SYOLO: An Efficient Pedestrian Detection

Zhen Wang¹, Longhua Ma²,*
¹College of Control Science and Engineering, Zhejiang University, Zhejiang, China
²Ningbo Institute of Technology, Zhejiang University, Ningbo, China

*Corresponding author e-mail: lhma_zju@zju.edu.cn

Abstract. Pedestrian detection is an important branch of object detection. It plays a vital role in many fields such as intelligent monitoring systems. The premise of the pedestrian recognition algorithm in the application of the industrial scene is accurate pedestrian detection. Our paper proposes a model to solve real-time pedestrian detection with high accuracy base on YOLO v3. We provide a method to select the size and number of anchor boxes for predicting bounding boxes accurately. Then we use a modified shuffle unit to lightweight the backbone of YOLO v3, which reduces the 67.3% FLOPs and 65.1% parameters. We train and validate our model on CrowdHuman detection dataset, SYOLO gets 62.7 mAP for face and 62.0 mAP person with 0.748 average IOU. Our network processes images in real-time at 185.8 FPS for network and 12.3 FPS for the entire model on CrowdHuman.

1. Introduction

The one-stage object detection [1, 2, 3, 4, 5] are aimed at quickly and accurately detect as many targets as possible. Pedestrian detection is similar to object detection because it has few categories, only face and human. Due to background clutter, camera viewpoints, overlaps, occlusions and other complex changes, this problem is challenging and has attracted a lot of attention in recent years.

Pedestrian detection also plays an important role in other areas, for example person re-identification. In [6], two separate tasks, pedestrian detection and person re-identification, are jointly handled in a single Convolutional Neural Network (CNN). In order to make the algorithm feasible in the industrial scenario, real-time pedestrian detection is inevitable. This paper proposes an efficient pedestrian detection model for person re-identification, therefore, the accuracy of bounding box detection is obviously more important, for example, in person re-identification, the Part-based Convolutional Baseline (PCB)[7] and Visibility-aware Part Model (VPM)[8], are both hopes that the photos of the two characters can be aligned as much as possible, especially for PCB, where unaligned images may result in poor recognition.

YOLOv3 is a state of art model for object detection, detecting different proportions of objects in a hierarchical manner in different scales of feature maps. YOLOv3 uses Darknet53 as the backbone to generate feature maps, which is redundant and time-consuming. Many existing works [9, 10, 11, 12] focus on pruning, compressing, and low-bit representing a basic network architecture. We are inspired by MobileNet v1 [13], MobileNet v2 [14], ShuffleNet v1 [15], and ShuffleNet v2 [16], and propose an uneven channel shuffle method based on Shufflenet v2 [14] to accelerate model and boost the effect. we propose a more reasonable way to choose the size and quantity of anchor boxes to get a higher intersection over union (IOU) score.
We evaluate our models on the CrowdHuman [17] person detection task. There are totally 470k individual persons in the train and validation subsets, and average 22.6 person in a single image. A series of controlled experiments show the effectiveness of our metric and the more efficient performance over original YOLO v3.

2. Related Work

2.1. Object Detection

State of the art object detection method can be divided into two groups: region proposal based methods (two stage) and proposal-free methods (one stage). Two stage methods include [18, 19, 20, 21], which have high detection accuracy and recall rate. One-stage methods like SSD [4], DSSD [22], YOLO v1 [1], YOLO v2 [2], YOLO v3 [3], RetinaNet [5] have a fast detection speed. Our proposed model is built upon the YOLOv3 [3] framework and thus it inherits the accuracy and speed advantage of YOLO v3[3].

2.2. Cluster analysis

The K-means algorithm is widely used to cluster unlabeled numeric data. Choosing the right "k" gives you a more reasonable number of clusters. The elbow method [23] is to plot the curves of various k and the corresponding mean variance, and then find the point where the marginal gain drops sharply and gives the angle in the graph. Another method is to evaluate the clustering effect by combining cohesion and resolution contour coefficients [24]. Our approach is to consider the combination of the above two algorithms to avoid the failure of some single algorithms in some cases.

2.3. Lightweight design

In the last few years, the increasing needs of running deep neural networks on embedded device make efficient model designs play an important role. SqueezeNet [25] reduces parameters and computation significantly. MobileNet v1 [13] and MobileNet v2 [14] make use of depthwise separable convolutions to light weight models. ShuffleNet v1 [15] and ShuffleNet v2 [16] investigate the effectiveness of channel shuffle and the influence of group convolution on memory access cost (MAC). Our work follows the advice given by ShuffleNet v2 [16] and makes use of channel shuffle in a novel form.

3. Method

3.1. Select anchor boxes

YOLO v3 predicts bounding boxes using prior anchor boxes. The network predicts \( t_w \), \( t_h \) for width and height of each bounding box. Rely on bounding box prior, \( p_w \), \( p_h \), to calculate the final predictions correspond to:

\[
    w = p_w e^{t_w} \\
    h = p_h e^{t_h}
\]  

(1)  (2)

The priors are critical for the network to learn to predict detections, and better priors for the network to start with can make it easier and get higher IOU scores. Our paper starts with two aspects, dimensions and quantity of anchor box, for optimization.

3.1.1. Pick box dimensions for each class. We run k-means clustering on training set of CrowdHuman [17] to automatically pick great priors. We still make use of IOU metric to calculate distance instead of Euclidean distance. The formula is as follows:

\[
    d_{iou}(box, centroid) = 1 - IOU(box, centroid)
\]  

(3)

There are only two classes, face and person, to detect in CrowdHuman [17], which is much less than in COCO. We filter out the bounding box belonging to each class, and cluster dimensions respectively. We choose k=5 for face and k = 9 for person, then choose 9 priors out of two clustering results (box_face, box_person) by hand. The largest one of box_face is not greater than median of box_person, we keep the second half of box_person and fusion the front five dimensions, which calculated as:
\[ s = w_1 s_1 + w_2 s_2 \]  
\[ w_i = \frac{n_i}{n_1 + n_2}, \quad i = 1, 2 \]

where \([s_1, s_2] \) is width pair or height pair of bounding boxes which have the minimum IOU distance, \(n_i\) is the boxes number belonging to the centroid \(s_i\).

Under the above criteria, our anchor boxes half are square-like boxes and half are tall thin boxes. We compare the average IOU to the closest prior of our picking strategy and the direct clustering anchor boxes in Table 1. Obviously, our method gets higher average IOU, and too much anchor boxes do not bring much, but causes a lot of computational costs.

**Table 1.** Compare the average IOU by different box generation.

| Box Generation         | #  | Avg IOU(face) | Avg IOU(person) | Avg IOU(total) |
|------------------------|----|---------------|-----------------|---------------|
| YOLO v3 tiny           | 6  | 0.698         | 0.685           | 0.690         |
| YOLO v3                | 9  | 0.745         | 0.725           | 0.735         |
| Direct Cluster         | 9  | 0.753         | 0.727           | 0.741         |
| Direct Cluster         | 15 | 0.755         | 0.729           | 0.743         |
| Cluster by Class + hand pick (our) | 9  | 0.756         | 0.729           | 0.744         |
| Cluster by Class + hand pick (our) | 12 | 0.760         | 0.731           | 0.746         |

### 3.1.2. Select the suitable number of anchor boxes.

We use the k-means to cluster dimensions of the bounding box, however, as the value of \(k\) increases, the average IOU distance of samples with the closest centroid inevitably decrease. The choice of \(k\) value also affects the results of the detection. A simple evaluation method is the Elbow method, we plot the curve of average IOU distance between bounding boxes and their closest centroid with various values of \(k\), see Figure1. It’s hard to find the ‘elbow point’, so we use the silhouette coefficient to assist in pick \(k\), although the computational overhead is very large, especially for nearly 70000 sample boxes. For fast calculations, we randomly sample the bounding boxes belonging to each centroid to no more than 2000. The result is as shown in Figure 1. YOLO v2 picks \(k=5\) and YOLO v3 picks \(k=9\), however, they are not elbow points, and the silhouette coefficient curve show the rationality with more cluster centroids (the closer the value of silhouette coefficient is to 1, the better the cluster performance). We pick \(k=12\) for our SYOLO, because it performs better on clusters and does not add too many parameters and calculations.

![Figure 1. Building blocks of ShuffleNet v2 [16] and this work](image-url)
3.2. Shuffle Unit

We propose a novel shuffle unit that take advantage of the channel shuffle operation. ShuffleNet v2 [16] is a state-of-the-art net architecture. In each block, half of the channels pass through the block and directly join the next block, as shown in Figure 2(a). Although this can be regarded as a kind of feature reuse, it reduces the nonlinearity and expression ability to some extent. In order to solve this problem, we use the channel clip unit after 1×1 group convolution layer, where the group is 2, shown as Figure 2(b). In this way, a small group will not increase too much MAC and still reduce FLOPs. For spatial down sampling, we fellow the original ShuffleNet v2 [16], seen as Figure 2(c). The experiment proves our points, in order to control the variables, we use the same structure and parameters as ShuffleNet v2 [16] and just modify the unit when the stride is 1, and train our model on ImageNet dataset, then compared with other state-of-the-art networks, see in Table 2. A smaller classification error rate is achieved with a slight increase in parameters and FLOPs.

![Building blocks of ShuffleNet v2 and this work](image)

**Figure 2.** Building blocks of ShuffleNet v2 and this work

| Model                  | FLOPs  | #Params | Top-1 error | Top-5 error |
|------------------------|--------|---------|-------------|-------------|
| MobileNet v2           | 300M   | 3.4M    | 28.0        | -           |
| ShuffleNet v1 1.5x (group=3) | 292M   | 3.4M    | 28.4        | 9.8         |
| ShuffleNet v1 2.0x (group=3) | 524M   | 5.4M    | 25.9        | 8.6         |
| ShuffleNet v2 1.5x     | 299M   | 3.5M    | 27.4        | 9.4         |
| our                    | 317M   | 3.6M    | 26.8        | 8.7         |

3.3. Lightweight

YOLO v3 [3] uses Darknet-53 for performing feature extraction. It is obviously a time-consuming process. We use our shuffle unit to lightweight the backbone, structure shown in Table 3. Stage2 to Stage6 are all stacks of shuffle unit. We use the unit with stride of 2 to down sampling at the beginning of each stage. We do end-to-end training for SYOLO on CrowdHuman [17]. Since the average number of pedestrians per image is 22.6, there will be a lot of boxes found out, so filtering bounding box with Non-Maximum Suppression (NMS) is time consuming. We compare the processing speeds from two aspects, fully convolutional neural network and entire model (with NMS)
respectively, seen as Table 4. Our backbone speeds up the calculation of the model without loss of accuracy.

Table 3. The overall architecture of our backbone

| layer   | Output | KSize | filter | Stride | Repeat |
|---------|--------|-------|--------|--------|--------|
| Conv0   | 416    | 3     | 24     | 1      | 1      |
| Conv1   | 208    | 3     | 32     | 2      | 1      |
| Stage2  | 208    | 32    | 1      | 2      |        |
| Stage3  | 104    | 64    | 2      | 1      |        |
|         | 104    | 128   | 1      | 4      |        |
| Stage4  | 52     | 256   | 2      | 1      |        |
|         | 52     | 256   | 1      | 6      |        |
| Stage5  | 26     | 512   | 2      | 1      |        |
|         | 26     | 512   | 1      | 6      |        |
| Stage6  | 13     | 1024  | 2      | 1      |        |
|         | 13     | 1024  | 1      | 4      |        |

Table 4. Comparison of several network architectures

| Model     | Anchor | FLOPs  | Param | FPS (network) | FPS (model) | mAP | Avg IOU |
|-----------|--------|--------|-------|---------------|-------------|-----|---------|
| YOLO v3 tiny | 6      | 2812M  | 8.8M  | 514.9         | 14.9        | 41.23| 0.690  |
| YOLO v3    | 9      | 33123M | 61.9M | 114.4         | 12.4        | 60.67| 0.735  |
| SYOLO (our)| 12     | 10821M | 21.6M | 185.8         | 12.3        | 62.44| 0.748  |

4. Conclusion

We introduce a real-time pedestrian detection system. We provide a method to pick a reasonable number of anchor boxes and more suitable dimensions for getting higher IOU scores in detection. Precise bounding box is conducive to identification in the field of pedestrian. The improved ShuffleNet v2 [16] unit is demonstrated competitive accuracy to original unit. We stack the unit as backbone of YOLO v3 [3] make it more efficient. Our future work will joint this real-time detection and identification for fast person search system.

References

[1] J. Redmon, S. Divvala, R. Girshick, et al. You only look once: Unified, real-time object detection. Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 779-788.
[2] J. Redmon, A. Farhadi. YOLO9000: better, faster, stronger. Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 7263-7271.
[3] J. Redmon, A. Farhadi. Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767, 2018.
[4] W. Liu, D. Anguelov, D. Erhan, et al. Ssd: Single shot multibox detector. European conference on computer vision. Springer, Cham, 2016: 21-37.
[5] T. Y. Lin, P. Goyal, R. Girshick, et al. Focal loss for dense object detection. Proceedings of the IEEE international conference on computer vision. 2017: 2980-2988.
[6] T. Xiao, S. Li, B. Wang, et al. Joint detection and identification feature learning for person search. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017: 3415-3424.
[7] W. Ouyang, X. Zeng, X. Wang, et al. DeepID-Net: Object detection with deformable part based convolutional neural networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016, 39(7): 1320-1334.
[8] Y. Sun, Q. Xu, Y. Li, et al. Perceive Where to Focus: Learning Visibility-aware Part-level Features for Partial Person Re-identification. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019: 393-402.

[9] M. Jaderberg, A. Vedaldi, A. Zisserman. Speeding up convolutional neural networks with low rank expansions[J]. arXiv preprint arXiv:1405.3866, 2014.

[10] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi. Xnornet: Imagenet classification using binary convolutional neural networks. In European Conference on Computer Vision, pages 525–542. Springer, 2016. 1, 2

[11] X. Zhang, J. Zou, K. He, and J. Sun. Accelerating very deep convolutional networks for classification and detection. IEEE transactions on pattern analysis and machine intelligence, 38(10):1943–1955, 2016. 1, 8

[12] X. Zhang, J. Zou, X. Ming, K. He, and J. Sun. Efficient and accurate approximations of nonlinear convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1984–1992, 2015. 1, 8

[13] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861, 2017. 1, 2, 3, 5, 6, 7

[14] M. Sandler, A. Howard, M. Zhu, et al. Mobilenetv2: Inverted residuals and linear bottlenecks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 4510–4520.

[15] X. Zhang, X. Zhou, M. Lin, et al. Shufflenet: An extremely efficient convolutional neural network for mobile devices. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 6848-6856.

[16] N. Ma, X. Zhang, H. Zheng, et al. Shufflenet v2: Practical guidelines for efficient cnn architecture design. Proceedings of the European Conference on Computer Vision (ECCV). 2018: 116-131.

[17] S. Shao, Z. Zhao, B. Li, et al. Crowdhuman: A benchmark for detecting human in a crowd. arXiv preprint arXiv:1805.00123, 2018.

[18] R. Girshick, J. Donahue, T. Darrell, et al. Rich feature hierarchies for accurate object detection and semantic segmentation. Proceedings of the IEEE conference on computer vision and pattern recognition. 2014: 580-587.

[19] R. Girshick. Fast r-cnn. Proceedings of the IEEE international conference on computer vision. 2015: 1440-1448.

[20] S. Ren, K. He, R. Girshick, et al. Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems. 2015: 91-99.

[21] K. He, G. Gkioxari, P. Dollár, et al. Mask r-cnn. Proceedings of the IEEE international conference on computer vision. 2017: 2961-2969.

[22] C. Fu, W. Liu, A. Ranga, et al. Dssd: Deconvolutional single shot detector. arXiv preprint arXiv:1701.06659, 2017.

[23] P. Bholowalia, A. Kumar. EBK-means: A clustering technique based on elbow method and k-means in WSN. International Journal of Computer Applications, 2014, 105(9).

[24] S. Aranganayagi, K. Thangavel. Clustering categorical data using silhouette coefficient as a relocating measure. International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007). IEEE, 2007, 2: 13-17.

[25] F. Iandola, S. Han, M. Moskewicz, et al. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size. arXiv preprint arXiv:1602.07360, 2016.