Article

The Application of Improved YOLO V3 in Multi-Scale Target Detection

Moran Ju ^{1,2,3,4,5,*} , Haibo Luo ^{1,2,4,5}, Zhongbo Wang ^{1,2,3,4,5}, Bin Hui ^{1,2,4,5} and Zheng Chang ^{1,2,4,5}

¹ Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China

² Institutes for Robotics and Intelligent Manufacturing, Chinese Academy of Sciences, Shenyang 110169, China

³ University of Chinese Academy of Sciences, Beijing 100049, China

⁴ Key Laboratory of Opto-Electronic Information Processing, Chinese Academy of Science, Shenyang 110016, China

⁵ The Key Lab of Image Understanding and Computer Vision, Shenyang 110016, China

* Correspondence: jumoran@sia.cn; Tel.: +86-176-7839-3828

Received: 8 August 2019; Accepted: 5 September 2019; Published: 9 September 2019

Abstract: Target detection is one of the most important research directions in computer vision. Recently, a variety of target detection algorithms have been proposed. Since the targets have varying sizes in a scene, it is essential to be able to detect the targets at different scales. To improve the detection performance of targets with different sizes, a multi-scale target detection algorithm was proposed involving improved YOLO (You Only Look Once) V3. The main contributions of our work include: (1) a mathematical derivation method based on Intersection over Union (IOU) was proposed to select the number and the aspect ratio dimensions of the candidate anchor boxes for each scale of the improved YOLO V3; (2) To further improve the detection performance of the network, the detection scales of YOLO V3 have been extended from 3 to 4 and the feature fusion target detection layer downsampled by 4× is established to detect the small targets; (3) To avoid gradient fading and enhance the reuse of the features, the six convolutional layers in front of the output detection layer are transformed into two residual units. The experimental results upon PASCAL VOC dataset and KITTI dataset show that the proposed method has obtained better performance than other state-of-the-art target detection algorithms.

Keywords: target detection; YOLO V3; anchor box; computer vision; deep learning

1. Introduction

Target detection is one of the research hotspots in the field of computer vision. The location and the category of the targets can be determined by using target detection. Nowadays, target detection has been applied in many fields such as military and civil areas, including image segmentation [1,2], intelligent surveillance [3–5], autonomous driving [6,7], and intelligent transportation [8,9].

With the rapid development of graphics processing unit (GPU) hardware, deep learning has made significant progress [10] and a lot of target detection algorithms based on deep learning have been used in our daily life [11], such as pedestrian detection [12], face detection [13–15], and vehicle detection [16,17]. The convolutional neural network (CNN) is a kind of network with many layers used to extract features based on invariance of regional statistics in images [18]. By training on the dataset, CNN can learn the features of the targets that need to be detected autonomously and the performance of the model can be improved gradually. With the improvement of computer hardware, the structure of CNN becomes much deeper and the features of the targets can be better learned. The conventional LeNet [19] consisted of five layers and the number of the layers in Highway Networks [20] and Residual

Networks [21] has surpassed 100. DenseNet [22] connects layers in a feed-forward fashion, which is useful for reusing the features and avoiding gradient fading.

The state-of-the-art target detection algorithms based on CNN can be divided into two categories. The first category is two stage target detection algorithms, such as R-CNN [2], Fast R-CNN [23], Faster R-CNN [24], Mask R-CNN [25], etc. The target detection process is divided into two phases by these algorithms. Firstly they use a Region Proposal Network (RPN) to generate a sparse of candidate anchor boxes and then the location and category of the candidate targets can be predicted and identified by the detection network. These algorithms can achieve perfect performance for target detection. However, they are not end-to-end target detection algorithms. The second category is one stage target detection algorithms, such as OverFeat [26], SSD [27], YOLO [28], YOLO 9000 [29], YOLO V3 [30], You Only Look Twice [31], etc. There is no need to use RPN to generate the candidate targets. These algorithms predict the target location and category information directly through the network. They are end-to-end target detection algorithms. Therefore, the speed of the one stage target detection algorithm is faster.

As a representative algorithm of one stage target detection, YOLO V3 [30] predicts targets of different size at 3 different scales, which uses a similar concept to feature pyramid networks [32]. Instead of choosing anchor boxes by hand, YOLO V3 runs K-means clustering on the dataset to find good priors automatically. However, it divides up the priors evenly across scales, which may cause some anchor boxes to be placed at inappropriate scales because the clusters are arbitrarily allocated.

In this paper, a multi-scale target detection approach based on YOLO V3 is proposed. To improve the detection performance of the targets with different sizes, we extend the detection scale of YOLO V3 from 3 to 4. To avoid gradient fading and enhance the reuse of the features, the six convolutional layers in front of the output detection of each scale are transformed into two residual units. To select appropriate size of the anchor boxes for each scale, a mathematical derivation method based on Intersection over Union (IOU) is proposed.

The main contributions of our work can be concluded as follows: (1) a mathematical derivation method based on Intersection over Union (IOU) was proposed to select the number and the aspect ratio dimensions of the candidate anchor boxes for each scale of the improved YOLO V3; (2) To further improve the detection performance of the network, the detection scales of YOLO V3 have been extended from 3 to 4 and the feature fusion target detection layer downsampled by $4\times$ is established to detect the small targets; (3) To avoid gradient fading and enhance the reuse of the features, the six convolutional layers in front of the output detection layer are transformed into two residual units; (4) We compared our approach with the state-of-the-art target detection algorithms both on the PASCAL VOC dataset and the KITTI dataset to evaluate the performance of the improved network. We also provide quantitative and qualitative evaluations between our algorithm and other target detection algorithms.

The rest of this paper is organized as follows. We introduce the framework of YOLO V3 in Section 2. In Section 3, we describe the details of our approach about the main framework of improved YOLO V3 network and the mathematical derivation method to select the appropriate candidate anchor boxes for each scale. The comparative experiments and results on PASCAL VOC dataset and KITTI dataset are presented in Section 4. Finally, the conclusion of this paper is drawn in Section 5.

2. Brief Introduction to YOLO V3

YOLO (You Only Look Once) is a one stage target detection algorithm, which has been developed into the third generation called YOLO V3. YOLO V3 is end-to-end target detection algorithm. It can predict the category and the location of the targets directly, so the detection speed of the YOLO V3 algorithm is fast.

To perform feature extraction, YOLO V3 uses successive 3×3 and 1×1 convolutional layers and draws on the idea of the Residual Networks [21]. There are 5 residual blocks in YOLO V3. Each residual block is composed of multiple residual units. The residual unit is shown in Figure 1. With the residual unit, the depth of the network can be deeper and the gradient fading can be avoided.

Figure 1. The structure of the Residual unit.

The input image is downsampled by five times. YOLO V3 predicts the targets in the last 3 downsampled layers. There are 3 scales for YOLO V3 to detect the targets. At scale 3, the feature map downsampled by $8\times$ is used to detect small targets. At scale 2, the feature map downsampled by $16\times$ is used to detect medium-sized targets. At scale 1, the feature map downsampled by $32\times$ is used to detect big targets. Feature fusion is used to detect targets because small feature map provides deep semantic information and large feature map provides more finer-grained information of the targets. To perform feature fusion, YOLO V3 resizes the feature maps of the deeper layer by upsampling. Then the feature maps at different scales will have the same size. YOLO V3 merges the features from the earlier layer with the features from the deeper layer by concatenation. So YOLO V3 has good performance to detect both large and small targets. The network of YOLO V3 is shown in Figure 2.

Figure 2. The network of YOLO V3.

3. Our Approach

We ran a K-means clustering algorithm to perform clustering analysis on the PASCAL VOC dataset and KITTI dataset. Then we proposed a mathematical derivation method based on IOU to determine the number and the aspect ratio dimensions of candidate anchor box for each scale of the improved network. Finally, to enhance the detection performance of the YOLO V3, we improved the structure of YOLO V3.

3.1. Appropriate Size for Anchor Boxes

YOLO V3 introduced the idea of anchor boxes used in Faster R-CNN. Anchor boxes are a set of initial candidate boxes with a fixed width and height. The choice of the initial anchor boxes will directly affect the detection accuracy and the detection speed. Instead of choosing anchor boxes by hand, YOLO V3 runs K-means clustering on the dataset to find good priors automatically. The clusters generated by K-means can reflect the distribution of the samples in each dataset, which can make it easier for the network to get good predictions. In this paper, Avg IOU is used as a metric of target clustering analysis. The objective function of clustering Avg IOU is as follows:

$$\operatorname{argmax} \frac{\sum_{i=1}^k \sum_{j=1}^{n_k} \operatorname{IOU}(\text{box}, \text{centroid})}{n}, \quad (1)$$

where *box* is the sample, namely the ground truth of the target. *centroid* is the center of the cluster. n_k is the numbers of samples in the k th cluster center. k is the total number of samples. n is the numbers of clusters. $\operatorname{IOU}(\text{box}, \text{centroid})$ is the intersection over Union of the clusters and the sample.

We applied K-means clustering on the PASCAL VOC and KITTI dataset, respectively. The Figure 3 shows the average IOU we got with different value of k . With the increase of k , the change of the objective function became more and more stable. Considering Avg IOU and the number of detection layers, we selected 12 anchor boxes. The width and the height of the corresponding clusters on PASCAL VOC dataset and KITTI dataset are shown in Table 1.

Table 1. The corresponding clusters on PASCAL VOC and KITTI dataset.

Dataset	PASCAL VOC	KITTI
Clusters	(22, 23), (29, 39), (37, 43), (59, 59), (53, 89), (81, 98), (119, 117), (128, 163), (173, 185), (248, 229), (308, 231), (428, 320)	(21, 22), (32, 24), (26, 40), (45, 40), (63, 32), (62, 56), (73, 60), (106, 57), (92, 116), (131, 118), (200, 130), (316, 181)

Figure 3. K-means clustering analysis result. (a) KITTI dataset; (b) PASCAL VOC.

After K-means clustering on the dataset, cluster centers are generated. YOLO V3 divides up the clusters evenly across scales. This may cause some clusters to be placed at inappropriate scales because the clusters are arbitrarily allocated. It is essential to determine what size of the anchor box is suitable for each scale of the network. Inspired by the method of the proposal generation [33], we used mathematical derivation based on IOU to help select the appropriate size of the anchor boxes for each scale.

There are two extreme cases, as shown in Figure 4. The red box represents the anchor box and the black box represents the ground truth box, and the green box is the grid cell of the feature map. If S_a is the side length of an anchor box and S_g is the side length of the ground truth box, and 2^d is the side length of the grid cell in the feature map. d is the numbers of downsampling layers.

Figure 4. Two extreme cases between anchor box and ground truth box. (a) The ground truth box is right in the anchor box (b) The center of the anchor box is in the upper left corner of the grid cell and the center of the ground truth box is in the bottom right corner of the grid cell.

Consider the extreme case in Figure 4a: We assume the anchor boxes and the ground truth boxes are quadrate and the anchor box is bigger than the ground truth box ($S_a \geq S_g$). The IOU between the anchor box and the ground truth box can be defined as

$$IOU(anchor, GT) = \frac{S_g^2}{S_a^2} = \lambda \quad (2)$$

where GT is ground truth box. The common metric to decide the prediction results is to see if the IOU is greater than 0.5 ($\lambda \geq 0.5$). Then we can get the result as follows

$$\sqrt{2}S_g \geq S_a \geq S_g \quad (3)$$

Consider the extreme case in Figure 4b: The center of the anchor box is in the upper left corner of the grid cell of the feature map and the center of the ground truth box is in the bottom right corner of the grid cell. We assume that half the side length of the anchor boxes and the ground truth box is longer than the length of the grid cell in the feature map.

$$\frac{1}{2}S_a \geq 2^d \quad (4)$$

$$\frac{1}{2}S_g \geq 2^d \quad (5)$$

The IOU between the anchor box and the ground truth box can be expressed by (6)

$$IOU(anchor, GT) = \frac{(\frac{S_a}{2} - 2^d + \frac{S_g}{2})^2}{S_g^2 + S_a^2 - (\frac{S_a}{2} - 2^d + \frac{S_g}{2})^2} \quad (6)$$

$$\frac{(\frac{S_a}{2} - 2^d + \frac{S_g}{2})^2}{S_g^2 + S_a^2 - (\frac{S_a}{2} - 2^d + \frac{S_g}{2})^2} \geq \frac{(\sqrt{S_a S_g} - 2^d)^2}{S_g^2 + S_a^2 - (\sqrt{S_a S_g} - 2^d)^2} \quad (7)$$

when $S_a = S_g$, the equal sign in the inequality (7) is true. From (3), the IOU between anchor box and the ground truth box in Figure 4a can be 1. We hope the IOU in (6) is greater than 0.5 so we can get the inequality to appear as follows

$$\frac{(S_a - 2^d)^2}{2S_a^2 - (S_a - 2^d)^2} \geq \frac{1}{2}. \quad (8)$$

Then we can get the side length S_a and the area of an anchor box with the numbers of the downsampling d . This is shown in Table 2.

Table 2. The relationship between the numbers of downsampling and the size of the anchor box.

d	$S_a \geq$	$S_a^2 \geq$
2	21.8	476
3	43.6	1901
4	87.2	7604
5	174.4	30,416

With the results in Table 1, we got the appropriate size and area of the anchor boxes for each scale of the output detection layer. However, the clusters generated by K-means on each dataset are not quadrate. Like the cluster (63, 32) generated by K-means on KITTI dataset, the height of the anchor box is much longer than that of the width. What is more, the height of the anchor meet the demand of output detection layer which is downsampled by 4× and the width of the anchor box meet the demand of the detection layer, which is downsampled by 8× according to Table 1. To solve this problem, suppose the height and the width of the anchor box are h and w , we compare the value of $\sqrt{h \times w}$ with S_a to determine which scale is suitable for this anchor. The cluster (63, 32) is suitable for the output detection layer which is downsampled by 8× ($43.6 \leq \sqrt{63 \times 32} \leq 87.2$).

So the principle to select suitable anchor boxes for each scale of the output detection layer can be concluded as follows:

- Apply the K-means clustering to get the candidate anchor boxes on each dataset.
- Calculate the value of $\sqrt{h \times w}$ for each candidate anchor box.
- Compare $\sqrt{h \times w}$ with S_a in Table 1 and select the appropriate scale for each anchor box.

According to the principle above, we can allocate the clusters on PASCAL VOC dataset to each suitable scale. This is shown in Table 3.

Table 3. The allocation of clusters on PASCAL VOC dataset at each scale.

d	Range	Anchor Boxes
2	$21.8 \leq \sqrt{h \times w} \leq 43.6$	(22, 23), (29, 39), (37, 43)
3	$43.6 \leq \sqrt{h \times w} \leq 87.2$	(59, 59), (53, 89)
4	$87.2 \leq \sqrt{h \times w} \leq 174.4$	(81, 98), (119, 117), (128, 163)
5	$174.4 \leq \sqrt{h \times w}$	(173, 185), (248, 229), (308, 231), (428, 320)

The allocation of each cluster on KITTI dataset can be shown in Table 4.

Table 4. The allocation of clusters on KITTI dataset at each scale.

d	Range	Anchor Boxes
2	$21.8 \leq \sqrt{h \times w} \leq 43.6$	(21, 22), (32, 24), (26, 40), (45, 40)
3	$43.6 \leq \sqrt{h \times w} \leq 87.2$	(63, 32), (62, 56), (73, 60), (106, 57)
4	$87.2 \leq \sqrt{h \times w} \leq 174.4$	(131, 118), (92, 116), (200, 130)
5	$174.4 \leq \sqrt{h \times w}$	(316, 181)

3.2. Improving YOLO V3 with More Scales

The network of improved YOLO V3 is shown in Figure 5. We add scale 4 (red box) to improve the detection performance of the network.

Figure 5. The network of improved YOLO V3.

As shown in Figure 5, to improve the performance of YOLO V3, we proposed an enhanced object detection algorithm. YOLO V3 uses three scales to detect targets of different size and detects small targets with the feature map downsampled by 8×. To get more fine-grained features and location information from small targets, we explore to use the feature map downsampled by 4× in the original network, because it contains more finer-grained information of small targets. To concatenate the feature map in the earlier layer to the feature map in the deeper layer, upsample the feature map

downsampled by $8\times$ and concatenate it with the output of the second residual block in YOLO V3. The feature fusion target detection layer downsampled by $4\times$ is established to detect small targets. Three detection scales in the original YOLO V3 are extended to four detection scales. So the improved YOLO V3 has better performance to detect targets of different size.

In front of the target detection layer of the YOLO V3 network, there are six convolutional layers. Inspired by DSSD [34], to avoid gradient fading and enhance the reuse of the features, we transformed the six convolutional layers into two residual units. This is shown in Figure 6.

Figure 6. Output structure of each network. (a) YOLO V3; (b) Improved YOLO V3.

4. Experiments and Results

To evaluate the performance of the improved network and the proposed method, the comparative experiments of our approach with other target detection algorithms were done on the PASCAL VOC dataset and KITTI dataset, respectively. The experimental condition is as follows. The operating system: Ubuntu 14.04. Deep learning framework: Darknet. CPU: i7-5930k. Memory: 64GB, GPU: NVIDIA GeForce GTX TITAN X.

4.1. Experiment on PASCAL VOC

We combined the PASCAL VOC 2007 training set and the PASCAL VOC 2012 training set as the training set of the improved network and the PASCAL VOC 2007 test set was taken as the test set. We evaluated with the commonly used performance metric in object detection if IOU overlaps with the ground truth bounding box is greater than 0.5. In the training stage, the initial learning rate is 0.001, and the weight decay is 0.0005. When the training batches were 60,000 and 70,000, the learning rate was reduced to 0.0001 and 0.00001, respectively, which makes the loss function further converge. To improve the performance of training, we use data augmentation methods such as rotating the images with different angles and changing the saturation, exposure, and hue of the images.

The improved YOLO V3 network was used to detect twenty categories of targets in the dataset. Calculate the average accuracy of each kind of target and calculate mAP of twenty categories of targets. The test results of different target detection algorithms are shown in Table 5. The results of YOLO [28], YOLO V2 [29], Faster RCNN [24], SSD [27], and DSSD [34] were referred to in their homepages or corresponding papers. The mAP of our proposed network for the 416×416 and 512×512 input models was 82.88% and 86.73%, respectively. Compared with the state-of-the-art object detection algorithms, the mAP was enhanced by the improved network obviously. The mAP has been improved by 3.62% and 7.47% over YOLO V3.

Table 5. Results of the state-of-the-art algorithms in PASCAL VOC data.

Algorithm	YOLO	YOLOV2	Faster RCNN	SSD	DSSD	YOLO V3	Ours	Ours
Input	448	416		512	513	416	416	512
mAP	66.4%	76.8%	73.2%	79.8%	81.5%	79.26%	82.88%	86.73%

The detection results of the improved YOLO V3 network on the PASCAL VOC dataset are shown in Figure 7. The dark blue box is the ground truth and the green box is the prediction box. It is clear that both the big targets and the small targets can be detected by the improved YOLO V3 network.

Figure 7. Test results of improved YOLO V3 on the PASCAL VOC dataset.

4.2. Experiment on KITTI Dataset

We used 2012 2D left color images of object data in KITTI dataset to train and test our model. We combined 8 categories of targets into 3 categories, namely the ‘car’, ‘cyclist’, and ‘person’. In the training stage, the initial learning rate is 0.001, and the weight decay is 0.0005. When the training batches were 50,000 and 55,000, the learning rate was reduced to 0.0001 and 0.00001, respectively, which makes the loss function further converge. We use data augmentation by rotating the images and shifting the color.

The mAP comparison of each algorithm on KITTI dataset is shown in Table 6. Compared with the state-of-the-art object detection algorithms, the mAP was enhanced by the improved network. The mAP of our proposed network for 512×512 input models was 84.72%. Our proposed network has improved the mAP by 1.77% over the conventional YOLO V3. The average accuracy of each kind of target and mAP of three categories of targets are shown in Table 7. The AP of each category has been improved by our proposed approach.

Table 6. Results of each algorithm on the KITTI dataset.

Object Detection Algorithm	SSD	Improved SSD [35]	YOLO V3	Ours
Input	512	512	512	512
mAp	62.8%	68.1%	82.95%	84.72%

Table 7. The average precision of 3-classon the KITTI dataset by YOLO V3 and improved YOLO V3.

Detection Algorithm	AP/%			mAp/%
	Car	Person	Cyclist	
YOLO V3	93.41	71.84	83.6	82.95
Ours	93.39	76.76	84	84.72

The detection results of the improved YOLO V3 network on the KITTI dataset are shown in Figure 8. The dark blue box is the ground truth and the green box is the prediction box. It is clear that both the big targets and the small targets can be detected by the improved YOLO V3 network.

Figure 8. Results of improved YOLO V3 on the KITTI dataset.

4.3. Experimenton VEDAI Dataset

To evaluate the performance of our network for detecting small targets, we do comparative experiments on VEDAI dataset. VEDAI is a dataset for vehicle detection in Aerial Imagery [36]. There is an average of 5.5 vehicles per image, and they occupy about 0.7% of the total pixels of the image. It is a typical small vehicle dataset. The experimental results show that YOLO V3 achieved a mAP of 55.81% and our network achieved 62.36%. Our network improved the mAP greatly by 6.55% which clearly demonstrates that the performance of our network for detecting small targets is improved.

4.4. Quantitative and Qualitative Evaluation

Quantitative evaluation: In Tables 4 and 5, the experimental results on PASCAL VOC dataset and KITTI dataset show that our network achieves better performance than other state-of-the-art detection algorithms. Compared with YOLO V3, our proposed network has improved mAP by 7.47% on the PASCAL VOC dataset and by 1.77% on the KITTI dataset. The performance is much better for the PASCAL VOC dataset than for the KITTI dataset. That is because PASCAL VOC contains 20 categories and every category may have small targets. However, there are 3 categories, namely ‘car’, ‘cyclist’, and ‘person’ in the KITTI dataset. Compared with PASCAL VOC, there are less small targets in the KITTI dataset. By extending the scales from 3 to 4, our proposed network is better than YOLO V3, especially for detecting small targets. What is more, our proposed mathematical derivation method based on IOU can select appropriate size of the anchor boxes for each scale of the output detection layer. The comparative results of model size and the frames per second (FPS) between YOLO V3 and improved YOLO V3 are shown in Table 8. The model size of our network is 249.1 M, which is slightly than that of YOLO V3. The FPS of our network is less than that of YOLO V3. That is because our proposed network extended the scales of the YOLO V3 from 3 to 4.

Table 8. The comparative results of FPS and model size with YOLO V3 and improved YOLO V3.

Detection Algorithm	Model Size	FPS
YOLO V3	246.7 M	23
Ours	249.1 M	19

Qualitative evaluation: Figures 7 and 8 shows that our proposed network has perfect performance for detecting both big and small targets. A qualitative comparison of our network with YOLOV3 is also provided. The comparative results between YOLO V3 and improved YOLO V3 on KITTI dataset and PASCAL VOC dataset are shown in Figures 9 and 10. The light blue box is the ground truth. The green box is the prediction box. The red box shows the targets with fault. It is obvious that our proposed network achieved a better performance than YOLO V3. In Figures 9 and 10, a lot of targets are missed and detected with fault by YOLO V3, especially some small targets. The improved network can avoid these problems. That is because our network has 4 scales to detect the targets. The feature fusion target detection layer downsampled by 4× is helpful to detect small targets. What is more, with the mathematical derivation method based on IOU, targets of different sizes can be accurately assigned to corresponding scales.

Figure 9. Comparison of each network on the KITTI dataset. (a) YOLO V3; (b) Improved YOLO V3.

Figure 10. Comparison of each network on the PASCAL VOC dataset. (a) YOLO V3; (b) Improved YOLO V3.

5. Conclusions

In this paper, a multi-scale target detection approach based on YOLO V3 is proposed. The main contributions of the proposed network are as follows: First, a mathematical derivation method based on IOU was proposed to select the number and the aspect ratio dimensions of the candidate anchor boxes for each scale of the improved YOLO V3. Second, to further improve the detection performance of the network, the detection scales of YOLO V3 have been extended from 3 to 4 and the feature fusion target detection layer downsampled by 4× was established to detect the small targets. To avoid gradient fading and enhance the reuse of the features, the six convolutional layers in front of the output detection layer were transformed into two residual units. Finally, we did comparative experiments on the PASCAL VOC dataset and the KITTI dataset. Compared with the state-of-the-art target detection algorithms, our experimental results show that the mean average precision is enhanced by the improved network and show that the proposed method is a valid one to select the suitable anchors for each scale. We also provided quantitative and qualitative comparisons of our approach with YOLO V3. Ways to simplify the network and reduce the computational cost without reducing the detection performance will be our main future research direction.

Author Contributions: M.J. contributed to this work by setting up the experimental environment, designing the algorithms, designing, performing the experiments, analyzing the data and writing the paper. H.L., Z.W., Z.C. and B.H. contributed through research supervisory and reviewer roles and by amending the paper.

Funding: This research received no external funding.

Conflicts of Interest: We declare that we have no conflict of interest.

References

1. Van de Sande, K.E.A.; Uijlings, J.R.R.; Gevers, T.; Smeulders, A.W.M. Segmentation as selective search for object recognition. In Proceedings of the IEEE International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 1879–1886.
2. Girshick, R.; Donahue, J.; Darrelland, T.; Malik, J. Rich feature hierarchies for object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 24–27 June 2014.

3. Al-Nawashi, M.; Al-Hazaimah, O.M.; Saraee, M. A novel framework for intelligent surveillance system based on abnormal human activity detection in academic environments. *Neural Comput. Appl.* **2017**, *28*, 565–572. [[CrossRef](#)] [[PubMed](#)]
4. Nguyen-Meidine, L.T.; Granger, E.; Kiran, M.; Blais-Morin, L.A. A Comparison of CNN-based Face and Head Detectors for Real-Time Video Surveillance Applications. *arXiv* **2018**, arXiv:1809.03336.
5. Yu, R.; Wang, H.; Davis, L.S. RemoteNet: Efficient Relevant Motion Event Detection for Large-scale Home Surveillance Videos. *arXiv* **2018**, arXiv:1801.02031.
6. Chen, X.; Ma, H.; Wan, J.; Li, B.; Xia, T. Multi-view 3d object detection network for autonomous driving. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.
7. Uçar, A.; Demir, Y.; Güzeli, C. Object recognition and detection with deep learning for autonomous driving applications. *Simulation* **2017**, *93*, 759–769. [[CrossRef](#)]
8. Jin, X.; Davis, C.H. Vehicle detection from high-resolution satellite imagery using morphological shared-weight neural networks. *Image Vis. Comput.* **2007**, *25*, 1422–1431. [[CrossRef](#)]
9. Wang, L.; Lu, Y.; Wang, H.; Zheng, Y.; Ye, H.; Xue, X. Evolving Boxes for Fast Vehicle Detection. *arXiv* **2016**, arXiv:1702.00254.
10. Lecun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [[CrossRef](#)] [[PubMed](#)]
11. Liu, L.; Ouyang, W.; Wang, X.; Fieguth, P.; Chen, J.; Liu, X.; Pietikäinen, M. Deep learning for generic object detection: A survey. *arXiv* **2018**, arXiv:1809.02165.
12. Tian, Y.; Ping, L.; Wang, X.; Tang, X. Pedestrian detection aided by deep learning semantic tasks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 8–10 June 2015.
13. Levi, G.; Hassner, T. Age and gender classification using convolution neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 8–10 June 2015.
14. Hu, P.; Ramanan, D. Finding Tiny Faces. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 22–25 July 2017.
15. Attia, A.; Dayan, S. Detecting and counting tiny faces. *arXiv* **2018**, arXiv:1801.06504.
16. Zhou, Y.; Liu, L.; Shao, L.; Mellor, M. DAVE: A Unified Framework for Fast Vehicle Detection and Annotation. *arXiv* **2016**, arXiv:1607.04564.
17. Liu, K.; Mattyus, G. Fast multi class vehicle detection on aerial images. *IEEE Geosci. Remote Sens. Lett.* **2015**, *12*, 1938–1942.
18. Li, J.; Zhou, F.; Ye, T. Real-world railway traffic detection based on faster better network. *IEEE Access* **2018**, *6*, 68730–68739. [[CrossRef](#)]
19. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [[CrossRef](#)]
20. Srivastava, R.K.; Greff, K.; Schmidhuber, J. Training very deep networks. *arXiv* **2015**, arXiv:1507.06228.
21. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
22. Gao, H.; Zhuang, L.; Maaten, L.V.D.; Weinberger, K.Q. Densely Connected Convolutional Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 22–25 July 2017.
23. Girshick, R. Fast R-CNN. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 8–10 June 2015; pp. 1440–1448.
24. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *36*, 1137–1149. [[CrossRef](#)] [[PubMed](#)]
25. He, K.; Gkioxari, G.; Dollar, P.; Girshick, R. Mask R-CNN. In Proceedings of the International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
26. Sermanet, P.; Eigen, D.; Zhang, X.; Mathieu, M.; Fergus, R.; Lecun, Y. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv* **2013**, arXiv:1312.6229.
27. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y. SSD: Single shot multi Box detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp. 21–37.

28. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
29. Redmon, J.; Farhadi, A. YOLO 9000: Better, Faster, Stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 22–25 July 2017; pp. 6517–6525.
30. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. Available online: <https://arxiv.org/abs/1804.02767> (accessed on 8 August 2019).
31. Van Etten, A. You only Look Twice: Rapid Multi-Scale Object Detection in Satellite Imagery. Available online: <https://arxiv.org/abs/1805.09512> (accessed on 8 August 2019).
32. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 22–25 July 2017; pp. 2117–2125.
33. Eggert, C.; Zecha, D.; Brehm, S.; Lienhart, R. Improving Small Object Proposals for Company Logo Detection. In Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval, Bucharest, Romania, 6–9 June 2017; pp. 167–174.
34. Fu, C.; Liu, W.; Ranga, A.; Tyagi, A.; Berg, A.C. DSSD: Deconvolutional Single Shot Detector. Available online: <https://arxiv.org/abs/1701.06659> (accessed on 8 August 2019).
35. Wu, T.; Zhang, Z.; Liu, Y.; Pei, W.; Chen, H. A light weight small object detection algorithm based on improved SSD. *Infrared Laser Eng.* **2018**, *47*, 0703005. [[CrossRef](#)]
36. Razakarivony, S.; Jurie, F. Vehicle detection in aerial imagery: A small target detection benchmark. *J. Vis. Commun. Image Represent.* **2016**, *34*, 187–203. [[CrossRef](#)]

© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).