2018 IEEE Conference on Multimedia Information Processing and Retrieval

Vehicle Detection in UAV Traffic Video Based on Convolution Neural Network

Shulin Li[1], Weigang Zhang[2,3*], Guorong Li[3], Li Su[3*], Qingming Huang[3]
[1]School of Computer Science, Beijing University of Posts and Telecommunication, Beijing, China
[2]School of Computer Science and Technology, Harbin Institute of Technology, Weihai, China
[3]School of Computer and Control Engineering, University of Chinese Academy of Sciences, Beijing, China
shulin.li@vipl.ict.ac.cn, wgzhang@hit.edu.cn, {liguorong, suli, qmhuang}@ucas.ac.cn

***Abstract*—Vehicle detection technology is a key component of an intelligent transportation system, but most of the current vehicle detection technologies are based on road monitoring cameras. Compared with these fixed cameras, Unmanned Aerial Vehicles (UAVs) seem to have a lot of advantages such as more flexible, broader vision, higher speed, which make the vehicle detection more challenging. In this paper, a new dataset built on UAV traffic videos and a neural network which could fuse multi-layer features are proposed. Different from some networks with only a single layer, the proposed network merges the features from multiple layers firstly. Then a convolution layer is used to reduce the feature dimensions and a deconvolution layer is employed to do upsampling and enhance the response information. Finally, multiple fully connected layers are used to finish the detection. Furthermore, the proposed method combines the detecting and tracking for optimization and high detection speed. Experiments on the self-built UAV traffic video dataset demonstrate that the proposed method gets better results and higher speed.**

Keywords-vehicle detection; unmanned aerial vehicle (UAV); convolution neural network; traffic video

I. Introduction

Intelligent transportation systems are growingly important because the traffic congestions and traffic accidents are becoming increasingly serious. Vehicle detection is the key technology in intelligent transportation system and is also a specific task of object detection in the field of computer vision. The target of vehicle detection is getting all vehicles in the videos or images. It can be used in traffic flow calculation, traffic congestion forecast and so on. A lot of vehicle detection methods have been proposed, but these approaches are mostly based on the videos captured by the fixed road traffic cameras. Fixed camera has many drawbacks, such as high cost to install, not flexible enough to do vehicle detection, difficult to follow up and so on. Thus, vehicle detection technologies based on UAV aerial videos have received widespread concern. Compared to fixed cameras, UAVs have many advantages, such as flexibility, mobility, portability, etc.

UAV traffic videos have the following characteristics: 1) wider vision and more detection targets; 2) target size varies widely due to the flying altitude changes of UAV; 3) smaller target size and less description details; 4) vehicle orientation is more changeable; 5) surrounding environment is more complex. Therefore, there are more difficulties and challenges for vehicle detection in UAV traffic video. This task has attracted lots of research attention.

II. Related Work

The traditional vehicle detection technology is mainly including interframe difference, background difference, optical flow and some classifiers based on histogram of oriented gradient (HOG) [1], Haar-like [2] or other vision features, such as Support Vector Machine (SVM), Ada-Boost, etc. In recent years, deep learning has made breakthroughs in lots of computer vision areas, including object detection task. R-CNN [3] divides the object detection task into two parts, generates region proposals using selective search [4] firstly, then does classification and box regression for every region proposal. But its speed is limited by repeated CNN evaluation for region proposal generation. SPP-Net [5] proposed spatial pyramid pooling (SPP), which would allow the computation of CNN features once per image and handle different sizes of images without beforehand cropping or wrapping. Building on SPP-Net, Fast-RCNN [6] proposed ROI pooling layer and multi-task learning to improve detection speed and effect. However, it still depended on some algorithms to generate region proposals. Faster-RCNN [7] uses a single neural network to generate region proposals for detecting, which is named RPN and leads to a significant speedup. RFCN [8] achieved translation-invariance by position-sensitive score maps. YOLO [9] thinks the problem from another way and considers the detection as a regression task, which no longer needs to generate region proposals. This method results in a significant speedup, but with some compromise of detection accuracy. Based on YOLO, SSD [10] removes all fully-connected (FC) layers and learns from RPN's anchor mechanism. The detection speed and effect are improved.

In the DeepProposal [11], the authors draw a conclusion through the experiments: "the final convolutional layers can find the object of interest with high recall but poor localization due to the coarseness of the feature maps. Instead, the first layers of the network can better localize the object of interest but with a reduced recall". GoogLeNet [12] uses three weighted classification losses and applies intermediate layers to show that this type of regularization is useful for very deep models. The same idea is employed to semantic segmentation [13] and edge detection [14]. For object detection, MSCNN [15] trained multiple independent detectors at different layers to improve the detection effect of

[*] corresponding author

0-7695-6354-6/18/$31.00 ©2018 IEEE
DOI 10.1109/MIPR.2018.00009

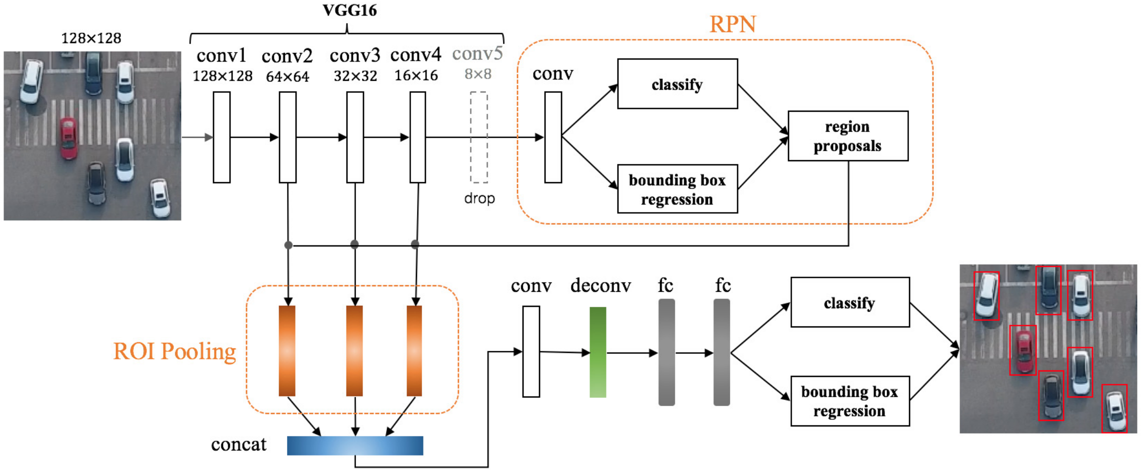

Figure 1. The network architecture for vehicle detection in UAV: (1) input a video frame, (2)extract features, (3) generate region proposals, (4) fuse multi-layer features, (5) classify and make adjustment for each region.

small targets. A good detection network should combine the advantages of both shallow and deep layers. Thus, we propose to integrate the features from different network layers to enhance the overall detection effect.

III. Vehicle detection

As shown in Fig.1, the proposed vehicle detection framework for UAV traffic video is illustrated. Initially, an entire video frame is forwarded through the convolutional layers and the feature maps are produced on multi-layers. The feature map of the last layer is used to generate region proposals. Then, a ROI Pooling layer is applied on the three layers: conv2, conv3 and conv4 for every region proposal and aggregates together by a concat layer. Next, a convolution layer is used to reduce the feature dimensions and a deconvolution layer is used to enhance the response information. Finally, these proposals are classified and adjusted based on the detection module.

A. Backbone Architecture

The proposed network is built on VGG-16 [16]. Other networks [12,17,18,19] are also applicable. VGG-16 has 13 convolution layers followed by 3 FC layers. These 13 convolution layers are divided into 5 groups which are corresponding to the five layers of AlexNet [17] by pooling layers. We remove the third FC layer and the last group of convolution layers. Because most targets are relatively small in the UAV traffic video dataset, the higher layers respond very weakly to small targets. For example, a 32×32 target is mapped into a 2×2 patch at the conv5 layer. The information provided to 7×7 ROI pooling is very limited. Thus, we retain the first four groups of convolution layers. The VGG-16 model is pre-trained on the ImageNet [20].

B. Multi-level Feature Fusion

The features from different CNN levels are complementary for object detection. Thus, we apply the multi-level feature fusion to improve vehicle detection results. One fusion scheme is to combine multi-level feature maps at the same resolution. HyperNet [21] uses a max pooling layer on layer-1 to do subsampling and a deconvolution layer on layer-5 to conduct upsampling. Finally, the three feature maps from layer-1, layer-3 and layer-5 are concatenated to output a single cube which called hyper feature. The main problem to be solved for multi-level feature fusion is that the resolutions of the obtained feature maps are usually not the same. Therefore, another improved scheme is that adding a ROI pooling layer on each layer for feature fusion. The resolutions of the multiple feature maps from different layers are same and they are concatenated. Different from the HyperNet, we fuse the feature maps of layer-2, layer-3 and layer-4 and the experiments show that the fusion can obtain the best performance. Compare to the first scheme, the improved scheme is easier to be implemented and more scalable. When to combine more layer features, no major changes will be made for the proposed network. In addition, there is no need to use subsampling or upsampling to keep the resolutions of the feature maps the same. Thus, it is not necessary to worry that the pre-trained features will be affected.

After feature combination, a 3×3 convolution layer is followed. The Conv operation not only reduces feature dimension but also makes feature fusion better. Besides, we add a deconvolution layer to increase the resolution of feature maps. The experiments show that this improvement can significantly improve the performance of object detection, especially for small targets.

C. Data Preprocessing

In addition to the vehicles on the road, there are many cars in the roadside parking lots and the parking density is usually high. Thus, to avoid affecting the detection performance of the proposed network, these vehicles in the parking lots and some too small cars will be masked firstly.

To obtain more training data, it needs to expand the UAV traffic video dataset in the following manners: 1) horizontal rotation and 90 degrees clockwise rotation to enrich the vehicle orientation; 2) scale down the video frame to 1/2, 1/3

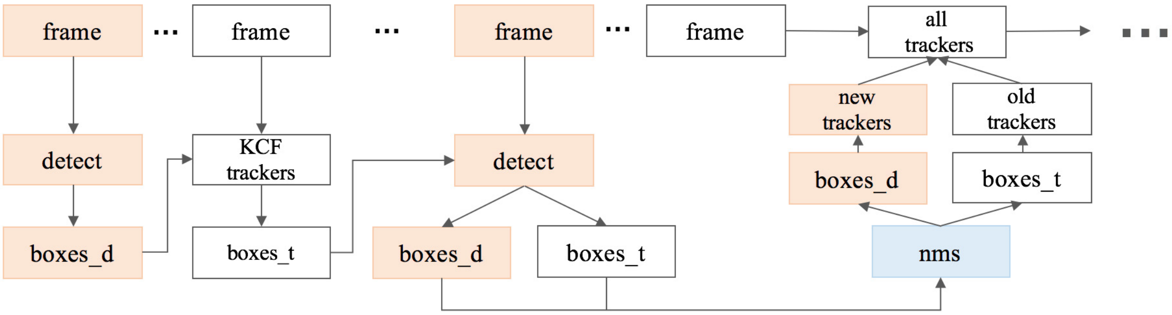

Figure 2. Combining tetection and tracking. The boxes_d and boxes_t represent the region proposals generated from the RPN and KCF separately.

of the original image, which can significantly improve the detection performance for small objects; 3) change the brightness and clarity of the video frame to simulation the environment of the UAV videos.

D. *Network Training*

For each region proposal generated from the RPN, a binary class label is assigned: vehicle or not vehicle. The positive label is assigned to the proposal which has a 0.45 or more IoU threshold with any ground truth bounding box. The proposals whose IoU threshold is lower than 0.3 with all ground truth bounding boxes are assigned to the negative label. The training aim is to minimize the multi-task loss function:

$$L(k,k^*,t,t^*) = L_{cls}(k,k^*) + \lambda L_{reg}(t,t^*) \qquad (1)$$

where the first part L_{cls} is the softmax classification loss of two classes and the second part L_{reg} is the bounding box regression loss only for positive boxes. k^* and k are the ground-truth and prediction separately. We set $\lambda=3$ in the RPN stage to get more accurate region proposals. In the detection stage, we set the same weight for classification and box regression. $t=(t_x,t_y,t_w,t_h)$, where (t_x,t_y) is the upper left corner of the coordinate t and t_w, t_h are the width and height of the bounding box separately.

As with Faster-RCNN, the proposed network is an end-to-end network. The stochastic gradient descent (SGD) algorithm is used to train the network and each mini-batch contains two images. The initial learning rate is set to 0.001, which will be adjusted to 0.0001 after 60,000 iterations. Then we have to continue running the learning 60,000 iterations at the same rate. Furthermore, we use a momentum of 0.9 and a weight decay of 0.0005.

IV. COMBING DETECTION AND TRACKING

Vehicle detection in each video frame could be completed by the proposed framework. But there still are two issues to solve: 1) the difference between adjacent frames is tiny. If the vehicle detection is performed on every video frame, it may cause too much computational waste; 2) the same vehicle is easily detected in one frame but may be difficult to be detected in another frame. Thus, we combine the vehicle detection and tracking to improve the detection performance. On the trade-off between vehicle detection speed and effect, we choose the kernel correlation filter (KCF) algorithm [22] for tracking.

As shown in Fig.2, the first video frame is input into the proposed detection network. The output detection results are used as the tracking objects and the KCF tracking algorithm is performed in the next few frames (*n* continuous frames). Then the tracking results of the last frame in the tracking process are used as region proposals and input into the detection network for the vehicle detecting in the next video frame. The boxes obtained from the detection network correspond to the region proposals generated from the RPN and the tracking process respectively. After the NMS (Non-Maximum Suppression), we reinitialize a KCF tracker for each box which corresponds to the region proposal generated from the RPN. And for those boxes which correspond to the region proposal generated from the tracking process, we still use the old trackers without reinitializing to track the objects continuously. Finally, the above detection and tracking continue running until the vehicle detection of the entire video is completed.

V. DATASET

The UAV traffic video dataset for experiments is collected and labeled by ourselves. The dataset contains 50 short videos, 30 clips for training and other 20 clips for testing. Some video samples are shown in Fig.3. The dataset can be used not only for vehicle detection but also for object tracking.

To make the dataset more challenging, we collect the UAV traffic video data under different conditions, such as day or night, UAV altitude, surroundings, traffic flow and weather, etc. We have not only labeled the bounding boxes of the vehicles but also marked the occlusion and outside attributes. According to the percentage of vehicle occlusion, the occlusion attribute includes three labels: if the vehicle occlusion is 30% or less, the bounding box is marked as occ-s; 30% to 60% is marked as occ-m; more than 60% is marked as occ-l. According to the percentage of the vehicle outside the video frame, the outside attribute includes two labels: if 20% or less of the vehicle is outside the video frame, the bounding box is marked as out-s; 20% to 50% is marked as out-m. If 50% or more of the vehicle is not in the video frame, the box will be ignored.

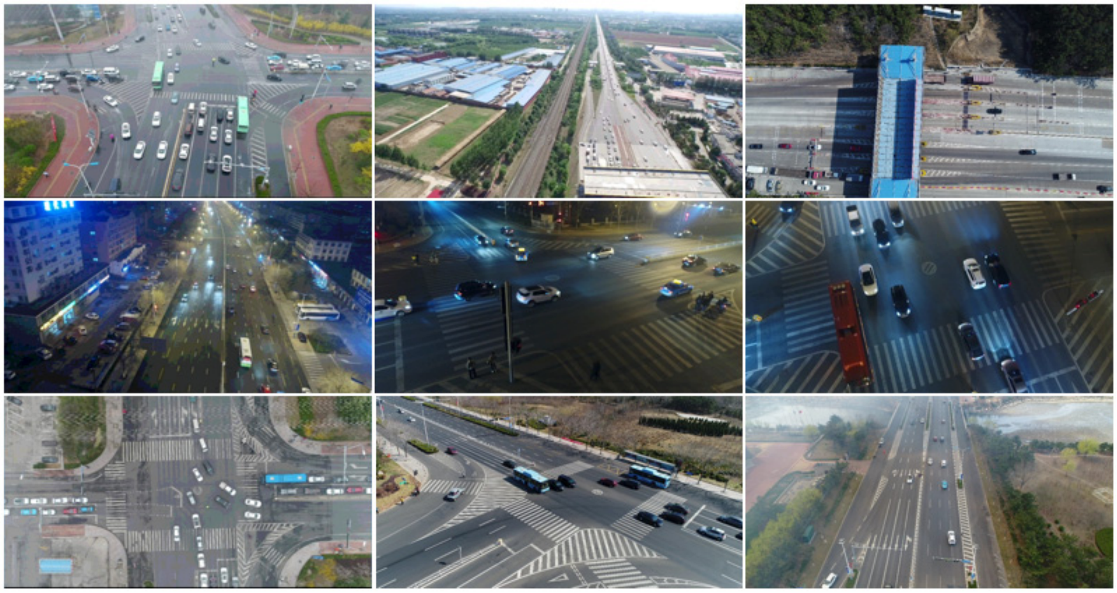

Figure 3. Samples of the self-built UAV traffic video dataset

VI. Experiments

We test the proposed network on the test set which consists of 20 video clips. The vehicle detection results are measured using the mean precision (mAP). As shown in Fig.4, most vehicle targets can be successfully detected even if the target is very small. In addition, we have done some comparative experiments to verify the effectiveness of the proposed network.

A. *Comparison of Different Networks*

We compare the proposed network with four important object detection frameworks which have well performance: Faster-RCNN [7], RFCN [8], MSCNN [15] and HyperNet [21]. Tab. I gives the detection results (mAP) obtained from different networks on the same UAV traffic video dataset.

TABLE I. Comparison of detection results (mAP)

Image Size / Networks	**300px**	**400px**	**500px**	**600px**	**Avg.**
Faster-RCNN	37.4	45.7	55.4	60.6	49.8
RFCN	39.9	48.6	56.8	61.9	51.8
MSCNN	41.6	49.4	56.2	61.1	52.1
HyperNet	43.8	53.2	58.1	62.3	54.4
Our	37.3	51.7	60.6	64.5	53.5
Our + Deconv	**50.4**	**61.0**	**63.3**	**66.1**	**60.2**

As shown in Tab. I, we compare the detecting performance of the networks under different image sizes. The size of the input images is represented by the minimum of the image width and height. For example, if the size of the original image is 600px, 300px means that the vehicle targets to be detected are scaled to half of the originals. From the table, we can see that the proposed network achieves the highest detection accuracy on the UAV traffic video dataset. And in the five networks, MSCNN, HyperNet and our network fuse the multi-level feature maps to optimize the detection performance and obtain higher mAP results. This shows that the fusion of multi-level features is indeed effective for object detection. Furthermore, we have verified that the deconvolution can significantly improve the detection effect and make the mAP results much higher. Especially when the size of the input image is smaller, the effect of the deconvolution is more obvious. This indicates that the deconvolution can effectively improve the detection performance and be suitable for small target detection.

B. *Different Layer Combination*

Experiments have shown that the fusion of multi-level features can improve the detection performance. Thus, to obtain the best performance, which layers should be combined for the fusion?

TABLE II. Recall and mAP results of the proposed network with different layer combination

Combination	**Recall**	**mAP**
1+2+3+4+5	84.7%	64.8%
1+2+3+4	85.2%	65.4%
1+2+3	82.3%	63.3%
2+3+4	**86.7%**	**66.1%**
2+3	81.6%	61.8%
3+4	83.8%	63.8%
2+4	82.5%	62.5%

We have trained multiple networks for experiments with

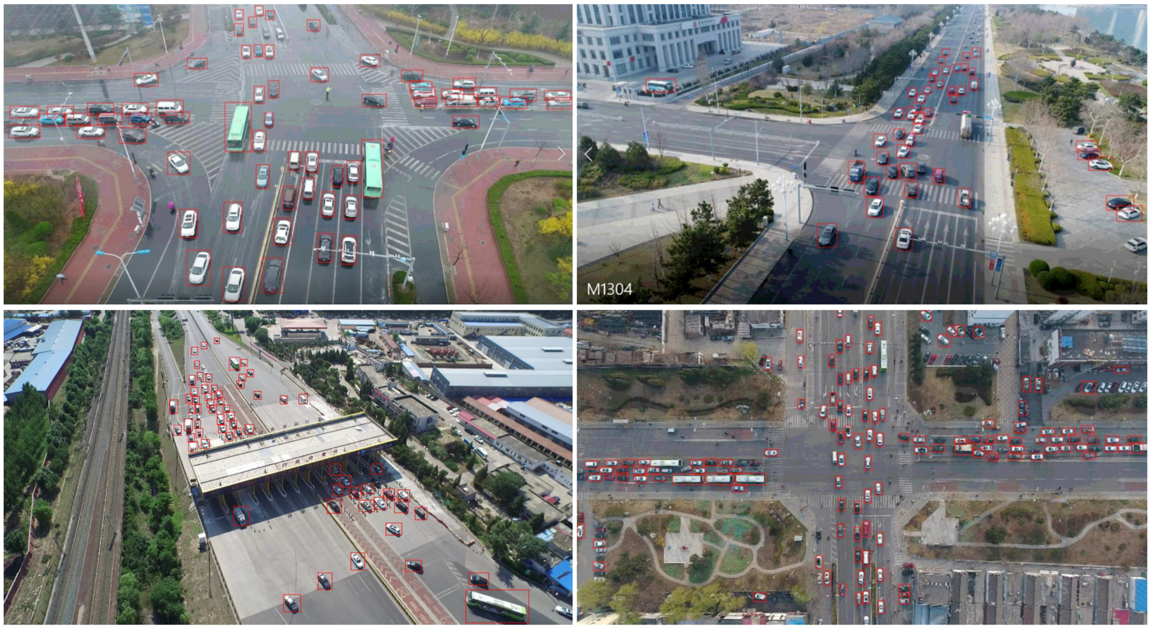

Figure 4. Some vehicle detection results on the UAV traffice videos

different layer combination schemes. As shown in Tab. II, the detection results of the 1+2+3+4+5 layer combination are lower than the 1+2+3+4 layer combination. And It also needs more computation cost because of the Conv-5 layer. It indicates that the features of the Conv-5 layer are useless for the vehicle detection on the UAV traffic video dataset. Thus, we could drop the last convolution layer. Besides, the 2+3+4 layer combination achieves the best performance and we make choice of this scheme for vehicle detection in the self-collected UAV traffic videos. Different layer combination schemes may obtain distinct performances according to the specific datasets and need to be verified by experiments.

C. Combination of Detection and Tracking

We combine the detection and tracking on the Faster-RCNN network and the proposed network to obtain the vehicle detection results on the testing dataset. The running speeds are measured on a NVIDIA Tesla K40 GPU. The combination strategy of detection and tracking is that one video frame is for detecting and the next three consecutive video frames are for tracking. The detection results and running speeds are shown in Tab. III.

TABLE III. THE MAP AND DETECTION SPEED OF THE PROPOSED NETWORK WITH OR WITHOUT TRACKING

Networks	**mAP**	**Detection Speed**
Faster-RCNN	60.6%	5 fps
Faster-RCNN + tracking	61.8%	18 fps
Our	66.1%	3 fps
Our + tracking	67.4%	11 fps

As shown in Tab. III, after the combination of tracking, the detection results of the Faster-RCNN and the proposed network both have being better. Although the detecting speeds are much slower than the networks without tracking, they are still faster than real-time and could meet the requirement of the practical applications.

VII. CONCLUSIONS

We have proposed a convolutional neural network for vehicle detection in UAV traffic videos. This network is also working for general object detection task. The network combines different layers' features and enhances the response information by using a deconvolution layer, which is helpful to improve the detection performance. In addition, we proposed a combination scheme of detection and tracking to achieve more accurate vehicle detection results. The proposed network achieves better performance on the UAV traffic video dataset than other four networks. In the future, we will do our best to find more efficient techniques for vehicle detection in UAV traffic video, especially for the detection of very small vehicles.

ACKNOWLEDGMENT

This work was supported in part by Natural Science Foundation of China: 61620106009, 61772494, 61332016, U1636214, 61650202, 61672497, 61472389, in part by Shandong Provincial Natural Science Foundation, China: ZR2017MF001, in part by Key Research Program of Frontier Sciences under Grant CAS: QYZDJ-SSW-SYS013.

REFERENCES

[1] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 2005, pp. 886-893 vol. 1.

[2] C. P. Papageorgiou, M. Oren and T. Poggio, "A General Framework for Object Detection," Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271), Bombay, 1998, pp. 555-562.

[3] E. Shelhamer, J. Long and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 4, pp. 640-651, April 1 2017.

[4] J. R. R. Uijling, K. E. A. van de Sande, T. Gevers and A. W. M. Smeulders, "Selective Search for Object Recognition," in International Journal of Computer Vision, vol. 104, no. 2, pp. 154-171, Sept. 1 2013.

[5] K. He, X. Zhang, S. Ren and J. Sun, "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 37, no. 9, pp. 1904-1916, Sept. 1 2015.

[6] R. Girshick, "Fast R-CNN," 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, 2015, pp. 1440-1448.

[7] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137-1149, June 1 2017.

[8] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object Detection via Region-Based Fully Convolutional Networks," Advances in Neural Information Processing Systems (NIPS), Barcelona, Spain, 2016, pp. 379-387.

[9] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 779-788.

[10] W. Liu, D. Anguelov, D. Erhan, et al., "SSD: Single Shot Multibox Detector," European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 2016, pp. 21-37.

[11] A. Ghodrati, A. Diba, M. Pedersoli, et al., "DeepProposal: Hunting Objects by Cascading Deep Convolutional Layers," 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, 2015, pp. 2578-2586.

[12] C. Szegedy et al., "Going Deeper With Convolutions," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, 2015, pp. 1-9.

[13] E. Shelhamer, J. Long and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 4, pp. 640-651, April 1 2017.

[14] S. Xie and Z. Tu, "Holistically-Nested Edge Detection," 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, 2015, pp. 1395-1403.

[15] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos, "A Unified Multi-Scale Deep Convolutional Neural Network for Fast Object Detection," European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 2016, pp. 354-370.

[16] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," CoRR abs/1409.1556 , 2014.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," Advances in Neural Information Processing Systems (NIPS), Lake Tahoe, Nevada, USA, 2012, pp. 1106-1114.

[18] M. D. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks," European Conference on Computer Vision (ECCV), Zurich, Switzerland, 2014, pp. 818-833.

[19] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 770-778.

[20] O. Russakovsky, J. Deng, H. Su, et al., "ImageNet Large Scale Visual Recognition Challenge" in International Journal of Computer Vision, vol. 115, no. 3, pp. 211-252, 2014.

[21] T. Kong, A. Yao, Y. Chen and F. Sun, "HyperNet: Towards Accurate Region Proposal Generation and Joint Object Detection," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 845-853.

[22] J. F. Henriques, R. Caseiro, P. Martins and J. Batista, "High-Speed Tracking with Kernelized Correlation Filters," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 37, no. 3, pp. 583-596, March 1 2015.