AFESSD: Object Detector for Natural Gas Pipeline Construction Scene

Zhen Yang$^{1,a}$, Qinhao Zheng$^{1,b,*}$, Jie Zhou$^{1,c}$, Zhijian Yin$^{1,d}$

$^1$School of Communication & Electronics, Jiangxi Science & Technology Normal University, Nanchang, Jiangxi, 330000, China

$^a$e-mail: yangzhenphd@aliyun.com, $^b$e-mail: 1020170701@jxstnu.com, $^c$e-mail: foryu0923@aliyun.com, $^d$e-mail: zhijianyin@aliyun.com

Abstract. Object detection is a challenging task in computer vision, which aims to classify and locate objects. SSD (Single Shot MultiBox Detector) is one of the classical algorithms in the current object detection field, which uses multiple feature maps with different scales to detect all objects in an image. Based on SSD, in this paper, we propose an improved AFESSD for object detection in natural gas pipeline construction scene: Firstly, the attention mechanism module is introduced; Secondly, Feature Fusion Block and Feature Enhancement Block are designed to achieve high-to-low level feature information fusion and enhancement. Finally, we verify AFESSD on two object detection datasets of natural gas pipeline construction scene. Experimental results on the first dataset demonstrate that SSD and AFESSD achieve 81.35% and 83.13% mean average precision (mAP) respectively. On the second dataset, SSD and AFESSD achieve 50.55% and 50.99% mAP respectively. Therefore, AFESSD improves SSD’s object detection accuracy in natural gas pipeline construction scene.

1. Introduction

Recently, with the development of low-carbon energy structure, natural gas will play an increasingly important role in the energy consumption structure. Pipeline is the main way of natural gas resource distribution. Whether the behavior is standardized during the construction process of natural gas pipelines is related to the life security of relevant personnel. Therefore, it is more and more significant to inspect irregularities of relevant construction personnel in monitoring video. Since manual processing is too time-consuming and laborious, it is very important to use object detection system. Currently, object detection methods can be divided into two types: 1) traditional object detection method based on hand-crafted features [1-4]; 2) object detection method based on deep learning. The traditional machine learning method firstly preprocess the image, then determine candidate regions through the sliding-window paradigm and extract features of these candidate regions, finally, the classifiers are used to judge the category of extracted features. As hand-crafted features cannot meet the needs of multiple features in complex scene, their generalization ability is poor, which brings many difficulties to the detection of related object in construction scene. In addition, the process of feature extraction is too complex, resulting in slow detection speed. Eventually, the object detection effect is poor. In recent years, with the rapid development of deep learning, image processing technology has been continuously improved. Meanwhile, it also has promoted the development of object detection to a large extent. Subsequently, various object detection methods based on Convolutional Neural Networks (CNN) have been proposed. Compared with traditional methods,
these methods have stronger feature extraction capability, better generalization ability and higher accuracy.

2. Related Works

2.1. Two-Stage Method
At present, the CNN-based object detection methods can be divided into two embranchments: two-stage method and single-stage method. The object detection of two-stage method is divided into two steps: firstly, a large number of candidate regions are generated on the image; secondly, the object category and regression information are output through CNN. Typical two-stage methods include R-CNN [5], SPP-Net [6], Fast R-CNN [7], Faster R-CNN [8] and R-FCN [9]. Early methods such as R-CNN and Fast R-CNN use selective search algorithm to generate regional proposal candidates, which results in time-consuming running of the framework. Later, Faster R-CNN innovatively proposed the idea of Anchor, pre-setting prior boxes of different scale and aspect ratio in feature maps. Then the Region Proposal Network (RPN) is used to perform binary classification and rough position adjustment on these prior boxes. Finally, final classification and secondary position refinement of these prior boxes are carried out through a sub-network, realizing the end-to-end training.

2.2. One-Stage Method
The two-stage method generally has high detection accuracy, but due to complex network structure and too many model parameters, its speed is too slow to meet the requirements of real-time detection. Based on this, the one-stage method was subsequently proposed. Different from the two-stage method, the one-stage method does not need to generate candidate regions, instead directly classifying and regressing the object in one step to achieve trade-off between speed and accuracy. The representative methods are YOLO [10] and SSD [11]. YOLO divides an image into $S \times S$ grids, abandoning regional proposal candidates, and each grid is responsible for predicting the object on which the center point falls. Finally, YOLO directly outputs the object category and location regression at the end of the network, which greatly improves the detection speed. However, YOLO does not consider objects with different sizes in the image and uses full connection layer, which will make spatial location information lost and lead to miss detection of intensive objects. Consequently, YOLO’s Recall is underperforming. SSD combines the idea of Anchor and Regression and simultaneously adopts multiple convolution layers with different scales to detect object, giving consideration to both detection accuracy and speed. However, SSD does not take into account the relationships between different layers and take full advantage of the semantic information from different layers.

Figure 1. The architecture of our AFESSD.
3. Methods
The AFESSD architecture we proposed is illustrated in Figure 1. Since AFESSD is based on the SSD, in this section, we first briefly review the structure of SSD and then introduce our improved module.

3.1. SSD Framework
The Single Shot MultiBox Detector (SSD) uses the VGG-16 [12] as based network and adds extra convolution layers to the end of it. Then selecting some layers of different sizes from basic and extra convolution layers and to predict scores and offsets for some prior boxes on each scale. The shallow and deep feature maps are responsible for detecting small and large object respectively.

3.2. FFBlock
Although, the SSD’s shallow feature map contains more detailed information, it has limited semantic information and insufficient feature representation ability, resulting in the detection performance for small objects on shallow layer is far weaker than that for large objects on deep layer. Based on this, we proposed Feature Fusion Block (FFBlock) to fuse different information from high levels to low levels and apply it to three detection layers of Conv4_3, Conv7, and Conv8_2. The specific structure of FFBlock is shown in Figure 2. In order to fuse the two adjacent feature maps in the original SSD, it is necessary to double spatial resolution of the latter layer to unify the size of feature maps. To avoid increasing the number of parameters, bilinear interpolation is used instead of deconvolution for up-sampling. Finally, the output results of two adjacent detection layers are multiplied by elements to obtain the output results of FFBlock.

![Figure 2. Illustration of the FFBlock.](image)

3.3. SEBlock
To enrich the attention features of FFBlock output feature maps, SEBlock [13] is connected after each FFBlock, which can improve the relationship between feature channels, facilitate the extraction of global information and improve the global receptive field of feature maps. Referring to SENet, more detailed introduction about SEBlock can be comprehended.
3.4. FEBlock
In order to fully enhance feature extraction and regression capability to object location of deep network, Feature Enhancement Block (FEBlock) was designed and applied in three detection layers Conv8_2, Conv9_2, Conv10_2 and Conv11_2, which is shown in Figure 3. Firstly, according to the Partial Transition [14] idea, input features maps are divided into two branches using two 1x1 convolution, one of which is connected to the residual module. Then the output features maps of two branches are connected in series. Finally, 1x1 convolution is used for feature fusion. Unlike CSPNet, which directly divides the features map into two parts according to the channel, we believe that using two 1x1 convolution can further improve the reusability of features, reducing the amount of computation concurrently. Combining residual module [15] and CSP strategy not only enables the network to learn more nonlinear relationships, but also different layers avoid to learn repeated gradient information, which can effectively strengthen the feature learning ability of detection layer and enrich the feature information of deep network.

4. Experiments
4.1. Dataset Establishment
Since there is no public dataset for object detection of natural gas pipeline construction scene, we had made the natural gas pipeline construction scene object detection dataset by ourselves. Most of pictures come from the photographing of dozens of natural gas pipeline construction scene by UAV, and a small part of pictures come from the web crawler. We screen for these pictures in the first place, removing ones which only contain background or very small number of objects. Then we mark images by labelImg. With this marking tool, when manually operating, we only need to mark various user-defined objects in the image, and the tool can automatically generate corresponding configuration file. Finally, they are saved in the format of Pascal VOC detection dataset [16], and the relevant files are
stored in three folders, Annotations, ImageSets and JPEGImages. Data annotation was completed in two stages, forming two datasets consequently, named First-Phase Dataset and Second-Phase Dataset. The labeled objects of datasets in the two periods are different. The specific object categories examples are shown in Figure 4, where ‘yes’ and ‘no’ indicate whether the pipeline is sealed. The number of pictures in First-Phase Dataset is 4267, with 8 positive sample categories. The number of pictures in Second-Phase Dataset is 3374, with 7 positive sample categories. Then the two datasets were divided into train set and test set in the ratio of 9:1.

![Figure 4](image)

**Figure 4.** The specific object categories examples of First-Phase Dataset (a) and Second-Phase Dataset (b)

4.2. Training Details
Our experiment was carried out on a computer whose GPU is NVIDIA Geforce GTX 1080Ti, and implementation code of the method is based on the Pytorch framework. In order to obtain a more robust and better performance model, it is necessary to make data enhancement for training images, including random clipping and flipping, which is consistent with the original SSD. Furthermore, we use Adam to optimize the objective functions of AFESSD, which are composed of the Cross Entropy Loss for classification and the Smooth L1 Loss for location. Due to the memory limitation of GPU, when the input image size is 300, we set the batch size to 16 and initial learning rate to 0.0005, training 100 epochs.

4.3. Experimental Results
Based on the divided two datasets, SSD and AFESSD are compared on mAP. As shown in Figure 5 and Figure 6, under our experimental conditions, in First-Phase Dataset, the mAP of SSD is 81.35% and that of AFESSD is 83.13%. In Second-Phase Dataset, the mAP of SSD is 50.55% and that of AFESSD is 50.99%. These experimental results show that AFESSD method we proposed increases SSD’s mAP by 1.78% and 0.44% respectively, achieving better detection performance.
4.4. Detection Samples
In order to intuitively compare the performance of AFESSD and SSD, we randomly selected a certain number of images from the test set and visualized the detection results. As shown in Figure 7, AFESSD detects more objects than SSD in the same image. In addition, AFESSD uses FFBlock to combine each layer from Feature Pyramid, thereupon gaining more image context information to improve the detection accuracy. FEBlock that we put forward can effectively enhance the semantic information for feature maps. Therefore, we can see clearly that AFESSD’s object confidence score is better than SSD.
5. Conclusions
We use attention mechanism module and design Feature Fusion Block and Feature Enhancement Block to improve SSD. Experiments are carried out on two object detection datasets of natural gas pipeline construction scene to verify our approach. Experimental results show that the accuracy of AFESSD method proposed in this paper is better than SSD, achieving better detection effect. AFESSD can be applied to detect violations of relevant construction personnel in surveillance video, protecting the life security of relevant personnel. In future work, we hope to enhance AFESSD with stronger backbone networks, in addition to applying the proposed fusion module and enhancement module to other computer vision tasks such as semantic segmentation.

Acknowledgments
This work was supported by the National Natural Science Foundation of China (61866016, 62061019), the General Project of Jiangxi Natural Science Foundation(20202BABL202014), the Key Project of Jiangxi Education Department(GJJ201107), and the Youth Top Talent Foundation of Jiangxi Science and Technology Normal University (2018QNBRC002).

References
[1] Ahonen, T., Hadid, A., Pietikäinen, M. (2006) Face Recognition with Local Binary Patterns. IEEE Trans. Pattern Anal. Mach. Intell. 28: 2037–2041.
[2] Lowe, D.G. (2004) Distinctive Image Features from Scale-Invariant Key points. Int. J. Comput. Vis. 60: 91–110.
[3] Dalal, N., Triggs, B. (2005) Histograms of oriented gradients for human detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Diego, CA, USA. pp. 886–893.
[4] Lienhart, R., Maydt, J. (2002) An extended set of Haar-like features for rapid object detection. In: Proceedings of the International Conference on Image Processing, Rochester, NY, USA. pp. 900–903.
[5] Girshick, R., Donahue, J., Darrell, T., Malik, J. (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA. pp. 580–587.
[6] He, K., Zhang, X., Ren, S., Sun, J. (2014) Spatial pyramid pooling in deep convolutional networks for visual recognition. In: Proceedings of the European Conference on Computer Vision, Zurich, Switzerland. pp. 346–361.
[7] Girshick, R. (2015) Fast r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, pp. 1440–1448.
[8] Ren, S., He, K., Girshick, R., Sun, J. (2015) Faster r-cnn: Towards real-time object detection with region proposal networks. In: Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada. pp. 91–99.
[9] Dai, J., Li, Y., He, K., et al. (2016) R-fcn: object detection via region-based fully convolutional networks. Advances in neural information processing systems, 29: 379-387.
[10] Redmon, J., Divvala, S., Girshick, R., Farhadi, A. (2016) You only look once: Unified, real-time object detection. In: Proceedings of the International Conference on Computer Vision & Pattern Recognition, Las Vegas, NV, USA. pp. 779–788.
[11] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C. (2016) Ssd: Single shot multibox detector. In: Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands. pp. 21–37.

[12] Simonyan, K., Zisserman, A. (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint, arXiv:1409.1556.

[13] Hu, J., Shen, L., Sun, G. (2018) Squeeze-and-excitation networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, USA. pp. 2011-2023.

[14] Wang, C-Y., Liao, H-Y. M., Wu, Y-H., et al. (2020) CSPNet: A new backbone that can enhance learning capability of CNN. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops, pp. 390-391.

[15] He, K., Zhang, X., Ren, S., Sun, J. (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA. pp. 770–778.

[16] Everingham, M., Eslami, S.M.A., Gool, L.V., Williams, C.K.I., Winn, J., Zisserman, A. (2005) The Pascal Visual Object Classes Challenge. A Retrospective. Int. J. Comput. Vis, 111 (1) :98-136.