IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 18, 2019, accepted February 4, 2020, date of publication February 10, 2020, date of current version February 18, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2972562

Moving Object Detection With Deep CNNs

HAIDI ZHU[1,2], XIN YAN[1], HONGYING TANG[1], YUCHAO CHANG[1], BAOQING LI[1], AND XIAOBING YUAN[1]

[1]Wireless Sensor Network Laboratory, Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, Shanghai 201800, China
[2]University of Chinese Academy of Sciences, Beijing 100049, China

Corresponding author: Baoqing Li (sinoiot@mail.sim.ac.cn)

ABSTRACT In large field of view for open country, the real-time detection and identification of moving objects with high accuracy is a very challenging work due to the excessive amount of data. This paper proposes a novel framework that consists of a coarse-grained detection as well as a fine-grained detection. To solve the problem of noise-induced object fracture during the coarse-grained detection process, we present a low-complexity connected region detection algorithm to extract moving regions. Furthermore, in the fine-grained detection, Deep Convolution Neural Networks are leveraged to detect more precise coordinates and identify the category of objects. To the best of our knowledge, this is the first work that proposes a coarse-to-fine grained framework to detect moving objects on high-resolution scenes. Experimental results show that the proposed framework can robustly work on the high resolution video frames (1920*1080p) with complex situations more fastly and accurately over existing methods.

INDEX TERMS Connected region detection, deep convolution neural networks, foreground extraction, high resolution, moving object detection.

I. INTRODUCTION

Field unattended monitoring systems are required to possess accurate and real-time image processing ability to detect, identify and track moving objects. Accurate detection of moving objects is the necessary prerequisite for a tracking system. While ensuring the effectiveness of monitoring, it is challenging to achieve real-time requirements on high-resolution scenes with a large field of view. What's more, in practical field scenarios, complex background, illumination changes, local motion such as waving trees, dust trailing, camouflage objects and etc make the system suffer from poor performance.

For moving detection, the existing state-of-the-art methods mainly include optical flow [1], [2], background subtraction [3]–[6], frame difference [7] and deep learning methods [8]–[11]. Regrettably, these methods can not work on high-resolution scenarios with large noises very well and have their own weaknesses. For example, the input of the deep learning algorithm is usually much smaller than $1920 * 1080$. Networks with too large input size consume too much computing resources and are hard to achieve the high speed. Background subtraction, which builds the background model and detects foreground, is sensitive to the background modelling step.

In addition, many video surveillance systems with these methods can only detect moving objects without obtaining the category and precise coordinates of each moving object. To deal with this issue, feature extraction and classification are combined in [12]. However, a very serious drawback of this method that can not be ignored is that the object regions obtained by moving detection can not be classified as a group or single object. Moreover, the precise coordinates of each moving object are not available, which further disturbs the classification result.

In this paper, a course-to-fine grained framework is proposed for moving object detection and identification. Firstly, moving regions are obtained by course-grained detection. Then connected regions detection is performed. Finally, in the fine-grained detection, coordinates of each object are corrected and the category of the object is obtained.

II. RELATED WORK

Generally, each video frame is divided into the foreground and background to detect moving objects according to the difference of pixel intensity or color distribution. In literature [1]–[7], [13], [14], researchers propose different methods of artificial design for foreground extraction. As a most

The associate editor coordinating the review of this manuscript and approving it for publication was Weiping Ding.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see http://creativecommons.org/licenses/by/4.0/

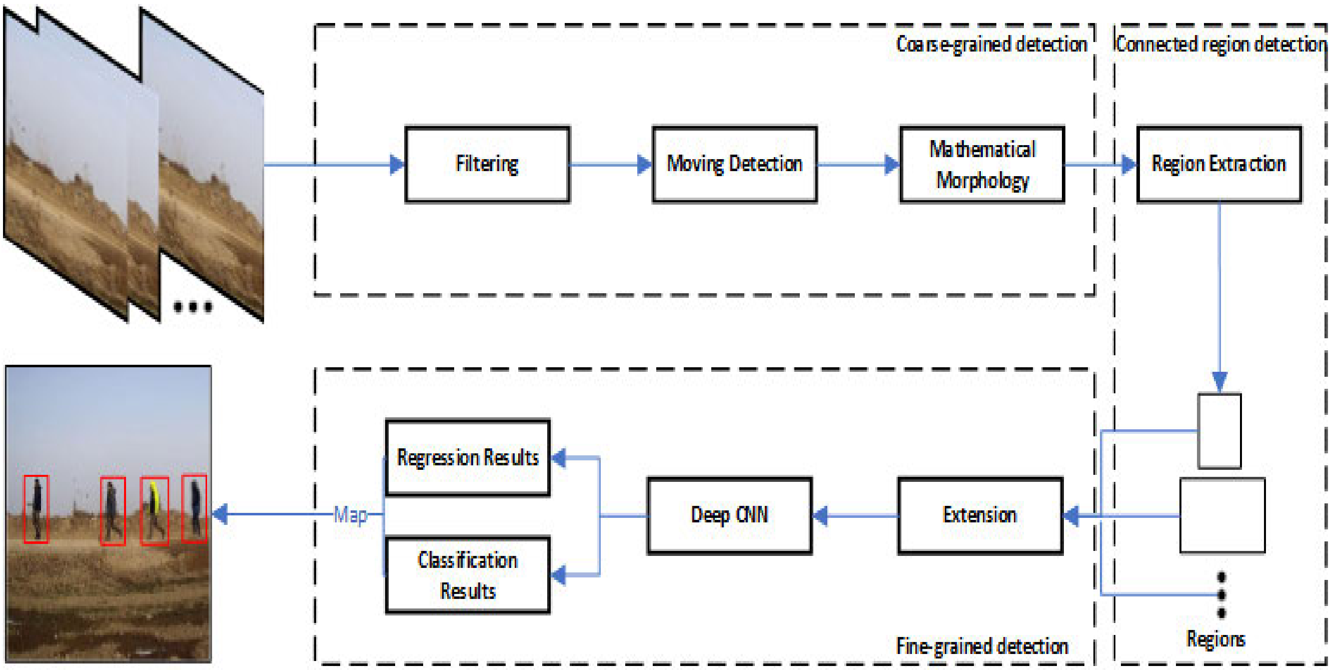


FIGURE 1. **Framework schema.**

generally applied method, frame difference [7] is based on the gray level difference between two adjacent frames of video to judge the motion. Without modelling background, it is simple to implement, yet is vulnerable to noise or complex scenes with local motion in the background. Meanwhile, Gaussian mixture model (GMM) [13]–[15] is more robust, but it needs multiple frames for modelling and updates the background iteratively, which suffers from high computational complexity and is hard to handle the video frames with illumination variation, infrequently moving object and camouflage.

After moving detection, coordinates of regions can be obtained by connected region labelling algorithms. By scanning the input image several times, scan mask techniques [16]–[18] attach a label to each pixel and divide the target according to the label. To obtain faster speed, block-based methods [19]–[21] are presented. However, these algorithms still consume too many computing resources and can not merge the noise-broken object.

In recent years, Deep Convolution Neural Networks (DCNNs) for object detection have gained a lot of interest for their powerful learning ability. By learning parameters themselves, it can achieve a high degree of accuracy. State-of-the-art object detection networks comprise RCNN and its variants, SSD and its variants, YOLO and its variants, etc. RCNN and its variants [22]–[26] are based on region proposal, which is accurate but time-consuming. YOLO and its variants [27]–[31] are known because of their fast speed and high efficiency. SSD and its variants [32]–[36] blend the advantages of these two methods. However, for moving object detection, these deep learning architectures face several critical issues:

1) The size of images with 1920*1080p resolution is too large for existing deep learning architectures. Just sampling image creates detection lost of small objects.
2) The movement can not be recognized with these networks [22]–[36], and it is hard to remove the stationary objects.

Consequently, in order to get a high speed and accuracy detection on high-resolution scenes, we propose a coarse-to-fine grained framework which combines moving detection and identification to get the category and bounding box of each moving object. Coarse-grained detection is performed firstly to obtain moving regions. Due to the persence of complex scene, local motion, illumination variation and so on, the regions obtained may be inaccurate such as containing none object or broken object. To merge the regions that contain the fractured objects, we propose a low-complexity method to extract the connected regions. Then, by the fine-grained detection with DCNNs, coordinates and category of each moving object are obtained. Furthermore, in order to get faster speed, YOLOV3 and its tiny version are modified for the fine-grained detection. Extensive experiments show that our framework can work well on the high-resolution $(1920 * 1080)$ scene in high accuracy and fast speed.

The innovation points proposed in this paper are summarized in the follwing:

1) Coarse-to-fine grained moving object detection framework which combines moving detection with DCNNs is proposed.
2) An efficient algorithm is proposed to detect the connected regions.
3) The structure of the tiny version of YOLOV3 is modified to make the detection faster.

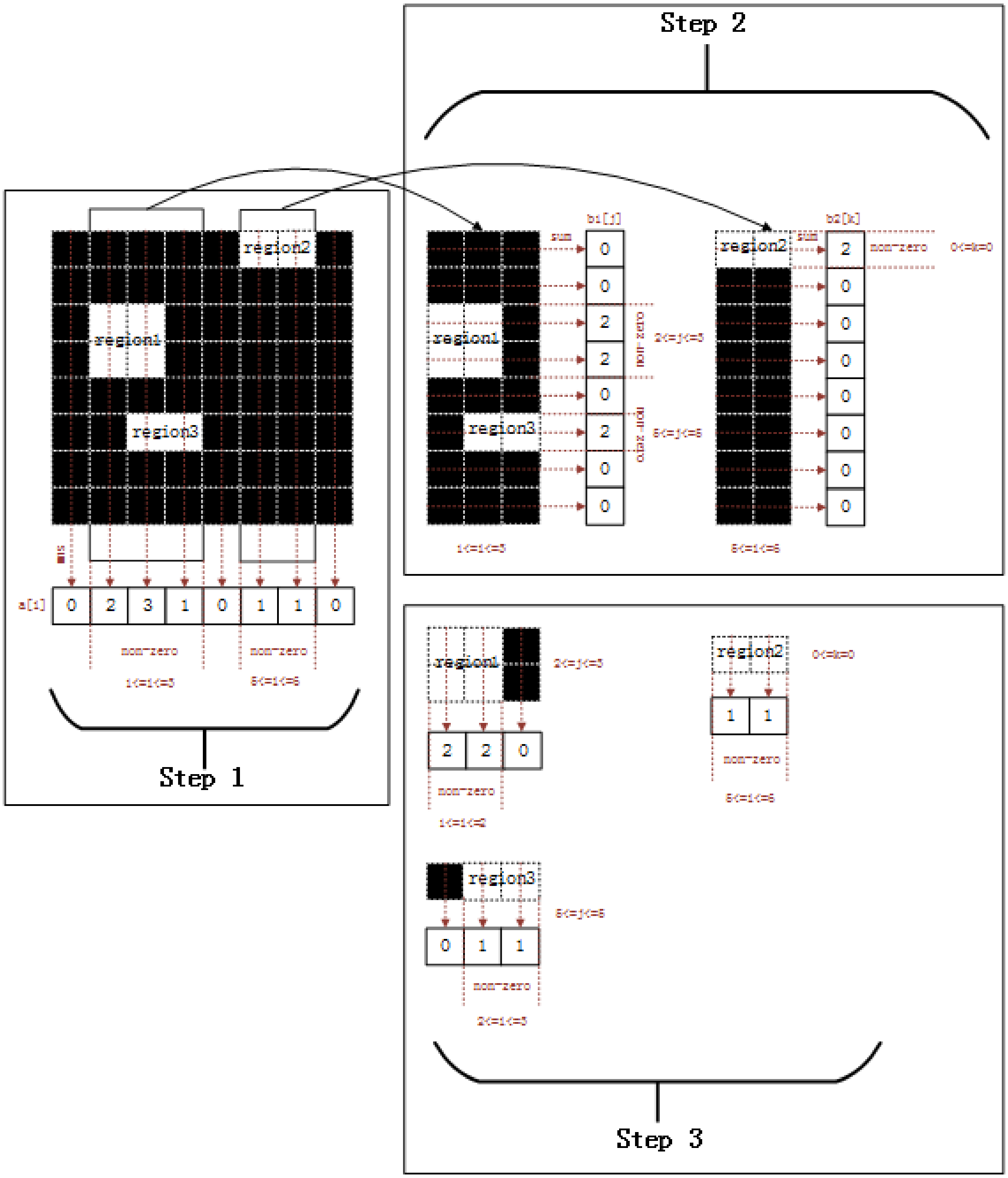
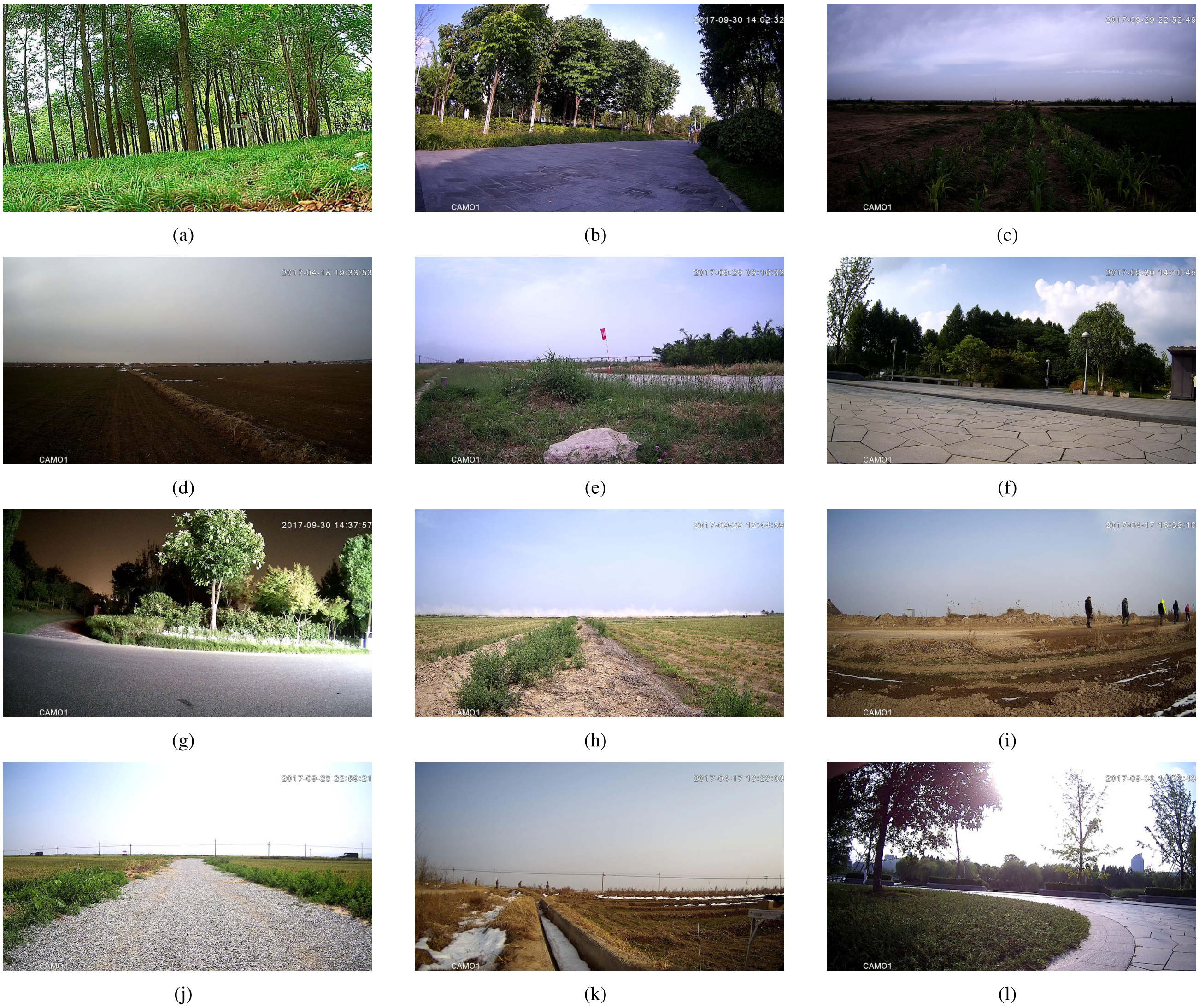


FIGURE 2. An example of connected region detection algorithm on binary image 8 × 8.

The rest of paper is organized as follows: In Section III, we introduced our method and the various parts of our framework in detail. For Section IV, we describe experiments and analyse results. Finally, Section V concludes this paper.

III. PROPOSED METHOD

Figure 1 shows the whole proposed framework. The system consists of two parts: moving detection and object classification and regression. Firstly, moving detection is implemented by moving detection module with filter module and mathematical morphology module (opening operation). Then the connected region extraction is performed to obtain the moving regions. According to the position information of the moving regions, the original image is cut and the cropped regions are fed into the neural network to get the bounding box and category of each object. Finally, according to the position information of the object in the moving region, coordinates are mapped to the original image. The key modules in the overall architecture include moving detection, region extraction and object detection.



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j) (k) (l)

FIGURE 3. Some typical video frames (1920*1080p) in SimitMovingDataset.

A. COARSE-GRAINED DETECTION

In the coarse-grained detection step, filtering and mathematical morphology are also performed to reduce the adverse effect of noises. Firstly, image frames are filtered by low pass filter to eliminate the high frequency noises. After that, moving detection algorithm is performed to detect motion. Finally, mathematical morphology (opening operation) is used to further suppress ill effects of noises. In the content of the high resolution scenes, we choose frame difference as the moving detection algorithm, which is simple to implement and is more responsive to almost all movements. Furthermore, as an initial trial in the coarse-grained detection, its potential noise-contaminated results can be corrected by fine-grained detection.

Note that other methods such as GMM can also be applied for moving detection in the coarse-grained detection stage, especially for the situation with a small illumination transformation. According to the distribution of each pixel in the time domain, the distribution model is built to achieve the purpose of background modeling. Compared the pixels of the input image with the background model, the pixels with high similarity to the background model are regarded as the background, and the existing model is updated, while the pixels with low similarity to the background model are regarded as the foreground. Therefore, it is more time-consuming. In the situation with suddenly exceptionally large illumination changes, both GMM and frame difference fail to handle it. However, such situation rarely happens in the wild. We will also show in the simulation parts that frame difference is both cost-effective and performance-efficient.

B. CONNECTED REGION DETECTION

Due to the complex background, occlusion, etc, the same object may be segmented. To deal with this issue, we propose an efficient region extraction algorithm as shown in Algorithm 1.

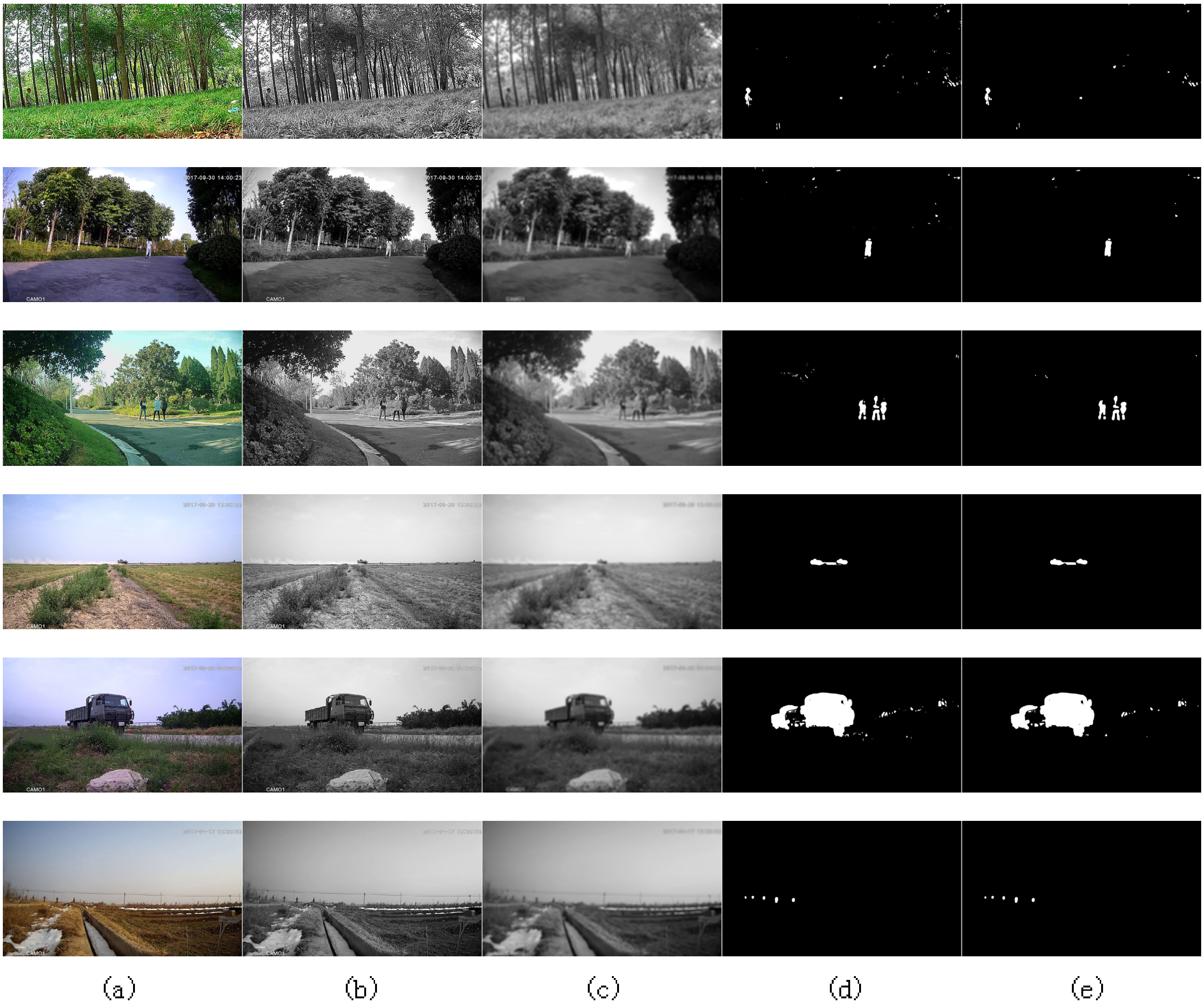


FIGURE 4. Visual results of each step in coarse-grained detection. (a) input video frames. (b) corresponding gray images. (c) results of gray images after filtering processing. (d) results of frame difference. (e) final results after filtering and mathematical morphology operation.

In order to show this algorithm more clearly, we give an example shown in Figure 2. Resolution of the binary image is 8 × 8. The column is marked as column i $(0 <= i <= 7)$ and the row is marked as row j $(0 <= j <= 7)$. The threshold is set as zero. In step 1, values of each column are summed as $a[i]$. The value of $a[1]$ - $a[3]$ and $a[5]$ - $a[6]$ is non-zero. In step 2, pixel values of each row from column 1 to column 3 are summed as $b1[j]$. Values of $b1[2]$ - $b1[3]$ and $b1[5]$ - $b1[5]$ are non-zero. In parallel, pixel values of each row from column 5 to column 6 are summed as $b2[k]$. Value of $b2[0]$ - $b2[0]$ is non-zero. In step 3, sum pixel values of each column for every region as shown in the step 3 of Figure 2 to refine the result according to columns which are non-zero. Therefore, the upper-left coordinate of region1 is (2, 1) and lower-right coordinate is (3, 2). We represent coordinates of region1 as [(2, 1), (3, 2)]. Similarly, coordinates of region2 and region3 are [(0, 5), (0, 6)], [(5, 2), (5, 3)].

In Algorithm 1, there are only some addition and judgment operations. Therefore, compared with existing connected region detection algorithms, the computational complexity of Algorithm 1 is much lower.

C. FINE-GRAINED DETECTION

After coarse-grained detection and region extraction, it is much easier to detect objects. However, multiple objects such as human beings and vehicles may appear in the moving region at the same time. To merge the segmented object, multiple objects may be detected as one moving region in some cases. In addition, owing to illumination changes, complex background, local motion, etc, some problems about obtained moving regions still exist such as incomplete edge or segmented object. The influence of dust and shadow also leads to the oversized obtained moving region. Therefore, fine-grained detection of particular importance is introduced to deal with these issues.

In fine-grained detection, objects occupy a large area of the region which is beneficial for object detection with DCNNs. Therefore, without a very complex network,

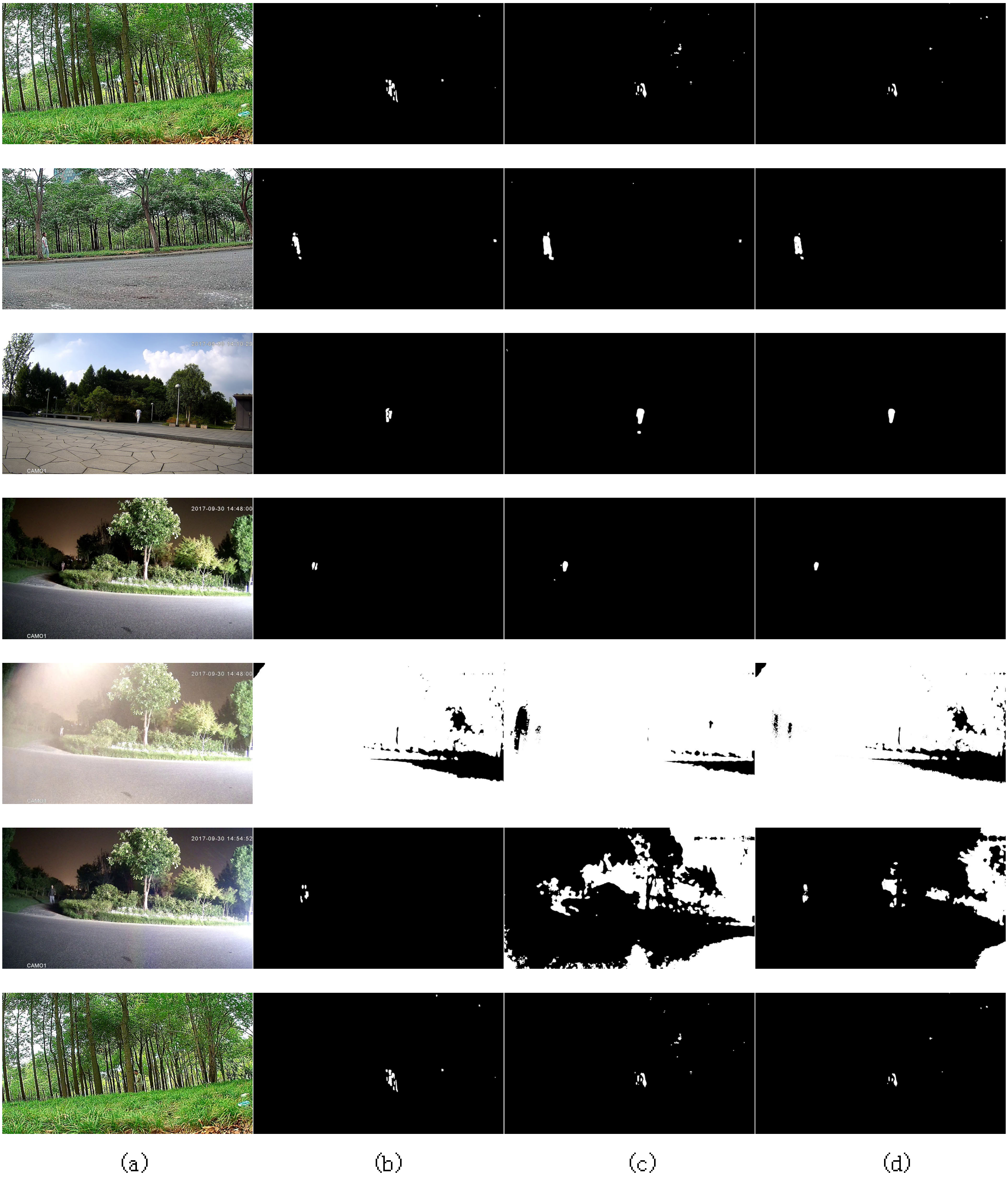


FIGURE 5. Visual results of different moving detection algorithms. (a) input video frames. (b) results of frame difference. (c) results of GMM. (d) results of KNN.

it can also get a satisfying detection result. Compared with the accuracy, the speed of the network is a more concerned issue. YOLOV3 [31] stands out among several typical object detection architectures with high speed, which uses darknet-53 as the base network for feature extraction, and uses three feature maps for prediction.

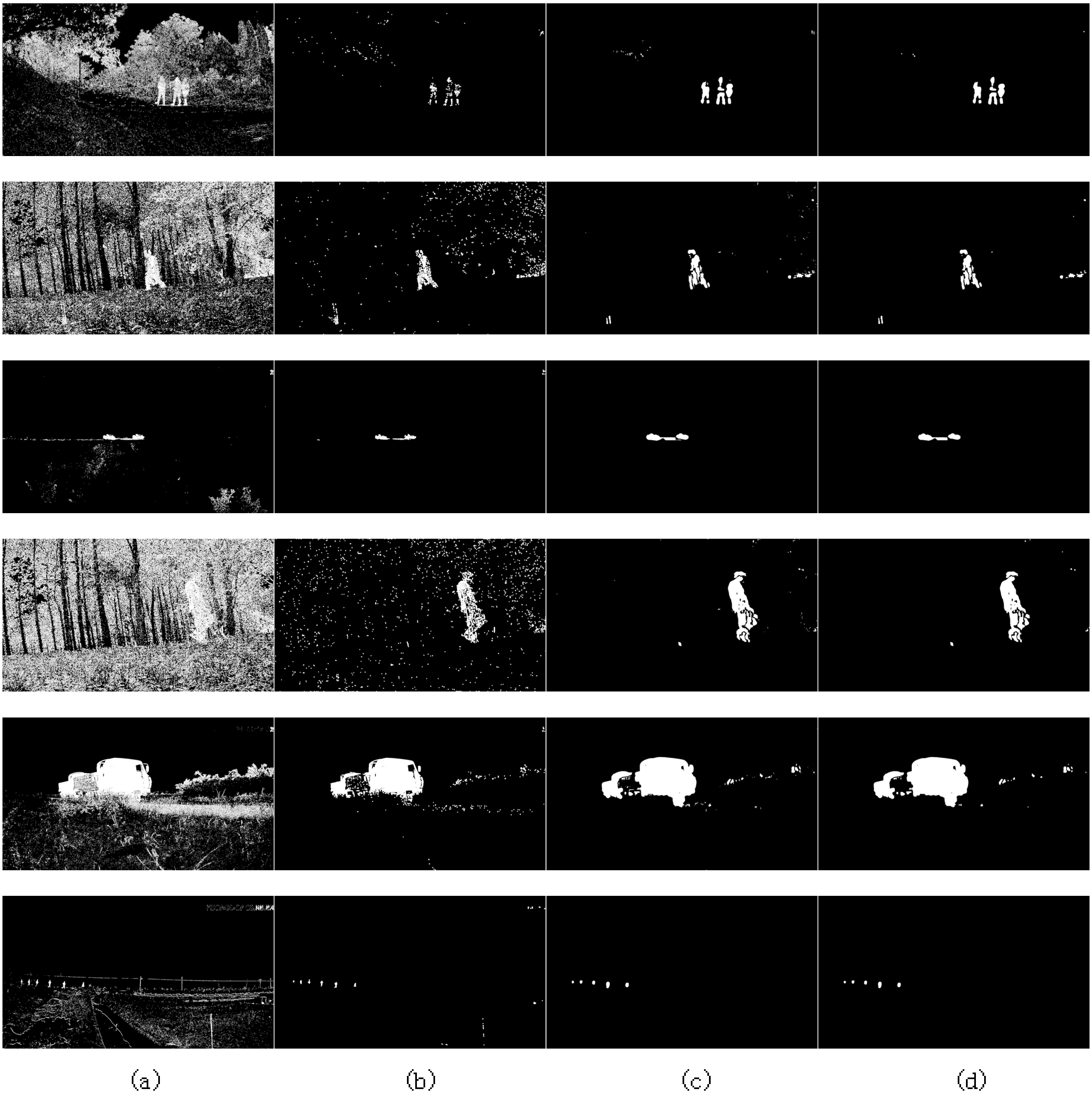


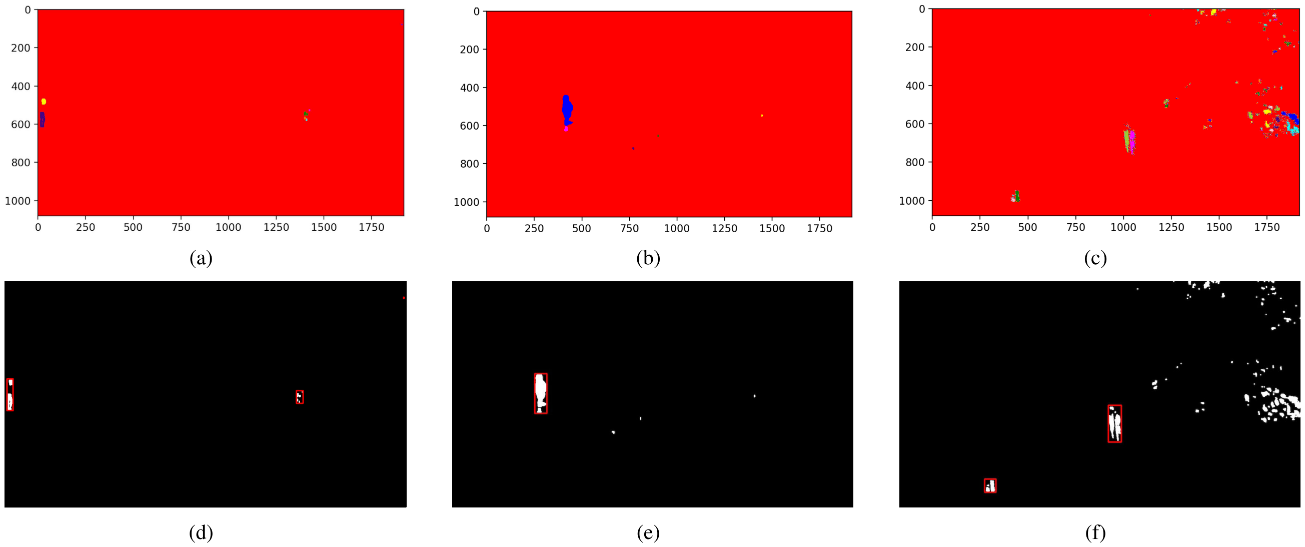
FIGURE 6. The effects of filtering and mathematical morphology. (a) without filtering and morphology processing. (b) only with the operation of mathematical morphology. (c) only with filtering. (d) both filtering and mathematical morphology.

In the context of the fine-grained detection, we encounter an issue: moving objects obtained by coarse-grained detection is likely incomplete caused by noises. Therefore, as shown in Figure 1, before detected by the network, we extend the regions obtained in the coarse-grained detection to ensure the integrity of the objects. A method is proposed to extend the region according to the prior knowledge of the anchors in modified YOLOV3 which is shown below. We choose the anchor that yields the largest IoU [37] value with the region and expand the region based on the aspect ratio of the anchor especially.

$$ratio = \frac{w}{h} = \frac{\Delta w}{\Delta h}, \tag{1}$$

$$\Delta w = \Delta h \times ratio. \tag{2}$$

The ratio is defined in Eq. (1), where w and h denote the width and height of the chosen anchor, and the Δw and Δh are the size of extent. Therefore, the extending method is $\Delta w = \Delta h \times rartio$ in Eq. (2).



(a) (b) (c)

(d) (e) (f)

FIGURE 7. Results obtained by skimage (top row). Results of our proposed method (bottom row). The top row and bottom row are corresponding to the same images.

```
Algorithm 1 Connected Region Extraction Algorithm
Input:
    The result of moving detection on the image;
Output:
    The coordinates of bounding boxes of moving objects;
1: Sum the pixel values of each column;
2: Divide the image vertically corresponding to non-zero columns. Set the threshold to determine how many pixels apart segmented object can be merged. If the number of all-zero columns is less than the threshold, the columns are considered to be connected to the adjacent non-zero columns. Obtain the list of index of columns with objects;
3: Sum the pixel values of each row in every continuous non-zero columns;
4: Divide the region horizontally using the same method in step 2;
5: Sum the pixel values of each column for every region obtained in step 4. Refine the horizontal coordinates of regions using the method in step 2;
6: return coordinates of bounding box of each connected region;
```

After extension, the regions will be further detected. Anchors are significantly important for modern object detection pipelines. Considering the fact that the moving region obtained by coarse-grained detection need to be extended and resized to the input size of network, the approach to obtain anchor boxes in YOLOV3 is modified to improve the detection performance. Instead of using dimension clusters, anchor boxes are obtained by clustering with the dimension ratio of the object to the image (relative dimension clusters).

Assuming that the upper-left coordinate of the obtained region in coarse-grained detection is (x_0, y_0). In the fine-grained detection stage, more accurate detection is completed. The coordinates are represented as $[(x_1, y_1), (x_2, y_2)]$. Finally, the coordinates are mapped to the original image as $[(x_{\min}, y_{\min}), (x_{\max}, y_{\max})]$ which is defined in Eq. (3).

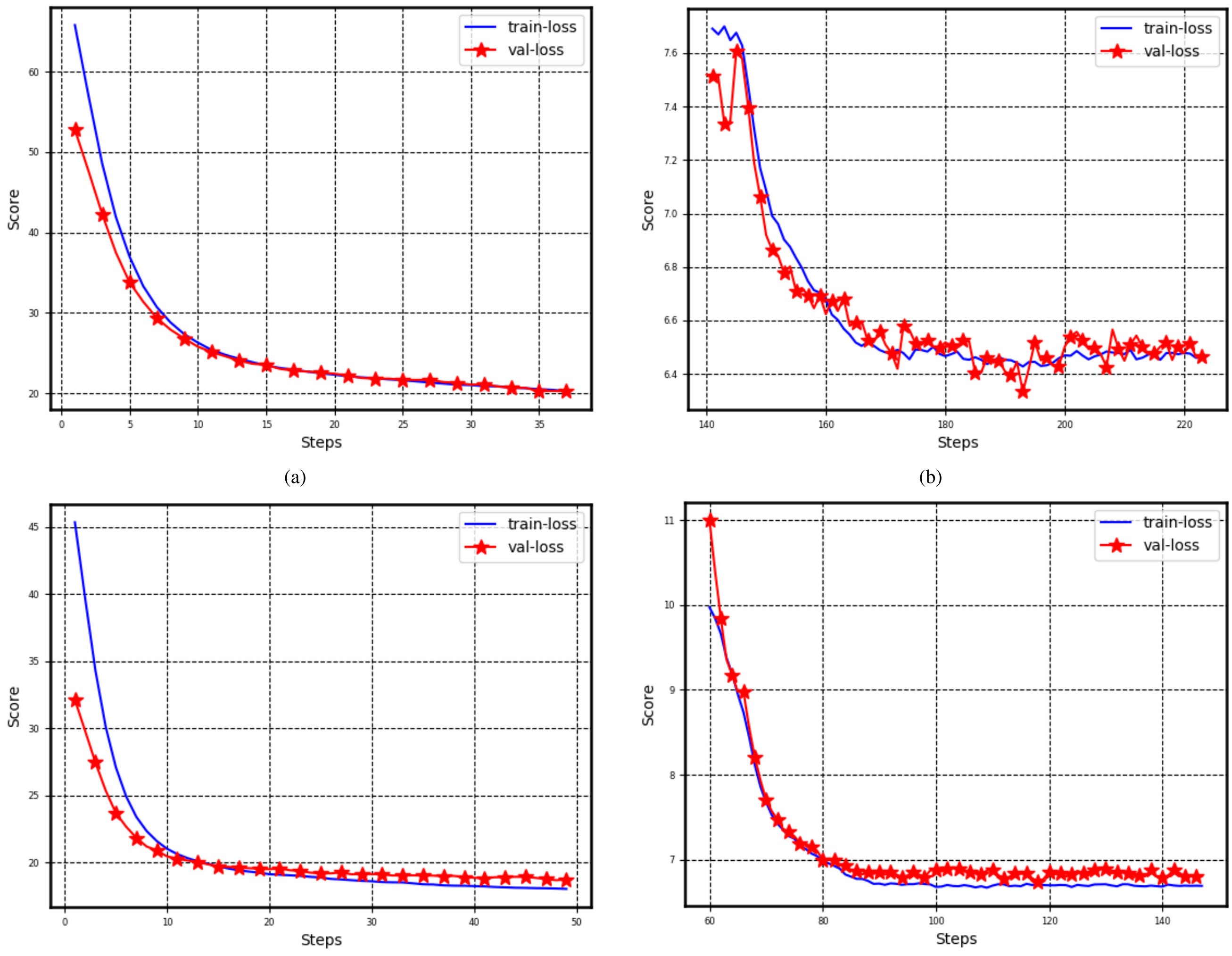
$$x_{\min} = x_1 + x_0, y_{\min} = y_1 + y_0,$$
$$x_{\max} = x_2 + x_0, y_{\max} = y_2 + y_0. \tag{3}$$

Detecting every moving region sequentially is relatively time-consuming. In order to improve the computational efficiency, for fine-grained detection, the region including all moving regions is detected by once if the number of objects is greater than the threshold (T0). Setting different thresholds can achieve a trade-off between the detection accuracy and computational efficiency. However, It is still hard to achieve a sufficiently high speed.

Therefore, we refine tiny version of YOLOV3 for fine-grained detection to get faster speed. The main structure of our modified Tiny YOLOV3 (MTiny YOLOV3) are essentially the same as that of Tiny YOLOV3. Therefore, MTiny YOLOV3 inherits from Tiny YOLOV3 those attributes desirable for object detection. However, the main difference is that, in MTiny YOLOV3, the input size is $96 * 96$ and the number of anchors per grid is 2. By slightly sacrificing some accuracy, we get faster speed and less consumption of computing resources.

IV. EXPERIMENTAL ANALYSIS

In this section, we evaluate the proposed coarse-to-fine grained framework detailly. Detection accuracy is measured in terms of mean Average Precision (mAP). The execution



(a)

(b)

(c)

(d)

FIGURE 8. **Training and validation loss of YOLOV3 (top row) and our framework (bottom row).**

time is measured on a standard GPU (NVIDIA GeForce GTX 1080TI), if not otherwise specified.

A. DATASET

We perform experiments on SimitMovingDataset for moving object detection. SimitMovingDataset[1] is collected and annotated by our laboratory members. SimitMovingDataset contains various changing scenarios, such as multi-scale objects, camouflage, occlusion, illumination changes, dust trailing, complex background and local motion (e.g. waving trees and drifting clouds). From Figure 3, some of the complexities of the dataset can be glimpsed, the resolution of which is $1920 * 1080$.

B. COARSE-GRAINED DETECTION

To examine the computational efficiency of coarse-grained detection, we tested frame difference with low pass filtering and mathematical morphology in terms of runtime, as shown in Table 1.

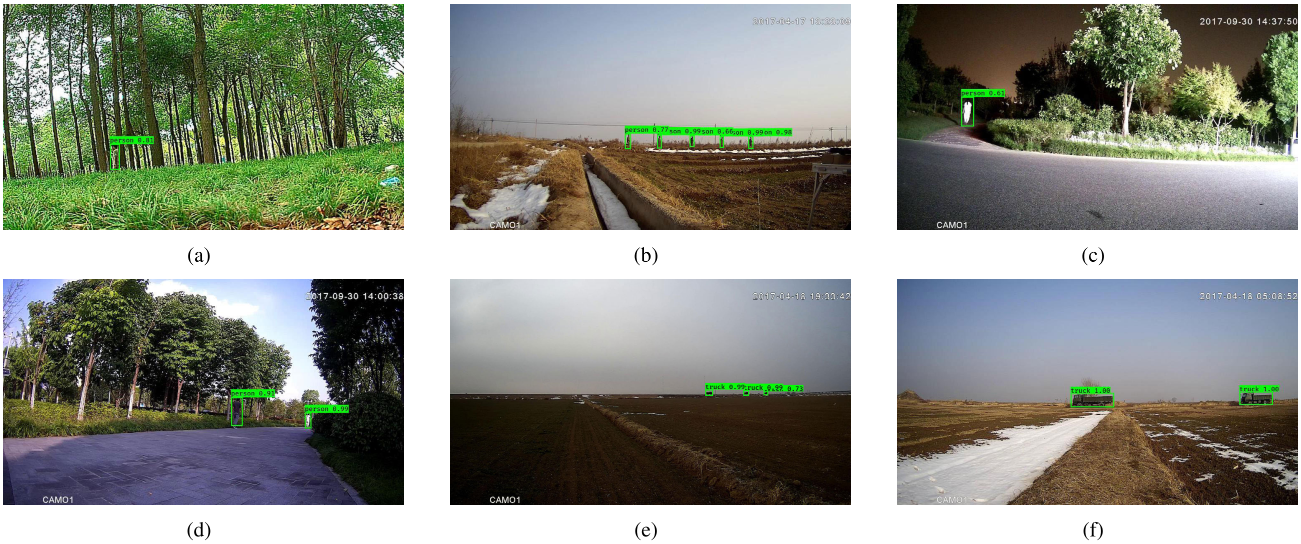
In order to further investigate the effect of each step in coarse-grained detection, Figure 4 shows visual results of each step including low pass filtering, frame difference and mathematical morphology. It is observed that each component in coarse-grained detection plays its own role.

- Ablation Study

1) moving detection algorithm

We compare the moving detection algorithms including frame difference, GMM and KNN in coarse-grained detection. The computational time comparison is shown in Table 1. From Table 1, it is observed that frame difference runs faster than the other two methods. Figure 5 illustrates the visual analysis of these methods. From the results in Figure 5, we observe that: (1) With filtering and mathematical morphology operations, coarse-grained detection with frame difference are not inferior to GMM and KNN unless in some complex situations such as large local motion. (2) Frame difference is more responsive to motion which is more suitable

[1] Since we intend to detect bounding boxes and categories of moving objects, Pascal VOC Annotation Rules are used to annotate SimitMovingDataset.



(a) (b) (c) (d) (e) (f)

FIGURE 9. Visual results of proposed framework.

TABLE 1. Results of different moving detection algorithms. Average time consumed of processing a batch of image frames is obtained on Intel core i7-6700.

Method	frame difference	GMM	KNN
time(s)	0.028	0.123	0.100

TABLE 2. Results of our framework with different moving detection algorithms.

Method	MAP(%)	Time(s)
framework with frame different	88.59	0.112
framework with GMM	78.28	0.229

for the later fine-grained detection. (3) In the situations with large illumination changes, the performance of all the methods are very poor, but frame difference method has a fast adaptability to illumination changes ((e),(f)). From Table 1 and Figure 5, it is shown that results of GMM and KNN are similar. Therefore, KNN is not discussed later.

To further investigate the detection accuracy and execution time of frame difference, experiments are performed on the comparison between our framework with frame difference and that with GMM. As shown in Table 2, the results show great advantages of frame difference over GMM. We suppose that the improvement not only comes from the efficiency of frame difference, but also results from the responsiveness of frame difference.

2) Filtering and mathematical morphology

We omit different components in coarse-grained detection to study the effectiveness of each component, including low pass filtering and mathematical morphology. Visual results are shown in Figure 6. We observe that: (1) Without filtering and morphology operation, the results are seriously disturbed by noise. (2) Low pass filtering module eliminates the influence of high frequency noise. (3) Opening operation of mathematical morphology removes isolated small noisy regions to further suppress the adverse impact of noises. The necessity of filtering and mathematical morphology operations is reflected in their effectiveness of suppress the influence of noises.

TABLE 3. Results of our method and the method from the package skimage based on a standard CPU (Intel core i7-6700).

method	Average time of one thousand times(s)
our method	0.001
another method	0.021

C. CONNECTED REGION DETECTION

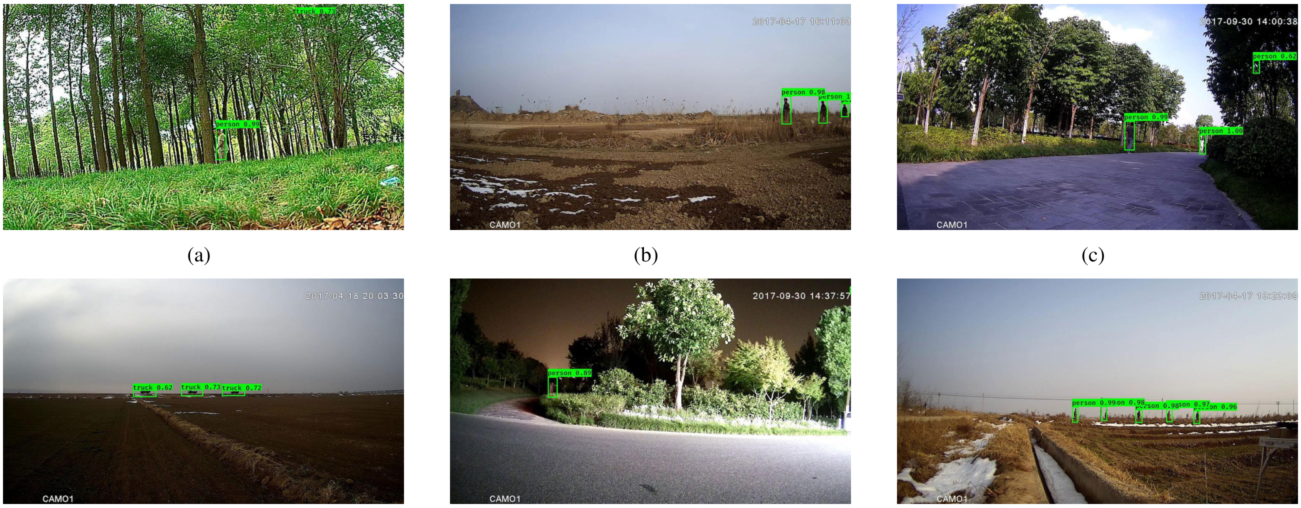
To study the computational efficiency of our proposed connected region detection algorithm, experiments are performed on the comparison between our algorithm and the connected region labeling method from the package skimage which is based on 4-connected method or 8-connected method. Results in Table 3 illustrate that our proposed method performed approximately 21 times faster than the method from the package skimage. Visual results[2] are shown in Figure 7 which illustrates that using our proposed method can merge the segmented object while the other method can not.

D. FINE-GRAINED DETECTION

1) FRAMEWORK WITH YOLOV3 IN FINE-GRAINED DETECTION

In this experiment, the proposed framework is evaluated with YOLOV3 in fine-grained detection in terms of mAP and

[2] To suppress the ill effect of noises, regions are discarded if its area < 50 and compactness (number of target pixels / number of all pixels in the region) < 0.2.



(a) (b) (c)

(d) (e) (f)

FIGURE 10. Visual results of the framework with MTiny YOLOV3.

TABLE 4. Results on SimitMovingDataset in terms of mAP and runtime.

Network	mAP(%)	time(s)
YOLOV3	81.73	0.165
our framework with YOLOV3	88.59	0.112

runtime. The results in Table 4 illustrate that our framework outperforms YOLOV3 by a large proportion in terms of detection accuracy and execution time. In addition, YOLOV3 [31] can not detect motion, which need to be further detected. From Figure 8, we observed that the proposed framework took fewer iterations to stabilized, which make it possible to perform more efficient training.

The effectiveness of the proposed framework is further illustrated by the visual results in Figure 9. Based on the qualitative results, it is evident that the detection results are encouraging even on some complex scenarios such as camouflage, multi-scale objects, local motion, etc.

- Ablation Study

1) Clustering methods

To reveal the performance of the approach to obtain anchor boxes, in terms of detection accuracy and execution time, experiments are performed on dimension clustering and relative dimension clustering. From Table 5, the relative dimension clustering outperforms the other method in terms of accuracy, which brings a gain of 5.1%. The importance of relative dimension clustering is reflected in its usefulness of obtaining more adaptable anchor boxes leading to a significant improvement in accuracy.

2) Speed up

In order to reflect the influence of T0 (defined in Section III-C) on efficiency and accuracy, we perform experiments on differnet thresholds which yeild different maximum number of regions detected sequentially on one video frame. From Table 6, it is observed that setting the threshold brings some improvement in terms of computational efficiency by sacrificing some accuracy.

TABLE 5. Results of two clustering methods in fine-grained detection step.

Clustering approach	mAP(%)	time(s)
Dimension clusters	84.30	0.122
Relative dimension clusters	88.59	0.112

TABLE 6. Results of different threshold T0.

T0	mAP(%)	time(s)
50	88.59	0.112
10	84.24	0.105
8	81.71	0.103
6	73.34	0.093
4	66.13	0.081
0	48.24	0.059

2) FRAMEWORK WITH MTINY YOLOV3 IN FINE-GRAINED DETECTION

In order to further study the computational efficiency of our method, we evaluate the framework with MTiny YOLOV3 instead of YOLOV3 in fine-grained detection. We present the comparison in Table 7 in terms of computational efficiency and detection accuracy. Table 7 illustrates great advantages of our framework with MTiny YOLOV3 over that with YOLOV3 in terms of computational efficiency. Simplified network leads to a lower accuracy, but will improve the performance in terms of efficiency by a large proportion. It can be noted that the framework with MTiny YOLOV3 performed 2.6 times faster than the framework

TABLE 7. Results of the framework with MTiny YOLOV3 and YOLOV3.

Network	time(s)	mAP(%)	input size
our framework with YOLOV3	0.112	88.59	416*416
our framework with MTiny YOLOV3	0.043	80.77	96*96

TABLE 8. Results of different extending methods.

extending methods	mAP(%)	time(s)
random extending	78.72	0.049
extending according to the anchor boxes	80.77	0.043

with YOLOV3, with a satisfying accuracy. The significant improvement on computational efficiency offers a possibility to achieve realtime field unattended monitoring systems. Our results are supported by visual results in Figure 10.

- Extending methods

To examine the advantage of the proposed extending method according to the anchor boxes, experiments compared with random extending are performed in terms of detection accuracy and execution time. From the results in Table 8, we observed the following points:

1) The proposed method improves the detection accuracy significantly.
2) The importance of proposed extending method is further reflected in its computational efficiency.

We also try to introduce it into the framework with YOLOV3, but performance of the two methods are similar. YOLOV3 has a stronger learning ability. We do not need to design it artificially.

V. CONCLUSION

This paper addresses the problems associated with moving object detection on high resolution scenarios for open country, with different kinds of challenging scenes such as local motion, camouflage, complex background, dust trailing, illumination variation and so on.

We have presented a coarse-to-fine grained framework and evaluated its effectiveness with extensive experiments. Considering the segmented object, the use of an efficient connected region detection algorithm we proposed gives it a significant advantage to merge segmented regions. To further improve the performance in terms of computational efficiency, we replace YOLOV3 to MTiny YOLOV3, which achieve 2.6 times faster speed with slightly sacrifying the accuracy. Specifically, our framework with MTiny YOLOV3 and YOLOV3 is able to achieve approximately 23 FPS (Frames Per Second) with 80.77% accuracy and 9 FPS with 88.95% accuracy, respectively.

REFERENCES

[1] Z. Wang, X. Sun, W. Diao, Y. Zhang, M. Yan, and L. Lan, "Ground moving target indication based on optical flow in single-channel SAR," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 7, pp. 1051–1055, Jul. 2019.

[2] Y. Xin, J. Hou, L. Dong, and L. Ding, "A self-adaptive optical flow method for the moving object detection in the video sequences," *Optik*, vol. 125, no. 19, pp. 5690–5694, Oct. 2014.

[3] H. Sajid and S.-C.-S. Cheung, "Universal multimode background subtraction," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3249–3260, Jul. 2017.

[4] S. Jiang and X. Lu, "WeSamBE: A weight-sample-based method for background subtraction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 9, pp. 2105–2115, Sep. 2018.

[5] Z. Zhong, B. Zhang, G. Lu, Y. Zhao, and Y. Xu, "An adaptive background modeling method for foreground segmentation," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 5, pp. 1109–1121, May 2017.

[6] P.-L. St-Charles, G.-A. Bilodeau, and R. Bergevin, "SuBSENSE: A universal change detection method with local adaptive sensitivity," *IEEE Trans. Image Process.*, vol. 24, no. 1, pp. 359–373, Jan. 2015.

[7] G. Shi, J. Suo, C. Liu, K. Wan, and X. Lv, "Moving target detection algorithm in image sequences based on edge detection and frame difference," in *Proc. IEEE 3rd Inf. Technol. Mechatronics Eng. Conf. (ITOEC)*, Oct. 2017, pp. 740–744.

[8] X. Ou, P. Yan, Y. Zhang, B. Tu, G. Zhang, J. Wu, and W. Li, "Moving object detection method via ResNet-18 with encoder–decoder structure in complex scenes," *IEEE Access*, vol. 7, pp. 108152–108160, 2019.

[9] P. W. Patil and S. Murala, "MSFgNet: A novel compact end-to-end deep network for moving object detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 11, pp. 4066–4077, Nov. 2019.

[10] W. Wang, J. Shen, and L. Shao, "Video salient object detection via fully convolutional networks," *IEEE Trans. Image Process.*, vol. 27, no. 1, pp. 38–49, Jan. 2018.

[11] Y. Chen, J. Wang, B. Zhu, M. Tang, and H. Lu, "Pixelwise deep sequence learning for moving object detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 9, pp. 2567–2579, Sep. 2019.

[12] C.-W. Liang and C.-F. Juang, "Moving object classification using a combination of static appearance features and spatial and temporal entropy values of optical flows," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 6, pp. 3453–3464, Dec. 2015.

[13] T. S. Haines and T. Xiang, "Background subtraction with dirichletprocess mixture models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 4, pp. 670–683, Apr. 2014.

[14] Z. Bian and X. Dong, "Moving object detection based on improved Gaussian mixture model," in *Proc. 5th Int. Congr. Image Signal Process.*, Oct. 2012, pp. 109–112.

[15] J. Zuo, Z. Jia, J. Yang, and N. Kasabov, "Moving target detection based on improved Gaussian mixture background subtraction in video images," *IEEE Access*, vol. 7, pp. 152612–152623, 2019.

[16] L. He, X. Zhao, Y. Chao, and K. Suzuki, "Configuration-transition-based connected-component labeling," *IEEE Trans. Image Process.*, vol. 23, no. 2, pp. 943–951, Feb. 2014.

[17] B. Bataineh, "A fast and memory-efficient two-pass connected-component labeling algorithm for binary images," *Turkish J. Elect. Eng. Comput. Sci.*, vol. 27, no. 2, pp. 1243–1259, 2019.

[18] L. He, X. Ren, X. Zhao, B. Yao, H. Kasuya, and Y. Chao, "An efficient two-scan algorithm for computing basic shape features of objects in a binary image," *J. Real-Time Image Process.*, vol. 16, no. 4, pp. 1277–1287, Aug. 2019.

[19] J. Chen, K. Nonaka, H. Sankoh, R. Watanabe, H. Sabirin, and S. Naito, "Efficient parallel connected component labeling with a coarse-to-fine strategy," *IEEE Access*, vol. 6, pp. 55731–55740, 2018.

[20] Y. Jang, J. Mun, K. Oh, and J. Kim, "Block-Based connected component labeling algorithm with block prediction," in *Proc. 40th Int. Conf. Telecommun. Signal Process. (TSP)*, Jul. 2017, pp. 578–581.

[21] D. J. C. Santiago, T. I. Ren, G. D. C. Cavalcanti, and T. I. Jyh, "Fast block-based algorithms for connected components labeling," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 2084–2088.

[22] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[23] Y. Yu, K. Zhang, L. Yang, and D. Zhang, "Fruit detection for strawberry harvesting robot in non-structural environment based on mask-RCNN," *Comput. Electron. Agricult.*, vol. 163, Aug. 2019, Art. no. 104846.

[24] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[25] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

[26] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," in *Proc. 2018 IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6154–6162.

[27] Y.-L. Chang, A. Anagaw, L. Chang, Y. Wang, C.-Y. Hsiao, and W.-H. Lee, "Ship detection based on YOLOv2 for SAR imagery," *Remote Sens.*, vol. 11, no. 7, p. 786, Apr. 2019.

[28] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[29] G. Li, Z. Song, and Q. Fu, "A new method of image detection for small datasets under the framework of YOLO network," in *Proc. IEEE 3rd Adv. Inf. Technol., Electron. Automat. Control Conf. (IAEAC)*, Oct. 2018, pp. 1031–1035.

[30] W. Lan, J. Dang, Y. Wang, and S. Wang, "Pedestrian detection based on YOLO network model," in *Proc. IEEE Int. Conf. Mechatronics Autom. (ICMA)*, Aug. 2018, pp. 1547–1551.

[31] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*. [Online]. Available: https://arxiv.org/abs/1804.02767

[32] Z. Wang, L. Du, J. Mao, B. Liu, and D. Yang, "SAR target detection based on SSD with data augmentation and transfer learning," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 1, pp. 150–154, Jan. 2019.

[33] X. M. Xie, X. Xu, L. H. Ma, G. M. Shi, and P. F. Chen, "On the study of predictors in single shot multibox detector," in *Proc. Int. Conf. Video Image Process.*, 2017, pp. 186–191.

[34] C. F. Xu, B. Bo, Y. Liu, and F. B. Tao, "Detection method of insulator based on single shot multibox detector," in *Proc. J. Phys. Conf.*, vol. 1069, 2018, Art. no. 012183.

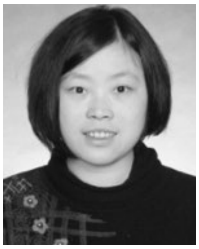
[35] K. Zhao, X. Ren, Z. Kong, and M. Liu, "Object detection on remote sensing images using deep learning: An improved single shot multibox detector method," *J. Electron. Imag.*, vol. 28, no. 03, p. 1, Jun. 2019.

[36] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.

[37] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6517–6525.

HAIDI ZHU received the B.S. degree in electronic information science and technology from Nantong University, Nantong, China, in 2017. She is currently pursuing the Ph.D. degree with the Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, Shanghai, China. Her research interests include computer vision, pattern recognition, and moving object detection.

XIN YAN received the M.S. degree in optical engineering from the Shanghai Institute of Optics and Fine Mechanics, Chinese Academic and Sciences, Shanghai, China, in 2015. He is currently an Engineer with the Shanghai Institute of Microsystem and Information Technology, Chinese Academic and Sciences. His research interests include computer vision and deep learning.



HONGYING TANG received the Ph.D. degree from Shanghai Jiao Tong University (SJTU), China, in 2015. She is currently an Engineer with the Science and Technology on Microsystem Laboratory, Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences. Her research interests include unmanned aerial vehicle (UAV) communications, linear transceiver design, and wireless sensor networks.

YUCHAO CHANG received the joint Ph.D. degree from the Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, and the University of Chinese Academy of Sciences, China, in 2019. He joined the Prof. Naofal Al-Dhahir Group, The University of Texas at Dallas, as a Visiting Ph.D. Student, in 2018. His research interests include wireless communications and machine learning applications.

BAOQING LI received the Ph.D. degree from the State Key Laboratory of Transducer Technology, Shanghai Institute of Metallurgy, Chinese Academy of Sciences, Shanghai, China, in 2000. He is currently a Professor and a Ph.D. Supervisor with the Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences. His research interest includes the application of wireless sensor networks.

XIAOBING YUAN received the Ph.D. degree from the Changchun Institute of Optics, Fine Mechanics, and Physics, Chinese Academy of Sciences, Shanghai, China, in 2000. He is currently a Professor and a Ph.D. Supervisor with the Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences. His research interests include wireless sensor networks, information transmission, and processing.