

Detect and Calculate the Speed of Moving Objects Using R-CNN Technique

Batool R. Abd¹ and Heba Kh. Abbas^{2*}

^{1,2}Department of Physics, College of Science for Women, University of Baghdad, Baghdad, Iraq.

*Correspondence Author.

Received: 25 May 2024

Accepted: 29 August 2024

Published: 20 April 2025

doi.org/10.30526/38.2.4020

Abstract

Artificial intelligence (AI) algorithms depend on different algorithms to track moving objects. The surrounding environment, like lightness, affects detection accuracy. A new algorithm is designed to detect moving objects in real time automatically and calculate the speed of tested objects based on the deep learning algorithm RCNN using MATLAB software. The suggested system consists of a phone camera, four balls in different colors (red, green, blue, and black), and various environmental lights consisting of eight lights. Two luxmeters were used to check the light intensity around the ball and the camera phone. Detecting moving objects has gained a lot of desirability because of its applications like video surveillance, person movement tracking, traffic investigation, and security systems like security systems or surveillance. are four parameters used to evaluate the performance of the algorithms and the system setup: accuracy, average time, detection percentage, and speed. Results show that the quality of detecting and tracking the ball is almost 100%.

Keywords: Detect moving object, RCNN, Algorithm in MATLAB software, Artificial intelligence.

1. Introduction

Over recent years, moving object detection has become the task of determining the physical movement of an object in a specific area or region (1). Detecting moving objects has gained a lot of desirability because of its applications like video surveillance, person movement tracking, traffic investigation, and security systems like security systems or surveillance (2). Detecting moving objects is used in computer vision and video processing fields because complex processes can be explored and solved like video object classification and tracking (3). Thus, determining the exact shape of a tested moving object from a tested video frame becomes easy to implement (4). However, detecting the precise shape of the tested moving object has challenges such as dynamic scene changes, lightness changes, shadow, opacity, and homogeneity problems (5). The benefit is to reduce the impact of these issues, and surveillance systems have gained importance because of the increasing demand (6). Researchers studied identifying and detecting moving objects and targets. The following



works were chosen as a selective choice due to their proximity to our research field. Manikandaprabu et al. presented a method to detect and track human beings in 2021. A continuously Adaptive Mean Shift algorithm is used to track the moving person. The background is subtracted using the Region of Interest (ROI) and a centroid pixel point over the ROI to be used by the Continuously Adaptive Mean Shift algorithm. Only three different lights were used, and there was a problem with a reduction in time complexity (7). K N V Khasim et al. used RADAR and LIDAR devices for traffic applications in 2022. Video sequence frames are used to subtract and mask the moving objects. Therefore, the speed of any moving target speed calculated in real time without any sensor calibration (8). S.M. Tilon and F. Nex used Unmanned Aerial Vehicles (UAV) videos that can be deployed on UAV-embedded edge devices to track vehicles in 2023. NVIDIA Jetson Xavier NX edge device used which could be executed at 8 Frames Per Second (FPS). Multi-segmentation regions are used to track the vehicles without affecting road details. The speed object was calculated using the VisDrone Multi-Object. Tracking (MOT) benchmark dataset. There was an error while they measured vehicle speed (9). The suggested work aims to use a deep learning algorithm RCNN using MATLAB software to track a moving object consisting of four balls with different colors, with the lightness of the environment controlled using eight lights, and intensity measured using two lux meters.

2. Materials and Methods

2.1. Region-Based Convolutional Neural Network (R-CNN)

R-CNN is classified as one of the deep learning methods used to track any object in the computer vision method (10,11). It depends on combining the convolutional neural networks and region-based approaches to detect the object. The target image is divided into sub-regions or multi-regions that have objects within them, using selective search or edge boxes to generate these sub-regions (12,13). Then features are extracted by the CNN algorithm by generating 2000 regions as input images. These regions are resized to be 16 pixels for warped frames. The high dimensional feature that resulted from CNN represents the region that is used for feature extraction in a separate machine-learning classifier. Support Vector Machines (SVM) are used for object classification to determine whether the region proposal contains an instance of that class. There are positive and negative samples in the training step. R-CNN eliminates duplication or overlapping bounding boxes in the classifying step. This led to high object detection as a final step (14-16).

2.2. Quality metrics

Quality and performance measures were relied upon in deep learning and data analysis, as the study included several efficient measures in evaluating the quality of model performance. Four scales were used.

1-The first measure is accuracy, which was calculated based on four factors (17-19).

- T_P (true positive): A test result that correctly indicates the presence of the object.
- T_N (true negative): A test result that correctly indicates the absence of any object.
- F_P (false positive): A test result that falsely indicates the presence of the object.
- F_N (false negative): A test result that incorrectly indicates the absence of any object. The parameters are specified as shown in **Table 1**.

Table 1. Quality metrics parameters.

Actual classification	1	1	0	0
Predicated classification	0	1	0	1
result	F_N	T_P	T_N	F_P

Accuracy is a metric for evaluating the performance of classification models. It is the ratio of correct predictions to the total number of predictions. For binary classification, accuracy can also be calculated in terms of positives and negatives as shown in equation (1)(20-22):

$$\begin{aligned} \text{Accuracy (Acc)} \\ = \frac{T_P + T_N}{T_P + T_N + F_P + F_N} \end{aligned} \quad (1)$$

2-Detection is determined in each video frame (30 f/s). If the ball is detected, then the ratio is the ball frame over the total number of frames. The detection rate equation is (23):

$$\begin{aligned} \text{detection rate} \\ = \frac{T_P}{T_P + F_N} \end{aligned} \quad (2)$$

Real-time Detection is programmed within MATLAB software to identify and locate objects of interest in real-time video sequences. The time is extracted as data within the code and presented accordingly(24). 3-Speed of moving object is determined by calculating the displacement between two reference points on the path of the moving object as a function of time, as given in equations (3 & 4). The difference was measured as: $dX = X_{i+1} - X_i$ and $dy = y_{j+1} - y_j$ (25-27):

$$dr = \sqrt{dX^2 + dy^2} \quad (3)$$

$$\text{speed} = \frac{dr}{dt} = \frac{dr}{\frac{1}{30}} = dr \times 30 \quad (4)$$

The speed was calculated in an image using the relationship:

$$\text{speed} = \sqrt{dX^2 + dy^2} \times 30 \frac{\text{pixel}}{s} \quad (5)$$

The actual speed was calculated based on the scale factor:

$$\text{scale factor} = \frac{\text{speed} \left(\frac{m}{s} \right)}{\text{speed} \left(\frac{\text{pixel}}{s} \right)} \quad (6)$$

Therefore, if we need to know the speed in m/s then we multiply 0.2419 m/pixel by speed (pixel/sec).

2.3. Movie Object Detection Algorithms

The object detection process includes four algorithms: cropping process(28), labelling(29,30), training data and detection, and testing(20). The cropping process depends on calculating the video length. In MATLAB software, the function `imcrop` is used in one frame to be a reference for the remaining frames as a detected process. This step enables cropping any object using a PC pointer with any size and position with a rectangle shape and saving the divided frames in a selected folder using write code in MATLAB. The second step is image labeling used to delimit the region of interest. The code of the `image datastore` function in MATLAB is used to generate properties of the select object and the `imageLabeler` property is used to label ground truth in images. Training data and detection algorithm used as the third step by knowing the number of layers (iz) and the number of training times (epoch). The number of layers is 64 and the number of training times epoch =50. The training file and the model are stored with the same name based on the time in terms of hours and seconds. The specific folder is selected and labeled as the file (ball. mat). The training process began as instructed based on the positive layers option `[dete, info] = train RCNN Object Detector (pstv2, layers, options)`. The model is obtained by stored it and stored information about the model where the date represents the model contains loss and accuracy. The testing algorithm is used to detect objects as a final step.

A folder selected and selected all images from 1 to number of images. If there is a target, then detect it and put box, score, and label. After detection, the target is deducted with a yellow square placed on each detected target. This means the score is greater than 9 otherwise it is not detected. The center (x, y) is calculated because the object is moving and estimated accuracy. The extraction time, detection time, total time, and average time in seconds are computed, and finally, the calculation of object speed.

3. Results and Discussion

Four balls with different colors (red, blue, green, and black) were used as a test target to get the perfect model for detection. For this purpose, the iPhone 14 Pro Max mobile camera is used to record 96 videos divided into three videos for each ball and eight different light intensities. The intensity details are listed in **Table 2** and distributed in the lab, as shown in **Figure 1**. The lux meter is located in two positions, next to the phone camera and close to the wall. The resulting 96 videos were cropped into 1036 frames, the number of layers (iz = 46), and the number of training times (epoch = 50). This helps us to generate a model to detect any moving object.

Table 2. The lightness intensity with the number of lights.

Number of lamps	Lux1 (Next to the camera)	Lux2 (Next to the wall)
0	2	2
1	32	16
2	63	41
3	101	63
4	127	81
5	198	127
6	203	198
7	253	158
8	317	127

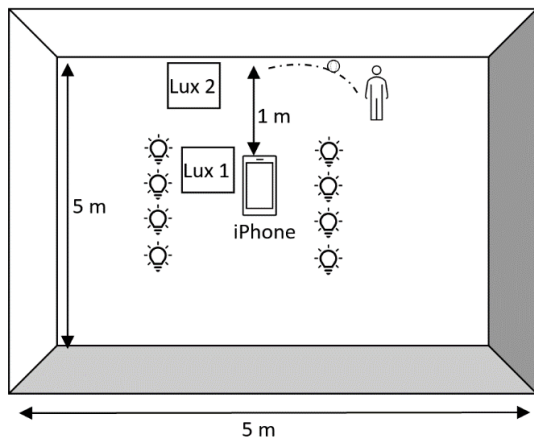


Figure 1. Diagram of the suggested system showing light distribution and object location.

3.1. Detection percentage

Figure 2 shows the detection percentage for each colored ball regarding the light intensity at the best number of layers (iz=64) and the best training number (epoch=50) for three videos. Lighting is an important parameter used to detect moving objects. Therefore, this study focused on this parameter and measured the detection of different colored balls to check the quality of the suggested system. It is noticeable that at all light intensities, there is a good detection percentage for all cases. This means that the resulting model succeeded in this purpose.

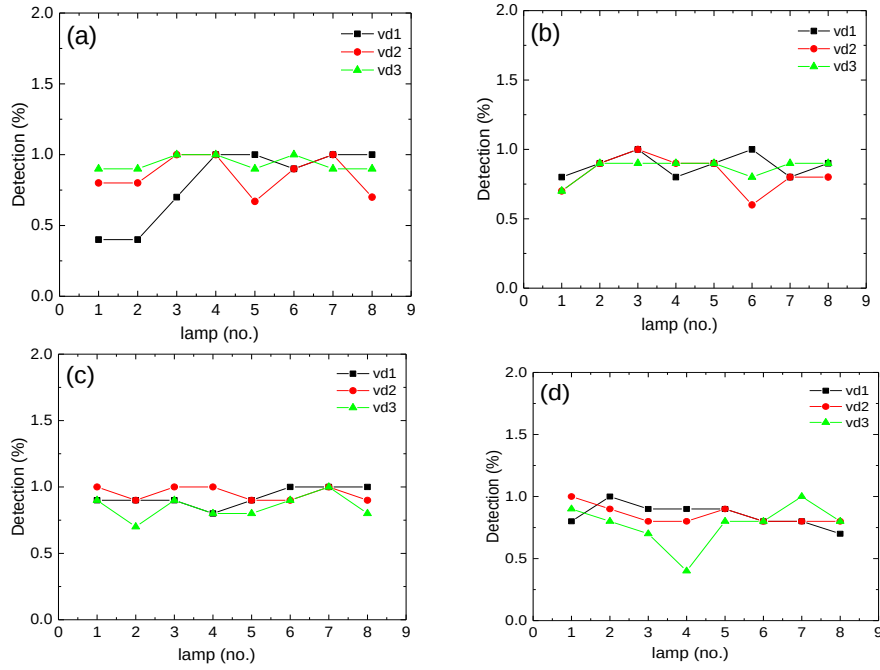


Figure 2. Detection percentage as a function of lamp number for the ball (a) red, (b) blue, (c) green, and (d) black.

3.2. Accuracy Percentage

The accuracy of the suggested model is calculated depending on Equation 1. **Figure 3** shows the accuracy of four colored balls for three videos. All colored balls have high accuracy, even with the lightness change.

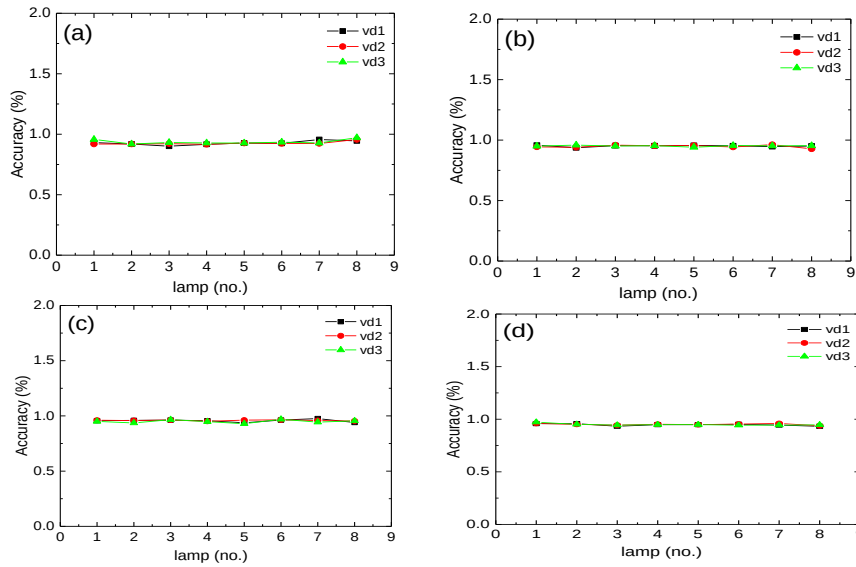


Figure 3. Accuracy percentage as a function of lamp number for ball (a) red, (b) blue, (c) green, and (d) black.

3.3. Average Time

The average time of the proposed model is calculated to show the performance of the suggested model and system. **Figure 4** shows the average time of four colored balls for three videos. All colored balls have almost the same behavior with the lightness change.

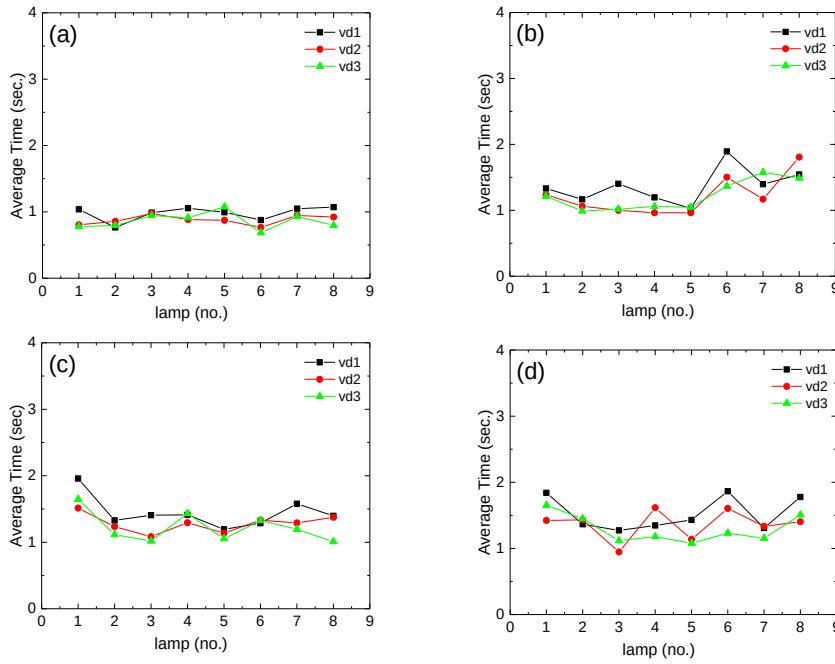


Figure 4. Average time as a function of lamp number for ball (a) red, (b) blue, (c) green, and (d) black.

3.4. Speed

The speed of the suggested model is calculated depending on equation 6. **Figure 5** shows the speed of four colored balls for three videos. All colored balls have almost the same speed because the person was the same and used to through the ball in the same way with the lightness change.

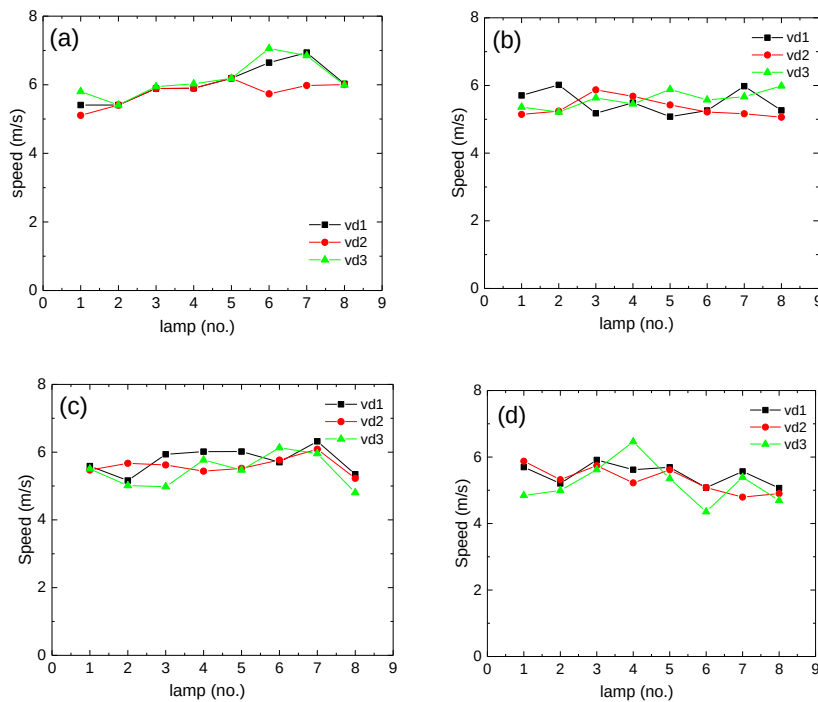


Figure 5. Speed as a function of lamp number for ball (a) red, (b) blue, (c) green, and (d) black.

4. Conclusion

The suggested system with the designed algorithm gave a clear view of detecting and calculating a moving object. There are many parameters used to evaluate the performance of

the suggested method. The detection percentage is acceptable at around 90% because the lightness is changed all the time and is considered a challenging parameter in this work. The speed of the object was almost fixed at 6 m/s because the user is the same for all cases. The time of detection is around 1.5 s, and the difference is due to the time it takes for the software to cut the video frames and track the object. The detection accuracy is nearly 100% for all objects, indicating the success of this suggested system.

Acknowledgment

Thanks, and appreciation to everyone who contributed to helping me, and I thank my supervisor who contributed greatly to this work as well.

Conflict of Interest

The authors declare that they have no conflicts of interest.

Funding

There is no funding received for this work.

References

1. Chen W, Luo J, Zhang F, Tian Z. A review of object detection: Datasets, performance evaluation, architecture, applications and current trends. *Multimedia Tools Appl.* 2024. <https://doi:10.1007/s11042-023-17949-4>.
2. Cui B, Créput J-C. A Systematic Algorithm for Moving Object Detection with Application in Real-Time Surveillance. *SN Comput Sci.* 2020;1(2):106. <https://doi:10.1007/s42979-020-0118-5>.
3. Delibaşoğlu İ. Moving object detection method with motion regions tracking in background subtraction. *Signal Image Video Process.* 2023;17(5):2415-2423. <https://doi:10.1007/s11760-022-02458-y>.
4. Chacon-Murguia MI, Guzman-Pando A. Moving Object Detection in Video Sequences Based on a Two-Frame Temporal Information CNN. *Neural Process Lett.* 2023;55(5):5425-5449. <https://doi:10.1007/s11063-022-11092-1>.
5. Yara S, Ganivada A. Detection of Moving Objects and Enhancement Using Motion Features in Various Video Sequences. In: *Proceedings of Innovations in Computer Science and Engineering*; Singapore, 2022. p. 665-674.
6. Abbas HK, Mohamad HJ. Feature Extraction in Six Blocks to Detect and Recognize English Numbers. *Iraqi J Sci.* 2021;62(10):3790-3803. <https://doi:10.24996/ij.s.2021.62.10.37>.
7. Nallasivam M, Senniappan V. Moving Human Target Detection and Tracking in Video Frames. *Stud Inform Control.* 2021;30(1):119-129. <https://doi:10.24846/v30i1y202111>.
8. Khasim KNV, Anilkumar G, Vamshi G, Ch SK, Nikhil Yadav M. Moving Object Detection and Speed Estimation by Digital Image Processing. *J Phys Conf Ser.* 2022;2325:012035. <https://doi:10.1088/1742-6596/2325/1/012035>.
9. Tilon SM, Nex F. Vehicle Tracking and Speed Estimation From Unmanned Aerial Vehicles Using Segmentation-Initialised Trackers. *ISPRS Ann Photogramm Remote Sens Spatial Inf Sci.* 2023;X-1/W1-2023;10:431-437. <https://doi:10.5194/isprs-annals-X-1-W1-2023-431-2023>.
10. Chaudhuri A. Smart traffic management of vehicles using faster R-CNN based deep learning method. *Sci Rep.* 2024;14(1):10357. <https://doi:10.1038/s41598-024-60596-4>.
11. Saraf P, Watve S, Kulkarni A. Object Detection: Literature Review. In: *Proceedings of the 14th International Conference on Soft Computing and Pattern Recognition (SoCPaR 2022)*; Cham, 2023. p. 485-494.

12. Pu C, Ju L, Gao F, Yu J. RatiO R-CNN: An Efficient and Accurate Detection Method for Oriented Object Detection. In: Proceedings of Image and Graphics; Cham, 2023. p. 28-40. <https://doi:10.1088/1757-899X/571/1/012117>.
13. Gorelik AJ, Gorelik MG, Ridout KK, Nimarko AF, Peisch V, Kuramkote SR, Low M, Pan T, Singh S, Nrusimha A. Evaluating efficiency and accuracy of deep-learning-based approaches on study selection for psychiatry systematic reviews. *Nat Ment Health*. 2023;1(9):623-632. <https://doi:10.1038/s44220-023-00109-w>.
14. Noor MH, Heba Kh A, Mohamad HJ. Detection and Recognition of Car Plates in Parking Lots at Baghdad University. *Iraqi J Sci*. 2023;64:1018-1029. <https://doi:10.24996/ij.s.2023.64.2.43>.
15. Abbas HK, Al-Saleh AH, Mohamad HJ, Al-Zuky AA. New algorithms to Enhance Fused Images from Auto-Focus Images. *Baghdad Sci J*. 2021;18(1):0124. <https://doi:10.21123/bsj.2021.18.1.0124>.
16. Awad R, Al-Zuky AA, Al-Saleh AH, Mohamad HJ. Enhance Video Film using Retnrx method. *J Phys Conf Ser*. 2018;1003(1):012124. <https://doi:10.1088/1742-6596/1003/1/012124>.
17. Abbas HK, Mohamad HJ, Al-Saleh AH, Al-Zuky AA. Modelling vision angles of optical camera zoom using image processing algorithm. *IOP Conf Ser: Mater Sci Eng*. 2019;571(1):012117. <https://doi:10.1088/1757-899X/571/1/012117>.
18. Derea AS, Abbas HK, Al-Saleh AH, Mohamad HJ. Adopting Texture Features to Detect and Recognize Brain Tumors in Magnetic Resonance Images. *IOP Conf Ser: Mater Sci Eng*. 2020;757(1):012029. <https://doi:10.1088/1757-899X/757/1/012029>.
19. Rasham NH, Abbas HK, Abdul Razaq AA, Mohamad HJ. Simulation of Speckle Noise Using Image Processing Techniques. In: Proceedings of the Computer Networks and Inventive Communication Technologies; Singapore, 2022. p. 489-501.
20. Jawad Mk, Daway HG, Mohamad HJ, Daway EG. Lightness Enhancement by Fuzzy Logic Depending on Power Membership Function. *J Phys Conf Ser*. 2021;1999:012129. <https://doi:10.1088/1742-6596/1999/1/012129>.
21. Zhan C, Duan X, Xu S, Song Z, Luo M. An improved moving object detection algorithm based on frame difference and edge detection. In: Proceedings of Image and Graphics, Fourth International Conference; IEEE, 2007. p. 519-523.
22. Hu W, Tan T, Wang L, Maybank S. A survey on visual surveillance of object motion and behaviors. *IEEE Trans Syst Man Cybern Part C Appl Rev*. 2004;34(3):334-352. <https://doi:10.1108/1757-899X/571/1/012117>.
23. Murshed M, Kabir MH, Chae O. Moving object tracking-an edge segment based approach. *Int J Innov Comput Inf Control*. 2011;24(4):23-35.
24. Yutaka SS, Kaneko S, Satoru I. Robust object detection and segmentation by peripheral increment sign correlation image. *Trans IEICE*. 2001;84(12):2585-2594.
25. Sudha D, Priyadarshini J. An intelligent multiple vehicle detection and tracking using modified vibe algorithm and deep learning algorithm. *Soft Comput*. 2020;24(22): 17417-17429.
26. Zhang B, Zhou L, Zhang J. A methodology for obtaining spatiotemporal information of the vehicles on bridges based on computer vision. *Comput Aided Civil Infrastruct Eng*. 2019;34(6):471-487. <https://doi:10.1088/1757-899X/571/1/012117>.
27. Liu L. Moving object detection technology of line dancing based on machine vision. *Mobile Inf Syst*. 2021;2021:23-45. <https://doi:10.1088/1757-899X/571/1/012117>.
28. Ji Q, Dai C, Hou C, Li X. Real-time embedded object detection and tracking system in Zynq SoC. *EURASIP J Image Video Process*. 2021;2021(1):12-23.
29. Zhang C, Liang B, Ma S, Xiang C, Bai Z, Wang J. Research on the attribute extraction algorithm of people and vehicles based on video structure. *J Phys Conf Ser*. 2021;1802:34-45. <https://doi:10.1088/1757-899X/571/1/012117>.
30. Zhang W, Sun W. Research on small moving target detection algorithm based on complex scene. *J Phys Conf Ser*. 2021; 1: 012093.