Research on hierarchical pedestrian detection based on SVM classifier with improved kernel function

Yin Zhang¹, Lunhui Xu² and Yikuan Zhang¹

Abstract
The research of pedestrian target detection in complex scenes is still of great significance. Aiming at the problem of high missed detection rate and poor timeliness of pedestrian target detection in complex scenes. This paper proposes an improved classification method. First, Haar features were extracted from the images to be detected, and the candidate areas of pedestrians were determined by Adaboost classifier. Then, the traditional SVM classifier was improved by using the combined kernel function instead of the single kernel function, and the optimal proportion of each function in the combined kernel function was found by using the adaptive particle swarm optimization algorithm. Finally, the improved SVM classifier was combined with the fusion feature to further detect the candidate area to accurately locate the pedestrian’s position. Experimental results show that compared with the traditional detection framework, the proposed method can effectively improve the detection speed and the detection accuracy. This method has certain practical significance for pedestrian target detection in complex scenes.

Keywords
SVM, kernel function, APSO, pedestrian detection

Date received: 5 October 2021; accepted: 6 June 2022

Introduction
The purpose of pedestrian detection is to predict the location of all pedestrians in the images, which provides sufficient technical support for later behavior detection, identity recognition, path tracking, and other technologies.¹

Nowadays, the more commonly used method of pedestrian detection is to extract features based on statistical learning methods to detect after training. Feature extraction methods use computers to extract detailed information of images to be detected and determine whether each image point belongs to an image feature. This method needs to be based on a large number of training samples used to construct a pedestrian detection classifier. Papageorgiou and Poggio² used sliding Windows for pedestrian detection, Viola and Jones³ proposed the integral graph method, in 2005, Dalal and Triggs⁴ proposed a robust description feature: Histogram of gradient (HOG) feature for pedestrian feature description, Ojala et al.⁵ proposed local binary pattern(LBP), which gave the classification results of one-dimensional eigenvalue distribution of single feature and two-dimensional distribution of complementary feature pairs, Wang et al.⁶ combined LBP features with HOG pending features, used the support vector machines(SVM) classifier to complete classification, and proposed feature fusion pedestrian detection algorithm, Felzenszwalb et al.⁷ proposed a target detection based on the multi-scale deformable part model mixing, Felzenszwalb et al.⁸ also proposed a cascade variable model. Due to the good performance of the above features in pedestrian detection, researchers have proposed improved features such as FHOG, LTP, SLBP, ACFFCF, and so on. As Krizhevsky et al.⁹ applied convolutional neural networks to solve large-scale image classification problems and achieved remarkable results, more and more researchers have also paid attention to the research of deep learning detection methods. According to the idea of cascading classifiers in the Adaboost algorithm, Angelova et al.¹⁰ proposed a pedestrian detection algorithm based on cascaded convolutional neural networks, which can
quickly eliminate most of the background area in the images. Ouyang and Wang\textsuperscript{11} proposed a joint deep learning algorithm, which combines HOG features with Cascading Style Sheets (CSS) features and uses SVM classifier classification to design the first-level detector to prefilter the samples, and then use the convolutional neural network makes the next judgment.

Among many pedestrian detection algorithms, the excellent performance of HOG-LBP pedestrian detection fusion feature in detection accuracy and handling the occlusion problem has always been a focus of research. However, the HOG features can not describe the spatial characteristics of gradient well. The binary coding strategy of LBP features makes it more robust to illumination and noise. Moreover, the algorithm complexity of the traditional kernel SVM classifier is relatively large, and the real-time performance of detection also needs to be improved. Based on the HOG-LBP fusion feature and SVM classifier framework, this paper proposes a pedestrian detection algorithm based on the cascaded candidate classifier for the above problems.

In detection, the environment in which pedestrian recognition is more complicated and there are many obstacles.\textsuperscript{12,13} In the algorithm of pedestrian detection, the shape of the human body is changeable, such as standing, squatting or obstructing, and there are differences between individuals due to high and low weight and other reasons, so feature extraction is relatively difficult.\textsuperscript{13}

In pedestrian detection, there is a large amount of noise interference in the images caused by changes in light, which will lead to poor detection accuracy and timeliness.\textsuperscript{14} Therefore, in the process of pedestrian detection, the first step is to process the images to reduce the interference factors in the images. In this paper, before the detection, the images are grayed and normalized.\textsuperscript{15}

The detection is divided into two stages. In the first stage, the region of interest is extracted by using head and shoulder contours. In the second stage, the improved SVM classifier is used to train parameters, and multi-feature fusion is used to re-detect the areas identified as pedestrians in the first stage. This method can greatly reduce the judgment of candidate areas and shorten the detection time.

Materials and methods

In this paper, two-stage detection is used to accurately locate pedestrians in images. In first stage detection, the classifier trained by Haar feature and AdaBoost algorithm is used as a detector to extract the region of interest. Then, feature fusion was used to further enhance the detection accuracy, the fused features were processed by principal component analysis (PCA) for dimension reduction,\textsuperscript{16} and the SVM classifier was trained. The region of interest extracted in the first stage was input into the trained improved SVM classifier for two-stage detection, and the final pedestrian detection result was obtained. The overall detection process is shown in Figure 1.

It makes full use of the high speed of Haar features in pedestrian detection, and quickly selects the candidate areas of head and shoulder.\textsuperscript{17} On the other hand, it uses the fusion features to describe the precise characteristics in pedestrian detection to complete pedestrian head and shoulder detection.

Level 1 Haar + AdaBoost detection scheme

In order to obtain better detection results, partial characteristics of pedestrians can be tested.\textsuperscript{18} The head and shoulders of pedestrians are similar to Ω shape and are relatively stable.\textsuperscript{19} In this paper, the head and shoulder are used as the detection target in the first level detection. On the one hand, it can effectively reduce the impact of occlusion, on the other hand, it can reduce the detection area and improve timeliness.

Table 1 compares the performance of the Adaboost classifier trained by Haar-like features, LBP features and HOG features in detection, and it can be concluded that Haar features have better performance through the comparison of the tests. Haar feature and Adaboost classifier were used to train the first level detector for pedestrian head and shoulder detection. By adding the weight coefficient to the traditional arithmetic average method, the final prediction function becomes:

$$H(x) = \frac{1}{n} \sum_{i=1}^{n} w_i h_i(x)$$

(1)

The constraint function is:
In pedestrian detection, the head and shoulder model is very stable and has excellent contour features.\textsuperscript{19,20} When using contours to detect targets, contour features need to be intact, and ensure the effectiveness of the training features in order to obtain highly accurate detection results. It is a stable and reliable way to detect pedestrians with head and shoulder contours to avoid the occurrence of some occlusion. In this paper, Haar features are used to train AdaBoost classifier, and scale of the obtained strong classifier is changed to accommodate different sizes of images to be detected, and the position of head and shoulder in the image is detected.\textsuperscript{21–23} The Haar feature is the difference between the sum of pixels of two matrices. It is proposed by Paul Viola to calculate Haar features by means of integral graph, which improves the convenience of Haar feature calculations.

The training process is shown in Figure 2. For the existing $N$ training samples, the weights of all samples have the same at the beginning of training, then the initial weight distribution is as follows:

$$D_1 = \left( w_{11}, w_{12}, \ldots, w_{1N} \right); w_{1N} = 1/N \quad (3)$$

In the training, the sample weight of accurate identification is reduced, while the sample weight of wrong identification is increased, and the key training is carried out. The weak classifier $h_m(x): X \rightarrow \{-1, 1\}$ can be obtained by using the data set for training, and the

| Algorithm        | Number of heads and shoulders tested | Detection rate (%) | Omission factor (%) | False positive (%) | Detection speed (s) |
|------------------|--------------------------------------|--------------------|---------------------|--------------------|--------------------|
| Haar + AdaBoost  | 785                                  | 97                 | 1.5                 | 15                 | 0.03               |
| HOG + AdaBoost   | 744                                  | 91                 | 6.7                 | 3                  | 0.92               |
| LBP + AdaBoost   | 723                                  | 80                 | 20                  | 5.5                | 0.12               |

Table 1. Comparison of classifier detection with different features.

Figure 2. Adaboost cascade classifier constitutes a strong classifier.
error rate of the classifier on the training data set is collected:

$$e_m = P(h_m(x_i) \neq y_i) = \sum_{i=1}^{N} w_i \cdot R(G_m(x_i) \neq y_i)$$  \hspace{1cm} (4)

Calculate the coefficient $\alpha_m$ of $h_m(x)$, the weight of the weak classifier can be calculated by the following formula:

$$\alpha_m = \frac{1}{2} \log \left( \frac{1 - e_m}{e_m} \right)$$  \hspace{1cm} (5)

When the strong classifier is trained, the previous best-performing classifier has the largest weight. It can be seen from equation (5), $\alpha_m$ varies from 0 to 1/2. $\alpha_m$ increases with the decrease of $e_m$, which indicates that the classifier with better performance and lower error rate will play a more important role in classification. After the change of the weighting coefficient, the weight distribution is:

$$D_{m+1} = (w_{m+1,1}, w_{m+1,2}, \ldots, w_{m+1,N})$$  \hspace{1cm} (6)

$$w_{m+1,i} = \frac{w_m \cdot \exp \left( -\alpha_m y_i h_m(x_i) \right)}{z_m}, i = 1, 2, \ldots, N$$  \hspace{1cm} (7)

In the formula, $z_m$ is the normalized constant, and $D_{m+1}$ is transformed into probability distribution:

$$z_m = \sum_{i=1}^{N} w_{m,i} \cdot \exp \left( -\alpha_m y_i h_m(x_i) \right)$$

After the changes, the final strong classifier is obtained:

$$G(x) = \text{sign} \left( \sum_{m=1}^{m} \alpha_m h_m(x) \right)$$  \hspace{1cm} (9)

The first stage detector is trained based on AdaBoost algorithm combined with sample features, which requires a large number of positive and negative samples. The detection object in the first stage is the position of pedestrian’s head and shoulder, so the positive sample is the image of pedestrian’s head and shoulder, and the negative sample is the image of non-row human head and shoulder. The positive samples are composed of two parts, one is from the INRIA pedestrian database, and the other are from the actual scene images taken by the network camera of laboratory, from which the head and shoulders of the pedestrians are intercepted. A total of 2360 positive samples were obtained. With respect to the selection of negative samples, a total of 9440 negative samples were obtained by using the negative samples provided in the data set and the non-head with shoulder images captured in the actual scene. By calling the resize function in OpenCV, all the images are normalized to $32 \times 32$ and then graying processing.

The detector based on Haar features obtained by AdaBoost cascade classifier has a fast detection speed. However, in complex backgrounds, other objects in the background may be mistaken for detection targets, such as backpacks that are mistakenly detected as pedestrians. In this paper, based on the first stage detection, the second stage detection is carried out to further improve the accuracy of detection results.

### Level 2 pedestrian detection scheme

The extracted head and shoulder areas were extracted for feature extraction and fusion by secondary detection, and the fused features were then trained on the improved SVM.

In order to balance the globality and locality of the SVM classifier, and make the classifier have the ability of learning and generalization capability, this paper uses the combined kernel function SVM algorithm based on particle swarm optimization algorithm (PSO) to obtain a pedestrian detection classifier with better classification performance. By analyzing the learning and generalization capability of polynomial kernel function and radial basis kernel function, uses linear combination polynomial kernel function and radial
basis kernel function as the kernel function of SVM to construct a pedestrian detection classification model to improve classification performance. In addition, the parameters that affect classification performance of the combined kernel function SVM are analyzed, mainly analyzing the effects of polynomial kernel function parameters, radial basis kernel function parameters, combined kernel function coefficients and penalty factors on classification performance.

**Feature fusion processing.** Features tend to be used independently, and the importance of feature in the overall representation is expressed by a weighted value. Feature fusion is a mutual complement between features. Another local feature is introduced to compensate for the defect of a local feature, so that each local feature can be fused together more effectively, and ultimately achieve the purpose of improving the robustness of image features and getting better accuracy. Therefore, the secondary detection is completed by the fusion feature combined with the improved SVM classification algorithm. The process of feature fusion is shown in Figure 3. Firstly, the HOG algorithm is used to extract features of the pedestrian’s external contour, and then the LBP algorithm is used to extract features of the pedestrian area. Perform preprocessing operations on the obtained initial features, and the weighted features are fused to get the final fusion features. The weighting formula is shown in (10):

\[
\lambda = \alpha . L_1 + \beta . L_2
\]

(10)

In the formula, \(\lambda\) refers to the fusion feature obtained after the weighting operation \(L_1\) and \(L_2\) in the formula respectively refer to the weighting of corresponding features, and the sum of them is 1, \(\alpha\) and \(\beta\) refer to the pedestrian edge features and texture features obtained after pretreatment.

**Dimension reduction of fusion features.** The curse of dimensionality is also a major problem for feature fusion. Feature fusion can showing better performance, but too high dimensionality of fusion features will increase the calculation time, and will also have some influence on the final result of pedestrian detection. PCA is a data dimensionality reduction algorithm, which is a linear feature extraction algorithm based on K-L transformation, and can greatly boost the learning rate. The variance in dimension is calculated as:

\[
\text{var}(X) = \frac{\sum_{i=1}^{n} (X_i - \bar{X})^2}{n - 1}
\]

(11)

The calculation of covariance as follows:

\[
\text{cov}(X, Y) = \frac{\sum_{i=1}^{n} (X_i - \bar{X})(Y_i - \bar{Y})}{n - 1}
\]

(12)

The result of covariance can show the degree of correlation between the two dimensions. If the result is 0, it means that there is no relationship between the two dimensions. Suppose that given \(m\) spatial samples \(x_1, x_2, ..., x_m\) with dimension \(n\) of eigenvectors, the covariance matrix can be calculated as:

\[
S = \frac{1}{m} \sum_{j=1}^{m} (x_j - \bar{x})(x_j - \bar{x})^T, \bar{x} = \frac{1}{m} \sum_{j=1}^{m} x_j
\]

(13)

The covariance matrix \(S\) is decomposed to obtain matrix \(P_k = [p_1, p_2, ..., p_k]\) composed of eigenvectors corresponding to the first \(k\) largest eigenvalues of covariance \(S\), and the final eigenvector can be obtained by dimensionality reduction calculation:

\[
y = P_k^T (x - \bar{x})
\]

(14)

**Improved APSO - SVM model construction**

**Build composite kernel functions**

The kernel functions commonly used in SVM classifier are as follows:

- **Linear function**: \(K(x_i, x_j) = x_i^T x_j\)
- **Polynomial function**: \(K(x_i, x_j) = (\delta x_i^T x_j + r)^d, \delta > 0\)
- **Radial basis function**: \(K(x_i, x_j) = \exp(-\frac{|x_i - x_j|^2}{\sigma^2}), \delta > 0\)
- **Sigmoid function**: \(K(x_i, x_j) = \tanh(\delta x_i^T x_j + r), \delta > 0\)

In the kernel functions, \(\delta\) and \(\delta, \sigma\) are all adjustable parameters, and \(r\) is the punishment function, which is used to measure losses.

The global kernel function has a strong generalization ability, while the local kernel function has a strong learning ability. If these two kinds of kernel functions are combined, their respective advantages can be brought into fully exploited, which makes the combined kernel function have good generalization ability and learning ability, and improves the recognition performance to a certain extent. The combination kernel function can be formed by linear recombination of the aforesaid several kernel functions, and the combination kernel function is used as the kernel function of SVM. The following expression can be used to express the linear combination of kernel functions:

\[
\begin{align*}
G_k = & \sum_{a=1}^{n} d_a K_i, d_a \geq 0 \\
d_1 + d_2 + ... + d_n = 1
\end{align*}
\]

(15)
$G_k$ represents the combinatorial function, $n$ represents the number of kernel functions, and $d_r$ represents the weight value of each kernel function.

In the global kernel function, the polynomial kernel function has the best performance, while in the local kernel function, the radial basis kernel function has the best performance. In this paper, the two functions are combined to construct a combinatorial kernel function. The combined kernel function can be expressed as:

$$G_k = m(\delta x^T x_j + r)^d + (1 - m) \exp\left(-\frac{||x_i - x_j||^2}{2\sigma^2}\right)$$

Equation (16)

$m$ refers to the weight coefficient of the combined kernel function, $m \in (0,1)$. In this paper, the adaptive particle swarm optimization algorithm is used to obtain the optimal parameters.

**APSO algorithm is used to find the optimal parameters**

The particle swarm optimization (PSO) is to simulate the migration and swarming behavior of birds in the foraging process. Its principle is that the particles start from the random solution and iterate to find the optimal solution, that is, to find the global optimal value by following the current optimal value. By using the information sharing of individuals in the group, the whole group realizes the transformation from disorder to order in the process of solving, so as to get the optimal solution. Adaptation introduces mutation operation into the original algorithm, that is to reintialize some variables with a certain probability. The mutation operation makes the particle jump out of the previously found optimal value position, and carries out the search in a larger space, while maintaining the diversity of the population and improving the possibility of searching for a better value. In the process of solving the problem, an optimization function needs to be defined, and the fitness value of each particle is calculated by the optimization function.

Suppose the problem search space is $D$ dimensional, and the total number of particles in the population is $N$, it position of particle is $D_x = (D_1, D_2, \ldots , D_N)$, the velocity vector is $v_x = (v_1, v_2, \ldots , v_N)$. The optimization is carried out in the form of gradual iteration. Equation (15) represents the update of speed and position at the current moment and the previous moment:

$$v_x(t+1) = w v_x(t) + c_1 r_1 (P_x - D_x) + c_2 r_2 (G_x - D_x)$$

$$D_x(t+1) = D_x(t) + v_x(t+1)$$

Equation (17)

Where $P_x$ represents the individual optimal value of the particle, $G_x$ represents the global optimal value, $w$ is the inertial weight, $c_1$ and $c_2$ are learning factors that vary in the range (0,2), $r_1$ and $r_2$ are random numbers that vary in the range (0,1). The formula 17 adds weight on the basis of the original PSO algorithm position update formula, so that the particle swarm can be better judged in the optimization problem.

$$w_d = \left\{ \begin{array}{ll}
(f_{\text{max}} - f_d) \frac{w_{\text{max}, f}}{\sum_{i=1}^{n} f(x_i)} + w_{\text{min}, f}, & f(x_d) \geq \sum_{i=1}^{n} \frac{f(x_i)}{n} \\
\frac{w_{\text{max}, f}}{n}, & f(x_d) < \sum_{i=1}^{n} \frac{f(x_i)}{n}
\end{array} \right. $$

Equation (18)

where $w_d$ is the weight which we want to find, as can be seen from the equation (18), with the increase in number of iterations, the value of weight is constantly decreasing. With the increase of the number of iterations, the particle swarm motion is getting closer to region of the optimal solution. At this time, the proportion of the global search kernel function decreases, while the proportion of the local kernel function increases. By slowing down the search speed, the accuracy is improved, and the speed of convergence to the optimal solution is accelerated. In pedestrian detection, the APSO algorithm is used to obtain the optimal parameters in the combined kernel function, and the pedestrian detection classifier is constructed. The main steps are shown in Figure 4.

**Results**

In the calculation, APSO uses individual extreme values and group extreme values to achieve gradual optimization, and follows the current optimal solution in the search process, with extremely fast convergence speed. The influence of the combined kernel function coefficient $m$ on the classification performance was verified, and the following results were obtained:

By comparing from Figure 5, it can be seen that when $m$ is 0.1, performance of the combination kernel function is the best, and the result of detection and classification is the best.

It can be seen from the parameter setting that the parameters affecting classification performance mainly include the optimal parameter of the combined kernel function and the punishment function. When the punishment factor $r$ takes different values, the recognition rate of different kernel functions changes as shown in the Figure 6:

The data in Figure 6 show that the recognition rate of the combination kernel function is the highest when the punishment factor $r$ is 100. After feature fusion and dimensionality reduction processing, it is jointly trained with the improved SVM classifier. The obtained classifier is connected in series with the classifier in first stage, and the output of the classifier in first stage is used as the input of the classifier in the second stage to realize the head shoulder detection of the combined structure, and the final detection result is obtained. The detection results are shown in Figure 7.
It can be seen that the second-level detection based on the first-level detection will effectively reduce the false detection rate and distinguish the real target. The feasibility of the method used in this paper is also proved through comparative experiments. The secondary detection based on the primary detection can effectively improve the accuracy. Relatively speaking, the method used in this paper will increase the detection time to a certain extent, but the increase in detection time is very small, while the reduction in false detection rate is significant.

The development tool of the experiment is Microsoft Visual Studio 2015. It is carried out under the 64-bit Windows 10 operating system, Intel(R) Core(TM) CPU i5@3.3GHz; the memory is 8 GB.

In traditional methods, some features perform better, such as HOG features and LBP features. Some researchers propose to fuse features for classifier training to obtain better detection results. The method in this paper is compared with them, and the results are presented in Table 2.

Nowadays, more and more researchers use the method of deep learning to complete the detection research, and compare the methods used in this article with the popular SSD and Faster. The experimental results were compared with other commonly used methods on INRIA and Caltech data sets respectively. In order to reflect the fairness of the experiment, different methods will take the same configuration. The comparison result shows in Figures 8 and 9.

Through data analysis, it can be seen that the method used in this paper is obviously superior to the recognition results of traditional methods in detection results. Compared with SSD, YOLO, etc., it also shows certain advantages, especially in the INRIA dataset. Detection background is relatively simple, no excessive occlusion cases, excellent performance. Comprehensive analysis shows that the classification detection method proposed in this paper has the best recognition effect.

**Discussion**

In order to achieve accurate and fast pedestrian detection, this paper proposes a hierarchical detection algorithm. In the first stage of detection, the classifier trained by Haar feature + Adaboost algorithm is used as the detector to extract the ROI. The combination function is used as the kernel function of SVM, and the adaptive particle swarm optimization algorithm is used to obtain the optimal weight of the combination kernel function. The improved SVM is used as the classifier of the second stage detection, and the head shoulder detector is trained to generate the relevant features for the second detection of the image of the “region to be detected.” The fusion features are used to greatly improve detection performance. In addition, the cascading structure is used to effectively refine the detection area and provide faster results. However, in
complex scenes such as occlusion, the performance needs to be improved. Good results can be obtained on simple INRIA data sets, but the detection results on more complex datasets are not very good, and it needs to be improved and optimized. Future work will focus on pedestrian occlusion, and an effective and lightweight feature extraction network without preliminary training is also worth considering.

**Declaration of conflicting interests**

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

**Funding**

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was partially supported by the Characteristic Innovation research project of Teachers in Guangdong Colleges and universities (2020DZXX07) and the Quality Engineering project of Education Department of Guangdong Province: Guangdong Higher Education Document no. 29 [2021] and Youth Scientific Research Project of Guangdong Universities (ky202015, ky202103).

**ORCID iD**

Yin zhang https://orcid.org/0000-0001-7728-9058

**References**

1. Cai Y, He Y, Wang H, et al. Pedestrian detection algorithm in traffic scene based on weakly supervised hierarchical deep model. *Int J Adv Robot Syst* 2016; 14(1): 1729881417692311.
2. Papageorgiou C and Poggio T. A trainable system for object detection. *Int J Comput Vis* 2000; 38(1): 15–33.

3. Viola P and Jones MJ. Robust real-time face detection. *Int J Comput Vis* 2004; 57(2): 137–154.

4. Dalal N and Triggs B. Histograms of oriented gradients for human detection. In: 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR’05), 2005, pp.886–893. New York, NY: IEEE.

5. Ojala T, Pietikäinen M and Harwood D. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognit* 1996; 29(1): 51–59.

6. Wang X, Han TX and Yan S. An HOG-LBP human detector with partial occlusion handling. In: *IEEE international conference on computer vision*, 2009. New York, NY: IEEE.

7. Felzenszwalb PF, Girshick RB, McAllester D, et al. Object detection with discriminatively trained part-based models. *IEEE Trans Pattern Anal Mach Intell* 2010; 32(9): 1627–1645.

8. Felzenszwalb PF, Girshick RB and McAllester D. Cascade object detection with deformable part models. In: 2010 IEEE Computer society conference on computer vision and pattern recognition, 2010, pp.2241–2248. New York, NY: IEEE.

9. Krizhevsky A, Sutskever I and Hinton G. ImageNet classification with deep convolutional neural networks. *Adv Neural Inf Process Syst* 2012; 25(2): 1097–1105.

10. Angelova A, Krizhevsky A, Vanhoucke V, et al. Real-time pedestrian detection with deep network cascades. In: *British machine vision conference*, 2015, pp.32.1–32.12. Swansea: BMVA Press.

11. Ouyang W and Wang X. Joint deep learning for pedestrian detection. In: *IEEE international conference on computer vision*, 2014. New York, NY: IEEE.

12. Guo T, Huynh CP and Sohl M. Domain-adaptive pedestrian detection in thermal images. In: 2019 IEEE international conference on image processing (ICIP). 2019. New York, NY: IEEE.

13. Guan D, Cao Y, Yang J, et al. Fusion of multispectral data through illumination-aware deep neural networks for pedestrian detection. *Inf Fusion* 2019; 50: 148–157.

14. Wang L and Siddique AA. Facial recognition system using LBPH face recognizer for anti-theft and surveillance application based on drone technology. *Meas Control* 2020; 53(7–8): 1070–1077.

15. Zhang X, Gao H, Xue C, et al. Real-time vehicle detection and tracking using improved histogram of gradient features and Kalman filters. *Int J Adv Robot Syst* 2018; 15(1): 1729881417749949.

16. Balli S, Sağbaş EA and Peker M. Human activity recognition from smart watch sensor data using a hybrid of principal component analysis and random forest algorithm. *Meas Control* 2019; 52: 37–45.

17. Yu C, Huiyong L, Fang Z, et al. Adaptive Kalman filtering-based pedestrian navigation algorithm for smartphones. *Int J Adv Robot Syst* 2020; 17(3): 172988142093093.

18. Chen X, Liu L, Deng Y, et al. Vehicle detection based on visual attention mechanism and adaboost cascade classifier in intelligent transportation systems. *Opt Quantum Electron* 2019; 51(8): 263.1–263.18.

19. Li W, Li H, Wu Q, et al. Headnet: an end-to-end adaptive relational network for head detection. *IEEE Trans Circuits Syst Video Technol* 2020; 30(2): 482–494.

20. Eckardt M, González JA and Mateu J. Graphical modeling and partial characteristics for multitype and multivariate-marked spatio-temporal point processes. *Comput Stat Data Anal* 2021; 156: 10713.

21. Wang X, Zhou Z, Li Y, et al. An algorithm for detecting the HOG features of head and shoulder of football players based on SVM classifier. In: 2020 International conference on intelligent transportation, big data & smart city (ICITBS), 2020.

22. Sahoo PK, Kanungo P, Mishra S, et al. Entropy feature and peak-means clustering based slowly moving object detection in head and shoulder video sequences. *J King Saud Univ Comput Inf Sci*. Epub ahead of print 13 January 2021. DOI: 10.1016/j.jsuci.2020.12.019.

23. Kumar A and Srivastava S. Object detection system based on convolution neural networks using single shot multi-box detector[J]. *Procedia Comput Sci* 2020; 171: 2610–2617.

24. Taherkhani A, Cosma G and McGinnity TM. AdaBoost-CNN: an adaptive boosting algorithm for convolutional neural networks to classify multi-class imbalanced datasets using transfer learning. *Neurocomputing* 2020; 404: 351–366.

25. Li P, Wang H, Li Y, et al. Analysis of face detection based on skin color characteristic and AdaBoost algorithm. *J Phys Conf Ser* 2020; 1601(5): 052019.

26. Zhi-Jin LI, Liang Z, Peng WU, et al. Fine-grained vehicle models based on feature fusion convolutional neural network. *Comput Eng Design* 2020; 2020.01.037. 226–230.

27. Li A, Qi J and Lu H. Multi-attention guided feature fusion network for salient object detection. *Neurocomputing* 2020; 416: 426–427.

28. Zhang Y, Dai M and Ju Z. Preliminary discussion regarding SVM kernel function selection in the twofold rock slope prediction model. *J Comput Civ Eng* 2016; 30(3): 04015031.