Received August 7, 2019, accepted August 16, 2019, date of publication August 20, 2019, date of current version September 5, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2936454

A Pedestrian Detection Method Based on Genetic Algorithm for Optimize XGBoost Training Parameters

YU JIANG[1], GUOXIANG TONG[1], HENAN YIN[1], AND NAIXUE XIONG[2,3]

[1]Shanghai Key Laboratory of Modern Optical System, School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China
[2]College of Intelligence and Computing, Tianjin University, Tianjin 300350, China
[3]Department of Mathematics and Computer Science, Northeastern State University, Tahlequah, OK 74464, USA

Corresponding author: Guoxiang Tong (tonggx@usst.edu.cn)

This work was supported by the National Key Research and Development Program of China under Grant 2018YFB1700902.

ABSTRACT In this paper, we present a machine learning classifier which is used for pedestrian detection based on XGBoost. Our approach, the Genetic Algorithm is introduced to optimize the parameter tuning process during training an XGBoost model. In order to improve the classification accuracy, HOG and LBP features are used to describe pedestrians in a way of tandem fusion, then input into GA-XGBoost classifier proposed in this paper to form a new static image pedestrian detection algorithm. The pedestrian feature extraction and machine learning are decoupled by storing the extracted pedestrian feature as feature files in the experiment, so that training can be exacuted many times and algorithms can be camparied conveniently. Experimental show that our pedestrian detection algorithm has improved the accuracy of pedestrian detection in the static image. The Area Under the ROC Curve (AUC) value reaches 0.9913.

INDEX TERMS Pedestrain detection, histogram of oriented gradient features (HOG), local binary patterns (LBP) XGBoost classifier, genetic algorithm.

I. INTRODUCTION

Pedestrian detection is widely used in intelligent monitoring, automotive assisted driving, security and intelligent transportation areas. It has been concerned for several years in many computer vision applications. Specially machine learning become more and more popular for pedestrian detection. In general, it is realized by building pedestrian feature extraction model and building feature classifier.

HOG features [1] are the most commonly used for feature extraction that depicts the local gradient amplitude and direction feature of the image, which was evolved from SIFT [2] and proposed by Dalal et al in 2005.However, due to the large feature dimension extracted by HOG algorithm, it takes more time for training and detection through classifier. Dollar et al. proposed the integral channel feature (ACF) [3], HOG and Haar and other features were combined, and the computing speed was accelerated by means of integral graph vector diagram. In this paper, the HOG and LBP proposed in literature [4] are adopted as feature sets to deal with the pedestrian detection method with partial occlusion.

Dai *et al.* [5] proposes a part-based detection model. This module was composed of several subdetectors to constitute a complete classifier. Each subdetector detects different parts of the target. The disadvantage of this module was that it needs to manually mark different parts of the target, which is tedious. With the continuous development of deep learning that has been used for pedestrian detection [6], [7]. Ouyang and Wang [8] discover the imperfection of the part model, proposed a probabilistic pedestrian detection framework based on deep learning.

Currently, pedestrian detection generally adopts machine learning algorithm for target classification. The common machine learning algorithms are: SVM (support vector machine), neural network, random forest, Adaboost, etc. SVM [9] is a classical classfication algorithm that was first introduced by Vapnik, and regared as the current best algorithm by many scholars. Its principle is linear classifier based on maximum interval, while it does not work well in solving nonlinear problems. Neural network deep learning is also excellent in various fields. The Go ai AlphaGo from

The associate editor coordinating the review of this article and approving it for publication was Chun-Wei Tsai.

Deep Mind [10] is based on a variety of deep neural network technology. Convolution Neural Network (CNN) is used most in the field of image recognition model, a Neural Network by using local receptive field and hier archical characteristics of the depth of the incremental processing method, can automatically extract the image feature which had solved the time-consuming problem in traditional computer visio of artificial extracting features, but need high performance computing equipment to reach the demand of the real scene. LiH et al. proposed a target tracking method based on CNN [11], in order to use the CNN into the field of image tracking, and improve the tracking precision and speed, they put forward a new structure of CNN. After comparing with other tracking algorithm, this algorithm achieved high accuracy after comparing with other tracking algorithm, but depends on high performance computing devices. Boosting is a machine learning ensemble meta-algorithm for primarily reducing bias, and also variance [12] in supervised learning, and a family of machine learning algorithms that convert weak learners to strong ones. While boosting is not algorithmically constrained, most boosting algorithms consist of iteratively learning weak classifiers with respect to a distribution and adding them to a final strong classifier. Among many boosting algorithms, the Gradient Boosting [13] exhibited excellent and practical performance, which is a machine learning technique for regression and classification problems to produce a prediction model in the form of an ensemble of weak prediction models. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function. This functional gradient view of boosting has led to the development of boosting algorithms in many areas of machine learning and statistics beyond regression and classification. The XGBoost mentioned in [14] is development from a GBDT(Gradient Boosting Decision Tree), and was proved to have very good convergence speed and generalization ability. As XGBoost has come to be known, some scholars have proposed scenario-based optimization algorithms. Optimized XGBoost-LMT [15] is used to improve the accuracy and stability of credit evaluation model. Artificial Chemical Reaction Optimization Algorithms (ACROA) is used to solve the problem of slow parameter search in XGBoost. At the same time, a Bayesian optimization algorithm based on Gauss method is proposed to optimize the parameters of XGBoost [16], and AUC-H measurement and Brier score are used.

Because the basic structure of XGBoost algorithm is decision tree, which can correlate the original independent features, this paper combines HOG and LBP features in series as the feature input of XGBoost to find the association between HOG features and LBP features to improve the classification accuracy in the way of using this ‘‘black box’’ learning. Moreover, for dealing with the numerous parameters of XGBoost, the cross-validation is selected to search the parameters that make the model the best. To deal with the complex global searching, we introduce the Genetic algorithm (GA) to improve its parameters search process and the GA-XGBoost algorithm formed, which is used as a classifier for pedestrian detection. In this paper, based on INRIA data set, under the condition of using HOG and HOG-LBP as pedestrian feature input respectively, we compare the performance of GA-XGBoost with that of SVM classifiers which are widely used in pedestrian detection. The experimental results show that the proposed algorithm has higher classification accuracy.

TABLE 1. Advantages and disadvantages of various feature algorithm.

Feature Algorithm	Advantage	Disadvantage
Color feature	Good stability to various transformations of the target	The feature description is not detailed enough, and the two targets with similar color distribution are not easy to distinguish
Shape feature	It is easy to distinguish objects with distinct shapes and has fast calculation speed.	Sensitivity to noise and deformation
Haar	Good description of light and shade changes	Weak description of object side
Hog	Describe shape contours, etc.	Poor description of the texture
Lbp	Describe the texture characteristics of objects	Unable to describe the contour change of the object

II. KEY TECHNOLOGIES OF PEDESTRIAN DETECTION

When matching recognition or classifier recognition is performed in pattern recognition, the basis for judgment is image features, so feature extraction plays an important role in image matching and classification. The common feature extraction algorithms are mainly divided into the following three categories: 1. Based on color features: such as color histogram, color set, color moment, color aggregation vector, etc; 2. Based on texture features: such as Tamura texture features, autoregressive texture models, Gabor transforms, wavelet transforms, MPEG7 edge histograms, etc; 3. Based on shape features: such as Fourier shape descriptors, invariant moments, wavelet contour descriptors, etc;For the feature extraction algorithm of pedestrians in the image, several classical feature extraction algorithm are chosen to compare as shown in Table 1:

Comprehensive analysis, this paper selects HOG+LBP [17] as the feature fusion algorithm to extract features for pedestrians.

A. HOG FEATURE

HOG (Histograms of Gradient), which is used to describe the direction change and shape contour of a local region by calculating the direction density distribution of the gradient or the edge, and forms the gradient histogram of each region by sliding window operation. The processing flow is shown in Fig. 1.

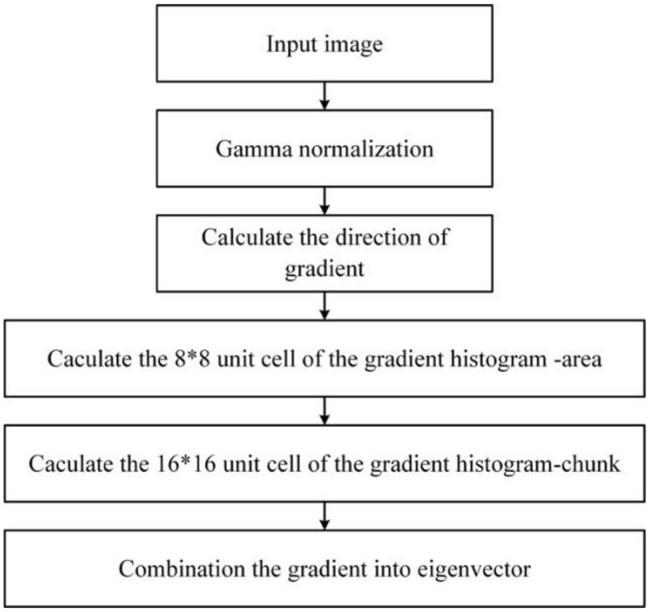


FIGURE 1. HOG feature extraction process.

The steps are as follows:

Step1: In order to reduce the influence light changes, we are using (1) Gamma correction method to normalization color space of the input image. Since the color information is not importance in this paper, so convert a RGB image to grayscale image reducing the memory in calculating.

$$I(x, y) = I(x, y)^{gamma} \tag{1}$$

Step2: Calculating the gradient of the horizontal and vertical directions of the image, and use convolution kernels $[-1, 0, 1]$ and $[-1, 0, 1]^T$ to calculate the gradient component of each pixel in the horizontal and vertical directions. The size and direction of the pixel (x,y) are computed from (2) and (3), in which $G_x(x, y)$ and $G_y(x, y)$ are respectively represent the horizontal gradient and the vertical gradient at the pixel point (x, y) in the input image.

$$G(x, y) = \sqrt{G_x(x, y)^2 + G_y(x, y)^2} \tag{2}$$

$$\mathrm{Obj}^{(t)} = \sum_{i=1}^{n} \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) \tag{3}$$

Step3: As shown in Fig. 2, the image is divided into several small regions, which can generally be set to 9, and the gradient direction in the small region is quantized, so that each region finally obtains a 9-dimensional feature vector. Then, place a block in multiple region, generally each region having 8×8 pixels, and each block having16×16 pixels, so we can get 36-dimensional features in each block.

Setp4: Using this block as a sliding window to scan the whole image. For 128×64 resolution images, it needs to slide 7 blocks in the horizontal direction, 15 blocks in the vertical direction, and finally combine the feature vectors in all blocks. Then we get 3780 dimensions Of Histogram of Oriented Gradients, and combined into final feature vectors for classification.

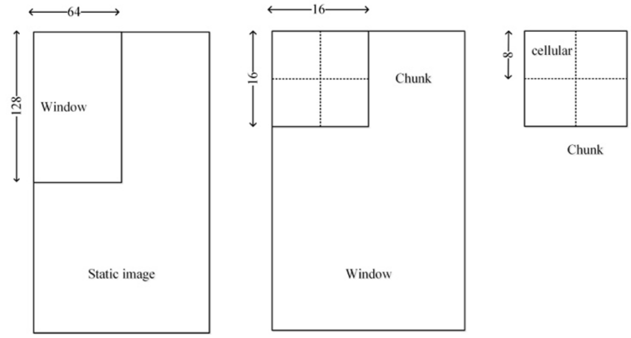


FIGURE 2. Cell and block in HOG.

B. LBP FEATURES

LBP (Local Binary Patterns), which is an algorithm for extracting local image texture features with rotation invariance and gray invariance, and widely used for extracting image texture features. LBP processing steps are as follows.

Step1: The first step is dividing the sliding window into 16×16 blocks;

Step2: Comparing all edge points with their central points in each 9-pixel region. If the surrounding gray value is greater than the central gray value, then the pixel point is assigned a value of 1; otherwise, it is assigned a value of 0. Finally, the edge point in the 9-pixel region will generate a binary string, which is the LBP feature value.

Step3: The LBP feature histogram in each block is computed and normalized to ensure that it will not affect the overall situation;

Step4: The histogram in each block is connected in series as the histogram of the whole image, which is the LBP feature vector of the whole picture.

The LBP features extracted by the above steps correspond to each pixel. After extracting the features of the whole image, the LBP features can still form an image

C. XGBoost

XGBoost [14] (Extreme Gradient Boosting) algorithm is a boosting algorithm for classifying regression tree models [18] which is coming from the gradient lifting decision tree. XGBoost is used for pedestrian detection classifiers in this paper. a general flow of the Boosting algorithm based on the classification regression tree is shown in Fig. 3. First, we need to learn a tree from the sample to obtain the first estimation result Y1, and the second tree is learned with Y based on the difference between the real label and the predictive label in the previous step. By analogy, the algorithm error can be effectively reduced.

Formulas (4) to (8) give the calculation flow of gradient boosting training. The following formulas is to calculate the target of the n-th tree model. The first behavior is the model defines a regularization term, which can reduce overfitting to

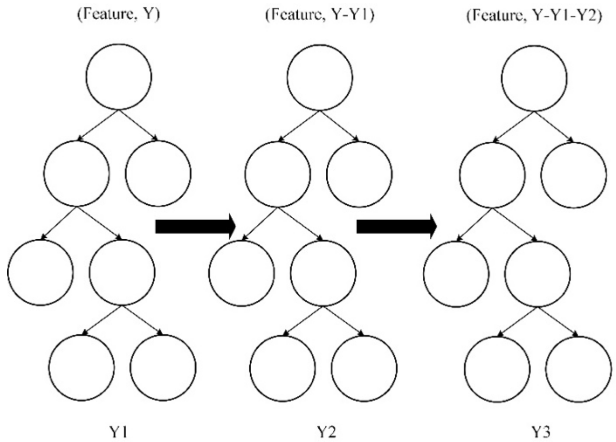


FIGURE 3. Boosting algorithm.

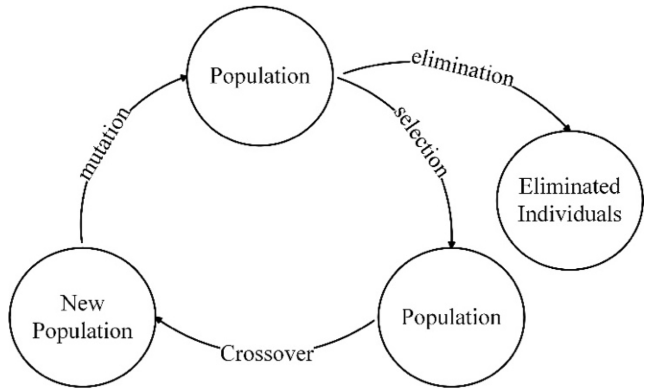


FIGURE 4. The process of biological evolution.

improve the generalization ability of the model. The second behavior is the first three terms from Taylor's formula, which contains constant terms, first and second derivatives. And the first three terms represent the original very well and not complexity. From the formulates we can see that one of the advantages of XGBoost is that it's accurate to the second derivative.

Among them, the objective function of each round is calculated by formula (4), and an f_t is chosen to minimize our objective function, that is, the error between the predicted result and the actual result is reduced after adding f_t . Where l is the error function and Ω is a regularization term, the error function tries to fit the training data as much as possible, and the regularization term encourages a simpler model. When the model is simple, the randomness of the results of the finite data fitting is relatively small, which is not easy to overfitting, which makes the prediction of the final model more stable.

$$\mathrm{Obj}^{(t)} = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)\right) + \Omega(f_t) + constant \quad (4)$$

When the error function l is not a square error, the first three terms of Taylor expansion are used to approximate the original objective function as the formula (5).

$$\mathrm{Obj}^{(t)} = \sum_{i=1}^{n} \left[l\left(y_i, \hat{y}_i^{(t-1)}\right) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) + constant \quad (5)$$

where g_i is the first derivative of the error function and h_i is the second derivative of the error function.

$$g_i = \partial_{\hat{y}_i^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)}) \quad (6)$$

$$h_i = \partial^2_{\hat{y}_i^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)}) \quad (7)$$

Then, by removing the constant term, namely, the difference between the real value and the predicted value of the previous round. The objective function only depends on the first and second derivatives of the error function of each data point.

$$\mathrm{Obj}^{(t)} = \sum_{i=1}^{n} \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) \quad (8)$$

Based on the realization of XGBoost, the algorithm first ranks the eigenvalues, because the tree model needs to determine the best segmentation points, and then stores them in a number of blocks. This structure is reused in later iterations, which has greatly reduced the computational complexity. In addition, the information gain of each feature needs to be calculated in the process of node splitting, so the calculation of information gain can be parallelized by using this data structure.

D. GENETIC ALGORITHM

GA (Genetic Algorithm) is a random adaptive global search algorithm. As shown in Fig. 4, it includes the process of group selection, crossover, mutation and elimination. In the GA, the main body running through the whole algorithm is "chromosome", which is equivalent to the individuals in the population, such a sample or a combination of values, in which each coding unit is called "gene".

The GA distinguishes the merits of each chromosome or sample by comparing the fitness function values. and the larger the value, the better the algorithm. The fitness value of each chromosome is determined by an evaluation function. There are three kinds of operators in general Genetic Algorithm: selection operator, crossover operator and mutation operator. The selection operator is to select the chromosomes in the population according to some logic or rules to get a population that can be crossed. The crossover operator is used to determine the gene rules of two crossing chromosomes, and then the operator determines the gene sequence of the offspring chromosomes after crossing between two chromosomes. The mutation operator is to make some chromosome gene change directly under certain probability and pass it directly into the next generation population. This method can generate new individuals and make available for global optimization of algorithm. The process of the Genetic Algorithm is shown in Fig. 5. Firstly, the population is initialized.

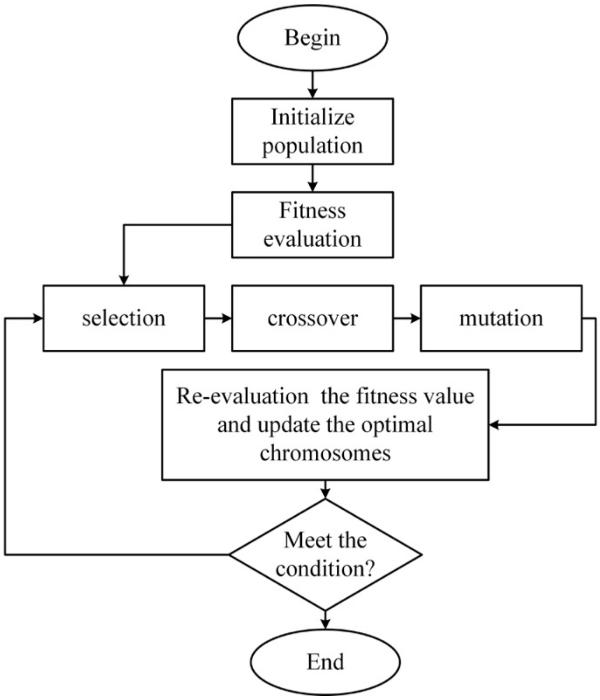


FIGURE 5. Genetic Algorithm flow.

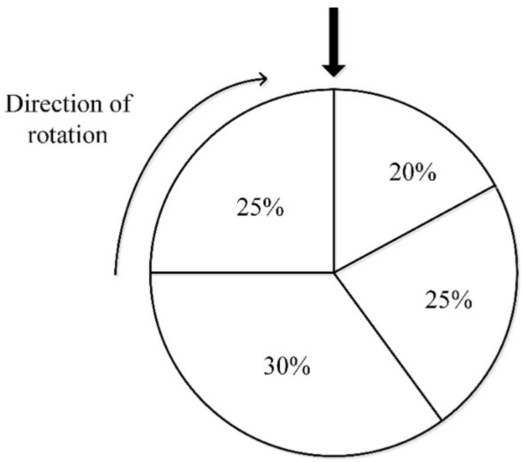


FIGURE 6. Probability-based selection operator.

In this stage, chromosomes need to be encoded, and the population is initialized with random numbers. Secondly, the individuals are calculated and selected according to the evaluation of fitness value. Then the individuals with the highest fitness can be retained to keep the population from degenerating.

In the selection phase, the probability-based operators are shown in Fig. 6. Firstly, we need to calculate the fitness value of each individual, and divide the probability according to the fitness value. Secondly, select the crossable chromosome randomly. Due to the randomness, the individuals with poor fitness are also retained in the selection process, which provided the possibility for the population evolution

In the crossover phase, a random threshold needs to be set to determine whether each individual should be crossover,

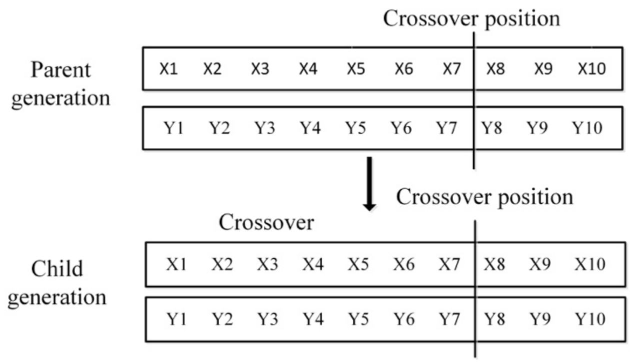


FIGURE 7. Crossover operator.

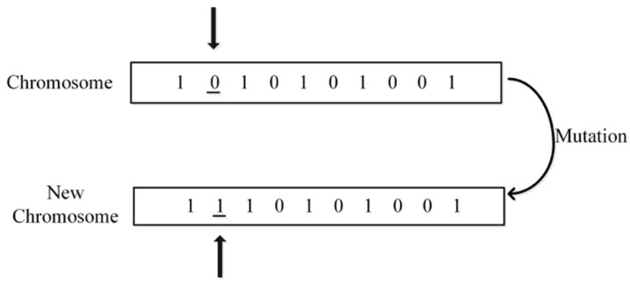


FIGURE 8. Mutation operator.

which determines the number of iterations of the algorithm. The wider the threshold is, the greater the chance of crossover and the fewer iterations. For the individuals that can be crossed, the cross-genes are determined according to the constraints of the problem and random numbers, as shown in Fig. 7. the general case is to exchange the two selected genes.

According to the probability the transformed gene position is selected by the mutation operator, and then changes the gene as shown in Fig. 8. Its probability determines the degree of dispersion of the population

Finally, the termination condition of iteration is used to determine whether genetic evolution is stopped or not

III. IMPROVED XGBoost ALGORITHM BASED ON GA

Although XGBoost has excellent results in all aspects, there are still some problems, one of which is that it has many parameters, and different combinations of parameters get different evaluation scores. If global search is used, it is relatively cumbersome. For genetic algorithm has the unique advantage of solving the optimal solution, and has good results in the field of algorithm parameter search. In view of the problems existing in XGBoost, The Genetic Algorithm proposed is used to optimize the parameter search process of XGBoost in this paper. Booster, General and Task which are the three main parameter sets for XGBoost, the main parts of each parameter sets are listed in Table 2. In most of case, the Booster is used to define the details of the boosting tree, which the definition of each tree can be precise;

TABLE 2. List of XGBoost parameters.

Typel	Paremeter	Default	Explain
Booster	eta	0.3	Shrinking the weights on each step
	Min_child_weight	1	Defines the minimum sum of weights
	max_depth	6	control over-fitting
	gamma	0	Specifies the minimum loss reduction required to make a split
	max_delta_step	0	Help in logistic regression
	subsample	1	Control the sample's proportion
	colsample_bytree	1	column's fraction of randomly samples
	colsample_bylevel	1	the subsample ratio of columns for each split, in each level.
	lambda	1	L2 regularization term on weights
	alpha	1	L1 regularization term on weights
	scale_pos_weight	1	Helps in faster convergence
General	booster	gbtree	Select the model for each iteration
	silent	0	Output message switch
	nthread	max	Parallel processing and input the system core number
Task	objective	reg:linear	Minimizing the loss function
	eval_metric	according to objective	validation data
	seed	0	Random seed

The General is used to control the whole algorithm, such as what boosting model to use, and the number of parallel and so on; The Task is aimed at the objective function and the evaluation indicators which are varies depending on the selected objective function.

We don't need to adjust all of the parameters in Table 2. In this paper, we select three common parameters to be tuning, they are eta, Min_child_weight and max_depth.

1. eta: The shrinkage parameter, which is used to update the leaf node weight, then multiply this factor to avoid excessive step size. The larger the parameter value, the more likely it will not converge. Setting the learning rate eta to a smaller level that can make the learning later more careful.

2. min_child_weight: Minimum sum of instance weight (hessian) needed in a child. For the 0-1 classification when the positive and negative samples are unbalanced, let hessian be around 0.01 and min_child_weight be 1 means that at least 100 samples need to be included in the leaf node. This parameter greatly affects the result, controlling the minimum value of the sum of the second order in the leaf node. The smaller the value of the parameter, the easier it is to overfitting.

3. max_depth: The maximum depth of each tree, the deeper the tree height, the easier it is to overfit.

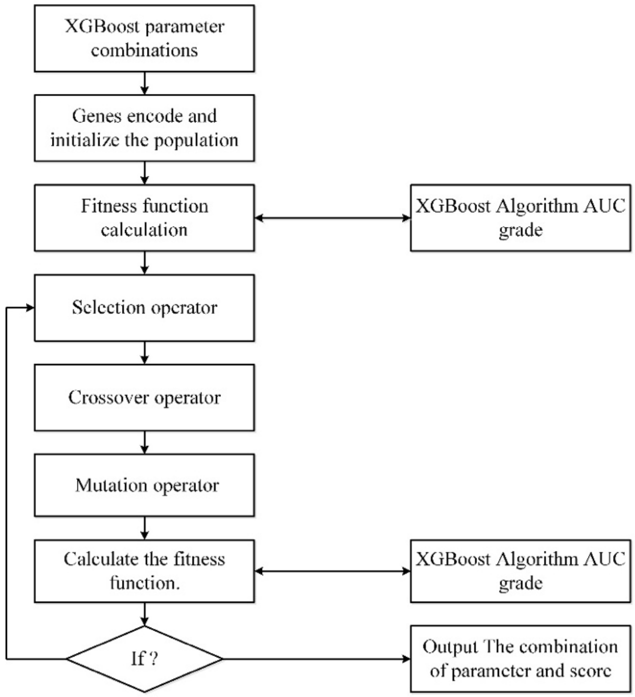


FIGURE 9. GA-XGBoost algorithm flow chart.

Because the eta makes the model more robust by shrinking the weights on each step and improve the robustness of the model, and the mid_child_weight defines the minimum sum of weights of all observations required in a child, which is used to avoid overfitting. Meanwhile, the max depth is used to control over-fitting as higher depth will allow model to learn relations very specific to a particular sample. while these three parameters are selected to be optimized by the Genetic Algorithm, several other important parameters are set as boosting model trees. In the other hand, the logistic regression is selected to be objective function, which is suitable for pedestrian classification processing in the improved GA-XGBoost algorithm is shown in Fig. 9

The basic idea of the improvement is, the parameters which are often modified in XGBoost are regarded as the individual population of Genetic Algorithm, and the XGBoost final training score is used as the fitness function to search out the optimal parameter by selecting, crossing and mutating processes. The steps are as follows of selection, crossover and mutation, so as to improve the performance of the classifier. The steps are as follows:

Step1: Generating a set of default parameters according to the XGBoost algorithm. Then initialize the individual of population.

Step2: The fitness function is got from the output score of the XGBoost algorithm. the evaluation scores can be obtained in this step, such as precision, recall or AUC score.

Step3: Population evolution is performed according to selection operator, crossover operator and mutation operator. the next generation population can be obtained.

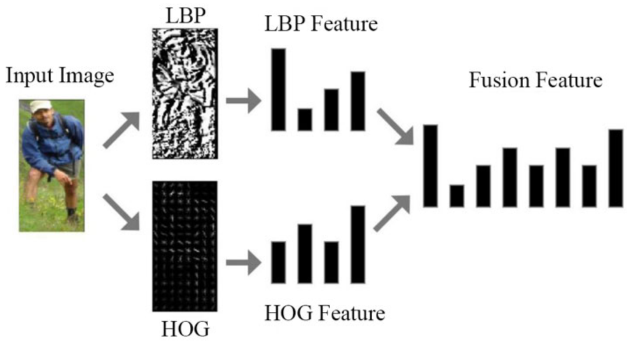


FIGURE 10. HOG-LBP feature fusion.

Step4: The fitness function is recalculated to determine whether it meets the cyclic termination condition. Because it is difficult to set the upper bound of fitness function, the iteration times are used to terminate the algorithm cycle.

IV. PEDESTRIAN DETECTION BASED ON FUSION FEATURE AND GA-XGBoost

A. HOG-LBP FEATURE SERIES FUSION

In this paper, HOG and LBP features are used to fuse in series. As shown in Fig. 10, HOG features are good at describing gradients, while LBP features are good at describing texture changes. The classification accuracy can be improved by inputting complementary features into the classifier.

In machine learning stage, feature sets are formed by feature extraction and fusion of HOG and LBP, and used as the input of XGBoost algorithm. During processing, a small number of samples are selected to optimize the parameters through Genetic Algorithm iteration. Finally, using this optimal parameter combination, the whole training set features are input into GA-XGBoost algorithm to obtain pedestrian detection classifier.

B. SLIDING WINDOW DETECTION WITH NMS

In a static image, pedestrians may be distributed in various areas and need to be located after classification. In this paper, pedestrian candidate regions are extracted from the image by using sliding window location method. Firstly, the original image is transformed into multi-scale images. Then, the sliding window is recognized by GA-XGBoost pedestrian classifier, and the coordinates and probability of the window are recorded. The recorded window is mapped to the original image with the same probability. As shown in Fig. 11, at this time we get many windows near the pre-determined pedestrian. In this paper, non-maximum suppression method is used to eliminate redundancy.

Firstly, according to the coordinates extracted by the sliding window classifier and the corresponding probability, the window with the largest probability value is put into the queue, and the remaining windows are compared with the current maximum probability value window. If the overlap area of the current maximum probability box is larger than the threshold, the window is removed and iterated until the end of the iteration.

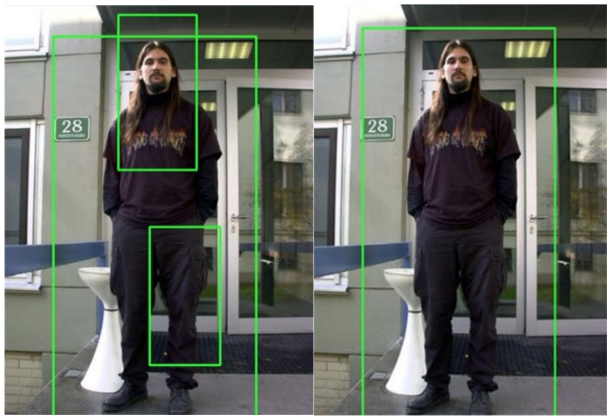


FIGURE 11. NMS window fusion.

TABLE 3. Feature file formats.

Category identifier	Character 1: Value	Character 2 : Value	Character 3 : Value	Character 4: Value
0	1: 0.132	2: 0.398	3: 0.653	4: 0.148
1	1: 0.154	2: 0.242	3: 0.191	4: 0.231
……	……	……	……	……

V. EXPERIMENTAL RESULT AND ANALYSIS

A. DATASETS AND FEATURE FILES

In this paper, INRIA pedestrian dataset is used to extract pedestrians from each image according to the boundary of positive samples, and the negative samples are randomly extracted according to similar proportions. After extraction, the original positive samples 2416 are obtained. And then expanding the sample size to 4832 by mirror. In order to ensure the sample equilibrium, the negative sample is randomly selected, so the total sample size is about 10000. Because different algorithms need to be trained and compared in machine learning stage, features are stored as LIBSVM format feature files, and uses feature files to decouple feature extraction stage from machine learning stage, As shown in Table 3, each row of data is a sample, each column is a feature except the first column, before the colon is the feature mark, after the colon is the feature value, and the first column is the sample label, i.e. the positive sample or the negative sample.

B. EXPERIMENTAL RESULTS AND ANALYSIS

In order to validate the pedestrian detection algorithm proposed in this paper, for the same set of test data sets, SVM classifier with outstanding classification effect is selected to combine with HOG and HOG-BLP feature extraction algorithm respectively, and GA-XGBoost classifier with HOG and HOG-BLP feature extraction algorithm respectively. The experimental results are shown in Tables 4 to 7.

TABLE 4. Results for training set based on HOG+SVM.

HOG+SVM				
Label	Precision	Recall	F1 Scores	Sample size
Negative	0.95	0.93	0.94	241
Positive	0.94	0.95	0.94	250
Average	0.94	0.94	0.94	491

TABLE 5. Results for training set based on HOG-LBP+SVM.

HOG&LBP+SVM				
Label	Precision	Recall	F1 Scores	Sample size
Negative	0.95	0.93	0.94	241
Positive	0.94	0.95	0.94	250
Average	0.94	0.94	0.94	491

TABLE 6. Results for testing set based on HOG+SVM.

HOG+SVM				
Label	Precision	Recall	F1 Scores	Sample size
Negative	0.94	0.95	0.94	906
Positive	0.96	0.95	0.96	1178
Average	0.95	0.95	0.95	2084

TABLE 7. Results for testing set based on HOG-LBP+SVM.

HOG&LBP+SVM				
Label	Precision	Recall	F1 Scores	Sample size
Negative	0.96	0.95	0.95	906
Positive	0.96	0.97	0.96	1178
Average	0.96	0.96	0.96	2084

Keeping the datasets and features unchanged, the GA-XGBoost algorithm proposed in this paper is used as a classifier for training and testing. Firstly, the parameters of XGBoost algorithm are optimized by GA algorithm, and then combined with HOG and HOG-BLP features algorithms respectively for training and testing.

In the individual initialization stage, the Genetic Algorithm is randomly assigned, and the combination of default parameters of XGBoost is taken as an individual in the initialization population. by Genetic Algorithm iteration, an optimal set of parameters (0.376, 5, 41) is obtained. The process parameters in the parameter optimization stage are shown in Table 8.

The results of machine learning training and test using the fusion features of HOG and HOG-LBP as input are shown in Tables 9 to 12. The results show that under the same GA-XGBoost classifier, the effect of HOG-LBP fusion features is still better than that of HOG single feature.

Fig. 12–Fig. 14 shows the comparison from the precision, recall, and F1 scores of several algorithm combinations on the training set. Fig. 16–Fig. 18 shows the comparison from the same characteristics of several algorithm combinations on the test set. The results show that the HOG-LBP features

TABLE 8. Optimization results of GA-XGBoost algorithm parameter.

Iterations	Optimal parameters of every step	AUC Scores
1	0.200，5，20	0.9784
2	0.238，6，25	0.9796
3	0.332，5，33	0.9742
4	0.270，3，29	0.9815
5	0.351，5，39	0.9865
6	0.356，4，37	0.9902
7	0.343，2，45	0.9837
8	0.295，3，40	0.9877
9	0.376，5，41	0.9913
10	0.270，3，29	0.9815
11	0.355，5，36	0.9812
12	0.313，4，35	0.9826
13	0.238，6，25	0.9796
14	0.374，5，39	0.9901
15	0.344，5，41	0.9911
……	……	……
Optimum	0.376，5，41	0.9913

TABLE 9. Results for training set based on HOG+GA-XGBoost.

HOG+GA-XGBoost				
Label	Precison	Recall	F1 Scores	Sample size
Negative	0.95	0.94	0.95	241
Positive	0.94	0.95	0.95	250
Average	0.95	0.95	0.95	491

TABLE 10. Results for training set based on HOG-LBP+ GA-XGBoost.

HOG-LBP+GA-XGBoost				
Label	Precision	Recall	F1 Scores	Sample size
Negative	0.98	0.96	0.97	291
Positive	0.96	0.98	0.97	250
Average	0.97	0.97	0.97	491

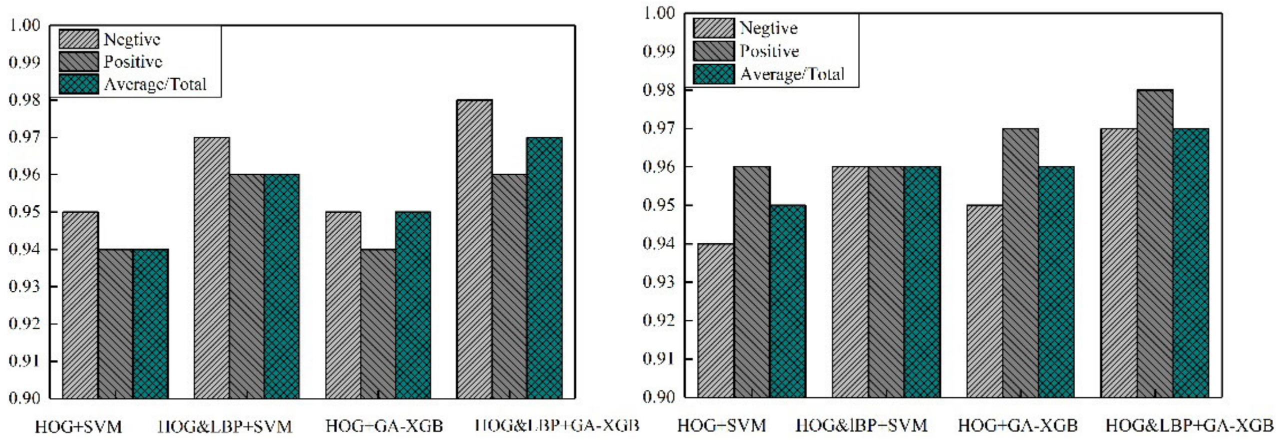


FIGURE 12. Performance comparion from precision between defferen algorithm on the training set(left) and test set(right).

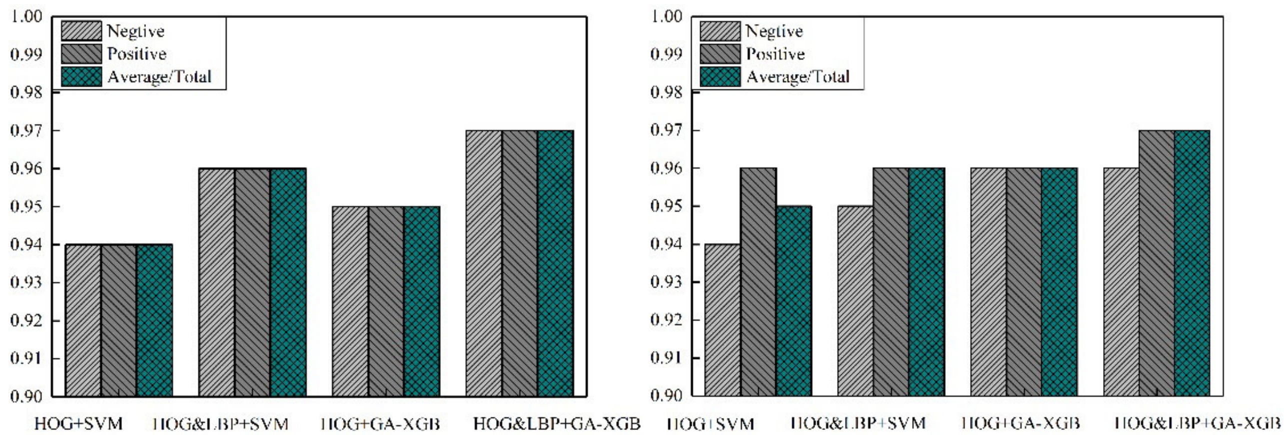


FIGURE 13. Performance comparion from recall between defferen algorithm on the training set(left) and test set(right).

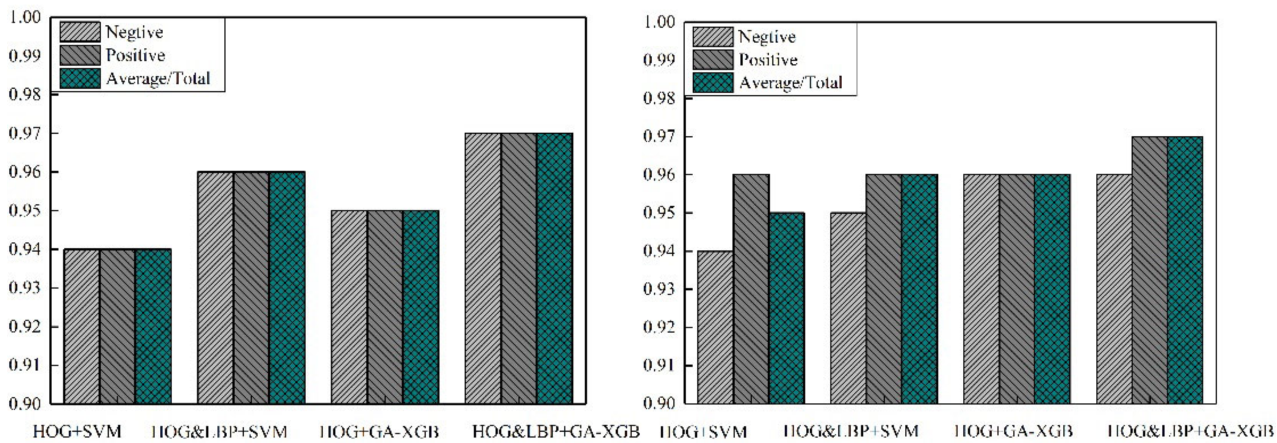


FIGURE 14. Performance comparion from F1 scores between defferen algorithm on the training set(left) and test set(right).

can improve the score of the classifier on various indicators. On the training set and test set, the average accuracy, recall rate and F1 score of the positive and negative samples are 0.97 for the proposed algorithm, respectively.

In order to verify the generalization ability of the GA-XGBoost algorithm proposed in this paper, the MOT_2DMOT2015 data set was added for testing. We random select 338 samples on this data set. The experimental

TABLE 11. Results for test set based on HOG+ GA-XGBoost.

HOG+GA-XGBoost				
Label	Precision	Recall	F1 Scores	Sample size/500
Negative	0.95	0.96	0.96	906
Positive	0.97	0.96	0.96	1178
Average	0.96	0.96	0.96	2084

TABLE 12. Results for test set based on HOG-LBP+ GA-XGBoost.

HOG-LBP+GA-XGBoost				
Label	Precision	Recall	F1 Scores	Sample size
Negative	0.97	0.96	0.96	906
Positive	0.98	0.97	0.97	1178
Average	0.97	0.97	0.97	2084

TABLE 13. Results for test set based on HOG-LBP+ GA-XGBoost.

HOG-LBP+GA-XGBoost				
Label	Precision	Recall	F1 Scores	Sample size
Micro Average	0.97	0.97	0.97	338

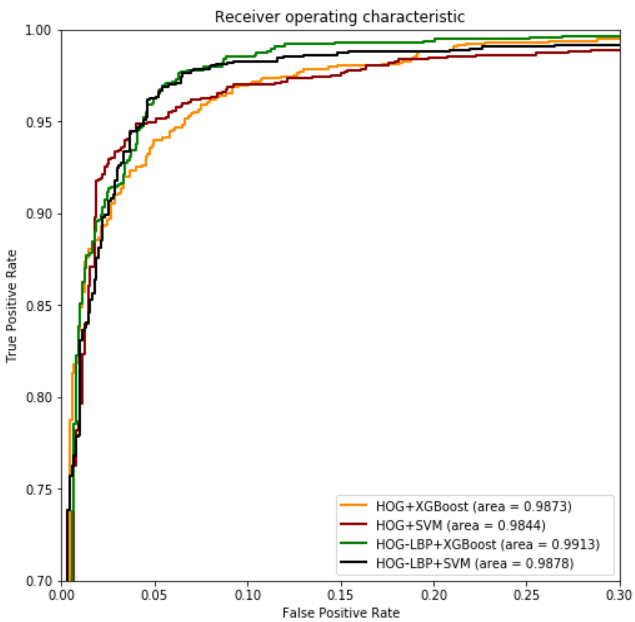


FIGURE 15. Comparison of pedestrian classification algorith.

results are shown in Table 13. The testdate accuracy is 0.97 and the testdate reall is 0.99. It indicates that the proposed algorithm on this paper still has strong generalization ability in various data sets.

TABLE 14. Comparison of AUC scores of other classifiers.

Classifier	HOG features	HOG-LBP features
SVM	AUC=0.9844	AUC=0.9878
GA-XGBoost	AUC=0.9873	AUC=0.9913

In order to verify the influence of sample distribution on the performance of the algorithm, the area under the ROC curve is used to measure the performance of the algorithm. Fig. 15 shows the HOG feature and the HOG-LBP fusion feature corresponding to the ROC curve when SVM and GA-XGBoost are used as classifiers respectively. Table 14 shows the results of ACU score comparison. From the ROC curve, the green curve corresponds to the pedestrian detection algorithm proposed in this paper, and can enclose other curves, which shows that the algorithm performs better. Its AUC score can reach 0.9913, which is obviously higher than other algorithms. It has good stability and generalization ability.

VI. CONLUSION AND FUTURE WORKS

In order to improve the accuracy of pedestrian detection algorithm, Genetic Algorithm is used to improve the parameter optimization process of XGBoost algorithm to forms the called GA-XGBoost as pedestrian detection classifier. Meanwhile, the HOG-LBP fusion features as pedestrian features are used to input into GA-XGBoost algorithm for training and testing pedestrian classifier. Experimental results show that this method can improve the performance of the algorithm. Especially for ACU values. However, in real scenes, it is still a challenge to accurately detect and track every pedestrian on the road because of the complexity and variety of pedestrians on the road. Therefore, the next research includes the following parts:

(1) Although HOG and LBP feature fusion methods are used to complement each other by their respective advantages, sliding window is used to extract features to slow down the processing speed in this paper. Therefore, further research is needed to improve the speed of feature extraction without losing accuracy, such as using ensemble learning method.

(2) Find new methods to improve the score of classifier and reduce the rate of missed detection.

(3) Although the XGBoost parameter search method based on Genetic Algorithm proposed in this paper effectively simplifies the parameter search process. It is easy to fall into the local minimum in the Genetic Algorithm. The next work will focus on this problem.

(4) For mobile vehicles, the camera picture is dynamic in the form of video, so a mature pedestrian detection system should not only include pedestrian detection, but also further track the pedestrians after detection. The Template Matching algorithm [19], when each frame comes, looks for the most similar to this target in the whole image, relying on the parameter method as the evaluation criterion parameter. The main process of the algorithm is to slide the frame in the input source image to find the similarity between each position

and the template image, meanwhile the result is saved in the result matrix. And the brightness of each point of the matrix represents the degree of matching with the template. Then the maximum value in the matrix R can be located by the function minMaxLoc (the minimum value can be also determined by this function). Moreover, this algorithm has low quality requirements for infrared images, so it's suitable for target tracking under conditions of low signal-to-noise and complex background.

For infrared video [19], [20] images have the characteristics of high resolution and concealment, and have more accurate recognition camouflage ability than conventional visible light, which can not only effectively reduce noise, etc., especially in target tracking under complex environmental.

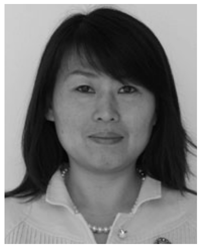
So infrared and visible light video fusion methods can used to enhance the detection effect that make the results of pedestrian movements at night more reliable and effective, and the information of pedestrians and their environment is clearer and more intuitive. Firstly, the infrared video is filtered by the spatio-temporal filtering technique and the target thermal imaging feature. Then, according to the brightness information in the infrared video, the regional seed growth algorithm is used to segment the moving target, and then the target is filtered according to the shape and color information of the region. finally, Finally, the infrared and visible light video are combined to enhance the detection results. so that the information provided in the video is more abundant.

REFERENCES

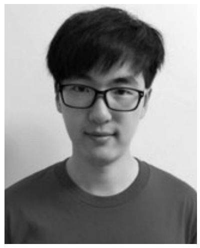
[1] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, San Diego, CA, USA, Jun. 2005, pp. 886–893.
[2] J. Zhao, H. Liu, Y. Feng, S. Yuan, and W. Cai, "BE-SIFT: A more brief and efficient SIFT image matching algorithm for computer vision," in *Proc. IEEE Int. Conf. Comput. Inf. Technol.; Ubiquitous Comput. Commun.; Dependable, Autonomic Secure Comput.; Pervasive Intell. Comput.*, Liverpool, U.K., Oct. 2015, pp. 568–574.
[3] P. Dollár, Z. Tu, P. Perona, and S. Belongie, "Integral channel features," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, London, U.K., 2009, pp. 1–11.
[4] X. Wang, T. X. Han, and S. Yan, "An HOG-LBP human detector with partial occlusion handling," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Kyoto, Japan, Sep./Oct. 2009, pp. 32–39.
[5] S. Dai, M. Yang, Y. Wu, and A. Katsaggelos, "Detector ensemble," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Minneapolis, MN, USA, Jun. 2007, pp. 1–8.
[6] Y. Tian, P. Luo, X. Wang, and X. Tang, "Deep learning strong parts for pedestrian detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1904–1912.
[7] D. Tomè, F. Monti, L. Baroffio, L. Bondi, M. Tagliasacchi, and S. Tubaro, "Deep convolutional neural networks for pedestrian detection," *Signal Process., Image Commun.*, vol. 47, pp. 482–489, Sep. 2016.
[8] W. Ouyang and X. Wang, "A discriminative deep model for pedestrian detection with occlusion handling," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, Jun. 2012, pp. 3258–3265.
[9] V. N. Vapnik, "An overview of statistical learning theory," *IEEE Trans. Neural Netw.*, vol. 10, no. 5, pp. 988–999, Sep. 1999.
[10] D. Silver, J. Schrittwieser, K. Simonyan, I Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, and D. Hassabis, "Mastering the game of go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, 2017.
[11] H. Li, Y. Li, and F. Porikli, "DeepTrack: Learning discriminative feature representations online for robust visual tracking," *IEEE Trans. Image Process.*, vol. 25, no. 4, pp. 1834–1848, Apr. 2015.
[12] L. Breiman, "Bias, variance, and arcing classifiers," Tech. Rep., Apr. 1996.
[13] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Statist.*, vol. 29, no. 5, pp. 1189–1232. Oct. 2001.
[14] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 785–794.
[15] W. XingFen, Y. Xiangbin, and M. Yangchun, "Research on user consumption behavior prediction based on improved XGBoost algorithm," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Seattle, WA, USA, Dec. 2018, pp. 4169–4175.
[16] Y. Xia, C. Liu, Y. Li, and N. Liu, "A boosted decision tree approach using Bayesian hyper-parameter optimization for credit scoring," *Expert Syst. Appl.*, vol. 78, pp. 225–241, Jul. 2017.
[17] F. Zhang, Y. Liu, C. Zou, and Y. Wang, "Hand gesture recognition based on HOG-LBP feature," in *Proc. IEEE Int. Instrum. Meas. Technol. Conf. (IMTC)*, May 2018, pp. 1–6.
[18] S. Paisitkriangkrai, C. Shen, and J. Zhang, "Fast pedestrian detection using a cascade of boosted covariance features," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 8, pp. 1140–1151, Aug. 2008.
[19] F. Lamberti, R. Santomo, A. Sanna, and P. Montuschi, "Intensity variation function and template matching-based pedestrian tracking in infrared imagery with occlusion detection and recovery," *Proc. SPIE*, vol. 54, no. 3, 2015, Art. no. 033106.
[20] A. Fernández-Caballero, M. López, and J. Serrano-Cuerda, "Thermal-infrared pedestrian ROI extraction through thermal and motion information fusion," *Sensors*, vol. 14, no. 4, pp. 6666–6676, 2014.
[21] P. Viola and M. J. Jones, "Robust real-time face detection," *Int. J. Comput. Vis.*, vol. 57, no. 2, pp. 137–154, 2004.
[22] R. Benenson, M. Mathias, T. Tuytelaars, and L. Van Gool, "Seeking the strongest rigid detector," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Portland, OR, USA, Jun. 2013, pp. 3666–3673.
[23] P. Dollár, R. Appel, S. Belongie, and P. Perona, "Fast feature pyramids for object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 8, pp. 1532–1545, Aug. 2014.
[24] S. Hinterstoisser, V. Lepetit, S. Ilic, P. Fua, and N. Navab, "Dominant orientation templates for real-time detection of texture-less objects," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, San Francisco, CA, USA, Aug. 2010, pp. 2257–2264.
[25] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *Amer. Statist.*, vol. 46, no. 3, pp. 175–185, Feb. 2012.
[26] M. Hentrich, "Methodology and coronary artery disease cure," Social Sci. Electron. Publishing, Rochester, NY, USA, Tech. Rep., 2015.
[27] J. Friedman, T. Hastie, and R. Tibshirani, *The Elements of Statistical Learning* (Springer Series in Statistics), vol. 1, no. 10. New York, NY, USA, 2001.
[28] F. L. Chang, X. Liu, and H.-J. Wang, "Target tracking algorithm based on meanshift and Kalman filter," *Jisuanji Gongcheng yu Yingyong (Comput. Eng. Appl.)*, vol. 43, no. 12, pp. 50–52, 2007.
[29] A. Salhi and A. Y. Jammoussi, "Object tracking system using Camshift, Meanshift and Kalman filter," *World Acad. Sci. Eng. Technol.*, vol. 6, no. 64, pp. 674–679, 2012.
[30] J. Sun, "A fast MEANSHIFT algorithm-based target tracking system," *Sensors*, vol. 12, no. 6, pp. 8218–8235, 2012.
[31] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.
[32] A. J. Lipton, H. Fujiyoshi, and R. S. Patil, "Moving target classification and tracking from real-time video," in *Proc. 4th IEEE Workshop Appl. Comput. Vis. (WACV)* Oct. 1998, pp. 8–14.
[33] S. Arseneau and J. R. Cooperstock, "Real-time image segmentation for action recognition," in *Proc. IEEE Pacific Rim Conf. Commun., Comput. Signal Process. (PACRIM)*, Victoria, BC, Canada, Aug. 1999, pp. 86–89.
[34] P. Shih and C. Liu, "Face detection using distribution-based distance and support vector machine," in *Proc. 6th Int. Conf. Comput. Intell. Multimedia Appl. (ICCIMA)*, Las Vegas, NV, USA, Aug. 2005, pp. 327–332.
[35] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 4, pp. 743–761, Apr. 2012.
[36] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: A benchmark," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Miami, FL, USA, Jun. 2009, pp. 304–311.
[37] D. M. Gavrila and S. Munder, "Multi-cue pedestrian detection and tracking from a moving vehicle," *Int. J. Comput. Vis.*, vol. 73, no. 1, pp. 41–59, Jun. 2007.

[38] F. Xu, X. Liu, and K. Fujimura, "Pedestrian detection and tracking with night vision," *IEEE Trans. Intell. Transp. Syst.*, vol. 6, no. 1, pp. 63–71, Mar. 2005.
[39] R. O'Malley, E. Jones, and M. Glavin, "Detection of pedestrians in far-infrared automotive night vision using region-growing and clothing distortion compensation," *Infr. Phys. Technol.*, vol. 53, no. 6, pp. 439–449, Nov. 2010.
[40] A. Jain, J. Tompson, Y. LeCun, and C. Bregler, "MoDeep: A deep learning framework using motion features for human pose estimation," in *Proc. Asian Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 302–315.
[41] T. Yu, X. Fan, and H. Shin, "An efficient pedestrian detection method by using coarse-to-fine detection and color histogram similarity," in *Proc. Int. Conf. Hybrid Inf. Technol.* Berlin, Germany: Springer, 2012, pp. 357–364.
[42] G. Overett and L. Petersson, "Fast features for time constrained object detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops*, Miami, FL, USA, Jun. 2009, pp. 23–30.
[43] P. F. Felzenszwalb, R. B. Girshick, D. Mcallester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.
[44] H. Zhang, Y. Du, S. Ning, Y. Zhang, S. Yang, and C. Du, "Pedestrian detection method based on Faster R-CNN," in *Proc. 13th Int. Conf. Comput. Intell. Secur. (CIS)*, Dec. 2017, pp. 427–430.
[45] M. S. Oughali, M. Bahloul, and S. A. El Rahman, "Analysis of NBA players and shot prediction using random forest and XGBoost models," in *Proc. Int. Conf. Comput. Inf. Sci. (ICCIS)*, Apr. 2019, pp. 1–5.
[46] L. Zhu, G. Xiong, and W. Yu, "Radar HRRP group-target recognition based on combined methods in the backgroud of sea clutter," in *Proc. Int. Conf. Radar (RADAR)*, Aug. 2018, pp. 1–6.
[47] T. Sledevič, A. Serackis, and D. Plonis, "FPGA-based selected object tracking using LBP, HOG and motion detection," in *Proc. IEEE 6th Workshop Adv. Inf., Electron. Elect. Eng. (AIEEE)*, Nov. 2018, pp. 1–5.
[48] W.-C. Kao, K.-J. Huang, Y.-C. Chiu, M.-Y. Hung, and C.-W. Shen, "Real-time image magnifier with visible-spectrum gaze tracker," in *Proc. Int. Symp. Consum. Technol. (ISCT)*, May 2018, pp. 22–23.
[49] F. Lamberti, A. Sanna, G. Paravati, and L. Belluccini, "IVF3: Exploiting intensity variation function for high-performance pedestrian tracking in forward-looking infrared imagery," *Proc. SPIE*, vol. 53, no. 2, 2014, Art. no. 023105.

YU JIANG is currently pursuing the degree in computer technology with the School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai, China. Her research interests include image processing and target detection.



GUOXIANG TONG is currently pursuing the Ph.D. degree with the School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology (USST), where she is also an Assistant Professor. Her major research interests include application of computer control, design and develop of embedded systems, digital data processing, and artificial intelligence.



HENAN YIN is currently pursuing the degree in computer technology with the School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai, China. His research interests include machine learning, data mining, and computer vision.

NAIXUE XIONG received the Ph.D. degree in sensor system engineering from Wuhan University, and the Ph.D. degree in dependable sensor networks from the Japan Advanced Institute of Science and Technology. Before he attended Northeastern State University, he was with Georgia State University, the Wentworth Technology Institution, and Colorado Technical University (full professor about five years) about ten years. He is currently with the School of Computer and Science, Tianjin University, China. His research interests include cloud computing, security and dependability, parallel and distributed computing, sensor networks, and optimization theory.