IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 8, 2019, accepted June 29, 2019, date of publication July 11, 2019, date of current version July 25, 2019.

*Digital Object Identifier 10.1109/ACCESS.2019.2928224*

# A Cascade Learning Approach for Automated Detection of Locomotive Speed Sensor Using Imbalanced Data in ITS

**BO LI[1], (Member, IEEE), SISI ZHOU[1], LIFANG CHENG[1], RONGBO ZHU[1], (Member, IEEE), TAO HU[1], ASHIQ ANJUM[2], ZHENG HE[3], YONGKAI ZOU[1]**

[1]College of Computer Science, South-Central University for Nationalities, Wuhan 430074, China
[2]College of Engineering and Technology, University of Derby, Derby DE22 1GB, U.K.
[3]Wuhan Japir Technology Company Ltd., Wuhan 430000, China

Corresponding author: Rongbo Zhu (rbzhu@mail.scuec.edu.cn)

This work was supported in part by the Fundamental Research Funds for the Central Universities under Grant CZP19004, in part by the National Natural Science Foundation of China under Grant 61772562, in part by the Hubei Provincial Natural Science Foundation of China for Distinguished Young Scholars under Grant 2017CFA043, in part by the Hubei Provincial Science and Technology Innovation Foundation of China under Grant 2018ABB1485, and in part by the Youth Elite Project of State Ethnic Affairs Commission of China under Grant 2016.

**ABSTRACT** Automatic and intelligent railway locomotive inspection and maintenance are fundamental issues in high-speed rail applications and intelligent transportation system (ITS). Traditional locomotive equipment inspection is carried out manually on-site by workers, and the task is exhausting, cumbersome, and unsafe. Based on computer vision and machine learning, this paper presents an approach to the automatic detection of the locomotive speed sensor equipment, an important device in locomotives. Challenges to the detection of speed sensor mainly concerns complex background, motion blur, muddy noise, and variable shapes. In this paper, a cascade learning framework is proposed, which includes two learning stages: target localization and speed sensor detection, to reduce the complexity of the research object and solve the imbalance of samples. In the first stage, histogram of oriented gradient feature and support vector machine (HOG-SVM) model is used for multi-scale detection. Then, an improved LeNet-5 model is adopted in the second stage. To solve the problem of the imbalance of positive and negative samples of speed sensor, a combination strategy which draws on four individual classifiers is designed to construct an ensemble of classifier for recognition, and the results of three different algorithms are compared. The experimental results demonstrate that our approach is effective and robust with respect to changes in speed sensor patterns for robust equipment identification.

**INDEX TERMS** Cascade learning, speed sensor detection, imbalanced data, ensemble learning.

## I. INTRODUCTION

Railway train fault inspection is an important procedure which has to be performed regularly for fear of possible accidents [1]–[3]. Among the many types of railway trains, locomotives are popular self-propulsion vehicles that is responsible for pulling railway vehicles and do not carry loads of their own, in which the locomotive speed sensor plays a vital role in the safe operation of locomotives. For a long time, the safety inspections of the speed sensor are conducted by maintenance workers by means of eye-measurement, running the risks of neglecting hidden dangers. In addition, it is time consuming and inefficient. In order to change the way of manual detection and improve work efficiency, automatic detection approaches based on advanced technologies such as computer vision have been proposed and gradually applied to the field of railway and train fault detection and intelligent transportation system (ITS) [4], [5]. At present, some common railway fault detection systems judge faults by analyzing fault images from repair station site. For example, a train fault detection system collects the image information of train devices (ie. the brake device, the bogie and the related equipment, etc.) through a series of high-definition cameras and records the speed of the train

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

  

(a) (b)

**FIGURE 1.** Locomotive image. (a) With speed sensor. (b) Without speed sensor.

and the type of train, displays the captured information on the computer, and determines whether there are cracks or missing parts in the key parts of the train.

In the last few decades, computer vision researchers have done a lot of work on train fault detection. Santur *et al.* [1] performed feature extraction on video images of normal railway tracks to examine faults which may occur on the surface of tracks, including scouring, rupture and weak fasteners, such as bolts and sleepers. In another approach, Aytekin *et al.* [2] proposed a system for detecting defective bolts on railway tracks using a laser camera. The classification system is trained by dimension reduction extraction of all seven-dimensional features. A detection system was designed which can detect missing or defective track fastener connection problems based on the combination of gradient direction histogram features and linear SVM classifiers [3]. Li *et al.* [4] presented an automatic rail track inspection for detecting railway pads and nails. Resendiz *et al.* [5] adopted texture classification by a set of Gabor filters and SVM to judge the position of the track device.

Different from the above work, this paper proposes an approach for speed sensor detection. The speed sensor device is a component in the locomotive that exhibits a change in speed through a change in the amount of electricity. It can be divided into a contact speed sensor and a non-contact speed sensor. The contact speed sensor works mainly based on the principle of electromagnetic induction, and coil induced voltage around the sensor is proportional to the speed of the train change. However, the non-contact speed sensor is a photoelectric type component, and the photoelectric pulse signal emitted by it is used to express the change of speed. Since it is minimally affected by external interference noise, its stability is better than that of the electromagnetic induction type speed sensor. The speed sensor equipment studied in this paper is shown in Fig.1(a). The red circular area is the area where the speed sensor is mounted. It belongs to the photoelectric type speed sensor, and the change of the running speed of the locomotive is transmitted to the device for monitoring the locomotive by a pulse signal. In general, from the outside, the speed sensor is a circular bulging device that is fixed by four screws (the four screws are distributed over the four corners of a square) on a disc. As shown in Fig. 1(b), the speed sensor does not exist.

This paper proposes an approach for detecting speed sensor based on machine learning by means of computer vision. The research of object detection via machine learning theory mainly includes two categories: traditional methods (such as SVM) and deep learning methods. In recent years, many researchers have carried out a lot of research work around these two kinds of methods.

In this paper, based on the characteristics of the locomotive images, a cascade learning approach is proposed which consists of two stages, including learning stage of object area localization and learning stage of speed sensor detection. Furthermore, according to sample distribution from the locomotive repair station, this paper chooses an optical sample training strategy for imbalanced samples.

The main contributions of this paper are summarized as follows.

1) Different from the common speed sensor equipment detection, which detects the speed sensor mostly using the internal detection method, that is, the signal band information of the speed sensor is used to judge whether there is a fault, this paper is based on computer vision and machine learning. And a detection approach is proposed to judge the presence of a speed sensor in the locomotive image in terms of the appearance of the image.

2) According to the imbalance of positive and negative samples, this paper proposes a cascade learning framework, combined with the ensemble learning method, to reduce the influence of sample imbalance and the complexity of objects to be recognized. In the ensemble learning, this paper designs a combination strategy of individual classifiers to construct an ensemble classifier for target recognition.

The remainder of this paper is organized as follows. Section II summarizes the relevant work in recent years. Section III introduces the framework of our appr-oach. Section IV presents the first stage learning for object region locating. In section V, the second stage learning for speed sensor detection is introduced. Section VI shows experimental results and illustrates the effectiveness of the proposed approach. Conclusions and prospects for future research are presented in section VII.

## II. RELATED WORK

In this paper, image analysis and machine learning method are used to locate the speed sensor device area in the locomotive image, and determine whether the speed sensor device exists or misses.

The speed sensor detection belongs to object detection problem. The research goal of our work is to locate the speed sensor installation area in complex background in the locomotive image and determine whether the speed sensor device is installed (or missing) in the installation area. Because speed sensors are often exposed to the field for a long time, their edges are often fuzzy, and many samples from the field are of poor quality (as shown in Fig. 2), the method of detecting the circle by image processing (i.e. Hough transform [6]) is not effective. Therefore, this paper locates and judges speed sensors using machine learning methods.

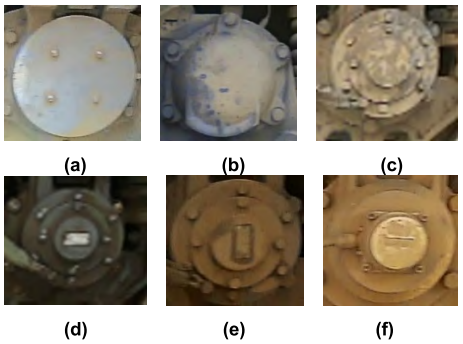


**FIGURE 2.** Poor quality samples. (a), (b), (c) Muddy. (d) and (e) Dark light. (f) Blurry edges.

In recent years, more and more researchers conducted object detection research by machine learning methods in computer vision field. Reference [7], the active learning and semi-supervised learning was adopted to improve the performance of object detection. Residual network and color and texture features are used in Reference [8] to obtain simplified image with multiple scales to increase network depth and to reduce the errors caused by depth increase simultaneously. In addition, Feature Pyramid Network with DenseNet method was proposed to overcome the problem of class confusion in the sample images [9]. Chen *et al.* [10] proposed a new multi-task learning method to jointly model object detection with a Cartesian product-based multi-task combination strategy. In particular, a completely automatic machine learning method is presented for quantifying the preferential mineral-microfracture relationships in intact and deformed shales in Reference [11]. Moreover, deep learning has been widely and successfully applied in recent years (i.e., References [12], [13]). It is especially worth mentioning that the R-CNN model [14] has also been widely used in target detection [15]. In traditional target detection and recognition methods, feature extraction and fusion [16], [17] need to be considered, but depth learning method itself can effectively extract features.

However, deep learning algorithm is easy to overfitting on small data sets, not better than nonlinear and linear kernel SVM, Bayesian classifier methods, etc. The SVM only needs a small amount of data to find the hyperplane of the classification between the data, and gets very good classification results. Many successful applications were reported in the field of target detection based on SVM (i.e., [18]–[21]). Therefore, it is necessary to choose the deep learning method and the traditional learning method according to the characteristics of the problem and the data.

In practical problems, sometimes the target and background are easily confused, and sometimes the number of positive and negative samples varies greatly, which leads to the imbalance of the samples. According to the characteristics and complexity of the research object, some researchers proposed an ensemble of classifiers, the structure of which is a cascade of binary classifiers, which allows each binary classifier in the cascade to process only one part of the non-object class [22]. In Reference [23], a cascade learning approach was proposed for the segmentation of tumour epithelium in colorectal cancer. Also, a coarse-to-fine pupil detection method was proposed based on shape augmented cascade regression in Reference [24]. Moreover, an iCascade model was presented which can search the optimal partition point of each stage by directly minimizing the computation cost of the cascade [25]. In Reference [26], a cascaded CNN architecture was proposed which consists of two stages for stereo matching. In addition, Prashant et al. presented a cascaded unpaired learning for background estimation and foreground segmentation [27]. Cascade classifiers are used in unbalanced sample classification [22] and are sometimes used to detect targets from coarse to fine [24], [26]. The measures used in the literature [22] and [25] for learning effects are not accuracy but rank metrics [22] and computation cost of the cascade [25].

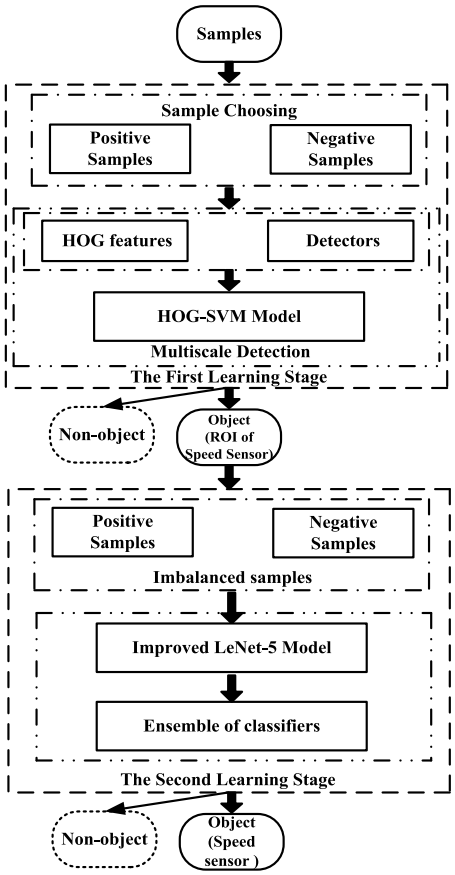


**FIGURE 3.** An overall flowchart of proposed framework.

## III. PROPOSED FRAMWORK

The proposed framework is shown as in Fig. 3.

There are several positions on two sides of a locomotive where the speed sensor can be installed, but under normal conditions, only one speed sensor is installed on a locomotive. If the samples containing the speed sensor are used

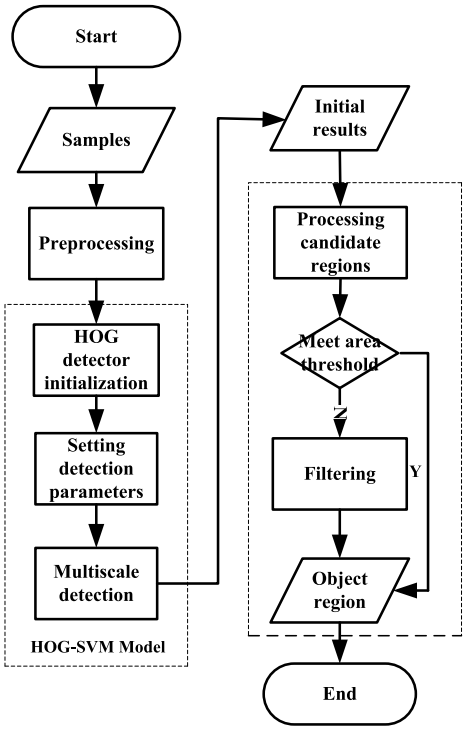


**FIGURE 4.** A flow chart of object region localization.

as positive samples, and the samples containing no speed sensors are used as negative samples, the natural ratio of the positive samples to the negative samples is unbalanced. Furthermore, if the background areas outside the speed sensor installation areas in the locomotive images are also used as negative samples, the proportion of negative and positive samples will be greater. Therefore, the number of positive and negative samples in this paper is very unbalanced. We adopt the idea of cascade learning, which is divided into two stages to learn and exclude non-objective categories, to reduce the impact of unbalanced samples and the complexity of the research object.

As shown in Fig. 3, the cascade learning used in this paper is divided into two stages. In these two stages, different positive samples and negative samples are selected respectively. In the first stage, we carry out multi-scale target area detection based on SVM-HOG model. In the second stage, taking into account the imbalance of positive and negative samples, we construct four individual learners using the improved LeNet-5 model, design a combination strategy, and form an ensemble learning classifier. According to the target area obtained in the previous step, the target judgment is carried out.

## IV. FIRST STAGE LEARNING FOR OBJECT REGION LOCALIZATION

The research goal of this section is to locate the region of interest (ROI), that is, determine the installation area of the speed sensor. The flow chart of object region localization is shown in Fig. 4.

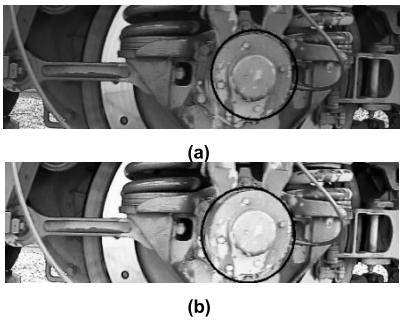


**FIGURE 5.** Preprocessing. (a) Original sample. (b) Result after histogram equalization.


**FIGURE 6.** HOG feature extraction. (a) Original image. (b)HOG extraction.

### A. PREPROCESSING

Because the quality of the samples is not good, it is necessary to pre-process the samples. The histogram equalization [28] is suitable for the processing of original images, which are often dark or bright overall, and after histogram equalization the image contrast is greatly enhanced (As shown in Fig. 5).

The research goal of this section is to locate the target area (ROI) where speed sensor may be installed, from images captured by the locomotive repair station when a locomotive arrives, and that is, the region with black circle in Fig. 5. This paper uses HOG features and SVM classifier (this paper calls it HOG-SVM classifier, as Reference [29]) in the first learning stages. At this stage, the positive samples used in this paper are the samples of the region of interest (including two cases: containing the speed sensor and not containing the speed sensor), and the negative sample is the background region outside the region of interest.

### B. HOG FEATURES

The Histogram of Oriented Gradient (HOG) feature was proposed in Reference [29]. This feature is often used in conjunction with SVM for object detection. The HOG feature is a feature descriptor for detecting an object, which is obtained by calculating a gradient histogram of a local region of an image (As Fig. 6).

The HOG feature calculation steps are as follows:

(1) Calculate the gradient direction and size of each pixel in the sample:

$$G_x(x, y) = H(x + 1, y) - H(x - 1, y) \quad (1)$$

$$G_y(x, y) = H(x, y + 1) - H(x, y - 1) \quad (2)$$

$$G(x, y) = \sqrt{G_x(x, y)^2 + G_y(x, y)^2} \quad (3)$$

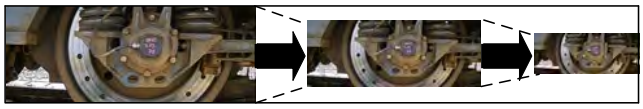

**FIGURE 7.** Multiscale detection.

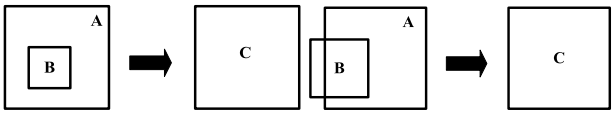


**FIGURE 8.** Relation of candidate rectangular areas. (a) Overlap. (b) Intersection.

$$\alpha\,(\mathrm{x},\mathrm{y}) = tan^{-1}\frac{G_y(x,y)}{G_x(x,y)} \tag{4}$$

where G (x, y) and $\alpha$ (x, y) represent the amplitude value and the direction of the gradient respectively.

(2) Divide the sample into several $8\times8$ size cells.

(3) The gradient histogram information of each cell is counted as its feature vector, and then the adjacent four cells are combined into one block to obtain the feature vector of each block.

(4) Combining the feature vectors of all blocks forms the HOG feature vector required for classification.

### C. MULTISCALE DETECTION USING HOG-SVM MODEL

SVM classifier is employed to determine the weight coefficient and threshold of the optimal hyperplane by training, and obtain an optimal classification surface that can divide the training positive and negative samples into two categories.

The HOG-SVM model detection mainly relies on the HOG feature generated by SVM classifier training, and the HOG detector performs multiscale detection [30], [31] on the image to be detected. Since the size of the object to be detected is not fixed, and the size of the sliding window for image traversal is constant, the image to be detected needs to continuously reduce its size so that the target area can be detected when the sample image is traversed. Different detection results can be obtained by adjusting the sliding step of the detection window and the proportion of the image to be detected. As shown in Fig. 7, in order to detect targets of different sizes, multiscale detection method is used to traverse and detect the sample image, gradually reducing the size of the image according to a certain ratio.

For candidate rectangular areas obtained by multiscale detection, it is determined whether there is overlap or intersection, and if it exists, the outermost rectangle (as Fig. 8(a)) or the rectangle with a larger area (as Fig. 8(b)) is reserved, as Formula (5). The result of post-processing candidate areas is shown as Fig. 9.

$$\mathrm{C} = \max\{\mathrm{A},\mathrm{B}\}\ (\text{if } \mathrm{A}\cap\mathrm{B}\neq\emptyset) \tag{5}$$

Then, a filtering using area threshold is performed on the candidate rectangular areas, and non-target areas


**FIGURE 9.** Post-processing candidate areas. (a) Multi-candidate areas. (b) Final result.

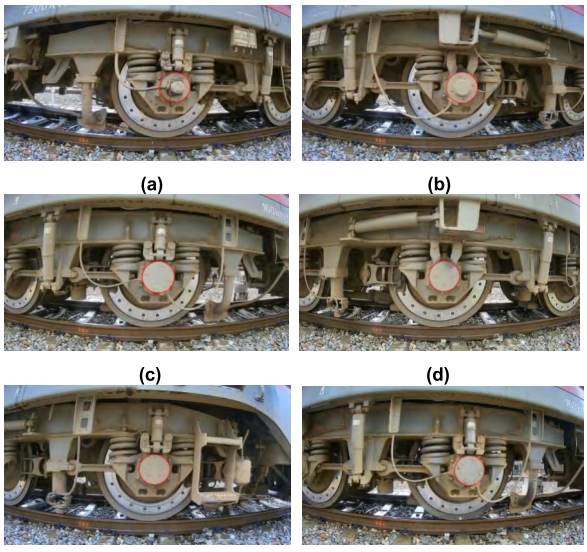


**FIGURE 10.** Regions of interest of speed sensor in a locomotive. (a) With speed sensor. (b)-(f) Without speed sensor.

are excluded, and the final remaining rectangular area is marked in the image.

## V. SECOND LEARNING STAGE FOR SPEED SENSOR DETECTION

After locating the installation area of the speed sensor, this section will judge whether a speed sensor is installed (or missing) in the area.

The positive samples in this section are the areas where the speed sensor is installed (as Fig. 10(a)), and the negative samples are the areas without speed sensor (as Fig. 10 (b)-(f)). In fact, because the speed sensor installation position of each locomotive is different, the locomotive repair station needs to shoot six suspected areas each time, as shown in Fig. 10, the round red frame in the figure is the region of interest.

### A. SAMPLE CLASS IMBALANCE PROBLEM

If the samples with speed sensors are taken as positive samples and those without speed sensors as negative samples, the natural proportion of these two types of samples shows that the number of positive samples is smaller than that of negative samples. Therefore, this is a class imbalance problem. For general classifier training, it is assumed that the positive and negative samples are proportionally balanced, but in practical applications, the training sample data

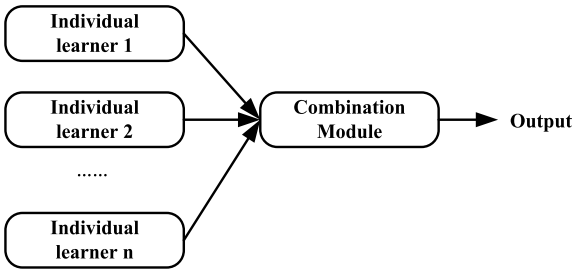


**FIGURE 11.** Ensemble learning.

is often unbalanced. Imbalanced learning hitherto remains a valuable research topic. There are several common methods for dealing with data of imbalance: sampling, ensemble, weighting, and classification [32]–[38]. An in depth review of rare event detection from imbalanced learning is provided in Reference [32]. In Reference [33], the assessment metrics for evaluating learning performance under the imbalanced learning scenario were proposed. Many techniques are adopted to solve the imbalance data problem, such as attribute selection [34], resampling algorithm [35], clustering [36], modified classifier [37], [38], etc.

Sampling is a case where both positive and negative sample data are sufficiently large and the ratios are not very large. It includes both oversampling and undersampling. Oversampling is the process of copying a few classes many times, making a few classes have a similar number to most classes. There are two extreme results in this type of oversampling, either the training results are very good or there are a lot of misclassifications. The solution is to add lightweight random interference each time new data is generated by replication. Undersampling is the removal of a portion of the data from most classes. Although this method does not produce overfitting as in oversampling, the lack of data may result in inaccurate classification results.

In cases where both positive and negative samples are large and the ratio is not particularly disparate, sampling or weighting should be considered. EasyEnsemble [32], [39] uses multiple undersampling (putting back the samples so that the resulting training sets are independent of each other) to produce multiple different training sets, and then trains multiple different classifiers. The final result is obtained by combining the results of multiple classifiers. This work adopts undersampling to eliminate the imbalance of sample data, and uses ensemble learning method to avoid the lack of classification accuracy due to missing data.

A common classifier combination strategy is to use the averaging method, as show in Fig. 11, the individual classifiers are combined as an ensemble classifier by a certain strategy, and then the output of the ensemble classifier is generated.

For numerical output, $c_i(x) \in R$, then the output of ensemble classifier C (x) is shown as:

$$C(x) = \frac{1}{N} \sum_{i=1}^{N} c_i(x) \tag{6}$$

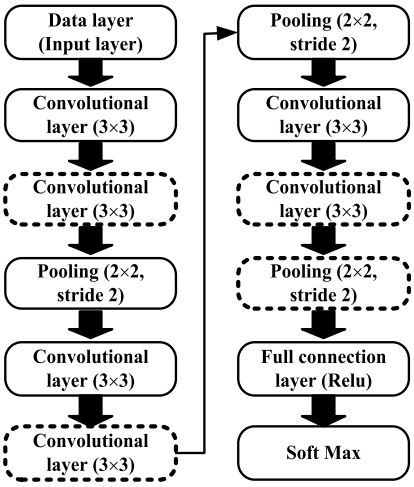


**FIGURE 12.** The improved LeNet-5 model.

### B. OBJECT DETECTION

In the target detection module, based on the typical LeNet-5 model [40], we design an improved model, as shown in Fig. 12, the convolution layer and pooling layer indicated by the dashed wireframe are the added parts of the improved model.

In the improved network structure, some convolution layers and pooling layers are added to the original LeNet-5 network model, which consists of 6 convolution layers and 3 pooling layers, a full connection layer, deepens the original network structure. At the same time, the number of convolution cores per convolution layer is increased.

The reasons for the improvements are as follows:

1) The LeNet-5 model was originally designed for handwritten character set recognition. In the input part, the handwritten data sets originally trained by LeNet-5 model are images with a size of 28×28. Since the samples in this work are locomotive speed sensor images, which are more complex than that of the handwritten dataset. Therefore, this work uses 64× 64 size images as data input, considering color information, the resolution of input data samples is 3×64×64.

2) In convolution layer, we use two 3×3 convolution cores instead of original one 5×5 convolution layer. Two 3×3 convolution layers have fewer parameters than a larger one. Compared with a convolution layer with a large convolution core, two 3×3 convolution layers can increase the nonlinearity and complexity of the network.

The research goal of this section is to judge whether the speed sensor is installed in the region of interest obtained from the previous step. Unlike the first step, we reset the positive and negative samples. The positive samples include speed sensors, and the negative samples do not contain speed sensors.

In this section, the improved LeNet-5 network model is used to judge whether the speed sensor device is installed, the object is marked with a thick white rectangle, and for the sample without the speed sensor, the ROI is marked with a fine white rectangle, as shown in Fig.13.

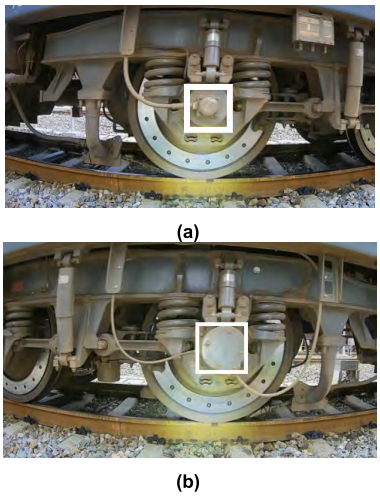

**FIGURE 13.** An example of detection results. (a) Containing speed sensor. (b) No speed sensor.

**FIGURE 14.** Monitoring images at locomotive repair station.

## VI. EXPERIMENT AND DISCUSSION

### A. PARAMETER SETTINGS

Our approach is developed based on Visual Studio 2010 and OpenCV2.4.9, and the improved LeNet-5 model is developed using PyCharm and TensorFlow 1.6.

The samples are obtained from the locomotive repair station. Fig. 14 shows some examples of monitoring images at locomotive repair station.

The experimental parameters are set as follows.

#### 1) OBJECT REGION LOCALIZATION

The training of HOG-SVM model for localization mainly includes sample feature extraction and classifier training. The sample size is uniformly set to 64×64 pixels. This paper chooses C_SVC as the type of SVM, the kernel function selects the linear kernel, and the penalty parameter C is set to 0.01. SVM training is performed on the training parameters, and the 1764 dimensional feature vector and the positive and negative marker matrix.

In the detection process, the size of original test images is 1920×1152. In initializing a HOG detector, we set the detection window size to 64×64, the block size to 16×16, the cell size to 8×8, and the sliding step of the detection window to 8×8.

In multiscale detection, the sliding step parameter setting of the detection window is the same as the sliding window size of the HOG detector, and the reduction ratio of the detection image is set to 1.05.

#### 2) SPEED SENSOR DETECTION

①Initializing the network

According to the improved LeNet-5 structure, the size and characteristic plane of each layer parameters are initialized by normal distribution function, and all bias terms are set to 0.

The input layer is a 64×64 resolution image, and each pixel of the first convolution layer is connected to the 3×3 sensory fields of the input layer. The first convolution layer has 32 convolution cores, therefore, the first convolution layer extracts a total of 32 feature planes and 32 features from the input layer. The second convolution layer extracts 32 feature planes with a feature plane size of 32×32. The third convolution layer and the fourth convolution layer consist of 64 feature planes, but the feature plane size becomes 16 ×16. The fifth convolution layer consists of 128 feature planes, each of which is 16 × 16 in size using of padding filling. The sixth convolution layer is also composed of 128 feature planes, and the size of the feature plane becomes 8×8. An intermediate layer consisting of 128 neurons is fully connected, followed by an output layer of two neurons, representing sensors and non-sensors, respectively.

The initial learning rate is 0.002, adjusted to 0.01 after 30 epochs to speed up the training.

②Forward propagation

Every time the convolution operation is completed, the ReLu function is activated. The activation process is run to process each element of the output feature map by the ReLu function.

The pooling operation uses the maximum pooling, with a window of 2×2 and a sliding step of 2. The last pooling layer outputs a 3-D feature map of 8×8×128. The input of the full junction layer is fully connected with the intermediate layer with 128 neurons in the back. Because of the binary classification, the number of neurons in the output layer is 2, and the index number of the elements with higher confidence is taken as the result of classification (0 means there is a speed sensor, and 1 means there is no speed sensor).

③Backpropagation

In backpropagation process, we use the cross-entropy loss function to calculate the cross-entropy between the two output results of the output neuron and the label made by ourselves, and then get the loss.

In the training process, batch is 30, that is, every 30 samples for weight adjustment, training step is 50000.

Regularization is adopted to avoid overfitting, as Formula (7).

$$C = C_0 + \frac{\lambda}{2n} \sum_w w^2 \tag{7}$$

**TABLE 1. Comparison of experimental results of three localization methods.**

| Algorithm | Hough | Haar-Adaboost | HOG-SVM |
|---|---|---|---|
| Localization precision (%) | 96.9 | 96.8 | 99.4 |

where the L2 regularization method is used to prevent overfitting, that is to say, a regularization term is added after the cost function. C0 represents the original cost function, and the latter term is the L2 regularization term, w is the training parameter, n is the sample size of the training set, and λ is the regular term coefficient.

### B. EVALUATION METRICS

Some evaluation metrics are frequently adopted to provide comprehensive assessments of imbalanced learning, namely, precision, recall, F-measure, and G-mean [33]. These metrics are defined as follows:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{8}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{9}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{10}$$

$$\text{F} - \text{Measure} = \frac{(1 + \beta^2) \cdot Recall \cdot Precision}{\beta^2 \cdot Recall + Precisioin} \tag{11}$$

where $\beta$ is a coefficient to adjust the relative importance of precision versus recall (usually $\beta = 1$).

$$\text{G} - \text{mean} = \sqrt{\frac{TP}{TP + FN} \times \frac{TN}{TN + FP}} \tag{12}$$

where precision, recall and accuracy are composed of true positive (TP, true speed sensors are detected), false positive (FP, false speed sensors are detected), false negative (FN, true speed sensors are not detected), true negative (TN, false speed sensors are not detected).

### C. EXPERIMENTAL RESULTS AND COMPARISONS

#### 1) OBJECT REGION LOCALIZATION

In this stage, we adopted about 6700 positive samples and 16000 negative samples. The comparison experiments were carried out using 1000 test samples. The same test samples are used for each experimental comparison. In addition to HOG-SVM model, the other two algorithms for target area localization are compared, which are Hough detection algorithm and Haar-Adaboost model (using Haar features and Adaboost classifier) [41], [42].

Some training samples are shown as in Fig. 15.

The experimental results are shown in Table 1. The HOG-SVM algorithm is 2.6% higher in localization accuracy than the Haar-Adaboost model. The reason why the accuracy of the Haar-Adaboost model is lower is that the type change of the positive and negative samples is not large, and the false alarm is easily reached during the training process.

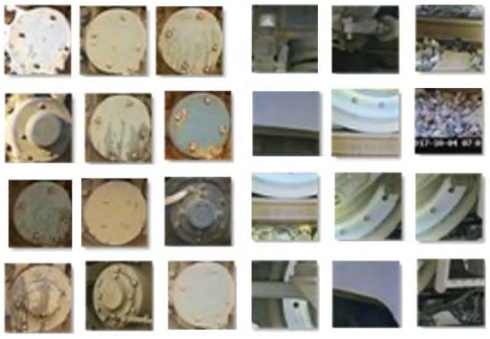


**FIGURE 15. Training samples in the first learning stage. (a) Positive sample with a ROI region. (b) Negative sample without a ROI region.**

In addition, the results detected by the Haar-Adaboost model are more than one in most cases, that is, the false detection rate is high. The background to be detected is complex and the feature distinguishing between the target localization area (ROI) and its surrounding area is not very obvious, which leads to locate more than one target area. If the detection window is set too large, the miss detection rate will increase, otherwise, the multi-detection rate will increase.

The HOG-SVM model can meet the real-time requirement in the practical application scenario. In the case where the time performance is acceptable, it is more focused on the localization accuracy. Therefore, the HOG-SVM algorithm is more suitable as a localization algorithm for the target region.

Also, Hough circle detection algorithm has similar localization accuracy to Haar-Adaboost model. However, the correct rate detected by the Hough algorithm is related to the quality of the detected image, and the accuracy of the Haar-Adaboost model is related to training samples. From the results shown in Table 1, the machine learning based detection methods is more robust than image processing based method.

#### 2) SPEED SENSOR DETECTION

In this stage, this paper uses the ensemble learning method for target detection. We have carried on the new arrangement and the selection to the sample set, including a small number of positive samples with speed sensors. The training set consists of 600 positive samples and 2400 negative samples, and the test set includes 200 positive samples and 200 negative samples. Because of the large difference in the number of positive and negative samples, it belongs to the imbalance problem of sample category. We train based on ensemble classifier method and divide the negative samples

**TABLE 2.** Experimental results of ensemble classifier using SVM.

| Training strategy | TP | FP | TN | FN | ACC(%) | Precision(%) | Recall(%) | F-Measure(%) | G-mean(%) |
|---|---|---|---|---|---|---|---|---|---|
| Pos600+Neg2400 | 191 | 2 | 198 | 9 | 97.25 | 98.96 | 95.50 | 97.20 | 97.23 |
| Pos600+Neg600_1 | 196 | 10 | 190 | 4 | 96.50 | 95.15 | 98.00 | 96.55 | 96.49 |
| Pos600+Neg600_2 | 195 | 3 | 197 | 5 | 98.00 | 98.48 | 97.50 | 97.99 | 98.00 |
| Pos600+Neg600_3 | 195 | 20 | 180 | 5 | 93.75 | 90.70 | 97.50 | 93.98 | 93.67 |
| Pos600+Neg600_4 | 193 | 3 | 197 | 7 | 97.50 | 98.47 | 96.50 | 97.47 | 97.49 |
| Ensemble classifier (Averaging) | 195 | 8 | 192 | 5 | 96.75 | 96.06 | 97.50 | 96.77 | 96.75 |
| Ensemble classifier (Extreme value) | 194 | 9 | 191 | 6 | 96.25 | 95.57 | 97.00 | 96.28 | 96.25 |

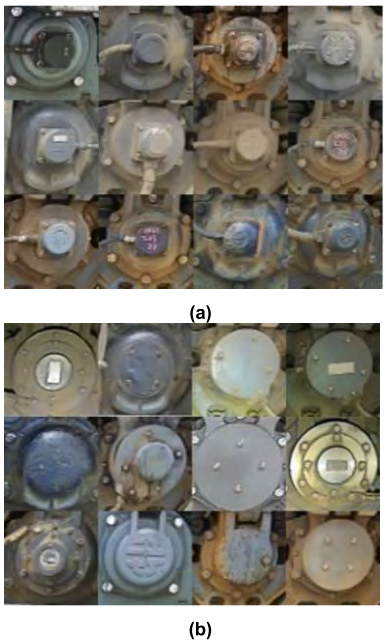


**FIGURE 16.** Positive samples and negative samples in the second learning stage. (a) Positive sample. (b) Negative sample.

into four groups. These four groups of negative samples and 600 positive samples form a small training set of 1:1 to obtain four individual classifiers. Then, four individual classifiers are combined to obtain the final classification results. In this way, for each individual classifier, the positive and negative samples in the training set are balanced, but the negative samples in the four small training sets are not repeated, and they can cover the overall information of the negative sample set together.

Some training samples are shown as Fig. 16.

In this stage, we reset the positive and negative samples. The positive samples include speed sensors, and the negative samples do not contain speed sensors.

According to the number of positive samples and negative samples, the negative samples are equally divided into four groups, and four individual classifiers are used for ensemble learning. We use the SVM (still using HOG features), LeNet-5 and the improved LeNet-5 model to carry on the detection experiment. Considering that the number of individual classifiers is 4 (even), the SVM ensemble classifier may not be able to classify effectively according to the conventional averaging and voting methods. Since the number of individual classifiers is 4, if the average method and the voting method are used, the result may be zero and the correct result may not be obtained. Therefore, this paper proposes an improved averaging method which averages the distance from the support vector to the classification plane in individual classifiers, and an extreme value method which chooses the extreme value of the distance between the support vectors and the classification plane.

The distance from the sample to the hyperplane is used as the result of the individual classifier, as

$$C = \frac{1}{n}\sum_{i=1}^{n} Dist_i \tag{13}$$

where $Dist_i$ is the distance from the $i^{st}$ sample to the hyperplane, and C denotes the final output result of the SVM ensemble classifier.

In LeNet-5, and improved LeNet-5 model, ensemble classifiers are used to sum the two confidence levels of the output of four individual classifiers, and then take the corresponding categories of the higher confidence levels as the final classification results, as:

$$C = Max(\sum_{i=1}^{n} CFy_i, \sum_{i=1}^{n} CFn_i) \tag{14}$$

where let $CFy_i$, $CFn_i$ be the confidence corresponding to the binary judgment result (Yes, No) in the LeNet-5 individual classifier, and C denotes the final output result of the LeNet-5 ensemble classifier. Max represents the maximum value.

The complete set of negative samples is divided into four subsets, which are respectively $A_1$,$A_2$,$A_3$, $A_4$.

$$A_i \cap A_j = \emptyset \text{ and } \bigcup_{i=1}^{4} A_i = I\,(i \neq j) \tag{15}$$

where I is a full set of negative samples.

We use three classifiers, including SVM, LeNet-5, improved LeNet-5 model. Among them, in the SVM, two kinds of ensemble methods, the improved average method and the extreme value method are adopted. As can be seen from Tables 2, Table 3 and Table 4, Pos600+Neg2400 indicated that 600 positive samples and 2400 negative samples are used. Pos600+Neg600_1, Pos600+Neg600_2,

**TABLE 3.** Experimental results of ensemble classifier using original LeNet-5.

| Training strategy | TP | FP | TN | FN | ACC(%) | Precision(%) | Recall(%) | F-Measure(%) | G-mean(%) |
|---|---|---|---|---|---|---|---|---|---|
| Pos600+Neg2400 | 181 | 5 | 195 | 19 | 94.00 | 97.31 | 90.50 | 93.78 | 93.93 |
| Pos600+Neg600_1 | 173 | 7 | 193 | 27 | 91.50 | 96.11 | 86.50 | 91.05 | 91.36 |
| Pos600+Neg600_2 | 190 | 17 | 183 | 10 | 93.25 | 91.79 | 95.00 | 93.37 | 93.23 |
| Pos600+Neg600_3 | 188 | 27 | 173 | 12 | 90.25 | 87.44 | 94.00 | 90.60 | 90.17 |
| Pos600+Neg600_4 | 193 | 21 | 179 | 7 | 93.00 | 90.19 | 96.50 | 93.24 | 92.93 |
| Ensemble classifier | 195 | 11 | 189 | 5 | 96.00 | 94.66 | 97.50 | 96.06 | 95.99 |

**TABLE 4.** Experimental results of ensemble classifier using improved LeNet-5.

| Training strategy | TP | FP | TN | FN | ACC(%) | Precision(%) | Recall(%) | F-Measure(%) | G-mean(%) |
|---|---|---|---|---|---|---|---|---|---|
| Pos600+Neg2400 | 183 | 3 | 197 | 17 | 95.00 | 98.39 | 91.50 | 94.82 | 94.94 |
| Pos600+Neg600_1 | 189 | 7 | 193 | 11 | 95.50 | 96.43 | 94.50 | 95.45 | 95.49 |
| Pos600+Neg600_2 | 196 | 11 | 189 | 4 | 96.25 | 94.69 | 98.00 | 96.31 | 96.23 |
| Pos600+Neg600_3 | 194 | 5 | 195 | 6 | 97.25 | 97.49 | 97.00 | 97.24 | 97.25 |
| Pos600+Neg600_4 | 195 | 18 | 182 | 5 | 94.25 | 91.55 | 97.50 | 94.43 | 94.19 |
| Ensemble classifier | 197 | 3 | 197 | 3 | 98.50 | 98.50 | 98.50 | 98.50 | 98.50 |

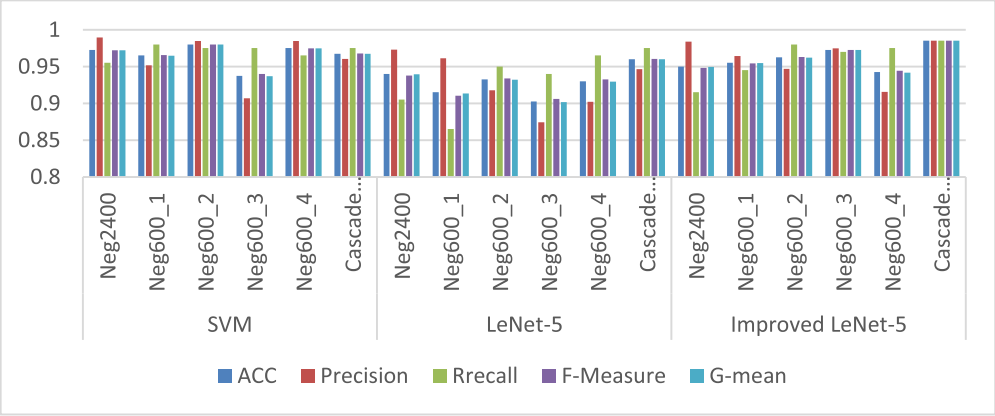


**FIGURE 17.** Comparison of the judging results of speed sensors based on three kinds of algorithms.

Pos600+Neg600_3, and Pos600+Neg600_4, represents the combination of training samples of four individual classifiers, all of which are 600 positive samples and 600 negative samples. Neg600_1, Neg600_2, Neg600_3, and Neg600_4 are a small subset that averages 2400 negative samples into 4 non-repetitive subsets.

Although the Recall of ensemble classifier in Table 3 is higher than that in Table 2, in general, the other indicators of SVM are higher than the LeNet-5 model. The average concatenation of SVM is better than that of extreme concatenation. In the SVM ensemble classifier, there is no obvious advantage, which may be related to the combination strategy of the ensemble classifier.

In Tables 2, Table 3, Table 4 and Fig.17, it can be seen from the Pos600+Neg2400 training set that both TN and FN are very high, indicating that under the unbalanced training of 2400 negative samples, the individual classifier has a certain tendency to recognize negative samples. In the unbalanced sample data set, F-Measure and G-mean are more persuasive to the performance of the classifier. Compared with individual classifier and unbalanced training set classifier, the ensemble classifier of LeNet-5 model in Table 3 has shown better performance, and the ensemble classifier of the improved LeNet-5 model has the best performance. Because the two-stage cascade learning is used in this work, the final result should be the product of the optimal result of each stage in the two stages. At the same time, it also shows that SVM, as an individual classifier, can show good performance for the usual binary classification problem, but it is not necessarily optimal in ensemble learning.

## VII. CONCLUSION

This paper presents an approach to automatically detect the locomotive speed sensor equipment using computer vision and machine learning. A cascade learning framework which includes two learning stages is proposed. Aiming at the imbalance of positive and negative samples of speed sensor, based on four individual classifiers, a combination strategy is designed to construct an ensemble of classifier for recognition. This approach has high accuracy and good application value, which can also be applied to other similar scenarios. Although the present work contributes to research on speed sensor detection at locomotive repair station, there are some limitations to be considered. We are aware that processing ability for low quality samples and model adaptability need to be improved. These are our research directions in the future.

## CONFLICT OF INTEREST

The authors declare that they have no conflict of interest.

## REFERENCES

[1] Y. Santur, M. Karaköse, and E. Akin, "Random forest based diagnosis approach for rail fault inspection in railways," in *Proc. Nat. Conf. Elect., Electron. Biomed. Eng. (ELECO)*, Dec. 2016, pp. 745–750.

[2] Ç. Aytekin, Y. Rezaeitabar, S. Dogru, and I. Ulusoy, "Railway fastener inspection by real-time machine vision," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 7, pp. 1101–1107, Jul. 2015. doi: 10.1109/TSMC.2014.2388435.

[3] X. Gibert, V. M. Patel, and R. Chellappa, "Robust fastener detection for autonomous visual railway track inspection," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Jan. 2015, pp. 694–701.

[4] Y. Li, H. Trinh, N. Haas, C. Otto, and S. Pankanti, "Rail component detection, optimization, and assessment for automatic rail track inspection," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 2, pp. 760–770, Apr. 2014.

[5] E. Resendiz, J. M. Hart, and N. Ahuja, "Automated visual inspection of railroad tracks," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 2, pp. 751–760, Jun. 2013. doi: 10.1109/TITS.2012.2236555.

[6] R. O. Duda and P. E. Hart, "Use of the Hough transform to detect lines and curves in pictures," *Commun. ACM*, vol. 15, no. 1, pp. 11–15, 1972.

[7] P. K. Rhee, E. Erdenee, S. D. Kyun, M. U. Ahmed, and S. Jin, "Active and semi-supervised learning for object detection with imperfect data," *Cogn. Syst. Res.*, vol. 45, pp. 109–123, Oct. 2017. doi: 10.1016/j.cogsys.2017.05.006.

[8] H. Wang, L. Dai, Y. Cai, X. Sun, and L. Chen, "Salient object detection based on multi-scale contrast," *Neural Netw.*, vol. 101, pp. 47–56, May 2018. doi: 10.1016/j.neunet.2018.02.005.

[9] Y. Liang, Z. Tang, M. Yan, and J. Liu, "Object detection based on deep learning for urine sediment examination," *Biocybernetics Biomed. Eng.*, vol. 38, no. 3, pp. 661–670, 2018. doi: 10.1016/j.bbe.2018.05.004.

[10] Y. Chen, D. Zhao, L. Lv, and Q. Zhang, "Multi-task learning for dangerous object detection in autonomous driving," *Inf. Sci.*, vol. 432, pp. 559–571, Mar. 2018. doi: 10.1016/j.ins.2017.08.035.

[11] T. Xiao and H. Daigle, "Preferential mineral-microfracture association in intact and deformed shales detected by machine learning object detection," *J. Natural Gas Sci. Eng.*, vol. 63, pp. 27–37, Mar. 2019. doi: 10.1016/j.jngse.2019.01.003.

[12] R. Huang, W. Feng, M. Fan, L. Wan, and J. Sun, "Multiscale blur detection by learning discriminative deep features," *Neurocomputing*, vol. 285, pp. 154–166, Apr. 2018. doi: 10.1016/j.neucom.2018.01.041.

[13] W. Lin, X. Ren, T. Zhou, X. Cheng, and M. Tong, "A novel robust algorithm for position and orientation detection based on cascaded deep neural network," *Neurocomputing*, vol. 308, pp. 138–146, Sep. 2018. doi: 10.1016/j.neucom.2018.04.061.

[14] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Columbus, OH, USA, Jun. 2014, pp. 580–587. doi: 10.1109/CVPR.2014.81.

[15] J. H. Kim, G. Batchuluun, and K. R. Park, "Pedestrian detection based on faster R-CNN in nighttime by fusing deep convolutional features of successive images," *Expert Syst. Appl.*, vol. 114, pp. 15–33, Dec. 2018. doi: 10.1016/j.eswa.2018.07.020.

[16] J.-G. Yu, C. Gao, and J. Tian, "Collaborative multicue fusion using the cross-diffusion process for salient object detection," *J. Opt. Soc. Amer. A*, vol. 33, no. 3, pp. 404–415, 2016. doi: 10.1364/JOSAA.33.000404.

[17] T. Guan, Y. Wang, L. Duan, and R. Ji, "On-device mobile landmark recognition using binarized descriptor with multifeature fusion," *ACM Trans. Intell. Syst. Technol.*, vol. 7, no.1, p. 12, 2015. doi: 10.1145/2795234.

[18] M. Tan, G. Pan, Y. Wang, Y. Zhang, and Z. Wu, "L1-norm latent svm for compact features in object detection," *Neurocomputing*, vol. 139, pp. 56–64, Sep. 2014. doi: 10.1016/j.neucom.2013.09.054.

[19] R. Bhuvaneswari and R. Subban, "Novel object detection and recognition system based on points of interest selection and SVM classification," *Cognit. Syst. Res.*, vol. 52, pp. 985–994, Dec. 2018. doi: 10.1016/j.cogsys.2018.09.022.

[20] A. Radman, N. Zainal, and S. A. Suandi, "Automated segmentation of iris images acquired in an unconstrained environment using HOG-SVM and GrowCut," *Digit. Signal Process.*, vol. 64, pp. 60–70, May 2017. doi: 10.1016/j.dsp.2017.02.003.

[21] D. Cheng, J. Wang, X. Wei, and Y. Gong, "Training mixture of weighted svm for object detection using em algorithm," *Neurocomputing*, vol. 149, pp. 473–482, Feb. 2015. doi: 10.1016/j.neucom.2014.08.048.

[22] A. Bria, C. Marrocco, M. Molinara, and F. Tortorella, "An effective learning strategy for cascaded object detection," *Inf. Sci.*, vols. 340–341, pp. 17–26, May 2016. doi: 10.1016/j.ins.2016.01.021.

[23] M. M. Abdelsamea, A. Pitiot, R. B. Grineviciute, J. Besusparis, A. Laurinavicius, and M. Ilyas, "A cascade-learning approach for automated segmentation of tumour epithelium in colorectal cancer," *Expert Syst. Appl.*, vol. 118, pp. 539–552, Mar. 2019. doi: 10.1016/j.eswa.2018.10.030.

[24] C. Gou, H. Zhang, K. Wang, F. Wang, and Q. Ji, "Cascade learning from adversarial synthetic images for accurate pupil detection," *Pattern Recognit.*, vol. 88, pp. 584–594, Apr. 2019. doi: 10.1016/j.patcog.2018.12.014.

[25] Y. Pang, J. Cao, and X. Li, "Cascade learning by optimally partitioning," *IEEE Trans. Cybern.*, vol. 47, no. 12, pp. 4148–4161, Dec. 2017. doi: 10.1109/TCYB.2016.2601438.

[26] J. Pang, W. Sun, J. S. J. Ren, C. Yang, and Q. Yan, "Cascade residual learning: A two-stage convolutional neural network for stereo matching," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Oct. 2017, pp. 878–886. doi: 10.1109/ICCVW.2017.108.

[27] P. Patil and S. Murala, "FgGAN: A cascaded unpaired learning for background estimation and foreground segmentation," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Jan. 2019, pp. 1770–1778. doi: 10.1109/WACV.2019.00193.

[28] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 2nd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2002.

[29] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2005, vol. 1, no. 1, pp. 886–893. doi: 10.1109/CVPR.2005.177.

[30] Y. Lei, X. Liu, J. Shi, C. Lei, and J. Wang, "Multiscale superpixel segmentation with deep features for change detection," *IEEE Access*, vol. 7, pp. 36600–36616, 2019. doi: 10.1109/ACCESS.2019.2902613.

[31] J. Jiao, Y. Zhang, H. Sun, X. Yang, X. Gao, W. Hong, K. Fu, and X. Sun, "A densely connected end-to-end neural network for multiscale and multiscene SAR ship detection," *IEEE Access*, vol. 6, pp. 20881–20892, 2018. doi: 10.1109/ACCESS.2018.2825376.

[32] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing, "Learning from class-imbalanced data: Review of methods and applications," *Expert Syst. Appl.*, vol. 73, pp. 220–239, May 2017. doi: 10.1016/j.eswa.2016.12.035.

[33] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009. doi: 10.1109/TKDE.2008.239.

[34] S. H. Ebenuwa, M. S. Sharif, M. Alazab, and A. Al-Nemrat, "Variance ranking attributes selection techniques for binary classification problem in imbalance data," *IEEE Access*, vol. 7, pp. 24649–24666, 2019. doi: 10.1109/ACCESS.2019.2899578.

[35] F. Charte, A. J. Rivera, M. J. del Jesus, and F. Herrera, "Dealing with difficult minority labels in imbalanced mutilabel data sets," *Neurocomputing*, vols. 326–327, pp. 39–53, Jan. 2019. doi: 10.1016/j.neucom.2016.08.158.
[36] W.-C. Lin, C.-F. Tsai, Y.-H. Hu, and J.-S. Jhang, "Clustering-based undersampling in class-imbalanced data," *Inf. Sci.*, vols. 409–410, pp. 17–26, Oct. 2017. doi: 10.1016/j.ins.2017.05.008.
[37] H. K. Lee and S. B. Kim, "An overlap-sensitive margin classifier for imbalanced and overlapping data," *Expert Syst. Appl.*, vol. 98, pp. 72–83, May 2018. doi: 10.1016/j.eswa.2018.01.008.
[38] J. Liu and E. Zio, "Integration of feature vector selection and support vector machine for classification of imbalanced data," *Appl. Soft Comput. J.*, vol. 75, pp. 702–711, Feb. 2019. doi: 10.1016/j.asoc.2018.11.045.
[39] X.-Y. Liu, J. Wu, and Z.-H. Zhou, "Exploratory undersampling for class-imbalance learning," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 39, no. 2, pp. 539–550, Apr. 2009. doi: 10.1109/TSMCB.2008.2007853.
[40] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998. doi: 10.1109/5.726791.
[41] J. Zhu and Z. Chen, "Real time face detection system using Adaboost and Haar-like features," in *Proc. 2nd Int. Conf. Inf. Sci. Control Eng.*, Apr. 2015, pp. 404–407. doi: 10.1109/ICISCE.2015.95.
[42] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 1, Dec. 2001, p. I. doi: 10.1109/CVPR.2001.990517.

**BO LI** (M'19) received the Ph.D. degree in space information science and technology from the Huazhong University of Science and Technology, China. He is currently employed as an Associate Professor with the College of Computer Science, South-Central University For Nationalities, China. His research interests includes digital Image processing and pattern recognition and machine learning.

**SISI ZHOU** graduate in computer technology from South-Central University for Nationalities. Her research interests include digital image processing, pattern recognition, and machine learning.

**LIFANG CHENG** is currently pursuing the degree from the South-Central University for Nationalities, in computer technology and studying digital image processing, pattern recognition, and machine learning.

**RONGBO ZHU** (M'10) received the Ph.D. degree in communication and information systems from Shanghai Jiao Tong University, China, in 2006. He is currently a Professor with the College of Computer Science, South-Central University for Nationalities, China. He has authored or coauthored over 70 papers in international journals and conferences in the areas of wireless networks and mobile computing. He serves as a guest editor for several journals. He is also an Associate Editor of IEEE Access and the *International Journal of Radio Frequency Identification Technology and Applications*.

**TAO HU** graduated from the Wuhan University of Technology, China. He is currently pursuing the degree in computer technology from the South-Central University for Nationalities. His research interests include machine learning and data mining.

**ASHIQ ANJUM** is currently a Professor of distributed systems at the University of Derby, U.K. His research interests include data intensive distributed systems, block chain, the Internet of Things, and high performance analytics platforms. He is currently investigating high performance distributed platforms to efficiently process video and genomics data. He has more than 70 international academic publications. He has been part of the EC funded projects in distributed systems and large-scale analytics such as Health-e-Child (IP, FP6), neuGrid (STREP, FP7), and TRANSFORM (IP, FP7). He has been a general chair, programmer chair, organizing chair, track chair, publicity chair, and workshop chair for more than 20 international conferences. He has been a member of the technical program committees for more than 40 international conferences. He is a member of British Computer Society, IEEE, ACE, and SIGHPC. He is the fellow of Higher Education Academy and the champion of European Grid Infrastructure.

**ZHENG HE** graduated in computer science and technology from the Hubei University of Technology. He is currently a Senior Software Engineer with Wuhan Japir Technology Company Ltd. He has extensive experiences in product development and algorithm. His research interest includes computer vision.

**YONGKAI ZOU** is currently pursuing the degree in computer technology with the South-Central University for Nationalities. His research interests include digital image processing and machine learning.