Research Article

Research on a Fast Human-Detection Algorithm for Unmanned Surveillance Area in Bulk Ports

Chao Mi,1 Xin He,2 Haiwei Liu,2 Youfang Huang,1 and Weijian Mi1

1 Container Supply Chain Tech. Engineering Research Center, Shanghai Maritime University, Shanghai 201306, China
2 Logistics Engineering College, Shanghai Maritime University, Shanghai 201306, China

Correspondence should be addressed to Haiwei Liu; liuhaiwei@michao.org

Received 12 March 2014; Revised 4 July 2014; Accepted 14 July 2014; Published 25 September 2014

Copyright © 2014 Chao Mi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the development of port automation, most operational fields utilizing heavy equipment have gradually become unmanned. It is therefore imperative to monitor these fields in an effective and real-time manner. In this paper, a fast human-detection algorithm is proposed based on image processing. To speed up the detection process, the optimized histograms of oriented gradients (HOG) algorithm that can avoid the large number of double calculations of the original HOG and ignore insignificant features is used to describe the contour of the human body in real time. Based on the HOG features, using a training sample set consisting of scene images of a bulk port, a support vector machine (SVM) classifier combined with the AdaBoost classifier is trained to detect human. Finally, the results of the human detection experiments on Tianjin Port show that the accuracy of the proposed optimized algorithm has roughly the same accuracy as a traditional algorithm, while the proposed algorithm only takes 1/7 the amount of time. The accuracy and computing time of the proposed fast human-detection algorithm were verified to meet the security requirements of unmanned port areas.

1. Introduction

1.1. Engineering Background. Bulk cargo wharfs have an important place in the logistics industry of China as the main loading and unloading means of bulk cargo from water-based transportation. Currently, the total number of bulk terminals in China has reached more than 1,000. Moreover, with the development of the nation’s economy and an increasing demand for coal and ore, the safety of staff members at port terminals has attracted greater attention. As shown in Figure 1, the safety rules of a bulk port prohibit people from entering the operation site. However, a large number of drivers and irrelevant personnel ignore these rules, which has resulted in serious accidents. The current method of enforcing these rules is to install cameras wherever needed; however, such traditional monitoring requires a large security staff. Under the premise of reasonable requirements and feasible technologies, it is necessary to search for a more effective monitoring and control system [1]. Intelligent video surveillance may be a valid selection meeting the requirements of unmanned port surveillance. Intelligent video surveillance [2] based on computer vision technology is aimed at building mapping relations using graphical and image descriptions. This type of surveillance is used to detect and analyze unusual circumstances in a video image through digital image processing [3] and to take control of the situation based on the results of an image analysis.

It is very difficult to apply intelligent human detection to bulk ports for the following reasons. First, the background is very complex. Since working machines are complicated and freight vehicles are continually moving about, dynamic interference is very significant when compared with a simple background environment of a container terminal. In addition, the image resolution is affected by a large amount of dust and spray during the loading and unloading operations. Second, the system should require a high-quality real-time performance and high accuracy. Because the scale of the machinery is quite large, the detection system must sound an alarm within a sufficient time frame to prevent someone breaking into the facility from causing a delay through their trespassing.
1.2. Related Works and Researches. Human detection is the key technology of intelligent video surveillance, especially for static images. Despite various difficulties, the development of human detection has made a number of achievements in recent years. Generally speaking, the detection process consists of the following three parts: namely, feature description, classification, and image processing. First, the characterization of the human form usually contains scale-invariant feature transforms (SIFTs) features, edge features [4], gait characteristics [5], and characteristics of the gradient direction. Second, there are different types of classifiers, with neural networks [6], support vector machines (SVM), AdaBoost, and Cascade [7] as graded by the AdaBoost classifier, being the most commonly used. Third, image processing consists of global scanning and subblock processing. The former is the processing and analysis of a whole image, whereas the latter divides a picture into multiple parts [8] and processes each part separately based on the inner link. Ngoc et al. proposed a new method for detecting a human region from still images using raw edges [9]. This method can indeed detect the human body but has problems in finding a suitable edge descriptor and does not have a training phase, which results in region estimation failures. As another method, Dalal and Triggs decided to use HOG features to describe the human body [10]. This method relies on the contrast between a human contour and the background, which was proposed as a useful feature for detecting humans standing in front of various kinds of backgrounds. However, the overall recognition process, combined with an SVM classifier, can be very slow. Zhang and Liu presented a novel method based on an Affine-SIFT detector to capture motion for human action recognition. However, this method requires a greater number of calculations compared with the use of HOG, which has no rotation or scaling invariance, because every feature in SIFT needs to be described using 128-dimensional vectors [11]. Shu et al. described the parts of the body as having a joint effect on the local direction characteristics [12]. The features are once extracted, the body parts are then detected [13], and the correlations among the parts are finally analyzed. However, the computational complexity of this method is large; Jia et al. derived a novel method of template matching [14, 15] based on a head-shoulder model [16], which only requires the correlation between the image edges and the template to be calculated. Nonetheless, because of the diversity and complexity of human postures, it is generally difficult to construct a sufficient number of templates to cover all of the possible postures.

1.3. Contributions of This Paper

(1) This paper describes an optimized HOG feature extraction algorithm. The traditional HOG needs to extract 3,780 feature vectors for each subimage when considering the impact of each possible factor, which results in low efficiency when processing an image. There is no doubt that Dalal’s HOG is one of the most successful feature description algorithms for the human body. It needs intensive scanning with a small rectangular window for different scale images, where each detection window can extract 3,780 HOG feature vectors as the classification base of the SVM classifier. Since a 720p image layer may have more than ten thousand windows that need to be computed for a traditional HOG algorithm, this leads to a slow computing process. An optimized algorithm that maintains the same accuracy as a traditional HOG feature extraction algorithm has been proposed. This optimized algorithm can avoid a large number of repeated operations of the original HOG and, by ignoring insignificant features, improves the efficiency. The experimental results show that the optimized algorithm can reduce the calculation time observably without a decrease in accuracy.

(2) This paper proposes a combined classifier for human recognition within an image. Whereas a traditional algorithm uses only an SVM as a classifier, in the proposed algorithm, the SVM classifier is combined with the AdaBoost classifier to shorten the detection time. Through training, the proposed algorithm can automatically search for support vectors that have a better classification capability. Thus, the classifier can maximize the spatial distance between two different classifications based on the principle of structural risk minimization. In addition, a better performance can be obtained using less sample training. However, because it takes all feature vectors into consideration, the detection process becomes time consuming. AdaBoost only selects certain HOG vectors as weak classifiers and trains the strong nonlinear classifiers, which avoids the unnecessary and useless calculations of HOG feature vectors. Although it can effectively reduce the number of computations and largely improve the detection speed, the accuracy is a little lower than for an SVM. The experimental results show that the combined classifier can perform better than a traditional classification method.

This paper proposes a fast human-detection method that combines an SVM with AdaBoost. To improve the recognition accuracy of the human-detection algorithm for an unmanned area in a bulk port, the training sample set is made up of images captured by camera views of a bulk port scene. AdaBoost can be treated as a filter for ignoring a large number of negative samples during the initial stage.
of detection. The remaining samples will be classified by the SVM, which is regarded as the final decision maker. A field experiment shows that the optimized HOG algorithm with the combined classifiers can improve the efficiency by more than sevenfold over a traditional algorithm.

2. Fast Human-Detection Algorithm

The proposed fast human-detection algorithm consists of three main steps: feature extraction, classification, and the fusion of the detection results.

Section 2.1 presents the optimized HOG algorithm used to perform the feature extraction. HOG is the most successful and popular feature for its contrast of the outline and background. Since the silhouettes of different people are similar, HOG is very effective for identifying humans from nonhumans. Specifically, to ensure the detection accuracy, the HOG feature vectors calculated can be reused for overlapping detection windows. Meanwhile, the block-based projection interpolation is improved to ignore some useless or unimportant features. In general, the optimized HOG algorithm can be increased by almost 100%.

Section 2.2 mainly describes the classification method. The SVM classifier cannot meet the real-time requirements of a bulk port, and AdaBoost’s classification accuracy is slightly worse than that of the SVM. Therefore, this section describes a novel classification method in which the SVM classifier is combined with AdaBoost to shorten the classification time. In addition, the vertical gradient intensity of the first and last columns, such as the origin and end points, are similar, HOG is very effective for identifying humans and background. Since the silhouettes of different people are similar, HOG is very effective for identifying humans from nonhumans. Specifically, to ensure the detection accuracy, the HOG feature vectors calculated can be reused for overlapping detection windows. Meanwhile, the block-based projection interpolation is improved to ignore some useless or unimportant features. In general, the optimized HOG algorithm can be increased by almost 100%.

Section 2.3 describes the method for fusing the detection results. Because the detection window scans the image to be detected under multiple scales, there will be a large number of marked results overlapping around only one human body. A mean shift method has been proposed for fusing the results into a final description of a detected human.

2.1. Optimized HOG Algorithm. Through research and comparison, it was found that sparse local features [17] could not be used to create a complete description of the human body. Considering the various clothes colors and illumination intensity of a complex environment, HOG has become an outstanding tool for outlining the contour and synthesizing the characteristics of a human body into a very high robust feature vector space. However, there is a disadvantage to the traditional method provided by Dalal and Triggs [10], which is that the object’s description is included in the representation of the whole image, leading to a large number of repeated feature calculations. This is the primary reason why the traditional HOG algorithm takes a very long time to process. Therefore, this paper describes how to reduce the calculation time of the traditional HOG algorithm.

Before describing the proposed fast HOG feature computation method, it is necessary to provide an introduction to the extraction process.

The process of extracting HOG features can be summarized in the following five steps: color space standardization, gradient calculations, generating gradient statistics on the space and direction, contrast standardization, and feature vector generation. The following paragraphs provide the detailed principles of the algorithm for each step and the core aspect of the entire algorithm. Steps 1, 2, and 4 are roughly the same as in Dalal’s method except for determining the corresponding parameters of the ports and the processing methods based on the actual situation. Steps 3 and 5 are the core aspects of the optimized HOG algorithm.

(1) Considering the strong illumination used for practical applications at a port, it is necessary to preprocess the images to improve the detector’s robustness to light and shade. As a consequence, the image noise is proportional to the square root of the light intensity; (1), which is the square root compression of each color channel (i.e., a gamma correction operation), is used to make the maximum balance limit and achieve a better balance:

\[
\begin{align*}
\bar{r}_{x,y} &= \sqrt{r_{x,y}}, \\
\bar{g}_{x,y} &= \sqrt{g_{x,y}}, \\
\bar{b}_{x,y} &= \sqrt{b_{x,y}},
\end{align*}
\]

(1)

where \(r_{x,y}, g_{x,y}, \) and \(b_{x,y}\) are the original pixel values of the red, green, and blue channels, respectively; \(\bar{r}_{x,y}, \bar{g}_{x,y}, \) and \(\bar{b}_{x,y}\) are the gamma-corrected pixel values; and \(f_{x,y}\) is the color space of a pixel in a gamma corrected image.

(2) The effective extraction of human body contours is the core aspect of HOG. The sharpness of the above-mentioned contour is determined through the gradient of the image intensity [18] or illumination. The gradient is a vector indicating where the biggest change in gray scale is located, which determines the final feature vector space. Here, \(G_x^x\) is the gradient of the \(x\) direction, whereas \(G_y^y\) is that of the \(y\) direction at point \((x, y)\). Herein, a one-dimensional gradient template \([-1, 0, 1]\) is used to calculate the gradient magnitude \(||f_{x,y}||_2\) and gradient direction \(\theta_{x,y}\), because it works best with a minimum number of calculations as compared to other templates, such as a \(2 \times 2\) diagonal matrix or \(3 \times 3\) Sobel operator; the equations for \(||f_{x,y}||_2\) and \(G_x^x\) are shown in (2) and (3) [19]. Assuming that the size of an image is \(m \times n\), if \(1 < x < m\) and \(1 < y < n\), then

\[
||f_{x,y}||_2 = \sqrt{G_x^x + G_y^y},
\]

(2)

\[
G_x^x = \text{Max Element} (f_{x+1,y} - f_{x-1,y}) ,
\]

(3)

\[
G_y^y = \text{Max Element} (f_{x,y+1} - f_{x,y-1}).
\]

The horizontal gradient intensity of the first and last column points can be expressed through the first two equations of (3). In addition, the vertical gradient intensity of the first
Table 1: Gradient amplitude of the red block shown in Figure 2.

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>6.40</td>
<td>8.00</td>
<td>8.25</td>
<td>4.27</td>
<td>7.28</td>
</tr>
<tr>
<td>10.05</td>
<td>7.50</td>
<td>2.83</td>
<td>4.12</td>
<td>4.24</td>
</tr>
<tr>
<td>1.50</td>
<td>1.00</td>
<td>2.50</td>
<td>3.35</td>
<td>1.41</td>
</tr>
<tr>
<td>8.06</td>
<td>11.60</td>
<td>11.54</td>
<td>4.03</td>
<td>2.24</td>
</tr>
<tr>
<td>0.25</td>
<td>4.33</td>
<td>2.12</td>
<td>2.00</td>
<td>0.71</td>
</tr>
<tr>
<td>1.41</td>
<td>3.00</td>
<td>3.16</td>
<td>4.47</td>
<td>5.59</td>
</tr>
<tr>
<td>7.02</td>
<td>6.58</td>
<td>9.22</td>
<td>10.77</td>
<td>6.02</td>
</tr>
<tr>
<td>5.00</td>
<td>3.54</td>
<td>4.12</td>
<td>5.59</td>
<td>5.97</td>
</tr>
<tr>
<td>6.52</td>
<td>7.52</td>
<td>7.52</td>
<td>9.50</td>
<td>12.01</td>
</tr>
<tr>
<td>12.51</td>
<td>13.51</td>
<td>5.59</td>
<td>5.00</td>
<td>6.02</td>
</tr>
<tr>
<td>4.47</td>
<td>7.43</td>
<td>7.16</td>
<td>4.72</td>
<td>2.50</td>
</tr>
<tr>
<td>1.50</td>
<td>1.50</td>
<td>1.26</td>
<td>2.55</td>
<td>2.69</td>
</tr>
<tr>
<td>1.12</td>
<td>2.50</td>
<td>4.24</td>
<td>4.53</td>
<td>4.50</td>
</tr>
<tr>
<td>4.03</td>
<td>4.03</td>
<td>2.83</td>
<td>2.83</td>
<td>2.24</td>
</tr>
<tr>
<td>1.00</td>
<td>1.41</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>7.43</td>
<td>12.42</td>
<td>14.16</td>
<td>7.16</td>
<td>2.69</td>
</tr>
<tr>
<td>2.50</td>
<td>1.58</td>
<td>0.00</td>
<td>2.92</td>
<td>3.54</td>
</tr>
<tr>
<td>3.00</td>
<td>3.16</td>
<td>2.50</td>
<td>11.2</td>
<td>1.12</td>
</tr>
<tr>
<td>14.21</td>
<td>18.03</td>
<td>14.15</td>
<td>8.86</td>
<td>7.28</td>
</tr>
<tr>
<td>9.01</td>
<td>11.00</td>
<td>9.62</td>
<td>9.50</td>
<td>8.02</td>
</tr>
<tr>
<td>5.70</td>
<td>4.24</td>
<td>2.24</td>
<td>0.50</td>
<td>1.58</td>
</tr>
<tr>
<td>2.69</td>
<td>12.17</td>
<td>12.59</td>
<td>7.38</td>
<td>2.83</td>
</tr>
<tr>
<td>4.53</td>
<td>4.30</td>
<td>5.52</td>
<td>8.63</td>
<td>6.08</td>
</tr>
<tr>
<td>7.02</td>
<td>7.02</td>
<td>5.50</td>
<td>3.04</td>
<td>2.00</td>
</tr>
<tr>
<td>1.58</td>
<td>2.69</td>
<td>10.31</td>
<td>16.12</td>
<td>16.99</td>
</tr>
<tr>
<td>9.71</td>
<td>4.03</td>
<td>5.52</td>
<td>7.38</td>
<td>6.50</td>
</tr>
<tr>
<td>4.92</td>
<td>2.50</td>
<td>0.50</td>
<td>1.12</td>
<td>1.50</td>
</tr>
<tr>
<td>1.58</td>
<td>2.06</td>
<td>4.72</td>
<td>11.41</td>
<td>14.84</td>
</tr>
<tr>
<td>8.51</td>
<td>6.18</td>
<td>8.02</td>
<td>5.52</td>
<td>2.50</td>
</tr>
<tr>
<td>4.03</td>
<td>4.03</td>
<td>4.72</td>
<td>4.72</td>
<td>6.32</td>
</tr>
<tr>
<td>7.07</td>
<td>7.57</td>
<td>7.07</td>
<td>7.57</td>
<td>7.07</td>
</tr>
<tr>
<td>18.03</td>
<td>22.90</td>
<td>21.52</td>
<td>18.34</td>
<td>17.18</td>
</tr>
<tr>
<td>18.03</td>
<td>17.26</td>
<td>16.62</td>
<td>16.00</td>
<td>15.01</td>
</tr>
<tr>
<td>13.15</td>
<td>13.73</td>
<td>13.09</td>
<td>13.04</td>
<td>13.51</td>
</tr>
<tr>
<td>13.51</td>
<td>13.51</td>
<td>13.51</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11.70</td>
<td>13.46</td>
<td>12.82</td>
<td>8.20</td>
<td>7.52</td>
</tr>
<tr>
<td>4.53</td>
<td>8.14</td>
<td>12.01</td>
<td>10.01</td>
<td>9.62</td>
</tr>
<tr>
<td>9.51</td>
<td>8.73</td>
<td>7.07</td>
<td>7.57</td>
<td>9.50</td>
</tr>
<tr>
<td>11.18</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Gradient angle of the red block shown in Figure 2.

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>38.66</td>
<td>90.00</td>
<td>104.04</td>
<td>159.44</td>
<td>15.95</td>
</tr>
<tr>
<td>174.29</td>
<td>143.13</td>
<td>135.00</td>
<td>165.96</td>
<td>135.00</td>
</tr>
<tr>
<td>90.00</td>
<td>90.00</td>
<td>90.00</td>
<td>90.00</td>
<td>26.57</td>
</tr>
<tr>
<td>135.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>60.26</td>
<td>82.57</td>
<td>85.03</td>
<td>82.87</td>
<td>153.43</td>
</tr>
<tr>
<td>173.66</td>
<td>135.00</td>
<td>165.96</td>
<td>135.00</td>
<td>0.00</td>
</tr>
<tr>
<td>6.47</td>
<td>11.41</td>
<td>14.84</td>
<td>8.51</td>
<td>6.18</td>
</tr>
<tr>
<td>8.02</td>
<td>5.52</td>
<td>2.50</td>
<td>4.03</td>
<td>4.03</td>
</tr>
<tr>
<td>4.03</td>
<td>4.72</td>
<td>6.32</td>
<td>7.07</td>
<td>7.57</td>
</tr>
<tr>
<td>7.07</td>
<td>18.03</td>
<td>22.90</td>
<td>21.52</td>
<td>18.34</td>
</tr>
<tr>
<td>17.18</td>
<td>16.62</td>
<td>16.00</td>
<td>15.01</td>
<td>13.15</td>
</tr>
<tr>
<td>13.73</td>
<td>13.09</td>
<td>13.04</td>
<td>13.51</td>
<td>13.51</td>
</tr>
<tr>
<td>13.51</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11.70</td>
<td>13.46</td>
<td>12.82</td>
<td>8.20</td>
<td>7.52</td>
</tr>
<tr>
<td>4.53</td>
<td>8.14</td>
<td>12.01</td>
<td>10.01</td>
<td>9.62</td>
</tr>
<tr>
<td>9.51</td>
<td>8.73</td>
<td>7.07</td>
<td>7.57</td>
<td>9.50</td>
</tr>
<tr>
<td>11.18</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

and last line points can be expressed through the final two equations of (3):

\[
G_{x,y}^x = f_{x,y} - f_{x-1,y}, \\
G_{m,y}^x = f_{m,y} - f_{m-1,y}, \\
G_{x,1}^y = f_{x,1} - f_{x,0}, \\
G_{x,n}^y = f_{x,n} - f_{x,n-1}, \\
\theta_{x,y} = \arctan \frac{G_{x,y}^x}{G_{x,y}^y}, \\
\theta_{x,y} = \begin{cases} 
\theta_{x,y} + \pi & \text{if } \theta_{x,y} < 0 \\
\theta_{x,y} & \text{if } \theta_{x,y} \geq 0.
\end{cases}
\]

In addition, if an image has three color channels, then a three-channel gradient value is calculated and the maximum gradient value of all channels will be taken as the final result. Tables 1 and 2 show the gradient amplitude and angle of the red block shown in Figure 2.

(3) The next step is to process the statistics of HOG. This is the core of the optimized algorithm and reduces the number of calculations by 44%. According to the direction and magnitude of the gradient, the gradient magnitude of each pixel is accumulated into orientation bins over cells with weight. The statistical area used in this paper is a square block, which is divided into \(2 \times 2\) cells. Each cell is 8 pixels \(\times 8\) pixels in size. This method avoids filtering out useful information and ensures the best effect of the following contrast standardization. Until now, the statistical method used has been a linear interpolation [20], where each of the gradient angles can be any value between 0 and 180 degrees,
and the entire region is discretized into nine bins, as shown in Figure 3; that is to say, each bin represents 20 degrees. Thus, for bins adjacent to the gradient direction of the pixel, the closer one obtains greater weight when the gradient of each pixel is allocated to these nine discrete bins. However, based on the fact that four cells of a block are all adjacent to each other and avoiding a mutation of the feature vectors, a cell projection needs to be weighted to the adjacent cell as well. Therefore, the following trilinear interpolation is used.

The so-called trilinear interpolation refers to an interpolation in a three-parameter space \([x, y, \theta]^T\), that is, the \(x\) direction, \(y\) direction, and gradient angle, as shown in Figure 4. When pixel point \([x, y]^T\) adopts the gradient amplitude as the voting weight according to the distance to the center of each cell, the pixel gradient direction also needs an interpolation in the adjacent interval at the same time.

The 36 statistics, \(V_{\text{block}} = [v_1, v_2, \ldots, v_{36}]^T\), in a block used to compute the final HOG features can then be calculated one by one. The detailed computation process of a traditional operation for these 36 statistics is shown in Algorithm 1.

In a traditional interpolation method, pixels in each cell have the same influence on the adjacent cells in the current block; that is to say, each pixel gradient direction must be projected onto four cells while ignoring the distance weighting factors. For one block, if the cells are considered as a whole, the total number of interpolations is \(16 = 4 \text{ (violet)} + 8 \text{ (red)} + 4 \text{ (black)}\).

Considering the phenomenon in which different positions of a cell will not have the same influence on an adjacent pixel cell, each cell is divided into four subareas to reduce the number of computations, as shown in Figure 4. In traditional HOG, the pixel area away from the other cells
INPUT: $\theta_{x,y}, \|v_{f_{x,y}}\|_2$
OUTPUT: the matrix $V_{\text{block}} = [v_1, v_2, \ldots, v_{36}]^T$

Begin Algorithm
For $x = 1$ to 16
  for $y = 1$ to 16
    $\theta_1 = (\theta_{x,y} - 10)\%20$; $\theta_2 = (\theta_{x,y} - 10)\%20 + 1$;
    if $(\theta_{x,y} - 10)\%20 = 0$
      $d_x = 8; d_y = 8; d_\theta = 20$;
      $h(1,1,\theta_1) = h(1,1,\theta_1) + \|v_{f_{x,y}}\|_2 (1 - \frac{x - 4}{d_x}) (1 - \frac{y - 4}{d_y}) (1 - \frac{\theta_{x,y} - \theta_1}{d_\theta})$
      $h(2,1,\theta_1) = h(2,1,\theta_1) + \|v_{f_{x,y}}\|_2 (\frac{x - 8}{d_x}) (1 - \frac{y - 4}{d_y}) (1 - \frac{\theta_{x,y} - \theta_1}{d_\theta})$
      $h(1,2,\theta_1) = h(1,2,\theta_1) + \|v_{f_{x,y}}\|_2 (1 - \frac{x - 4}{d_x}) (\frac{y - 8}{d_y}) (1 - \frac{\theta_{x,y} - \theta_1}{d_\theta})$
      $h(2,2,\theta_1) = h(2,2,\theta_1) + \|v_{f_{x,y}}\|_2 (\frac{x - 8}{d_x}) (\frac{y - 8}{d_y}) (1 - \frac{\theta_{x,y} - \theta_1}{d_\theta})$
    else
      $h(1,1,\theta_1) = h(1,1,\theta_1) + \|v_{f_{x,y}}\|_2 (1 - \frac{x - 4}{d_x}) (1 - \frac{y - 4}{d_y}) (1 - \frac{\theta_{x,y} - \theta_2}{d_\theta})$
      $h(2,1,\theta_1) = h(2,1,\theta_1) + \|v_{f_{x,y}}\|_2 (\frac{x - 8}{d_x}) (1 - \frac{y - 4}{d_y}) (1 - \frac{\theta_{x,y} - \theta_2}{d_\theta})$
      $h(1,2,\theta_1) = h(1,2,\theta_1) + \|v_{f_{x,y}}\|_2 (1 - \frac{x - 4}{d_x}) (\frac{y - 8}{d_y}) (1 - \frac{\theta_{x,y} - \theta_2}{d_\theta})$
      $h(2,2,\theta_1) = h(2,2,\theta_1) + \|v_{f_{x,y}}\|_2 (\frac{x - 8}{d_x}) (\frac{y - 8}{d_y}) (1 - \frac{\theta_{x,y} - \theta_2}{d_\theta})$
    end if
  end for
end for

End Algorithm

Algorithm 1

is also interpolated. Actually, the influence of this area on the other three cells is limited, and thus this interpolation is redundant. By improving the algorithm, the pixels in the middle black sub-regions will have influence on all four cells surrounding them; the pixels in four corners lined up by red will have influence on the cells they belong to; and the pixels in other areas lined up by red will have influence on two cells adjacent to them. After this simplification, each cell of the gradient calculation is $9 = 1$ (violet) + 4 (red) + 4 (black), and therefore the optimized method reduces the calculated quantity of HOG features by $44\% = (16 - 9)/16$ compared with a traditional method. In addition, the time required for the entire feature extraction is reduced by nearly half. Figure 5 shows an example of a projection. The detailed computation process of our optimized operation for these 36 statistics is shown in Algorithm 2.
INPUT: \(\theta_{x,y}, \|\nabla f_{x,y}\|_2\)

OUTPUT: the matrix \(V_{\text{block}} = [v_1, v_2, \ldots, v_{36}]^T\)

Begin Algorithm:

define function \(vote(\theta_{x,y}, X, Y)\);

for \(x = 1\) to 16
  for \(y = 1\) to 16
    if \(x \leq 4\) \& \(y \leq 4\)
      vote(\(\theta_{x,y}, 1, 1\));
    else if \(12 < x \leq 16\) \& \(y \leq 4\)
      vote(\(\theta_{x,y}, 1, 2\));
    else if \(x \leq 4\) \& \(12 < y \leq 16\)
      vote(\(\theta_{x,y}, 2, 1\));
    else if \(12 < x \leq 16\) \& \(12 < y \leq 16\)
      vote(\(\theta_{x,y}, 2, 2\));
    elseif \(4 < x \leq 12\) \& \(y \leq 4\)
      vote(\(\theta_{x,y}, 1, 1\));
      vote(\(\theta_{x,y}, 1, 2\));
    elseif \(4 < y \leq 12\) \& \(x \leq 4\)
      vote(\(\theta_{x,y}, 1, 1\));
      vote(\(\theta_{x,y}, 2, 1\));
    elseif \(4 < y \leq 12\) \& \(12 < x \leq 16\)
      vote(\(\theta_{x,y}, 1, 2\));
      vote(\(\theta_{x,y}, 2, 2\));
    elseif \(4 < x \leq 12\) \& \(4 < y \leq 12\)
      vote(\(\theta_{x,y}, 1, 1\));
      vote(\(\theta_{x,y}, 1, 2\));
      vote(\(\theta_{x,y}, 2, 1\));
      vote(\(\theta_{x,y}, 2, 2\));
  end if
end for
end for

\(i = 1;\)

for \(y = 1\) to 2
  for \(x = 1\) to 2
    for \(\theta = 1\) to 9
      \(V_i = h(x, y, \theta);\)
      \(i = i + 1;\)
    end for
  end for
end for

End Algorithm

function \(vote(\theta_{x,y}, X, Y)\)

\(\theta_1 = (\theta_{x,y} - 10)\times20; \theta_2 = (\theta_{x,y} - 10)\times20 + 1;\)

if \((\theta_{x,y} - 10)\%20 = 0\) then
  \(d_x = 8; d_y = 8; d_\theta = 20;\)
if \(X = 1\) \& \(Y = 1\) then
  \(h(1, 1, \theta_1) = h(1, 1, \theta_1) + \|\nabla f_{x,y}\|_2 \left(1 - \frac{x - 4}{d_x}\right) \left(1 - \frac{y - 4}{d_y}\right) \left(1 - \frac{\theta_{x,y} - \theta_1}{d_\theta}\right);\)
else if \(X = 2\) \& \(Y = 1\)
  \(h(2, 1, \theta_1) = h(2, 1, \theta_1) + \|\nabla f_{x,y}\|_2 \left(1 - \frac{x - 8}{d_x}\right) \left(1 - \frac{y - 4}{d_y}\right) \left(1 - \frac{\theta_{x,y} - \theta_1}{d_\theta}\right);\)
else if \(X = 1\) \& \(Y = 2\)
  \(h(1, 2, \theta_1) = h(1, 2, \theta_1) + \|\nabla f_{x,y}\|_2 \left(1 - \frac{x - 4}{d_x}\right) \left(1 - \frac{y - 8}{d_y}\right) \left(1 - \frac{\theta_{x,y} - \theta_1}{d_\theta}\right);\)
else if \(X = 2\) \& \(Y = 2\)
  \(h(2, 2, \theta_1) = h(2, 2, \theta_1) + \|\nabla f_{x,y}\|_2 \left(1 - \frac{x - 8}{d_x}\right) \left(1 - \frac{y - 8}{d_y}\right) \left(1 - \frac{\theta_{x,y} - \theta_1}{d_\theta}\right);\)

Algorithm 2: Continued.
\[
\text{Algorithm 2}
\]

\[
M_i = \frac{\nu_i}{\sqrt{\|V_{\text{Block}}\|_2^2 + \sigma^2}},
\]

where \(\sigma\) is a small constant to prevent the denominator from being 0 and \(M_i\) is the HOG vector. There are 36 HOG vectors in a block.

(5) The HOG feature vector space is then generated. As a large number of blocks are repeated in adjacent detection windows, calculating the HOG feature vector of each scan window will lead to many unnecessary calculations. Hence, the previous calculation mode needs to be changed. First, all block feature vectors of the current scaling image should be calculated and stored, and a \(64 \times 128\) (according to the proportions of the human body) \([21]\) detection window is then used to scan the entire image from top to bottom and from left to right. In each detection window, all HOG feature vectors of 105 blocks are obtained in accordance with the correct indexes. Thus, all \(105 \times 36 = 3,780\) dimensional feature vectors of the detection window are generated. The feature vectors of a block are shown in Figure 6. Renumbering all values of \(M_i\) from 1 to 3,780 for a detection window, the

(4) Block normalization is used primarily to strengthen the correlation of different regions, such that the feature vector space will be robust to background illumination, edge mutations, and shadows. Each block’s HOG features are calculated using an L2-Norm normalized function, which can be defined through the following equation:

\[
M_i = \frac{\nu_i}{\sqrt{\|V_{\text{Block}}\|_2^2 + \sigma^2}},
\]

where \(\sigma\) is a small constant to prevent the denominator from being 0 and \(M_i\) is the HOG vector. There are 36 HOG vectors in a block.

(5) The HOG feature vector space is then generated. As a large number of blocks are repeated in adjacent detection windows, calculating the HOG feature vector of each scan window will lead to many unnecessary calculations. Hence, the previous calculation mode needs to be changed. First, all block feature vectors of the current scaling image should be calculated and stored, and a \(64 \times 128\) (according to the proportions of the human body) \([21]\) detection window is then used to scan the entire image from top to bottom and from left to right. In each detection window, all HOG feature vectors of 105 blocks are obtained in accordance with the correct indexes. Thus, all \(105 \times 36 = 3,780\) dimensional feature vectors of the detection window are generated. The feature vectors of a block are shown in Figure 6. Renumbering all values of \(M_i\) from 1 to 3,780 for a detection window, the
2.2. Combined Classifier. After calculating the HOG feature vector space of positive and negative samples, the next step is to use these samples to train the classifier and obtain the best classification parameters.

AdaBoost is short for Adaptive Boosting, which is a representative algorithm of the Boosting family. As this method adaptively adjusts the assumed error rate according to the results of the weak learning feedback, it does not need to know the lower limit of the error rate in advance. In addition, it does not need any prior knowledge regarding the performance of the weak classifiers and may have the same efficiency as the Boosting algorithm. It has therefore been widely used since its initial proposal.

The SVM was first proposed by Cortes and Vapnik [22]. It can automatically search for support vectors that have a better classification capability. Thus, the classifier can maximize the spatial distance between two different classifications based on the principle of structural risk minimization. This has obvious advantages when solving the problems of a small sample for nonlinear and high dimensional pattern recognition.

In the proposed fast human-detection system, a linear SVM [23] and AdaBoost are combined for a better performance. Because of the high-dimensional feature vectors, AdaBoost [24] selects certain vector components as a weak classifier and trains the strong nonlinear classifier to avoid unnecessary or useless feature vectors. Although it can effectively reduce the number of computations and greatly improve the detection speed, the accuracy is still somewhat unsatisfactory. Meanwhile, an SVM has high classification accuracy, although its time cost is a little longer than that of AdaBoost. Therefore, the combined classifier integrates the advantages of both classifiers and achieves a higher classification accuracy and efficiency. The combined classifier first uses the AdaBoost cascade classifier to filter out most of the negative examples. The remaining image features of the detection windows will then be classified by the SVM classifier, which can greatly improve the classification speed.

2.2.1. Training and Selection for AdaBoost Weak Classifiers. A matrix of all sample features should first be created. In Figure 7, the abscissa is the sample number, and the ordinate is the dimension number of the feature vectors. In addition, \( M \) indicates the value of the feature vectors. For each feature, as shown in the green rectangle, all values of the training samples should be calculated and sorted from smallest to largest. By rescanning the sorted samples, an optimal threshold can be determined for the operating feature. A weak classifier is created using this feature. The optimal threshold can be determined through the following steps.

First, the following four values should be calculated for any sample for this feature vector: \( R_i \), the total weight of the positive samples before sample \( n \); \( W_i \), the total weight of negative samples before sample \( n \); \( R_X \), the total weight of all positive samples; and \( W_X \), the total weight of all negative samples. In this paper, \( X \) is \( 11,536 = 2,416 + 9,120 \), where 2,416 is the number of positive samples and 9,120 is the number of negative samples. These samples were all collected at Coal Terminal of Tianjin Port. \( M_{(i,j)} \) is the \( n \)th HOG vector in sample \( j \).

Thus, the weak classifier will classify samples before \( n \) as human (or nonhuman) and classify the samples after \( n \) (including) as nonhuman (or human). The classification error can be determined using

\[
\varepsilon_n = \min \left[ (R_n + W_X - W_n), (W_n + R_X - R_n) \right]. \tag{7}
\]

For such a feature vector, the minimum classification error can be found by traversing all samples.

Then, the best weak classifier of each feature is calculated by scanning all feature vectors in Figure 7 using the above method. In addition, the threshold (determined by (8)) of feature \( i \) that minimizes the classification errors is selected as the optimal threshold. The corresponding weak classifier \( h_i \) is also the best:

\[
\text{Threshold } = \frac{[M_{(i,n)} - M_{(i,n-1)}]}{2}. \tag{8}
\]

Table 4 shows example training results of different weak classifiers. The optimal threshold means the samples are separated at the minimum error rate. The characteristic dimension with the minimum error rate is chosen and a tree node is created. Each branch of the node indicates the corresponding classification results. The classification can be regarded as complete if it stops at a leaf node while traversing the tree. The classification weights are updated for all iterations.

A training sample set is shown in Table 3. All samples were taken from a real unmanned area of a port.
The table shows the training results of a weak classifier.

<table>
<thead>
<tr>
<th>Weak classifier</th>
<th>Threshold</th>
<th>Feature no.</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.173217</td>
<td>1708</td>
<td>0.469284</td>
</tr>
<tr>
<td>2</td>
<td>0.217191</td>
<td>2275</td>
<td>0.863482</td>
</tr>
<tr>
<td>3 and 4</td>
<td>0.266998</td>
<td>2050</td>
<td>-0.548016</td>
</tr>
</tbody>
</table>

Figure 8: Training process of CART decision tree.

2.2.2. Training of Decision Tree. To improve the classification accuracy during the weak classifier training, we use the CART decision tree as the training unit. A decision tree has a flowchart-like structure in which an internal node represents the test of an attribute, each branch of the node indicates the corresponding classification results, and each leaf node represents the classification label, that is, +1 for a human or −1 for a nonhuman. The path from a root to a leaf represents the classification rules. This therefore helps in identifying the strategy most likely to reach the goal of the decision analysis.

Figure 8 shows the process of training the decision tree. The entire algorithm is as follows.

1. Create root node 1, and then generate nodes 2 and 3 after the weak classifier is trained for the first time using all of the samples.

2. For nodes 2 and 3, weak classifiers trained, respectively, under the environment of their corresponding samples are divided by node 1. In addition, node 2 with a larger error decreasing rate will be the leaf node.

3. Repeat step 2 until the error rate is equal to 0, or until the specified number of iterations runs out. Herein, this number is set to 4. As a result, four weak classifiers are generated, that is, 1-3, 1-2-5, 1-2-4-6, and 1-2-4-7.

2.2.3. Strong Classifier Training. The role of a strong classifier is to make all weak classifiers vote and then calculate the weighted summation of the voting results in accordance with the error rate of the weak classifiers. The final classification result can be obtained by comparing the summation with 0. Figure 9 describes the training algorithm of a strong classifier.

1. The weights are normalized, where \( n \) is the sample number:

\[
w_i = \frac{w_i}{\sum_{n=1}^{X} w_n}
\]

2. Weights \( \alpha_t \) of \( h(x) \) are calculated, where \( h(x) \) is a weak classifier:

\[
\alpha_t = 1 - 2\epsilon_t = \frac{P_t - N_t}{P_t + N_t}
\]

Here, \( P_t \) is the probability of a tree node's positive samples, and \( N_t \) is the probability of a tree node's negative samples. Please note that (10) is the linearization of the traditional weight function, \( \alpha_t = (1/2) \ln((1 - \epsilon_t)/\epsilon_t) \), which can accelerate the computer calculation process and avoid the weight from becoming an infinity number when \( \epsilon_t \) is too small.

3. According to the optimal weak classifier, the weights of the samples are adjusted for the next iteration. In this paper, training samples are separated into 4 nodes after the completion of classification of a decision tree, and the 4 nodes are symbolized as node 3, node 5, node 6, and node 7. In order
Begin

Sample matrix

Maximum iterations (T)

Normalize weights

Train decision tree (CART)

Calculate h(x)’s weights

Reweighting

Generate strong classifier

End

Figure 9: Flowchart of strong classifier training.

to perform weight updating for all samples, sum formula $e^{-\sum_{t=1}^{T} \alpha_{t} y_{i} h_{t}(x_{i})}$ is used to combine samples of the 4 nodes. The reweighting equation is shown below:

$$ w_{i} = \frac{1}{e^{\sum_{t=1}^{T} \alpha_{t} y_{i} h_{t}(x_{i})}} \quad \text{where } i = 1, \ldots, X. \quad (11) $$

(4) Strong classifier $H(x)$ is generated:

$$ H(x) = \text{sign} \left( \sum_{t=1}^{T} \alpha_{t} h_{t}(x) \right). \quad (12) $$

2.2.4. Cascade Classifier. A cascade classifier is composed of multiple strong classifiers. The numbers of weak classifiers required by different levels of strong classifiers differ. Each level is more complicated than the former level. During the initial stage of the detection, the classifier abandons a large number of complex negative samples, as shown in Figure 10, which improves the detection speed.

From Figure 11, we can see that 80% of the negative samples have been rejected for the first four levels. As the cascade level increases, the rejection rate rises to a flat level. The final classification result may be equal to 100% if the level is sufficient. For this research, the first five stages, that is, about 100 weak classifiers, are used as a filter. Using the cascade classifier, most classifiable negative samples are quickly rejected. Thus, only a few samples will be entered into a more accurate SVM classifier for further classification. Compared to a traditional method, in which all samples are input into an inefficient and high-precision SVM classifier, this combined classifier takes into account the efficiency and accuracy.

2.2.5. Training of SVM Classifier. For human-detection system, the SVM classifier is aimed at finding a hyperplane among all 3,780 HOG feature vectors of the detection windows. This hyperplane is used not only to correctly distinguish between people and objects but also to maximize the classification interval of the samples. The support vector machine classification is shown in Figure 12. The yellow area is the hyperplane.

The hyperplane can be expressed through

$$ w^{T} x + b = 0. \quad (13) $$

If the samples and relevant categories are given, $w$ is determined through

$$ w = \alpha_{1} x_{1} y_{1} + \alpha_{2} x_{2} y_{2} + \cdots + \alpha_{n} x_{n} y_{n}. \quad (14) $$

Because the processed data are nonlinear, it is necessary to use a kernel function to accept two low-dimensional space vectors and calculate the inner product value of a high-dimensional space through the transformation. Thus, it can convert data into a higher dimensional space with...
linear separability. This problem can be transformed into the following equation, where $C$ is the penalty factor:

$$
\min \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{j} \xi_i
$$

(15)

$$y_i \times (w^T x + b) - 1 \geq 0 \quad (i = 1, 2, \ldots, l).
$$

The main process of the training SVM classifier is listed as follows:

(1) A general human database contains thousands of positive and negative sample images including different postures, clothing, backgrounds, and partial covering. However, the background environment of a port is so complex that the experimental effect is not satisfactory when simply using such a database. Therefore, after a period of testing at a port, a large number of manual annotations from video data were conducted to build a better private database. The areas where humans are found are cropped into $64 \times 128$ sized windows and saved as positive samples, whereas areas without humans are used as negative samples.

(2) The positive and negative samples are classified and marked in the completed sample library. All positive samples are marked as $+1$, and negative samples are marked as $-1$. A HOG function for each sample is then called to extract 3,780 HOG features.

(3) The extracted features of both positive and negative samples are put into the SVM to be trained for obtaining the initial classifier.

(4) The initial detection model is used to detect negative samples of the actual port background. If human regions are found, the current sample is categorized as a hard sample.

(5) If there are too many training samples, they should be subsampled. Some of the samples from the initial positive and negative sample set are selected for retraining. The final classifier can then be developed.

2.3. Multiscale Detection and Fusion. The proposed human-detection process uses the trained detection classifier to intensively scan a static image at multiple scales. From the top-left corner of the image, the human body is detected using a $64 \times 128$ detection window by scanning the entire image from top to bottom and left to right. Because the size of the detection window is fixed and the objects can be very flexible, the general solution is to zoom in on the image until the image scale is reduced to the training scale such that it can detect objects of all sizes. Each scaling layer will then be scanned to detect all human body sizes. This method solves the problem of human detection at different scales.

During the detection process, the adjacent detection windows usually overlap. These rectangles are all marks of the same human target, which is shown in Figure 14. However, this is not what we ultimately want. The final purpose is to find a person’s exact position. Thus, there will generally be a plurality of overlapping rectangles near the target’s position in the image, and thus the results need to be fused into one final description of the detected object. In this paper, the optimization of this process is treated as a nonmaxima suppression response. The solution is to provide density estimation in positioning mode. A pattern recognition program based on the mean shift [25] is used to point to the position of maximum probability density, namely, the most accurate description of the current object. The purpose of an image fusion is to minimize the redundancy and maximize the information of interest in the input images [26].

Figures 13 and 14 show examples of a multiscale test at Shanghai Maritime University. Figure 13 shows an original camera image, and Figure 14 shows multiscale detection results projected onto the original image.
Figure 15: Three-dimensional spaces in multiscale detection.

Figure 15, and expressed as point \( y_i = [x_i, y_i, s_i] \); finally, the coordinate and scale information is saved:

\[
\begin{align*}
y_i & \quad \text{result} = 1 \\
\text{ignore} & \quad \text{result} = -1.
\end{align*}
\]

(16)

(2) For each point of the set, the bandwidth matrix is calculated using the following covariance matrix:

\[
\text{diag} \left[ H_i \right] = [ (\exp(s_i) \sigma_x)^2, (\exp(s_i) \sigma_y)^2, (\sigma_s)^2 ].
\]

(17)

(3) The mean shift vector for each point on the list is iteratively calculated until they are all merged into a single model, which means that the density is the highest when the mean shift vector of (18) is 0

\[
m(y) = H_n \frac{\nabla \hat{f}(y)}{\hat{f}(y)} \equiv y_m - y.
\]

(18)

According to (19), starting from a random point, \( y_i \) is calculated in step 1 until \( y_m \) no longer changes

\[
y_m = H_n (y_m) \left[ \sum_{i=1}^{n} w_i (y_m) H_i^{-1} y_i \right].
\]

(19)

(4) The mode centroids \( y_m \) obtained above are the final fusion results.

(5) For each mode centroid, the detection rectangles are drawn according to the center position and scale.

The green rectangles in Figure 16 are the discrete results of the initial detection. Starting from a random rectangle, the top-left point is obtained, and its neighborhood points are iterating to calculate the convergent location. To guarantee the accuracy of the person's position, it is necessary to consider one of the discrete points closest to the convergent location as the convergence result [27]. Finally, all green rectangles are grouped into a red rectangle and the feasibility of the fusion algorithm is certified.

During the classification process, the AdaBoost cascade classifier can filter out 90% of nonhuman-detection windows, and the rest of the detection windows will then be accurately classified by the SVM classifier. The time needed for the combined classifier is only 1/8 of the time needed for a classification method using an SVM as a single classifier. Since the optimized HOG algorithm can save 50% of the time needed for feature extraction, the entire human-detection algorithm in theory improves the efficiency by 500 to 800% compared with a traditional algorithm.

3. Experimental Analysis

3.1. Experimental Installation. The hardware architecture for artificial intelligence based on proposed human-detection system consists of four parts, as shown in Figure 17: high-speed HD cameras, a mounting platform (including a camera bracket, galvanized outdoor lamp posts, and an outdoor waterproof plug pole box), Ethernet communications equipment, and a server used to run the human-detection algorithm.

Considering the demands for multiangle human detection in a field environment, high-speed HD cameras were installed on the poles in Coal Terminal at Tianjin Port, as shown in Figure 18.

3.2. Experimental Results and Analysis. The numbers of positive and negative samples used to train the classifier model are 2,416 and 9,120, respectively. The results show that the accuracy of the optimized algorithm is 90% during the daytime, which is roughly the same as a traditional algorithm, although the optimized algorithm only takes 1/7 the amount of time (300 ms) to analyze a 500 × 300 image.
A random selection of detection images taken during the day and night from the operational fields of Coal Terminal is shown in Figures 19 and 20.

To verify the detection results through the use of the proposed algorithm, we collected a test sample set consisting of 1,780 scene images from Coal Terminal of Tianjin Port. The experimental results are provided in Table 5.

The positive samples in Table 5 are images with a human body, and the negative samples were images without a human body. The precision is the probability of a positive sample being accurately detected; the undetected rate is the probability of a positive sample being undetected; and a false detection rate is the probability of a negative sample being falsely detected.

By statistically analyzing the results of this experiment, good detection results were obtained during the daytime. The system has a recognition precision of 90.51%, and the error recognition rate was controlled to within 10%. However, the recognition precision was decreased to around 74.47% at night. Finally, the false recognition rate reached 21.95%, mainly owing to the less-visible local image characteristics resulting from the dim lighting at night and the complex background environment.

Since the Tianjin Port Coal Terminal has a complex background in the monitored area, the texture information [28], color, and other characteristics of the image are not combined with HOG, and some mistaken identifications will occur, as shown in Figure 21, which must be resolved in the future.
To evaluate our improved human-detection algorithm [29], we used Dalal’s evaluation standard. The test results can be described using a recall-precision (RP) curve. The recall means the coverage, that is, the ratio of correct detections to the number of practical positive test samples used. Precision is shooting, that is, the ratio of detected correctly to all detected samples. It is considered that if the distance between the center of the detection rectangle and the center of the calibrated rectangle is within 5 pixels and the coverage is above 75%, the algorithm output is correct. The RP curve of our human-detection system according to the above criteria is shown in Figure 22.

Figure 22 shows that the PR performance of our improved optimized human-detection algorithm is almost the same as Dalal’s traditional algorithm because our innovation is mainly on the detection time, that is, the testing time is greatly reduced by ensuring the detection precision. The tests use the following hardware environment: an Intel core i7-276QM with 8 GB of memory and the Windows 7 operation system. Our improved human-detection system using the proposed fast detection algorithm greatly improves the detection speed, and Figure 23 shows a comparison of the testing time between the improved system and the original system for different layers.

The number of detection windows for each scale layer during the multiscaledetection is shown in the abscissa, and the $y$ coordinate shows the corresponding testing time. We can see from the figure that the traditional HOG is more than seven times longer than the method introduced herein. For a $500 \times 300$ image, the total number of detection windows is 15,365, and the detection time is less than 300 ms.

It takes about 2,400 ms to complete the multiscaledetection of a 720p image, as shown in Figure 24, but the desired effect was not achieved. However, in engineering practice, it is not necessary to scan the entire image. The region of interest (ROI), which may be a road or an operation square in the original image, will be cut out from the original image and sent to the human-detection systems. It can be seen from Figure 24 that when the size of the ROI area is not more than $640 \times 360$, the processing time is less than 500 ms, whereas the processing time for a $500 \times 300$ image is less than 300 ms. The performance of proposed human-detection algorithm meets the requirements of bulk ports.

4. Conclusion

The surveillance system used to maintain port security must satisfy two requirements, that is, accuracy and real-time capability. For this research, Dalal’s HOG was used for human feature descriptions, because experiments have proven that
it can describe an outline of the human body well, ensuring the accuracy of the detection algorithm. However, it must be explicitly pointed out that if the windows need to be computed one by one, the detection process becomes quite slow. This is not in conformity with the real-time demands of a port and is therefore the basis of our proposed optimization method. At the same time, to maintain the detection accuracy, the original feature vectors are not partially deleted, and the use of a proposed optimized algorithm can avoid a large number of repeated calculations required by the original HOG; in addition, ignoring insignificant features doubles the efficiency. The general feature extraction speed is twice as fast as the traditional HOG algorithm. Moreover, the features are classified using the new classifier trained by images taken at a port. The entire detection time is shortened to 1/7th that of the original algorithm. This paper also contains the overall framework of the software and hardware platform used, as well as the field experiment results. The results of a field experiment at Coal Terminal of Tianjin Port show that the entire system is able to accurately achieve moving human detection, human positioning, and human target matching and detection at the same level as a traditional method. During the experiment, the overall effects of the system met the desired design requirements, and the human-detection accuracy remained at 90% during the day and 74% at night. The overall effect was improved by more than 700%.

Although the overall effect of the system reached the expected design requirements, a number of false positives occurred. This is because the method proposed in this paper only focuses on static images, which have a lack of dynamic information [30]; moreover texture information, color, and other image characteristics were not combined with HOG for Coal Terminal, which has a complex background for the monitored areas, producing some mistaken detections, which should be resolved in future work.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors would like to thank Editor Hung-Yu Wei and anonymous referees for their helpful and very delicate comments. This research was supported by the “Young University Teachers’ Training Program” of Shanghai Municipal Education Commission, “Local University Capacity Promotion Special Program” of the Science and Technology Commission of Shanghai Municipality (no. 13510501800), “Scientific Research Innovation Project” of Shanghai Municipal Education Commission (no. 14Z2140), and the “Ph.D. Innovation Program” of Shanghai Maritime University (no. 2014ycx040).

References


