Research Article

A Fast and Effective Image Preprocessing Method for Hot Round Steel Surface

Xuguo Yan, Long Wen, and Liang Gao

State Key Lab of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology, Wuhan 430074, China
School of Mechanical Engineering and Electronic Information China University of Geosciences, Wuhan 430074, China

Correspondence should be addressed to Liang Gao; gaoliang@mail.hust.edu.cn

Received 6 September 2019; Revised 12 October 2019; Accepted 14 October 2019; Published 29 October 2019

Guest Editor: Marco Perez-Cisneros

Copyright © 2019 Xuguo Yan 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 computer vision technology, more and more enterprises begin to use computer vision instead of manual inspection for steel surface defect detection. However, classical image processing methods often face great difficulties when dealing with images containing noise and distortions, which leads to low computational efficiency and poor accuracy of detection. In view of the particularity of hot round steel production, a computational intelligence method is proposed in this paper. On the basis of preliminary image preprocessing, we combine the improved PCA with genetic algorithm for feature selection and then use evolutionary computing and CUDA-based parallel computing to screen out the suspected defective image of round steel surface intelligently, quickly, and accurately. This method can provide decision support for subsequent defect analysis and production process improvement.

1. Introduction

Digital image processing originated in the 1920s. In the 1950s, people began to study digital image systematically [1]. In the past 30 years, with the development of computer and other related fields, digital image processing has been widely valued and made great achievements in many fields. For example, in the field of industrial engineering, the computer vision is used for quality test [2–5]. Some mainstream scientists believe that the computing is based on computing devices and algorithms, and there are many problems that can be solved by people but cannot be solved by computing equipment [6]. But other scientists disagree. They believe that the use of algorithm is not the only way to use computing devices [7]. As an important branch of artificial intelligence, the development of computer vision should draw lessons from and develop the ability of intelligent computing in artificial intelligence, such as knowledge learning and evolution, so as to break through the bottleneck of adaptability and intellectualization of computer vision.

As early as the 1950s, scientists have noticed that there may be a close relationship between human intelligence and machines. Subsequently, scientists tried to use machines to realize and simulate human intelligence and produced the subject of artificial intelligence. But up to now, the level of intelligence is still very limited [8]. In general, there are three basic fields of intelligent computing: fuzzy computing, neural computing, and evolutionary computing. In evolutionary computing, the genetic algorithm was founded in 1975 by professor Holland and his students at the University of Michigan. It was the first evolutionary computing method to be studied and applied and has been successfully applied to different fields and solved many problems, such as image processing [9–11] and data mining. Nowadays, the research on genetic algorithm is still continuing, and more and more scholars are involved in the research and application of genetic algorithm. Among them, the most important work is to improve the traditional genetic algorithm to solve the problem of high-dimensional computing.

The surface defect detection technique initiated in the United States and Britain in the 1960s [12]. In the mid-1970s,
Japan and Netherlands also joined the field of surface quality testing and improved the ability of defect detection [13]. In the 1990s, with the development of CCD technology, computer pattern recognition theory, artificial intelligence theory, and any other related technologies, the research work of steel surface automatic defect detection system is becoming more and more extensive and many representative technologies have emerged [14]. There are many literature studies about the methods of industrial products surface defect detection. For example, Liu et al. proposed a CNN-based defect classification method, which can detect six kinds of strip steel defects with high accuracy and meet the real-time requirements of actual production line [15]. Soukup and Huber-Mörk trained CNNs on a database of photometric stereo images of metal surface defects; by using this method, defects of rail can be recognized early in order to take countermeasures in time [16]. In these two literature studies, CNN is used to automatically extract the image features. However, CNN needs a lot of training data and has poor interpretability, which restricts its application in the practical production line. Fekri-Ershad and Tajeripour proposed a method for detecting abnormalities in surface textures based on single dimensional local binary patterns. High detection rate and low computational complexity are advantages of the proposed approach [17]. Fekri-Ershad and Tajeripour also proposed a new noise-resistant and multi-resolution surface quality detection method based on colour and texture features. This method has a high detection rate, low computational complexity, low sensitivity to noise, and rotation invariance [18]. Those literature studies rely on prior knowledge to select the features that best represent image information for defect detection. The disadvantage is that it may ignore other useful feature information and results in inaccurate classification results. At present, there are few literature studies on surface defect detection of round steel. In this case, it is necessary to find an effective method for surface defect detection of round steel. We hope the method can meet the requirements of the actual production line and can find out the features that can best reflect the image information comprehensively and accurately, so as to quickly and effectively identify the defect image.

Original images acquired from the image acquisition system are affected by different conditions, such as uneven brightness and noise. Therefore, it is inefficient to use original images directly for defect detection. For this reason, the original images must be preprocessed at the early stage of image processing [19]. In general, the image preprocessing method can be divided into three processes: image graying, image geometric changing, and image enhancement. The purpose of image graying is to transform the colour images into grayscale images to reduce the amount of data. The purpose of image geometric changing is to correct the image error caused by the image acquisition system. The purpose of image enhancement is to improve the image effect, remove the background noise, expand the difference between different object features in the image, and improve the image quality [20]. At present, most image preprocessing literature studies are basically based on these three processes to optimize the image preprocessing methods. For example, Li and Liu proposed an improved image preprocessing method. In this method, image size normalization, median filtering, and image enhancement are adopted to achieve image denoising and enhancement. This literature proposed some new ideas based on the traditional image preprocessing method [21]. Li et al. proposed another new image preprocessing method. This method takes full account of the skew, blur, and damaged images caused by various reasons and gets better image effect than the traditional preprocessing method [22]. However, the methods in these literature studies are all the innovations of the traditional preprocessing method. They only consider the effectiveness of the methods in terms of image effect without considering the processing speed requirement. Therefore, in a modern steel production process, these preprocessing methods are difficult to meet the real-time requirements.

The basic idea of parallel computing is to use multiple processors to solve the same problem cooperatively. The significance of parallelism is to shorten the time of problem solving, increase the scale of problem solving and get better solution. GPU processing scheme has the advantages of easy components, convenient programming, and high cost-effective, and it has been widely accepted and used to improving the processing speed [23–25]. In 2007, NVIDIA introduced the CUDA programming interface as a new hardware and software architecture for parallel computing. It regards GPU as a data parallel computing device and distributes and manages computing on it without mapping it to graphical APU [26]. With CUDA, GPU can be more easily used for general purpose computing [27]. Many literature studies have proved that using CUDA computing in image processing can improve processing efficiency [28–35]. For example, Zhan et al. proposed a fast CUDA-based image preprocessing method which includes image graying, Gaussian filtering, histogram equalization, and other processes of image preprocessing and achieved high-speed parallel processing. Experiments on images with different image resolutions were provided to prove the effectiveness of the method [35]. However, the experimental data used in these literatures are all from public database, without experimental support from actual production line. Some literature studies proved the efficiency of CUDA-based parallel computing in one or two specific processes in image preprocessing [36–40]. For example, Xia et al. proposed a CUDA-based image denoising method for steel plate and proved that the speed of the CUDA method was faster than that of the traditional CPU method [40].

By taking the surface images from a hot round steel production line as research objects, this paper proposed a fast and effective image preprocessing method. The main contributions of this research are as follows:

(1) A preliminary preprocessing method is designed by considering the particularity of hot round steel surface images. CUDA is used in several processes in
3. Image Preliminary Preprocessing

A simple surface defect detection system in the hot round steel production line is given in Figure 2. Two image preprocessing units cooperate in parallel, and then the images are transmitted to the cloud for defect recognition and analysis. As a prerequisite step in defect detection, image preprocessing is extremely important to realize rapid defect detection.

The original images of this research are the hot round steel surface images acquired by six linear-array CCD cameras. The diameter of the round steel products is 14 mm–27 mm. The images are grayscale images. In order to facilitate image processing, the resolution of the images after captured with a specific size is normalized to 1024 × 1024. The example of the original image of 20 mm diameter round steel is shown in Figure 3.

3.1. Effective Region Extraction. Figure 3 gives the original image (20 mm in diameter) of round steel taking by one CCD camera, where the effective region of the round steel is about 1/2. The effective region may vary according to the diameter of the round steel. The round steel is mostly in the centre of the image, with a black background on either side. The gray threshold algorithm is the mostly used effective region extraction method. However, due to the high surface temperature of hot round steel, the oil particles in the air and the complex construction site environment, the qualities of images taken by the cameras can be varied. The effect will be unstable if the gray threshold algorithm is used, which may cause background misjudgement. In this case, the Sobel operator is adopted in this research for edge detection by considering the particularity of surface images of hot round steel.

The Sobel operator is one of the most important operators in edge detection, composed by two templates corresponding to the X-axis and Y-axis edges, respectively (Figure 4). For each pixel in the image, these two templates are used for correlation calculation to obtain the edge of the image. When using $f(i, j)$ as the input image, the output image $f'(i, j)$ can be obtained by the following formula:

$$
\begin{align*}
  f'_1(i, j) &= f(i + 1, j - 1) + 2f(i + 1, j) + f(i + 1, j + 1) \\
  &- f(i - 1, j - 1) - 2f(i - 1, j) - f(i - 1, j + 1), \\
  f'_2(i, j) &= f(i - 1, j + 1) + 2f(i, j + 1) + f(i + 1, j + 1) \\
  &- f(i - 1, j - 1) - 2f(i, j - 1) - f(i + 1, j - 1), \\
  f'(i, j) &= |f'_1(i, j)| + |f'_2(i, j)|,
\end{align*}
$$

(1)

where (1) $f'_1$ and $f'_2$ are the convolution in the horizontal (X-axis) and vertical (Y-axis) directions, respectively; (2) $\cdot|$ represents the absolute value.

Referring to the original images in Figure 3, in general, the effective edge of the image on the Y-axis is the two ends of image after normalized by the specific size (1024 × 1024 in this research). However, the effective edge on the X-axis needs to be obtained by the horizontal direction convolution...
calculation with the Sobel operator. Traditionally, we can use the CPU method to get the result by serial processing of each pixel in the image. However, due to the high resolution of the image and large amount of computation, this research proposed a CUDA-based parallel Sobel edge detection method. In CUDA, the CPU is responsible for logical task processing and serial computation, while the GPU focuses on executing highly thread parallel structures. The CUDA thread structure has three important concepts: grid, block, and thread. The relationship among them is shown in Figure 5. The CUDA thread configuration affects the final execution effect, which is determined by the task characteristics and the computer hardware capacity. Multiple experiments are required to get the optimal method.

In the CUDA-based parallel Sobel edge detection method, we firstly introduce an edge pixels autocompletion technique for the pixels at the boundary position. The Sobel operator needs to read the edge pixels. However, for the pixels at the boundary position, the pixels at the top and left have exceeded the normal index range. In this case, pixels at the top and left need to be completed with appropriate value. Since the size of the Sobel operator is $3 \times 3$, Figure 6 gives the circle of pixels that needs to be completed. The purpose of edge pixels autocompletion is to make the boundary pixels...
and nonboundary pixels get the same processing, so as to speed up the processing speed. Traditional edge pixels autocompletion methods include symmetry method, adjacent region replication method, and zero-padding method. Because of the high resolution of the image, this research adopts the zero-padding method to fill the gray value of the boundary pixels as 0 directly.

Secondly, an optimal CUDA thread configuration is designed after experiments. The size of the input image in this step is $1024 \times 1024$, and the data type is unsigned char. After fully considering the task characteristics and the computer hardware capacity, the optimal thread configuration obtained by the experiments is as follows: the size of block is $64 \times 4$, and the size of grid is $1 \times 256$. The 256 blocks process the entire 1024 rows of data. Because each pixel requires a domain operation, each block must pass six rows of data including the original three rows of image data into the shared memory. On the basis of this thread configuration, each thread in one block serially calculates the output at the position of four pixels (Figure 7). Because the adjacent pixels overlap the areas required for Sobel correlation operations, when the output of a position is calculated, it shifts to the right column by one. At this point, the first two columns have been read before, so only the last column needs to be read to calculate the output. In this case, the memory access can be reduced to improving the processing speed. Furthermore, experiments prove that this configuration conforms to the access requirements of share memory and does not result in bank conflict. Therefore, the thread configuration proposed in this research is reasonable.

The result of Sobel edge detection is shown in Figure 8:

![Figure 3: Original image (20 mm in diameter).](image-url)

![Figure 4: Convolution of Sobel operator. (a) $G_x$. (b) $G_y$.](image-url)

After obtaining the Sobel edge detection result, in order to avoid background misjudgement, it is necessary to use projection calculation to search the image from left and right sides to the middle to obtain more accurate and clear images of the effective region of round steel. Since the round steel is generally located in the centre of the image, in order to avoid misjudgement caused by interference factors, the left and right starting position of the search is set at GrayScaleX. The GrayScaleX is a configurable parameter, and a certain left and right region is skipped according to the diameter of the round steel. When there is a “mutation” in the projection, it is judged that the edge point of the round steel is reached, and the left (right) edge is defined at this position. Similarly, the top and bottom boundaries can be found, but no region is skipped. The parameters of GrayScaleX and GrayScaleY are shown in Table 1.

At the same time, invalid images need to be removed. If the width of the effective region of the image after projection calculation is less than the DropWidth or the height is less than the DropHeight, the image is considered invalid. The parameters of DropWidth and DropHeight are shown in Table 2:

Then, we can obtain the final effective region extraction result, as shown in Figure 9:

3.2. Image Denoising. The surface image of hot round steel is collected by six cameras. The brightness of the image may vary with the angle of the camera, the intensity of the light source in the production site, and the diameter of the round steel, which will affect the effect of subsequent image analysis. Therefore, before image processing, it is necessary to keep the brightness of all images consistent. The brightness of a gray image ranges from 1 to 256, so we normalized the brightness of all images to the mean value 128 to ensure the overall brightness of all images is uniform. In this research, we proposed an illumination equalization method based on column projection. The specific steps are as follows:

1. Use column projection to obtain the grayscale vector $G$ of the image in the column:

   $$G = \left\{ \begin{array}{c} G_{i+9} \\ \vdots \\ G_{i-9} \end{array} \right\}. \quad (2)$$

2. Calculate the average gray value $\overline{G}$ of the images in the column:

   $$\overline{G} = \sum_{i=9}^{i+9} G_{k}.$$ \quad (3)

3. Calculate the compensation coefficient $K$ of the images in the column:
Figure 5: Grid, block, and thread in CUDA.

Figure 6: Autocompletion of pixels at the boundary position.
Multiply the value of each pixel by the compensation coefficient $K$.

This formula ensures that the obtained compensation coefficient will compensate the average grayscale value of each column to 128, and the difference between columns will not be too great. In this case, the overall brightness of the images can be improved and the details can be well retained.

The result of illumination equalization method is shown in Figure 10:

The noise component analysis is also necessary before image denoising. The images of round steel surface are collected using a linear-array CCD camera. In general, the zero-mean Gaussian white noise can be used as the model noise inside the CCD camera [41]. At the same time, as the image acquisition is carried out on the production line, external factors such as vibration of the lens and material and coating of the round steel will affect the image quality. In addition, the imperfection and sudden failure of the image acquisition system will also cause the noise interference to the images. These noises have a strong randomness, which is usually shown as the signal-independent Gaussian additive noise [42].

Therefore, the noises in the round steel images are all signal-independent Gaussian additive noise. In this case, the Gaussian filter is selected as the basic algorithm for image denoising.

Gaussian filter is a linear smoothing filter, the template used is called a Gaussian kernel, and it can be calculated by formula. The core size is called the GaussSize, and it is a configurable parameter. The larger the GaussSize, the much smoother the image effect, but the more the details of image lost. The kernel size of the Gaussian filter is determined by the GaussSize parameter. For round steels with different diameters, the optimal GaussSize parameters obtained by experiments are shown in Table 3. This table contains the parameters of round steel with most diameters. If the round steel diameter is not in the table, look for the nearest parameter to use.

Traditionally, we can use the CPU method to get the result of Gaussian filter denoising by serial processing each pixel in the image. However, due to the high resolution of the image and large amount of computation, this research proposes a CUDA-based parallel Gaussian filter denoising method. This step is similar to (Section 3.1) the CUDA-based parallel Sobel edge detection method. The design of the edge pixels autocompletion method is the first step in this process. Because the Gaussian filter needs to read edge pixels, the pixels at the boundary position need to be completed according to the filter size. Traditional edge pixels autocompletion methods include symmetry method, adjacent region replication method, and zero-padding method. Because of the high resolution of the image, this research adopts the zero-padding method to fill the gray value of the boundary pixels as 0 directly.

Secondly, an optimal CUDA thread configuration is designed after experiments. The size of the input image in this step is $1024 \times 512$, and the data type is unsigned char. Taking the images of round steel with 15 mm–19 mm diameter as the example, the GaussSize is 7 and the blur radius is 3. After fully considering the task characteristics and the computer hardware capacity, the optimal thread configuration obtained by the experiments is as follows: the size of block is $16 \times 8$, and the size of grid is $1 \times 64$. Each thread processes 64 pixels. Based on this configuration, each thread in the block serially calculates the output at the position of 8 pixels. Because the adjacent pixels overlap the areas required for the Gaussian filter, when the output of a position is calculated, it shifts to the right column by one. At this point, the first six columns have been read before, so only the last column needs to be read to calculate the output. In this case, the memory access can be reduced to improving the processing speed. Furthermore, experiments prove that this configuration conforms to the access requirements of share memory and does not result in conflict. Therefore, the thread configuration proposed in this research is reasonable.

Then, we can obtain the final effective region extraction result, as shown in Figure 11:

4. Image Screening Algorithm

The purpose of image screening algorithm is to divide the surface images of round steel into normal and defect images. The defect images will be numbered and uploaded to the
database for subsequent defect recognition and analysis. In this step, feature selection and extraction is the most important part. The commonly used image features include color feature, texture feature, shape feature, and spatial relationship feature. According to different purposes, there are great differences in the selection of features. The judgment of image screening is closely related to image feature selection.

4.1. Feature Extraction Based on Improved PCA and Genetic Algorithm. Genetic algorithm is a highly parallel, stochastic, and adaptive optimization algorithm based on “survival of the fittest.” By replication, crossover, and mutation, the “chromosome” group represented by the problem solution coding evolved from generation to generation, and eventually converged to the most suitable group, so as to obtain the optimal or satisfactory solution of the problem. Using the genetic algorithm to find the optimal feature subset of the problem space, which can greatly reduce the space of classification system and improve the search efficiency, is one of the mainstream ideas of intelligent computing at present. But one of the disadvantages of the traditional genetic algorithm is that it takes a long time to deal with and optimize the problem with high dimension. In this case, we propose a feature extraction method based on improved PCA and genetic algorithm. Firstly, we use an improved PCA method to reduce the feature dimension, and then the features after PCA are selected as the input features to reduce the time consumption of feature extraction in the genetic algorithm.

4.1.1. Improved PCA Method. PCA is a multivariate statistical method, and it is one of the most commonly used dimension reduction methods. By orthogonal transformation, a group of variables that may be correlated can be transformed into a group of linear uncorrelated variables. The transformed variables are called the principle components. In this research, we propose an improved PCA method. We firstly use traditional PCA to extract the main features (principal component) of normal and defective samples, respectively, and then according to the projection residuals of these selected features in two principal component spaces, the expression of the difference values can be evaluated. The larger the difference is, the stronger the ability of selected features that can distinguish between the normal and defect samples. 1000 images including 200 labeled defect images are selected as the samples in this step. The main steps are as follows:

1) Using matrix $A_n(N_n \times Q)$ and matrix $A_f(N_f \times Q)$ to represent the normal sample matrix and defect sample matrix, respectively, where $N$ is the number of two kinds of samples and $Q$ is the total number of features. Then, the PCA model of a normal sample is

$$A_n^e = A_n - \text{repmat}(M_n, N_n, 1),$$

$$A_n^{e^T} = T_n^P + E_n,$$  \hspace{1cm} (5)

where $A_n^e$ is the centralized matrix of $A_n$, $M_n$ is the mean vector of normal samples, and $\text{repmat}(M_n, N_n, 1)$ represents that the vector $N_n$ is used as a module, which is flattened and repeated into a matrix according to $N_n$ rows and one column. The number of rows of the matrix is $N_n$, and the number of column is the length of

| Diameter (mm) | 14  | 15  | 16  | 17  | 18  | 19  | 20  | 21  | 22  | 23  | 24  | 25  | 26  | 27  |
|------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| GrayScaleX | 230 | 220 | 220 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 200 | 170 | 170 |
| GrayScaleY | 25  | 25  | 25  | 25  | 25  | 25  | 25  | 25  | 25  | 25  | 25  | 25  | 25  | 25  |

| Diameter (mm) | 14  | 15  | 16  | 17  | 18  | 19  | 20  | 21  | 22  | 23  | 24  | 25  | 26  | 27  |
|------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| DropWidth  | 970 | 970 | 970 | 970 | 970 | 970 | 970 | 970 | 970 | 970 | 970 | 970 | 970 | 970 |
| DropHeight | 160 | 170 | 190 | 200 | 210 | 220 | 230 | 240 | 250 | 250 | 250 | 250 | 260 | 270 |

| Diameter (mm) | 14  | 15  | 16  | 17  | 18  | 19  | 20  | 21  | 22  | 23  | 24  | 25  | 26  | 27  |
|------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| DropWidth  | 970 | 970 | 970 | 970 | 970 | 970 | 970 | 970 | 970 | 970 | 970 | 970 | 970 | 970 |
| DropHeight | 160 | 170 | 190 | 200 | 210 | 220 | 230 | 240 | 250 | 250 | 250 | 250 | 260 | 270 |

**Figure 9:** Effective region extraction result (20 mm in diameter).
vector $M_n, T_n, P_n$, and $E_n$ represent the score matrix, load matrix, and residual matrix obtained by decomposition of principal components of normal samples, respectively. The number of features (principal components) we selected in this step is represented by $K_n$. Similarly, we can get the PCA model of defect sample:

$$
A_f^c = A_f - \text{repmat}(M_f, N_f, 1),
$$

$$
A_f^t = T_pf + E_f.
$$

(2) Establish the residual projection of the defect sample in the principle component space of the normal sample:

$$
e_{nf,i} = (x_{f,i} - M_N)(I - Q_fQ_f^T),
$$

where $i$ represents the sample number, $x_{f,i}$ is the feature expression data of defect samples in normal samples, and $T$ is the matrix transposition.

(3) Calculate the average residual variance of projection of the defect sample in the principal component space of the normal sample:

$$
S_{nf}^2 = \frac{1}{(N_f - K_n - 1)(Q - K_n)} \sum_{i=1}^{N_f} e_{nf,i}^T e_{nf,i}.
$$

(4) Calculate the differences between the mean residual variance of the projection of the defect sample in the principle component space and the mean residual variance of the projection of the normal sample in the principle component space:

$$
|D| = S_{f,n}^2 \times S_{n,n}^2 - S_{nf}^2 \times S_{f,n}^2.
$$

Figure 10: (a) Original image; (b) image after illumination equalization (20 mm in diameter).

Table 3: GaussSize parameters of round steel with different diameters (mm).

| Diameter | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 |
|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| GaussSize | 5  | 5  | 7  | 7  | 7  | 7  | 9  | 9  | 9  | 9  | 9  | 9  | 9  | 9  |
(5) Replace the types of defect samples, return to step 1, until the first 32 features (principle component) that best reflect the image features are obtained.

4.1.2. Feature Extraction Based on Genetic Algorithm. After improving the PCA method, the first 32 features with the strongest distinguishing ability are obtained. In this step, we use these 32 features as the input features for feature extraction in order to reduce the time consumption in genetic algorithm. The main processes of the genetic algorithm are shown in Figure 12:

The establishment of evaluation criterion is the key in this step. The purpose of this research is to find the defect images of round steel. Therefore, we use the variance between different regions and within same region as the evaluation criterion. In this case, the fitness value calculation formula is designed to evaluate the advantage of the feature subset. Assuming the sample type is \( C \), the number of samples in each type is \( N_i \), and \( p_i \) represents the prior probability of each sample. Then the mean vector \( m_i \) and the total mean vector of each sample \( m \) can be calculated as follows:

\[
m_i = \frac{1}{N_i} \sum_{k=1}^{N_i} x_k^{(i)},
\]

\[
m = \sum_{i=1}^{c} p_i m_i.
\]

Then, we can get the total discreteness matrix \( S_d \) between different sample classes:

\[
S_d = \sum_{i=1}^{c} p_i (m_i - m)^T (m_i - m).
\]
and the total discreteness matrix $S_j$ within the sample class:

$$S_j = \sum_{i=1}^{N} \frac{1}{N} (x_k^{(i)} - m_j)^T (x_k^{(i)} - m_j).$$  \hspace{1cm} (12)

The bigger the dispersion between different sample classes, the better the separateness; and the smaller the dispersion with sample class, the better the separateness, too. Therefore, the fitness value calculation formula of the genetic algorithm for feature optimization is

$$\text{eval} = \text{trace}(S_d - S_j).$$  \hspace{1cm} (13)

Other parameters are selected as follows: the group size $M = 80$, crossover probability $P_c = 0.8$, and the mutation probability $P_m = 0.01$.

4.2. Defective Image Preliminary Screening. After the genetic algorithm, 16 dimensional feature vectors of the image are selected. However, when deciding whether an area is defective, in addition to analyzing the features of region, the image features around it must also be considered. In this case, this research proposed a parallel defective image preliminary screening method based on CUDA and evolutionary computing. 1000 images including 200 defective images are selected as the samples in this step. The main steps are as follows.

(1) Design of the decision algorithm of defect images:

Step 1: input image $P$ (1024 $\times$ 512)
Step 2: divide image $P$ into $B$ blocks ($B = 128$)
Step 3: divide each block into $Z$ regions ($Z = 64$), and record the adjacent relations of these regions with the adjacency matrix
Step 4: carry out 64 computations, parallel computing the following tasks:

(i) The determination feature $T$ of each region
(ii) The determination feature $T'$ of the input image $P$
(iii) The difference between the determination feature of each region and the overall determination feature ($T - T'$):

(a) If the difference value ($T - T'$) is greater than the specified threshold value ($MT$), mark the region as suspicious.
(b) If the number of adjacent suspicious areas reaches the specified number (3 in this research), then judge the image as suspected defective images and record the suspicious location. Otherwise, it is normal.

(2) Design of learning algorithm of defect images:

Step 1: input defective image $P'$ (1024 $\times$ 512)
Step 2: divide image $P'$ into $B'$ blocks ($B' = 128$)
Step 3: divide each block into $Z'$ regions ($Z' = 64$) and record the adjacent relations of these regions with the adjacency matrix
Step 4: record the corresponding area of defects
Step 5: carry out 64 computations, parallel computing the following tasks:

(i) Randomly generate a group of criteria according to $T$ and set the initial global evaluation function $MT$: $MT = T_i - T_m$, where $T_i$ is the internal feature difference of one defective region and $T_m$ is the internal feature difference of one nondefective region.
(ii) Calculate the evaluation function $MT_z$:

(a) If $MT_z < MT$, the current judgment basis $T$ is recorded and the $MT_z$ is updated as the global evaluation function $MT$
(b) If $MT_z > MT$, update the judgment according to $T$
(iii) Return the last step, until the evaluation function value remains unchanged for 100 times (the difference value is less than 0.000001)

Step 6: each calculation is synchronized to obtain the optimal judgment basis $T$ and the threshold value $MT$ to minimize the judgment error of the defect region
Step 7: replace the input defective image, return to step 1, and calculate the new judgment feature $T$ and threshold value $MT$, until the $T$ and $MT$ are stable
Step 8: output the optimal judgment feature $T$ and threshold value $MT$

5. Technological Setup and the Comparisons of Experimental Results

5.1. Technological Setup. The computer hardware configuration for program execution in this research is

(i) Graphic card: NVIDIA GeForce GTX 960 and
(ii) CPU: Intel(R) Core(TM) i5-4660 3.20 GHz.

More details are shown in Table 4.

According to the hardware and software configuration of the computer in Table 4, the computer performance parameters are shown in Table 5. The CUDA thread configuration proposed in this research is the optimal configuration after considering the selected computer performance.

5.2. Comparison of Experimental Results. The main purpose of this research is to divide the surface images of round steel into normal and defective images. In order to facilitate the comparison, 2000 round steel surface images samples (including 300 defective images) from a hot round steel production line are selected. There are four common defect types in round steel surface: scratch defect, rolling skin, crack defect, and wire ear defect. We firstly use the traditional PCA method to detect the four kinds of defect samples, and get the detection rate. Then the combination of improved PCA and the genetic algorithm proposed in this research is applied to the same samples. The experimental results are shown in Table 6:
We can see clearly from the table, traditional PCA cannot detect the rolling skin, crack defect and wire ear defect completely correctly. However, by combining genetic algorithm to reduce the dimension of the feature, the defect images can be screened out completely correctly. The reason is in traditional PCA, if the feature dimension is high but the training sample is small, the estimation of model parameters will be incorrect. In this case, we should combine the genetic algorithm to reduce the number of features to achieve better detection results.

At the same time, in order to prove the effectiveness of the parallel computing, the tasks of Sobel edge detection, Gaussian denoising and defect image preliminary screening are also carried out with OpenCV on the CPU side. The time comparison of CPU and CUDA is shown Table 7:

As can be seen from Table 7, the CUDA-based parallel computing greatly improves the processing speed in the three tasks. In Sobel edge detection task, it gets the highest degree of parallelism, the acceleration ratio reaches 74.24.

### 6. Comprehensive Conclusion and Future Research

By taking the surface images from a hot round steel production line as research objects, this paper proposed a fast and effective image preprocessing method. Firstly, a preliminary preprocessing method is designed by considering the particularity of hot round steel surface images. Then, an image screening algorithm based on intelligent computing is designed. We combine improved PCA with genetic algorithm for feature selection to solve the time-consuming problem of traditional genetic algorithm due to the high dimensions. Finally, a defect image screening algorithm is designed to efficiently screen out the normal images and the defect images. The CUDA-based parallel computing is also used in several processes to improve the speed of the method.

Future research can be carried out from the following aspects:

1. The proposed image preprocessing method is based on the particularity of round steel surface images from a hot round steel production line. Future research can focus on combining CUDA parallel computing with other preprocessing processes by considering the task particularity, in order to get more accurate results.

2. The CUDA thread configuration and the computer capacity have great influence on the processing speed. In actual the production process, we should focus on the tasks particularity and computer capacity and find the more suitable CUDA thread configuration.

3. The genetic algorithm used in this paper is a classical intelligent computing algorithm. At present, many advanced intelligent algorithms are emerging, so we can combine these advanced intelligent algorithms with production practice to better serve the production.

4. Although the propose method in this research can quickly and effectively screen out the defect images from normal images, how to classify the identified defect type still needs to be matched with the appropriate classification method and actual production line to achieve better defect classification results.
The proposed method can be applied to the steel surface images with high contrast, and the background texture of the image is similar. The applicability of this method to other images with low contrast and changeable background texture still needs to be verified in practice.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was funded by the National Natural Science Foundation for Distinguished Young Scholars of China under Grant no. 51825502 and the Natural Science Foundation of China (NSFC) under Grant no. 51454009 and supported by “Program for HUST Academic Frontier Youth Team under Grant no. 2017QYTD04.”

References

[1] A. Sanchez-Romero, J. A. Gonzalez, J. Calbo, and A. Sanchez-Lorenzo, “Using digital image processing to characterize the Campbell-Stokes sunshine recorder and to derive high-temporal resolution direct solar irradiance,” Atmospheric Measurement Techniques, vol. 8, no. 1, pp. 183–194, 2015.

[2] P. Viola and M. J. Jones, “Robust real-time face detection,” International Journal of Computer Vision, vol. 57, no. 2, pp. 137–154, 2004.

[3] Q. Wu, H. Li, K. N. Ngan, and K. Ma, “Blind image quality assessment using local consistency aware retriever and uncertainty aware evaluator,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 28, no. 9, pp. 2078–2089, 2018.

[4] R. J. Hopcroft, R. Motwani, and J. D. Ullman, Introduction to Automata Theory, Language and Computation, Pearson Addison-Wesley, Boston, MA, USA, 3rd edition, 2007.

[5] P. Wegner, “Why interaction is more powerful than algorithms,” Communications of the ACM, vol. 40, no. 5, pp. 80–91, 1997.

[6] J. Franco, “The brick wall: NP completeness,” IEEE Potentials, vol. 16, no. 4, pp. 37–40, 2002.

[7] M. Sonka, V. Hlavac, and R. Boyle, Image Processing, Analysis, and Machine Vision, Chapman & Hall Computing, London, UK, 2nd edition, 2007.

[8] B. Huang, S. Lee, and J. Ming, “Adaptive image segmentation using a genetic algorithm,” IEEE Transactions on Systems Man & Cybernetics, vol. 25, no. 12, pp. 1543–1567, 2002.

[9] P. S. Brumby, J. P. Theiler, S. J. Perkins et al., “Investigation of image feature extraction by genetic algorithm,” Proceedings of SPIE—The International Society for Optical Engineering, vol. 3812, pp. 24–31, 1999.

[10] W. L. Orton, W. J. Hill, and R. A. Brook, “Automatic visual inspection of moving steel surface,” British Journal of NDT, vol. 9, pp. 242–258, 1977.

[11] D. J. Purll, “Automated surface inspection with solid-state image sensors,” in Proceedings of the Industrial Applications of Solid State Image Scanners, vol. 148, pp. 18–25, London, UK, January 1978.

[12] X. Wang and Z. J. Xie, “Research on automatic inspection system for defects on precise optical surface based on machine vision,” Journal of Chongqing University: English Edition, vol. 148, no. 2, pp. 89–93, 2006.

[13] Y. Liu, J. Geng, Z. Su, W. Zhang, and J. Li, “Real-time classification of steel strip surface defects based on deep CNNs,” in Proceedings of the 2018 Chinese Intelligent Systems Conference, vol. 5298, pp. 257–266, Hong Kong, China, June 2018.

[14] D. Soukup and R. Huber-Mörk, “Convolutional neural networks for steel surface defect detection from photometric stereo images,” in Proceedings of the International Symposium on Visual Computing (ISVC 2014): Advances in Visual Computing, pp. 668–677, Las Vegas, NV, USA, December 2014.

[15] S. Fekri-Ershad and F. Tajeripour, “A robust approach for surface defect detection based on one dimensional local binary patterns,” Indian Journal of Science and Technology, vol. 5, no. 8, pp. 3197–3203, 2012.

[16] S. Fekri-Ershad and F. Tajeripour, “Multi-resolution and noise-resistant surface defect detection approach using new version of local binary patterns,” Applied Artificial Intelligence, vol. 31, no. 5–6, pp. 395–410, 2017.

[17] B. Qi, H. Shi, Y. Zhuang, H. Chen, and L. Chen, “On-board, real-time preprocessing system for optical remote-sensing imagery,” Sensors, vol. 18, no. 5, pp. 1328–1330, 2018.

[18] C. Bradley, “Automated surface roughness measurement,” The International Journal of Advanced Manufacturing Technology, vol. 16, no. 9, pp. 668–674, 2000.

[19] Z. Li and W. M. Liu, “Research on license plate image preprocessing in vehicle license plate recognition system,” Science and Technology Engineering, vol. 8, pp. 2081–2084, 2011.

[20] D. F. Li, Y. F. Ding, and X. Y. Qiu, “Research on license Plate Image preprocessing method based on VCC++,” Electron Mass, vol. 11, pp. 35–37, 2011.

[21] S. Zuo, D. G. Donoro, Y. Zhang, Y. Bai, and X. Zhao, “Simulation of challenging electromagnetic problems using a massively parallel finite element method solver,” IEEE Access, vol. 7, pp. 20346–20362, 2019.

[22] Q. R. Qiu, Q. Wu, M. Bishop, R. E. Pino, and R. W. Linderman, “A parallel neuromorphic text recognition system and its implementation on a heterogeneous high-performance computing cluster,” IEEE Transactions on Computer, vol. 62, no. 5, pp. 886–889, 2013.

[23] F. G. Tinetti, M. Mariano, and A. D. Giusti, “Restructuring fortran legacy applications for parallel computing in multiprocessors,” The Journal of Supercomputing, vol. 64, no. 2, pp. 638–659, 2013.

[24] L. Fatone, M. Giacinti, M. Mariani, M. C. Recchioni, and F. Zirilli, “Parallel option pricing on GPU: barrier options and
realized variance options,” *The Journal of Supercomputing*, vol. 62, no. 3, pp. 1480–1501, 2012.

[27] V. V. Grubov and V. O. Nedaivozov, “Stream processing of multichannel EEG data using parallel computing technology with NVIDIA CUDA graphics processors,” *Technical Physics Letters*, vol. 44, no. 5, pp. 435–455, 2018.

[28] P. D. Gutiérrez, J. M. Lastra, J. M. Benítez, and F. Herrera, “SMOTE-GPU: big data preprocessing on commodity hardware for imbalanced classification,” *Progress in Artificial Intelligence*, vol. 6, no. 4, pp. 347–354, 2017.

[29] Y. K. Wang and W. B. Huang, “A CUDA-enabled parallel algorithm for accelerating retinex,” *Journal of Real-Time Image Processing*, vol. 9, no. 3, pp. 407–425, 2014.

[30] S. Aydin, R. Samet, and O. F. Bay, “Real-time parallel image processing applications on multicore CPUs with OpenMP and GPGPU with CUDA,” *The Journal of Supercomputing*, vol. 74, no. 6, pp. 2255–2275, 2018.

[31] M. Yano, H. Mori, Y. Akiyama, T. Yamada, and K. Kurokawa, “CLAST: CUDA implemented large-scale alignment search tool,” *BMC Bioinformatics*, vol. 15, no. 1, p. 406, 2014.

[32] Y. U. Reyes, L. M. José, L. C. Erika, N. D. Lizzie, and T. S. Joel, “A parallel algorithm for the computing of ellipse present in conglomerates using GPU,” *Mathematical Problems in Engineering*, vol. 2018, Article ID 5714638, 17 pages, 2018.

[33] F. Wang, X. Jiang, and X. P. Hu, “A TBB-CUDA implementation for background removal in a video-based fire detection system,” *Mathematical Problems in Engineering*, vol. 2014, Article ID 692921, 6 pages, 2014.

[34] O.-K. Ha, K. Lee, W.-J. Kim, and K. S. Yoon, “Effective parallelization method for object recognition in 2D sonar images based on task partitioning,” *Scientific Programming*, vol. 2019, Article ID 8908950, 9 pages, 2019.

[35] Z. F. Zhan, G. Li, X. H. Zhang, and X. Y. Yin, “Parallel study of image preprocessing based on CUDA,” *Mechanical and Electronic*, vol. 91, no. 1, pp. 426–431, 2014.

[36] L. S. Wang, D. Li, and S. H. Huang, “An improved parallel fuzzy connected image segmentation method based on CUDA,” *Bio Medical Engineering Online*, vol. 15, no. 1, p. 56, 2016.

[37] X. Yang, L. Jian, W. Wu et al., “Implementing real-time RCF-retinex image enhancement method using CUDA,” *Journal of Real-Time Image Processing*, vol. 16, no. 1, pp. 115–125, 2018.

[38] T. Matić, I. Aleksi, H. Željko, and K. Dieter, “Real-time biscuit tile image segmentation method based on edge detection,” *ISA Transactions*, vol. 76, pp. 246–254, 2018.

[39] G. W. Lee, J. Ko, and T. Y. Choe, “Two-way partitioning of a recursive Gaussian filter in CUDA,” *EURASIP Journal on Image and Video Processing*, vol. 2014, no. 1, p. 33, 2014.

[40] H. L. Xia, Y. P. Chen, S. Y. Zhou, and Y. G. Tan, “Denoising method of steel sheet image based on CUDA,” *Information Technology*, vol. 11, pp. 35–39, 2017.

[41] B. T. Jiang, Y. H. Qiu, Y. Wen, and Z. Chen, “Modeling and analyzing the noise of CCD,” *Electro-Optic Technology Application*, vol. 25, no. 2, pp. 64–67, 2010.

[42] L. Luo, “A survey of noise in modern communication systems,” *Journal of Beijing Institute of Graphic Communication*, vol. 25, pp. 51–57, 2017.
