Application of Parallel Algorithm in Image Processing of Steel Surfaces for Defect Detection

Mostafa SADEGHI1,*, Hossein SOLTANI2, Kamran ZAMANIFAR3

1Department of Computer, Zavareh Branch, Islamic Azad University, Zavareh, Iran
2Department of Computer, Ashkezar Branch, Islamic Azad University, Ashkezar, Iran
3Department of Computer Engineering, University of Isfahan, Isfahan, Iran

Received: 22.03.2015; Accepted: 29.05.2015

Abstract. As the industry of steel manufacturing progresses all over the world, application of image processing has also developed in quality control of products. One of the main problems in image processing of steel surfaces in large volume is performing matrix multiplication. In this paper, we will discuss that by using the algorithm of matrix multiplication in parallel for applying filter on the image of steel surfaces, we are able to detect the defects made on them with higher speed and accuracy.

Keywords: Image Processing, Gabor Wavelet, Quality control, steel process, Parallel Multiplication Matrix

INTRODUCTION

Quality control is one of the major issues in the industry of steel sheets production. Detection of surface defects devotes a great percent of quality control process to itself. Steel manufactures would like to know unexpected defect so that to prevent them occur continually, and also make sure that quality of products satisfy customer’s need [1]. On the other hand, the defects made must be recorded on the statistical quality reports. Without an inspection system for surveying surface defects, record and investigation for causes of defects may take a long time.

During this delay, the problem can be repeated so that quality of production reduces. Many researchers have surveyed the defects produced on the steel surfaces in different approaches. Some of them utilize the method of edge finding [2]. In [3], by applying one of methods of tissue analysis, named “concurrent matrix”, detection of surface defects of steel sheets is discussed. Concurrent matrix approach has shown low efficiency for the defects found on tissues, in comparison with other techniques such as filter-based method.

In [4] and [5] considerable tissue features are extracted from images, using Gabor filters, which includes both different directions and different frequencies. In [6] some approaches are presented in all of which the task of feature extraction from images is performed by means of Gabor filters. Because of optimality in both local and frequency domain, Gabor filters makes it possible to utilize advantages of signal processing in both domains.

In this paper, we investigate the algorithm of parallel matrix multiplication in section2. In section 3, the method of tissue analysis and introduction of Gabor wavelet is presented, and then extraction of features from images by help of Gabor wavelet is discussed. The presented model and results achieved from various tests is introduced in section 4. In section 5, by showing the results of experiments conducted on the set of images, the efficiency of suggested method is indicated in two forms, namely the table of results and the image resulted from the experiment. Finally, in section 6 the results of the method introduced in this paper will be presented.

*Corresponding author. Email: msadeghi@khuisf.ac.ir

Special Issue: Technological Advances of Engineering Sciences
PRESENTING A NEW APPROACH FOR MULTIPLICATION OF MATRIX

In the method presented here, we used the idea of polynomial multiplication to reduce the time required for multiplying matrices\(^{[7]}\). At first we convert each matrix to a polynomial. Powers of two polynomials must be such that in case of multiplying two polynomials, when all elements of row I are multiplied by column \(j\) of second matrix, the resulted powers are equal so that they can be added together. Consider the following example:

\[
\begin{bmatrix}
3 & 4 \\
6 & 2
\end{bmatrix}
\begin{bmatrix}
2 \\
4
\end{bmatrix}
= (3\times2) + (4 \times6) = 30
\]

\((3x^0 + 4x^1)(2x^1 + 6x^0) = 6x^1 + 18x^0 + 8x^2 + 24x^1 = 8x^0 + 30x^1 + 8x^2\)

As it could be seen, the product stays in the coefficient of \(x^1\). In this solution, we extend two matrices in the following form:

First matrix: The value of \(x\) with even powers which increase in column from last to first must be added together. Consider the following example:

\[
n = 2, A = \begin{bmatrix}
a_{11} & a_{12} \\
a_{21} & a_{22}
\end{bmatrix} \Rightarrow \begin{bmatrix}
a_{11}x^0 & \rightarrow \ a_{12}x^2 \\
a_{21}x^2 & \rightarrow \ a_{22}x^0
\end{bmatrix}
\]

\[
n = 3, A = \begin{bmatrix}
a_{11} & a_{12} & a_{13} \\
a_{21} & a_{22} & a_{23} \\
a_{31} & a_{32} & a_{33}
\end{bmatrix} \Rightarrow \begin{bmatrix}
a_{11}x^0 & \rightarrow \ a_{12}x^2 & \rightarrow \ a_{13}x^4 \\
a_{21}x^8 & \rightarrow \ a_{22}x^{10} & \rightarrow \ a_{23}x^{12} \\
a_{31}x^{16} & \rightarrow \ a_{32}x^{18} & \rightarrow \ a_{33}x^{20}
\end{bmatrix}
\]

Second matrix: The value of \(x\) with odd powers which increase in column from last to first must be added to each element. The reason for this inverse order is that while multiplying rows by columns, the powers of elements become equal. Below, we will constitute a matrix for \(n=2\) and \(n=3\) as an example.

\[
n = 2, B = \begin{bmatrix}
b_{11} & b_{12} \\
b_{21} & b_{22}
\end{bmatrix} \Rightarrow \begin{bmatrix}
b_{11}x^3 & \rightarrow \ b_{12}x^7 \\
b_{21}x^{11} & \rightarrow \ b_{22}x^{15}
\end{bmatrix}
\]

\[
n = 3, B = \begin{bmatrix}
b_{11} & b_{12} & b_{13} \\
b_{21} & b_{22} & b_{23} \\
b_{31} & b_{32} & b_{33}
\end{bmatrix} \Rightarrow \begin{bmatrix}
b_{11}x^5 & \rightarrow \ b_{12}x^{11} & \rightarrow \ b_{13}x^{17} \\
b_{21}x^7 & \rightarrow \ b_{22}x^{15} & \rightarrow \ b_{23}x^{23} \\
b_{31}x^{11} & \rightarrow \ b_{32}x^{19} & \rightarrow \ b_{33}x^{25}
\end{bmatrix}
\]

The reason for choosing odd and even numbers as power in the matrices is that we don’t want the elements to be repeated, and the reason for adding 4 at the end of matrix A is that similar numbers do not appear in matrix C. Now, we multiply the two matrices A and B (\(n=2\)):
Application of Parallel Algorithm in Image Processing of Steel Surfaces for Defect Detection

\[
\begin{bmatrix}
0 & 2 & 4 & \cdots & 2(n-1) \\
2(n+1) & 2(n+2) & 2(n+3) & \cdots & 2(2n) \\
2(2n+2) & 2(2n+3) & 2(2n+4) & \cdots & 2(3n+1) \\
\vdots & \vdots & \vdots & & \vdots \\
2(n^2-1) & \cdots & \cdots & \cdots & 2(n^2+n-2)
\end{bmatrix}
\]

(1)

\[
\begin{bmatrix}
2n-1 & 4n-1 & 6n-1 & \cdots & 2n^2-1 \\
5 & 2n+5 & 4n+5 & \cdots & 2n^2-2n+5 \\
3 & 2n+3 & 4n+3 & \cdots & 2n^2-2n+3 \\
1 & 2n+1 & 4n+1 & \cdots & 2n^2-2n+1 \\
\vdots & \vdots & \vdots & & \vdots \\
(2n^2+2n-3) & \cdots & \cdots & \cdots & 4n^2-3
\end{bmatrix}
\]

(2)

\[
\begin{bmatrix}
(2n-1) & (4n-1) & (6n-1) & \cdots & (2n^2-1) \\
(4n+1) & (6n+1) & (8n+1) & \cdots & (2n^2+2n+1) \\
(6n+3) & (8n+3) & (10n+3) & \cdots & (2n^2+4n+3) \\
\vdots & \vdots & \vdots & & \vdots \\
(2n^2+2n-3) & \cdots & \cdots & \cdots & 4n^2-3
\end{bmatrix}
\]

(3)

The relationship 3 shows the power of product matrix:

\[
C_{ij} = \sum_{k=0}^{n-1} a_{ik} b_{kj} x^{2[(n+1)i+k]+2[(j+1)n-k]-1}
\]

(4)

For example, for \(n=7\) powers are calculated as follows (by using 4, 5 and 6):

\[
A_{power} =
\begin{bmatrix}
0 & 2 & 4 & 6 & 8 & 10 & 12 \\
16 & 18 & 20 & 22 & 24 & 26 & 28 \\
32 & 34 & 36 & 38 & 40 & 42 & 44 \\
48 & 50 & 52 & 54 & 56 & 58 & 60 \\
64 & 66 & 68 & 70 & 72 & 74 & 76 \\
80 & 82 & 84 & 86 & 88 & 90 & 92 \\
96 & 98 & 100 & 102 & 104 & 106 & 108
\end{bmatrix}
\]
PROVING THE CORRECTNESS OF POWER

It should be proven that power of each element of $C$ is unique, because otherwise, when it is converted to a polynomial, equal powers are added together, which is not desired. To prove this, we consider three cases:

1) $\begin{cases} i_1 = i_2 & 0 \leq i_1, i_2 \leq n - 1 \\ j_1 \neq j_2 & 0 \leq j_1, j_2 \leq n - 1 \end{cases}$

\[ C_{i_1 j_1} = C_{i_2 j_2} \]

\[ C_{i_1 j_1} = 2[n(i_1 + j_1 + 1) + i_1] - 1 \Rightarrow C_{i_1 j_1} = C_{i_2 j_2} \]

\[ C_{i_2 j_2} = 2[n(i_2 + j_2 + 1) + i_2] - 1 \Rightarrow C_{i_1 j_1} = C_{i_2 j_2} \]

\[ \Rightarrow 2[n(i_1 + j_1 + 1) + i_1] - 1 = 2[n(i_2 + j_2 + 1) + i_2] - 1 \]

\[ \Rightarrow n(i_1 + j_1 + 1) + i_1 = n(i_2 + j_2 + 1) + i_2 \]

\[ \Rightarrow i_1 + j_1 + 1 = i_2 + j_2 + 1 \]

\[ \Rightarrow j_1 = j_2 \]

2) $\begin{cases} i_1 = i_2 & 0 \leq i_1, i_2 \leq n - 1 \\ j_1 \neq j_2 & 0 \leq j_1, j_2 \leq n - 1 \end{cases}$

\[ C_{i_1 j_1} = C_{i_2 j_2} \]

\[ C_{i_1 j_1} = 2[n(i_1 + j_1 + 1) + i_1] - 1 \Rightarrow C_{i_1 j_1} = C_{i_2 j_2} \]

\[ C_{i_2 j_2} = 2[n(i_2 + j_2 + 1) + i_2] - 1 \Rightarrow C_{i_1 j_1} = C_{i_2 j_2} \]

\[ \Rightarrow 2[n(i_1 + j_1 + 1) + i_1] - 1 = 2[n(i_2 + j_2 + 1) + i_2] - 1 \]

\[ \Rightarrow n(i_1 + j_1 + 1) + i_1 = n(i_2 + j_2 + 1) + i_2 \]

\[ \Rightarrow n(i_1 + j_1 + 1) + i_1 = n(i_2 + j_2 + 1) + i_2 \]

\[ \Rightarrow i_1 (n + 1) = i_2 (n + 1) \]

\[ \Rightarrow i_1 = i_2 \]

3) $\begin{cases} i_1 = i_2 & 0 \leq i_1, i_2 \leq n - 1 \\ j_1 \neq j_2 & 0 \leq j_1, j_2 \leq n - 1 \end{cases}$

\[ C_{i_1 j_1} = C_{i_2 j_2} \]
Application of Parallel Algorithm in Image Processing of Steel Surfaces for Defect Detection

\[ C_{i,j_1} = 2[n(i_1 + j_1 + 1) + i_1] - 1 \]
\[ C_{i,j_2} = 2[n(i_2 + j_2 + 1) + i_2] - 1 \]
\[ \Rightarrow 2[n(i_1 + j_1 + 1) + i_1] - 2[n(i_2 + j_2 + 1) + i_2] - 1 \]
\[ \Rightarrow n(i_1 + j_1 + 1) + i_1 = n(i_2 + j_2 + 1) + i_2 \]
\[ \Rightarrow i_1(n + 1) + j_1 = i_2(n + 1) + n j_f \]
\[ \Rightarrow i_1 + j_1 - i_2 = n(j_2 - j_1) \]
\[ \Rightarrow \frac{n}{(i_1 - i_2)} = (j_2 - j_1) \]

OVERHEAD OF MULTIPLICATION OPERATION

When a matrix is converted to a polynomial according to relationships 4 and 5, although it is true that equal powers of rows and columns are added,

There are some other elements with the same power that are added to the answer. This part called “multiplication overhead”. As an example, we obtain multiplication overhead for \( n=2 \) in the following operation:

\[
\begin{bmatrix}
  a_{11} & a_{12} \\
  a_{21} & a_{22}
\end{bmatrix} \times \begin{bmatrix}
  b_{11} & b_{12} \\
  b_{21} & b_{22}
\end{bmatrix} = \begin{bmatrix}
  c_{11} & c_{12} \\
  c_{21} & c_{22}
\end{bmatrix}
\]

\[
\begin{bmatrix}
  a_{11}x^0 & a_{12}x^2 \\
  a_{21}x^6 & a_{22}x^8
\end{bmatrix} \times \begin{bmatrix}
  b_{11}x^3 & b_{12}x^7 \\
  b_{21}x^1 & b_{22}x^5
\end{bmatrix} = \begin{bmatrix}
  c_{11}x^3 & c_{12}x^7 \\
  c_{21}x^9 & c_{22}x^{13}
\end{bmatrix}
\]

\[
= (a_{11}x^0 + a_{12}x^2 + a_{21}x^6 + a_{22}x^8)(b_{11}x^3 + b_{12}x^7 + b_{21}x^1 + b_{22}x^5 + b_{12}x^7)
\]

\[
= a_{11}b_{21}x^4 + a_{11}b_{12}x^2 + a_{12}b_{22}x^5 + a_{11}b_{12}x^7 + a_{12}b_{21}x^3 + a_{12}b_{11}x^5 + a_{12}b_{22}x^7 + a_{12}b_{12}x^9 + a_{22}b_{21}x^9 + a_{22}b_{11}x^11 + a_{22}b_{22}x^{13} + a_{22}b_{12}x^{15}
\]

We consider the powers which belong to \( C \), and delete the others:

\[
\begin{bmatrix}
  a_{11}b_{11}x^3 + a_{11}b_{12}x^7 \\
  a_{12}b_{21}x^3 + a_{12}b_{22}x^7 + a_{12}b_{21}x^7 + a_{12}b_{11}x^9 + a_{22}b_{21}x^{13} + a_{22}b_{11}x^{15} \\
  a_{21}b_{11}x^3 + a_{22}b_{21}x^9 + a_{21}b_{21}x^9 + a_{22}b_{22}x^{13}
\end{bmatrix}
\]

Terms that are underlined, are system overheads which should be omitted. This procedure is accomplished for each element separately.

If we count matrix elements and call the new matrix “C-Count “, we will have the following matrix:

\[
\begin{bmatrix}
  (a_{11}b_{11} + a_{12}b_{21})x^3 + (a_{11}b_{12} + a_{12}b_{22} + a_{21}b_{21})x^7 \\
  (a_{21}b_{11} + a_{22}b_{21} + a_{12}b_{21})x^9 + (a_{21}b_{12} + a_{22}b_{22})x^{13}
\end{bmatrix}
\]

267
Therefore, the matrix \( C_{\text{-} \text{Count}} \) can simply be concluded as follows:

\[
C_{\text{-} \text{Count}} = \begin{bmatrix}
  n & 2n - 1 & 3n - 2 & \cdots & n^2 - n + 1 \\
  2n & 3n - 1 & 4n - 2 & \cdots & 2n^2 - 2n \\
  3n & 4n - 1 & 5n - 2 & \cdots & 3n^2 - 3n \\
  n^2 - n + 1 & \cdots & 3n - 1 & 2n - 1 & n
\end{bmatrix}
\]

(7)

In order to achieve response matrix, the overhead part must be omitted. Since the time order of overhead omitting for each element is \( O(n^2) \), we can use parallel processing to perform this job, in order to increase the speed of overhead omitting procedure.

PERFORMING PRESENTED ALGORITHM

\[
\text{NEWMULTMATRIX}(A[1..n,1..n],B[1..n,1..n])
\]

\[
\begin{align*}
&\text{For } i \leftarrow 0 \text{ to } 2(n^2 + n - 2) \\
&\quad A_{\text{temp}}[i] \leftarrow 0 \\
&\text{For } i \leftarrow 0 \text{ to } 2n^2 - 1 \\
&\quad B_{\text{temp}}[i] \leftarrow 0 \\
&\quad \text{For } i \leftarrow 0 \text{ to } n \\
&\quad \text{For } j \leftarrow 0 \text{ to } n \\
&\quad A_{\text{temp}}[2((n + 1)i + j)] \leftarrow A[i,j] \\
&\quad B_{\text{temp}}[2((j + 1)n - i) - 1] \leftarrow A[i,j] \\
&\quad C_{\text{temp}} \leftarrow \text{FFTMULTIPLY} \left[ A_{\text{temp}}[0..n^2 + n - 2], B_{\text{temp}}[0..2n^2 - 1] \right] \\
&\quad O(n^3) \\
&\text{For } i \leftarrow 0 \text{ to } n \\
&\quad \{ \}
\end{align*}
\]

The time to perform the algorithm which was presented above is \( O(n^2 \log n) \). However, if we execute FFT algorithm in parallel, the time of performance is reduced to \( (n^2) \).

TEXTURE ANALYSIS AND FEATURE EXTRACTION WITH USING GABOR WAVELET-TODAY

In various stage of steel sheet production, different effects are made on the sheet surface. In order to assure a good quality of products, producers should detect unexpected defect to prevent them to occur continually, and make sure that their products meet the requirements of the users [10, 11].
Application of Parallel Algorithm in Image Processing of Steel Surfaces for Defect Detection

In Mobarakeh Steel Complex of Isfahan, about 210 coded defects have been inspected from the viewpoint of similarity, lighting, features, imaging, and process algorithms. In this paper by considering such parameters as the frequency of defect occurrence, diversity in defect from, and importance of separation among them, 4 types of defects, namely hole, scrape, lateral wrinkle, and corrosion are chosen for inspection (Figure 1).

Hole, Wrinkle, Corrosion, Scrape (From Right to Left)

The dominant technology for detection and classification of objects based upon apparent features is the technology of image processing. Generally speaking, the operation of image processing is performed in 2 steps: feature extraction & classification. In the former, by determining the favored features, parameter selection, and method of extraction, these features are separated from the raw image to be optimized. In the latter, regions with similar texture are recognized and borders between different textures are determined.

Of the most applicable methods used in the extraction of textural features, is the method of directional multi-frequency. In this approach Gabor Wavelet is extensively utilized.

Gabor Wavelet extracts considerable textural features from image which includes both different directions and different frequencies [12, 14]. This Wavelet due to begin optimal in both frequency and local domain, can utilize the benefits of signal processing in both domains [15-18]. If Gabor Wavelet is defined in local domain, it will be convoluted with the respected image and makes partial image. If it is defined in frequency domain, by taking a Fast Fourier Transform (FFT) from respected image, transfers it to frequency domain and then multiplies it by the Gabor Wavelet in that domain. By transferring the product to local domain, partial image is provided. Since the convolution in local domain is performed less quickly than multiplication in frequency domain, in this paper we use Gabor Wavelet in latter domain to extract the image feature faster.

Because most defect in steel surface are not made accidentally and production, a specific Gabor Wavelet may not be used for detection of any kind of defect. Therefore, in this paper we have Utilized Gabor Wavelet bank for feature extraction. Existence of various direction and frequencies in such a bank causes extract features to include lost of information about image texture, so that they can detect any defect in different frequency and direction very well.

The two-dimensional Gabor Wavelet Utilized in this paper is presented by (1).

\[
G(u, v) = e^{-\pi \left[ \frac{u_p^2}{\sigma_u^2} + \frac{v_p^2}{\sigma_v^2} \right]} e^{-2\pi i (u x + v y)}
\]  \hspace{1cm} (1)

In (1), \( u_p \) and \( v_p \) are determined by using (2).

\[
u_p = (u - w_x) \cos(\theta) + (v - w_y) \sin(\theta)
\]

\[
u_p = -(u - w_x) \sin(\theta) + (v - w_y) \cos(\theta)
\]  \hspace{1cm} (2)

269
In (2), \( W_x \) and \( W_y \) are central frequencies of the wavelet in directions X and Y. \( X_0 \) and \( Y_0 \) are horizontal and vertical displacement in local domain, respectively. In the experiments conducted, these values are placed: \( X_0 = 0, Y_0 = 0, W_x = W_y, \sigma_x = \sigma_y \). Since no analytic approaches for optimization of Gabor Wavelet bank have yet been suggested, it is necessary to use a long trend of trial and error to find the best configuration for the Wavelet banks. After inspection of experiments conducted, it was concluded that by using a bank consisting of 12 Gabor Wavelet with frequencies \( \left\{ \Omega_{m2}, \Omega_{m4}, \Omega_{m8} \right\} \) and angles \( \{0^\circ, 45^\circ, 90^\circ, 135^\circ\} \), texture feature can be extracted well. \( \Omega_m \) is the maximum image frequency and equals to half of image dimension m frequency domain. It should be noted that increasing the number of Wavelet above some extent, does not increase the efficiency much, but makes the calculation more complicated.

**RECOMMENDED APPROACH**

In this paper, a new approach is suggested in which classification of defects is based on the calculation of dispersion in partial images. The procedure is as follows: feature extraction by Gabor Wavelet bank is performed with 12 Wavelets. After feature extraction from the defective image, construction of partial image, and calculation of energy, partial images are selected for combination which include defective region distinctly. The category of images in which data dispersion is less, include regions of defect more distinctly, because this region causes the feature of these pixels to be more obvious than the rest parts of image. In images on which defect is not obvious, usually the difference of calculated energy is nearly zero and is not situated in the region of calculated dispersion. With algorithms presented images with lower dispersion are selected. In a defective image it is possible that all partial images are chosen, while in another image none of partial images are selected. In order to combine the chosen partial images by this approach, two methods are used, namely simple addition and Bernoulli combination law. Since the former method gave the best results, in this paper only results achieved by simple addition method is mentioned. In this method, statistical basis for calculation of dispersion in experiments conducted is Variance. In these experiments after feature extraction by Gabor Wavelets bank, constitution of partial images, and calculation of energy, the variance for each partial image is calculated. Experiments are implemented in MATLAB software. To calculate the variance, by this software, (3) was utilized.

\[
\sigma^2 = \frac{1}{N-1} \sum_{n=1}^{N} (x_n - \bar{x})^2
\]

(3)

The resulted variance for all partial images is divided by its maximum value to be situated in similar area in the interval \([0, 1]\).

This is performed because in some partial images, especially those with lower frequency, total value of pixels is less than the rest of partial images. If in this partial image there is an obvious defect, because the amount of its pixels is low, the achieved value for the variance will also be low, which does not have appropriate result. After normalizing the variance values, in order to select partial images for combination with each other, we need a threshold level. If variance is higher than a definite level of threshold, that partial image is chosen for combination. With respect to the images, this threshold level is considered as 0.2 by using the low of simple addition to combine partial images; a feature map is constituted which is a useful tool.

**SET OF DATA AND EXPERIMENTS**

To conduct such an experiment, we need various pictures which include different defects. The pictures used in this experiment consist of 100 real images 55 of which belongs to Kanpur university of India, and the other 45 are pictures taken from Mobarakeh Steel complex. In the
Application of Parallel Algorithm in Image Processing of Steel Surfaces for Defect Detection

experiments of this section, after feature extraction by Gabor Wavelet bank, construction of partial images, and calculation of energy, variance of each partial image is computed. The algorithm for selection of partial images with regions of defect is achieved based on variance value. Obviously, the image situated in the limits of favored variance includes defect regions. In figure 2 an example of corrosion defect is shown. After performing the algorithm of defect detection, figure 3 is achieved.

![Corrosion Defect](image1)

![Result of Applying the New Algorithm](image2)

The results from executing this algorithm are indicated in Table 1.

In these experiments partial images are considered as non-normalized. Variances are normalized by dividing into their maximum value in the interval $[0, 1]$. Threshold level is 0.2 and combination of partial images is performed by simple addition.

**CONCLUSION**

In this paper an approach is suggested to detect the location of defect on the surface of steel sheet by using Gabor Wavelet and Parallel Algorithm. Tasking use of parallel matrix multiplication can lead to speed execution increase on filtering image. Values of partial images are inspected both normalized and non-normalized in the limit of $[0, 1]$. The best elimination by normalizing the resulted variance of partial images, the values of variance changes and inappropriate partial images are selected for combination. All together, the amount of correct detection of defective regions is improved in this approach with respect to other approaches.
RESULT OF APPLYING RECOMMENDED ALGORITHM

<table>
<thead>
<tr>
<th>Image Type</th>
<th>Percentage of Correct Detection</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Correct detection of defective regions (%)</td>
</tr>
<tr>
<td>Real Image of Hole</td>
<td>93.14</td>
</tr>
<tr>
<td>Generated Image of Hole</td>
<td>99.21</td>
</tr>
<tr>
<td>Real Image of Wrinkle</td>
<td>97.13</td>
</tr>
<tr>
<td>Generated Image of Wrinkle</td>
<td>97.43</td>
</tr>
<tr>
<td>Real Image of Corrosion</td>
<td>96.82</td>
</tr>
<tr>
<td>Generated Image of Corrosion</td>
<td>95.35</td>
</tr>
<tr>
<td>Real Image of Scrape</td>
<td>93.84</td>
</tr>
<tr>
<td>Generated Image of Scrape</td>
<td>97.33</td>
</tr>
</tbody>
</table>

REFERENCES


Application of Parallel Algorithm in Image Processing of Steel Surfaces for Defect Detection


