metals

MDPI

Article

An Algorithm for Surface Defect Identification of Steel Plates Based on Genetic Algorithm and Extreme Learning Machine

Siyang Tian [1] and Ke Xu [2,*]

[1] National Engineering Research Center of Advanced Rolling Technology, University of Science and Technology Beijing, Beijing 100083, China; tsycnh@126.com

[2] Collaborative Innovation Center of Steel Technology, University of Science and Technology Beijing, Beijing 100083, China

* Correspondence: xuke@ustb.edu.cn; Tel.: +86-10-6233-2159

Received: 30 June 2017; Accepted: 8 August 2017; Published: 15 August 2017

Abstract: Defects on the surface of steel plates are one of the most important factors affecting the quality of steel plates. It is of great importance to detect such defects through online surface inspection systems, whose ability of defect identification comes from self-learning through training samples. Extreme Learning Machine (ELM) is a fast machine learning algorithm with a high accuracy of identification. ELM is implemented by a hidden matrix generated with random initialization parameters, while different parameters usually result in different performances. To solve this problem, an improved ELM algorithm combined with a Genetic Algorithm was proposed and applied for the surface defect identification of hot rolled steel plates. The output matrix of the ELM's hidden layers was treated as a chromosome, and some novel iteration rules were added. The algorithm was tested with 1675 samples of hot rolled steel plates, including pockmarks, chaps, scars, longitudinal cracks, longitudinal scratches, scales, transverse cracks, transverse scratches, and roll marks. The results showed that the highest identification accuracies for the training and the testing set obtained by the G-ELM (Genetic Extreme Learning Machine) algorithm were 98.46% and 94.30%, respectively, which were about 5% higher than those obtained by the ELM algorithm.

Keywords: surface inspection; steel plate; genetic algorithm; ELM

1. Introduction

Surface defects detection technique is widely applied in industrial scenarios [1]. The surface quality inspection of steel plate has passed through three stages of development, including manual visual inspection, traditional non-destructive testing, and machine vision detection. Artificial visual inspection commonly uses the stroboscopic method [2], which sets up a high-frequency flashing light source above the production lines, and uses the persistence of human vision to achieve high-speed inspection of steel plates. This kind of detection causes great damage to the human body, and could result in optic fatigue as well as a higher false inspection rate. Due to the high-frequency flashing, the workers cannot inspect all parts of the steel surface, so a large number of defects are thus ignored.

The traditional non-destructive detecting techniques include eddy current testing, infrared detection, magnetic flux leakage detection, and laser detection. Since these techniques are limited by the detection principles, only a small number of defect types can be detected. At the same time, the resolution of acquired images is not high enough, so these techniques cannot effectively evaluate the product quality. With the development of computer image processing technology, machine vision with a Charge Coupled Device (CCD) has become widely used in industrial visual inspection. The invention of high-speed CCD cameras also enables fast image acquisition. In the 1980s, some organizations [3] developed online surface defect inspection systems for steel rolling production lines with fast CCD imaging of the steel surface and real-time image processing algorithms. Since then, workers do not have to stare at steel surfaces in the harsh environment of production lines. All they need to do is watch computer screens in an air-conditioned room, and verify the defects reported by the computer. This greatly improves production efficiency and lightens the labor intensity of the workers. Due to insufficient learning ability, early surface inspection systems had high false identification rates of defects. Workers had to identify the real defects based on their experiences of thousands of defects reported by the system. With the development of artificial intelligence, different defect identification algorithms were developed to improve defect identification rates [4].

In 1983, American company Honeywell conducted a survey of a continuous slab surface detecting system [5]. The system used a linear CCD camera to capture images. Specifically, they designed a parallel image processing machine and a classification algorithm based on syntactic pattern recognition theory. This study established the method of inspecting surface defects with a CCD sensor and pattern recognition technology. In 1986, financially supported by American Iron and Steel Institute (AISI), Westinghouse adopted linear CCD cameras and a high intensity linear light source for monitoring steel surface [6], and the method of combining bright-field, dark-field, and glimmer-field illumination was presented to illuminate the steel plates. Horizontal and vertical resolutions were 1.7 mm and 2.3 mm separately at a high rolling speed. At the same period, Centro Sviluppo Materiali in Italy developed a prototype for stainless steel plate surface detecting [7]. This prototype could measure the width of strips and detect pore defects of both sides at the same time. However, few types of defects could be identified. After the 1990s, the automatic defect identification ability of surface inspection systems was improved to a more practical level. Rautaruukki New Technology Company in Finland developed a surface detecting system using machine learning to optimize the decision tree classifier [8]. Cognex Corporation in the United States developed a self-learning classifier named iLearn [9] in 1996, which was applied to its automatic detecting system iS-2000 [10]. This system greatly improved the online detecting speed. Parsytec Company in Germany developed the HTS-2 cold rolling strip surface inspection system [11]. This system adopted an artificial neural network classifier that could detect strip surface defects with a minimum dimension of 0.5 mm under a rolling speed of 300 m/min. In 2000, four sets of HTS-2W surface inspection systems were installed on the hot rolling strip lines of ThyssenKrupp Corporation in Germany [12]. These systems could detect defects on the millimeter scale, and changed the situation of the hot rolling strip line that had previously relied only on manual detection. In 2005, VAI SIAS in France developed the first surface inspection system of hot rolled steel plates for Arcelor Group [13].

Currently, machine learning has been a hot topic in surface inspection area. Much research has focused on developing algorithms for this target, including ANNs (Artificial Neural Networks) [14], SLFNs (Single hidden Layer Feedforward neural networks) [15], CNN (Convolution Neural Networks) [16], SVM (Support Vector Machine) [17], and so on. Over the past two decades, the invention of Artificial Networks has solved many previously unimaginable problems, such as convolution neural network (CNN) for character recognition [16], neural networks for certain object classification [18], and so on. Among the single hidden layer neural networks (SLFNs), BP (Back Propagation) neural network [19] is one of the most representative and most widely used algorithms. However, this method requires many hidden nodes, and the network requires frequent adjustment to converge.

In recent years, a new type of SLFN named Extreme Learning Machine (ELM) [20] achieved great success in training efficiency and accuracy. In ELM, the activation function of the output layer is linear and indefinite, and it only needs a generalized matrix inverse operation to complete network training. Comparing to traditional BP networks, ELM has a greatly reduced training time. It has been demonstrated that, under given conditions [19], given N training samples and any small positive value $\varepsilon > 0$, there exists a number of hidden nodes $\tilde{N}$ ($\tilde{N} \leq N$), and the learning approximation error of the SLFN is less than ε with certainty.

A number of improved ELM-based algorithms have been developed, such as: I-ELM (incremental ELM) [20], OS-ELM (on-line sequential ELM) [21], EI-ELM (enhanced incremental ELM) [22], OP-ELM (Optimally Pruned ELM) [23], EM-ELM (esrror minimized ELM) [24], EOS-ELM (ensemble OS-ELM) [25], and so on. In the field of defect identification, ELM also plays a significant role. Li et al. [26] employed ELM for the identification of glass bottle defects and achieved high identification rates. Zhang et al. [27] applied ELM for the identification of solder joint defects. On the other hand, as the weights and bias of the ELM are randomly chosen, the results are different for each training session. Even if training results are fine enough, the result is not stable. The random weights limit the application of the ELM, that is, the random weights cannot reach the optimal of the current networks. In this paper, an improved ELM algorithm named G-ELM is proposed, which can effectively avoid the instability result caused by randomization with the help of a Genetic Algorithm [28].

The chapters of the paper are organized as follows: Section 2 briefly describes the principles of the ELM algorithm and Genetic Algorithm. Section 3 introduces the G-ELM algorithm, including its principles, elements, and implementation. The advantages of this proposed algorithm are as follows: (1) GA helps to eliminate the instability caused by parameters randomization initialization; (2) some new iterations are added to improve the efficiency of the evolution; and (3) some evolution methods are proposed to speed up converge. In Section 4, the online detecting system is introduced and some types of defect origins are discussed. In Section 5, the original ELM and G-ELM are compared and analyzed experimentally. Section 6 is the summary.

2. ELM and Genetic Algorithm

2.1. ELM Algorithm

The ELM algorithm was proposed by Huang Guangbin [20] as a kind of single hidden layer feedforward neural network (SLFN). The main idea of ELM is stochastic SLFN weight and bias, in which the activation function of the output node is infinitely differentiable. Thus, the optimal solution can be obtained by a generalized matrix inverse operation. The training network takes only a few simple steps and trains very quickly. However, the stochastic weights and bias will result in instability caused by parameters randomization initialization.

Suppose we have M mutually exclusive samples (x_i, y_i), $x_i \in \mathbb{R}^d$, $y_i \in \mathbb{R}$; then an SLFN network with N hidden nodes can be expressed as follows:

$$\sum_{i=1}^{N} \beta_i f(w_i x_i + b_i), j \in [1, M] \tag{1}$$

where f is the activation function, and $w_i,\ b_i,\ \beta_i$ are the input weights, bias, and output weights of the ith neuron nodes of the hidden layer, respectively.

If the SLFN can perfectly predict the data, that is, the difference between the observed value $\hat{y}_i$ and the ground truth y_i is 0, then the above equation can be written as:

$$\sum_{i=1}^{N} \beta_i f(w_i x_i + b_i) = \sum_{i=1}^{N} y_i, j \in [1, M] \tag{2}$$

The equation can be abbreviated as:

$$H\beta = Y \tag{3}$$

where H is the output matrix of the hidden layer, defined as follows:

$$H = \begin{pmatrix} f(w_1 x_1 + b_1) & \cdots & f(w_N x_1 + b_N) \\ \vdots & \ddots & \vdots \\ f(w_1 x_M + b_1) & \cdots & f(w_N x_M + b_N) \end{pmatrix} \tag{4}$$

β as:

$$\beta = (\beta_1 \cdots \beta_N)^T \tag{5}$$

Y as:

$$Y = (y_1 \cdots y_N)^T \tag{6}$$

Given a randomly initialized input layer, as well as the training data $x_i \in \mathbb{R}^d$, the output matrix of the hidden layer H can be calculated. And with H and the target output $y_i \in \mathbb{R}$, we can calculate the output weight β, $\beta = H^{\dagger}Y$, which $H^{\dagger}$ is Moore-Penrose pseudoinverse [29].

In general, the flow of the ELM algorithm is as follows:

Step 1: Given training set (x_i, y_i), $x_i \in \mathbb{R}^d$, $y_i \in \mathbb{R}$, activation function $f : \mathbb{R} \mapsto \mathbb{R}$, hidden nodes N.

Step 2: Randomize initial input weights w_i and bias b_i, $i \in [1, N]$.

Step 3: Calculate the hidden layer output matrix H.

Step 4: Calculate the output weight $\beta = H^{\dagger}Y$.

2.2. Genetic Algorithm

The Genetic Algorithm [26] is an algorithm that simulates the evolution of a species in nature. It is widely applied to optimization and search problems. Inspired by biology, mutation, chromosome crossover, and artificial selection; these and other concepts were applied to the Genetic Algorithm.

The flow of the Genetic Algorithm is as follows:

Step 1: Produce an initial generation G_0.

Step 2: Mutate the initial generation by chromosomal crossing, artificial selection, or other operations, denoted as:

$$G_i = f_m(G_{i-1}), 1 \leq i \leq i_{max} \tag{7}$$

Step 3: Termination condition of G_i

$$f_t(G_i) < \varepsilon \tag{8}$$

Step 4: If step 3 is established, then the calculation is completed, otherwise return to step 2 to continue to produce offspring.

Step 5: If the maximum number of iterations is reached, the algorithm is exited. In the above process, G_i represents the ith generation, f_m represents mutation function, i_{max} represents the maximum number of iterations, f_t represents the fitness function, and ε is a customized small value which is greater than 0.

There are two important aspects of the Genetic Algorithm. One is the mutation operator to generate offspring; the other is the fitness operator to determine which offspring is to survive.

3. G-ELM

The input weights and bias of ELM are combined together to form a input matrix, such as $X = [w, b]$. The matrix X is considered as a single individual or a chromosome, and each element in it is a minimum mutation unit. Taking the characteristics of the matrix X into account, some operations are not compatible to this problem, such as crossover.

Mutation is a key point in this algorithm; to increase the correctness and success rate of the mutation, some novel mutation rules are added to the G-ELM which could produce new offspring more effectively.

3.1. The G-ELM Procedure

The process is as follows:

Given hidden matrix nodes number $\widetilde{N}$, training samples $N\left(\widetilde{N} \leq N\right)$ and normally $\widetilde{N} \ll N$. A single training sample (x_i, y_i), $x_i \in \mathbb{R}^d$, $y_i \in \mathbb{R}$, d is the number of features of the sample. ELM randomizes an initial input weight matrix $iw = \text{Matrix}\left(\widetilde{N} \times d\right)$, while a column vector

$b = \text{Matrix}\left(\widetilde{N} \times 1\right)$ is also randomized. The weight matrix iw is combined with bias b to obtain the initial parent $G = [iw,\ b]$.

The algorithm is initialized m times, so m initial parents are obtained, named the initial parents group. The fitness function is defined as:

$$f_{f\text{itness}}(G_0) = \sum_{i=1}^{N} |y_i - \hat{y}_i| \tag{9}$$

in which $\hat{y}_i$ is the ground truth.

Then set ε and the maximum iteration limit k_{max}. if $f_{f\text{itness}}(G_0) < \varepsilon$ or $k = k_{max}$. Then terminate the training, otherwise enter the evolution iteration. When one generation of the elements of G_k change from one status to another, it is called a mutation operation. Every time a mutation operation proceeds, certain rules need to be followed.

Suppose the kth generation is G_k, and the elements in the matrix are $g_{i,j}$ $(1 \le i \le \widetilde{N}, 1 \le j \le d+1)$. There are three steps in the mutation operator O_m:

Step 1: Choose all selectable elements, excluding the locked ones.

Step 2: Determine the number of variation elements according to the mutation rate v_t.

Step 3: Determine the new element value $\overline{g_{i,j}}$ according to the variation fluctuation range r.

Step 4: Mutate m times with different randomized parameters.

Step 5: Choose the model of smallest fitness among m models as the candidate for the generation $k + 1$.

Step 6: Check if this model is performs better than the model of smallest fitness and, if so, choose it to be G_{k+1}, otherwise give up the current generation of groups and restart from G_k to produce a new set of offspring.

Step 7: Cycle the above steps until the training condition is reached or the maximum number of iterations is reached.

The detailed procedure is shown in Figure 1.

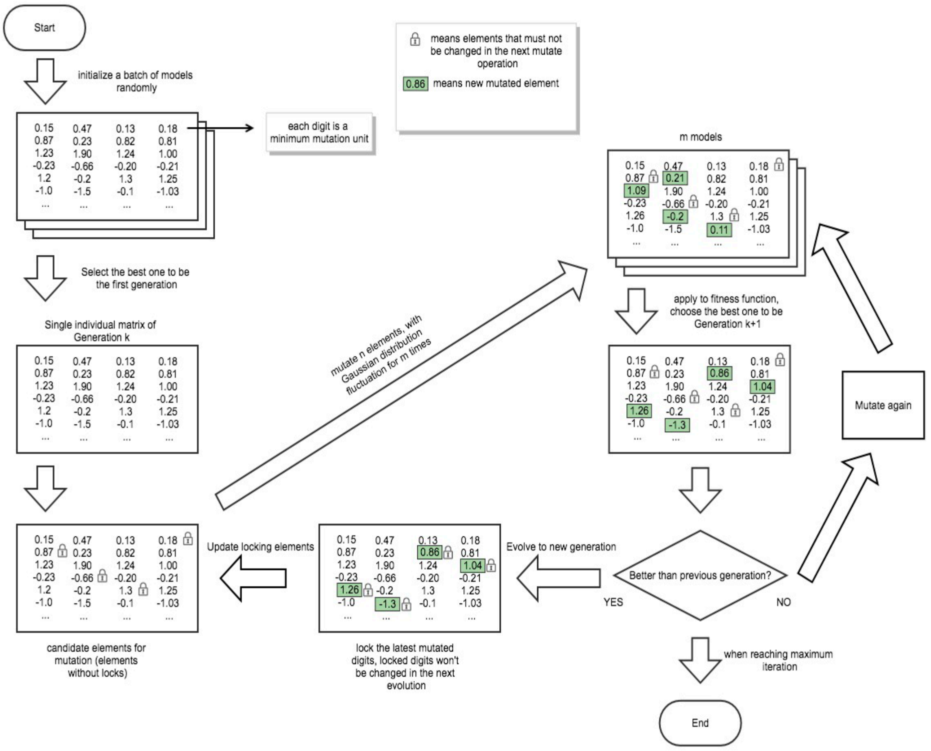
Start
initialize a batch of models randomly
each digit is a minimum mutation unit
🔒 means elements that must not be changed in the next mutate operation
0.86 means new mutated element
m models
Select the best one to be the first generation
Single individual matrix of Generation k
mutate n elements, with Gaussian distribution fluctuation for m times
apply to fitness function, choose the best one to be Generation k+1
Mutate again
Update locking elements
Evolve to new generation
Better than previous generation?
YES
NO
candidate elements for mutation (elements without locks)
lock the latest mutated digits, locked digits won't be changed in the next evolution
when reaching maximum iteration
End

Figure 1. Flow chart of the G-ELM (Genetic Extreme Learning Machine) procedure.

3.2. Mutation Operation Rules

In order to speed up the evolution rate, some new mutation methods are proposed, including superior gene selection, dynamic variability, and regional mutation.

(1) Superior gene selection:

Whenever a new offspring is generated, it is assumed that all the elements of variation become an element group M. If it is obvious that element group M is the main reason for this successful mutation, then in order to preserve these "better genes", the next generation of variation elements will not contain the elements in element group M. Experimentation shows that this practice can greatly improve the efficiency of the algorithm.

(2) Dynamic variability:

Set the value of base variability v_b, the highest variability v_h, and variation step rate v_s. For each iteration t, the mutation rate v_t is defined as:

$$\begin{cases} v_t = v_{t-1} + v_s(t \geq 1,\ v_t \leq v_h) \\ v_0 = v_b \\ v_t = v_b(v_t > v_h) \end{cases} \tag{10}$$

When one generation reaches the bottleneck of evolution, mutation rate v_t will be changed initiatively to improve the success rate of evolution. However, a too high mutation rate, such as 0.7, will also make it difficult to mutate successfully. Furthermore, the mutation rate will be gradually increased until the highest rate v_h is reached, after which it will be returned to the base mutation rate v_b so as not to fall into a local optimal. In general, the training accuracy will be improved in a limited number of iterations.

4. Surface Inspection and Defects of Hot Rolled Steel Plates

4.1. Surface Inspection of Hot Rolled Steel Plates

There are three main procedures of the steel production industry, including continuous casting, hot rolling, and cold rolling. In every procedure, the surface status of steel products varies with different features. This paper is focused on the surface defect identification of hot rolled steel plates, and all samples were collected from several surface inspection systems installed on hot rolling steel plate lines, which were developed by Xu et al. [30]. Due to the hostile environment of hot rolling lines, the contrast and sharpness of samples are often not very good.

The system is installed after the hot straighter, and the surface temperatures of the steel plates are about 600 °C–800 °C. As demonstrated in Figure 2, the hot rolled steel plates are illuminated with green linear laser lighting. The wavelength of the lasers is 532 nm, which is very far from the spectrum of high temperature radiation. Furthermore, a narrow-banded color filter with a central bandwidth of 532 nm is installed on the front of each camera lens, and only laser lights with the wavelength of 532 nm reflected by the steel plates are allowed to enter into the cameras. Then, the surface of high temperature steel plates is imaged with high quality, and defects are visible in the images. In a surface inspection system for a 5000 mm steel plate production line, eight line-scanning CCD cameras with 4096 pixels are used, four of which are for image acquisition of the top side of the steel plates, while the other four cameras are for the bottom side. Each camera view is 1200 mm, and four cameras can cover 4700 mm of the width, subtracting overlaps between two cameras. The resolution of images in the width direction is $1200/4096 \approx 0.3$ mm/pixel. The resolution of images in the height direction is the distance between two adjacent lines captured by the camera. To keep images in the same resolution for the width direction and height direction, the distance between two adjacent lines captured by the camera is also 0.3 mm. A rotator is installed on the roller to acquire the real-time speed of the production line, and all cameras of the system are triggered once every 0.3 mm by the rotator. As the height of an image is 1024 pixels, the length of plate covered by an image is $1024 \times 0.3 \approx 307$ mm.

Normally, a hot rolled steel plate is about 25 m long, and about 8 × 25,000/307 ≈ 648 images are needed to cover the whole length and width of both sides of the steel plate.

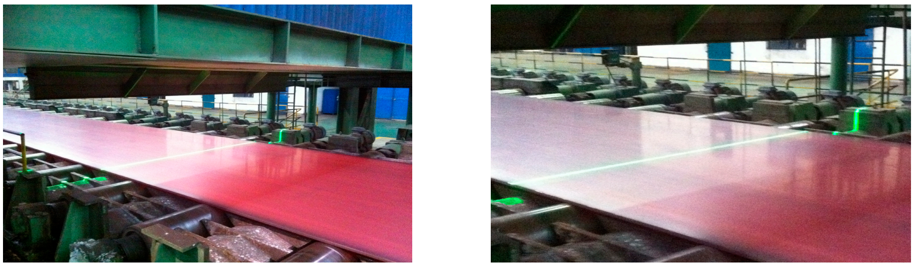

Figure 2. Surface inspection system of hot rolled steel plates with green linear laser lighting.

4.2. Surface Defects of Steel Plates

Figure 3 shows nine kinds of typical defects of hot rolled steel plates, which are pockmark (Figure 3a), chap (Figure 3b), scar (Figure 3c), longitudinal crack (Figure 3d), longitudinal scratch (Figure 3e), scale (Figure 3f), transverse crack (Figure 3g), transverse scratch (Figure 3h), and roll mark (Figure 3i).

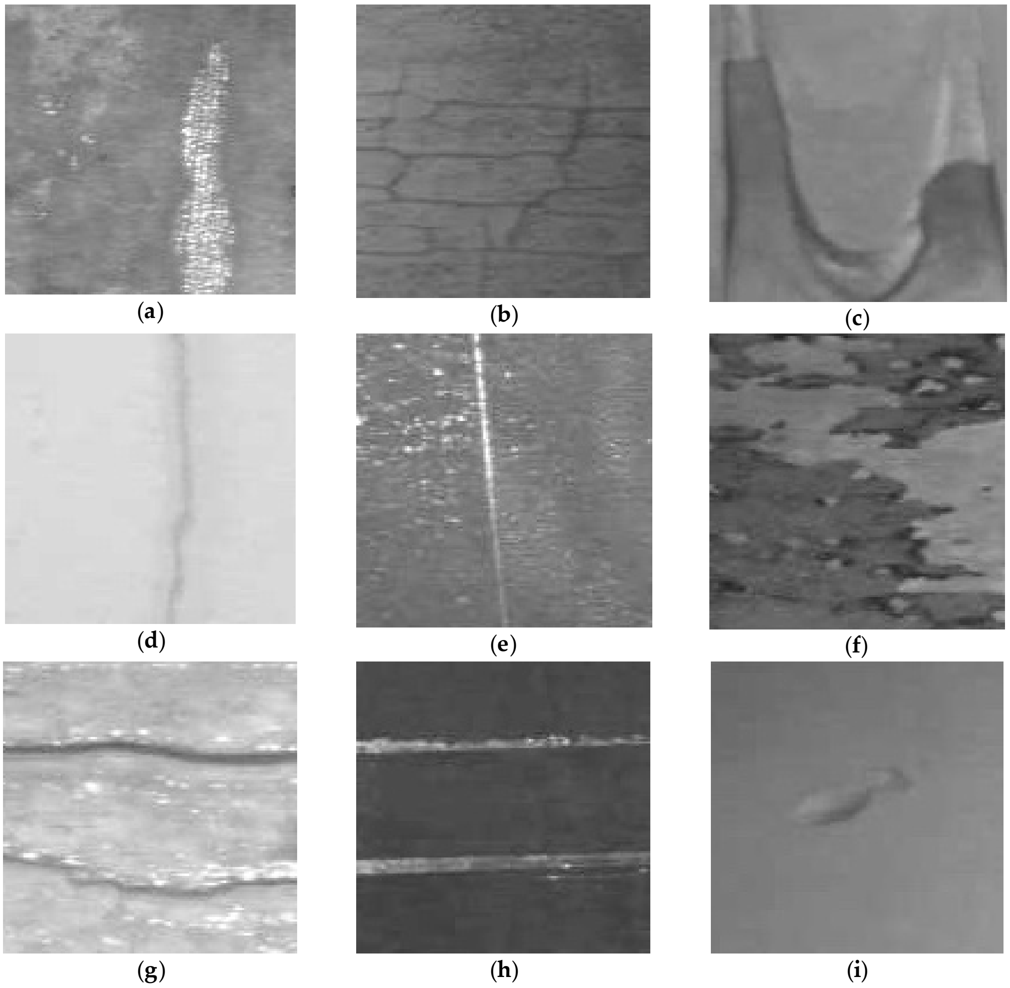

Figure 3. Nine kinds of typical defects of hot rolled steel plates: (**a**) pockmark; (**b**) chap; (**c**) scar; (**d**) longitudinal cracks; (**e**) longitudinal scratches; (**f**) scale; (**g**) transverse cracks; (**h**) transverse scratches; (**i**) roll mark.

One of most common defects of hot rolled steel plates is scratch, including longitudinal scratch (Figure 3e) and transverse scratch (Figure 3h). Scratches are usually caused by contact with hard objects or corners, or by the relative movement of steel plates and rollers resulting from a change in velocity.

Another frequent kind of defect is scale (Figure 3f). Some scales are covered on the steel surface, while some are rolled in the steel plates. Identification of scales is very difficult because of their diverse shapes and distributions.

Roll mark (Figure 3i) is another common kind of defect. They are periodical, as they are caused by foreign matter and pits in the rollers. By calculating the cycle of roll marks, the problem roller can be determined.

5. Experiments

The dataset uses 1675 samples of hot rolled steel plates, which are collected with the surface inspection system of a 5000 mm hot rolling line. Of these, 836 samples are used as a training set, while another 839 samples are used as a testing set. During each experiment, only the training samples are used to optimize the model. There are much more testing samples than training samples (normally 10% of total samples) to testify the generalization of the model. As illustrated in Section 4.1, there are nine types of defects in the dataset, including pockmarks, chaps, scars, longitudinal cracks, longitudinal scratches, scales, transverse cracks, transverse scratches, and roll marks. The image size is 128 × 128 pixels. The feature extraction procedure is carried out by the original Local Binary Pattern (LBP) operator [31], and the feature vector is 256 dimensions. All features of the training samples consist of a matrix of 836 × 256. The algorithm was developed with Matlab (R2016b, Version 9.1, MathWorks Inc., Natick, MA, USA), and operated in a MacBook Air computer (version 2011, Apple Inc., Cupertino, CA, USA) (CPU 1.4 GHz Intel Core i5, memory 4 GB 1600 MHz).

5.1. Comparison between ELM and G-ELM

In order to verify the efficiency of the proposed G-ELM, the training set is also imported into the ELM model. ELM and G-ELM algorithms both use 200 hidden nodes. The number of G-ELM iterations is 1000. The first-generation individual is selected from the best 20 random ELM models. Table 1 gives the experimental results of the G-ELM algorithm, which are compared with the ELM algorithm.

Table 1. Experiments result of ELM (Extreme Learning Machine) and G-ELM (Genetic Extreme Learning Machine).

Algorithm	v_b	v_h	Training Set (%)	Testing Set (%)
ELM	-	-	93.32	89.12
G-ELM	0.001	0.003	94.98	89.39
	0.003	0.01	95.81	90.35
	0.01	0.03	96.14	91.29
	0.03	0.1	96.92	91.71
	0.1	0.3	98.46	94.30
	0.3	0.7	95.69	90.23
	0.6	0.9	94.74	89.27

In Table 1, v_b and v_h are the minimum and maximum boundary of the evolution mutation rate. Compared to the results of ELM, the improved G-ELM has better accuracy for both the training set and testing set because of its additional mutation operation rules. For example, because of v_b and v_h, the worst accuracies for the training set and testing set by G-ELM are 94.98% and 89.93%, which are 1.78% and 0.30% higher than that of ELM, respectively. Moreover, note that the accuracies for both the training set and the testing set are high, which means that the generalization of G-ELM is high enough.

For G-ELM, with the increase of the value of v_b and v_h, the accuracies of the training set and testing set continue to increase until the maximum values are reached, after which the values decrease, as shown in Figure 4. In Table 1, when the value of v_b increases from 0.001 to 0.1, the accuracy for the training set improves from 94.98% to 98.46%, which is the highest accuracy. However, the accuracy decreases from 98.46% to 94.74% when v_b increases to 0.6. This can be explained by the fact that the lower the mutation rate, the fewer the number of mutation elements, which makes it more difficult to mutate successfully. When the mutation rate is too high, there become too many mutation elements, causing the loss of superior genes. Only an appropriate mutation rate and reasonable mutation rules can achieve a higher rate of successful mutation and preserve effective genes.

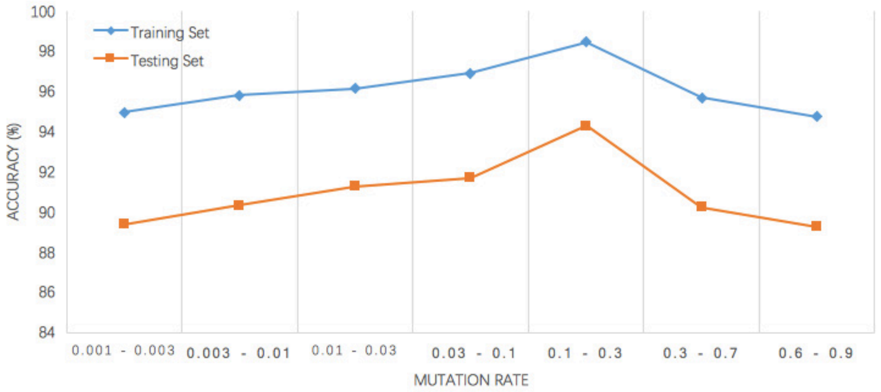

Figure 4. Accuracies of training set and testing set of ELM and G-ELM.

5.2. Analysis of the Perfomance of G-ELM

In this section, the performance of G-ELM is discussed, including the training history and the number of iterations. Figure 5 shows the training history of the training set and testing set at v_b = 0.1, v_h = 0.3, in which there are 10 successful evolutions after about 7000 iterations in total. It can be seen that during the 10 generations, the accuracy of the training set increases more steadily than that of the testing set. Moreover, in the 10th generation, the accuracies of both the training set and testing set are the highest, meaning that the model is optimal. For the first six generations, the accuracy of the testing set rises rapidly. However, the accuracy of the testing set fluctuates significantly from the sixth generation to the 10th generation because the generalization ability is weakened. Therefore, the accuracy at the 10th generation is optimal.

Note that when the number of generations increases, it becomes harder to achieve successful evolution. In Table 2, the larger the number of generations, the more iterations are needed. For instance, in the eighth generation, there needs to be 830 iterations, while 5690 iterations are required for the generation. Secondly, when the number of generations is very large, it is very difficult to achieve evolution. With the increase of the number of generations, the accuracies of the training set and testing set become higher, since it is closer to the optimal solution. From the first generation to the 10th, the accuracy of the testing set increases from 90.93% to 94.43%. Furthermore, in Table 2, regarding the training set, some accuracies are the same. For example, the accuracy of the seventh generation is the same as that of the fifth and ninth generations, because the generalization ability of the model is instable, especially when the number of iterations is high.

There is an increase of time consumption, especially from the seventh to 10th generation, and its CPU time increases from 7.35 s to 250.95 s. Under certain circumstances, such as online detection, there is not enough time to search for the global optimal, thus limited iterations are also acceptable. In this case, the seventh generation is a highly cost-effective solution based on accuracy and time.

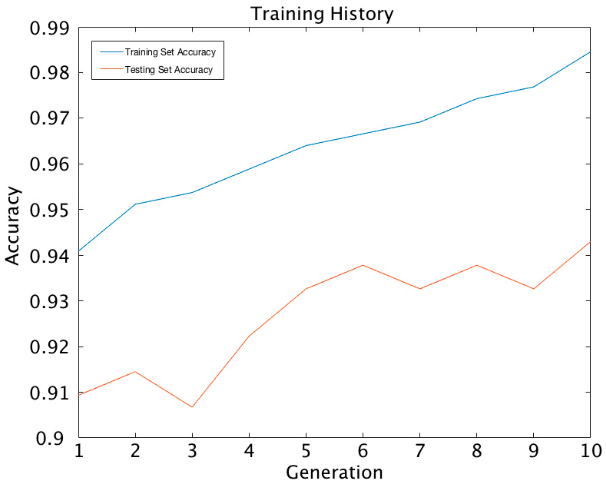

Figure 5. G-ELM Training History at v_b = 0.1, v_h = 0.3.

Table 2. Iterations for each generation of conditions v_b = 0.1, v_h = 0.3.

Generation	1	2	3	4	5	6	7	8	9	10
Iterations	-	10	50	60	170	200	210	830	5690	7170
Training Accuracy (%)	94.09	95.12	95.37	95.89	96.40	96.66	96.92	97.43	97.69	98.46
Testing Accuracy (%)	90.93	91.45	90.67	92.23	93.26	93.78	93.26	93.78	93.26	94.30
TCPU (s)	0.035	0.35	1.75	2.1	5.95	7	7.35	29.05	199.15	250.95

6. Conclusions

An improved ELM algorithm named G-ELM was proposed and applied for the defect identification of steel plates. The G-ELM algorithm employs some additional mutation rules, which can offset the uncertainties caused by ELM randomization. Moreover, the G-ELM algorithm has a large gap between the different mutation rate training results. Results of experiments with nine typical defect samples showed that the G-ELM algorithm effectively improved the identification accuracy of the ELM algorithm. Under conditions of v_b = 0.1 and v_h = 0.3, the G-ELM algorithm performs best. The highest identification accuracy of the training and testing set obtained by the G-ELM algorithm are 98.46% and 94.30% respectively, which are about 5% higher than that obtained by the ELM algorithm.

Acknowledgments: This work is sponsored by The National Natural Science Foundation of China (No. 51674031).

Author Contributions: Siyang Tian conceived, designed and performed the experiments; Ke Xu contributed experiment data, materials and experiment equipments; Siyang Tian and Ke Xu analyzed the data; Siyang Tian wrote the paper; Ke Xu revised the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Xu, K.; Xu, J.W.; Chen, Y.L. On-line Inspection System for Surface Defects of Cold Rolled Strip. *J. Univ. Sci. Technol. Beijing* **2002**, *24*, 329–332.
2. Qin, D.G.; Xu, K. Study on the On-line Detection Technology of Continuous Casting Slab Surface "Trajectory". *Met. World* **2010**, *5*, 13–16.
3. Ai, Y.H.; Xu, K. Development and Prospect of Online Detection Technology for Steel Plate Surface. *Met. World* **2010**, *5*, 37–39.
4. Han, B.A.; Xiang, H.Y.; Li, Z.; Huang, J.J. Defects Detection of Sheet Metal Parts Based on HALCON and Region Morphology. *Appl. Mech. Mater.* **2013**, *365–366*, 729–732. [CrossRef]
5. Suresh, B.R.; Fundakowski, R.A.; Levitt, T.S. A real-time automated visual inspection system for hot steel slabs. *IEEE Trans. Pattern Anal. Mach. Intell.* **1983**, *6*, 563–572. [CrossRef]

6. Jouetetal, J. Defect Classification in surface inspection of strip steel. *Steel Times* **1992**, *16*, 214–216.
7. Canella, G.; Falessi, R. Surface inspection and classification plant for stainless steel strip. *Nondestruct. Test.* **1992**, *72*, 1185–1189.
8. Badger, J.C.; Enright, S.T. Automated surface inspection system. *Iron Steel Eng.* **1996**, *73*, 48–51.
9. Rodrick, T.J. Software controlled on-line su–rface inspection. *Steel Times Int.* **1998**, *22*, 30.
10. Carisetti, C.A.; Fong, T.Y.; Fromm, C. Selflearning defect classifier. *Iron Steel Eng.* **1998**, *75*, 50–53.
11. Parsytec Computer Corp. Software controlled on 2 line surface inspection. *Steel Times Int.* **1998**, *22*, 30.
12. Ceracki, P.; Reizig, H.J.; Rudolphi, U.; Lucking, F. On-line surface inspection of hot-rolled strip. *Metall. Plant Technol. Int.* **2000**, *23*, 66–68.
13. Bailleul, M. Dynamic surface inspection at the hot-strip mill. *Steel Times Int.* **2005**, *29*, 24.
14. Jordan, M.I.; Bishop, C.M. Neural networks. *ACM Comput. Surv.* **1996**, *28*, 73–75. [CrossRef]
15. Vishwakarma, V.P.; Gupta, M.N. A New Learning Algorithm for Single hidden Layer Feedforward Neural Networks. *Int. J. Comput. Appl.* **2011**, *28*, 26–33. [CrossRef]
16. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
17. Cortes, C.; Vapnik, V. Support-Vector Networks. *Mach. Learn.* **1995**, *20*, 273–297. [CrossRef]
18. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2012; pp. 1097–1105.
19. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning representations by back-propagating errors. *Nature* **1986**, *323*, 533–536. [CrossRef]
20. Huang, G.; Zhu, Q.; Siew, C. Extreme learning machine: Theory and applications. *Neurocomputing* **2006**, *70*, 489–501. [CrossRef]
21. Liang, N.Y.; Huang, G.B.; Saratchandran, P.; Sundararajan, N. A fast and accurate online sequential learning algorithm for feedforward networks. *IEEE Trans. Neural Netw.* **2006**, *17*, 1411–1423. [CrossRef] [PubMed]
22. Huang, G.B.; Chen, L.; Siew, C.K. Universal approximation using incremental constructive feedforward networks with random hidden nodes. *IEEE Trans. Neural Netw.* **2006**, *4*, 879–892. [CrossRef] [PubMed]
23. Miche, Y.; Sorjamaa, A.; Lendasse, A. OP-ELM: Theory, experiments and a toolbox. *Artif. Neural Netw.* **2008**, *5163*, 145–154.
24. Feng, G.; Huang, G.B.; Lin, Q.; Gay, R. Error minimized extreme learning machine with growth of hidden nodes and incremental learning. *IEEE Trans. Neural Netw.* **2009**, *20*, 1352–1357. [CrossRef] [PubMed]
25. Lan, Y.; Soh, Y.C.; Huang, G.B. Ensemble of online sequential extreme learning machine. *Neurocomputing* **2009**, *72*, 3391–3395. [CrossRef]
26. Goodman, E.D. Introduction to genetic algorithms. In Proceedings of the Conference Companion on Genetic and Evolutionary Computation, Dublin, Ireland, 12–16 July 2011; pp. 839–860.
27. Zhang, C.; Liu, H.; Zhang, C. The Detection of Solder Joint Defect and Solar Panel Orientation Based on ELM and Robust Least Square Fitting. In Proceedings of the Chinese Control and Decision Conference (CCDC), Mianyang, China, August 2011; pp. 561–565.
28. Li, S.X.; Huang, Z.H. Research on the detection method of glass mouth defect based on extreme learning machine. *J. Computing Technol. Autom.* **2016**, *4*, 117–120.
29. Barata, J.C.; Hussein, M.S. The Moore-Penrose Pseudoinverse: A Tutorial Review of the Theory. *Braz. J. Phys.* **2011**, *42*, 146–165. [CrossRef]
30. Xu, K.; Zhou, P.; Yang, C.L. Online Detection Technology of Surface Defects in Continuous Casting Slab and Its Application. In Proceedings of the Scientific and Technological Progress and Fine Arts Symposium on Continuous Casting Equipment Technology, Xi’an, China, May 2013; pp. 1195–1200.
31. Ojala, T.; Pietikainen, M.; Maenpaa, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 971–987. [CrossRef]