IOP Conference Series: Materials Science and Engineering

PAPER • OPEN ACCESS

An enhanced real-time steel band wrinkling detection algorithm based on homomorphic filter and boost decision tree

IOP Conf. Series: Materials Science and Engineering **493** (2019) 012005 doi:10.1088/1757-899X/493/1/012005

An enhanced real-time steel band wrinkling detection algorithm based on homomorphic filter and boost decision tree

Sheng Xu [a], **Xiong Chen** [b, *]

School of Fudan University, Shanghai 200433, China

[a] 17210720045@fudan.edu.cn,
[*, b] Corresponding author email: chenxiong@fudan.edu.cn

Abstract. Strip wrinkling is a kind of recurrent defect in metal production, which leads to the waste of raw material and the decline of production efficiency. The way to reduce loss is to detect this kind of defects as soon as possible. The computer vision algorithms are often used in this kind of situations, but the specific challenge we have to face in this problem is that the steel band is reflective to flash light. This causes the unevenness of exposure in the images we analyze. In order to solve this problem, we propose a method combining homomorphic filtering and region proposal based GBDT algorithm. Our method receives a satisfactory result within both accuracy and detection speed.

Keywords: Strip wrinkling, Homomorphic filter, GBDT.

1. Introduction

The cold-strip steel is an important material which is widely used in different departments. In the production of strip steel, the strip steel needs to keep smooth and flat. However, in the actual production, the non-uniform surface tension caused by several complex reasons is likely to form the defection called longitudinal curve. When the strip passes through the rolling convexity roll, the irreversible strip wrinkling is easy to happen because of plastic deformation. Another possible cause of strip wrinkling is called off-tracking. Due to the horizontal disturbance, the strip will be deviated from its center line and causes the phenomenon of off-tracking. When off-tracking happens, it’s likely to exacerbate the strip wrinkling, even causes the strip to be fractured.

To the consideration of cost saving and quality control, it’s important to detect the strip wrinkling as soon as possible. In some production field, people choose to apply manual vision inspection to find out whether strip wrinkling occurs. But this comes with the problem of expensive human cost. Besides, the human detection relies on subjectivity judgement, which means this method is often not robust.

According to the disadvantages enumerated above, some works have been done to realize the automatic and real time monitoring on strip wrinkling. Pan [1] introduced an operation based on AIM saliency model. The algorithm tracks the movement of edges in sequential images. Wang [2] introduces a feature-based image processing method. Other methods include neural network [3]. These methods all provide constructive solution for strip wrinkling detection. But they are either not real-time or not good enough in accuracy. To solve these two problems, we must get images with higher degree of distinction.

In our paper, we take the non-uniform luminance into consideration. Considering the actual working condition, the production line is under a low-light level environment. To get the clear image, we often use flashlight as the supplementary lighting. Unfortunately, the strip steel is light-reflective. This feature causes the images captured to have extremely intensive variance in pixel value. Some sections are extremely bright while some sections are so dark that we can almost recognize no detail. It also leads to serious disturbance to the classification afterwards. In this paper, we will first apply homomorphic filter to solve the uneven illumination problem. Then we use the gradient boost decision tree as our classification algorithm. Our algorithm outperforms in both accuracy and computation time.

2. Methodology

2.1. Framework of the proposed method

The aim of our algorithm is to detect the defect fast and accurately. Considering the uneven light and shade of the image, we first pass the image through the homomorphic filter. Then we apply the Gradient Boost Decision Tree (GBDT) to generate the classifier model. The following images can be classified quickly by this model. The flow chat is shown in figure 1.

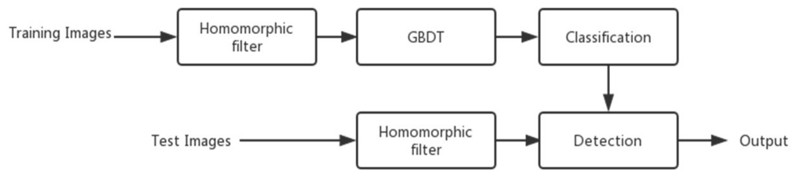


Figure 1. Framework of the proposed method

2.2. Principal of homomorphic filter

Homomorphic filter is a kind of nonlinear filter. This filter can remove multiplicative noise and is therefore effective for images that have uneven illumination. This is exactly the situation we are facing with. The images we have to classify has the characteristic of huge dynamic range, which means part of the image has similar high pixel value, while some other parts may have similar low pixel value. The bright part of the image will have similar high pixel values that we can almost see nothing. While the dark part will be close to black and lose a lot of detail. The pixel values of the interested part will be varying in a large scale and concentrate in both ends. This leads to the difficulty for distinguishing the details of the image both for human visual system and machine learning algorithms. The traditional gray level linear transformation methods are not useful. Extending grayscale can strengthen the contrast, but it will make the dynamic range larger. Compressing the grayscale can decrease the dynamic range but also make the details more difficult to distinguish. The homomorphic filtering is an effective enhancement algorithm for this kind of uneven illumination distribution problem. The filter can compress the grayscale while keeping details.

In our model, we consider the digital image as the product of illumination and reflection:

$$s(x,y) = s_i(x,y)s_r(x,y) \tag{1}$$

In equation (1), $s(x,y)$ is the origin image. $s_i(x,y)$ is the illumination component and $s_r(x,y)$ is the reflection component. We have the prior assumption that $0 < s_r(x,y) < 1$ and $0 < s(x,y) < s_i(x,y) < \infty$. The illumination component has small variance, which means it is composed most by low frequency components. On the other hand, the reflection is mainly composed of high frequency components, which is the detail of the image. We want to keep details meanwhile eliminate the influence of illumination intensity, which makes our goal extracting the reflection from the image. The image after passing the homomorphic filter conforms to human nonlinear response to luminance. This homomorphic filter is also effective for our classifier.

In order to apply high pass filter, we have to transform the image to the frequency domain. The usual signal and noise we want to separate is linear combined, but this time the two components are multiplied. It is not convenient to operate on illumination and reflectance respectively, so we use the logarithmic operation to change the product expression into additive expression:

$$ln[s(x,y)] = ln[s_i(x,y)] + \ln[s_i(x,y)] \tag{2}$$

The image is expressed as the linear combination of two parts after the logarithm. Then we can apply the Fourier transform to equation (2):

$$\ln\big(s(x,y)\big) = \ln\big(s_i(x,y)\big) + \ln\big(s_r(x,y)\big) \tag{3}$$

We mark the three parts of (3) as $M(x,y)$, $I(x,y)$ and $R(x,y)$:

$$M(x,y) = I(x,y) + R(x,y) \tag{4}$$

The high-pass filter is applied to $M(x,y)$:

$$N(x,y) = H(x,y)\, M(x,y) \tag{5}$$

The $N(x,y)$ lies in frequency domain. We have to use the Fourier inversion to map the signal back to spatial domain for the following work.

$$n(x,y) = inv\mathcal{F}(N(x,y)) \tag{6}$$

The wanted reflection component $\hat{s}_r(x,y)$ is recovered after exponent:

$$\widehat{s_r}(x,y) = e^{n(x,y)} \tag{7}$$

2.3. Principal of Gradient Boosting Decision Tree

After passing through the homomorphic filter, we try Gradient boost decision Tree (GBDT) [4] as our classification algorithm. GBDT is widely applied in regression and classification, used in recommendation system [4], traffic prediction [5], flight traffic [6]. But we apply GBDT to image classification for that we notice that the wrinkling only appears in specific locations, as shown in figure 1:

(a) (b)

Figure 2. Wrinkling positions in images

The relationship among particular pixels can identify whether it's positive or negative, so our image classification is changing to extracting the dependence among specific pixels, so the GBDT is workable for our problem.

Gradient boosting decision tree is a kind of iterative decision tree algorithm. For supervised learning, if we suppose that there are N training samples:

$$\{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\} \tag{8}$$

Each x_i can be represent as $x_i = \{x_i^{(1)}, x_i^{(2)}, \dots, x_i^{(n)}\}$, for there are n characteristics in one training sample. We want to find a function that makes the map of x_i close to y_i as much as possible:

$$x_i \xrightarrow{F} y_i \tag{9}$$

It is hard to use only one decision tree to match y_i well, so GBDT uses multiple trees to match the labels iteratively. When coming to the single decision tree, we choose CART algorithm as our principle to generate the decision trees. We recursively divide the input space into two parts by choosing the optimal characteristic and splitting point subjected to:

$$\min_{j,s}\left[\min_{c_1}\sum_{x_i\in R_1(j,s)}(y_i - c_1)^2 + \min_{c_2}\sum_{x_2\in R_2(j,s)}(y_i - c_2)^2\right] \tag{10}$$

The region $R_1(j,s)$ and $R_2(j,s)$ are decided by the splitting variable and splitting point: $\mathrm{R}_1(j,s) = \{x|x^{(j)} \le s\}$,

$$\mathrm{R}_2(j,s) = \{x|x^{(j)} > s\} \tag{11}$$

Then we will use the chosen pair (j, s) to decide the correspond output:

$$\hat{\mathrm{c}}_{\mathrm{m}} = \frac{1}{N_m}\sum_{x_i\in R_m(j,s)} y_i\,, \mathrm{x} \in \mathrm{R}_{\mathrm{m}}, m = 1,2 \tag{12}$$

We continuously apply (10), (12) to each sub region to until it satisfies the suspensive condition. We use the divided M regions to generate our decision tree:

$$f(x) = \sum_{m=1}^{M} \hat{c}_m I(x \in R_m) \tag{13}$$

The GBDT algorithm is an additive model. Each new tree is decided both by the training set and the former trees. It trains the residual error of the last turn to approach target. The structure of GBDT could be shown as figure 2:

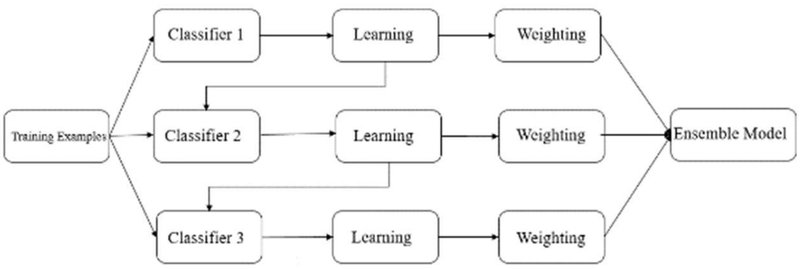


Figure 3. Structure of GBDT

The final model can be represented as:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), f_k \in F \tag{14}$$

F is the function space spanned by the K decision trees. The learning objects we aim at are no longer weights, but the decision trees. The loss function we take into consideration is not only the fitting error, but also the complexity of the decisions:

$$L = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{15}$$

Function l is the loss function which can be taken as the square error:

$$l(y_i, \hat{y}_i) = (y_i - \hat{y}_i)^2 \tag{16}$$

Function Ω represents for the complexity of the decision tree. A normal choice is to use the l_1 norm as judgement of the complexity of the tree:

$$\Omega(f_k) = \alpha |T_k| \tag{17}$$

$|\mathrm{T_k}|$ is the number of leaf nodes. The additive model is learned by forward progressive algorithm, generating one tree by one step:

$$\begin{aligned} \hat{y}_i^1 &= 0 \\ \hat{y}_i^1 &= f_1(x_i) = \hat{y}_i^0 + f_1(x_i) \\ \hat{y}_i^2 &= f_1(x_i) + f_2(x_i) = \hat{y}_i^1 + f_2(x_i) \\ \hat{y}_i^t &= \sum_{k=1}^{t} f_k(x_i) = \hat{y}_i^{t-1} + f_t(x_i) \end{aligned} \tag{18}$$

In the tth step, $\hat{y}_i^t$ is predicted by the last prediction and the current decision tree, $\hat{y}_i^t = \hat{y}_i^{t-1} + f_t(x_i)$, the decision tree $f_t(x_i)$ learned this turn is subjected to the target function:

$$\begin{aligned} L^t &= \sum_{i=1}^{N} l(y_i, \hat{y}_i^t) + \sum_{j=1}^{t} \Omega(f_i) \\ &= \sum_{i=1}^{N} l\left(y_i, \hat{y}_i^{t-1} + f_t(x_i)\right) + \Omega(f_t) + constant \end{aligned} \tag{19}$$

We do the second order Taylor expansion to (19) and get:

$$L^t = \sum_{i=1}^{n} [l\left(y_i, \hat{y}_i^{t-1}\right) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t) + constant \tag{20}$$

We choose the square loss in (16) as our loss function l, then g_i is the first derivative of loss function:

$$g_i = \frac{\partial l(y_i, \hat{y}_i^{t-1})}{\partial y^{t-1}} \tag{21}$$

h_i is the second derivative of loss function:

$$h_i = \frac{\partial^2 (y_i, \hat{y}_i^{t-1})}{\partial (y^{t-1})^2} \tag{22}$$

The constant terms are removed for the reason that they have no influence on the minimization of loss function. The loss function can be now written as:

$$L^t = \sum_{i=1}^{n}\left[g_i f_t(x_i) + \frac{1}{2}h_i f_t^2(x_i)\right] + \Omega(f_t) \tag{23}$$

From (23) we can find out that the tth loss function is only determined by the tth decision tree.

For a generated decision tree with T leaf nodes, the vector composed by all leaf nodes is $\omega \in R^T$. The complexity of the decision tree is expressed as $\Omega(f_t) = \gamma|\mathrm{T}| + \frac{1}{2}\lambda\sum_{j=1}^{T}\omega_j^2$. The complexity is decided by both the number of leaf nodes and the $l2$ norm.

Suppose $I_j = \{i|q(x_i) = j\}$ is the sample set for the samples belong to the jth leaf node, (23) can be rewritten as

$$\begin{aligned} L^t &= \sum_{i=1}^{n}\left[g_i\omega_{q(x_i)} + \frac{1}{2}h_i\omega_{q(x_i)}^2\right] + \gamma T + \frac{1}{2}\lambda\sum_{j=1}^{T}\omega_j^2 \\ &= \sum_{j=1}^{T}\left[\left(\sum_{i\in I_j} g_i\right)\omega_j + \frac{1}{2}\left(\sum_{i\in I_j} h_i + \lambda\right)\omega_j^2\right] + \gamma T \end{aligned} \tag{24}$$

Making the definition that $\mathrm{G_j} = \sum_{i\in I_j} g_i$ and $\mathrm{H_j} = \sum_{\mathrm{i\in I_j}} h_i$, we can make (24) changed to:

$$L^t = \sum_{\mathrm{j}=1}^{\mathrm{T}}\left[G_j\omega_j + \frac{1}{2}(H_j + \lambda)\omega_j^2\right] + \gamma T \tag{25}$$

If the depth of the tree is decided, which means the number of leaves is determined, then the only uncertain variable is ω. Make the first derivative of the lost function $\frac{\partial L}{\partial \omega_j} = G_j + (H_j + \lambda)\omega_j = 0$, the ideal value of ω_j should be:

$$\omega_\mathrm{j}^* = -\frac{G_j}{H_j+\lambda} \tag{26}$$

The ideal min loss is:

$$L_{min}^t = -\frac{1}{2}\sum_{j=1}^{T}\frac{G_j^2}{H_j+\lambda} + \gamma T \tag{27}$$

Thus we can get the minimum loss of the decision trees.

The GBDT algorithm can be generalized as:

a. Algorithm: GBDT

b. Decide the max depth of a decision.

c. Calculate the first order derivative g_i and second order derivative h_i of each sample.

d. Applying equation (8) to calculate the values of leaf values of the tree. Use the greedy strategy to generate the new tree.

e. Put the generated tree into the model $\hat{y}_\mathrm{i}^\mathrm{t} = \hat{y}_i^{t-1} + \epsilon f_t(x_i)$, ϵ is the learning rate of overall to suppress overfitting.

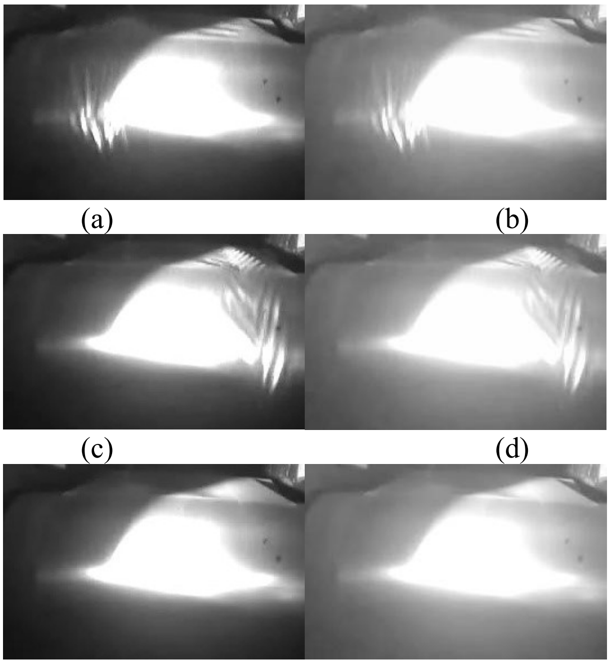
f. Go back to (b) to iterate or it achieves the end condition.

3. Experiments

The origin pixel size is 352×288, we focus on the interest part and select the pixels from 75 to 250 in rows and 30 to 300 in columns. The image is resized to 270×175.

3.1. Applying homomorphic filter on images

For the steel band wrinkling images, RGB offers little information. So our work is processed on grayscale images. Figure3 is the origin image and the image after passing the homomorphic filter (expanded to the pixel value range from 0-255):



(a) (b)

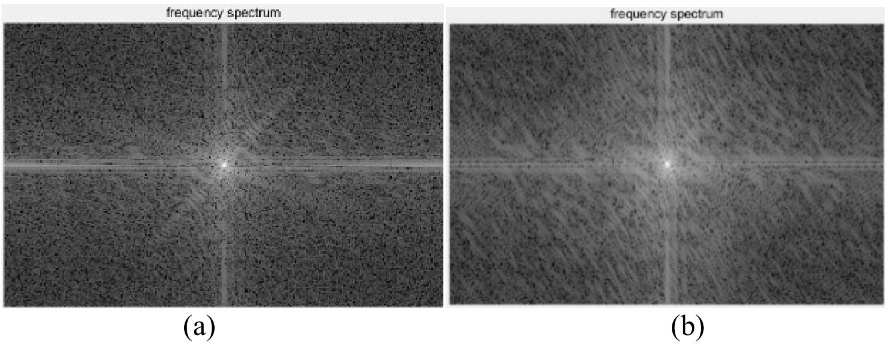
(c) (d)

(e) (f)

Figure 4. Origin image and image passing homomorphic filter

From figure3 we demonstrate the images of steel strip with wrinkling in different position like (a) and(c). (e) is an example of steel band with no obvious defection. Image (b), (d) and (f) are the corresponding images after passing the homomorphic filter(pixel value expanded to 0-255). From figure3 we can easily find that the flashlight is inclined. This induces a shadow line from the up right to left bottom. The steel band wrinkling is mainly located along this line. The sharp contrast of luminance makes some part of the wrinkling obvious to observe. However, the other part is close to the background, making it hard to recognize. We can also find that up-left and the up-right part of figure3 (b) and (d)is clearer than in figure3 (a) and (c).

Figure 4 shows the average frequency spectrums of the origin image and the image passing through the homomorphic. In order to distinctly demonstrate the difference, we plot the logarithmic frequency. Figure 4(b) shows that it contains more high frequency components.

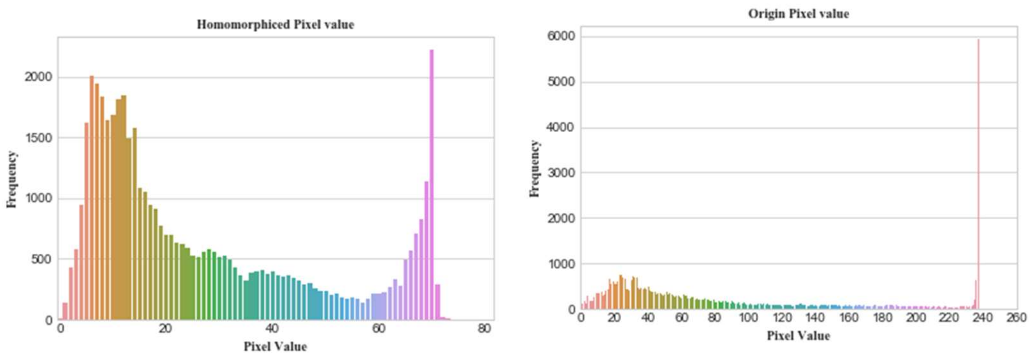


(a) (b)

Figure 5. Frequency spectrum of the image

IOP Conf. Series: Materials Science and Engineering **493** (2019) 012005 doi:10.1088/1757-899X/493/1/012005

We take the average pixel value of all the training set and make a count of the distribution of the pixel value from the origin image and after passing the homomorphic filter. From these two chats we come to some conclusion.



(a) (b)

Figure 6. Pixel value distribution of the image

The first property we noticed is that in the origin image the value scope is very large, ranging from 0 to 255. Most components only occupy a small and approximate share. Certainly, this is not a good distribution for the classification. Meanwhile the frequency of pixel value 240 is extremely high, corresponding to the reflectance of camera flash in the image. This part of the image contains very little information and is thus useless for the following work. These two issues are solved after passing he homomorphic filter. The homomorphic filter adjusts the distribution of grayscale to the maximum of about 80. Besides the ratio of frequency between high pixel value and low pixel value has observably decreased, reserving more useful information.

To test our classification algorithm, we randomly choose part of the data set as training set and the other as test set. We have 2503 positive samples which have no obvious wrinkling, 486 negative samples which have wrinkling. We change the parameters to see whether they would make a difference to our result. We change the proportion of train set and compare the training time of GBDT-only method and our homomorphic-GBDT algorithm. From table 1, we can find that our algorithm shows an advantage in computation time. And with the rising proportion of training examples, the time gap between two methods is enlarged.

Table 1. Classification time with and without homomorphic filter

Train/All	GBDT(s)	Homomorphic-GBDT(s)
0.60	27.93	25.18
0.70	31.09	29.04
0.80	33.25	30.30
0.90	39.35	33.20

We also record the accuracy of test set under different proportion of train samples and get table 2.

We repeat our algorithm 500 times to get the average accuracy of our algorithm. Our algorithm get a high accuracy and get the 100% correct at the train proportion of 90%.

Table 2. Classification accuracy with proposal method

Train/All	Accuracy
0.60	0.9908
0.70	0.9900
0.80	0.9983
0.90	1.00

3.2. Classification Model

We compare our homomorphic-GBDT method to other classification algorithms. As an illustration, we choose SVM as our comparison algorithm. Figure6 shows the computation time to train the model for SVM and homomorphic-GBDT. We can find that our method takes much less time than SVM. Especially when we take 90% as training samples, our algorithm takes almost half time of SVM.

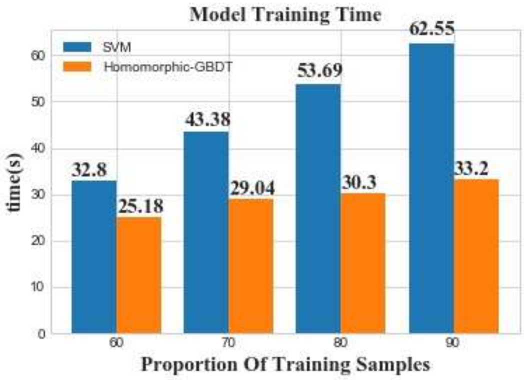


Figure 7. Model training time compared with SVM

On the other hand, our method also shows advantage over the traditional SVM in accuracy. We also do the same experiment 500 times to get the average accuracy by SVM. Our method can generate a better model to fit the origin data set. Besides, our model reaches the accuracy of 100% when we use 90% of our data as training set. This is important in practical production, for one missing may cause the waste of several hundred meters of steel.

3.3. Hyperparameter and loss function

We choose the hyperparameters as the following table 3:

Table 3. Hyperparameters of proposal method

num_leaves:	16
learning_rate:	0.4
max_depth:	4
reg_alpha:	0.1
reg_lambda:	0.1
n_jobs	10

These parameters can ensure that the loss function decreases rapidly and will not lead to the problem of overfitting. For binary classification, we use log loss as loss function:

$$l(\theta) = \sum_{i=1}^{N} y_i \log(h(x_i) + (1 - y_i)\log\big(1 - h(x_i)\big) \tag{28}$$

We test the time for online detection. The average time is shown in table4. Once the model is trained, our method need only about 3ms to detect one new image, which means about 330 frames per second. This is almost double of other methods, which is conform to the real time require.

In equation (28), y_i is the label for training sample, $h(x_i)$ is the output of the classification. We show the descent of loss function by different proportion of data used as training set in figure 8. We set the termination condition of iteration as the loss function do not improve for 5 rounds. The convergence value is decreasing with the rising proportion of train samples.

Table 4. Time comsuption of online test

Numbers	Total Time(s)	Average Times(s)	Accuracy
100	0.312228	3.10	1
200	0.650889	3.25	1
300	1.002834	3.34	1

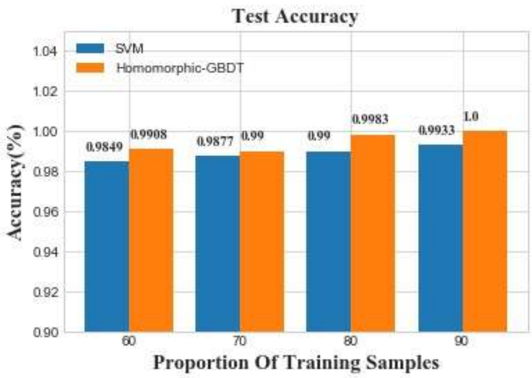


Figure 8. Test accuracy compared with SVM

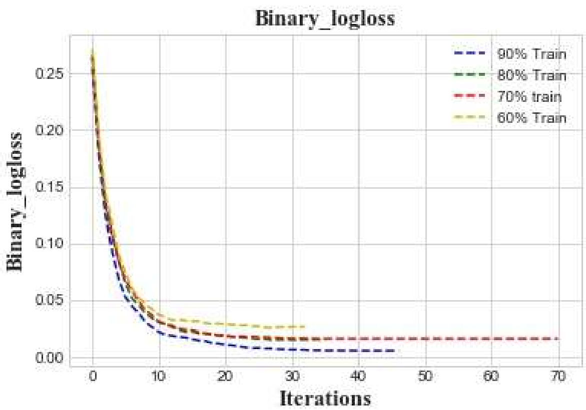


Figure 9. Loss function with iterations

4. Summary

In this paper, we introduce a real time detection method for strip defects. We use the homomorphic filter to solve the problem of uneven illumination in the origin image. Then we use the Gradient Boost Decision Tree as our classification. Our method reaches the accuracy of 100% and about 330 frames for the new incoming images. This is satisfied with the actual production need.

Acknowledgements

This work was supported by Shanghai Science and Technology Commission Project of China, No. 15DZ1202803 and No.17DZ1201605.

References

[1] Pan Y, Lin L, Chen X. Research of online detection of strip misalignment and wrinkling[J]. International Journal of Wireless & Mobile Computing, 2014, 7(7):153-158.
[2] Wang W, Chen X, Pan Y. Strip wrinkling detection based on feature extraction and sparse representation[J]. International Journal of Wireless & Mobile Computing, 2017, 12(1):36.

[3] Kang G W, Liu H B. Surface defects inspection of cold rolled strips based on neural network[C]// International Conference on Machine Learning and Cybernetics. IEEE, 2005:5034 - 5037 Vol. 8.
[4] Wang Y, Feng D, Li D, et al. A mobile recommendation system based on logistic regression and Gradient Boosting Decision Trees[C]// International Joint Conference on Neural Networks. IEEE, 2016:1896-1902.
[5] Kong X, Xia F, Wang J, et al. Time-Location-Relationship Combined Service Recommendation Based on Taxi Trajectory Data[J]. IEEE Transactions on Industrial Informatics, 2017, PP(99):1-1.
[6] Liu J Y, Ding Y, Tao L I. Classification of Flight Delay Based-on GBDT[J]. Mathematics in Practice & Theory, 2018.
[7] Friedman J H. Greedy Function Approximation: A Gradient Boosting Machine[J]. Annals of Statistics, 2001, 29(5):1189-1232.
[8] Yeni L I, Lin X, Chen J, et al. Defects Identification for Carbonization Bamboos Based on GLCM and SVM[J]. Journal of Jimei University, 2017
[9] Im J, Jensen J R. A change detection model based on neighborhood correlation image analysis and decision tree classification[J]. Remote Sensing of Environment, 2005, 99(3):326-340.
[10] Esseiva J, Caon M, Mugellini E, et al. Feet Fidgeting Detection Based on Accelerometers Using Decision Tree Learning and Gradient Boosting[C]// International Conference on Bioinformatics and Biomedical Engineering. Springer, Cham, 2018:75-84.
[11] Sheng Y, Rovnyak S M. Decision tree-based methodology for high impedance fault detection[J]. Power Delivery IEEE Transactions on, 2004, 19(2):533-536.
[12] Jindal A, Dua A, Kaur K, et al. Decision Tree and SVM-Based Data Analytics for Theft Detection in Smart Grid[J]. IEEE Transactions on Industrial Informatics, 2016, 12(3):1005-1016.
[13] Alonso I P, Llorca D F, Sotelo M A, et al. Combination of Feature Extraction Methods for SVM Pedestrian Detection[J]. IEEE Transactions on Intelligent Transportation Systems, 2007, 8(2):292-30
[14] Xu L, Lü J. Bayberry image segmentation based on homomorphic filtering and K-means clustering algorithm[J]. Transactions of the Chinese Society of Agricultural Engineering, 2015, 31(14):202-208.
[15] Tiwari M, Gupta B. Brightness preserving contrast enhancement of medical images using adaptive gamma correction and homomorphic filtering[C]// Electrical, Electronics and Computer Science. IEEE, 2016.
[16] Duan Q, Fen-Xia W U, Hong L I. An Enhanced Algorithm of Color Image Based on Block Homomorphic Filter in Two Channels[J]. Computer Technology & Development, 2016.