

# An Intelligent Real-time Vision System for Surface Defect Detection

<sup>1</sup>Hongbin Jia, <sup>1</sup>Yi Lu Murphey, <sup>2</sup>Jianjun Shi, <sup>3</sup>Tzyy-Shuh Chang

<sup>1</sup>Department of Electrical and Computer Engineering, The University of Michigan-Dearborn, Dearborn, MI 48128-1491

<sup>2</sup>Department of Industrial and Operations Engineering, The University of Michigan-Ann Arbor, Ann Arbor, MI 48105

<sup>3</sup>OG Technologies, Inc., 58 Parkland Place, Suite 200, Ann Arbor, MI 48103

## Abstract

*In recent years, there is an increased need for quality control in the manufacturing sectors. In the steel making, the rolling operation is often the last process that significantly affects the bulk microstructure of the steel. The cost of having defects on rolled steel is high because it takes more than 5000 KW-Hr to produce a ton of steel. Early detection of defects can reduce product damage and manufacturing cost. This paper describes a real-time visual inspection system that uses Support Vector Machine to automatically learn complicated defect patterns. Based on the experimental results generated from over one thousand images, the proposed system is found to be effective in detecting steel surface defects. The speed of the system for feature extraction and defect detection is less than 6 msec per one-megabyte image.*

## 1. Introduction

In recent years, there is an increased need for quality control in the manufacturing sector. The aerospace and automotive industries reject any materials with defects in manufacturing processes because a minor defect in a manufactured part might result in a disaster at a later stage. Without automatic machine vision technology, steel rolling operations is not able to perform real-time inline surface defect inspection. The failure to identify some defects can lead to factory down time and significant economic losses.

The following issues make automatic inspection challenging in real-time in a hot rolling process: (1) The rolling speed can be as fast as 225 MPH and adequate sampling requires a data acquisition rate of 80 MB per second or more. Analyzing data at this speed requires sophisticated real-time algorithms in addition to processing power. (2) The many factors make defect features difficult to detect: material impurities, fluctuation of lighting, object twitter and other unexpected movement between the sensor and the object.

Our research efforts have been put in modeling defect detection as a pattern classification problem. There are two important issues to be addressed. First, effective defect features must be explored, extracted and optimized.

Defect features characterize surface defects and determine the complexity of the classification system. The second important issue is the classifier design, which determines the performance of the entire vision system in classifying defects. We have developed a machine learning system based on the Support Vector Machine. During the design of the machine vision system, we address three main requirements: (1) it should be robust and capable of providing good discrimination even in the case of noisy input data. (2) It must be fast enough in order to meet in-line speed requirements. (3) It should be incrementally scalable to incorporate new known objects, particularly new known defect classes, without retraining with the whole data set. In addition, we address the issues including parameter selection in SVM.

The system has been implemented and tested on image data downloaded directly from hot rolling manufacturing productions. The experiments showed that the machine vision system meets the real-time requirement with a high defect detection rate.

## 2. Real-time Surface Defect Detection in Hot Rolling Process

Figure 1 illustrates the machine vision system we developed. The system consists of two major components, defect feature extraction and defect classification, which are described in detail in the following subsections. The defects generated during a steel rolling process are typically low resolution and appear in different shapes. In this paper, we present our research in detecting an important defect category, seams, as shown in Figure 2 (a).

### 2.1 Rough Filtering and Seam Candidate Finding

The defects in steel rolling processes typically occupy small areas of a steel bar. We developed a two-step feature extraction approach, rough filtering, and defect candidate finding, in order to achieve computational efficiency.

The objective of the rough filtering is to quickly pass the majority of the normal surface images as a steel bar going through the rolling process. For those images that

do not pass, it identifies the regions in these images that have a high probability of containing defects.

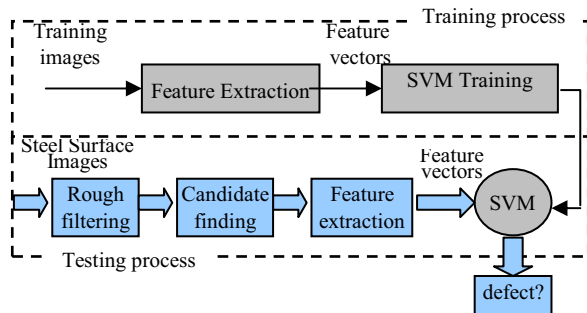


Figure 1. A machine vision system for defect detection

The rough filtering algorithm explores several defect features including the orientation of the seams, which is approximately vertical in the images. The rough filtering algorithm applies a horizontal gradient operator to the image to detect the edges of the seams. If a filtered image has no significant gradient values, the image is considered normal. However the horizontal gradient filtering also generates false seams as shown in Figure 2 (b), which is considered as a normal image since the thin lines were caused merely by an uneven surface on the steel. Our experiments showed that more than 50% edges generated by the horizontal filtering are not seam defects, but true seams defects are always detected in almost 100% success rate.

The defect candidate finding algorithm explores fact that a seam is thin, and usually has high contrast with the seam being always darker than its surrounding area. This implies that a horizontal gradient filter usually produces significant negative values on the left side and significant positive values on the right side of a seam. The algorithm first searches for the initial seam points, called seeds. A seed point is the local maxima in the gradient image and has either negative-to-positive or positive-to-negative neighbors on the two sides. If the minimum seam length is set to be  $n$  pixels long, then the gradient image is sampled every  $n$  rows to search for seeds. The second step is to grow the seam region starting from the seed points. It begins by examining six-direction neighboring pixels of the seed to determine if they also belong to the same surface feature. This can be explained by Figure 2 (c). Once a specific candidate region cannot be grown further, the search terminates and the candidate region is identified.

## 2.2 Feature Extraction from Defect Candidate Region

Once the candidate regions are located, the following features are extracted for defect classification.

- The length of the potential seam in pixels, which is obtained by the defect candidate finding algorithm described above.

- Gray scale contrast between the possible seam and its neighboring areas. This feature measures how dark the seam appears relative to the background.
- The disparity that indicates the horizontal intensity difference from one side of the region to the other side. This feature is added to distinguish the seams from shadows.
- Average and variance of gray scale values of the candidate regions, which are effective in eliminating noise.

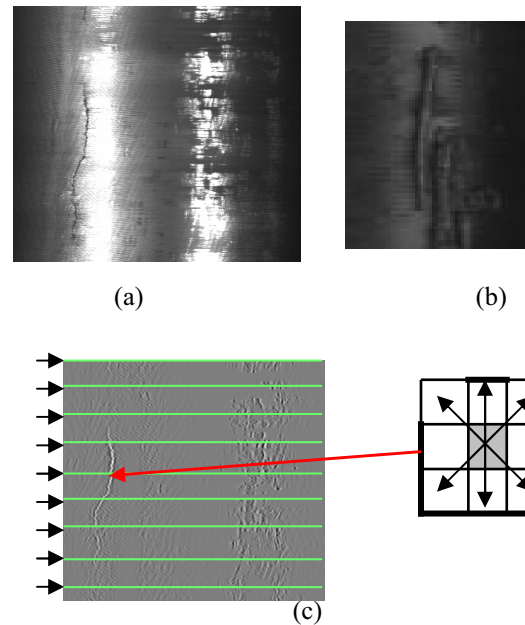


Figure 2. Examples of seam defect images and normal images.

## 2.3 Defect Detection using SVM

The SVM learning algorithms were developed based on statistical pattern recognition theories[1]. Successful implementations of SVMs have been reported in various applications [2, 3, 4]. In general SVM classification approach is considered the better choice for noisy datasets in comparison with Naive Bayes, C4.5, and neural networks in terms of accuracy and computational complexity [6].

A SVM maps the input features to a high dimensional feature space through a kernel scheme, generates a classification hyperplane based on the maximum margins in the feature space. Unlike traditional classifiers such as artificial neural networks (ANNs), which minimize the empirical training error, a SVM aims to minimize the upper bound of the generalization error. This upper bound represents the error that could be resulted in the unseen data for a classifier. Therefore, high generalization performance can be achieved with an SVM.

We model rolling defect detection as a non-linear classification problem using a SVM with RBF as the kernel function. Let a training data set have the form  $TR = \{(x_i, y_i), i = 1, \dots, N\}$ , where  $(x_i, y_i) \in \mathbb{R}^m, y_i \in \{-1, +1\}$  for  $i = 1, 2, \dots, N$ , and the RBF kernel  $K(x_i, x_j) = \exp^{-\gamma \|x_i - x_j\|^2} = \phi(x_i)\phi(x_j)$ . The objective function used in SVM learning is  $\frac{1}{2} \omega^T \omega + C \left( \sum_{i=1}^N \xi_i \right)$ , with the following constraints:  $y_i((\omega^T \phi(x_i)) + b) \geq 1 - \xi_i$  and  $\xi_i \geq 0, i = 1, \dots, N$ .

The parameter  $C$  is the coefficient of a penalty term that can be chosen by the user, a larger  $C$  corresponding to assigning a higher priority to training errors. The solution to the above optimization is:

$$\omega = \sum_{i=1}^{N_s} \alpha_i y_i \phi(s_i), \quad b = w \cdot \phi(x_i) - y_i \quad \text{for } \alpha_i > 0$$

where  $s_i$  are support vectors,  $N_s$  is the number of support vectors and  $0 \leq \alpha_i \leq C$ .

Based on the above description, it is clear that we need to determine the following parameters at the SVM learning stage:  $\gamma$ ,  $C$ , support vectors  $s_i$  and  $\alpha_i$ , for  $i = 1, \dots, N_s$  and  $b$ .

We decompose the SVM learning into two stages. At the first stage, we used a multi-step k-fold cross validation algorithm,  $\gamma$  C-generation algorithm to determine the effective learning parameters,  $\gamma$  and  $C$  [7]. At the second stage, we developed an incremental SVM learning algorithm that uses the parameters  $\gamma$  and  $C$  selected at the first stage to generate support vectors  $s_i$ ,  $\alpha_i$  and  $b$ , from a small subset of training data.

In machine learning, *incremental learning* refers to the process of accumulating and managing knowledge over time. An incremental learning system updates its hypotheses as new instances arrive without reexamining the old instances. Such a learning strategy is both spatially and temporally economical, because there is no need to store and reprocess old instances. More importantly, incremental learning provides a framework for the development of scalable learning systems, which is important in manufacturing environment. We developed the following incremental SVM learning algorithm to build SVM models when training data become available in time. When the first training data set  $TR_1$  is available, we obtain a SVM model, SVM1 using the  $\gamma$  and  $C$  obtained at the first stage. We collect all the support vectors in SVM<sub>1</sub> and the margin data in a new set called

$\Omega_1$ . A margin point  $x$  is a training data example that satisfies

$$|f(x)| = |\sum \alpha_i y_i K(s_i, x) + b| < a$$

where  $a$  is predefined margin threshold.

When a new training data,  $TR_2$  come in, we retrain a SVM using the data,  $\Omega = \Omega_1 \cup TR_2$  to obtain SVM<sub>2</sub>. We collect the margin points and errors data and store them in  $\Omega_2$ . The process repeats whenever new training data becomes available.

The next problem in defect classification we need to address is the noise data problem. There are two types of noise existing in training data. The first case is called wrongly-labeled outlier, a training point that is wrongly labeled in the training data set. There are two negative effects with this kind of noise. First, the decision hyperplane in the feature space can be distorted if a significant number of outliers are present in the training set, or if one outlier is sufficiently far away from the decision hyperplane. Second, the existence of outliers increases the number of the support vectors in the model. This is due to an increased value of the objective function resulting in a larger margin between the classes in the high dimensional feature space. The second case is called out-of-range outlier, a training point that is far away from the main cluster of the class it belongs to. For the wrongly labeled outlier, we use the information of its  $k$  nearest neighbors to remove it. For a given training point, all its  $k$  nearest points must have a selected target values that are "similar" for all these near-to-each-other training points. A training point that has a significantly different target value will be removed. For the out-of-range outlier, we estimate the distribution of the entire training data without labeling information. If a given training point is out of the sphere of the entire training data set, it will be removed as an out-of-range outlier.

The result from the SVM learning is a SVM model that includes kernel parameters  $\gamma$  and  $C$ , support vectors ( $s_i$ ), coefficients of the support vectors ( $\alpha_i$ ), and the target values of each support vector ( $y_i$ ), and the bias ( $b$ ).

During the testing phase, the unseen data will be classified as a specific category according to the decision function in the model. That is, a new sample  $x^*$  can be classified with the label  $sign \left( \sum_{i=1}^{N_s} \alpha_i y_i K(s_i, x^*) + b \right)$ . The

$sign$ , positive or negative, indicates which side the new sample belongs to with respect to the hyperplane.

### 3. Experimental Results

We have tested our system on image data collected directly from hot rolling processes. The training data have 920 images, and the testing data have 306 images.

We applied  $\gamma$ C-generation algorithm discussed to the training data and obtained the learning parameters  $C = 8$ ,  $\sigma = 0.707$ , and then trained a SVM model. Table 1 summarizes the results, along with results obtained using a feed forward artificial neural network (ANN) classifier (backpropagation) obtained from the data without pre-processing. The ANN classifier has one hidden layer with eight nodes. The ANN gave very low detection rate on the seam class, and relatively high detection rate on non-seam images. In manufacturing, it allows a low percentage of false alarms, but requires high detection rate on defect products.

Table 1. Performances of SVM and ANN on testing data

Predict \ True	SVM classifier				ANN classifier		
	Total	Seam	Non-seam	Accuracy	Seam	Non-seam	Accuracy
Seam	49	40	9	85.1%	27	22	55.1%
Non-seam	257	20	237	91.5%	9	248	96.5%
Total	306	60	246	90.5%	34	272	89.9%

Table 2 shows the performances of the SVM system and the ANN system trained on the data being pre-processed for noise removal. As we can see both systems, SVM and ANN performed better on the same test data. SVM has improved significantly in both classes, seam and normal.

Table 2. Performances on testing data by SVM and ANN trained on data being pre-processed for noise removal

Predict \ True	From SVM classifier				From ANN classifier		
	Total	Seam	Non-seam	Accuracy	Seam	Non-seam	Accuracy
Seam	47	41	6	87.2%	34	13	72.3%
Non-seam	259	11	248	95.8%	15	244	94.2%
Total	306	52	254	94.4%	49	257	90.8%

The speed of the SVM classifier system was tested on all images from both training and test data. Table 3 delineates the time used in feature extraction from the original images, time needed for the SVM prediction, and the total processing time on all 1226 images. The execution was performed on a PC with Intel Pentium IV CPU 1.5 GHz, 256M RAM. From the table it is evident that the most time-consuming process is feature extraction. The SVM classification is extremely fast, which can be negligible in comparison with the feature extraction portion. The entire system provides a process rate of 172 images per second, which is acceptable in a real-time manufacturing environment.

Table 3. Speed on test data

	Feature Extraction (s)	SVM prediction (s)	Total (s)	Detection speed (images/second)
Time used	6.920	0.201	7.121	172.17

## 4. Conclusions

This paper presented a defect detection approach based on machine vision and intelligent systems. The approach fully explored visual defect features, and developed a SVM learning algorithm that automatically learns complex decision boundaries when data noise exists. Experimental results demonstrate the potential of a Support Vector Machine as a promising classifier for defect (seam) data with application in real-time manufacturing environment. The in-line tests are currently underway. We expect that if successful, the system will help the steel industry to improve the consistency of product quality and lower production costs.

## 5. Acknowledgement

This material is based upon work supported, in part, by a REEDF grant from the State of Michigan, NIST-ATP grant and Charter Steel.

## 6. References

- [1] V. N. Vapnik, *The Nature of Statistical Learning Theory*. Springer, New York, 2nd edition, 2000.
- [2] B. Schölkopf, C.J.C. Burges, A. J. Smola, *Advances in Kernel Methods – Support Vector Learning*, MIT Press, Cambridge, Massachusetts, 1998
- [3] S. Mukkamala, G. Janoski, A. Sung, *Intrusion Detection Using Neural Networks and Support Vector Machines*, IEEE IJCNN (May, 2002)
- [4] B. Lerner and N. D. Lawrence, *A Comparison of State-of-the-Art Classification Techniques with Application to Cytogenetics*, *Neural Comput & Applic* (2001)10:39–47
- [5] A B M S. Ali and A. Abraham, *An Empirical Comparison of Kernel Selection for Support Vector Machines*, 2nd International Conference on Hybrid Intelligent Systems, Chile, IOS Press Amsterdam, The Netherlands, December'02.
- [6] A. Kelemen, Y. Liang and S. Franklin, *A Comparative Study of Different Machine Learning Approaches for Decision Making*, in: "Recent Advances in Simulation, Computational Methods and Soft Computing" (N. E. Mastorakis, ed.), WSEAS Press, Piraeus, Greece, pp. 181-186, 2002
- [7] Y. L. Murphey, Z. Chen, M. Putrus, and L. A. Feldkamp, "SVM learning from large training data set," IEEE International Joint Conference on Neural Networks, July, 2003.