DEFECT DETECTION AND CLASSIFICATION USING MACHINE LEARNING CLASSIFIER

Mitesh Popat[1] and S V Barai[2]
[1] Johns Hopkins University, Baltimore, USA
[2] Indian Institute of Technology, Kharagpur, India.

Abstract: In most cases visual inspection of the hot strip by an inspector (in real time or video-taped) is a difficult task. The issues in this project study are data modeling, Machine Learning (ML) model - neural networks (NN) modeling and reliability of such models for automatic detection and classification of defects of hot strips. The proposed study intends to develop general guidelines for developing NN model for automatic surface inspection for hot strip mills.

Introduction: In steel industry, visual inspection of the hot strip by an inspector is, in most cases not possible because of the high speeds and high temperature involved. In recent times, only video monitors and video recorders have been used where inspectors check on-line or taped video sequences for defects. In this way, only small parts of the strip's top or bottom side are viewed. Additionally, this visual inspection is subjective and dependent on a large number of human factors such as the problems of working through night shifts, attention being drawn to other events, subjectivity in terms of defect severness assessment and restricted capabilities to cover the entire strip at high line speeds. Developing automatic detection and classification of surface defects of hot strips has been really a challenging problem in the field of steel manufacturing industry (Rinn et al., 2002).

An automatic detection and classification system requires knowledge of data concerning the current state of the hot strips, and a methodology to integrate various types of information into decision-making process of evaluating the quality of the product. The need for surface detection technologies for surface defect classification has long been recognized. Real time image processing typically involves the application of high-speed camera, which may give the defect images in real time. The large amount of information is gathered during this process in the form of images. Human experts evaluate this information and give the level of defects, types of defects of the hot strips and will suggest remedial measure to over come those defects. However, the complexity of such program could be avoided by creating a Decision Support System (DSS). This study will take the initial steps in developing a DSS for hot strip evaluation in the manufacturing plant based on the process carried out in the practice. Typical paradigm is shown in Figure 1.

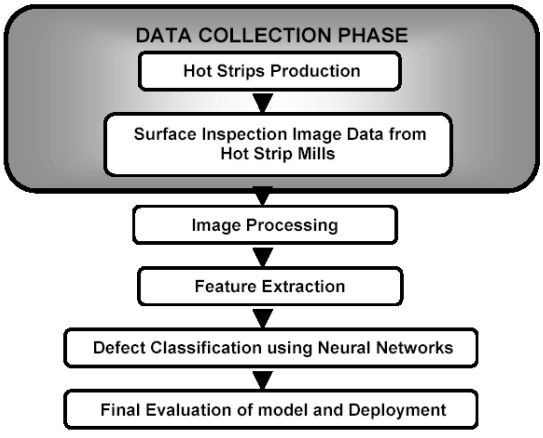


Figure 1: Defect detection classifier paradigm

A methodology is developed for defect image classifier using machine-learning model - Artificial Neural Networks (ANN). ANN has an excellent generalization capability to learn from the set of data obtained during real-time images (Anderson 1995, Bishop 1995, Haykin 1999). The issues related to data modeling, ML model - neural networks modeling and reliability of neural models

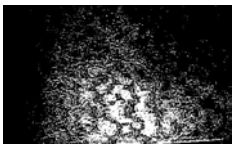
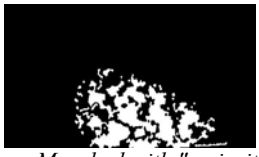
for automatic detection and classification of defects of hot strips are addressed. The study demonstrates general guidelines for developing ANN model for automatic surface inspection for hot strip mills taking into account some real but offline examples of defects and developing a model to classify them.

Data Collection Phase: In this particular case, the data collection from one of the reputed steel mill of India was really a challenging task (DLM, 2002). Basic difficulty lies in identifying those features or set of features in the images, which are more or less unique for a particular defect. The images are studied and those images which are ideal and can be used for the modeling is chosen. Following paragraphs describe some common defects (DLM 2002), which have been used in the present study.

- *Shell* is irregular, flaky overlapping material permeated with non-metallic inclusions. The overlapping portions of the surface phenomenon known as shell are manifested in various forms and vary in size. The overlapping material has an irregular perimeter and is separated by nonmetallic or oxidic inclusions, or scale, from the base material.
- *Holes* are discontinuities in the material that extend right through from the top to the bottom surface.
- *Scale pits/Scabs* are caused by scale being embedded into the surface of the material during hot rolling. The scale may be removed again by pickling, but the defect may not be completely eradicated during the cold rolling process. The appearance of these flaws ranges from punctiform and linear to wide-area defects.
- *Residue scale* is scale which has not been completely removed by pickling and which is rolled into the surface during the cold rolling operation. Its appearance range from punctiform or linear to wide-area flaws.
- *Coil breaks* are cross breaks orientated at right angles to the direction of rolling. They may run across the width of the strip or be located at the edge of the strip, with either regular or irregular spacing.
- *Coiling slip marks* are the result of mechanical damage to the surface and usually appear in clusters.
- *Rust* is a surface layer of the corrosion products of iron rust appears in the form of reddish -yellow to black patches on the strip, the shape pattern and extension of which may vary considerably.

Image Processing: For image analysis (Gose et al. 1997, Gonzalez and Woods 1992) the image processing toolbox version 2.1 (IMPT 2002) of MATLAB (2002) is used. Following are the major steps in image processing and description of artificial noise to simulate field noise on the data (IMPT 2002).

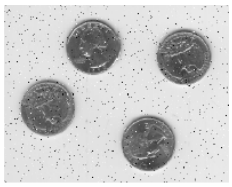
- *Standard format*: Resizing of all the images to a standard size of 600 X 400 pixels.
- *Binary:* Binary images creation from indexed, intensity, or RGB images. These are images with only black and white pixels.
- *Morphing:* The binary image is morphed (Figure 2) with ‘majority morphing’ to take care of noise in the image. By trial and error morphing is found to be very effective to take care of some very common noises. It sets a pixel to 1 if five or more pixels in its 3-by-3 neighborhood are 1’s; otherwise, it sets the pixel to 0.

Original image *Morphed with "majority morphing"*

Figure 2: Example of morphing

- *Salt & pepper:* If I is the image, and d is the noise density, this affects approximately d*prod (size (I)) pixels. Shown below is original and noisy image (Figure 3) with salt and pepper noise.

Original image *Image with added 'salt and pepper' noise*

Figure 3: Example of noise in the image

- *Speckle:* It is a multiplicative noise added to the image I, using the equation J (noisy image) = I+n*I, where n is uniformly distributed random noise with mean 0 and variance v.
- *Gaussian:* Gaussian white noise of mean m and variance v is added to the image.

Feature extraction: This is a very crucial phase of the image processing. There are many feature vectors, which can be extracted from a binary image, but we need minimum number of features, which can be easily extracted and also are good enough to distinguish between the common defects found on the strip. The objective is that if one of the features is same for two images at-least there are some other features, which can differentiate them. Following paragraphs discuss about the various features in the present domain context. The selection is purely on the basis of observation on the available images.

- *Number of objects*: It is the number of connected white pixels. It may use four-connected or eight-connected objects. The default used here is eight. The four-connected takes the pixels only above it and in its line as neighbors and in eight it takes the diagonal pixels also as its neighbors (Figure 4).

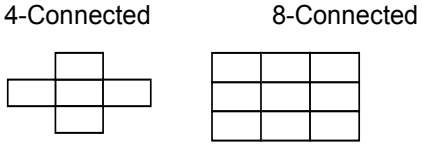


Figure 4: Identification of number of objects

- *Euler number*: is a scalar whose value is the total number of objects in the image minus the total number of holes in those objects. It can have a 4-connected objects and 8-connected objects. e.g. for the image shown in Figure 5 has Euler number as -2.

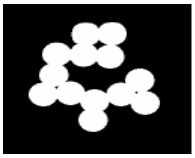


Figure 5: Identification of Euler number

- *Perimeter:* The image is changed to a perimeter image. A pixel is part of the perimeter if its value is 1 and there is at least one zero-valued pixel in its neighborhood. The value of perimeter (Figure 6) is simply the number of objects in the perimeter image.

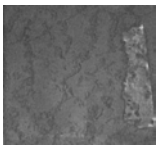
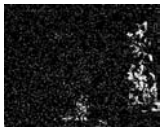
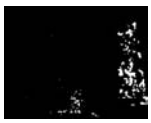
Original *Perimeter*

Figure 6: Example of perimeter

- *Area*: It is the area of all of the on pixels in an image. The area of an individual pixel is determined by looking at its 2-by-2 neighborhood. There are six different patterns distinguished, each representing a different area:
    - Patterns with zero on pixels (area = 0)
    - Patterns with one on pixel (area = 1/4)
    - Patterns with two adjacent on pixels (area = 1/2)
    - Patterns with two diagonal on pixels (area = 3/4)
    - Patterns with three on pixels (area = 7/8)
    - Patterns with all four on pixels (area = 1)

Each pixel is part of four different 2-by-2 neighborhoods. This means, for example, that a single on pixel surrounded by off pixels has a total area of 1. In Table 1, two examples are shown to demonstrate standard operations performed on image and noise-removal from the image.

Table 1: Typical example of noise removal

Image Processing	Example 1	Example 2
Standard image (Without noise)		
After 'salt and pepper' noise	 Area = 2600.2, Perimeter = 1986.2, No. of objects = 203, Euler. no. = 195	 Area = 6179.2, Perimeter = 5741.2, No. of objects = 285, Euler. no. = 2789
After processing with 'majority morphing'	 Area = 2028, Perimeter = 1255, No. of objects = 63, Euler no. = 63	 Area = 2048.8, Perimeter = 1296.8, No. of objects = 74, Euler no. = 71

Note: Noise is practically removed automatically using morphing

For the present study the data (DLM, 2002) collected from a typical steel industry of India are used. Additional datasets are created using the image processing and feature extraction

approaches as described earlier by introducing different level of noise. Details of the datasets are given in Table 2.

Table 2: Dataset details

Dataset Name	Data processing	Samples
Dataset A	The images chosen for modeling from the images collected from the mill.	15
Dataset B	Dataset A + "salt and pepper" noise.	30
Dataset C	Dataset A + "Speckle" noise	30
Dataset D	Dataset A with three different intensity of the "salt and pepper" noise.	45
Dataset E	Dataset A with three different intensity of the "Speckle" noise	45
Dataset F	Dataset D + Dataset E.	90

Defect Classification using Machine Learning Model*:* Following sections explains about the input data modeling and neural networks model.

- *Input:* Input in the network is four feature vector namely *area, perimeter, number of object and Euler number* of the binary image. Tan-sigmoid function is used to map normalized feature vector to [-1, 1]. So the input will be a 4xP vector, where P is the number of training sets.
- *Network Architecture:* To make a decision regarding how many hidden layers should be used, one should consider the overall performance of the network i.e. its generalization and mapping capabilities as a final goal. Mostly the choice is limited to one or two hidden layers. In most of the cases it is documented that three-layered network is sufficient to perform any non-linear mapping leaving some esoteric exceptions. On the basis of hit and trial with one and two hidden layers, one hidden layer is found to be most optimal. From literature and previous works it's evident that in classification problems, mostly one layer is efficient. Nodes in the hidden layer are again like the choice of number of hidden layers in the network. After trial and error, with whole range of number of nodes and comparing results, 10 nodes in the hidden layer seem to be optimal. The number of training sets available also influences the choice of nodes in the hidden layer. Nodes in the output layer are naturally the number of images into which the set of images have to be classified. In the output nodes the image under which the input is classified will contain value 1 (or near to one) and all other 0 (or near to zero). Therefore number of output nodes is seven equal to the defect classes into which the classification has to be done. These defect classes are discussed earlier. The transfer function between hidden layer and the output layer should map real values to [0, 1], so that we can put a threshold value e.g. 0.5, and thus classification is achieved. The ideal function for this is log-sigmoid function, which has desired properties and is also very common in the classification problems. After going through various functions, log-sigmoid is found to be efficient for this mapping. Finally the network paradigm of back-propagation neural network for the classification problem is shown Figure 7.

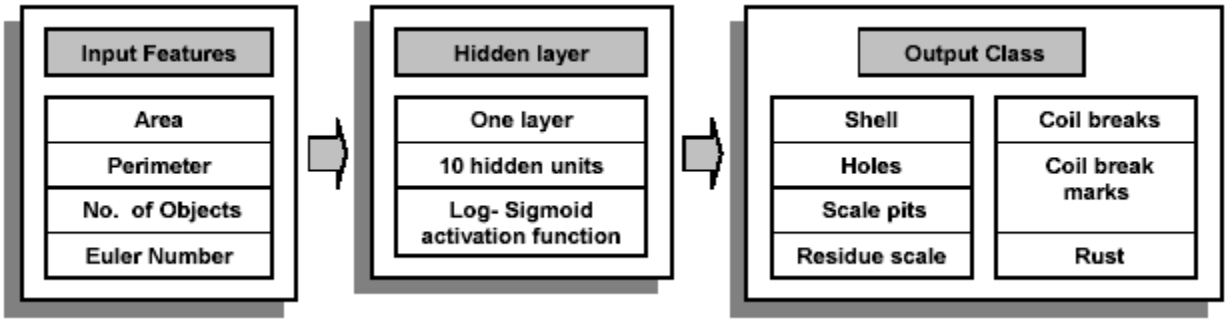


Figure 7: Neural network paradigm

- *Training:* Once the network weights and biases have been initialized, the network is ready for training. The choice of the training algorithm (Haykin, 1999) depends on many factors like no. of training sets, nodes in each layer, transfer functions used, time it takes to train, and above all the accuracy. The training process requires a set of examples of proper network behavior network inputs and target outputs. The performance function used here is mean square error (MSE). All of these algorithms use the gradient of the performance function to determine how to adjust the weights to minimize performance. The gradient is determined using a technique called back-propagation, which involves performing computations backwards through the network.
- *Back-propagation Algorithms:* There are many variations of the back-propagation algorithm. The simplest implementation of back-propagation learning updates the network weights and biases in the direction in which the performance function decreases most rapidly. There are two different ways in which this gradient descent algorithm can be implemented: incremental mode and batch mode. In the incremental mode, the gradient is computed and the weights are updated after each input is applied to the network. In the batch mode all of the inputs are applied to the network before the weights are updated. The functions described above have two back-propagation training algorithms: gradient descent and gradient descent with momentum. These two methods are often too slow for practical problems. There exist high performance algorithms, which can converge from ten to one hundred times faster than the algorithm discussed above. All of the algorithms here operate in the batch mode. These faster algorithms fall into two main categories. The first category uses heuristic techniques, which were developed from an analysis of the performance of the standard steepest descent algorithm. One heuristic modification is the momentum technique. The second category of fast algorithms uses standard numerical optimization techniques.
- *Resilient Back-propagation:* Multi-layer networks typically use sigmoid transfer functions in the hidden layers. These functions are often called *squashing* functions, since they compress an infinite input range into a finite output range. Sigmoid functions are characterized by the fact that their slope must approach zero, as the input gets large. This causes a problem when using steepest descent to train a multi-layer network with sigmoid functions, since the gradient can have a very small magnitude, and therefore cause small changes in the weights and biases, even though the weights and biases are far from their optimal values. The purpose of the resilient back-propagation training algorithm is to eliminate these harmful effects of the magnitudes of the partial derivatives. Only the sign of the derivative is used to determine the direction of the weight update; the magnitude of the derivative has no effect on the weight update.

In the present study, resilient back-propagation is used. For all the operations related to Neural Network modeling, Neural Network toolbox (NNET, 2002) of MATLAB (2002) is used extensively.

Results: The machine learning model performance is evaluated using holdout exercise (Reich and Barai, 1999). The performance based on time taken to train the network is not considered here; reason being the time taken is so less (usually some minutes) that it hardly makes a difference for the model. The datasets (Table 2) were considered for the machine learning model training and testing. The major results for various exercises are summarized in Table 3.

Table 3: Neural networks performance

Trained with:	Tested on:	Misclassification:	% Accuracy	Remarks on misclassified images
1. Dataset A	Dataset B	Zero of fifteen	100 %	-
2. Dataset B	Dataset A	One of fifteen	93.33 %	Correct answer also included.
3. Dataset D	Dataset A	One of forty-five	93.33 %	Correct answer also included
4. Dataset E	Dataset A	Zero of forty-five	100 %	-
5. Dataset F	Dataset A	Zero of ninety	100 %	-

The results in above table demonstrated that the performance accuracy was more than 93% even though the images were noisy.

Validation of the Model: A machine-learning model always requires rigorous validation before deployment. The test used here is known as leave one out (Reich and Barai, 1999). The dataset D is divided into disjoint training and testing sets. After training, the network is evaluated on the test set and errors give an estimate of the generalization error. The results of this exercise have high variability that depends upon this random sub-division. In smaller database (as is the case here), a variation of this exercise is carried out. The dataset is randomly subdivided many times and then trained and tested for finding out misclassification by model. And finally the average error is computed for misclassification.

The dataset (Dataset F of 90 images) is divided such that except one all others are used for training and then tested on that left out image and also a standard set (Dataset A of 15 images) of data. Total of ten iterations are carried out. The dataset is divided 90 times into two sub-divisions, each time leaving one out and then trained on the rest. So basically training and testing is done 90 X 10 =900 times.

In Tables 4 and 5, results of the image leave one out exercise for Dataset F and testing of trained model on Dataset A respectively are given.

Table 4: leave one out performance for Dataset F

Exercise number	1	2	3	4	5	6	7	8	9	10	Total misclassifications (Out of 900 images)
Number of Misclassification	6	5	5	5	7	5	5	5	5	5	53

Accuracy = ((900 - 53)/900) 100 = 94.11 %*

Table 5: Performance for Dataset A for the trained model of leave one out of Dataset F

Exercise number	1	2	3	4	5	6	7	8	9	10	Total misclassifications (Out of 13500 images)
Number of Misclassification	24	15	15	15	24	15	15	15	15	15	168

$$Accuracy = ((13500 - 168)/13500) * 100 = 98.75\ \%$$

Results have demonstrated robustness of the model and accuracy more than 98% in classifying the defects.

Discussion: The model developed aims at initiating a research work in the direction of defect classification on steel surfaces, particularly in Indian context. Some of the major difficulty faced in this area is related to field operations for example, the data collection from the field, understanding the problem for various mills. The model has used image processing and neural networks as tool to come up with a complete model. Various operations are studied and tried before the model is finalized as is now. The results of the holdout as well as leave one out exercises show a promising performance of the model and its architecture. The model can be generalized to a higher degree if sufficient data become available for use. Thus, the present work is just a penultimate model before a live model for the industry can be designed. Various issues have to be addressed before the model can be deployed. First of all, the image capturing at such a high speed of 10-15 m/sec has to be taken care of then it has to be provided to the present model in a standard format after using some infra filters, as the temperature is very high. Secondly, this work has to be developed independently rather than using image processing and neural network toolboxes of MATLAB (2002) to increase the speed of operation. Further, recent advances in wavelet analysis in image processing can improve the performance of machine learning classifiers.

Conclusions: The work presented here is a first step, where a system with limited capability for offline data is developed which can handle certain kind of hot strip defects. The results have demonstrated that machine learning model – neural networks can handle and successfully remove some standard noises from the images. Further, the classification accuracy was found to be as high as 98.75% in leave one out exercise. General guidelines for developing machine learning model have been identified based on defect classification paradigm. Study showed that there is a wide scope of such models to be deployed for online defect detection and classification problems of hot strip mills.

References:

1. Anderson, J. (1995). *An Introduction to Neural Networks.* MIT Press, Cambridge, UK.
2. Bishop, C. (1995). *Neural Networks for Pattern Recognition.* Clarendon Press, Oxford, Birmingham, UK.
3. DLM (2002). *Defect library manual, (Private communication).*
4. Gonzalez, R.C., and Woods, R.E. (1992). *Digital Image Processing.* Addison-Wesley Publishing Company.
5. Gose, E., Johnsonbaugh, R., and Jost, S. (1997). *Pattern Recognition and Image Analysis.* Prentice Hall of India.
6. Haykin, S. (1999). *Neural Networks.* Pearson Education Asia.
7. IMPT (2002), *Image Processing Toolbox, User's Guide (version 2),* www.mathworks.com.
8. NNET (2002). *Neural Network Toolbox User's Guide (version 3),* www.mathworks.com.
9. MATLAB (2002). *MATLAB 60 User's Guide,* www.mathworks.com.
10. Rinn, R., Lorbach, N., and Lucking, F. (2002) *First automatic surface inspection for hot strip mills.* Parsytec Inc and Parsytec Computer GmbH. USA and Germany, www.parsytec.de.
11. Reich, Y., and Barai, S. V. (1999). *Evaluating machine learning models for engineering problems.* Artificial Intelligence in Engineering, Vol. 13, No. 3, pp: 257-272.