International Journal of Engineering Research
Volume No.4, Issue No.2, pp : 76 - 84
ISSN:2319-6890)(online),2347-5013(print)
01 Feb. 2015

Determination of Rate of Degradation of Iron Plates due to Rust using Image Processing

Priyanka Choudhary, Mr. Rohit Anand[2]
ECE, Kurukshetra University, N.C.college of engineering and technology,Panipat, Haryana, India
prianca.ece@gmail.com[1] , anandroh852@gmail.com[2]

Abstract: Most of industries and bridges around us make use of iron for manufacturing their products. On the other hand corrosion is a natural process that deteriorates the integrity of iron surface. Therefore, rusting of iron takes place. To avoid unwanted accidents in industries and bridges, it is necessary to detect rusting in earlier stage, so that it can be prevented. Digital image processing for the detection of the rusting provides fast, accurate and objectives results. In this research paper, we have done a systematic review of algorithms that help us to detect the rust area from a metal (iron).it has been found that most of researches are bring their images, processing series in usage for this purpose due to its simplicity in implementing and due to fact the images help capturing the visual inspection process easily and due to the ground teeth. The image processing techniques explored by other peoples based on in-depth analysis, we have also proposed a novel technique to overcome the limitation.

Keywords: Rust detection; Visual Inspection; Image processing

I. Introduction

Rust defect assessment is important in order to maintain a good quality of iron based fabrication painting. Rusting caused by corrosion causes wastage of iron materials, reduction in efficiency and costly maintenance [1]. So, Iron fabrications are more realistically to develop long-term cost-effective maintenance programs if they have dependable coating condition data. In order to detect rust defects in advance it is possible for engineers to initiate corrective action, whether to paint immediately or later or either replace the iron part or to add support like fish plates. Detection of onset of rust is relatively easier in machines, but difficult in bridges which may be located in remote locations and as such not accessible regularly for inspection. Otherwise also even if the bridges are located in populated areas certain portion of them may not be clearly visible thereby making detection of rust difficult [2]. So, due to unreachable location of rust formation either we use some moving robotic instrument or any stationary camera which continuously monitoring or after some interval the effective place respectively. In these cases image processing can be of immense help as rust formation and rate of decay can be calculated using the images captured at different intervals by digital camera or by moving robotic. Image processing refers to any form of signal processing in which input is an image, like a photograph or a video frame, the output of image processing may be either an image or characteristics or some parameters relating to the image[2].Image processing is carried out on a digital photograph or a video. Digitization includes sampling of image or video and quantization of sampled values. After converting the image into bit information, processing is performed. This processing technique may include some pre-processing (like Image enhancement, Image reconstruction, and Image compression, Image crop, Image Rotate) and post-processing steps (unsupervised clustering) which can help us to detect the rust better and in least possible time[3]. Each of this technique has advantages and disadvantages which are discussed in further sections and also there are multiple methods for detection of rust which are discussed in related work.

II. RELATED WORK

Most research has gone into identification of rust area detection using varied methods. One of them was Sindhu et al [1], he demonstrated the detection of rust on highway steel bridges using **Wavelet Domain Detection of Rust** Technique. It was a non-iterative approach based on the concept of wavelet transforms for the calculation of the rust percentage in the image. The method followed the concepts of principal component analysis and classification of rust and non rust part. Since, in this technique colored images are directly processed, therefore, there is no loss of information. They had implemented training and learning algorithm to classify a given image as a rust or non rust. Since it used the fuzzy logic which was very complex process as it required a lot of training images and all the images used had similar dimensions so because of this, some part of images remains undetected which is the one of the drawback. However Pidaparti *et al* [17] used an image analysis based on wavelet transforms and fractals to study the corrosion morphology of nickel aluminum bronze metal under varying corrosion conditions. Image feature parameters were extracted and analyzed to classify the pits/cracks in the metal samples. The results indicated that classification of pits/cracks is possible with image analysis and may be used for correlating service/failure conditions based on corrosion morphology.

MichikoYamana and Tohru Ohashi [5] have proposed idea about classification of rusted images with the help of **Support Vector Machines.** In this technique, the images that are taken by a digital camera are classified into “reuse” or “retire” on the basis of the color of the rust. The image taken by the digital camera is fed to the attached computer which compares the same with database images and the classifier function classifies the image as “Reuse” or “Retire”. However the results of this paper was promising their accuracy was 100 %,but work was limited by its

ISSN:2319-6890)(online),2347-5013(print)
01 Feb. 2015

learning process which was very time consuming and also once we had constructed the classification function, it cannot be change again and due to this it gives limited results.

Mariana P. Bento and Geraldo L. B. Ramalho [6] proposed an approach to detect corrosion of metals based upon **Nondestructive Evaluation (NDE) and Self Organizing Mapping (SOM) using Gray Level Co-occurrence Matrix (GLCM**) for the detection of the change in texture of metal surfaces. Further, Self Organizing Mapping (SOM) is used for the classification of the images as rusted or non-rusted areas of metals. In this experiment, 93% of validation data set was correctly classified but it was a complex process which required multistep methodology. Choi and Kim [13] have also used Co-occurrence matrix for texture analysis. To calculate corrosion surface damage color they used the interpretation of Hue Intensity Saturation (HIS) model. For the texture attributes, the method of co-occurrence matrix was used. Five types of corrosion damage were examined by the author. Multidimensional scaling procedure was used to define the classification plane. The study suggested a probabilistic method of decision-making for that. Zaidan *et al* [15] have also used texture analysis technique for the detection of corrosion in metals.

Besides image processing, other methods have also been used for detecting corrosion. Grinzato et al[16] have used transient infra red (IR) thermography for hidden corrosion detection in thick metallic components. Silva *et al* [18] have used laser ultrasonics and wavelet transform signal analysis for hidden corrosion detection in aircraft aluminum structures.

The potentiality of image processing techniques for automatic rust steel detection was investigated and the methodology introduced an iterative multivariate data analysis to examine the effects of rust steel descriptors, that was texture and color distribution on a set of classification algorithms. In this analysis, a selector of classifiers indicated that algorithm provides good classification results (high sensitivity) and acceptable time response for the automation of the system.

In 1981, Itzhak et al employed computer image processing techniques for statistical evaluation of pitting corrosion in a plate of AISI 304L stainless steel exposed to a corrosive water solution containing 10% Iron (III) Chloride. The purpose of this work was to introduce and to evaluate new tools for analyzing the effects of pitting corrosion process [15]. The algorithm was capable of estimating the number and area of pits in the binary image and therefore provided better evaluation of pitting corrosion damages.

A popular image processing algorithm for texture analysis extracts features from the gray level co-occurrence matrix (GLCM). In study conducted by Medeiros et al [14], the power of these features to deal with the stochastic pattern of corrosion for damage detection in metallic surfaces has been explored. Parameters extracted from the GLCM were used to define similarity properties for corrosion detection purpose in image segmentation methods based on region approach. This approach consists in determining the regions that contain neighbor pixels in the image that have similar properties, that is, gray level and spatial relationship. Two GLCM parameters, namely contrast and energy, are considered to be the most efficient for discriminating different textural patterns.

A wide variety of literature works had reported that texture features are proper to characterize corroded surfaces. In addition, typical color changes of metallic surfaces are often related to corrosion. Thus, color attributes carry out relevant information to design corrosion detection systems and help in building better.

III. Methodology

After conducting a systematic review and after studying all other possible resources we propose a methodology that covers up the limitation of previous work done and it can be summarized into the following steps:

1. Image Capturing
2. Read Image
3. Convolution of images
4. Run De-noising filter
5. Run Contrast enhancement and Contrast stretching
6. Image Sharpening
7. Run K-means Machine Algorithm
8. Calculate Rust statistics
9. Calculate Rate of Rust Spread
10. Neural Network
11. Result

Image Acquisition & Capturing

Images of the object under study were captured by digital camera SX20IS with specification 12.1 megapixel , 2.8-inch type charge coupled device(CCD) having optical and digital zoom 4x and low light sensitivity .These captured images will then be processed for the detection of the rust.

Read Image

After capturing the RGB images which is of size 256*256. The images are then read by function as a matrix and after reading the images some pre processing steps like image cropping and image rotate (if necessary) are done.

Convolution

After reading an image, now we convolve the two images to obtain a resulting image of same dimension. Convolution of images is important in this paper because we want to convert it into frequency domain to make calculations easy and to get idea of frequency response. [7].

Mathematically we can write the convolution as:

(i)

where i runs $$O(i,j) = \sum_{k=1}^{m} \sum_{l=1}^{n} I(i+k-1, j+l-1) K(k,l)$$

ISSN:2319-6890)(online),2347-5013(print)
01 Feb. 2015

from 1 to $M - m + 1$ and j runs from 1 to $N - n + 1$.

De-noising filter

Now after convolving, De-noising [12] aims at suppressing as much as possible of noise that perturbs a signal or an image. This noise accounts for measurement imperfections (captors of bad quality, quantification noise, etc) and is often model as a Gaussian white noise. Mathematically we can calculate the Gaussian Noise as:

$$h(t) = \frac{\exp\left(\frac{-t^2}{2\delta^2}\right)}{\sqrt{2\pi}\cdot\delta}$$

where

$$\delta = \frac{\sqrt{\ln(2)}}{2\pi BT}$$

(ii)

where B is the filter's 3-dB bandwidth

Contrast Enhancement and Contrast stretching

After removing the noise, we improve the perceptibility of objects in the image by enhancing the brightness i.e., the difference between objects and their backgrounds. Contrast enhancements are typically performed as a contrast stretch followed by a tonal enhancement, although these could both be performed in one step [8] and after enhancing we improve the contrast in an image by `stretching' the range of intensity values it contains to span a desired range of values. It differs from the more sophisticated histogram equalization in that it can only apply a linear scaling function to the image pixel values. As a result the `enhancement' is less harsh [10]. For enhancing the brightness we use a gamma factor that lies in between 0and1. so,if gamma value<1= the image is darkened

if gamma value> 1= the image is brightened overall (iv)

Image Sharpening

To get the crisp boundary and crisp edges, we do image sharpening which is one of the most impressive transformations. This will be applying to an image to bring out image detail that was not there before [9]. Paradoxically, the first step in sharpening an image is to blur it slightly. Next, the original image and the blurred version are compared one pixel at a time. If a pixel is brighter than the blurred version it is lightened further; if a pixel is darker than the blurred version, it is darkened. The result is to increase the contrast between each pixel and its neighbours.

K-means Algorithm

K-means (MacQueen, 1967)[11] is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. The procedure follows a simple and easy way to classify a given data set through a 4 clusters. The aim of this algorithm is to grouping the pixels of clusters. In this algorithm we are using RGB images and predominately our images having gray and black colour more than the red and orange colour .More less we try to make 4 types of clusters. Therefore, gray or black part comes under one cluster and light red or dark red in another, then we were able to cluster these group of image pixels into four kinds of clusters representing rust , non-rust and other parts which do not qualify as proper rust and non-rust part and eliminate the remaining ones by using k means and create a logical image which having particular pixels represent rust. Therefore in order to calculate the rate of spread we subtract the whole area from that.

Calculate Rust statistics

Calculating *Total Rust Area*

Now, the total rusted area of the surface will be calculated. The step will be performed to make sure that the images are partially rusted or totally rusted. Depending on the area found rusted important decisions are to be made, whether to repair or discard.

Therefore area can be calculated as:

Non rust area = total count of pixels – total count of rust pixels

Rust area = total count of pixels – total count of non-rust pixels (v)

Calculating the Rate of Decay

Rate of decay of the metal will be calculated using time series analysis. Images captured at different times will be compared for reduction in thickness and then rate of decay will be calculated .

Artificial Neural Network (ANN)

Artificial Neural Network (ANN) has been successfully used classifier in numerous fields. So, it is of interest to use it for use identification defective gears . It can be modeled on a human brain. The basic processing unit of brain is neuron which works identically in ANN. The neural network is formed by a set of neurons interconnected with each other through the synaptic weights. It is used to acquire knowledge in the learning phase. The number of neurons and synaptic weights can be changed according to desired design perspective. The basic neural network consists of 3 layers.

- Input layer: The input layer consists of source nodes. This layer captures the features pattern for classification. The number of nodes in this layer depends upon the dimension of feature vector used at the input.
- Hidden layer: This layer lies between the input and output layer. The number of hidden layers can be one or more. Each hidden layers have a specific number of nodes (neurons) called as hidden nodes or hidden neurons. The hidden nodes can be varying to get the desired performance. These hidden neurons play a significant role in performing higher order computations. The output of this layer is supplied to the next layer.

ISSN:2319-6890)(online),2347-5013(print)
01 Feb. 2015

- Output layer: The output layer is the end layer of neural network. It results the output after features is passed through neural network. The set of outputs in output layer decides the overall response of the neural network for a supplied input features.
- Selection of Real Valued Input Feature Vectors

Selection of real valued input features vectors that influence defect identification was done based on identification of critical parameters which influence the pattern of these defects(flash, warping, bubbles, unfilled sections, sink marks, ejector marks etc.) in terms of their spectrogram and coherence . Since neural network consist of three layers. This step basically implements the source nodes of neural network. The number of node in this layer depends upon the dimensions of feature vectors used at the input layer.

All feature vectors must have strong association relationship with the function variable f(*y*). The feature vectors normally are independent and the dependent variable is f(*y*) must have some relationship with the predictor (type of defects).In case some relationship exists between feature vectors and predictor their co-variance must be also close to one.

Neural Network Hidden Layer Number Detection

A major problem in designing a neural network is about establishing the optional number of hidden layers. Input and output layers are normally clear and easy to read, showing hidden layer to get the accurate result is designing issue of the classifiers. We have used the process called 'Grow up'.

In this process a layer is added after performing empirical testing of neural network for achieving better level of results. We designed the optimal architecture of a neural network with different architectures and making them learn on the input dataset created by simulating the red to traffic data. This was carried on until the performance criterion was fulfilled.

The typical structure of neural network is shown in Figure 1.1 below which consists of m input neurons in general and n hidden neurons with single hidden layer. The output layer has only three neurons. The network is called as fully connected network when all the neurons are connected with the adjacent neurons

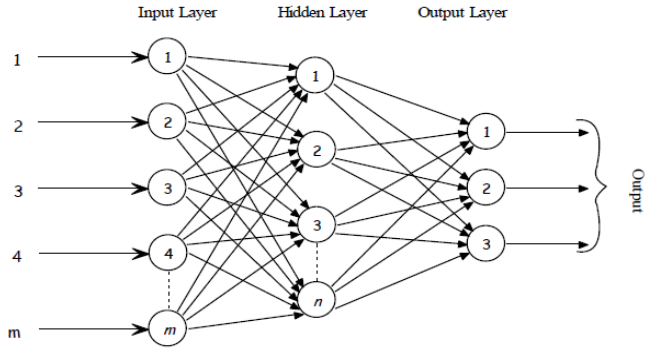

Figure 1.1 Typical Neural Network

Initialization of Weights

All the weight and biases are set to small read random values between 0 and 1 using twister random algorithm. This algorithm is also called the Mersenne twister is a pseudorandom number generator developed in 1997 by Makoto Matsumoto and Takuji Nishimura that is based on a matrix linear recurrence over a finite binary field. It provides for fast generation of very high-quality pseudorandom numbers, having been designed specifically to rectify many of the flaws found in older algorithms.

Its name derives from the fact that period length is chosen to be a Mersenne prime. There are at least two common variants of the algorithm, differing only in the size of the Mersenne primes used. The newer and more commonly used one is the Mersenne Twister MT19937, with 32-bit word length. There is also a variant with 64-bit word length, MT19937-64, which generates a different sequence.

For a k-bit word length, the Mersenne Twister generates numbers with an almost uniform distribution in the range [0, 2^k-1]. The advantage of using this algorithm are as follows :

The Mersenne twister is the default random number generator for Python, Ruby, R, PHP , MATLAB and also available in C++ since C++11, therefore we choose this random generator for giving weights to our neural network classifier .However for our research work, all the weights and biases are set to small real random values between 0 and 1.

Choosing the Appropriate Learning Method; In artificial neural network the learning methods are classified into three basic types.

a) Supervised Learning
b) Unsupervised Learning
c) Reinforcement Learning

Supervised Learning

It incorporates an external teacher, so that each output unit is told what its desired response to input signals ought to be. Learning process is based on comparison between network's computed output and the correct expected output, generating error. Depending on the deviation weight adjustment is determined. Figure 1.2 explains the supervised learning process. As shown in the diagram the input features are applied to neural net which is then given to a comparison box that compares the net output and target. The error vector is then applied to a supervised learning algorithm box which is feedback to neural net.

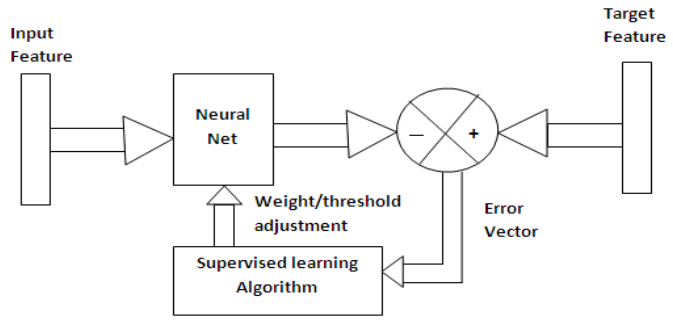

Figure 1.2 Supervised Learning Process

b) Unsupervised learning:

Unsupervised learning discovers features in a given set of patterns, and organizes the patterns accordingly. There is no specified desired output in this case. This type of learning uses mostly local information to update the weights. It is also referred to as self-organization, in the sense that it self-organizes data presented to the network and detects their emergent collective properties.

ISSN:2319-6890)(online),2347-5013(print)
01 Feb. 2015

c) Reinforcement Learning

This is output based learning. In this type of learning a teacher is present but does not present the expected or desired output. It only indicates if the computed output is correct or incorrect. The information provided helps the network in its learning process. A reward is given for correct answer and a penalty for a wrong answer.

The Validation Phase

When the training is complete, we want to check the network performance and determine if any changes need to be made to the training process, the network architecture or the data sets. The default generalization feature for the multilayer feed forward network is early stopping. The error on the validation set is monitored during training, and the training is stopped when the validation increases over net. Train.Param.max_fail iterations.

The provisional classification mining model is then adjusted to minimize the error rate on the test set.

The adjusted data classification model is then applied to a validation data set, another holdout data set, where the values of the target variable are again hidden temporarily from the model. The adjusted model is itself then adjusted, to minimize the error rate on the validation set. Estimates of model performance for future, unseen data can then be computed by observing various evaluative measures applied to the validation set.

Testing Phase

After satisfactory training error level obtained, the training was terminated; test metrics from each group were presented alternatively to the network without weight adjustment to test the performance. The network calculated the output of this test pattern with values of weight, bias and other parameters determined during training phase. The output was then compared with the desired output to check if the model performs well and minimizes to the goal, then present design is finalized otherwise the classifier is retrained by changing its parameters.

Output layer Design

The design of output layer depends upon the number of different defects (flash, warping, bubbles, unfilled sections, sink marks, ejector marks etc.) or classes as labeled below:

Table 1.1 Number of Defects types

Target	**Defect Class**	**Target Pattern (NN)**
Class A (Non-defective)	Defect A	1 0 0 0 0
Class B(Defective)	Defect B	0 1 0 0 0

Design of Optimized Classifier

After completing the design of all the layer of the NN, we finally implement the complete design setting, the challenge remains us to find which design will give maximum accuracy with minimum possible performance error. Therefore, we design about 13 designs to find which is the most optimized, accurate and one which use less resources.

Table 1.2 Training Parameters (NN) used in our research work

S.No	Type	Default Values	Description
1	Epochs	100	Maximum number of epochs to train
2	Goal	0	Performance goal(MSE)
3	Maximum failure	5	Maximum validation failures
4	Memory reduction factor	1	Factor to use for memory/speed tradeoff
5	Minimum gradient	1e-10	Minimum performance gradient
6	Mu	0.001	Initial mu
7	mu_dec	0.1	mu decrease factor
8	mu_inc	10	mu increase factor
9	mu_max	1e10	Maximum mu

Table 1.3: Neural Network Classification Algorithms used in experiment to design highly optimized defect classifier

Classifier Design No	**Algorithm**	**Number of Hidden Layers**	**Accuracy**
1	LM	3	67
2	LM	5	68
3	LM	7	77
4	LM	8	79
5	LM	10	90
6	Gradient Descent with Momentum(traingdm)	3	55
7	Gradient Descent with Momentum(traingdm)	5	65
8	Gradient Descent with Momentum(traingdm)	6	70
9	Gradient Descent with Momentum(traingdm)	8	73
10	Gradient Descent(traingd)	3	62
11	Gradient Descent(traingd)	7	68
12	Gradient Descent(traingd)	9	73
13	Gradient Descent(traingd)	10	78

Back propagation Training Algorithms

a) Levenberg-Marquardt (trainlm)
b) Gradient Descent with Momentum(traingdm)
c) Gradient Descent(traingd)

Levenberg-Marquardt (trainlm)

Back propagation algorithm utilizes the Levenberg-Marquardt algorithm for training of the network. The 'trainlm' is a network training function that updates weight and bias values according to Levenberg-Marquardt optimization. This training function is often the fastest back propagation algorithm in the toolbox, and is highly recommended as a first-choice supervised algorithm, although it does require more memory than other algorithms.The

ISSN:2319-6890)(online),2347-5013(print)
01 Feb. 2015

Levenberg-Marquardt consists basically in solving **(H + λI) δ = g** with different **λ** values until the sum of squared error decreases. So, each learning iteration (epoch) will consist of the following basic steps:

1. Compute the Jacobian (by using finite differences or the chain rule)
2. Compute the error gradient
   $g = J^t E$
3. Approximate the Hessian using the cross product Jacobian
   $H = J^t J$
4. Solve **(H + λI)δ = g** to find **δ**
5. Update the network weights **w** using **δ**
6. Recalculate the sum of squared errors using the updated weights
7. If the sum of squared errors has not decreased, discard the new weights, increase **λ** using **v** and go to step 4.
8. Else decrease **λ** using **v** and stop.

Variations of the algorithm may include different values for **v**, one for decreasing **λ** and other for increasing it. Others may solve **(H + λdiag (H)) δ = g** instead of **(H + λI) δ = g**, while others may select the initial **λ** according to the size of the elements on H, by setting λ_0 **= t max (diag (H))**, where **t** is a value chosen by the user. Identity matrix equation can be chosen .There can be a problem if the error does not decrease after some iteration. In this case, the algorithm also stops if **λ** becomes too large.

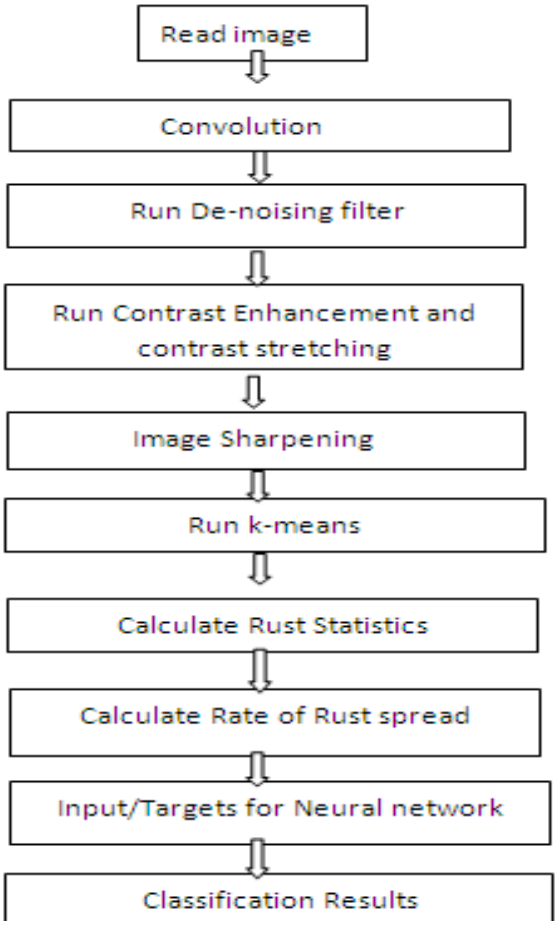

Fig 1.3: Block schematic of rust detection

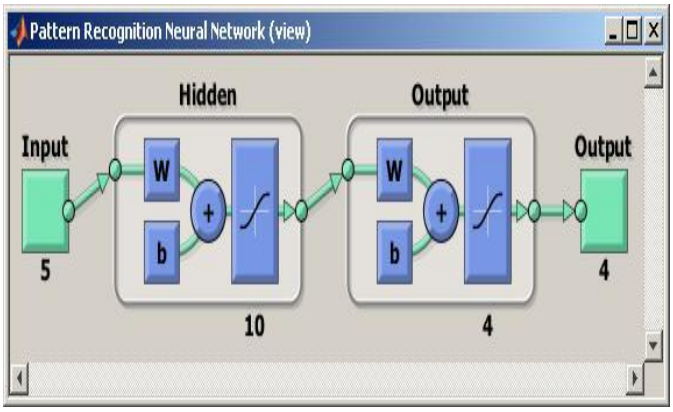

Figure 1.4 MATLAB based view of the Neural Network Architecture Defect identification

IV. **Results:**

In this section, we shall discuss the result we have built which allows us to achieve the objectives including the detection of rust area in metal using simple digital camera and finally find the rate of rust and non rust pixels using K-means and we will explore all the possible relevant results that we need to evaluate to really assess the accuracy of the rust defect recognition system, the main criteria for measuring the accuracy of the system was develop following performance graphs, The neural network is responsible for making intelligent classification based on observations made for various types of gear defects. In our research work we have taken observations as spectrogram and coherence properties of rust and non-rust of the metal and here are the findings with respect to our work:

a) Mean Square Error Value graph in all Phases (Training, Testing and Validation :

This performance graph helps use to find , how much close and successfully the neural network has been able to achieve with respect to fitting the data towards performance goal of zero mean square value , mathematically Mean Square Error (MSE) is explained as follows :

Mean Squared Error is the average squared difference between outputs and targets. Lower values of (MSE) indicate better performance of the network and zero means no error.

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(e_i)^2 = \frac{1}{N}\sum_{i=1}^{N}(t_i - a_i)^2$$

Where, t – Target
a - Actual output
e – Error
N – Number of exemplars

Interpretation of MSE graph: As is evident, we can observe that the best validation performance is 0.1042 at epoch 2. In simple terms this means that the neural network does not need to learn about the grear defects further than 8 iterations. Since the objective of neural network is to run simulation until it reaches the minimum possible mean square error (Lower values of (MSE) indicate better performance of the network and zero means no error). We can apparently observe that there is not

ISSN:2319-6890)(online),2347-5013(print)
01 Feb. 2015

much variation of mean square error beyond 8 epoch. It is more or less study graph after that.

b) Data Fitting R square graph in all phases (Training , Testing and Validation)
c) Confusion Matrix (Overall):
d) Classification matrix

Classification matrix is a specific table layout that allows visualization of performance of an algorithm that may be supervised or unsupervised in its learning, each column of the matrix represents instances in predicted class, while each row represents instances in actual class. We have taken care of giving a balanced dataset (observations) in terms of creating a dataset of equal proportions of samples of color and texture features. Therefore we can rely on the accuracy of our designed neural network based classifier.

For this purpose comparison between target and network's output is done in testing set using various parameters to estimate the classifier performance. The performance of classifier is analyzed using confusion matrix. Also known as table of confusion, it displays the number of correct and incorrect predictions made by the model compared with the actual classifications in the test data. It is an *n*-by-*n* array showing relationships between true and predicted classes, where *n* is the number of classes. In the field of artificial intelligence, a confusion matrix is a visualization tool typically used in supervised learning. Each column of the matrix represents the instances in a predicted class, while each row represents the instances in an actual class. One benefit of this matrix is that it is easy to see if the system is confusing two classes (i.e. commonly miss labelling one as another). A confusion matrix contains information about actual and predicted classifications done by a classification system. Performance of such classification systems is commonly evaluated using the data in the matrix. A confusion matrix can be made by using values of True Negatives, False Positives, False Negatives and True Positives. These are the standard terms for performance analysis of a classifier. They represent the four different possible outcomes of a single prediction for a two-class case with classes "1" ("yes") and "0" ("no"). A false positive is when the outcome is incorrectly classified as "yes" (or "positive"), when it is in fact "no" (or "negative"). A false negative is when the outcome is incorrectly classified as negative when it is in fact positive True positives and true negatives are obviously correct classifications. Thus, false negative and true negative rates are complements of true positive and false positive rates, respectively. Figure 5.1 shows the confusion matrix for three class classifier for the metal defect

Output

From the classification matrix as shown in Table 5.1it can be seen that the lower triangular matrix shows the number of misclassifications, while the upper triangular matrix shows the correctly classified metal defects. In fact, for calculating the true positive rate we sum up the total number of observations that fall diagonally along this matrix. For each other phase (training, testing and final simulation) we have developed the same type of classification matrix and the accuracy values as shown below in the table.

Sr. No	Phase	Overall accuracy
1	Training	81
2	Validation	80
3	Testing	

Classification Accuracy

The classification accuracy is the extent to which the classifier is able to correctly classify the exemplars and is summarized in the form of confusion matrix to the test data. This is defined as the ratio of the number of correctly classified patterns (TP and TN) to the total number of patterns (beats) classified.

$$Accuracy = \frac{Number\ of\ beats\ correctly\ classified}{Total\ Number\ of\ samples}$$

Or

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Interpretation of Results

The strength of our algorithm is that it identifies nature of defect classification errors as well as their quantities so that the round truth is properly matched with the dedicated results. Once this metal defects frame work is put into practice. Therefore after designing multiple classifiers with various possible parameters of input observations, hidden layers and fixed no. of output classes. We have tried to build a low computational resource intensive as well as less time consuming framework to detect the various types of possible metal defects.

The selection of parameters for the design of classifier has been meticulously and empirically found after many experiments. The appropriate selection of initial weights for the learning function was found by using Twister Random Algorithm. So that the coverage is maximum and it occurs rapidly (-0.5 and +0.5). If initial weights are too small then net input to hidden or output unit will approach 0 which would have led to slow learning but if weight were too large the initial input signal to each hidden or output unit would fall in the saturation region where the derivative of the activation function (sigmoid) would have very small value 0.

The selection of learning rate was also done keeping in mind changes in weight factor must be small in order to reduce oscillations or any deviation. For deciding the training and testing patterns. We developed disjoint sets of training and testing datasets and got these validated using K4 cross validation method. In all the various deigns of classifiers the major focus was also to identify the no of hidden units care was taken that no unnecessary additional computational resource usage comes into play for each additional hidden layers. Therefore, hence we finally got the observation collected by using image processing for identifying the various classes of defects automated with the help of neural network based machine learning algorithms.

Conclusion Based on Machine Learning :

The defective metal image was collected from the standard data base of a manufacturing unit. The defect identification is sometimes not possible with human eyes or sometime may be

ISSN:2319-6890)(online),2347-5013(print)
01 Feb. 2015

ignored due to human error. We have used the image processing tools to identify the defective metal image signal.

- ❖ It was found that the feature extraction by coherence and spectrogram descriptive statistical features gives the best performance and more number of features can be extracted using methods explained in methodology section.
- ❖ Data Normalization must be used to reduce the number of samples and the complexity of the neural network and the computation time of the neural network.
- ❖ For the classification schemes, it was found that training the model with a large number of test data and with fast training algorithm would greatly enhance the accuracy and hence the reliability of the system.
- ❖ The design of our classifier was done by running the neural network with different number of hidden layers and it was apparent from the bar graphs that it affected the accuracy.
- ❖ It was found that as we increase the number of hidden layers there was also an increase in computation time but high order of accuracy is also achieved until we have reached the maximum of hidden layers.

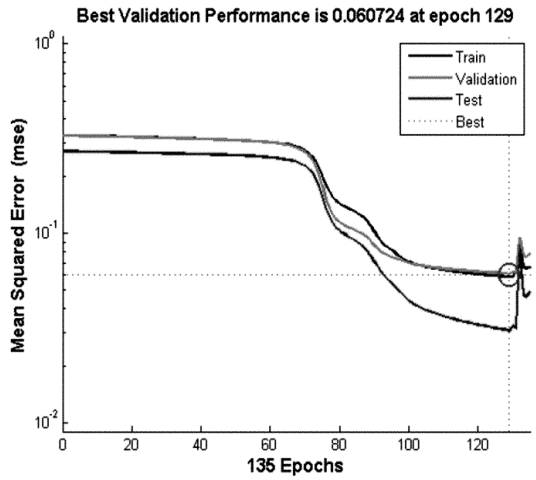

Fig: 1.5 graph b/w mse and epochs

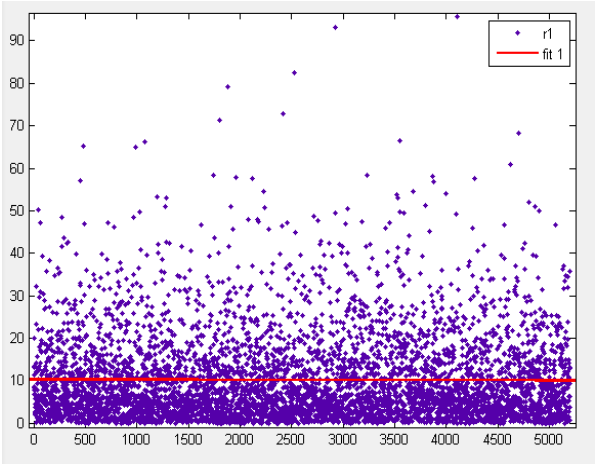

Fig: 1.6 rust and non rust on plate 1

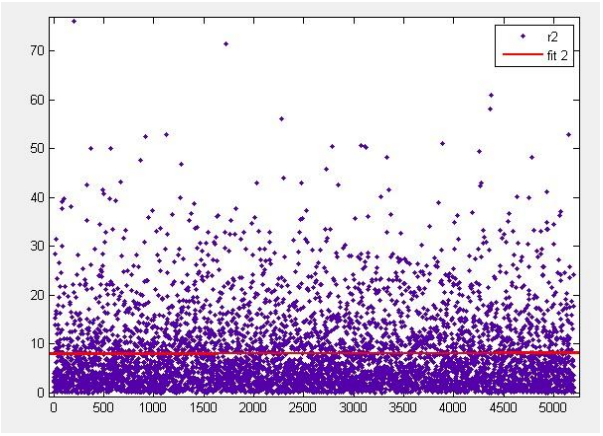

Fig: 1.7rust and non rust on plate 1

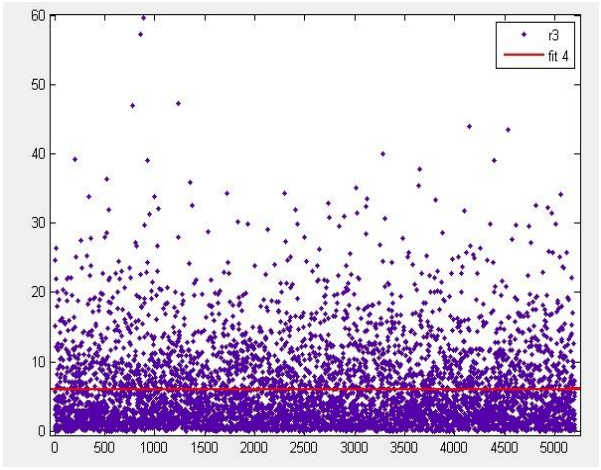

Fig: 1.8rust and non rust on plate 1

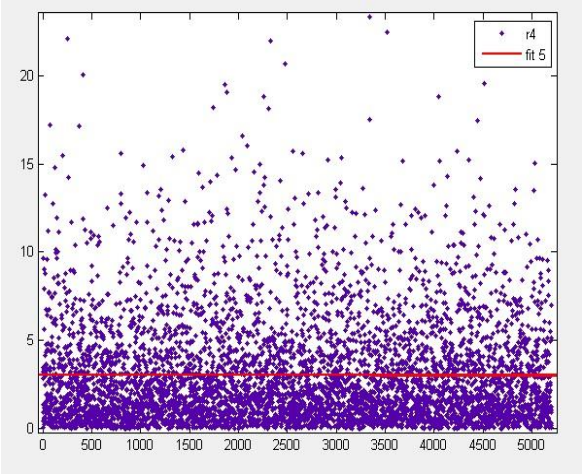

Fig: 1.9 rust and non rust on plate4

V.Conclusion

This paper helps us to detect the rust area from a metal (iron).in this paper we are using unsupervised machine algorithm, with the advantage of having large number of variables, it is

ISSN:2319-6890)(online),2347-5013(print)
01 Feb. 2015

computationally faster (if K is small) and also it may produce tighter clusters, especially if the clusters are globular. In this research work, images are captured right from the acquisition stage undergoes certain steps where rate of gradient of the pixels having rust is manipulated in such a way that it leads to reconstruction of new image from which a logical image is built to calculate the rust and non rust part.

VI. Future Scope

For future scope, we suggests to develop a representative data set of images depicting rusting of iron at various stages and make a mathematical model depicting rust growth and metal decay using unsupervised machine learning and thresholding to evaluate the validity and performance using time series analysis.

VI. REFRENCES:

i..*Sindhu Ghanta, Tanja Karp, Sangwook Lee "WAVELET DOMAIN DETECTION OF RUST IN STEEL BRIDGE IMAGES" 978-1-4577-0539-7/11/$26.00 ©2011 IEEE.*

ii.http://www.sciencedirect.com/science/article/pii/S0041624X02004559

iii.http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=5946583&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D5946583

iv.http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=1562969&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D1562969

v.Michiko Yamana, Hiroshi Murata, Takashi Onoda, Tohru Ohashi ,Seiji Kato "Development of System for Crossarm Reuse Judgment on the Basis of Classification of Rust Images Using Support Vector Machine" Proceedings of the 17th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'05)1082-3409/05 $20.00 © 2005 IEEE

vi. Mariana P. Bento, Fatima N. S. de Medeiros, I´alis C. de Paula Jr,Geraldo L. B. Ramalho "Image Processing Techniques applied for Corrosion Damage Analysis"

vii. http://homepages.inf.ed.ac.uk/rbf/HIPR2/convolve.htm

viii. Chapter 9 Image Enhancement Processing - http://spie.org/samples/TT92.pdf

xi. Sharpening Images Written by Jonathan Sachs Copyright © 1996-1999 Digital Light & Color

x.file:///C:/Users/dell/Documents/Point%20Operations%20%20Contrast%20Stretching.htm

xi. file:///C:/Users/dell/Documents/Clustering%20-%20K-means.htm

xii. Lecture 3 De-noising Using Linear Filtering https://www.ceremade.dauphine.fr/~peyre/teaching/wavelets/tp3.html

xiii. http://www.sciencedirect.com/science/article/pii/S0010938X0400126X

xiv. http://asp.eurasipjournals.com/content/2010/1/817473

xv. http://www.ijcsi.org/papers/IJCSI-Vol-7-Issue-3-No--1.pdf#page=41

xvi.http://www.sciencedirect.com/science/article/pii/S135044950600082X

xvii. http://www.sciencedirect.com/science/article/pii/S0010938X10003550

xviii. http://www.sciencedirect.com/science/article/pii/S0041624X02004559