

Inspection System for Rail Surfaces Using Differential Images

F. J. delaCalle, Daniel F. Garcia*, Rubén Usamentiaga*

*Department of Computer Science and Engineering, University of Oviedo, Campus de Viesques
33204 Gijón, Asturias, Spain, Email: delacalle@uniovi.es

Abstract—In this paper a surface inspection system for rails is proposed. There is a lack of commercial systems and publications about surface inspection of rails. Therefore, the quality control developed by rail manufacturers depends on their own developed systems or on the equipments provided by very few sellers. Commercial systems must be configured manually by experts of seller companies. The configuration process requires large sets of all types of manufactured rails along several periods of time and the active participation of the quality engineers of the manufacturer. This is a long, cumbersome and very expensive process. In this paper we propose a new system that can be configured using a systematic method that can be performed by the quality engineers of the manufacturer. The proposed inspection system uses differential images of the rail surfaces obtained with a technique called the Spectral Images Differentiation Procedure. These images are processed using a computer vision algorithm that looks for variations in the pixel values among the images. In order to offer more information, a neural network approach is used for classifying detected defects into six types. The proposed system is an open solution for the inspection of rail surface, easy to implement at an affordable cost, which can be systematically configured by the quality engineers of the manufacturing company.

Index Terms—Long steel products, Rail inspection, Surface inspection, Defect detection, Differential images

I. INTRODUCTION

The steel industry needs quality control systems which can ensure high reliability and efficiency in order to control production. The manufactured rails must follow strict quality standards about flatness, profile and surface quality.

Surface inspection is performed automatically with diverse types of systems. These systems give information to quality inspectors about the location of defects and suspicious regions on rails and also information about them such as their shape or dimensions. Due to the speed on modern industrial production lines, surface inspection cannot be carried out manually; New reliable and efficient inspection systems are needed in order to ensure product quality.

Each time a defect is detected, a quality inspector must find the defect in the proper rail to check that its dimensions do not exceed the allowed tolerances. A system which detects each minimum imperfection on the surface as a defect, even when it is not, causes an overload of work for inspectors and lowers their productivity.

Surface defect detection can affect not only one rail in particular but the whole the production. Some types of defects are produced by protrusions or scratches in rolling mills. Early

detection of these kinds of situations may help in maintenance of the infrastructure.

This paper presents a new surface inspection system for rails based on the processing of differential images obtained from the surface of rails by a spectral image differentiation procedure. The parameters of this systems are configured with a systematical procedure that can be easily carried out by quality engineers of the manufacturer without specialized personal. In fact, the systems is configured according to the industrial needs of the manufacturer paying special attention on the amount of erroneous detections. The proposed system is compared with a commercial system available in the same factory improving the ratio of real defects detected and reducing the number of erroneous detections.

The rest of this paper is organized as follows: in Section II some of the most relevant related works about defect detection are introduced. In Section III a global vision of the proposed system is given. In Sections IV and V methods of acquisition and processing of images are described. In Section VI a method for optimally configuring the parameters of the system is described. Finally Sections VII and VIII give experimental results and conclusions.

II. RELATED WORK

Surface inspection based on computer vision techniques is suitable for diverse kind of materials and industries. Many papers study this kind of image processing algorithms and techniques applied to different fields [1][2][3]. These algorithms and techniques are frequently used in defect detection processes for metallic products [4] but they are also used in other fields such as fabric [5], phone screens [6] or food [7].

Surface inspection based on computer vision can be done with different techniques. One of these techniques uses lasers which project lines over the surface. These lines are captured by cameras and the resulting images are processed to produce a three dimensional reconstruction of the piece as it goes through the production line [8][9]. This technique requires expensive hardware and periodical maintenance due to the laser degradation but it provides an accurate measure of the defects.

The most traditional surface inspection technique uses one light source and gray level cameras to get images of the surface. The resulting gray level images are processed to detect variations of the gray level which can suggest the existence of

defects [6][10][11]. Using this technique, volumetric defects such as protrusions or seams cannot be measured and a large number of false detections are introduced in the diagnosis due to color variations in the product surface or irregular illumination.

Using more than one light source, well placed for image acquisition, a 3D reconstruction can be obtained. This kind of technique is based on photometric stereo algorithms [12] which use only one camera and several light sources to produce a 3D reconstruction of the piece using the shadows in the images. The obtained images give information about height variations on the surface using several images. If light sources are of different colors (red, green, blue), only one standard color camera is needed to obtain the image corresponding to each light source [13][14][15]. In fact, this technique is suitable for inspecting flat products as steel strips [16] due to the easy way of placing the different light sources. However, when the whole surface of very long products must be inspected, this technique has a problem with the position of the light sources in order to illuminate the entire surface homogeneously while generating the needed amount of shadows.

In many cases, surface inspection is applied to small pieces or flat products. Using these techniques in large products such as rails [13] or wood [17] requires surrounding the pieces with cameras and light sources or other type of sensors [9] in order to acquire images of the whole surface. This requirement makes these applications more complex as final images must be pieced together from the images acquired by each camera and some overlapping may be needed. The lack of published works or commercial systems about this kind of application forces industries to develop their own inspection systems or buy them from the very few enterprises that offer them.

In terms of processing the resulting images, there are three typical approaches: traditional pipeline with image processing algorithms, machine learning approach with elements like Convolutional Neural Networks (CNN) or an hybrid approach. The first approach has already proved its good performance in several fields as many algorithms can be used depending on the type of image that must be processed [5][18]. In this approach there are many parameters used by the algorithms that must be configured in order to give a good result. In many cases an ad-hoc pipeline of algorithms is needed to solve a particular problem. In some particular applications a final step of classifying the detected defects is needed, so in this cases a machine learning element is usually used for this task [19] if it cannot be solved using some easy rules according to some defect features.

The second approach of processing is commonly used for detecting defective pieces in a production line where all must be similar [20]. However this approach it can also be used on more complex images [21][22] so it can classify product images they contain defects or not what can be applied to long products giving a diagnosis meter by meter. Using this approach, the image can be the input of a CNN or similar. The developing of this kind of method includes only the design of the CNN and the creation of its training set, but no further

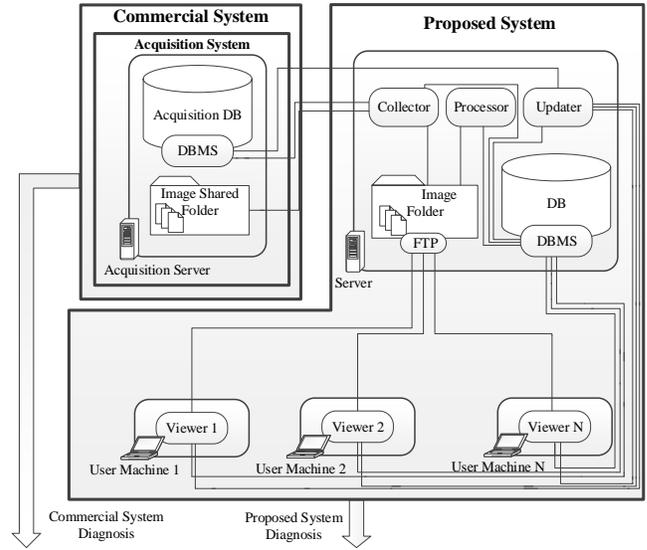


Fig. 1: System Architecture

configuration of parameters or feature selection is needed.

The third approach of processing is an hybrid approach in which some of the techniques of the first one are applied to the images in order to make defects more visible and increase the performance of the machine learning approach[23].

III. PROPOSED SYSTEM

The proposed inspection system is composed of two systems working in cascade: a commercial image acquisition system, which uses the technique proposed in [13], and an image processing system. The commercial system performs its own image processing and gives a diagnosis. Both systems works at the same time on the same rails and, as the image acquisition system is part of the commercial system, on the same images.

The image processing system is divided in four independent modules which must collect the images from the acquisition system, process the images, update information on demand, and allow users to view rail information and defects. A viewing module has been designed in order to allow several users to view the information simultaneously. Figure 1 shows the architecture of the proposed system.

IV. IMAGE ACQUISITION SYSTEM

The acquisition system uses the technique of spectral image differentiation. This technique is used to detect volumetric defects on the surface such as protrusions or scratches.

This technique works as follows. Two light sources are placed before and after a line scan camera. These light sources may be of different colors; in this case red and blue light sources are used. Flat surfaces will reflect the same amount of light from both sources, independently of their texture. Surfaces with volumetric defects will reflect different amounts of light from each source due the shadows that defects create. Using a color camera it is possible to divide the image in two

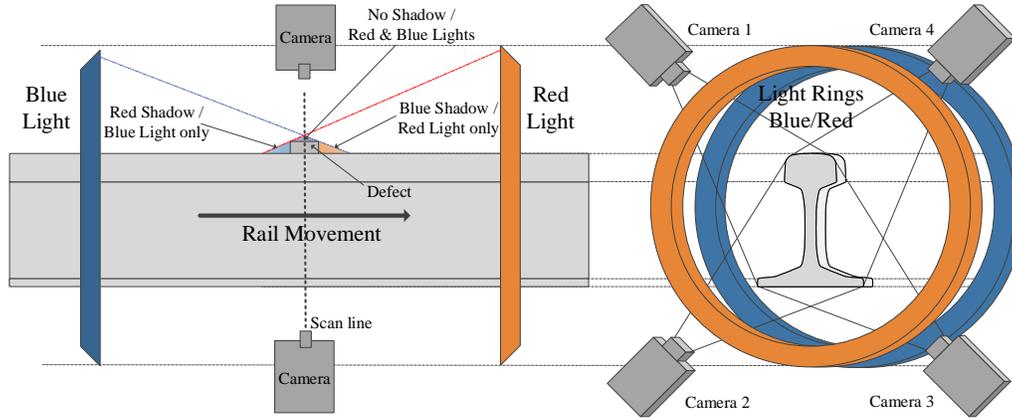


Fig. 2: Image Acquisition Method Scheme

channels, red and blue, and compare the light reflected from each source. Assuming that both light sources generate the same amount of light over the surface and that the rail has the same reflective properties all over its surface, the difference between the red channel and the blue channel will be zero on flat surfaces and non-zero on defects.

In Figure 2 a scheme of this system with a defect is shown. On the acquisition at instant T_0 the pixel values of red and blue channels will be similar. On the acquisition at instant $T-1$ the pixel values of the red channel will be higher than the blue ones because the defect creates a shade for the blue light on the surface of the rail. On the acquisition at instant $T+1$ the situation will be the opposite. This method can detect rises and falls on the surface through the rail movement direction.

Using this technique, the acquisition system provides differential images generated by the differences between the red and blue channels of the image captured by the color camera. Subtracting the values of the red and blue channels pixel by pixel, flat surfaces will have a near zero value. Otherwise, depending on whether the change is a rise or a fall, the value will be positive or negative but always different from zero. The position of the light sources determines whether positive or negative values represents rises or falls in the surface.

The resulting images contain values in $[-127, 127]$. In order to make them more suitable for user visualization, a value of 127 is added to all the pixels of the image. This gives a gray level image as output. These images still give the same information about the surface but flat surfaces will have a value of near 127, while the rest will have values closer to 0 or 255. Although these images give information about height variations, they do not give a quantitative measure about the variations. These images will be called "differential images" in this paper.

V. IMAGE PROCESSING SYSTEM

Image processing has two main objectives: defect detection and defect classification. Defect detection is performed using several computer vision algorithms working on differential images. Classification uses neural networks as classifiers.

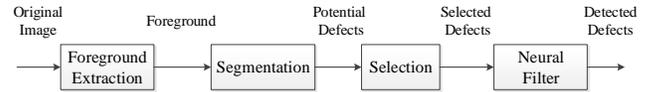


Fig. 3: Defect Detection Steps

On each image provided by the acquisition system, a sequence of steps is performed to detect defects in this particular type of images, see Figure 3.

In the first step the foreground of the image is extracted. Using a filter, defects can be easily differentiated from the background due to the kind of images that the acquisition method produces. As the shape of the rail is complex, the median gray level is not exactly 127 in all zones of the image, so an adaptive filter should be used instead of a simple subtraction of the mean value of the image.

In a second step, a segmentation must be performed for locating the potential defects in the image. Due to the type of images, a thresholding is selected over other methods as region growing or clustering as it fits better the needs of detection. Usually each defect is represented as two zones (white/rise and black/fall), and a closing operation is performed for joining them into one region per defect.

After these steps are done, all defects should be detected generating also so many erroneous detections so some filters are needed. First, in selection step, some erroneous detections are filtered using some features extracted from them and from their environment. This selection is useful for the next step, so it filter some defects that otherwise should be included into the set of potential defects degrading the neural networks training. After that two neural networks, specialized in some types of erroneous detections, are used to filter more erroneous defects.

A. Foreground Extraction

The differential images provided by the acquisition system contain zones with a small displacement from zero, positive

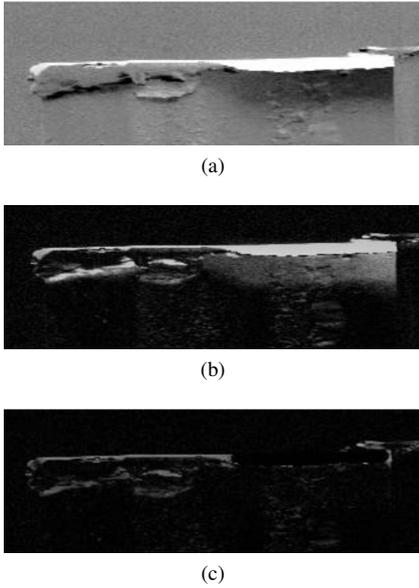


Fig. 4: Effect of Median Filter Mask. (a) Original image (b) Effect of large mask (c) Effect of small mask

or negative. These zones appear along both the longitudinal and the transversal direction. In order to work with the image uniformly, these displacements must be eliminated.

A differential image centered in zero value can also be obtained. In this kind of image the foreground is represented by pixels whose values tend to ± 127 . To perform this step, a median filter has been designed. The median of the environment of each pixel is subtracted from the pixel value. The environment of each pixel is defined by a mask whose size defines the intensity of the filtering. If the mask is too large, the foreground is retained, but the small displacements from zero of the pixel values are not eliminated. If the mask is too small, foreground is also filtered. These two situations are shown in Figure 4.

Subtracting the median from the pixel values, a differential image is obtained with negative and positive values. The relevant information for defect detection is the absolute value of the pixels. In terms of defects in rails, a deviation in the surface height is characterized by its absolute magnitude for both rises or falls. This characterization is useful for the next step, in which the absolute values of the pixels of the image are used. Thus, images in Figure 4 must be interpreted as follows: black represents pixels whose value is near zero and white or gray represent pixels whose value is significantly different from zero.

B. Segmentation

Segmentation is done by thresholding and morphological operations. Thresholding is performed on the foreground extracted from the differential image in the absolute value. The value of the threshold is one of the main values that determines the detection of defects. If this value is too low or too high, thresholding will produce an excess or lack of detections.

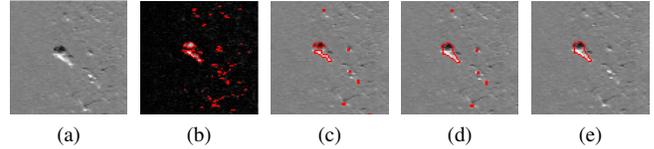


Fig. 5: Evolution of the segmentation of a rolled-in material. (a) Original image (b) After thresholding (c) After opening (d) After closing (e) After area filtering

After thresholding, several morphological operations are performed on the resulting pixels in the binary image.

First an opening operation is performed on the regions. This operation filters most of the noise in the image, eliminating the pixels with outlier values. After this, a closing operation is performed to join adjacent regions and to fill holes.

The closing operation is crucial due to the way that defects are represented in differential images; as a rise or a fall from the height level of previous pixel of the surface in the longitudinal direction of the rail. As long as the height level of the surface is constant in the longitudinal direction, the pixels will have zero values even if their height levels are greater than their surroundings in the transversal direction. Thus, any single defect will be detected in two parts that must be joined. The closing operation does this when the two parts of the defects are close enough.

Finally, an area filter is applied to the resulting regions. The area of a region is calculated as the count of its pixels. During this operation all the regions whose area is less than a minimum value are eliminated. This eliminates some of the noise that the opening operation could not filter due to the size of the noise region. This filter does not affect the real defects detected because these must have their size increased because of the closing operation.

In Figure 5 the evolution of the segmentation of a defect (a rolled-in material) during this process is shown.

C. Selection

After segmentation, the regions obtained are treated as potential defects. The main objective of this step is to choose those regions which represent real defects and discard those which represent rough textures (erroneous detections). For this purpose each potential defect is compared with the first result after the thresholding in the segmentation step called First Detection. Each potential defect is dilated to get a dilated region that contains both the defect and its environment. The environment is the result of subtracting the pixels in the defect from the set of pixels that defines the dilated region (1).

The relation between the number of the pixels filtered in each environment of a potential defect, I , and the size of the proper potential defect, D , can be used as the differentiating value. The value of I is calculated as the cardinality of the intersection between each environment and the first detection produced after segmentation thresholding (2). The value of D is the cardinality of the set of pixels that defines each defect

(3). Using these two values, all potential defects that do not satisfy (4) are eliminated. Thus, all the potential defects which are similar to their environment are filtered as noise.

$$Environment = DilatedDefect - Defect \quad (1)$$

$$I = \left| Environment \cap FirstDetection \right| \quad (2)$$

$$D = |Defect| \quad (3)$$

$$I < D \quad (4)$$

In the last operation in the selection step, the value of the pixels of each potential defect are analyzed in order to filter erroneous detections. If each pixel, i , has its own value, $p(i)$, then the volume of a Defect D can be defined as (5).

$$V = \sum_{i \in D} p(i) \quad (5)$$

Using this definition of the volume of a defect, a volume filter is designed to eliminate defects that do not reach a minimum value of volume. This filter eliminates potential defects that are not real defects but intensity changes in the images or residual noise.

After this step, the retained potential defects go through a new closing operation with the same purpose as in the segmentation step. Due to the lack of noise in the image, the radius of the closing operation is now larger and the parts of a defect can be properly joined.

D. Neural Filter

At the beginning of this step, the set of selected defects contains detections of engravings and erroneous detections, such as scales.

Engravings, embossed characters that identify a rail, are easily classified visually. They are imperfections on the surface, but they are needed and must not be considered as defects. They are also different from other types of erroneous detections. That is why engravings are filtered with an independent filter before the general filtering of erroneous detections.

Identifying an engraving automatically using a set of ad-hoc rules would make the detection method unstable and not automatically configurable. Because of this, a machine learning approach for classification is used. In this paper engravings and also erroneous detections are filtered using neural networks.

First, a specialized neural network filters engravings and then a second one filters the rest of erroneous detections. In Figure 6 both engravings and erroneous detections are shown. These regions are the targets of the neural networks.

In order to train the neural networks, 41 features are obtained from each instance. These features are divided in two sets [25].

The first set of features is based on the morphology of the region. In this set there are 14 features, among them are: length and width, center of the rectangular bounding box, both axis of the minor ellipsis that contains the region and its orientation, convexity, compactness, etc. This first set gives information

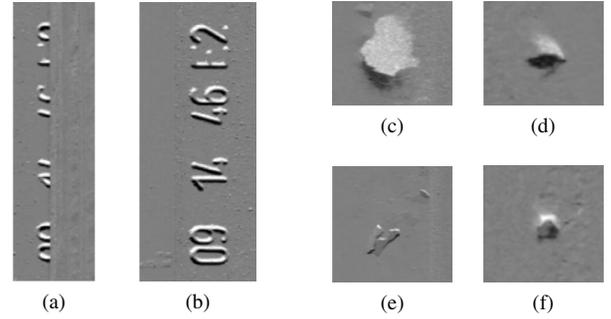


Fig. 6: Erroneous defect detections. (a-b) Engravings (c-f) Scales

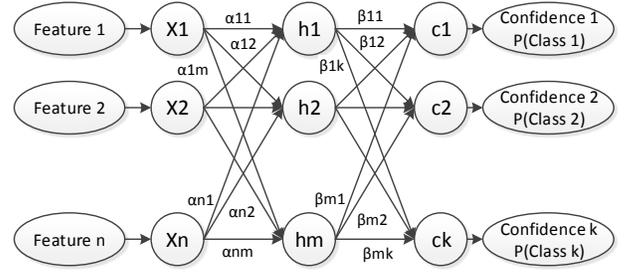


Fig. 7: Multilayer Perceptron

about where the defect is located and what shape it has. This is suitable for regions that likely appear on one particular zone of the rail as engravings that also should have a regular length and width depending on the image acquisition.

The second set of features contains 26 features that use the information of the gray level of the region. Features of this set gives information about the general appearance of the defect, as volume (5) or mean value, and also information about how the values are located using features as correlation, gravity center or homogeneity. This information can be useful for differentiate some False Positive that are very similar to some defects, specially rolled-in material.

The described set of features is the input to the neural networks which are multilayer perceptrons such as the one shown in Figure 7. A hyperbolic tangent is used as activation function in hidden layer nodes (6), where matrix α and vector b^1 are the weights of the input layer and the bias respectively.

$$h_j = \tanh\left(\sum_{i=1}^n \alpha_{i,j} x_i + b_j^1\right), j = 1 \dots m \quad (6)$$

The values of the hidden layer are obtained using a different activation function. This is a normalized exponential function shown in (7) and (8), where matrix β and vector b^2 are the weight of the hidden layer and the bias respectively.

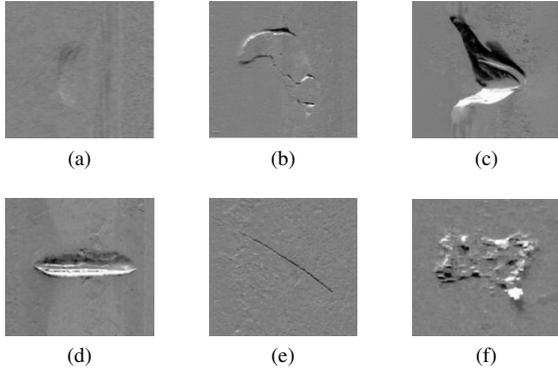


Fig. 8: Defect Types. (a) Roll Mark (b) Rolled-In Material (c) Lack of Material (d) Straightening Mark (e) Wire (f) Others

$$a_j = \sum_{i=1}^m \beta_{i,j} h_i + b_j^2, j = 1 \dots k \quad (7)$$

$$c_l = \frac{e^{a_l}}{\sum_{t=1}^k e^{a_t}}, l = 1 \dots k \quad (8)$$

The output layer is composed of k nodes. Each of these nodes represent one class. The output of each node is the confidence or probability of an instance belonging to that class. Thus, the sum of the confidences of all classes must be 1. The class with the highest confidence will be selected to classify each instance.

The described neural networks do not get the features as input but, rather the output of pre-processing them. To improve performance, a Principal Component Analysis (PCA) is done over the features in order to determinate which ones are most relevant for the classification process. From the 41 features, only n principal components will be used as input of the neural networks. This n is a parameter in neural network configuration.

E. Classification

After detection, the defects must be classified to offer inspectors all the information they might need.

Classification is done using a neural network with the same architecture used for neural filtering of engravings and erroneous detections. In this case, the neural network has as many output nodes as types of defects. Possible defect types are: roll marks, rolled-in material, lack of material, straightening marks, wire and others. In Figure 8 examples of these defect types are shown.

VI. PARAMETER CONFIGURATION METHOD

The defect detection method has several parameters that must be configured. These parameters are: median filter mask size, segmentation threshold, opening and closing radius, minimum value of area filter, minimum value of volume filter and neural network configuration.

Parameter configuration is done sequentially following the phases of detection method shown in Figure 3. The first parameters of the method, corresponding to the first phase, which are foreground extraction and segmentation, are configured with a factorial design. The parameters of the selection phase are configured analytically using the results of the previous phases. The neural filter is configured using an exhaustive exploration of the parametric space.

The classification neural network is configured using the same method used for the neural filter. These neural networks have two parameters: the number of nodes in the hidden layer and the number of principal components for PCA.

A. Evaluation metric

To configure the parameters of a method, a metric to evaluate the adequacy of the values given to the parameters is needed. In this case the metric estimates the goodness of a detection comparing it with a “perfect detection” which comes from a Knowledge Base (KB). In this work, the Knowledge Base is composed of images from 245 rails with 2017 defects produced in a factory of ArcelorMittal over four months in 2016. These images were checked by the quality inspectors of the company in order to assure that all defects were marked. Knowledge Base composition is shown in Table I.

Defect Type	Amount	Defect Type	Amount
Roll Mark	147	Straightening Marks	153
Rolled-in Material	1425	Wire	247
Lack of Material	19	Others	26
Total	2017		

TABLE I: Knowledge Base Composition

Comparison is done as an intersection between detected defects and marked defects in the Knowledge Base. If a region of a detected defect intersects with a defect of the KB, it is a True Positive (TP). If it does not it is a False Positive (FP). Using the same method if a defect marked in the KB does not intersect with any detected defect, it is a False Negative (FN). Thus the best performance is obtained by maximizing TP and minimizing FP and FN.

Using these values (TP, FP and FN), some metrics to estimate the goodness of a detection can be defined. Selected metrics must be coherent with the objectives of the system: detect the maximum number of defects and minimize erroneous detections. The fulfillment level of these objectives can be easily mapped to two metrics using TP, FP and FN values.

For the erroneous detections objective, the selected metric is the mean of the erroneous detections per rail obtained by processing the Knowledge Base rails. The lower the value of this metric, the better the detection.

A similar metric cannot be used to estimate the number of correct detections, because the metric must consider how many defects should have been detected. Therefore, a standard metric called Recall is used to do so. This metric is defined as the relation between what has been detected and what should have been detected (9). The higher the value of this metric, the

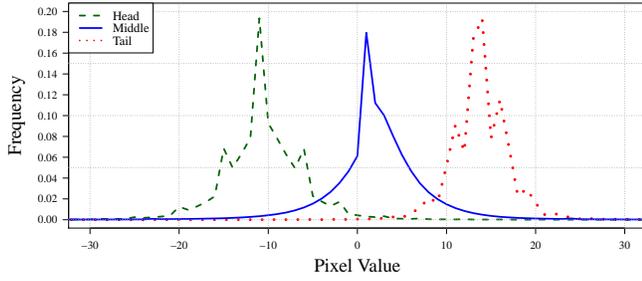


Fig. 9: Pixel value distribution in images of the Head, Middle and Tail of rails.

better the detection. Other common metrics that are used in the following sections are Precision (10) and F-Measure (11).

$$Recall = \frac{TP}{TP + FN} \quad (9)$$

$$Precision = \frac{TP}{TP + FP} \quad (10)$$

$$F_{\beta} = (1 + \beta^2) * \frac{Precision * Recall}{\beta^2 * Precision + Recall} \quad (11)$$

Detection of all defects ($Recall = 1$) leads to a large number of erroneous detections ($FP \gg 0$). Thus, a compromise must be found. When using two metrics at the same time, it is useful to represent them in a two dimensional plot to select the optimum configuration.

B. Foreground Extraction and Segmentation configuration

The configuration of the parameters of these phases is done in the same step due to their dependencies. The configuration of foreground extraction parameters directly influence the ones in segmentation, especially the threshold value.

The configuration of the size of the median filter must assure that its value captures the foreground of the image but does not filter any defects. Thus, a method to estimate an appropriate value is to look at the maximum size of any defect which has a radius of about 30 pixels in Knowledge Base. Based on this information, possible values of this factor in experimental design could be: 10, 30, 60 and 90 pixels. An extreme value is added in order to use the whole image size as the value for the filter.

The value of the segmentation threshold is directly related to the images. The background of the image has been eliminated in a previous step, so this value must represent how different a pixel value should be to be considered as a defect. This can be done by analyzing the distribution of pixel values in images. For this some images are selected including head, middle and tail rail images. Examples of density functions of these distributions are shown in Figure 9. According to these graphics, possible values for segmentation threshold in the experimental design could be: 20, 25, 30, 40, 60 and 70 in order to assure that background is not retained after thresholding.

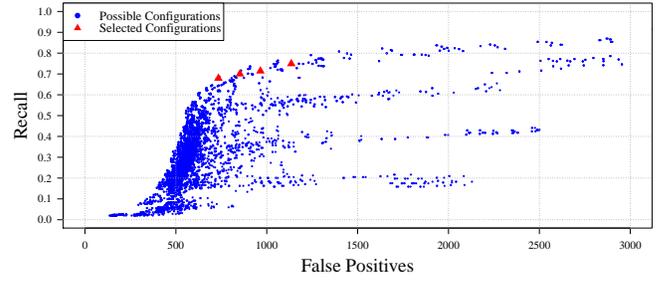


Fig. 10: Detection performance of all configurations

The value of the opening operation radius is kept as a constant established with ad-hoc tests. The closing operation starts working properly with radius values greater than or equal to 3, so 3, 6 and 9 are established as values for this factor.

The area filter must eliminate noise and preserve defects so the minimum area of the known defects must be used to determine the possible minimum values of the area filter. The smallest defects in the Knowledge Base is 73 pixels so appropriate values for this factor are 75, 100, 125 and 150 pixels.

Using these values for factors, the experiment is carried out and all possible configurations are evaluated using the rails in the Knowledge Base. After this, a graph for all possible configurations can be obtained representing Recall as a function of False Positives detected in the rails of the Knowledge Base. This graph is shown in Figure 10. It can be seen that in order to capture approximately an 80% of the defects detected, an overhead of nearly 3000 erroneous detections (approximately, 1.5 times more erroneous detections than correct detections) is needed.

From all the possible configurations, those with the maximum Recall value and low FP counts are selected. These can be obtained by analyzing the effect of each factor on the metrics. The selected configurations have a: median filter mask size of 60, a threshold value of [25;30], a closing radius of 6 and an area filter value of [75; 100]. These configurations also have maximum levels for F-Measure (11) using $\beta = 1$.

C. Selection Configuration

Using the regions of the images obtained from the selected configurations in the previous step, the distribution of the volume of True Positives and False Positives is analyzed.

In Figure 11 the cumulative distribution function is shown for TP and FP. According to this graph, a value of 8000 for the threshold of the volume filter should eliminate 10% of the FPs without filtering any TPs.

D. Neural filter Configuration

After the configuration of the parameters in previous phases, neural networks must be trained and their parameters must be configured. To do so, an exhaustive search is performed on their two parameters.

The number of principal components of PCA varies from 1 to 41 and the number of hidden nodes vary in the same range.

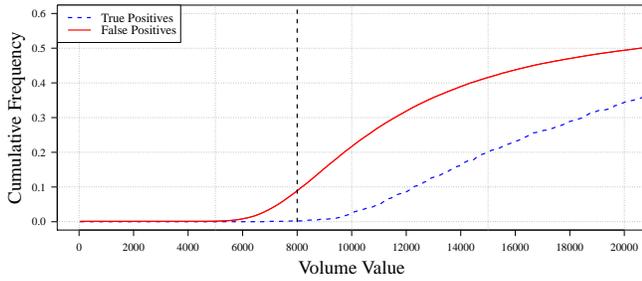


Fig. 11: Defect Volume Distribution

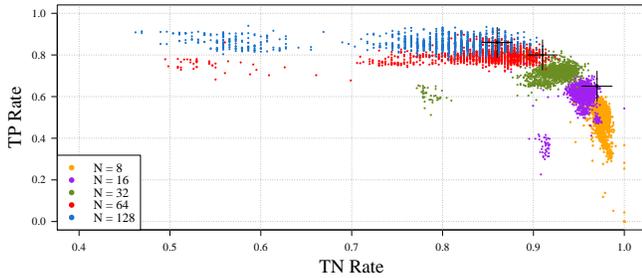


Fig. 12: Performance of Neural Network Configurations

In some studies the number of hidden nodes is selected as the 2-base logarithm of the number of training instances. As the optimization of this factor is still an unresolved problem [26], an exhaustive search is performed its configuration because it is computationally viable.

Initial results showed that the training set had to be modified to train the networks correctly. In this set, 98% of the instances are False Positives. This leads to an optimal configuration that classifies all instances as False Positives with a success rate of 98%. A situation of imbalanced classes is frequent in classification problems. There are several fix methods to modify the training set [27][28]. In this case, random undersampling is selected to fix the problem. This method selects one from N instances for training. In these experiments, N varies from 1 to 512. Each trained neural network is evaluated with the K-Fold Cross Validation method using 4 folds.

The results from all configurations of the networks are plotted in Figure 12 using True Positive and True Negative rates as axes. Each point of the figure represents the proportion of TP/TN classified correctly by a particular configuration of the neural network. The large crosses mark those selected. In this figure, several sets of points are represented for various N values. N is the value used for the undersampling method. Thus each combination of values of the two parameters of the neural net is represented several times depending on the training set used. This graph is used to select the configuration that maximizes the TN rate for filtering erroneous detections while keeping high enough the TP rate so as not to filter real defects.

To select the best neural filter for engravings and false positives, all configurations are plotted so that quality inspectors can decide which ones best fulfill their needs.

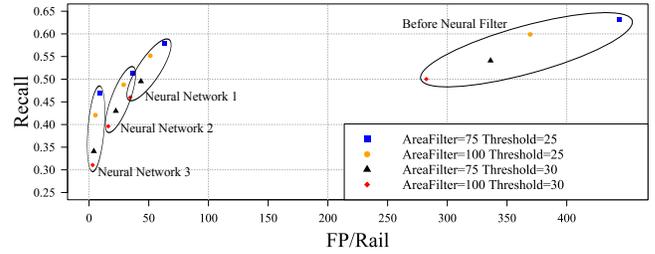


Fig. 13: Performance of proposed configurations

These configurations are shown in Figure 13. In this figure the initial selected configurations from Figure 10 are used. The performance of the configurations with no neural network is shown on the right side of the graph and the performance using three different neural networks for filtering false positives on the left side of the graph. The neural networks used are those marked with large crosses in Figure 12.

Using Figure 13 as an example, the configurations marked as blue squares are much better than the ones marked as red rhombuses in terms of Recall but they generate more erroneous detections (false positives). In order to select a configuration for the system, the quality inspector must make a compromise between the number of false positives and the Recall. Taking this into account, the best solution proposed is the blue square using Neural Network 3, which differs by only 0.1 from Neural Network 1 but also detects 55 fewer erroneous detections.

The final configuration must be selected by quality inspectors. These systems are very sensitive to the number of erroneous detections which is a very important value for quality inspectors. When the system detects a defect on a rail surface, an inspector must check the region of the rail manually in order to confirm that defect if it exists. Each erroneous detection leads quality inspectors to lose time and productivity.

E. Classifier Configuration

The configuration of the neural network that classifies the defects follows the same procedure as the networks that filter erroneous detections. From each defect 41 features are extracted. These features are the input of a PCA whose output is the input of the neural network. An exhaustive search is performed to configure both the number of hidden nodes and the number of principal components from 1 to 41.

All possible configurations are evaluated using a cross validation method with 4 folds. As the training set has only real defects, with no false positives, there is no need to undersample false positives instances.

In order to select a configuration, the rates of success in the classification of all the classes are calculated. Thus, the mean of these rates (MSR, Mean of Success Rates) gives information about the overall success rate without weighting the classes by their count of instances. The configuration with the maximum MSR value is selected.

In Figure 14 all the possible configurations are shown using their two parameters. In this figure, the color of each

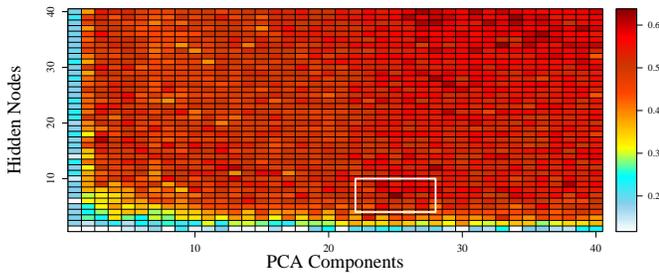


Fig. 14: Performance of possible configurations for classification

configuration represents the value of MSR. In this figure several configurations with a high MSR can be seen. Most of these configurations use a complex neural network with a high number of hidden nodes and high number of principal components. These complex solutions need all the information from all features to carry out an accurate classification. However, the one selected (marked in a white rectangle), needs much less information and gives better classification results.

The set of samples that compose the Knowledge Base can be seen in Table I. In this set, there is one class which has by far the largest number of samples, rolled-in material. This situation is similar to that of FP and TP instances. In this case, using a method like undersampling would waste real-defect instances with the consequent loss of information. The distribution of defect classes and the fact that there are more than two classes, allows the use of a metric like MSR to evaluate this neural network without undersampling. This metric weights all classes equally, so the set of rolled-in material instances has exactly the same weight as the set of any other class.

VII. RESULTS

Results obtained from the selected configuration can be validated by comparing them with the results of a commercial system over the same set of rails. The set used is the Knowledge Base.

The proposed system and the commercial system work simultaneously and the results of both are included in the viewer module of the proposed system to give quality inspectors both sets of detected defects. Table II shows the results of processing the Knowledge Base by these two systems.

Metric	Commercial System	Proposed System
Detected Defects	756	946
Total Defects	2017	2017
Recall	0.3748	0.4690
FP/Rail Mean	27.10	8.86
FP/Rail Median	24	7

TABLE II: Detection Results

The metrics of False Positives must be validated with a larger set of rails than the one in the Knowledge Base. To do so, the whole rail production of an ArcelorMittal factory over

four months was processed. This large set contains 24,869 rails. This validation provides similar results to the ones obtained with the Knowledge Base. In this case the value of metrics are 10 for the FP/Rail mean and 8 for the median.

The result of the defect detection, 46% of real defects detected, may be understood as a bad result even improving the commercial system results, but there are two reasons because it seems to be so low.

On the one hand, the aim of the parameter configuration is fulfill the needs of the industry. As it has been said before, quality inspectors loose time because of each erroneous detection. According to this, in Figure 13 the configuration represented by the blue square of Neural Network 3 is selected. However, the one of Neural Network 1 could be selected detecting 57.9% of the defects with a increment of 54 erroneous detections. Without the neural filter, 63% of the defects could be detected with an overhead of 444 defects per rail. Going at the beginning of the configuration, in Figure 10 the same election has been made, choosing configurations below 80% of defect detection instead of those near 90% for prevent the system to reach unacceptable erroneous detections rates.

On the other hand, roll-marks and rolled-in material, see Figure 8, are frequently too smooth in the images because of their low height or depth. This makes them difficult to differentiate from the background. This type of defects are part of the Knowledge Base even when distinguish them from the background is a hard task for the quality inspectors. This two types of defects can also be misclassified as scales, see Figure 6, because of their similarity. In order to improve the percent of defects detected without increasing the number of false positives, it would be needed to clean the scales from the surface before the inspection.

Classification evaluation can only be done by comparing the result of the system with the real defects manually classified by quality inspectors, which are those of the Knowledge Base. The results of this evaluation are shown in Table III. These results are not compared with a commercial system because it does not provide classification of defects. As a result of using the MSR metric to select a configuration for a neural network, all classes with a reasonable number of instances have an acceptable success rate.

Success Rate in	Value	Success Rate in	Value
Global	78%	Straightening marks	45%
Roll Marks	87%	Wire	67%
Rolled-in Material	86%	Others	44%
Lack of Material	51%	Mean	63%

TABLE III: Classification Results

VIII. CONCLUSION

This paper proposes a new surface inspection system for rails which uses images acquired with the Spectral Image Differentiation Procedure. System parameters are configured using a sequential method based on the phases of the detection method. This configuration method can be automated using a

training set of rails inspected manually by quality inspectors to compare the detections using standard metrics.

Using the configuration selected by the quality inspectors, the proposed system detects 46.9% of the existing defects in rails with an average overhead for quality inspectors of 10 erroneous detections per rail. These defects are classified into 6 different types with a global success rate of 78%. This value increases to 86 % for the most common defect, which is rolled-in material. The new system improves the results provided by an available commercial system, which detects 37.48% of the existing defects with an overhead of 27 erroneous detections per rail.

The number of erroneous detections is a critical value that can slow down the quality inspection speed of the factory because each erroneous detection must be checked by quality inspectors by hand. The configuration selected in this paper gives better performance than a commercial surface inspection system tested, detecting 9% more defects with 17 fewer erroneous detections per rail.

Surface inspection systems are developed and sold by very few companies, even fewer for those specifically designed for rails. The inspection performed by rail manufacturers depends on these commercial systems unless they develop their own systems. This work presents an open design for surface inspections systems that allows rail manufacturers to easily build and configure their own systems according to their needs in terms of real defect detection and erroneous detections.

The systematic configuration method described in this paper can be carried out easily by quality engineers of the manufacturer and is based on the user point of view of the system, which is focused in defect detection with the least amount of erroneous detections. In the last step of the method, quality inspectors and engineers much reach a compromise between the number of erroneous detections they can handle without slowing down the production speed, and the number of real defects detected.

ACKNOWLEDGMENT

The authors would like to thank quality inspectors and personnel of ArcelorMittal for their help in the development of this research. This work has been partially funded by the project TIN2014-56047-P of the Spanish National Plan for Research, Development and Innovation.

REFERENCES

- [1] W. H. Wan, "Product Quality Control and its Applications in Industry", Journal of System Simulation, 2001, 13, 154-155.
- [2] E. N. Malamas, E. G. Petrakis, M. Zervakis, L. Petit and J. D. Legat, A Survey on Industrial Vision Systems, Applications and Tools, Image and Vision Computing, 2003, 21(1) 171-188.
- [3] Z. Liu, H. Ukida, P. Ramuhalli and K. Niel, *Integrated Imaging and Vision Techniques for Industrial Inspection*, 2015, Springer.
- [4] F. M. Megahed and J. A. Camelio, *Real-time Fault Detection in Manufacturing Environments Using Face Recognition Techniques*, Journal in Intelligent Manufacturing, 2012, 23(3), 393-408.
- [5] M. Patil, S. Verma and J. Wakode, *A Review on Fabric Defect Detection Techniques*, International Research Journal of Engineering and Technology 2017.
- [6] C. Jian, J. Gao and Y. Ao, *Automatic surface defect detection for mobile phone screen glass based on machine vision*, Applied Soft Computing, 2017, 52, 348-358.
- [7] B. Zhang, W. Huang, J. Li, C. Zhao, S. Fan, J. Wu and C. Liu, *Principles, developments and applications of computer vision for external quality inspection of fruits and vegetables: A review*, In Food Research International, 2014, 62, 326-343.
- [8] X. J. Xue, C. K. Sum and S. H. Ye, *Laser Scanning On-Line Inspection System for Ball Grid Array Coplanarity Measurement*, Guangdong Gongcheng/Opto-Electronic Engineering, 2001, 28(1), 39-42.
- [9] Z. Xiong, Q. Li, Q. Mao and Q. Zou, *A 3D laser profiling system for rail surface defect detection*, Sensors, 2017, 17(8), 1791.
- [10] Y. Jeon, D. Choi, J. P. Yunm C. Park and S. W. Kim, *Detection of Scratch Defects on Slab Surface*, 11th international conference on Control, Automation and Systems in KINTEXm Korea, 2011.
- [11] F. G. Bulnes, D. F. García, F. J. de la Calle, R. Usamentiaga and J. Molleda, *A Non-Invasive Technique for Online Defect Detection on Steel Strip Surfaces*, Journal of Nondestructive Evaluation, Springer, 2016, 35(4), 1-17.
- [12] R. J. Woodham, *Photometric method for determining surface orientation from multiple images*, Optical engineering, 1980, 19(1), 139-144.
- [13] E. Deutchl, C. Gasser, A. Niel and J. Werschonig, *Defect Detection on Rail Surfaces by a Vision based System*, 2004, IEEE Intelligent Vehicles Symposium, 507-511.
- [14] O. Ikeda and Y. Duan, *Color Photometric Stereo for Albedo and Shape Reconstruction*, IEEE Workshop on Applications of Computer Vision, 2008.
- [15] F. J. delaCalle, Daniel F. García, Rubén Usamentiaga, *Inspection System for Rail Surfaces Using Differential Images*, In Proceedings IEEE Industry Applications Society Annual Meeting, Cincinnati, 2017.
- [16] L. Wang, K. Xu and P. Zhou, *Online detection technique of 3D defects for steel strips based on photometric stereo*, In Measuring Technology and Mechatronics Automation (ICMTMA), 2016, Eighth International Conference on (pp. 428-432). IEEE.
- [17] R. W. Connors, C. W. Mcmillin, K. Lin and R. R. Vasquez-Espinosa, *Identifying and Locating Surface Defects in Wood: Part of an Automated System*, IEEE Transactions on Patterns Analysis and Machine Intelligence, 1983, 6, 573-583.
- [18] Y. J. Zhao, Y. H. Yan and K.C. Song, *Vision-based automatic detection of steel surface defects in the cold rolling process: considering the influence of industrial liquids and surface textures*, The International Journal of Advanced Manufacturing Technology May 2017, 90(5), 16651678
- [19] Ga. Singh, Gu. Singh and M. Kaur, *Fabric defect detection using series of image processing algorithm and ANN operation*, Global Journal of Computers & Technology, 2016, 4(2).
- [20] M. Zhang, J. Wu, H. Lin, P. Yuan and Y. Song, *The Application of One-Class Classifier Based on CNN in Image Defect Detection*, In Procedia Computer Science, 2017, 114, 341-348.
- [21] D. Weimer, B. Scholz-Reiter and M. Shpitalni, *Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection*, In CIRP Annals, 2016, 65(1), 417-420.
- [22] T. Wang, Y. Chen, M. Qiao, et al. *A Fast and robust convolutional neural network-based defect detection model in product quality control*, The International Journal of Advanced Manufacturing Technology, 2017.
- [23] J. Ferguson, *Additive Manufacturing Defect Detection using Neural Networks*, Internal Report, 2016.
- [24] J. L. Rose, M. J. Avioli, P. Mudge and R. Sanderson, *Guided Wave Inspection Potential of Defects in Rail*, NDT International, ELSEVIER, 2004, 37(2), 153-161.
- [25] D. P. Tian, *A review on image feature extraction and representation techniques*, International Journal of Multimedia and Ubiquitous Engineering, 2013, 8, 385-395.
- [26] S. Xu and L. Chen, *A Novel Approach for Determining the Optimal Number of Hidden Layer Neurons for FNNs and Its Application in Data Mining*, 5th International Conference on Information Technology and Applications (ICITA), 2008.
- [27] N. V. Chawla, *Data mining for imbalanced datasets: an overview* Data Mining and Knowledge Discovery Handbook: A Complete Guide for Practitioners and Researchers, 2005, pp. 853867.
- [28] A. Ali, S. M. Shamsuddin and A. L. Ralescu *Classification with class imbalance problem: A Review* Int. J. Advance Soft Compu. Appl, 2015, Vol. 7, No. 3, pp. 176-204.