INSPECTION OF METALLIC SURFACES USING LOCAL BINARY PATTERNS

M. Mansano\textsuperscript{1}, L. Pavesi\textsuperscript{2}, L. S. Oliveira\textsuperscript{1}, A. Britto Jr.\textsuperscript{3,4}, A. Koerich\textsuperscript{1,3}

\textsuperscript{1}Federal University of Parana (UFPR), Curitiba, Brazil
\textsuperscript{2}Federal University of Technology - Parana (UTFPR), Curitiba, Brazil
\textsuperscript{3}Pontifical Catholic University of Parana (PUCPR), Curitiba, Brazil
\textsuperscript{4}State University of Ponta Grossa (UEPG), Ponta Grossa, Brazil

ABSTRACT

In this paper we propose the use of a texture feature called Linear Binary Patterns to inspect steal surfaces. To assess the proposed method, we have build two different databases. The first one contains 996 color images of steel bars illuminated with black light, where the defects were highlighted using penetrating liquid. The second dataset is composed of 1141 gray-scale images of steel bars without highlight. Comprehensive experiments using three different classifiers show that the proposed feature set is able to detect 91.5\% and 95.6\% of the defects on the first and second databases, respectively.

I. INTRODUCTION

The modernization of manufacturing processes made necessary the incorporation of techniques for evaluating the product at the end of its production. Computing is present in this process by replacing part of human attention with artificial intelligence and image processing, to automate the inspection and assessment activity in certain industry segments. In metallurgical industry the inspection of metallic surfaces is performed by experts in recognizing and identifying the gaps. They even determine the source of the problem and suggest fixes to production methods.

Most of the defects are imperceptible to the inspector, requiring auxiliary equipment to increase the accuracy of its recognition. According to Kang and Liu [1], humans detect between 60 and 75\% of the significant defects present in the inspected object. In order to improve the quality of inspection and reduce production costs, the automation of this service is being integrated into industrial production, mainly with the use of computer systems that use computer vision techniques.

Systems that monitor the products using computer vision can identify problems stemming from the production. The systems simulate the knowledge of an inspector, with the precision of human vision. Research is being developed in the area where the solutions found so far, show the effectiveness of the computing inspection systems used in the metallurgy industry of intermediate goods.

One of the great difficulties one has to face to develop a computer vision system for metal inspection is the lack of a public database. To overcome this problem, in this work we introduce two databases of steel bars. The first one contains 996 color images of steel bars illuminated with black light, where the defects were highlighted using penetrating liquid. The second dataset is composed of 1141 gray-scale images of steel bars without highlight.

In order to have a better insight about the databases, we have assessed different classifiers to discriminate between defects and faultless. The feature set used in this work is based on texture and extracted using Linear Binary Patterns (LBP). The classifiers considered in this work are the \textit{k}-Nearest Neighbors (\textit{k}-NN), Multi-Layer Perceptron (MLP), and Support Vector Machine (SVM). A series of comprehensive experiments shows that the SVM trained with LBP features is able to detect 91.5\% and 95.6\% of the defects on the first and second databases, respectively. It also shows that LBP is an interesting feature set that should be considered when building a system to inspect metallic surfaces.

II. RELATED WORKS

Ensuring greater efficiency in the inspection of metallic surfaces, several studies have been conducted in recent years. Practical applications involve the combination of pattern recognition with different fields of research in computer science, in order to help finding the solution of problems.

In the 90’s, Gunatilake [2] proposed the use of stereo vision and wavelets to identify defects in aircraft surfaces. It was even proposed differentiating defects from pseudo-faults, what is easily mistaken for human inspectors. Later, Jonker et al [3] extract PCA(Principal Component Analysis)-based features to rank six different defects found on metallic surfaces. The classifier used was an SVM.

Jig [4] proposed a real time system using an SVM to recognize small cracks of steel bars. The study was based on some characteristics like contrast, disparity, mean and variance of the grayscale in the image. These values were obtained from a region with a potential defect, after a series of pre-processing steps.
Kang and Liu [1] used a private database of strips cold rolled steel. With their database, they trained a forward neural network to classify defects from the production method. The observation was made by analyzing the image and checking the changes of the gray scale distribution of the pixels around a faulty region. So, they chose to use texture features from a windowed Fourier transform of $7 \times 7$ pixels.

Bonnot et al [5] used a technique that combines seven different images of the same region. The arrangement corresponds to a capture of an object part lighted with ring illumination with other six images of the same region, with only light rotation. He distinguished defects and pseudo-faults using several techniques of image processing. Recently, the method proposed by [6] uses a low-level analysis. It is purely statistics of the Gaussian profile in a defective region. The objective was identify defects that cause discontinuity in the metallic surfaces. Also with an own database and custom framework, the author could show the dependence between his method and the capture quality (good lighting and little noise).

Table I summarizes the results and datasets used by the aforementioned works. It is clear that a direct comparison is not possible since different datasets were used in each work. However, it allows us to have an idea of the performance of similar systems reported in the literature.

Table I. Summary of related works

<table>
<thead>
<tr>
<th>Ref.</th>
<th>Images in the Data Set (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>[2]</td>
<td>66 71.5</td>
</tr>
<tr>
<td>[4]</td>
<td>1226 94.4</td>
</tr>
<tr>
<td>[5]</td>
<td>4000 76.5</td>
</tr>
<tr>
<td>[3]</td>
<td>264 93.0</td>
</tr>
</tbody>
</table>

III. LINEAR BINARY PATTERNS

Ojala et al [7] outlined a model to describe texture called Local Binary Patterns (LBP). In this model each pixel $C$ contains a set of neighbors $P$, equally spaced at a distance of $R$ and $C$, as shown in Figure 1.

Fig. 1. The LBP operator. A pixel $C$, dark circle in the middle, and its neighbors $P_n$, lighter circles.

An histogram $h$ is defined by the texture intensity differences of $C$ and its neighbors $P$. When the neighbors do not correspond to an image pixel integer value, its value is obtained by interpolation. An important characteristic of this descriptor is its invariance to changes in the value of the average intensity of the central pixels, when compared with its neighbors.

Considering the resulting sign of the difference between $C$ and each neighbor $P$, it is defined that: if the sign is positive the result is 1 otherwise is 0. Thus, it is possible to obtain this invariance of the intensity value of pixels in grayscale format. With this, the LBP value can be obtained by multiplying the binary elements for a binomial coefficient. So, it is generated a value $0 \leq C' \leq 2^P$ (corresponding to the vector).

Observing the non-uniformity of the vector obtained, [7] introduced a concept based on the transition between 0's and 1's in the LBP image. It was described that a binary LBP code is considered uniform if the number of transitions is less than or equal to 2, also considering that the code is seen as a circular list. That is, the code 00100100 is not considered uniform, because it contains four transitions. But the code 00100000 is characterized as uniform because it has only two transitions. Figure 2 illustrates this idea.

Fig. 2. LBP uniform pattern. (a) the two transitions showed identifies the pattern as uniform. (b) with four transitions, it is not considered a uniform pattern.

So, instead of using the whole histogram, which size is $2^P$, it is possible to use only the uniform values, constituting a smaller feature vector. This version of the descriptor was called “u2”, a label accompanying the values of the radius $R$ and the neighborhood size $P$, making the LBP definition as follows: $LBP_{P,R}^{\text{label}}$.

It was also defined a version of the descriptor rotation invariant, called “riu2”. But in this job we decided to use the uniform descriptor “u”. It was because we observed that it was not interesting to consider the rotation invariance, since the defects found have longitudinal pattern and the images are extremely close to them.

Furthermore, we observed during the experiments that the feature extraction with $LBP_{8,2}^{u2}$ is fast and accurate enough for the proposed application. Then, we chose to use $P = 8$.
and \( R = 2 \) on the tests described in this paper.

**IV. DATABASES**

As stated before, two different databases were built to validate the proposed system. The first database (DATA 1) contains 996 images of \( 50 \times 50 \) pixels, captured from an industrial inspection process of steel bars, with 486 images containing defects and 510 without defects. The images of this database have a purple background and are highlighted by green fluorescent penetrating fluid. The highlighting of the defect from the bar is acquired by irradiating black-light over the bar surface after immerse it on a specific liquid.

![Fig. 3. Samples of DATA 1. (a) and (b) regular images and (c) defect.](image)

As we can notice, there is a bigger intra-class variability in the class of the regular images (non-defect). The class of the non-defect has images with homogeneous texture (Figure 3a) but also features images with irregular textures, e.g., Figure 3b. The class of defects generally has a non-homogeneous texture, like the one depicted in Figure 3c. We will discuss later in this paper that this lack of homogeneity for the class of non-defect is the main source of confusion between defect and non-defect.

It is worth of remark that the highlighting of the defects helps the inspector to classify types of defects and their gravity. But to be used in this study, the bar images were converted to gray-scale. It was done because in gray-scale it would be a simplified representation of the defects.

The second database (DATA 2) contains 1141 images, being 524 with and 617 without defects. The images were processed for use in gray-scale, as done with the previous images. The surfaces on this database have greater homogeneity than the former one. It is because there is no emphasis found on the defects (with penetrating fluid) and the capture was made in an ideal environment (pictures do not show shadows nor even noise). Some samples of this dataset are shown in Figure 4.

![Fig. 4. Samples of DATA 2. (a) and (b) regular images and (c) defect.](image)

As we can notice, there is a bigger intra-class variability in the class of the regular images (non-defect). The class of the non-defect has images with homogeneous texture (Figure 3a) but also features images with irregular textures, e.g., Figure 3b. The class of defects generally has a non-homogeneous texture, like the one depicted in Figure 3c. We will discuss later in this paper that this lack of homogeneity for the class of non-defect is the main source of confusion between defect and non-defect.

It is worth of remark that the highlighting of the defects helps the inspector to classify types of defects and their gravity. But to be used in this study, the bar images were converted to gray-scale. It was done because in gray-scale it would be a simplified representation of the defects.

The second database (DATA 2) contains 1141 images, being 524 with and 617 without defects. The images were processed for use in gray-scale, as done with the previous images. The surfaces on this database have greater homogeneity than the former one. It is because there is no emphasis found on the defects (with penetrating fluid) and the capture was made in an ideal environment (pictures do not show shadows nor even noise). Some samples of this dataset are shown in Figure 4.

Alike DATA 1, this dataset also have images without defects that are similar in some way to those images with defects. See for example Figures 4b and 4c. Figure 4b does not have the same homogenous texture as Figure 4a, but it belongs to the same classes, i.e., non-defect.

**V. EXPERIMENTAL ANALYSIS**

An important aspect in pattern recognition problems that is very often neglected is the class distribution. A tacit assumption in the use of recognition rate as an evaluation metric is that the class distribution among examples is constant and relatively balanced. In the database proposed in this work this is not the case. In this context, ROC (Receiver Operating Characteristic) curves are attractive due to its property of being insensitive to changes in class distribution. If the proportion of positive to negative instances changes in a test set, the ROC curves will not change [8]. For this reason we present the ROC curves and AUC (Area Under de Curve) values for all the experiments. The AUC has an important statistical property: the AUC of a classifier is equivalent to the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance.

As stated before, three different classifiers were used to assess the aforementioned feature sets: \( k-\)NN, MLP, and SVM. For the SVM, different kernels were tried out, but the Gaussian kernel produced the better results. The kernel parameters, \( \gamma \) and \( C \), were defined empirically through a grid search. All the experiments were performed considering a three-fold cross validation. Regarding the MLP, different configurations of networks were tried out, but the one that provided the best results contains three layers with 30 units in the hidden layers. The learning rate used for training was defined empirically.

For all reported results, we used the following definitions of the recognition rate and error rate. Let \( T \) be a test set with \( n \) string images. If the recognition system, classifies correctly \( N_{rec} \), and misclassifies the remaining \( N_{err} \), then

\[
\text{Recognition Rate} = \frac{N_{rec}}{n} \times 100
\]

\[
\text{Error Rate} = \frac{N_{err}}{n} \times 100
\]

In the first experiment we have used the three aforementioned classifiers (\( k-\)NN, MLP, and SVM) trained with the 59-dimensional \( LBP_{8,2} \) feature vector. The databases were
divided into 60%, 10%, and 30% for training, validation, and testing, respectively.

Table II shows the recognition rates and AUC (Area Under the Curve) achieved for the three classifiers on DATA 1. As we can see, the SVM outperforms the other two classifiers by a large margin. It is worth of noticing, that the nature of the SVM, which is a binary classifier by nature, fits well for the problem considered in this work.

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Rec. Rate (%)</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>k-NN</td>
<td>81.0</td>
<td>0.87</td>
</tr>
<tr>
<td>MLP</td>
<td>81.3</td>
<td>0.89</td>
</tr>
<tr>
<td>SVM</td>
<td>91.5</td>
<td>0.97</td>
</tr>
</tbody>
</table>

Table II. Performance of the classifiers on the test set of DATA 1.

Figure 5 compares the ROC of the three classifiers. As expected, the best trade-off between False Positives (FP) and True Positives (TP) is produced by the SVM classifiers as well.

Figure 5. ROC curves for the classifiers trained with GLCM features.

By analyzing the results provided by the SVM classifiers, we could notice that the majority of images with excessive penetrating fluid into a horizontal line or a set of small circles were misclassified as defectives.

The second strand of experiments were performed on DATA 2 using the same three classifiers trained with LBP. Again, the SVM outperformed the other two classifiers in terms of recognition rate. Table III reports all the results for DATA 2. However, in this case the MLP outperforms the SVM for low False Positive rates, as depicted in Figure 6.

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Rec. Rate</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>k-NN</td>
<td>89.0</td>
<td>0.93</td>
</tr>
<tr>
<td>MLP</td>
<td>93.3</td>
<td>0.97</td>
</tr>
<tr>
<td>SVM</td>
<td>95.5</td>
<td>0.98</td>
</tr>
</tbody>
</table>

Table III. Performance of the classifiers on the test set of DATA 2.

Figure 7 shows some False Positives, i.e., some defects detected by the classifiers that are not classified as defects by human inspectors. Figures 7a and 7b are misclassified images from DATA 1 while Figures 7c and 7d are misclassified images from DATA 2. In both cases the pseudo-defects are highlighted in red.

Fig. 6. ROC curves for DATA2

Fig. 7. False positives produced by the classifiers for DATA 1 (a and b) and DATA 2 (c and d)

VI. DISCUSSION

While many other metallic surfaces inspection systems only analyze the defect in a structural way, we show that using a texture descriptor faults can be identified, separating them from pseudo-defects and the surface. In the experiments with the two databases, we obtained quite different results. With the first database the difficulty was in determining if there was or not a defect in the midst of all interference caused by the penetrating liquid, that confuse the classification system.

When considering several features based on texture, a region with strong bright green pigmentation absorbed be-
comes a defect for the classifier. Pseudo-defects end up absorbing much of the penetrating fluid, as well as defects. This leads to confusing the classifier. Moreover, the surface roughness of the metal part just accumulates fluid.

There is a variety of steel bar images and a variety of fault shapes in the database. In some images there are regions where the imperfection do not characterize a defect, because its small depth. This situation confuses not only to the detection system, but also the human who carries out the inspection. This confusion is a recurring problem in this kind of image that uses the technique of highlighting defects by penetrating liquid. Since the imperfection that is not characterized as a defect has the same structural type of a defect which are highlighted by the fluid, such as width and proportionality between width and thickness.

The second database (DATA 2) brought better results with the proposed method. The classifiers produced less errors, however the misclassified images were with pseudo-faults named scratch. The scratches confuse the classifier because its structure is almost the same as defects like cracks and fissures. Because images are so near from the metallic surface, some scratches turn out highlighting themselves on the image. Nevertheless, the use of LBP in real-time inspection systems can be very well utilized, as shown in the method proposed in this paper.

The SVM has considerably performance, when compared to other methods of classification. Also the reduced feature vector helps improving the performance of the entire system.

VII. CONCLUSION

On inspection systems of metallic surfaces, that could be homogeneous or not, we can find many different types of problems. Where each kind of problem has its predominant characteristic. However, among these defects we found no harmful surface imperfections. The imperfections can be such as scratches or small wrinkles, which do not constitute a default, but end up being considered as one.

This study tested a texture descriptor called Local Binary Pattern or LBP to identify defects in metallic surfaces. Tests were performed on two different databases containing images of steel bars. Some of the images have defects like cracks and folds and other images were with perfect surfaces. The big difference between the databases is the use of penetrating fluid to highlight the defects. For the recognition process three different classifiers were assessed.

The results show how the use of penetrating fluid can hinder the recognition process, since 91.5% is the classification result. With the other database (DATA 2) we had a better method response, without the interference in the images was obtained 95.5% of accuracy between validation and classification.

There is an efficient representation of the defects and imperfections in homogeneous surfaces or not, when using texture features. And the results obtained in this paper help to demonstrate the benefits of using features related to texture. With a small feature vector and the $LBP_{8,2}^{5,2}$ algorithm, we reduced the processing time for feature extraction. So, it is possible to implement different tasks that assist in the inspection of these kind of surfaces. However, this method has some flaws describing regions with large amounts of components, as in the case of images with excessive penetrating fluid. Most of these interferences end up being described as holes and gaps in the image. This was evident in the view of the highlighted image classifications where there were excess of penetrating fluid.

Perhaps pre-processing the images before extract their characteristics, can reduce noise and thus keep only defective regions. Another job would be design a classifier able to inspect the defect and tell what type it is (straight, bending, cracking, cold solder, etc.). This classifier would be done with a combination of structural features and with LBP descriptors.

ACKNOWLEDGMENT

This research is supported by the National Council for Scientific and Technological Development (CNPq) grant number 306358/2008-5.

VIII. REFERENCES


