An Application of Machine Learning and Statistics to Defect Detection

R. Cucchiara[1], P. Mello[2], M. Piccardi[3], F. Riguzzi[3]

Abstract. We present an application of Machine Learning and Statistics to the problem of distinguishing between defective and non-defective industrial workpieces, where the defect takes the form of a long and thin crack on the surface of the piece. The images of the pieces are described by means of a set of visual primitives, including the Hough transform and the Correlated Hough transform. We have compared an attribute-value learner, C4.5, a backpropagation neural network, NeuralWare Predict, and the statistical techniques linear, logistic and quadratic discriminant for the classification of pieces. Moreover, two feature sets are considered, one containing only the Hough transform and the other one containing also the Correlated Hough Transform. The results of the experiments show that C4.5 performs best for both feature sets and gives an average accuracy of 93.3 % for the first dataset and 95,9 % for the second dataset.

1 INTRODUCTION

We present an application of Machine Learning and Statistics to a problem of Automated Visual Inspection (AVI) that consists of automatically inspecting the integrity of metallic industrial workpieces. The aim is to classify each piece as defective or non-defective depending on whether it contains or not surface defects, visible only under UV light. The surface defect is a crack that is visible under UV light as a bright, thin and roughly rectilinear shape.

In order to recognize cracks, a set of visual primitives has been selected for characterizing the images of pieces. In this way, each image is described by a set of numerical attributes and machine learning can be applied in order to find a classifier for new images.

In particular, we use the Hough transform (HT) that has been proposed in the literature of image analysis for detecting straight lines [1]. The HT transforms the image space into another two dimensional space (called Hough space) where each local maximum point corresponds to a straight edge in the image space. Moreover, another transformation is used, the Correlated Hough transform (CHT), that has the specific aim of detecting shapes that are bright, rectilinear and thin [2]. The CHT transforms an image from the Hough space to the Correlated Hough space where each local maximum point represents a couple of close, straight edges in the image.

In order to test the effectiveness of these different primitives on classification, we have considered two different datasets, one containing features from the Hough and the Correlated Hough space, and another one containing features from the Hough space only.

On the two datasets, we have compared an attribute-value learner, C4.5, a backpropagation neural network, NeuralWare Predict, and the statistical techniques linear, logistic and quadratic discriminant.

The paper is divided as follows: next section introduces the specific application. Section 3 discusses the adopted visual primitives. Section 4 discusses the results of experiments, providing a comparative analysis among the different algorithms. Finally, the last section provide final conclusions.

2 DEFECT DETECTION

The application goal is visual integrity inspection of metallic industrial workpieces and in particular the location of surface and subsurface defects in ferromagnetic materials.

This target can not be reached by normal, visible-light inspection but is usually accomplished by adopting a "Magnetic-Particle Inspection" technique (MPI) [3]. First, the piece is magnetized and dipped in a water suspension of fluorescent ferromagnetic particles; then, it is exposed under ultraviolet light and examined by a human inspector. When surface or subsurface defects are present, they produce a leakage field that attracts and concentrates the ferromagnetic particles. Defects can then be easily perceived by the human eye, since ultraviolet light greatly enhances fluorescence. Off-the-shelf CCD cameras and frame grabbers are used in order to acquire the images.

Examples of images with cracks are shown in figures 1, 2 and 3. Figure 1 shows a whole image, while figures 2 and 3 show two cracks in detail, more and less evident respectively.

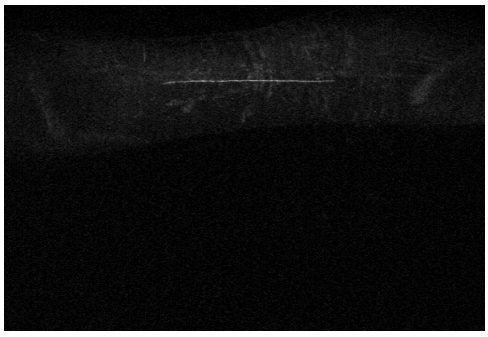

Figure 1. Image with a crack.

[1] Dipartimento di Scienze dell'Ingegneria, Università di Modena, Italy, e-mail: rita.cucchiara@unimo.it
[2] D.E.I.S., Università di Bologna, Italy, e-mail: pmello@deis.unibo.it
[3] Dipartimento di Ingegneria, Università di Ferrara, Italy e-mail: {mpiccardi, friguzzi}@ing.unife.it

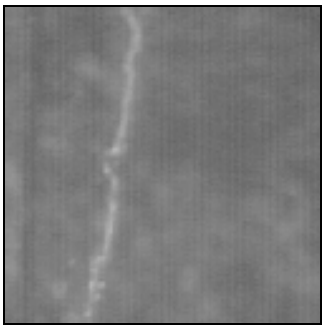

Figure 2. Detail of an evident crack.

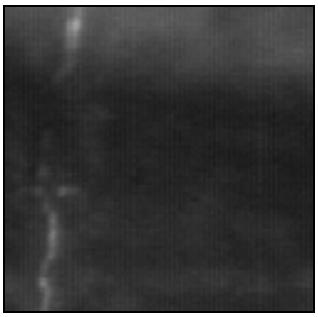

Figure 3. Detail of a less evident crack.

3 CLASSIFICATION BY VISUAL PRIMITIVES

The defect shape was a-priori known by means of a qualitative model provided by human inspectors. They defined it as a "thin, roughly rectilinear and very bright shape".

On the basis of this rather generic model, we elicited a set of measurable visual properties that can be used for describing the defects, by associating them with the aspects of the qualitative model:

- *bright shape* → high local gradient of luminosity in the proximity of its edges;
- *rectilinear* → with two main edges approximately straight;
- *thin* → with an upper-bounded distance between the two main edges.

Once elicited the visual properties, a set of quantitative image operators able to reflect them has to be defined. Usually, the approach consists of defining a rather large set of image operators, or features, each of them somehow related to one or more visual properties, which will be later used by a machine learning phase. The choice of the initial feature set is critical since the information lost at this step cannot be recovered later.

To this aim, we defined and compared two different feature sets, motivated by opposite rationale: in the first set, we included a specialized primitive called Correlated Hough transform (CHT [2]), which has been proposed for detection of objects corresponding exactly to our model; in the second set, we used only image operators of general use. The two feature sets reflect a different control of the visual aspects of the problem, the first one calling for the insight on image operators typical of a computer vision specialist, while the second one requires just the use of well-known image operators.

Both feature sets include the Hough transform (HT), which essentials is sketched hereafter. The HT has been proposed in the computer vision literature to detect straight lines [1]. It consists of a space transformation from the image space to a 2-coordinate parameter space: "collinear" points forming a straight line segment in the image space are collected into a single point of the parameter space, where the point's first co-ordinate, ϑ, is the slope of the straight line and the second co-ordinate, ρ, is its distance from the origin. Each point in the Hough space has a value which is exactly the number of collinear points in the straight line segment; thus, the longer the line segment, the higher is the point's value in the Hough space. Furthermore, in this work we adopted a refined version of the Hough transform, called gradient-weighted Hough transform (GWHT, [1]) in which each collinear point is weighted by its luminosity gradient. Therefore, peaks in the Hough space (i.e., points with high values) correspond to the existence of straight, bright lines in the image space, or we could also say that the problem of detecting lines in the image space is converted in the much easier problem of detecting peaks in the Hough space.

In the inspected images, a crack has two edges with similar gradient magnitude (with same direction but opposite orientation); since the crack is thin, the distance between the two edges is upper-bounded. Therefore, two peaks must be detected in the Hough space, with similar values and their ρ, ϑ parameters mutually constrained. In alternative to the separate detection of these two peaks, it is possible to exploit the Correlated Hough transform. The CHT performs a post-processing of the GWHT Hough space by correlating the area where the first peak is detected with the one where the second peak should be located: if it is actually present, the resulting correlation value is very high and can be easily detected. The CHT has been proven robust to non-ideality and noise, since the detection after correlation is more reliable than the detection of the two separate peaks in the Hough space. However, the CHT itself is not enough for detecting cracks when they strongly differ from their ideal aspect, and therefore we added in the feature set many other features related with the model.

The set based on the CHT (called CH dataset) contains the following features:

1. *CH (Correlated_Hough_Peak)*: this is the maximum value in the Correlated Hough space; its ρ, ϑ co-ordinates correspond to the parameters of a straight line in the image located on the crack, in case a crack is present.
2. *H1 (First_Hough_Peak)*: this is the value in the point of the Hough space with the same co-ordinates ρ, ϑ , where the first peak is formed in case a crack is present.
3. *H2 (Second_Hough_Peak)*: it is the peak in the Hough space between π and 2π at the ideal point where the second straight edge should be found.
4. *H22 (Second_Hough_Average)*: this feature is CH divided by H1; it measures how much the correlation operation increases the evidence of the crack with respect to the uncorrelated Hough space.
5. *Thickness*: the mutual distance between H1 and H2. It represents the object thickness.
6. *Number_of_Points*: the number of voting points accumulated in H1, which estimates the edge length.
7. *Average_Vote*: the average "vote" of the voting points, i.e. the average luminosity gradient of each point voting for H1 (it is computed by dividing H1 by the Number_of_Points); it

measures the average luminosity gradient along the crack profile.

8. *Average_Image_Gradient*: the average luminosity gradient of the image; it is a different property with respect to the others, since it is global, meaning that it is an overall feature of the whole image. It might be used by the classifier as a "normalization" attribute, since images with low values of the average gradient have proportionally lower CH and Hough space values.

Operationally, we acquire images with relevant views of the mechanical piece and for each image we compute the CHT. Then, we detect the CHT maximum (the CH feature) and record a tuple with CH and the other associated feature values. We then detect all the points of the correlated Hough space whose value is greater or equal an assigned percentage of the maximum (75% was used in the experiments), and record a tuple for each of them; this is done in order to catch multiple cracks that can be present in a single image. After acquiring the tuples, we pre-classify each of them into the two categories of *Defect* or *NoDefect* by checking manually if the straight line segment corresponding with the tuple was located on a real crack in the image or not.

In the approach followed, the CHT plays a major role, since the CH maximum is the feature that determines the position where the crack may be located. However, the CHT is a highly specialized operator, and it is interesting to approach the problem with a feature set with more standard features, and comparing the performance of the resulting classifiers.

Therefore, in the second dataset set (called H1 H2 dataset) we excluded the CH value and included the following features:

1. *H1*: the value of the Hough maximum in the range $\vartheta \in [0,\pi]$, where the first peak is formed in case a crack is present; its ρ, ϑ co-ordinates correspond to the parameters of a straight line located on one edge of the crack.
2. *H2*: the value of the Hough maximum in the range $\vartheta \in [\pi, 2\pi]$, where the second peak is formed in case a crack is present; its ρ', ϑ' co-ordinates correspond to the parameters of a straight line located on the other edge of the crack. However, if multiple cracks are present, H1 and H2 may not be associated with the same crack.
3. *Number_of_Votes*: the sum of the number of image points that were transformed into H1 and H2.
4. *Distance*: the mutual distance between H1 and H2 in the Hough space. It represents the object thickness if H1 and H2 correspond to the same crack.
5. *Delta_rho*: the $|\rho' - \rho|$ value, and
6. *Delta_theta*: the $|\vartheta' - \vartheta - \pi|$ value. *Delta_rho* and *Delta_theta* express the distance between the two peaks along the ρ and ϑ directions, respectively. In case of a same real crack, *Delta_theta* should be close to 0 and *Delta_rho* upper bounded. *Delta_rho* and *Delta_theta* are related to *Distance* by the following formula :

$$Distance = \sqrt{Delta\_rho^2 + Delta\_theta^2} \ .$$

7. *Delta_product*: the product delta_rho * delta_theta. It correlates the Delta_rho and Delta_theta values, expecting small values for the product in case of a same real crack.
8. *Average_Image_Gradient*: The average luminosity gradient of the image.

Since there is not an explicit correlation operation between H1 and H2, we also added some basic arithmetic functions of the H1 and H2 values:

9. *Product*: the product H1 * H2: should be high in case of a real crack (about the square of each of the two values).
10. *Ratio*: the ratio H1 / H2: should be close to 1 in case of a real crack.
11. *Sum*: the sum H1 + H2: should be high in case of a real crack (about double each of the two values).
12. *Difference*: the difference H1 - H2: should be close to 0 in case of a real crack.

These arithmetic functions are just combinations of other features and thus may be considered redundant, but they have been explicitely included in the feature set since they are related with the model and may improve the classifiers' performance in case the classifier does not explore linear or quadratic combinations or ratios of the feature values.

Operationally, we acquire images with relevant views of the mechanical piece and for each image we compute the Hough space with the GWHT. Then, we detect the H1 and H2 maxima and record them in a tuple with the other associated feature values. We then repeat the process for all the points of the Hough space in the range $[0, \pi]$ and $[\pi, 2\pi]$ whose value is greater or equal an assigned percentage of H1 and H2, respectively, and record a tuple for each couple; this is done in order to catch multiple cracks that can be present in a single image. After acquiring the tuples, we pre-classify each of them into the two categories of *Defect* or *NoDefect* by checking manually if the straight line segments corresponding with H1 and H2 were located on a same real crack.

4 EXPERIMENTS

We have experimented and compared two different machine learning techniques: attribute-value learning and backpropagation neural networks. Moreover, due to the numeric nature of all the attributes, we have used statistical techniques as well in order to compare their performance with that of machine learning tools.

For attribute-value learning we have used C4.5 [4] that is able to learn both decision trees and rules. For backpropagtion neural networks, we have used a commercial system, Predict by NeuralWare[1]. As regards statistical techniques, we have used the algorithms Discrim, Logdisc and Quadisc, developed under the Statlog project [5], that implement respectively linear discriminant, logistic discriminant and quadratic discriminant.

In the following, we first give a brief description of each algorithm and then we present the results of experiments.

4.1.1 Discrim

Discrim finds a linear discriminant, i.e., a hyperplane in the p-dimensional space of the attributes. Given the values of the attributes of a new pattern, its class is found by looking at the position of the corresponding point with respect to the hyperplane.

The hyperplane equation is found on the assumption of normal probability distribution: the attribute vectors for the examples of class A_i are independent and follow a certain probability

[1] More information about Predict can be found at http://www.neuralware.com/ .

distribution with probability density function (pdf) f_i. A new point with attribute vector x is then assigned to that class for which the probability density function $f_i(\mathbf{x})$ is greatest. This is a maximum likelihood method. The distributions are assumed normal (or Gaussian) with different means but the same covariance matrix. The probability density function of the normal distribution is

$$f_i(\mathbf{x}) = \frac{1}{\sqrt{|2\pi\Sigma|}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu_i)' \Sigma^{-1}(\mathbf{x}-\mu_i)\right) \quad (1)$$

where μ is a p-dimensional vector denoting the theoretical mean for class i and Σ, the theoretical covariance matrix, is a $p \times p$ matrix that is necessarily positive definite. In this case the boundary separating the two classes, defined by the equality of the pdfs, can be shown to be a hyperplane that passes through the mid-point of the two means. Its equation is

$$\mathbf{x}'\Sigma^{-1}(\mu_1-\mu_2) - \frac{1}{2}(\mu_1+\mu_2)'\Sigma^{-1}(\mu_1-\mu_2) = 0 \quad (2)$$

where μ_i is the population mean for class A_i. When using this formula for classification the exact distribution is usually not known and the parameters must be estimated from the available sample. With two classes, if the sample means are substituted for μ_i and the pooled sample covariance matrix for Σ, then Fisher's linear discriminant [6] is obtained. The covariance matrix for a dataset with n_i examples from class A_i is

$$S_i = \frac{1}{n_i - 1} X^T X - \bar{\mathbf{x}}^T \bar{\mathbf{x}} \quad (3)$$

Where X is the $n_i \times p$ matrix of attribute values and $\bar{x}$ is the p-dimensional row vector of attribute means. The *pooled covariance matrix S* is

$$S = \frac{\sum (n_i - 1) S_i}{n - q} \quad (4)$$

where the summation is over all the classes and *(n-q)* is chosen to make the pooled covariance matrix unbiased.

4.1.2 Quadisc

Quadisc performs a quadratic discrimination. Quadratic discrimination is similar to linear discrimination with the difference that the surface separating the two regions is quadratic. This means that the discriminating function will contain not only the attributes but also their squares and the products of two attributes. With respect to the case of probability maximization seen in the previous case, if we remove the assumption that the normal distributions have the same covariance matrix S, we obtain a quadratic surface, for example an ellipsoid or a hyperboloid.

The simplest quadratic discrimination function for a class is defined as the logarithm of the corresponding probability density function and is given by equation 5 in the case of differing prior probabilities. The suffix i is used to indicate class A_i.

$$\log \pi_i f_i(\mathbf{x}) = \log \pi_i - \frac{1}{2}\log(\det(\Sigma_i)) - \frac{1}{2}(\mathbf{x}-\mu_i)'\Sigma_i^{-1}(\mathbf{x}-\mu_i) \quad (5)$$

In this equation π_i stands for the prior probability of class A_i. As before, the means and covariance matrix are substituted by their sample counterparts obtained from the training set. In the same way, π_i is substituted by the sample proportion of class A_i examples. For classification, the discriminant is calculated for each class and the one giving the highest value is chosen.

The most frequent problem with quadratic discriminants is caused when some attribute has zero variance in one class, for then the covariance matrix can not be inverted. One way of avoiding this problem is to add a small positive constant term to the diagonal terms in the covariance matrix (this corresponds to adding random noise to the attributes).

4.1.3 Logdisc

Logdisc performs a logistic discrimination. As linear discriminants, a logistic discriminant consists of a hyperplane separating the classes in the best possible way, but the criterion used to find the hyperplane is different. The method adopted in this procedure is to maximize a conditional probability. In theory, when the attributes have a normal distribution with equal covariances and are independent from each other, linear and logistic discriminants are equivalent. Different result are obtained when these hypotheses are not satisfied.

The method here described is partially parametric, as the actual pdfs for the classes are not modeled, but rather the ratio between them. In particular the logarithms of the ratios of the probability density functions for the classes are modelled as linear functions of the attributes. Thus, for two classes

$$\log \frac{f_1(\mathbf{x})}{f_2(\mathbf{x})} = \alpha + \beta'\mathbf{x} \quad (6)$$

where α and the p-dimensional vector β are the parameters of the adopted model and must be estimated. The case of normal distribution is a special case in which these parameters are functions of the prior probabilities, of the class means and of the common covariance matrix.

The parameters are estimated by maximum conditional likelihood. The model implies that, given attribute values **x**, the conditional class probabilities for classes A_1 and A_2 take the forms:

$$P(A_1 \mid \mathbf{x}) = \frac{\exp(\alpha + \beta'\mathbf{x})}{1 + \exp(\alpha + \beta'\mathbf{x})} \quad (7)$$

$$P(A_2 \mid \mathbf{x}) = \frac{1}{1 + \exp(\alpha + \beta'\mathbf{x})} \quad (8)$$

Given independent samples for the two classes, the parameters are estimated by maximizing the probability:

$$L(\alpha, \beta) = \prod_{\{A_1, sample\}} P(A_1 \mid \mathbf{x}) \prod_{\{A_2, sample\}} P(A_2 \mid \mathbf{x}) \quad (9)$$

Iterative methods have been proposed in order to estimate the parameters for example like in [7] and [8]. Since in practice there is often little difference between logistic and linear discriminant, the latter are taken as a starting point for the former.

Table 1. Average accuracies

	Discrim	Logdisc	Quadisc	Predict	C4.5 tree	C4.5 rules
CH	0,853	0,857	0,853	0,873	0,959	0,959
H1 H2	0,855	0,928	0,316	0,864	0,933	0,933

Table 2. Average false negative (FN) and false positive (FP) errors

	Discrim		Logdisc		Quadisc		Predict		C4.5 tree		C4.5 rules	
	FN	FP	FN	FP	FN	FP	FN	FP	FN	FP	FN	FP
CH	37	9	36	9	31	15	15	25	6	7	6	7
H1 H2	26	19	14	9	40	177	3	40	13	8	12	9

Table 3. Values for the *t* statistics for the CH dataset

$\|t\|$	Discrim	Logdisc	Quadisc	Predict	C4.5 tree	C4.5 rules
Discrim		1,000	0,000	0,452	**1,947**	**1,947**
Logdisc			0,166	0,376	**1,959**	**1,959**
Quadratic				0,615	**2,352**	**2,352**
Predict					**2,031**	**2,031**
C4.5 tree						0,000

Table 4. Values for the *t* statistics for the H1 H2 dataset

$\|t\|$	Discrim	Logdisc	Quadisc	Predict	C4.5 tree	C4.5 rules
Discrim		**1,753**	**2,114**	0,118	**1,586**	**1,689**
Logdisc			**2,411**	0,867	0,127	0,135
Quadisc				**2,509**	**3,068**	**3,050**
Predict					0,843	0,858
C4.5 tree						0,000

4.1.4 NeuralWorks Predict

Predict by Neural Works is a system for training multi-layer neural nets. Predict uses an adaptive gradient learning rule which is a form of back-propagation. Predict does not start from a fixed network architecture but uses a constructive method for determining a suitable number of hidden nodes. This constructive method is referred to as "Cascade Learning" [10] and is loosely characterised by the fact that hidden nodes are added one or a few at a time. New hidden nodes have connections from both the input buffer and the previously established hidden nodes. Construction is stopped when performance on an independent test set shows no further improvement.

4.1.5 C4.5

C4.5 [4] is a system for learning rules and decision trees. Its peculiarity is the heuristics it adopts in order to select the test to perform at each steps. These heuristics are based on the notion of entropy from information theory that represents the amount of “dis-uniformity” of examples in the training set with respect to the class attributes: at each step a test is selected that makes the resulting subsets as uniform as possible with respect to the class attribute, i.e., subsets containing examples from only one class or from a small number of classes.

4.1.6 Results

All systems have been tested on the CH and H1 H2 datasets employing 10-fold cross validation. Both datasets contain 317 tuples of which 67 belong to the Defect class and 250 to the NonDefect class. The spread of attribute values is larger for the Defect class.

Table 1 shows the average accuracies of the classification algorithms for both datasets, while table 2 shows the total number of false negative and false positive errors summed over the ten folds. False negatives are defective pieces that are classified as non defective and false positive are non defective pieces that are classified as defective. It is important to distinguish between these two types of errors because the damage that derives from a false negative is much higher than the one deriving from a false positive. Therefore, we should prefer an algorithm that minimizes the number of false negatives.

In order to evaluate if the accuracy differences between algorithms are significant, we have computed a 10-fold cross-validated paired *t* test for every pair of algorithms (see [11] for an overview of statistical tests for the comparison of machine learning algorithms).

This test is computed as follows. Given two algorithms A and B, let $p_A^{(i)}$ (respectively $p_B^{(i)}$) be the observed proportion of test examples misclassified by algorrithm A (respectively B) in trial i. If we assume that the 10 differences $p^{(i)}=p_A^{(i)}-p_B^{(i)}$ are drawn independently from a normal distribution, then we can apply Student *t* test by computing the statistic

$$t = \frac{\bar{p} \cdot \sqrt{n}}{\sqrt{\frac{\sum_{i+1}^{n} (p^{(i)} - \bar{p})^2}{n-1}}} \quad (10)$$

where n is the number of folds (10) and $\bar{p}$ is

$$\bar{p} = \frac{1}{n} \sum_{i=1}^{n} p^{(i)} \quad (11)$$

In the null hypothesis, i.e. that A and B have the same accuracy, this statistic has a t distribution with n-1 (9) degrees of freedom. If we consider a probability of 90%, then the null hypothesis can be rejected if

$$|t| > t_{9,0.90} = 1.383 \quad (12)$$

Table 3 shows the values of the statistic for the CH dataset, while table 4 shows the values of the statistic for the H1 H2 dataset. The value of the statistic for algorithms A and B can be found at the crossing of line A with column B. The numbers in bold are those that provide a probability of 90% or more of rejecting the null hypothesis.

From these tables can be seen that, for the CH dataset, the accuracy difference is statistically significant only between C4.5 algorithms and the other ones, while it is not statistically significant among the statistical and neural algorithms. Therefore, for the CH dataset, we can state that the best performance has been obtained by C4.5, both for the case of trees and rules.

On the H1 H2 dataset there is a significant difference between the best performing algorithms, C4.5 and Logdisc, and Discrim and Quadisc. The difference among the best performing algorithms and Predict is instead a little less certain, having 80% probability.

In conclusion, for both datasets, the best overall accuracy has been obtained by C4.5 both for the case of trees and rules. The comparison of machine learning and statistical techniques shows that C4.5 performs better than statistical techniques for the CH dataset, while on H1 H2 dataset Logdisc is equivalent to C4.5. Instead, for Predict, the differences with statistical techniques are less significant.

As can be seen, the CH feature is very important because it leads to more accurate classifiers for all systems apart from Logdisc and Discrim.

As regards the number of false negatives, C4.5 yields the lowest number of them for the CH dataset, while for the H1 H2 dataset the lowest number is given by Predict.

These results show that machine learning tools can outperform statistical classifiers on the domain examined.

1 RELATED WORKS

Machine learning has been widely exploited for object classification in computer vision. Learning is often essential for defining an effective classifier in the case of unstructured objects or shapes, which are difficult to model in terms of geometric, topologic or other metric features. Examples of the use of learning in computer vision are for instance recognition of hand gestures, landscape inspection, medical images analysis, and appearance-based recognition [11,12,13,14]. However, the most comprehensive work concerning the use of learning for classification is the StatLog project [5]. StatLog includes several classification algorithms, covering machine learning, neural and statistical techniques. The algorithms are compared against several different classification tasks, nine of which consists of classifying images (Dig44, KL, Vehicle, Letter, Chrom, SatIm, Segm, Cut20, Cut50). As reported in the StatLog results in [5], the ranking of classifiers in terms of error rates varies with the image classification task. As stated in the analysis of results still in [5], some of these tasks address mostly classification of pixel areas, while others address classification of derived features computed from the pixel values. These tasks are very different in nature, and this may be a major reason for the different ranking of classifiers' error rates.

The k-NN classifier achieves generally the best error rate. However, one pitfall of the k-NN method is the fact that it typically treats variables with equal weight, and this may be the reason for the few exceptions (Vehicle and Segm); in these cases, k-NN is outperformed by many other algorithms, including C4.5.

Quadisc achieves the best error rates only for those image datasets considered as object recognition (Dig44, KL, Vehicle, Letter, Chrom), while performing badly on average on the image datasets considered as segmentation (SatIm, Segm, Cut20, Cut50).

The machine learning algorithm C4.5 tends to assess good performance for the segmentation tasks (Segm, Cut20, Cut50) and in particular, it largely outperforms Quadisc on the Cut dataset. On the other datasets, C4.5 ranks on average positions.

In the application we considered C4.5 achieves the best error rates for both feature sets, while k-NN ranks in the second position. Quadisc assesses significantly worse performance, similar to that of the other classifiers for the CH feature set and drastically lower for the H1 H2 one. The resu

5 CONCLUSION

We have presented an application of machine learning and statistics to the problem of recognizing surface cracks on metallic pieces. In order to learn from the images of the pieces, we have identified a set of visual features for characterizing each image. One of these features, the average gradient of luminosity, is computed on the image itself, while the others are computed on transformed versions of the image obtained with the Hough transform (HT) and the Correlated Hough transform (CHT). We use these features because they have been expressively designed for the recognition of straight lines and rectilinear shapes.

In order to test the effectiveness of these various features on classification, we have considered two different sets, one containing features from the Hough and the Correlated Hough space, and another one containing features from the Hough space only.

Various machine learning and statistical techniques have been applied to the problem. As regards machine learning, we have employed an attribute value learner, C4.5, and a neural network trainer, NeuralWare Predict. As regards statistical techniques, we have employed linear, logistic and quadratic discriminants.

The results of the experiments show that, of the two feature sets, the one containing the CHT leads to more accurate classifiers for all learning methods apart from Logdisc and Quadisc, thus confirming the usefulness of highly specialized operators for Computer Vision.

Among all systems, C4.5 had a performance significantly higher than the other systems on the CH dataset, while on the H1 H2 dataset it was significantly higher than Discrim and Quadisc.

These results are not easy to explain It is not easy to explain this behaviour C4.5 provided a very good performance. This is probably due to spread in the attribute values, especially for the Defect class, that requires the adaptiveness of machine learning tools.

6 REFERENCES

[1] J. Illingworth, J. Kittler, “A survey of the Hough transform”, Comp. Vision Graphics, Image Process. (43): 221-238.

[2] R. Cucchiara, M. Piccardi, “Eliciting visual primitives for detection of elongated shapes”, Image and Vision Computing, v. 17, n.5-6, pp. 347-355, Elsevier, 1999.

[3] R. Mason, editor, Magnetic Particle Inspection. Nondestructive Testing (33), pp. 6-12.

[4] J. R. Quinlan, C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, San Mateo, California, 1993.

[5] D. Michie, D.J.Spiegelhalter and C.C.Taylor (eds.), “Machine Learning, Neural and Statistical Classification”, Ellis Horwood, 1994.

[6] R. Fisher, “The use of multiple measurements in taxonomic problems”, Annals of Eugenics, 7, pp. 179-188, 1936.

[7] D. R. Cox, “Some procedures associated with the logistic qualitative response curve”, in Research Papers on Statistics: Festschrift for J. Neyman, Wiley, pp. 55-77, 1966.

[8] N. Day, D. Kerridge, “A general maximum likelyhood discriminant”, Biometrics, 23, pp. 313-324, 1967.

[9] S. E. Fahlmann, C. Lebiere, "The Cascade-Correlation Learning Architecture", Advances in Neural Information Processing Systems 2, Morgan Kaufmann, 1988.

[10] T. Dietterich, “Approximate satistical tests for comparing supervised classification learning algorithms”, Neural Computation, in press (draft version available at http://www.cs.orst.edu/~tgd/projects /supervised.html).

[11] K. Cho, S. M. Dunn, “Learning shape classes”, IEEE Trans. on PAMI 16 (1994) n. 9, pp. 882-887.

[12] B. A. Drapter, C. E. Brodley, P. E. Utgoff, “Goal directed classification using Linear Machine Decision tree”, IEEE Trans. on PAMI 16 (1994) n. 9, pp. 888-893.

[13] P. Pellegretti, F. Roli, S. Serpico, G. Vernazza, “Supervised learning of descriptions for image recognition purposes”, IEEE Trans. on PAMI 16 (1994) n. 1, pp. 92-98.

[14] H. Murase, S. K. Nayar, “Learning by a generation approach to appearance based object recognition”, Proc. of 13th ICPR, Vienna 1 (1996) , pp. 24-30.