Intell Ind Syst (2015) 1:373–393
DOI 10.1007/s40903-015-0018-5

ORIGINAL PAPER

Automatic Defect Classification on a Production Line

N. Alberto Borghese[1] · Manuele Fomasi[1]

Received: 16 March 2015 / Revised: 4 May 2015 / Accepted: 12 June 2015 / Published online: 14 July 2015

Abstract We describe here a novel defect classification system that works in real-time on the images of material running on the production line, provided by a video-inspection module. The classifier is constituted of a two-levels hierarchical architecture: a set of adequate features are extracted from the image first; the defect type is then identified from them. Statistical analysis allows greatly reducing the number of features, leaving only the most significant ones. A particular version of multi-class boosting has been developed for labeling: differently from its original version, it accepts only one label for each image, the true one, and does not require the rank for the other classes. Nevertheless, the classifier is able to produce a valid rank of the defect with respect to all the classes, information that can be used to achieve an identification rate of the dangerous defects very close to 100 % on a real data set.

Keywords Video-inspection systems · Adaptive defect classification · Boosting · Feature statistical evaluation · Real-time image processing

1 Introduction

Quality control of the production is a key element for the success on the market in very different fields [1–6] and different technologies have been adopted according to the peculiarities of the material and of the defects, ranging from X-rays [7], to laser [8–10], microscopy [11], ultrasound, [12], thermal [13] or electromagnetic field [13]. In the field of textile industry, defect identification is carried out mainly by video-inspection systems that analyze in real-time the material passing at high speed on a production line [14–18] (Fig. 1a). When a pattern different from the clean tissue is detected, its image is displayed to the operator to make the proper choice (Fig. 3). Some defects are not dangerous (e.g. small woollen threads, spots), while few of them must be eliminated: a mother who finds an insect inside the plastic cover of a pamper of her baby or a surgeon who finds a metallic staple inside the cellophane cover of a surgical instrument might spur a multi-million dollars suit. A conservative approach is often chosen in which, most defects are reanalyzed in the packaging room with a huge loss of time. Automatic real-time identification of dangerous defects is therefore desirable to reduce time loss and increase reliability as defects identification, in the current model, depends critically on the skills and status of the human operator.

As shown here, machine learning allows going one step further, recognizing automatically the type of the defect. This is required for several reasons: for instance, identification of environment contaminants, scheduling of cleaning sessions of the machines, assessment of the quality of raw material, identification of weaknesses in the production line. For this reason, systems for automatic defect classification (ADC) have been recently introduced. These are usually based on a hierarchical two-level structure [19,20], in which the first level extracts a set of features from the image of the potential defect and the second level classifies the image and, in case of dangerous defect, triggers a procedure for defect elimination.

Features definition is the stage most dependent on the application and it capitalizes often on the experience of a domain expert. In [21] a hybrid approach is pursued where features are extracted in both the spatial and the

✉ N. Alberto Borghese
alberto.borghese@unimi.it
http://ais-lab.dsi.unimi.it

[1] Applied Intelligent Systems Laboratory, AIS-Lab, Department of Computer Science, Università degli Studi di Milano, Via Comelico, 39, 20135 Milano, Italy

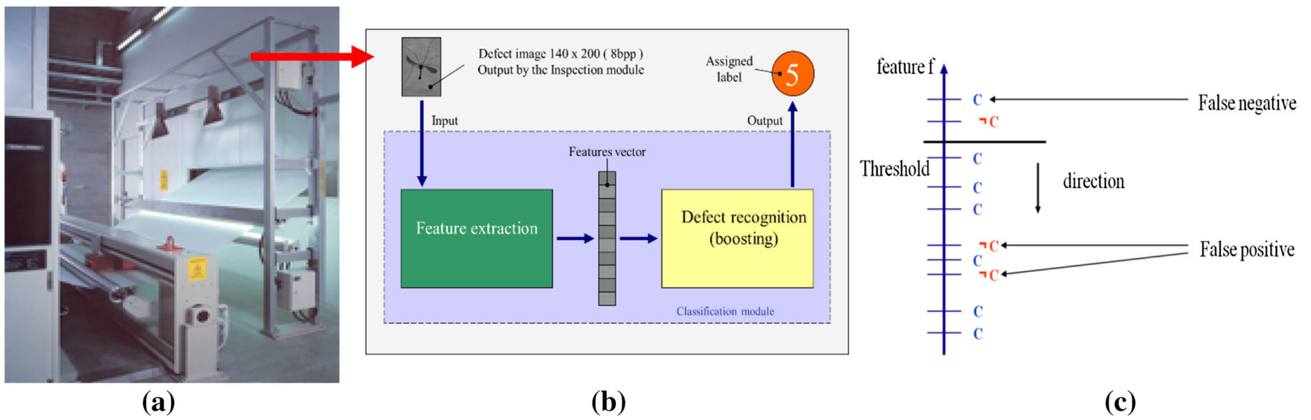


Fig. 1 A typical video-inspection system from [1] is shown in (**a**). When an image contains a defect e sent also to a classification stage (**b**) and/or displayed to an operator. The classification stage is constituted of two hierarchical modules: features extraction and defect classification. Classification is carried out by a linear combination of weak classifiers. Each of these is defined by a threshold and a direction (**c**) and it analyzes a single feature

frequency domain. Several approaches based on localized frequency analysis have been proposed. For instance, in [22,23] wavelets are used to identify the "signature" of the features in the different sub-bands. These approaches can be useful when the defect image is sufficiently large and the image exhibits some invariance properties. This is often not the case as the available images are usually small and defects appear with a different orientation and position, come with different shapes and change their appearance depending on how the defect is generated. For these reasons, only spatial features are considered here.

Extracted features are input to the defect classification module (Fig. 1b). Machine learning techniques have been recently introduced to achieve a robust classification as they allow "learning" the correct classification from a set of example images, collected on the field, called training set. Moreover, some of these techniques can learn incrementally and adapt to the current defect population. These techniques can be subdivided into three broad categories: soft-computing, [24–27], support vector machines [28,29] and boosting [30–32].

The methods belonging to the first class are based mainly on soft-clustering possibly combined with Gaussian mixture models and they are aimed at finding sub-regions in the features space that characterize each class. This approach can hardly be applied here as the number of features can be very large and they present large variability; therefore different approaches have been pursued. Support vector machines (SVM) identify non-linear boundaries between classes, by projecting the image features into a high dimensional space where the different classes can be separated through hyperplanes. This mapping is obtained through non-linear parametric functions, named kernels. However, the performance of the classifier depends critically on the parameters involved; these are determined through non-linear optimization that turns out time consuming and not always converges to the global minumum [29,33]. Moreover, SVMs do not have the flexibility to add new classes / images as they need to be retrained from scratch in this case.

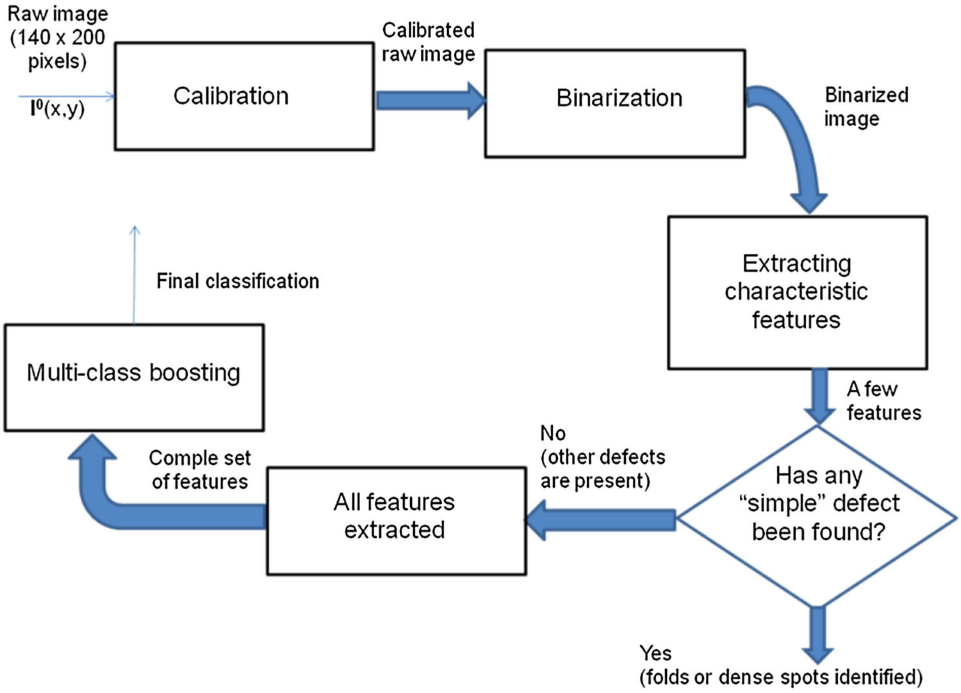
Boosting appears a more natural way to approach our problem. It is based on combining a set of weak classifiers, each working on a single feature, to obtain a robust classifier (cf. "Regular defects" section). Moreover, in its multi-label version, boosting, not only classifies a datum to a class, but provides also a rank of the datum with respect to the other classes [34–37]. However, we do not have such rank information as the domain expert provides only the label of one class: that to which the defect belongs.

We show here experimentally that, although this ranking order is not provided to the classifier, the system is indeed able to learn by itself a ranking order that reflects the similarity of the defect with the different classes. We take advantage of this to propose to the human operator, more than one class whenever the classifier identifies that the score of two classes is close to each other. We show with examples, that this is the case when defects have doubtful classification also for human experts. This classifier is complemented with a careful feature design based on a two-stage procedure: in the first stage a set of standard features is defined with the help of the domain expert. These features are then evaluated and pruned. Afterwards specific and innovative features have been realized to improve classification. Results on a reasonably large data set show the validity of the methodology and the ability to reach the detection of all the critical defects on the material.

2 Overview of the System

Defect images are sent by the inspection module to the classification stage shown in Fig. 2. This processes the image

Fig. 2 The processing flow-chart inside the classification stage. In brackets the issues considered in the actual example described here


Raw image (140 x 200 pixels)
I0(x,y)
Calibration
Calibrated raw image
Binarization
Binarized image
Extracting characteristic features
A few features
Has any "simple" defect been found?
Yes (folds or dense spots identified)
No (other defects are present)
All features extracted
Comple set of features
Multi-class boosting
Final classification

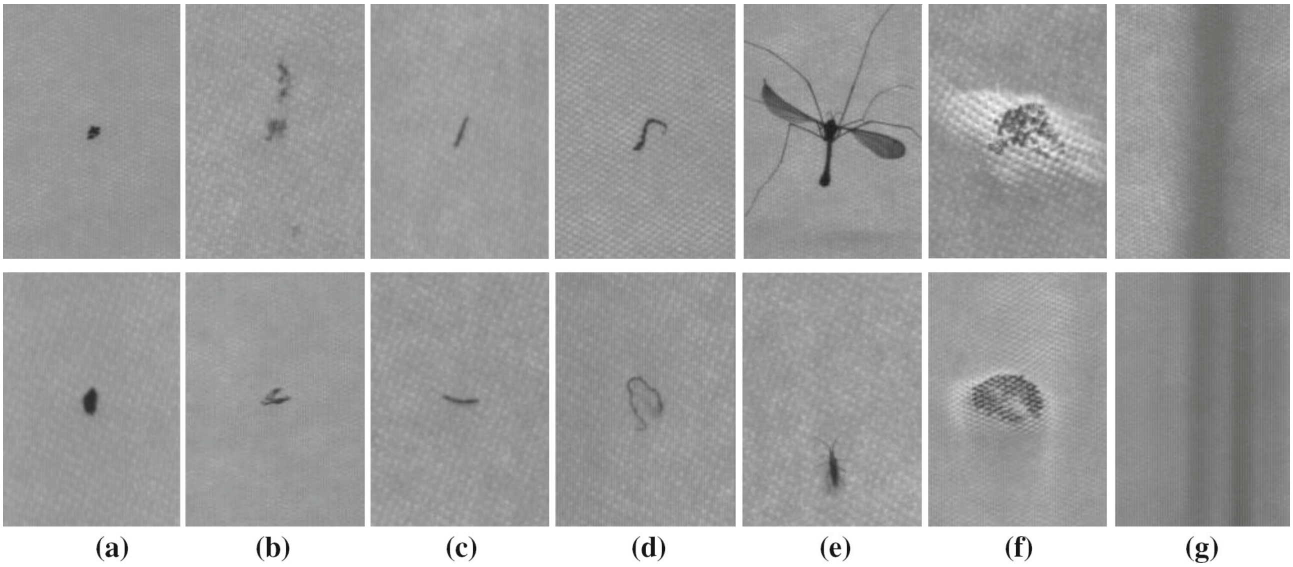


Fig. 3 Examples of defect images identified by the domain expert: regular contaminations (**a**), irregular contaminations (**b**), elongated defects (**c**), threads (metallic staples or cotton threads **d**) insects (**e**), dense spots (**f**), folds (**g**). As it can be seen, defects can change largely their appearance, position and orientation

to extract a set of features and, from them, it provides the classification of the image and its rank with respect to the other classes. The "simple" defects characterized by a single feature or a simple combination of features, identified through statistics ("Features Evaluation" section), are detected first while all the other defects are detected through boosting.

For the particular system described in the Results section, a total of 295 real images of the defects were made available. The domain expert identified seven different defect classes: regular contaminations, irregular contaminations, elongated defects, threads (metallic staple or cotton thread), insects, dense spots and folds (Fig. 3). For each image, correct classification was provided.

3 Pre-processing

Before extracting the features, image pre-processing is required to eliminate most intra image luminance variability. This step is composed of gray level calibration and binarization (Fig. 2).

3.1 Gray Level Calibration

Let us call $\mathbf{I^0} = \mathbf{I^0}(x, y)$ the image received from the inspection module, having dimension N x M and K bits per pixel. Often this is produced by two different line-scan cameras and a luminance difference between adjacent lines can be observed as it is evident in Fig. 4a. This difference is systematic and it is due to a different gain of the two sensors. It must be eliminated for making processing more robust. We adopt here a procedure for real-time gain equalization. The mean gray level value, $\bar{I}$, is computed separately for all the odd and even lines: $\bar{I}(\mathbf{I^0_{even}})$ and $\bar{I}(\mathbf{I^0_{odd}})$ and the gray level of the pixels in the odd lines is multiplied by the ratio: $\frac{\bar{I}(\mathbf{I^0_{even}})}{\bar{I}(\mathbf{I^0_{odd}})}$ (gain equalization):

$$\mathbf{I^0_{odd}} = \mathbf{I^0_{odd}} * \Delta \text{ with } \Delta = \frac{\bar{I}\left(\mathbf{I^0_{even}}\right)}{\bar{I}\left(\mathbf{I^0_{odd}}\right)} \tag{1}$$

The second step is the normalization obtained linearly stretching the image such as to cover the entire range of the gray levels.

3.2 Background Subtraction (Binarization)

We now extract the defect from the image. Although the material surface may have a weak texture, algorithms for texture analysis and identification [38–41], besides their computational cost, cannot be applied here as texture is often distorted at the boundaries of the defect. Moreover, some texture may easily be confounded with the defect itself (cf. Fig. 3). For this reason, a binarization approach based on global statistics bas been adopted here: the mean and standard deviation of the background gray levels, μ_B^o and σ_B^o, is computed on images without defects and the binarization threshold, T^B, is first set to:

$$T^B = \mu_B^o + k * \sigma_B^o \tag{2}$$

with k is set to the very large value $k = 5$, that guarantees that all the background pixels are eliminated. T^B is then refined according to Otsu clustering [42] decreasing k until the variance between the pixels of the defect and of the background is maximized. $\mathbf{I^0}(x, y)$ is then binarized through T^B to obtain the binarized image, $\mathbf{I^B}(x, y)$ (cf. Fig. 5b):

$$\begin{aligned} I^B(x, y) = 1 \quad &\mathit{iff} \quad I^0(x, y) \geq T^B \\ I^B(x, y) = 0 \quad &\mathit{iff} \quad I^0(x, y) < T^B \end{aligned} \tag{3}$$

that contains the pixels of the defect. This procedure has the advantage of simplicity and it allows tuning binarization to different lighting conditions produced by a change in material thickness, flash units and so forth. It should be remarked that binarization may not produce a perfect separation between the gray levels of the defect and those of the background when the two ranges of gray levels partially overlap. This is critical especially for insects (Figs. 3e, 13) and for dense spots (Figs. 5, 10).

Isolated background pixels inside the defect body may hamper feature extraction; for this reason, a second binarized image is obtained filling the holes in $\mathbf{I^B}$ through standard Region Growing [43]. This second image is referred to as *enhanced binarized image*, $\mathbf{I^E}$ (Fig. 5c).

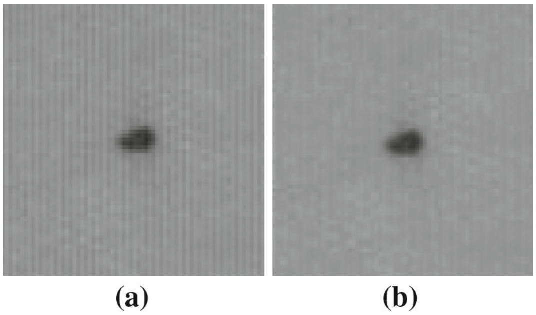


Fig. 4 A raw image is displayed on the *left*. The same image after calibration is shown on the *right*: the *vertical* stripes have been eliminated

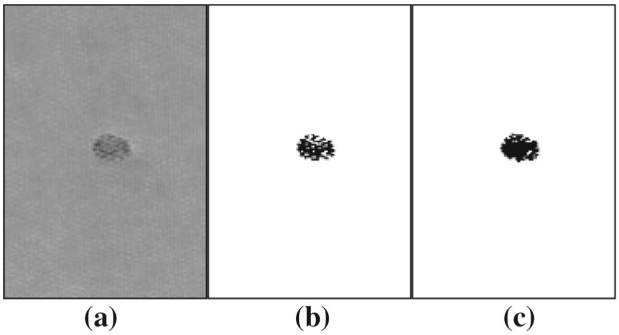


Fig. 5 Calibrated image of a dense spot, I^0(x,y) (**a**). The same image after thresholding, I^B(x,y) (**b**), and after region growing, I^E(x,y) (**c**)

3.3 Orientation Invariance

Defects can show up in any orientation and this has to be factored out to simplify their classification. To this aim we apply principal component analysis [43] to determine the elongation main direction of the defect. This is computed through the Singular Value Decomposition (SVD) of the data dispersion matrix, $\mathbf{D^B}$:

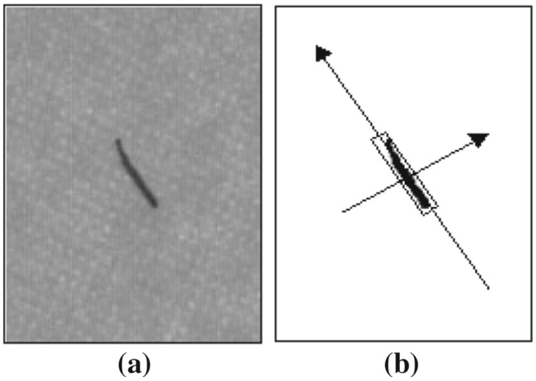


Fig. 6 A typical image of an elongated defect is shown in panel (**a**). The binarized image, with superimposed the principal directions, is shown in panel (**b**). The tight (oriented) rectangular bounding box is also reported

$$\mathrm{SVD}(\mathbf{D^B}) = \mathbf{U\,W\,V} \tag{4}$$

with:

$$\mathbf{D^B} = \begin{bmatrix} \sum_k (x_k - \bar{x})^2 & \sum_k (x_k - \bar{x})(y_k - \bar{y}) \\ \sum_k (x_k - \bar{x})(y_k - \bar{y}) & \sum_k (y - \bar{y})^2 \end{bmatrix} \tag{5}$$

where $\bar{\mathbf{p}}(\bar{x}, \bar{y})$ is the average position of the defect pixels (overthreshold) $\mathbf{I^B}$. $\mathbf{U}$ and $\mathbf{V}$ are 2 × 2 orthonormal matrixes, and $\mathbf{W}$ is a diagonal matrix whose elements contain the sum of the squared distance of the defect points from the two principal axes. Applying the matrix $\mathbf{V^T}$, $\mathbf{I^B}$ is rotated so that the defect major principal axis becomes vertical, making the image largely insensitive to defect orientation (cf. Fig. 6). The rotated binary defect image will be referred to as $\mathbf{I^{RB}}$.

4 Feature Extraction

At start, we have identified a basic set of 159 features following the experience of a domain expert. Although they allow identifying most of the defects, the classifier performed poorly on threads and insects. For this reason, ad-hoc features for these classes have been developed and features previously defined have been refined as reported in “Enhanced Features” section. A total of 221 features were designed. These have been evaluated a-posteriori, through the statistical framework described in “Features Evaluation” section, to assess their discriminative power, their redundancy and efficiency. Through this analysis, the features were reduced to a total of 71, on which the final classifier operates. In the following the main basic features are described for each class. We explicitly remark that features were not normalized as the elementary classifiers adopted here work on single features. Similar results were obtained for normalized features.

4.1 Regular Defects

These are characterized by a uniform area and by an approximately circular shape. To assess the degree of uniformity, the image inside B_B is clustered into two or three disjointed regions ([44], cf. Fig. 7), from which the following features are extracted. The *number of blobs*, where a blob is a group of 8-connected pixels, is typically two: background and defect when two clusters are considered or a few for three clusters. The *mean gray level of each cluster* is usually lower than in the other classes, as these defects are usually darker. The *variance of the gray levels within each cluster* gives a direct evaluation of the uniformity of each cluster. Moreover, in three-clusters clustering, the l^2 norm of the *difference between the mean gray level of the two darkest clusters* is usually smaller than in other classes as these two regions are associated respectively to the inner defect area and to its border respectively.

Another feature, characteristic of this class, is the *histogram profile* and, in particular, the horizontal and vertical profile of the minimum gray level, $\mathbf{h^{mgv}}$ and $\mathbf{v^{mgv}}$ that have usually a characteristic “U” shape with steep

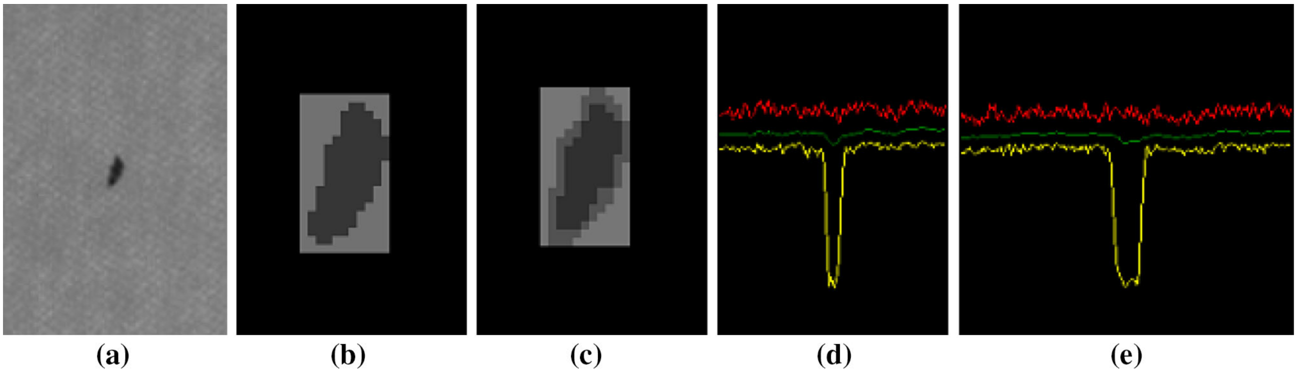
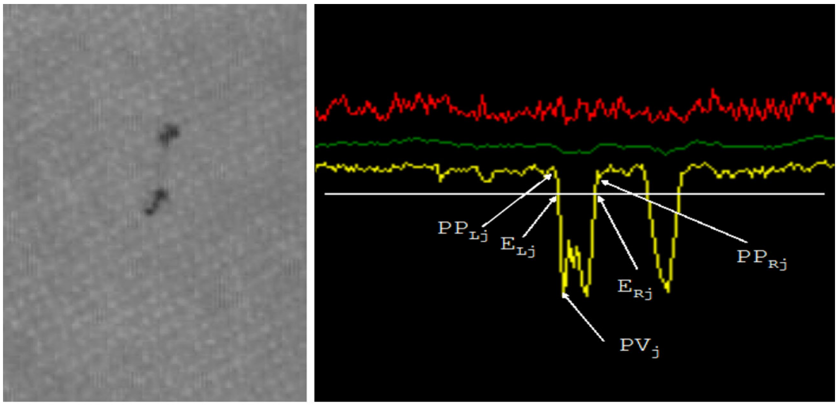


Fig. 7 The image of a typical regular defect (**a**). The same image clustered into two (**b**) or three (**c**) clusters is shown with a ×4 zoom. The *horizontal* profiles of the *gray* levels is reported in panel (**d**): *upper* curve—minimum gray level; *middle* curve—mean gray level; *lower* curve—maximum gray level. The same curves referred to the *vertical* profiles are reported in panel (**e**)



(a) (b)

Fig. 8 A typical irregular defect is reported in (**a**) along with its *vertical* profile (**b**). The points used to compute the peaks properties are evidentiated, see text for details

slopes (cf. Fig. 7d–e). To characterize the profile transition, we have proceeded as follows. Let us consider the horizontal profile (the same derivation applies to the vertical profile) and intersect it with the line $y = 0.9 * \bar{\mu}^{mgv}$, where $\bar{\mu}^{mgv}$ is the mean gray level of the profile. This produces a set of N intersection points, $\{\mathbf{IP}\}$, which identify N/2 peaks (Fig. 8b). Let us call $\mathbf{IP_{Lj}}$ and $\mathbf{IP_{Rj}}$ the two IPs associated to the j-th peak and $\mathbf{P_{Vj}}$ the point associated to the minimum, $x_{\mathbf{PV_j}}$, inside this interval. From these three points, the left base point of the peak, $\mathbf{PP_{Lj}}$ is identified by minimizing the following cost function:

$$H\left(\mathbf{PP_{L_j}}\right) = \sum_{t_1 = x_{\mathbf{IP_{L_j}}}}^{x_{\mathbf{PP_{L_j}}} - Z} \left\{\left(h^{mgv}\left(t_1\right) - \mu_{1,j}\right)^2\right\} + \sum_{t_2 = x_{\mathbf{IP_{L_j}}} + 1}^{x_{\mathbf{PV_j}}} \left\{\left(h^{mgv}\left(t_2\right) - \mu_{2,j}\right)^2\right\} \qquad (6)$$

where μ_{1j} and μ_{2j} are the mean value of the gray levels computed between $\mathbf{IP_{Lj}}$ and $\mathbf{PP_{Lj}}$ and $\mathbf{IP_{Lj}} + 1$ and $\mathbf{PV_j}$ respectively and Z allows considering several pixels outside the peak interval, a value Z = 10 is considered here. The right base point of the peak, $\mathbf{PP_{Rj}}$, is identified through an analogous cost function.

The properties of each peak are then evaluated inside the segment between the two points $\mathbf{PP_{Lj}}$ and $\mathbf{PP_{Rj}}$ through the following features: *mean and standard deviation of the gray levels profile* and *variability of the first and second derivative* inside the histogram segment associated to the peak. The latter are measures of uniformity inside the defect area. The variability of the derivatives is computed with the l^1 norm to penalize possible outliers. An additional feature is the *number of peaks*, which is usually equal to one for the regular defects.

4.2 Irregular Defects

No specific processing has been developed for this class as the defects in this class are those which cannot be classified in the other classes.

4.3 Elongated Defects

Elongation is evaluated as the ratio between perimeter and area. To this aim, the *major and minor singular values*, w_{ii} in (4), are normalized and assumed as features along with their *ratio*. The occupancy rate, that is the *percentage of defect pixels* inside the bounding box tight, B_{RB}, is also considered, as the defects of this class tend to better fill their bounding box (cf. Fig. 6b).

4.4 Threads

Threads come usually from packaging and they are characterized by a small almost constant width and some variability in the local orientation (cf. Figs. 3d, 9). They are identified analyzing the statistics of the local thickness and orientation determined from all the points, **p***, of the defect.

A voting scheme has been implemented. Eight orientations, equally spaced by $\pi/8$, $\{l_j, \mathrm{j} = 1..8\}$ are considered for each point **p***: the local thickness in **p*** in the direction l_j is estimated as the number of white pixels, $W_{\mathbf{p}*;l_j}$, measured along that direction. To get a metric measure we multiply this value for the length of l_j inside each pixel, that is by $\sqrt{5/2}$, for $l_j = \pi/8, 3/8\pi, 5/8\pi, 7/8\pi$ and by $\sqrt{2}$ for $l_j = \pi/4, 3/4\pi$. The minimum of the eight $W_{\mathbf{p}*;l_j}$ is assumed as the absolute local thickness in **p***, $W_{\mathbf{p}*}$, and the distribution of the $W_{\mathbf{p}*}$s over the whole image is evaluated through the *mean, the standard deviation and the maximum thickness*. These features tend to assume values lower in threads than in the other classes.

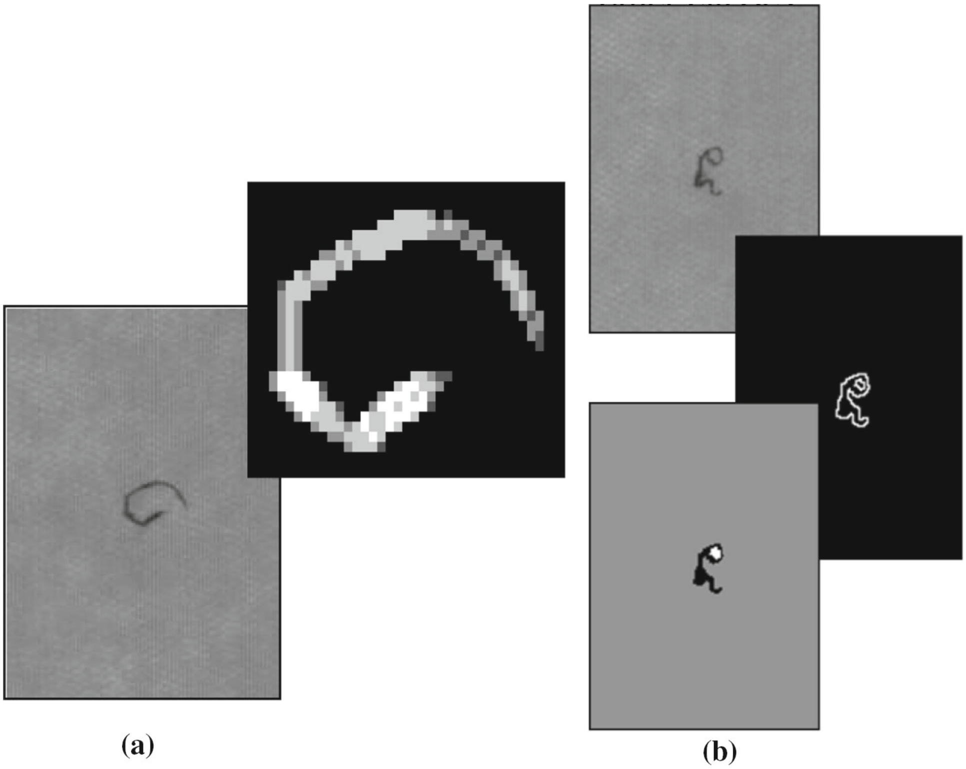


Fig. 9 Typical threads images: in **a** the distribution of local thickness is highlighted in a *gray* levels *scale*. In **b** a loop in the upper portion of the defect is identified

Another characteristic of threads is local orientation. In fact, they may have few preferred directions (cf. Fig. 3d), especially when they are metallic staples. Defect orientation can be assessed by using the distribution of the $W_{\mathbf{p}^*;l_j}$ in a somehow complementary way. First the eight $W_{\mathbf{p}^*;l_j}$ are normalized to obtain a vote, $V_{\mathbf{p}^*;l_j}$, between 0 and 1, associated to each direction:

$$V_{\mathbf{p}^*_j} = \frac{W_{\mathbf{p}^*;l_j}}{\sum\limits_{j=1}^{8} W_{\mathbf{p}^*;l_j}} \tag{7}$$

the larger is the vote, the more the defect is elongated in that direction. The votes collected from each point **p*** for each direction, j, is added in an accumulator, A_j, one for each direction. When all the **p*** have been examined, the A_j are normalized again between 0 and 1, to allow comparison of defects that have different dimension and occupy a different number of pixels in the image.

$$A_j = \frac{A_j}{\sum\limits_{j=1}^{8} A_j} \tag{8}$$

A_j represents therefore a vote for each of the eight directions for the whole defect. The A_js are then sorted in decreasing order, such that A_0 is associated with the preferred defect orientation. The following features are then computed: the *sum of the first two votes*, that is higher the more the defect assumes a predominant direction, and the *sum of the votes of the last three and the last two directions*, that indicate the presence of one or more predominant directions. Additional information is conveyed by the orientation *standard deviation* computed on the A_j with the rationale that the lower the standard deviation the more isotropic is the defect.

4.5 Insects

Insects fly by the running material and may become trapped and stamped on the tissue. They can be described as a body with legs and/or wings. The body is usually uniform and can be assimilated to an elongated or regular defect. Wings tend to assume a light gray colour and they could be identified by intra-cluster variability (cf. "Regular defects" section). Legs are usually constituted of small line segments which can be highlighted by high-pass spatial filtering, for instance through Sobel operator. Features aimed at determining the dispersion of the defect in the image help in the identification of the insects too. Therefore, the following two additional features are considered: *the ratio between the mean defect thickness and its perimeter and its connected component*. The latter is defined as the set of all the dark (defect) pixels connected to at least another dark pixel.

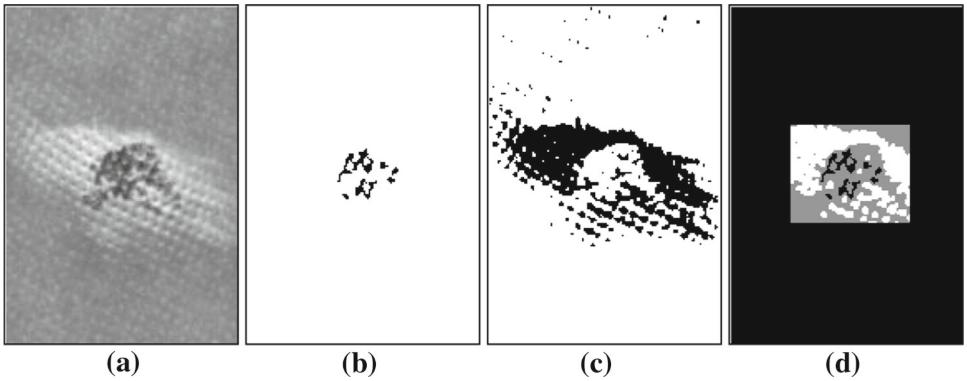


Fig. 10 A dense spot (**a**). Its binarized image, $\mathbf{I^B}$,(**b**). The image of the pale *gray* area detected through (9), $\mathbf{I^W}$: the "white" area is shown as *dark* pixels here (**c**). Three-clusters clustering carried out on the enlarged bounding *box*, B_{BE}, considering the pixels over threshold in both $\mathbf{I^W}$ and $\mathbf{I^B}$(**d**)

4.6 Dense Spots

These defects occur when stretched fibers pile up and appear on the image as a dark nucleus surrounded by a characteristic pale gray / white shadow (cf. Fig. 10a). This pale gray area is almost unique of this type of defect as all the other defects produce pixels darker than the background. To search for pale areas, we observe that the binarization produce an image $\mathbf{I^B}$ containing only the nucleus of the dense spot (Fig. 10b). A second binarized image containing only the pale gray pixels, $\mathbf{I^W}$, can be created thresholding $\mathbf{I^0}$ with the following threshold value, T^W (Fig. 10c), lower than T^B in "Background subtraction (binarization)" section:

$$T^W = \mu_B^0 - K_D * \sigma_B^0 \tag{9}$$

with K_D empirically set to 4 with a procedure similar to that described in "Background subtraction (binarization)" section. A bounding box enlarged by 50 %, B_{BE}, is created around $\mathbf{I^B}$ (cf. Fig. 10d) and the following features are computed inside B_{BE}: the *ratio* berween *the amplitude of the core area*, measured as the number of the defect pixels in $\mathbf{I^B}$, and the *overall dense area*, measured as the number of pixels in $\mathbf{I^W}$; and the *mean plus standard deviation of the gray levels computed for all the pixels belonging to* $\mathbf{I^W}$ and lie inside B_{BE}.

4.7 Folds

Folds are introduced by an error in the control of the tensing mechanism that drives the smooth motion of the tissue on the assembly line. They appear as dark stripes as large as the entire image (Fig. 11a). Therefore a first feature considered is the *height of their bounding box*, which is as large as the image for this type of defects. A second feature is derived from the observation that in the first few rows of the image, on the top and on the bottom, the standard deviation of the gray levels with respect to the clean image increases, while the mean gray value remains almost constant. This is captured by the *mean value less the standard deviation of the gray levels* of the first few rows of $\mathbf{I^0}$.

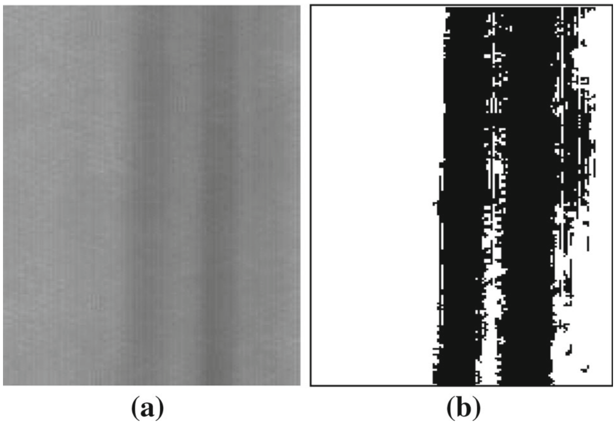


Fig. 11 A typical fold defect is shown in (**a**). It extends over the whole *vertical* dimension of the image. The *vertical* extension of the defect is evident in the binarized image shown in (**b**)

5 Adaptive Labelling Through Boosting

Classification is based on boosting [32]. This linearly combines a set of elementary binary classifiers, $h_{f,\vartheta}(\mathbf{I^0})$, each working on a single feature, to obtain a global robust classifier. $h_{f,\vartheta}(\mathbf{I^0})$ maps a defect image, $\mathbf{I^0}$, into a binary value, ±1, depending on the comparison of the value assumed by a feature f, computed from $\mathbf{I^0}$, with a threshold ϑ:

$$h_{(f,\vartheta)}\left(\mathbf{I^0}\right) = \begin{cases} -1 & if \quad f < \vartheta \\ +1 & if \quad f \geq \vartheta \end{cases} \quad \vartheta \in R \tag{10}$$

The elementary classifiers do not produce alone a satisfactory result. For instance, an elementary binary classifier working on the best feature (the *sum of the first two votes*, "Threads" section) is able to correctly classify no more than 16 % of thread defects. However, a robust classifier of $\mathbf{I^0}$, $H(\mathbf{I^0})$, can be obtained by linearly combining a set of T elementary classifiers into what is usually called a committee of weak classifiers [34]:

$$H\left(\mathbf{I^0}\right) = sign \sum_{t=1}^{T} \alpha_t h_{f(t),\vartheta(t),t} \left(\mathbf{I^0}\right) \tag{11}$$

with $\alpha_t \in$ R. The global classifier is built incrementally: at each iteration a new binary classifier is added in three steps: choice of the feature (and therefore of the binary classifier and of its threshold), computation of its associated coefficient and update of the weight of all the images. The choice is such that the maximum reduction of the classification error is obtained at each iteration for the given data set (DS) [31,32].

When more than one class is defined, as in this case, a multi-label version of boosting, named AdaBoost.MR [34–37], has been proposed. In AdaBoost.MR not only the correct label of an item has to be provided but also the rank of the image with respect to the different classes. Once trained, the global classifier is able to provide the rank of any new image with respect to each of the L different classes: $l_1, l_2 \ldots l_L$. In this case, the output of the classifier would be a function of both the current image, $\mathbf{I^0}$, and the class label, l_k:

$$H\left(\mathbf{I^0}, l_k\right) = \sum_{t=1}^{T} \alpha_t h_{f(t),\vartheta(t),t} \left(\mathbf{I^0}, l_k\right) = C_k \tag{12}$$

with $C_k \in \Re$ and $C_j > C_i$ holds if the image $\mathbf{I^0}$ is more likely to belong to class j than to class i. Each elementary classifier, $h(.)$, outputs a different value for each class, l_k, and the performance of the global classifier, $H(.)$, can be evaluated counting the number of ordering errors, called *ranking loss*, r_{loss}.

Such ranking information is not present here as the domain expert assigns only the true defect label to each image and, as far as we know, training results in this situation have not been examined so far. As only the defect class is given, we consider only the *crucial* ordering errors that occur when the first ranked class is not the true one and compute, r_{loss} as:

$$r_{loss} = \sum_i \left(\hat{l}_{\mathbf{I}_\mathbf{i}^\mathbf{0}} : \hat{l}_{\mathbf{I}_\mathbf{i}^\mathbf{0}} \neq \underset{l_k}{Max}\, H\left(\mathbf{I}_\mathbf{i}^\mathbf{0}, l_k\right)\right) \tag{13}$$

where $\hat{l}_{\mathbf{I}_\mathbf{i}^\mathbf{0}}$ is the highest ranked class, provided by $H(.)$ for image $\mathbf{I}_\mathbf{i}^\mathbf{0}$.

The classifier behaviour is uniquely determined by the parameters in (12), that is the number of elementary classifiers in the committee, T, the feature on which each of them operates, $f(t)$, the associated threshold, $\vartheta(t)$ and the coefficient α_t. The minimization of Eq. (13), with respect to these parameters, cannot be achieved in closed form and an iterative procedure is usually adopted [34]. At each step, all the images in the data set are examined and an elementary classifier is added to the committee, such that (13) can be maximally reduced. To this aim, the following weighted error, r_t, is minimized at each step, t, by comparing the classifier output for class l_j to the output of the classifier for the true class l^*:

$$r_t = \sum_{i,l_j \neq l^*} D_t\left(\mathbf{I}_\mathbf{i}^\mathbf{0}, l_j, l^*\right) e^{\frac{1}{2}\alpha_t\left(h_t\left(\mathbf{I}_\mathbf{i}^\mathbf{0}, l_j\right) - h_t\left(\mathbf{I}_\mathbf{i}^\mathbf{0}, l^*\right)\right)} \tag{14}$$

We observe that $h_t\left(\mathbf{I}_\mathbf{i}^\mathbf{0}, l_j\right) - h_t\left(\mathbf{I}_\mathbf{i}^\mathbf{0}, l^*\right)$ gives a negative contribution if ranking is correct, a positive one otherwise. Therefore, if the image $\mathbf{I}_\mathbf{i}^\mathbf{0}$ is correctly classified to the class l^*, r_t is decreased, while it is increased otherwise. The role of $D(.)$ is to give a larger emphasis to those images for which the correct classification is most problematic at step t.

The first step of each boosting iteration, is to identify the weak classifier, $h_t(.)$, that is to find the feature-threshold pair: $\{f_t, \vartheta_t\}$, which minimizes (14), and to add it to the global classifier, $H(.)$ in (12). Following [34], to reduce the memory storage required in (13) from N^2 to N x L, $D(.)$ is split into the following product:

$$D_t\left(\mathbf{I^0}, l_j, l^*\right) = v_t\left(\mathbf{I^0}, l_j\right) v_t\left(\mathbf{I^0}, l^*\right) \tag{15}$$

Plugging (13) and (15) into (14), we obtain the expression of the ranking loss, r_t, as:

$$r_t = \sum_{i,l_k} v\left(\mathbf{I}_\mathbf{i}^\mathbf{0}, l_k\right) e^{\frac{1}{2}\left[Y_i(l_k) h_t\left(\mathbf{I}_\mathbf{i}^\mathbf{0}, l_k\right)\right]} \tag{16}$$

with $Y_i[l_k] = +1$ if the image $\mathbf{I}_\mathbf{i}^\mathbf{0}$ belongs to class l_k, that is $l_k = l^*$, $Y_i[l_k] = -1$, otherwise. Therefore, at each step, we identify the $h_t(.)$ that miminizes (16).

The second step of each iteration is the computation of the linear coefficient, α_t in (12) that according to [32,34,35] is obtained as:

$$\alpha_t = \frac{1}{2} \ln \left(\frac{1 + r_t}{1 - r_t}\right) \tag{17}$$

with

$$r = \sum_{i,k} d\left(\mathbf{I}_\mathbf{i}^\mathbf{0}, l_k\right) Y_i[l_k] h\left(\mathbf{I}_\mathbf{i}^\mathbf{0}, l_k\right) \tag{18}$$

and

$$d\left(\mathbf{I}_\mathbf{i}^\mathbf{0}, l_k\right) = \frac{1}{2} v\left(\mathbf{I}_\mathbf{i}^\mathbf{0}, l_k\right) \sum_{l_j : Y_i[l_j] \neq Y_i[l_k]} v\left(\mathbf{I}_\mathbf{i}^\mathbf{0}, l_j\right) \tag{19}$$

The last step is the update of the weighting function: v(.),according to [32]:

$$v_{t+1}\left(\mathbf{I}_\mathbf{i}^\mathbf{0}, l\right) = \frac{v_t\left(\mathbf{I}_\mathbf{i}^\mathbf{0}, l\right) e^{-\frac{1}{2}\left(Y_i[l] h\left(\mathbf{I}_\mathbf{i}^\mathbf{0}, l\right)\right)}}{\sqrt{Z_t}} \tag{20}$$

where $\sqrt{Z_t}$ is a normalization factor which makes v_{t+1} a distribution and it is equal to:

$$Z_t = \sum_{i,l} v_t \left(\mathbf{I}_\mathbf{i}^\mathbf{0}, l\right) e^{\frac{1}{2}\left|a_t h_t\left(\mathbf{I}_\mathbf{i}^\mathbf{0}, l\right)\right|} \tag{21}$$

$\mathrm{v}_t(.)$ is set equal for all the images, and all the labels pairs at start. Taking into account that only the true label is considered in the ranking loss function (13), Z_t is set here equal to $\sqrt{L(N-1)}$.

6 Enhanced Features

As shown in the next Section, the classifier employing basic features does not exhibit good performances on defects such as "Insects" or "Threads". Although in upper panel of Fig. 13, the insect is indeed clearly different from other defects, in most of the cases (e.g. lower panel of Fig. 13), insects tend to get confused with other classes (cf. Fig. 21). Similar problems are observed with threads. To improve classification performance we had to design a set of enhanced features.

6.1 Enhanced Features for Threads

One of the main problems with threads is that, after binarization, the border of the defect becomes discontinuous and local thickness can be underestimated, producing a thickness variability larger than the true one. To avoid this, a more robust identification of the local width is achieved filling in the possible holes in the defect. Morphological operators have been avoided as they would blur dense spots making them unrecognizable. Instead, to this aim, the pixels are analyzed in rows and columns and filling in is carried out when up to Q consecutive white pixels are limited by two defect (black) pixels by turning into black (defect) the pixels in between (cf. Fig. 12). Q depends on image resolution and it was experimentally set equal to 4. The following features, similar to those defined in "Threads" section, are added: *mean, standard deviation and maximum local width.*

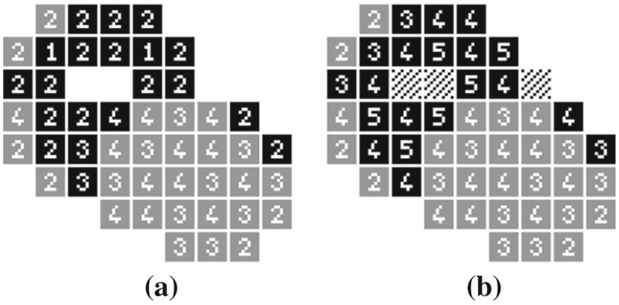


Fig. 12 In (**a**) the local width of a defect, computed with the method described in "Regular defects" section, is written inside each pixel. The local width after filling in rows and columns of pixels is reported in (**b**). The pixels for which the local defect width is different are shown in *black*, those for which it does not in *light gray*. Notice how the true local defect width is better approximated in (**b**)

6.2 Enhanced Features for Insects

Because of background inhomogeneity and texture, and because legs tend to assume gray levels close to the background, legs and sometimes wings are filtered out during binarization (Fig. 13b) and insects gets easily confounded with regular or irregular defects. To avoid this, specific processing has been developed to locally tune the binarization threshold around the insect body. Algorithms for structure detection like anisotropic diffusion [45], Sobel filtering [43], adaptive thresholding [46] have not been able to give reliable results, because of the limited area occupied by the defect itself. This tricky problem shares some similarities with vessel tracking in digital angiography [47], plants roots tracking in biology [48], or fibre tracking [49,50] in MRI for which different approaches have been proposed.

However, as insect legs do not have a high contrast with the background these algorithms do not show reliable results and an ad-hoc solution to "recover" insect legs from the image has been developed. This is based on an adequate set of matched filters associated with an a-posteriori evaluation of the filtering result, combined with adaptive thresholding. The procedure is illustrated with the help of Fig. 13. We start from $\mathbf{I}^\mathbf{B}$(Fig. 13b) and consider a rectangular bounding box, enlarged by 50 %, centred in the defect, B_{BE}. The image inside B_{BE} is processed by two banks of matched filters [43], targeted to highlight small linear segments. Each bank has a different resolution: the first bank extracts segments 1.5 pixels wide and 5 pixels long, while the second one slightly larger segments: 2 pixels wide and 6 pixels long (Fig. 13c). The filters in each bank have different orientations, equally spaced by $\pi/12$. The final filtered image, $\mathbf{I}^\mathbf{MF}(x, y)$, is obtained, considering, for each pixel, the maximum gray level obtained after applying the whole set of 24 filters. The gray levels are then stretched between 0 and $2^{\mathrm{K}-1}$ for improving visibility of the filtering result (Fig. 13c).

We can now further refine the binarization threshold. From $\mathbf{I}^\mathbf{MF}$, the co-occurrence matrix, *CR*, [51], is created as follows. For each pixel belonging to $\mathbf{I}^\mathbf{MF}$, $\mathbf{p}^\mathbf{MF}$, the mean and standard deviation, μ^{MF} and σ^{MF}, of the gray levels inside a 3 × 3 window centred in $\mathbf{p}^\mathbf{MF}$, is computed and μ^{MF} and σ^{MF} are then truncated to the closest integer. A histogram of the number of pixels for each pair (μ^{MF}, σ^{MF}) is then created. This has the typical shape shown in Fig. 14a. As can be seen, *CR* is a sparse matrix with values larger than zero concentrated along the main diagonal: for a given gray level, the variability around it is similar in different areas of the defect image and increases with the gray level. We also

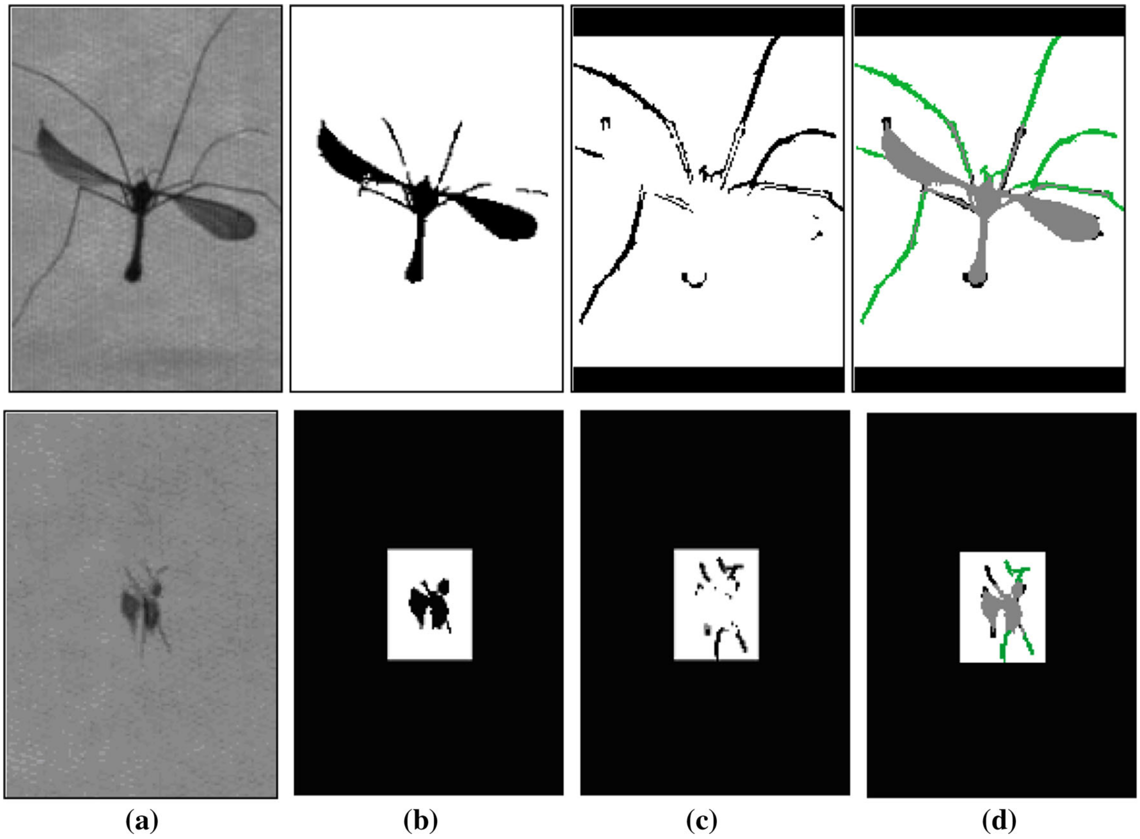


Fig. 13 The original image, $\mathbf{I}^0$, of a large *insect* in the *upper* row and of a small *insect* in the *lower* row (**a**). The same image after binarization, $\mathbf{I}^B$ (**b**); notice that legs are almost missing in both cases. The image, after the application of the two banks of matched filters (**c**). Notice that in the case of large *insect*, the enlarged bounding *box*, BL, is almost coincident with the defect image itself. The final defect shape extracted by the enhanced processing described here (**d**): notice the recovered parts, in *black*

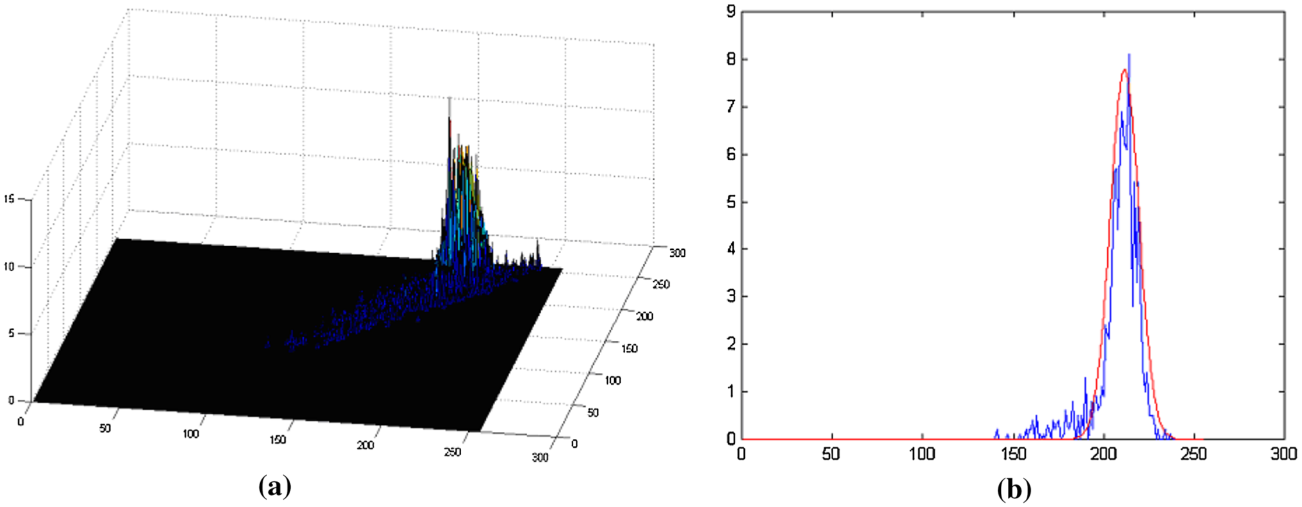


Fig. 14 The co-occurrence matrix associated to the defect in the lower row of Fig. 14 (**a**). Its elements contain the number of co-occurrences of a given mean *gray* level (on the abscissa) and a given standard deviation (on the ordinate), for all the 3 × 3 windows extracted from $\mathbf{I}^{\mathbf{MF}}$. The profile of the co-occurrence matrix computed along the best-fitting line is shown in (**b**) with superimposed its fitting by a Gaussian

remark that a 3 × 3 matrix around $\mathbf{p}^{\mathbf{MF}}$ will contain mainly background. This suggests some regularity in the background pattern. A linear regression is performed to determine the straight line, s^{MF}, which best fits the matrix entries. To the scope, the entries of *CR* are regarded as points in a 2D space (μ^{MF}, σ^{MF}), each weighted with the inverse of the associated co-occurrence count. The co-occurrence count measured on s^{MF} constitutes the co-occurrence profile of $\mathbf{I}^{\mathbf{MF}}$ whose typical shape is plotted in Fig. 14b. This profile is fitted with a Gaussian, whose standard deviation, σ^{HB}, represents a more robust estimate of the local background statistics. Therefore it can be used to obtain a tighter threshold, T^{HB} for local background subtraction (cf. Eq. (2):

$$T^{HB} = \mu - K_T * \sigma^{HB} \tag{22}$$

with K_T experimentally set to 2.5, but smaller values work as well. An enhanced defect image, $\mathbf{I}^{\mathbf{HB}}$, can now be obtained thresholding $\mathbf{I}^{\mathbf{0}}$ with T^{HB}. Such an image does contain legs, wings but it may contain also spurious elements. To filter them out, the over-threshold pixels in $\mathbf{I}^{\mathbf{HB}}$ are first grouped into blobs. These are then analyzed and only those blobs similar to legs or wings are maintained. To the scope, the statistics of the defect local width and local orientation is computed as described for threads in "Threads" section (Eqs. (7, 8). In particular, the mean and standard deviation of the local defect width, μ_S and σ_S, allow determining a degree of compactness, *c*, of the blob:

$$c = (\mu_s + g_s * \sigma_s)^2 / n \tag{23}$$

with *n*, number of pixels of the blob. In the present system, g_s was set to 0.25 but we have experimentally verified that the output of the classifier was robust with respect to variations in the coefficient g_s. If the blob has a small width and it is uniform, μ_s and σ_s assume small values and *c*, in turns, assumes a small value, too. To be considered a leg the blob should also be oriented along a preferred direction. This is evaluated through the mean value of the standard deviation of the votes assigned to each direction (7), μ_{σ_V} and the standard deviation of the votes assigned to the eight equally spaced directions, $\{A_j\}$ in (8), σ_{V}. These considerations are lumped as follows:

$$b = (\mu_{\sigma_V} + g_s * \sigma_v)/c \tag{24}$$

b turns out to be a reliable estimate of the spatial distribution of the blob: the larger is its value, the more the blob is close to a segment, and therefore close to a leg or a wing. Therefore spurious parts can be filtered out if $b > T^{SP}.T^{SP}$ was set to 0.75, considering that the matched filters were aimed in finding structures not larger than 2 pixels. This is a very conservative value as we prefer to loose part of a leg rather than consider a spurious part as an insect leg or wing.

At the end of this processing, we extracted the *Number of additional parts, their mean width and standard deviation.* Although these features are robust for insects they are not sufficient to discriminate them from the other classes and in particular from threads and irregular defects.

7 Features Evaluation

A statistical framework has been developed to evaluate the features in relationship to their capability of correctly classifying the defects. To this aim, the mean, the standard deviation, the minimum and the maximum value of each feature is computed for each class.

We first verify if any feature is able, alone, to distinguish one class from all the others (Fig. 2). Two features: *bounding box height* and *mean less standard deviation of the first rows* assume values in folds images that are significantly different from those assumed in all the other classes. The *bounding box height* is always equal to the maximum image height, 200 pixels, with zero standard deviation for folds, while for the other classes it assumes a value of 29 pixels ± 25 pixels (Fig. 15a). The *mean less standard deviation of the first rows* has less discriminative power: it assumes an average value of 103 ± 7 pixels for folds, and of 123 ± 10 pixels for all the other classes (Fig. 15b). Therefore, using a combination of the two binary classifiers working on these two features, the system is able to classify fold defects always correctly (Fig. 16).

The same result was obtained through boosting. The global classifier in (11) was able to achieve a 100 % detection rate on the fold defects by automatically selecting the two binary classifiers with these two features with an equal weight: $\alpha_t = 0.5$ each.

There are no separating features for dense spots; the two best features: the *ratio of the amplitude of the core and the dense area* and the *mean plus standard deviation of the gray levels* computed for all the pixels belonging to the dense or defect areas (cf. "Dense Spots" section) produce an error rate of 1.5 and 2.3 % respectively. However, a single classifier constituted of two binary classifiers working on these features linearly combined through boosting does allow separating the spots defects from all the others.

No features, with such a discriminative power, can be identified for the other classes (cf. Fig. 17) and we have to resort to multi-label boosting for all the other classes.

To reduce the processing time and avoid overfitting, the number of features used by boosting should be limited. To this aim, the capability of each feature in separating one class from the others is assessed through statistical analysis [52]. We first evaluate the spread of the p^{th} feature, f^p, for each class k, through the intra-class variance, which is defined as:

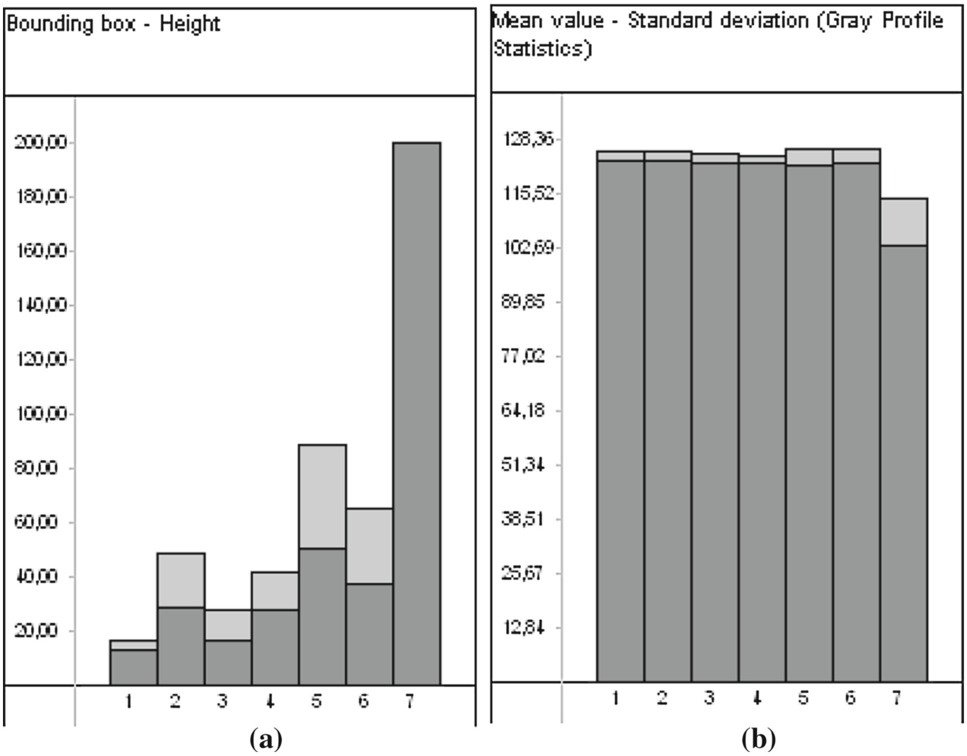


Fig. 15 Features able to identify fold defects univocally. In **a** the mean value (*dark gray*) and standard deviation (*light gray*) of the bounding box height is reported for the seven classes; in **b** the mean *gray* value less the standard deviation of the first rows is reported

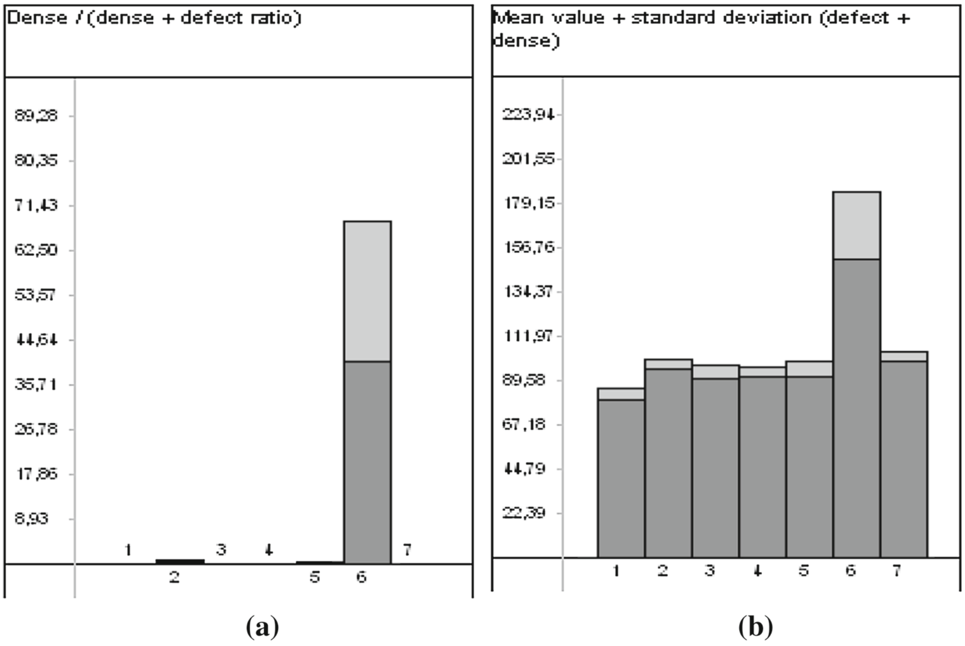
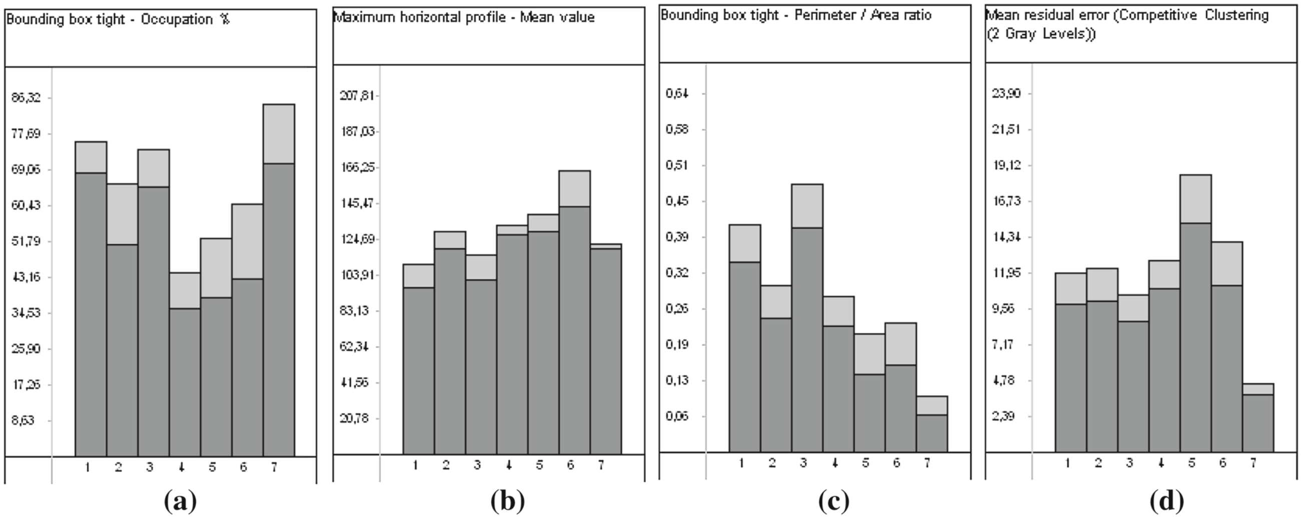


Fig. 16 The value assumed for the seven classes by the two best features for dense spots. Combining, through boosting, these two features a dedicated classifier for dense spots is obtained. The mean value is plotted in *dark gray*, the standard deviation in *light gray*

$$\sigma^2_{IC}\left(f^p_k\right) = \frac{\sigma^2\left(f^p_k\right)}{\left\langle\sigma^2\left(f^p\right)\right\rangle} \tag{25}$$

the smaller is $\sigma^2_{IC}\left(f^p_k\right)$, the more concentrated are the values of f^p for the *k-th* class and therefore the more the feature f^p is able to characterize that class. Another statistical index introduced to evaluate the discriminative power of a feature is the correlation between two features, f^q and f^p, for each class, k. This is computed as:

$$\rho_k(f^p, f^q) = \frac{\frac{1}{N_k}\sum_{i=1}^{N_k}\left(f^q_i - \mu\left(f^q_{k,i}\right)\right)\left(f^p_i - \mu\left(f^p_{k,i}\right)\right)}{\sigma\left(f^q_k\right)\sigma\left(f^p_k\right)} \tag{26}$$



(a) **(b)** **(c)** **(d)**

Fig. 17 The statistics of the following features: percentage of occupancy of the bounding *box* tight (**a**, cf. "Elongated Defects" section), mean value of the *horizontal* profile (**b**, cf. "Regular Defects" section), ratio between the defect perimeter and area (**c**) and mean residual *error* of clustering with two-clusters (**d**, cf. "Regular defects" section). The mean value is reported in *dark gray*, and the standard deviation in *light gray*. As it can be seen the features assume values which partially overlap among the different classes

Lastly, to evaluate the distance between the values assumed by the *p*-th feature in the two classes, the inter-class variability, $ircv(f_k^p)$, has been computed as:

$$ircv\left(f_k^p\right) = \sum_{l=1}^{L} \frac{\left|\mu\left(f_k^p\right) - \mu\left(f_l^p\right)\right|}{\sqrt{\sigma^2\left(f_k^p\right) + \sigma^2\left(f_l^p\right)}} \tag{27}$$

The larger is the inter-class variability, the better the two classes can be distinguished. Eqs. (25–27) are used as follows. First, when two features have a high correlation value, $\rho_k(f^p, f^q) > 0.9$ for all the classes, the one with the larger inter-class variability is discarded. Then, all the possible feature-class pairs are analyzed to discard those features with the least discriminative power. To this aim, for each feature, we evaluate the following conditions:

$$\text{a) } \sigma^2(f_k^p) > 3 \quad \&\& \quad ircv(f_k^p) < 8 \tag{28a}$$

$$\text{b) } ircv\left(f_k^p\right) < 1.5 \tag{28b}$$

If a feature satisfies both conditions for all the classes, that feature is discarded. Thresholds in Eq. (28) were set experimentally considering the images in the whole dataset without any optimization, and results are robust against variations of their values. As such Eq. (28) are general and could be used with other feature sets to evaluate correlation and discriminative power: parameters may make the choice more or less strict.

After the statistical analysis the number of features was reduced from 221 to 71.

8 Results

The defect images considered here were acquired through a Flexin$_{\mathrm{TM}}$ video-inspection module [14] from five different plants, from material running at speeds up to 2m/s. Lighting was adjusted such that defects appear as a dark shadow on a light background and that no defect goes undetected. The system compares in real-time the mean gray level of the current image with that of a reference clean image. When the difference is over a given threshold, the image contains a defect and it is sent to the classification module. Although more refined approaches can be used to detect when a defect is present on the material (e.g. [38,39]), this solution is very reliable in the present domain.

From the whole set of 295 defect images available, for each experiment, a balanced sub-set of images, I_{DS}, was randomly extracted. This set is constituted of 215 images: 40 images of regular defects, 40 of irregular defects, 40 of elongated defects, 50 of threads and 45 of insects. A larger number of thread and insect images was inserted because of the intrinsic difficulties of these two classes. As expected classification error goes to zero in a finite number of iterations (42 in Fig. 18). However, this does not guarantee that the classifier is able to correctly classify any other defect image. To evaluate accuracy in a robust way, cross-validation [52] has been implemented. To this aim, in each experiment, the images in I_{DS} were randomly partitioned into two sets: a training set, I_{DR}, constituted of 172 images (80 % of the defect images) and a test set, I_{DT}, constituted of 43 images. The training set was used to compute the parameters of the classifier (Eq. 12), while classifier accuracy was computed only on the images of

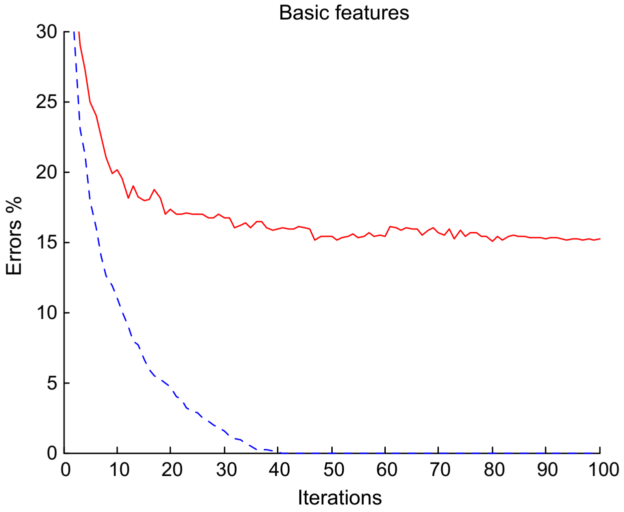


Fig. 18 The classification *error* for the training (*discontinuous line*) and test (*continuous line*) sets is reported for a typical set of experiments in which only basic features were considered. As it can be seen while the training *error* approaches zero quite fast, the test *error* saturates and does not fall below 15 %

the test set. Reported results are obtained averaging the output of 10 experiments in each of which the images were shuffled 5 times randomly. The maximum number of iterations for boosting was set to 100 as no significant improvement is obtained beyond this value; on the contrary, the number of binary classifiers would increase unnecessarily.

We have first implemented single class boosting (11) with the basic set of features. As it can be expected (Fig. 18) the training error on I_{DR} goes to zero quite fast: it is zero already at the 42th iteration on the average. The test error decays also rapidly until the 30th iteration, but afterwards it starts decreasing very little approaching a 15.2 % error at the 100th iteration.

The errors are distributed among the different classes as shown in Table 1. As it can be seen, most of the errors are due to the miss-classification of threads, insects and irregular defects; for each of this class the error is around a few percent, ranging from 2.64 % for irregular defects to 2.88 % for insects. For instance, insects are miss-classified as irregular defects 1.3 % of the times; this means that, on the average, 0.56 images (over the 43 images of the test set) are misclassified as irregular defects, over all the 10 repetitions with 5 different distributions of training and test set images. We explicitly remark that differently from classical applications of boosting, we do not have here false positives, as all the images sent to the classifier by the inspection module contain a defect and only misclassification errors are considered.

Although the misclassification rate is very small for each class, the overall number of errors committed, on the average, over the test dataset is of 15.2 %. Moreover, misclassification of the critical classes, namely that of the insects and that of the threads, has to be avoided completely. No errors, instead, are committed on folds and on dense spots both in training and test images: 100 % of these defects are correctly identified.

A closer analysis of Table 1 shows that the classifier tends to confound insects and threads, mainly with irregular defects. This suggests that the basic set of features assume too close values for these classes to be able to discriminate them. This has prompted us to design improved features for insects and threads (“Enhanced Features” section).

With the new set of features and after pruning (Section 7.2) the overall error drops to 13.6 % as shown in Table 2. Classification errors larger than 1 % are found for the elongated class, where few images are erroneously classified as threads or insects; and in threads, that can be miss-classified as irregular defects. False negatives on insects have been halved decreasing to 1.95 % while they decreased only slightly on threads. This is not sufficient yet to operate the classification system safely.

Another 1.5 % of improvement was obtained applying the enhanced multi-label version of boosting reported in “Adaptive Labelling Through Boosting” section. As shown in Fig. 19 the overall testing error drops to 11.9 %. This is mainly due to the fact that during training the classifier is forced at the same time, to assign an image to the correct class and not to classify it to the other (wrong) classes.

Table 1 Confusion matrix

Basic features single class boosting	Class #1		Class #2		Class #3		Class #4		Class #5	
	#	%	#	%	#	%	#	%	#	%
Confusion with #1	0.0	0.0	0.640	1.488	0.040	0.093	0.0	0.0	0.360	0.837
Confusion with #2	0.48	1.116	0.0	0.0	0.280	0.651	0.480	1.116	0.560	1.302
Confusion with #3	0.16	0.372	0.200	0.465	0.0	0.0	0.480	1.116	0.0	0.0
Confusion with #4	0.0	0.0	1.120	2.605	0.560	1.302	0.0	0.0	0.320	0.744
Confusion with #5	0.0	0.0	0.680	1.581	0.0	0.0	0.200	0.465	0.0	0.0
Total classification errors	0.64	1.488	2.640	6.140	0.880	2.047	1.160	2.698	1.240	2.884

An error average number equal to 1.49, 6.14, 2.05, 2.70 and 2.88 % was obtained for the five classes using basic features only with a total average error of 15.2 %

Table 2 Confusion matrix

Enhanced features	Class #1		Class #2		Class #3		Class #4		Class #5	
	#	%	#	%	#	%	#	%	#	%
Confusion with #1	0.000	0.000	0.520	1.209	0.000	0.000	0.000	0.000	0.320	0.744
Confusion with #2	0.360	0.837	0.000	0.000	0.240	0.558	0.520	1.209	0.320	0.744
Confusion with #3	0.000	0.000	0.160	0.372	0.000	0.000	0.360	0.837	0.000	0.000
Confusion with #4	0.040	0.093	0.600	1.395	0.520	1.209	0.000	0.000	0.200	0.465
Confusion with #5	0.040	0.093	0.680	1.581	0.760	1.767	0.200	0.465	0.000	0.000
Total	0.440	1.023	1.960	4.558	1.520	3.535	1.080	2.512	0.840	1.953

The error rate obtained using the enhanced features is reported; a total average error of 13.6 % is obtained

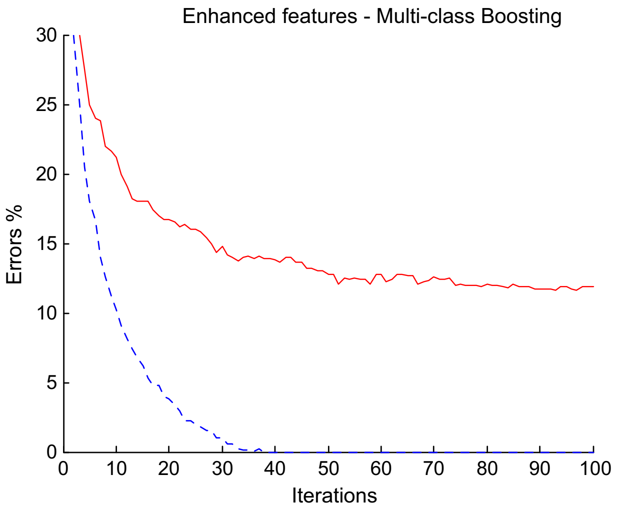


Fig. 19 The classification *error* on training (*discontinuous line*) and test (*continuous line*) sets is reported for a typical set of experiments in which enhanced features and multi-label boosting are considered. Notice that the *error* is reduced down to 11.9 % on the average

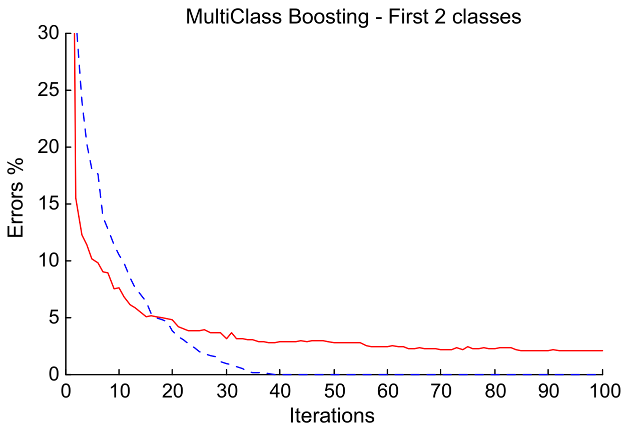


Fig. 20 The classification error for the training (*discontinuous line*) and test (*continuous line*) images is reported for a typical set of experiments in which enhanced features, enhanced multi-label boosting and the first two labels are considered

Although the error could be further reduced designing a second improved set of features, we have chosen here to take advantage of the ability of the classifier to automatically rank the defect images. In fact, a closer view of the classification errors (cf. Figs. 21 and 22) reveals that also the human expert may experience some difficulty in correctly labelling some images; we may regard these images as containing defects that lie ideally at the boundary between two classes. This analysis is supported by the output of the classifier (Fig. 21) that produces similar values for these images. To obtain a robust labelling that guarantees avoiding false negatives especially for the most critical classes, we consider both the first and the second label output by the classifier when their distance is much smaller than that between the second and the third label. This brings the overall error down to 2.33 % on the average (Fig. 20); the error goes to zero for the elongated defects, and almost to zero for all the other classes (Table 3). For instance, only 0.093 % of the insects are erroneously classified as irregular, that means that one image over 129 has been miss-classified. As test sets are constituted of 43 images, randomly extracted at each experiment from the whole data set, this means that one insect image was wrongly classified in a few experiments.

The overall processing time required by the classification module is less than 1s (upper bound), using not optimized code on an old Pentium IV, 2Gbyte of memory, 1.7Ghz. This time is compatible with the requirement of on-line real-time defect classification. We remark here that most of this time is consumed by enhanced feature extraction designed specifically for insect legs and thread identification, which, alone, requires 0.8s. The processing time required by the video-inspection module to extract the defect images is negligible as it requires only computing a difference between two images.

9 Discussion

The system presented here is aimed at Zero Defect Tolerance that, since its introduction in [53], has become the goal of last generation quality control systems. The simplest systems are

Table 3 Confusion matrix

First two classes	Class #1		Class #2		Class #3		Class #4		Class #5	
	#	%	#	%	#	%	#	%	#	%
Confusion with #1	0.000	0.000	0.040	0.093	0.000	0.000	0.000	0.000	0.000	0.000
Confusion with #2	0.000	0.000	0.000	0.000	0.000	0.000	0.080	0.186	0.040	0.093
Confusion with #3	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Confusion with #4	0.040	0.093	0.200	0.465	0.000	0.000	0.000	0.000	0.160	0.372
Confusion with #5	0.000	0.000	0.160	0.372	0.000	0.000	0.080	0.186	0.000	0.000
Total of false negative	0.040	0.093	0.400	0.930	0.000	0.000	0.160	0.372	0.200	0.465

The error rate obtained using enhanced features, enhanced multi-label boosting and the first two ranked classes is reported; a total average error of 2.33 % is produced

based on the identification of specific features on the images [5,10,54–56]. This is not always feasible and combination of features have to be considered to achieve a good classification as in the approach presented here based on boosting.

Features are designed starting from the knowledge of the domain expert who suggests typical characteristics of the different defects. Most relevant features are selected on the basis of their discriminative power: this has allowed discarding here a large number of features: 150 features over 221. We have implemented here a framework based on classical statistics, but any other method suitable to identify the discriminative power of a feature in a statistical framework, like for instance Linear Discriminant Analysis [57], could be used as well.

The analysis of the discriminative power of features has prompted us to design new advanced features targeted to classes difficult to disambiguate, here insects and threads: the first ones get confused with elongated non dangerous defects while the second ones get confused with the other classes as described in Section 6. A more refined consensus based strategy [58] has been introduced to characterize the local thickness and orientation of threads greatly reducing the false negatives on this class. Insects are more complex as they exhibit a large variability. Analysis of binarized images (Fig. 13) has suggested to develop algorithms to better detect their legs. The use of matched filters in combination with the co-occurrence matrix has provided the new features that allow a more robust classification of insects. These novel complex features can be of interest for domains different from the one for which they have been designed.

However, in a few cases, the feature detectors are not able to recover completely the characteristic for which they were designed. For instance, in Fig. 22a, the continuous thread has been captured as two distinct pieces; in Fig. 21e the entire insect wing is not found, due to the closeness of its gray levels to that of the background and in Fig. 22b, the frontal curved small segments could not be found by the matched filters. Although new features could be developed (e.g. curved matched filters), we have explored here a different solution: we accept that, in some cases, the classifier outputs similar values for two classes. In this case, both labels are considered. This is equivalent to hypothesizing that the classifier finds hard to classify an image to either one class or the other. Although the conditions on the features that guarantee this have still to be investigated, it has been verified experimentally that such approach allows dealing with situations that can be considered borderline (Figs. 21, 22). We remark that in these situations also the domain expert was not completely sure on which the correct classification was.

We have implemented classical boosting machinery here using only one label for each image and no ranking information. Enhanced mechanisms, recently proposed [59,60], implement explicit mechanisms to further enhance the output associated to the winning class, modifying the exponential loss function (15). However these implementations would increase artificially the distance between two classes, somehow distorting the similarity measure underlying ranking. The simple squashing function in (10) could be substituted with a continuous function like Gaussians or logistic [61,62]. However, such functions are more suitable to multi-variate regression problems and have been largely adopted in these domains, in which a continuous function is incrementally approximated (cf. also [63]), and simple binary functions have been used here.

We remark that other multi-class labeling procedures could produce a ranking that is proportional to the distance of the classifier output associated to two classes. Such classifiers can be based on soft-classification and be implemented through neural networks, discriminative or statistical methods. Nevertheless, boosting retains the simplicity of adding one feature at a time to the classifier while all other systems usually require that the features are all evaluated in parallel.

In the present work, we suppose that the inspection module does not fail in detecting defects. One of the reasons is that the production line considered are indoor in controlled environments: no variations in illumination are present and optimal

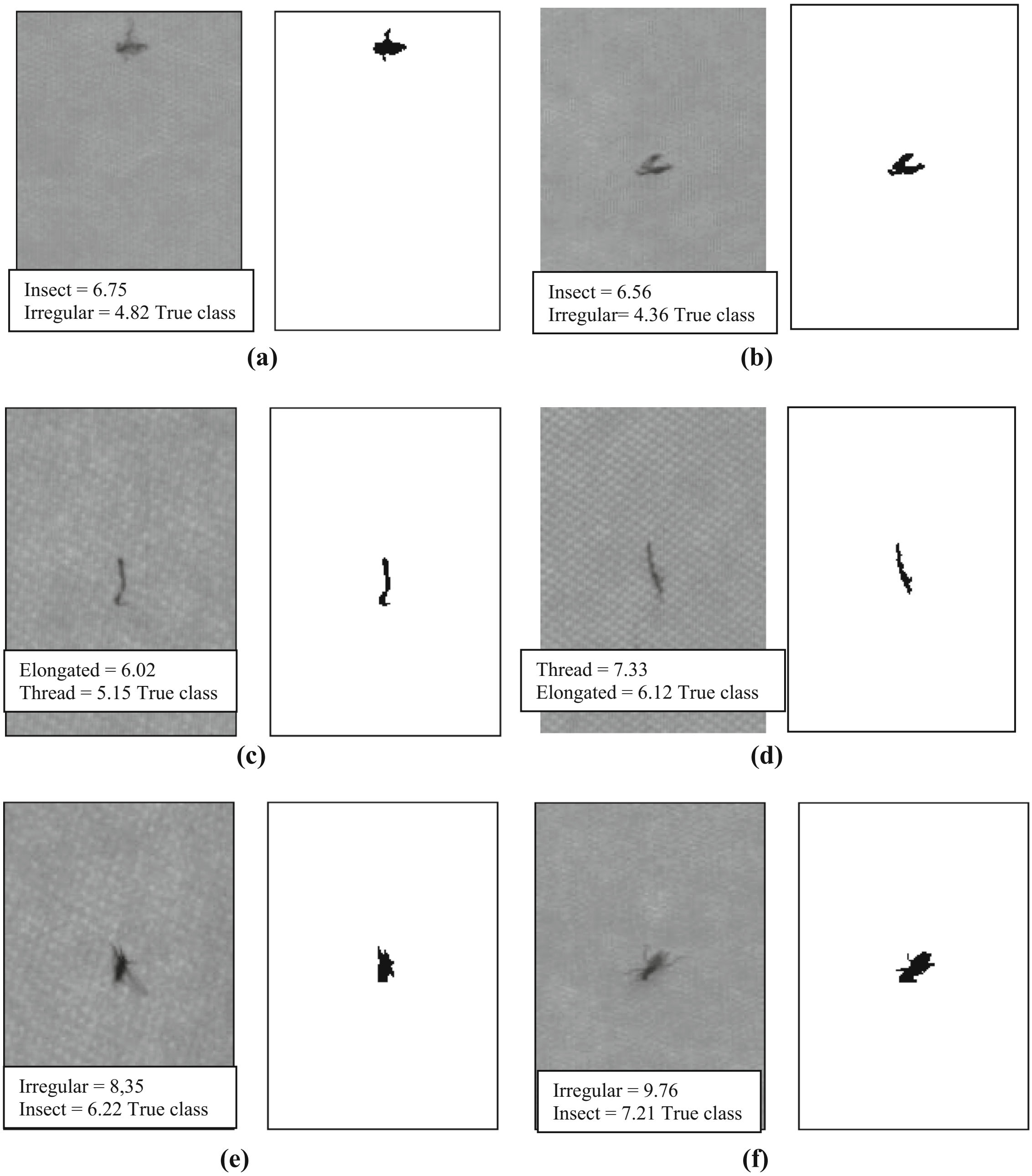


Fig. 21 Pairs of defect images for which the second label is the true one: on the left the original one, on the right the image after binarization. The rank value associated to the first two classes is over imposed. Notice that the value attributed to the first two classes is very close

lighting for a given material can be set. Such module is considered an external module that provides only and all the defect images to the classification stage. If this were not the case, features should be designed also taking into account that false positives may be produced by the inspection module.

Furthermore, the classification system does not depend on the acquisition modality. Whenever line cameras are employed, as it is often the case to work at high speed with reduced costs, gray level calibration is required ("Gray Level Calibration" section).

The system described here can also be easily maintained. When the number of irregular defects increases over average, the presence of new defects, a change in the defect type or issues with the plant can be hypothesized. This could be due to the presence of new insects, of new type of tissue damage besides white spots and folds and so forth. This makes the system able also to provide some form of supervision of the plant.

Updating of the system with new classes is very simple as boosting can be continuously adapted every time a new

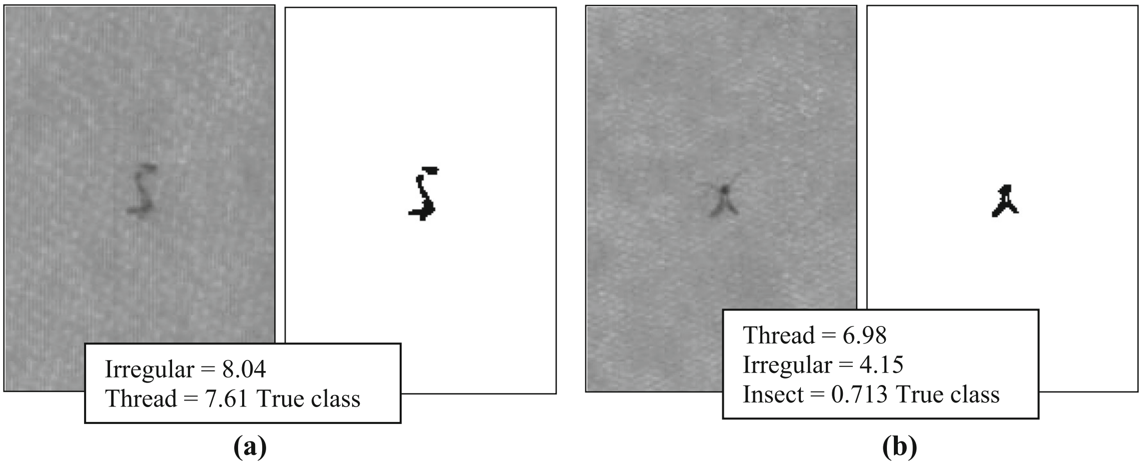


Fig. 22 Two other pairs of critical defect images: on the left the original one, on the right the image after binarization. The rank value associated to the first two classes is over imposed. Notice that in the thread image in panel **a** part of the thread body is missing; this produces the labeling of the defect as irregular. In panel **b** the possible insect antennas are missing in the binarized image; this produces the labeling of the defect as irregular or threads. We remark here that the domain expert, reconsidering these images, was not anymore sure about the true class of these defects

difficult image or a new class, with a new label, is defined. This feature makes boosting particularly appealing for being early deployed in the field in parallel to the human operator, so that it can be trained incrementally from the operator himself who can judge when the system is able to work alone. This makes boosting more flexible for instance than soft-computing approaches based on clustering or mixtures of Gaussians, or SVM classifiers, for which re-learning is often required when new classes are added.

The working hypothesis underlying such system is that defects are rare. If this were not the case, the produced material would be of very poor quality and the production line would have better be cleaned. In this respect, the response time of less than $1s$ can be compatible with real-time operation for a plant with a defect rate $< 1 defect/s$ [14]. We remark here that most processing time is devoted to features identification that can be drastically reduced using modern parallel computing architectures, essentially based on CUDA architectures [64,65].

Overall, the system presented here can be considered an evolution of the expert systems popular in the seventies and eighties based on explicit reasoning or fuzzy systems. Those systems had the limitation that explicit knowledge had to be provided and were not able to generalize to new cases or to learn from examples. The integration of such methodology with methods based on machine learning techniques that can "learn" from the examples and are able to generalize does produce a much higher quality in the results.

10 Conclusion

We have presented here a framework to design an effective defect classification system to be used in conjunction with modern video-inspection modules. Its key elements are robust features design and the use of ranking of an adaptive multi-label classifier trained without specifying ranking. For these reasons such a system can be adopted in new generation inspection systems in the most different industrial environments.

Acknowledgments The authors wish to thank Prof. N. Cesa-Bianchi for interesting and deep discussions on boosting, and the anonymous referees for the precious and focused comments.

References

1. Mery, D., Pedreschi, F.: Segmentation of colour food images using a robust algorithm. J. Food Eng. **66**(3), 353–360 (2005)
2. Ngan, H.Y.T., Pang, G.K.H., Yung, N.H.C.: Automated fabric defect detection–A review. Image Vis. Comput. **29**, 442–458 (2011)
3. Smith, M.: Defect review yields improvements in solar cell manufacturing, Photovoltaic World, May–June 2–9 (2010)
4. Yun, J.P., Choi, S., Kim, S.W.: Vision-based defect detection of scale-covered steel billet surfaces. Opt. Eng. **48**(3), 25–31 (2009)
5. Yeah, C., Wu, F.: An image enhancement technique in inspecting visual defects of polarizers. In: Proceedings international conference iccms, TFT-LCD Industry, pp. 257–261 (2009)
6. Rodil, S.S., Gomez, R.A., Bernardez, J.M., Rodriguez, F., Miguel, L.J., Peran, J.R.: Laser welding defects detection in automotive industry based on radiation and spectroscopical measurements. Int. J. Adv. Manuf. Technol. **49**(2), 133–145 (2010)

7. Frosio, I., Lissandrello, F., Venturino, G., Rotondo, G., Borghese, N.A.: Optimized acquisition geometry for X-ray inspection. Proceedings IEEE international instrumentation and measurement technology conference, pp. 300–305 (2011)
8. Rosati, G., Boschetti, G., Biondi, A., Rossi, A.: Real-time defect detection on highly reflective curved surfaces. Opt. Lasers Eng. **47**(3–4), 379–384 (2009)
9. Molleda, J., Usamentiaga, R., Garcia, D.F., Bulnes, F.G., Emay, L.: Uncertainty analysis in 3D shape measurement of steel strips using laser range finding. Proceedings IEEE International Instrumentation and Measurement Technology Conference, pp. 592–595 (2011)
10. Frosio, I., Tirelli, P., Venturino, G., Rotondo, G., Borghese, N.A.: Flexible and low cost laser scanner for automatic tyre inspection. Proceedings IEEE International Instrumentation and Measurement Technology Conference, pp. 260–264 (2011)
11. Sanchez-Brea, L.M., Gomez-Pedrero, J.A., Bernabeu, E.: Measurement of surface defects on thin steel wires by atomic force microscopy. Appl. Surf. Sci. **150**, 125–130 (1999)
12. Vogel, R., Pollard, L., Yates, R., Beller, M.: Ultrasound tool can combine metal loss and crack inspection of gas pipelines. Pipeline Gas J. **234**(8), 112–118 (2007)
13. Yildiza, K., Buldub, A., Demetgulc, M., Yildiz, Z.: A novel thermal-based fabric defect detection technique. J. Text. Inst. **106**(3), (2015)
14. http://www.electronicsystems.it/inspection-system/
15. http://www.cognex.com/ApplicationsIndustries/TypesOfApplications/
16. http://www.evs-sm.com
17. http://www.thomasnet.com/products/video-inspection-systems-91953661-1.html
18. http://www.key.net/technologies/vision-technology/imaging/lasers/default.html
19. Davies, E.R.: Principles emerging from the design of visual search algorithms for practical inspection tasks. IMVIP08, pp. 3–20 (2008)
20. Montilla, M., Orjuela-Vargas, S.A.: State of the art of 3D scanning systems and inspection of textile surfaces. Proceedings SPIE, Measuring, Modeling, and Reproducing Material Appearance (February 2014)
21. Iivarinen, J., Visa, A.: An adaptive texture and shape based defect classification. Proceedings International Conference on Pattern Recognition, ICPR, Vol. 1, pp. 117–123 (1998)
22. Elbehiery, H.M., Hefnawy, A.A., Elewa, M.T.: Visual inspection for fired ceramic tile's surface defects using wavelet analysis. ICGST Int. J. Graph. Vis. Image Process **5**(2), 1–8 (2005)
23. Jianlia, L., Baoqia, Z.: Identification of fabric defects based on discrete wavelet transform and back-propagation neural network. J. Text. Inst. **98**(4), 355–362 (2007)
24. Wang, J., Jheng, Y., Huang, G., Chien, J.: Artificial neural network approach to authentication of coins by vision-based minimization. Mach. Vis. Appl. **22**(1), 87–90 (2011)
25. Wunsch, D., Xui, R.: Clustering, IEEE press series on computational intelligence (2008)
26. Malhi, A., Gao, R.X.: PCA-based feature selection scheme for machine defect classification. IEEE Trans. Instr. Meas. **53**(6), 1513–1525 (2004)
27. Lee, K., Hyoung, P.: Image segmentation of UV pattern for automatic paper money inspection. Proceedings 11th International Conference on Control Automation Robotics & Vision (ICARCV), pp. 1175–1180 (2010)
28. Murino, V., Bicego, M., Rossi, I.A.: Statistical classification of raw textile defects. Proceedings 7th International Conference on Pattern Recognition (ICPR'04), Vol. 1, pp. 311–314 (2004)
29. Chen, P., Lin, C., Schölkopf, B.: A tutorial on v-support vector machines. Tech. Report. Max Planck Institute for Biological Cybernetics, Tübingen, Germany (2001)
30. Hwa Lee, S., Kim, H., Cho, N., Jeong, Y., Chung, K., Jun, C.: Automatic defect classification using boosting. Proceedings Fourth International Conference on Machine Learning and Applications (ICMLA'05), pp. 45–52 (2005)
31. Freund, Y., Iyer, R., Schapire, R.E., Singer, Y.: An efficient boosting algorithm for combining preferences in Machine Learning. Proceedings Fifteenth International Conference Machine Learning, ICML-98, pp. 176–184 (1998)
32. Schapire, R.E.: The boosting approach to machine learning: an overview. In: Denison, D.D., Hansen, M.H., Holmes, C., Mallick, B., Yu, B. (eds.) Nonlinear Estimation and Classification. Springer, New York (2003)
33. Bellocchio, F., Ferrari, S., Piuri, V., Borghese, N.A.: A hierarchical approach for multi-scale support vector regression. IEEE Trans. Neural Netw. Learn Syst. **23**(9), 1448–1460 (2012)
34. Schapire, R.E., Singer, Y.: Boostexter: a boosting-based system for text categorization. Mach. Learn. **39**(2/3), 135–168 (2000)
35. Tsoumakas, G., Katakis, I.: Multi-label classification: an overview. Int. J. Data Warehous. Min. **3**(3), 1–13 (2007)
36. Mukherjee, I., Schapire, R.E.: A theory of multiclass boosting. In: Advances in Neural Information Processing Systems **23** (2011)
37. Zhu, J., Zou, H., Rosset, S., Hastie, T.: Multi-class AdaBoost. Stat. Its Interface J. **2**, 349–360 (2009)
38. Raiser, S., Lughofer, E., Eitzinger, C., Smith, J.E.: Imapct of the object extraction methods on classification performance in surface inspection systems. Mach. Vis. Appl. **21**(5), 627–641 (2010)
39. Mozina, M., Tomazevic, D., Pernus, F., Likar, B.: Real-time image segmentation for visual inspection of pharmaceutical tablets. Mach. Vis. Appl. **22**(1), 145–156 (2011)
40. Broadhurst, R.E.: Statistical estimation of histogram variation for texture classification. Proceedings 4th international workshop on texture analysis and synthesis. Proceedings ICCV2005, pp. 25–30 (2005)
41. Tavakkoli, A., Nicolescu, M., Bebis, G., Nicolescu, M.: Non-parametric statistical background modeling for efficient foreground region detection. Mach. Vis. Appl. **20**(6), 395–409 (2010)
42. Otsu, N.: A threshold selection method from gray level histograms. IEEE Trans. Syst. Man Cybern. **9**, 62–66 (1979)
43. Gonzalez, R.C., Woods, R.E.: Digital Image Processing, 3rd edn. Prentice Hall, Upper Saddle River (2007)
44. Uchiyama, T., Arbib, M.: Color image segmentation using competitive learning. IEEE Trans. Pattan. Anal. Mach. Intell. **16**(12), 1197–1206 (1994)
45. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. IEEE Trans. Pattern Anal. Mach. Intell. **12**(7), 629–639 (1990)
46. Ng, K.: Automatic thresholding for defect detection. Pattern Recognit. Lett. **27**(14), 1644–1649 (2006)
47. Strother, C.M.: Interventional neuroradiology: 20-year retrospective. Am. J. Neurorad. **21**, 19–25 (2000)
48. Ertz, G., Posch, S.: A region based seed detection for root detection inn minirhizotron images. Lect. Notes Comput. Sci. **2781**, 482-489 (2003)
49. Mori, S., Crain, B.J., Chacko, V.P., Zijl, P.C.: Three-dimensional tracking of axonal projections in the brain in magnetic resonance imaging. Ann. Neurol. **45**, 265–269 (1999)
50. Masutani, Y., Kabasawa, H., Aoki, S., Abe, G., Ohtomo, K.: Pyramidal tract tracking based on presegmentation of superior longitudinal fasciculus and tensor field interpolation. In: Proceedings of ISMRM07, Berlin (2007)

51. Frosio, I., Borghese, N.A.: Statistical based impulsive noise removal in digital radiography. IEEE Trans. Med. Imaging **28**(1), 3–11 (2009)
52. Klotz, J.H.: A computational Approach to Statistics, available on line, http://pages.cs.wisc.edu/~klotz/Book.pdf (2006)
53. Crosby, P.B.: Quality is Free. McGraw-Hill, New York (1979)
54. Jun, J., Lee, J., Kim, J., Park, Y.: Real time crack inspection using the hand held magnetic camera. Proceedings IEEE International Instrumentation and Measurement Technology Conference, pp. 296–300 (2010)
55. Park, M., Jin, J.S., Au, S.L., Luo, S., Cui, Y.: Automated defect inspection systems by pattern recognition. Int. J. Signal Process. Image Process. Pattern Recognit. **2**(2), 124–129 (2009)
56. Fujita, Y., Hamamoto, Y.: A robust automatic crack detection method from noisy concrete surfaces. Mach. Vis. Appl. **22**(2), 245–254 (2011)
57. McLachlan, G.J.: Discriminant Analysis and Statistical Pattern Recognition. Wiley, New York (2004)
58. Yu, X., Guo, L.: Image registration by combining feature consensus with affine model. Proceedings International Conference on Image and Signal Proc., pp. 741–745 (2008)
59. Zhu, J., Zhou, H., Rosset, S., Hastie, T.: Multi-class Adaboost. A multi-class generalization of the Adaboost algorithm, based on a generalization of the exponential loss. Stat. Its Interface **2**, 349–360 (2009)
60. Mukherjee Robert, I., Schapire, E.: A theory of multi-class boosting. Proceedings NIPS, pp. 347–353 (2010)
61. Smola, A., Scholkopf, B.: A tutorial on support vector regression. Stat. Comput. **14**, 199–222 (2004)
62. Friedman, J., Hastie, T., Tibsirani, R.: Additive logistic regression: a statistical view of boosting. Ann. Stat. **28**(2), 337–407 (2000)
63. Ferrari, S., Bellocchio, F., Piuri, V., Borghese, N.A.: A hierarchical RBF online learning algorithm for real-time 3D scanner. IEEE Trans. Neural Netw. **21**(2), 275–282 (2010)
64. Albertazzi, A., Viotti, M.R., Buschinelli, P., Hoffmann, A., Kapp, W.: Residual stresses measurement and lnner geometry Inspection of Pipelines by Optical Methods, Conference Proceedings of the Society for Experimental Mechanics Series, Vol. 999, pp. 1–12 (2011)
65. Matic, T., Hocenski, Z.: Parallel processing with CUDA in ceramic tiles classification knowledge based and intelligent information and engineering systems. Lect. Notes Comput. Sci. **6276**, 300–310 (2010)
66. Geisser, S.: Predictive Inference. Chapman & Hall, New York (1993)