Real-time flaw detection on complex object: comparison of results using classification with SVM, Boosting and Hyperrectangle based method

J. Mitéran, S. Bouillant, M. Paindavoine, F. Mériaudeau, J. Dubois

Le2i Aile des Sciences de l'ingénieur Université de Bourgogne

BP 47870 21078 Dijon

miteranj@u-bourgogne.fr

Abstract

This paper presents a classification work performed on industrial parts using artificial vision, Support Vector Machine (SVM), Boosting and a combination of classifiers. The object to be controlled is a coated heater used in television set. Our project consists of detecting anomalies under manufacturer production as well as in classifying the anomalies among twenty listed categories. Manufacturer's specifications require a minimum of ten inspections per second without a decrease in the quality of the produced parts. This problem is here addressed by using a classification system relying on a real-time machine vision. To fulfill both real time and quality constraints, three classification algorithms and a tree based classification method were compared. The first one, Hyperrectangle based, has been proved to be well adapted for real-time constraints. The second one is based on the Adaboost algorithm, and the third one, based on (SVM), has a better power of generalization. Finally, a decision tree allowing improving classification performances is presented.

1 INTRODUCTION

This paper presents a classification work performed on industrial parts using artificial vision, Support Vector Machine (SVM), Boosting, hyperrectangles, and a combination of classifiers. The object to be controlled is a coated heater used in television set. As shown in Figure 1, a part consists of a spiraled wire (the body) and a straight shape called "legs". The inspection is complex, due to the 3D geometry of the part, and its high intrinsic variability in terms of geometry and coating.

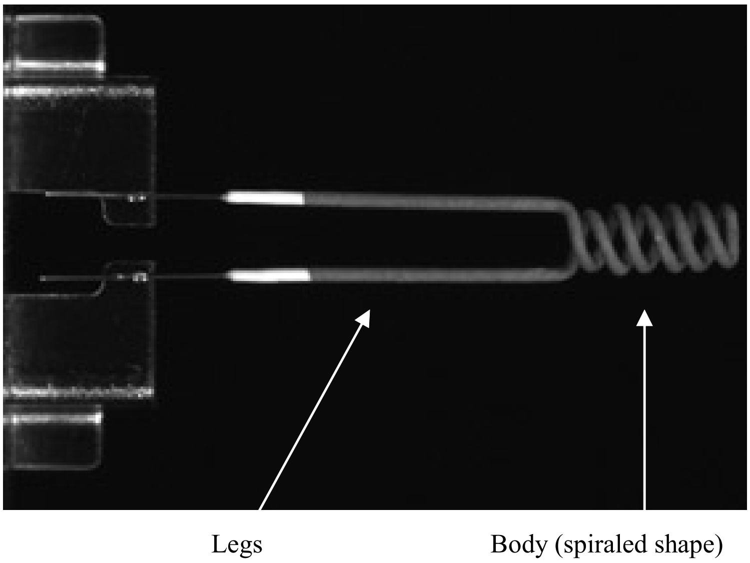

Figure 1: Part to be controlled

Prior to this study, human inspectors were in charge of detecting and classifying the defects. Unfortunately, the time required by the inspection procedure was far too long and the missclassification rate too high. Our project consists of detecting anomalies under manufacturer production as well as in classifying the anomalies among twenty listed categories. Such examples of anomalies are presented in Figure 2. Some flaws concern the coating (local discoloration, buds or scaled surface) and others concern the geometry (joined legs, joined spiral, global deformation of the spiral, etc). Anomalies will affect the temperature properties of the heater and have therefore to be detected. The principal complex task of this project is the presence of numerous acceptable flaws, such as the small local discoloration presented in the top right image of the Figure 2.

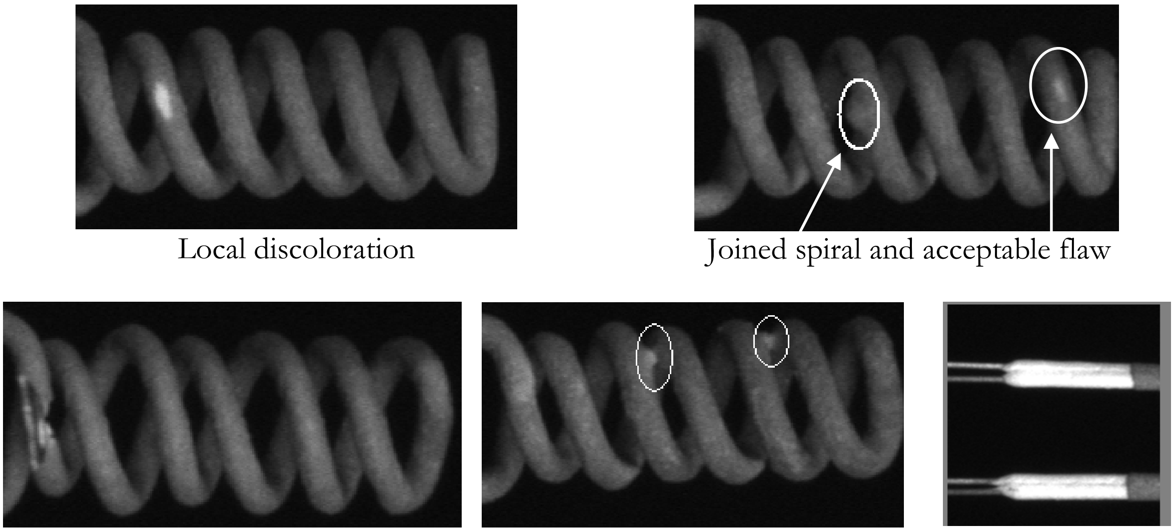

Figure 2: Anomalies to be detected

Manufacturer's specifications require a minimum of ten inspections per second without a decrease in the quality of the produced parts. This problem can be solved with a classification system relying on a real-time machine vision. Numerous papers in the literature [1], [2], [3], [4] describe anomalies detection using a classification method (often neural network based). However, it has been proved in the literature that other methods such as SVM [5] can lead to efficient results in real cases [6], [7], [8], with some advantages compared to classical neural networks (smaller number of parameters to be tuned, higher speed of training step, etc). This method is not yet commonly used in quality control by artificial vision and we will show in this paper that the results can fulfill industrial constraints. We will also show in this paper that in spite of the intrinsic good generalization power of the SVM method, a combination with a single classification tree can improve the final performances. In order to compare the SVM performances, results obtained by a Boosting method [9] and a Hyperrectangle based algorithm [10], [11] are presented.

The first part of the paper presents the acquisition system, the segmentation phase which lead to the features extraction necessary to characterize the spiral shape of the object. Different kinds of classification features (geometrical measurements, statistical information using grey level value, etc) are extracted from processed images. These features will be used as an input of a supervised classification method.

In the second part, the three main classification methods involved in this project are reviewed.

The third part is dedicated to experiments. The three classification algorithms are compared using a preliminary set of features. Two ways of improvement of classification results are then proposed: a feature selection with SVM decision used as a criterion function, a tree-based combination method with SVM. This combination allows mainly decreasing the False Alarm Rate while the Non Detection Rate is maintained within the manufacturer's constraints.

2 ACQUISITION SYSTEM, DEFINITION OF ANOMALIES AND FEATURE EXTRACTION

2.1 Acquisition and lighting system

Images of the parts are obtained with a monochrome mega pixel camera (Jai CV M1). The parts are illuminated by red pulsed LEDs synchronized with the part motion (Figure 3). A lighting system, based on pulsed LEDs allows robust and uniform illumination of the outside of the spiraled part. A rate of 10 (1024x1024 pixels) images per second using a PC-vision board and two Intel Pentium IV 1 GHz processors is reached. This resolution is requested to simultaneously perform dimensional and textural controls. The lighting system and the optical set-up were optimized as referenced in [12] to obtain good quality images.

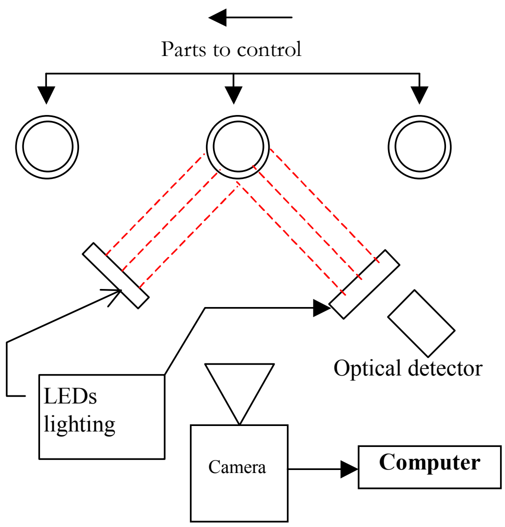

Figure 3: Top view of the acquisition system.

2.2 Definition of anomalies to be detected – sampling

An inventory of 20 anomalies of texture and structure, such as discoloration, scaled surface, stripes, cracks, surface flaws, buds and joined spiral was created.

In this study, our attention was only focused on the five predominant flaws. These five flaws have been selected by the industrial partner as the most critical for the production.

To classify these anomalies and evaluate the performances of the final system, training and test sets were built from 1606 sound pieces and 245 defected pieces (Table 1). Since the number of available samples per categories of flaw is very small, a ten-fold cross validated error was used to evaluate the classification performances.

Table 1: Flaws subsets size.

Description	Subset size
Flawless (Class B)	1606
Local discoloration (flaw D1)	18
Buds (flaw D2)	42
Joined spiral (flaw D3)	24
Scaled surface (flaw D4)	78
Joined legs (flaw D5)	83

2.3 Features extraction

Four major steps are successively achieved on the image:

<u>Localization</u>: during the first step, the spiral is localized in the image. It reduces the computing time of the next processes.

<u>Segmentation</u>: during the second step, the image is segmented in order to isolate the upper side body (spiral shape) from the background scene. This is easily achieved by combining Wen's thresheld

images [13] and Sobel's thresheld images with logical operators. The resulting binary shapes are labeled and each is referred as a seed in a morphological growing algorithm leading, as shown on Figure 4, to the segmentation of the very inner part for each foreground spiral. This part of the spiral will be call "arm" in the rest of the paper.

Figure 4: Image segmentation.

Indexation: the third step is the final labeling of each arm which is identified and classified using its gravity center abscissa (Figure 5). This step, obtained with the well known Connected Components Labeling algorithm, allows a thorough indexation of all arms. During this process, a surface filtering is achieved and artifacts (small regions) which areas are less than an arbitrary set threshold are removed.

Figure 5: Arms labeling.

Features extraction: the fourth step is the extraction of a large set of features on the original images as well as on the processed images. The dimension *d* of the initial feature space is 180. The features have been defined regarding the industrial constraints concerning the dimension of the part, and our prior knowledge about flaws to be detected:

- The "local discoloration" produces a small region brighter than other pixels in the arm. It will be detected by statistical measurements such as standard deviation or mean of luminance. This measurement is performed on the full-arm or can be reduced to smaller analysis windows. Acceptable local discolorations are usually of small area. The measurement of the area of such regions could then be performed.

- "Joined spiral" produces small deformation of the arm shape. The luminance of this deformation is similar to the arm-luminance. Therefore this flaw cannot be detected with the previous features. Nevertheless, it can be detected by measurements such as the mean and the standard deviation of the inter-arm distance, or with the features of the line (slope, y-intercept) which represents the best model of the arm.
- "Scaled surface" produces deformations of the arm shape and generates regions brighter or darker than the arm. It will be detected by a combination of the previous features.
- "Buds" produce small bright regions localised on the border of the arm. The area of this region is usually smaller than the area measured for a joined spiral.
- "Joined legs" produce a dark line in the white part of the leg, and a double bright line in the right part of the leg. The number of legs in this case is easily detected by the measurement of the number of maxima detected in the right part of the leg.

Some of these features are presented in Figure 6: the arms and the global orientation of the spiraled shape are modeled by lines, and the line features are used as classification features. The estimated gravity centers of each arm enable the "horizontal" line to be determined.

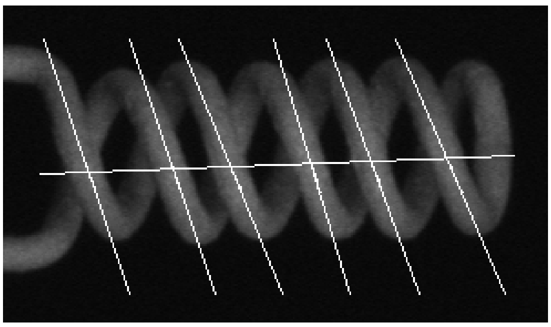

Figure 6: Dimensional features (characteristics of the white lines).

The whole features space (some details are given in Table 2) is made of:

- A subset of 77 (11 values per arm) standard statistical features (mean of luminance, Eq. 1, standard deviation of luminance, local contrast, Eq. 2) computed on each arm in window masks. Window sizes were decided regarding the size of the smallest flaws (mainly local discolorations and buds) to be detected (5x5, 7x7 and 9x9 pixels square).

The local mean $\mu(i,j)$ in a square [*(2N+1)* x *(2N+1)*] neighborhood is defined by:

$$\mu(i,j) = \frac{1}{(2N+1)^2} \sum_{c=-N}^{N} \sum_{l=-N}^{N} z(i+c, j+l) \qquad \text{Eq. 1}$$

where $z(i,j)$ is the grey level of the pixel of coordinates (i,j).

We also use the $V(i,j)$ local contrast in an [*(2N+1)* x *(2N+1)*] neighborhood, which can be expressed as follows:

$$V(i,j) = \frac{z_{\max}(i,j) - z_{\min}(i,j)}{z_{\max}(i,j) + z_{\min}(i,j)} \qquad \text{Eq. 2}$$

with

$$z_{\max}(i,j) = \max\{z(i+k, j+l), -N \leq k \leq N, -N \leq l \leq N\},$$

$$z_{\min}(i,j) = \min\{z(i+k, j+l), -N \leq k \leq N, -N \leq l \leq N\}$$

- A subset of features, also based on arm luminance, is used for the detection of local discoloration. As previously mentioned, this flaw is characterised by a small region of pixels brighter than their neighbours. In order to process these features, two types of arm regions named P_i and Q_i are defined using different thresholds :

The region P_i of the ith arm is defined by the pixels z_k for which luminance is greater than a fixed threshold s_i :

$$z_k \in P_i \Leftrightarrow z_k \geq s_i \text{ with } s_i = \mu_i + n\sigma_i$$

where μ_i and σ_i are the mean of luminance and standard deviation of the ith arm.

The region Q_i is defined using a threshold s'_i:

$$z_k \in Q_i \Leftrightarrow z_k \geq s'_i \text{ with } s'_i = \mu_i + m\sigma_i$$

The value of n and m are empirically defined with $n<m$, ensuring that the regions Q_i are brighter than the region P_i (typically n=3 and m=5).

In order to well classify small acceptable flaws, we compute the area (number of pixels) of each region P_i and Q_i, and the final used features are the maximum value of these areas:

$$A = \max_i(area(P_i))$$

$$B = \max_i\,(area(Q_i))$$

The next feature computed from the simple sum of pixel luminance of pixels that belong to region P_i:

$$C = \max_i(\sum_{j\in P_i} z_j)$$

Thus, if a large area of light pixels is present, the value of B should be high, and if a large area of middle range luminance is present the values of A and C should be high.

- A subset of dimensional and macroscopic textural features aimed at detecting joined arms of spiral flaw and scaled surfaces.

Table 2 Feature set

Feature names	Feature size
Standard statistical features computed using grey level in square windows	11x7
Number of detected arms – S	1
$A = \max_i(area(P_i))$	1
$B = \max_i\,(area(Q_i))$	1
$C = \max_i(\sum_{j\in P_i} z_j)$	1
Number of detected legs – P	1
Arm area	7
Mean grey level of each arm	7
Standard deviation of grey level in arms	7
Out the rectangle of interest	1
Line characteristics of gravity center arms fitting	3
Line characteristics of the 1st order inertial axis, left and right border of each arm fitting	3x7
Mean of inter-arm distance	6
Standard deviation of the inter-arm distance	6
Line characteristics of 1st order legs fitting (upper and lower sides)	2x2
Distance between gravity center of each arm	6
Difference between upper pixel's X-coordinate on two consecutive arm	6
Difference between upper pixel's Y-coordinate on two consecutive arm	6
Difference between lower pixel's X-coordinate on two consecutive arm	6
Difference between lower pixel's X-coordinate on two consecutive arm	6
Difference between line characteristics of 1st order arms inertial axis fitting on two consecutive arms	6

3 REVIEW OF CLASSIFICATION METHODS

As mentioned in the introduction, three families of classifiers were studied in this project. The choice of the first one, hyperrectangles based, is justified by the good results obtained with such a method in other real cases [14], [15] and its ability to be implemented at low cost in hardware [16]. The second one, based on the Adaboost algorithm, is also well suited to fast implementation. The choice of the third one, SVM based, is justified by its important theoretical background in terms of machine learning, its good generalization power often compared to neural networks, and by the existence of powerful training algorithms.

3.1 Hyperrectangles based method

This method divides the attribute (or feature) space into a set of hyperrectangles for which simple comparators may easily satisfy the membership condition. During the training step, one hyperrectangle is built for each learning sample. The bounds of this hyperrectangle are constrained by hyperrectangles of opposite classes. Resulting hyperrectangles are then merged together in the way that there is no possible overlap between hyperrectangles of opposite classes. During the decision step, a new sample is classified regarding its membership to one hyperrectangle of a given class. This method belongs to the same family as the NGE (Nested Generalized Exemplars) algorithm, which performances were compared to the k-nearest neighbor (kNN) [11]. These studies show that, although the division of the feature space is simple, the performance in terms of classification rate is similar to the kNN method and mainly significantly faster during the decision phase. Moreover, it has been shown that the method possesses a good generalization power and works quite well with a poor sampling. This method is easy to use, and can be implemented for real time classification, using hardware [17] or software optimization. It can also be seen as a particular case of Radial Basis Function (RBF) neural network with a simple Heaviside function used as activation function.

3.2 Boosting

The basic idea introduced by Schapire and Freund [18], [19] is that a combination of single rules or "weak classifiers" gives a "strong classifier". Each sample is defined by a feature vector $\mathbf{x}=(x_1, x_2, ..., x_d)^T$ in a d dimensional space and its corresponding class :

$C(\mathrm{x}) = y \in \{-1,+1\}$ in the binary case.

We define the learning set S of p samples as:

$\mathrm{S} = \{(\mathrm{x}_1, y_1), (\mathrm{x}_2, y_2), ..., (\mathrm{x}_p, y_p)\}$.

Each sample is weighted such as after each iteration of the process (which consists in finding the best weak classifier as possible), the weights w of the misclassified samples are increased, and the weights of the well classified sample are decreased.

The final class y is given by:

$$y(\mathrm{x}) = sgn\left(\sum_{t=1}^{T} \alpha_t h_t(\mathrm{x})\right)$$

Where both α_t and h_t are to be learned by the following boosting procedure (Adaboost):

a. Input $\mathrm{S} = \{(\mathrm{x}_1, y_1), (\mathrm{x}_2, y_2), ..., (\mathrm{x}_p, y_p)\}$ **, number of iteration T**

b. Initialise $w_i^{(1)} = 1/p$ **for all i=1, …,** ***p*****, where** $w_i^{(1)}$ **is the weight of the i**th **sample at iteration 1.**

c. Do for t=1, …, T

c.1 Train classifier with respect to the weighted samples set $\left\{\mathrm{S}, w^{(t)}\right\}$ **and obtain hypothesis**

$$h_t : x \rightarrow \{-1, +1\}$$

c.2 Calculate the weighted error ε_t **of** h_t **:**

$$\varepsilon_t = \sum_{i=1}^{p} w_i^{(t)} \mathrm{I}(y_i \neq h_t(\mathrm{x}_i))$$

c.3 Compute the coefficient α_t

$$\alpha_t = \frac{1}{2}\log\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$$

c.4 Update the weights

$$w_i^{(t+1)} = \frac{w_i^{(t)}}{Z_t}\exp\{-\alpha_t y_i h_t(\mathrm{x}_i)\}$$

Where Zt is a normalization constant: $Z_t = 2\sqrt{\varepsilon_t(1-\varepsilon_t)}$

c.5. Stop if $\varepsilon_t = 0$ **or** $\varepsilon_t \geq \frac{1}{2}$ **and set T=t-1**

end for

d. Output : $y(\mathrm{x}) = sgn\left(\sum_{t=1}^{T} \alpha_t h_t(\mathrm{x})\right)$

The weak classifier used in this study is a single hyper plan parallel the feature space axis. The main advantages of this method are the small number of parameters to tune (T, which can be automatically chosen), and the automatic feature selection performed using this weak classifier, since one feature is selected at each step of the Adaboost algorithm.

3.3 S.V.M method

A Support Vector Machine (SVM) is a universal learning machine developed by Vladimir Vapnik [20] in 1979. SVM are known to achieve good results in many pattern recognition problems. SVM performs a nonlinear mapping of the input vector (features) from the input space R_d into a high dimensional feature space H, the mapping is performed by a kernel function. The decision function is:

$$\mathrm{f(x)} = \mathrm{Sgn}\left(\sum_{i=1}^{Ns} y_i \alpha_i \cdot K(\mathrm{s}_i, \mathrm{x}) + b\right) \qquad \text{Eq. 3}$$

Where **x** is the input vector, y is support vector output and in the set {-1,1} and b is a scalar bias.

Here $\boldsymbol{s}_i$ belongs to the set of support vectors defined in the training step, as well as the set of α_i values. We used during this study mainly the RBF Kernel (Eq. 4):

$$K(\mathrm{x}, \mathrm{y}) = e^{\left\{-\frac{\|\mathrm{x}-\mathrm{y}\|^2}{\gamma}\right\}} \qquad \text{Eq. 4}$$

The main advantage of this kernel (which also allows seeing the SVM method as a RBF neural network), is that the only parameter to be tuned is γ. V. Vapnik, in [5], chose γ=1/d where d is the dimensionality of the input space. However, this choice doesn't guaranty that the classification error will be optimum. We then define an iterative algorithm which uses different values of γ and chooses the best following the decision rule, (according to the manufacturer's specifications, for which the False Alarm Rate (FAR) has to be less than 1%, and the Non Detection Rate (NDR) has to be less than 20%):

$$\gamma_{opt} = Arg\min(\lambda(\gamma)) \text{ where } \lambda(\gamma) = Max\left\{\frac{FAR(\gamma)}{1};\frac{NDR(\gamma)}{20}\right\} \text{ and } \gamma \in \left[\frac{1}{d};5\right] \qquad \text{Eq. 5}$$

The rates are here defined as relatives to the class to be detected (NDR=100xnumber of non detected flaw divided by the number of samples of the considered flaw). In terms of manufacturer's constraints, the global system is validated if $\lambda(\gamma) \leq 1$. For each value of γ varying from 1/d to 5, $FAR(\gamma)$, $NDR(\gamma)$ and the criterion $\lambda(\gamma)$ were computed and finally the value of γ which minimizes λ was chosen.

4 EXPERIMENTS

4.1 Preliminary results

As previously mentioned, results were obtained using a ten-fold based Cross-validation with a learning step performed on $\mathcal{B}$, (B is the learning subset) and test step on $\mathcal{T}$, with $\mathcal{T} \cup \mathcal{B} = \mathcal{D}$, and $\mathcal{T} \cap \mathcal{B} = \varnothing$ where *D* is the set of every available piece (flaw and flawless ones). As shown on Table 3, SVM based method performs better classification than both Adaboost and Hyperrectangles methods. These preliminary experiments are performed mainly in order to validate the ability of the candidate features and the candidate classification methods to detect each flaw type. Consequently, each flaw is detected in this case independently from each other. However, even with the SVM algorithm, manufacturer's specifications could not be reached. This is mainly due to the small number of defected pieces available in the training set, regarding to the feature space dimension. Moreover, in spite of its known robustness, the SVM method seems to be sensitive to discreet features such as the number of detected legs, the number of detected arms, or the position of the spiral in the rectangle of interest. We then developed two ways of optimization. The first one is a classical feature selection, which removes irrelevant features (improving in the same time the whole classification speed). The second one is a combination of SVM and a classification tree, extracting from the feature set a subset of binary features.

Table 3: Classification results of 1606 flawless observation vs. one type of flaw, using the full set of features.

	Flaw 1: Local discoloration – 18 pieces			*Flaw 2: Buds – 42 pieces*			*Flaw 3: Joined spires – 24 pieces*			*Flaw 4: Scaled surfaces – 78 pieces*			*Flaw 5 – Joined legs – 83 pieces*		
	λ_{opt}	FAR (%)	NDR (%)	λ_{opt}	FAR (%)	NDR (%)	λ_{opt}	FAR (%)	NDR (%)	λ_{opt}	FAR (%)	NDR (%)	λ_{opt}	FAR (%)	NDR (%)
SVM	**3.00**	0.15	60.00	**1.93**	0.53	38.57	**3.75**	0.09	75.00	**1.50**	0.29	30.00	**0.49**	0.19	9.86
Adaboost	**3,00**	0,00	60,00	**3.33**	0.25	66.67	**2.92**	0.06	58.33	**2.63**	0.56	52.56	**0.71**	0.06	14.29
Hyperrectangle	**5.00**	0.00	100.0	**4.49**	0.00	87.45	**4.37**	0.18	87.50	**3.21**	0.62	64.12	**3.51**	0.44	70.27

4.2 Optimization using feature selection

Several methods are commonly used to perform feature selection. One of the most popular is the Principle Component Analysis [21], [22]. This method has been tested and the results were not relevant for our kind of application. Indeed, the elastic structure of the spiral, creates high dependencies between some features. Moreover, some features had to be linked in order to take into account that some flaws are often not available into the sample set for each arm. Therefore the PCA method can not be applied without linking these features. It would be then necessary to classify manually every arm separately which is a very time-consuming operation incompatible with the manufacturer's constraints .

We implemented then a Sequential Backward Floating Selection (SBFS) [23], [24], [25] and performed a feature selection for each flaw, including the created link between the features in the procedure. The criterion function used to select a feature is based on the results obtained after a learning–test step of SVM method. This is a high computational time method but a more coherent way to select a good subset of features relative to the SVM decision function.

The selected features are for example:

D1 (local discoloration) : standard deviation of the mean of luminance and local contrast computed in 7x7 windows (other window sizes where removed) and the features A, B and C,

D2 (buds) : standard deviation of the mean of luminance computed in 7x7 windows, feature A, B, C, and standard deviation of inter-arm distance, D3 (joined spiral) : number of detected arms, standard deviation of the inter-arm distance, distance between gravity center of each arm, difference between line characteristics of 1st order arms inertial axis fitting on two consecutive arms, mean of luminance

D4 (scaled surface) : number of detected arms, the standard deviation of luminance computed in 9x9 windows, arm area, mean luminance of each arm, standard deviation of the inter-arm distance and of the luminance in arm, A, B, C,

D5 (joined legs) : number of detected legs, line characteristics of the 1st order inertial axis, left and right border of each arm fitting, Oroi, line characteristics of 1st order legs fitting (upper sides).

Table 4: Results after feature selection.

Flaw – Subset size	*SVM RBF method*		
	λ_{opt}	FAR (%)	NDR (%)
Flaw D1 – Subset 1: 20	1.25	0.39	25.00
Flaw D2 – Subset 2: 16	1.75	0.58	35.00
Flaw D3 – Subset 3: 22	3.38	0.24	67.50
Flaw D4 – Subset 4: 39	1.50	0.24	30.00
Flaw D5 – Subset 5: 24	0.37	0.12	7.5

Since flaws are still here considered independently, we chose to optimize the detection of each flaw independently. The final decision can be done by merging the 5 decisions of each elementary classifier. Results presented in table 4 clearly show that the feature selection step enables better detection for all flaws but the third one ("joined spiral"), which is clearly the most difficult to detect. Indeed, joined spiral produces local deformation of the arm, and this should be detected mainly by the measurement of the standard deviation of the arm's width. However, the intrinsic variability of the arm's width is high, explaining the poor results of the detection. Moreover, the too small number of samples (24) does not allow a good training of the classification method. The manufacturer's constraints are reached only in the last case where the detection is easy to perform due to the particular nature of the "joined legs" flaw.

However, the main interest of the manufacturer is to globally detect the defected pieces from the non-defected ones. Flaws from classes 1 to 5 were then merged in a flaw class and a cross-validation with SVM, Adaboost and Hyperrectangles algorithm were performed. Results are summarized in Table 5.

Table 5: Classification rate of flawless vs. flaws pieces.

Methods	*Flawless versus all flaw piece (1606 vs. 245)*		
	λ_{opt}	FAR (%)	NDR (%)
SVM RBF	1.36	1.12	27.20
Hyperrectangles	3.25	1.43	65.04
Adaboost	1.81	1.81	34.15
SBFS SVM	**0.98**	**0.73**	**19.59**

Features selection optimizes the classification rate; however the Non Detection Rate is still important, even with smaller subset of features (the final size of feature set is d=100, after elimination of redundancies between different flaws). In that case, the results obtained with the SVM are better than those obtained with the Adaboost or Hyperrectangles methods. This confirms the idea that SVM is a robust method, with a high generalization power, but still need a high number of samples during the training step.

4.3 Optimization using combination of SVM and a classification tree

The last way we explored to improve the final performances was to combine single threshold based decision tree with a SVM based model. As presented in [26], it is possible to improve classification performances by combining classifiers using single rule such as logical operations. This is also one of the principles of the Adaboost method. We herein combine logical operations and a tree-based method [27]. The first node of the tree is made from the features which are manually extracted using *prior* knowledge defined by the manufacturer:

- Piece has 2 legs – feature P– Yes or No,
- Piece has 2 white parts called boots– feature B – Yes or No,
- Piece has the right number of spirals (depends of the manufacturers specifications) – feature N– Yes or No,
- Piece out of the region of interest – feature D – Yes or No.

These features have been chosen because of their Boolean nature. If one of these constraints is not validated, then the piece is faulty and must be rejected. Other features are mainly continuous values, and are better adapted to the SVM method.

The combining rule is a logical AND with the aforementioned Boolean features. The second node is made with a SVM model or Hyperrectangle based classifier built using all other features. The learning step algorithm is presented below:

a-For every piece X with N dimensional pattern $\mathrm{x} \in \mathcal{B}$

If $P(\mathrm{x}).B(\mathrm{x}).N(\mathrm{x}).D(\mathrm{x}) = 1 \ \mathrm{x} \in \mathcal{B}'$ **,** $\mathcal{B}$**' a new learning set and where . is the boolean operator 'AND'**

Then the piece is rejected (flaw)

End if

End for

b- A learning step is performed with a classification method on $\mathcal{B}$**' and built a model** $\mathcal{M}$ **using a classification algorithm.**

And the test step:

For every piece X with N dimensional pattern $\mathrm{x} \in \mathcal{T}$

If $P(\mathrm{x}).B(\mathrm{x}).N(\mathrm{x}).D(\mathrm{x}) = 1$ **the piece is a flaw one.**

Then Use the classification model $\mathcal{M}$ **and predict the class of X**

Endif

End for

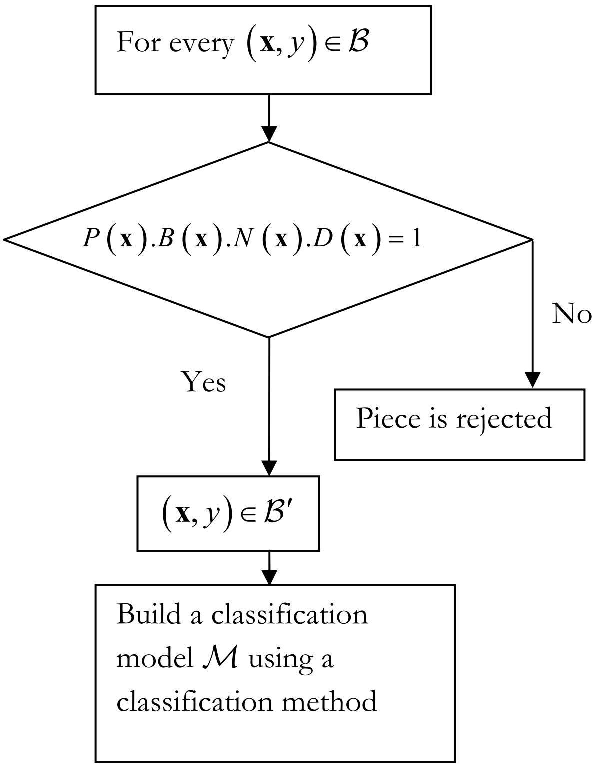

Figure 7: Learning tree based on boolean parameters

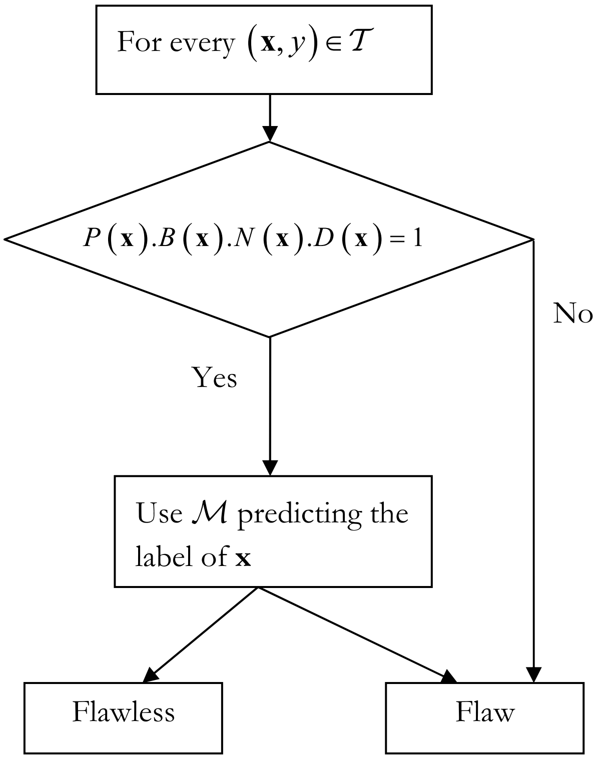

Figure 8: Decision tree using a model $\mathcal{M}$.

We depicted in Figure 7 and Figure 8 the learning tree and the decision tree. Due to the nature of the piece, this method is easy to use. Strong constraints on the part led us to build a "coarse to fine" approach which allows reaching the manufacturers specifications.

Results are summarized in Table 6. Since the priority of the detection was given by the manufacturer to the FAR, we can here conclude that the gain is important in the SVM case: FAR is 30% better than in the previous paragraph and the relative gain on λ_{opt} is 7%. The result of boosting is unchanged, due to the nature of the original algorithm which is a tree.

Table 6: Decision tree using SVM, boosting and Hyperrectangles classifiers.

	λ_{opt}	FAR	NDR
Decision Tree using SVM	0.91	0.50	18.37
Decision Tree using Adaboost	1.81	1.81	34.15
Decision Tree using Hyperrectangles	1.29	0.57	25.71

The main advantage of the combination is here to improve the performance regarding the FAR. However, this approach is less general than the standard SVM training procedure and cannot be applied systematically in quality control. Moreover, the use of the tree reduces the number of samples available for the training step of the supervised classification. This can be critical in some applications for which it is often difficult to obtain such samples.

The low-quality results of the hyperrectangle based method is mainly due to the presence of non-classifed samples (samples outside of all hyperrectangles). This situation occurs when the number of samples is small regarding to the feature space dimension. This is also the case for the Adaboost procedure: the quality of the optimum threshold estimation at each iteration depends of the number of samples in the database.

In order to analyze classification errors, we give in Figures 9-a through 9-d examples of good detection (sound piece and faulty piece), false alarm and non detection. As one can see, pieces b and c look alike; however b is the image of a faulty piece whereas c is not. The high variability of the parts explains why the error rates are still high even after all presented optimizations. However these error rates match with preliminary manufacturers specifications.

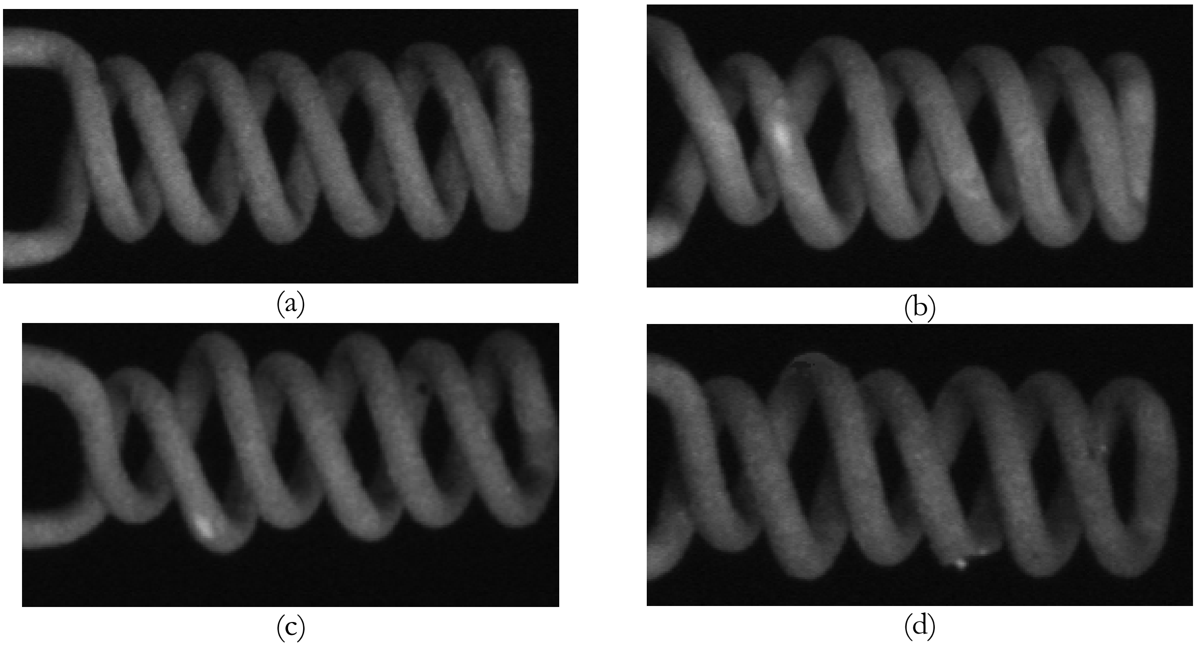

Figure 9: Different examples of classified images: flawless piece labelled as good (a), flaw piece labelled faulty (b), false alarm piece (c) and faulty piece labelled as good (non detection)(d).

5 CONCLUSION

We presented a full artificial vision system which allows high rate (10 p/s) anomaly detection on industrial objects. The whole process of part detection, low level-image segmentation, feature extraction, selection and comparison of results using three classification methods were exposed.

We showed through robust statistical methods that our classification methods, with or without any combination, allow to separate flaws from sound pieces with an accuracy which follows the industrial constraints. We also showed that the use of a combination of decision and SVM based decision rule improves the initial performance. The detection is performed in spite of the high variability of the part to be inspected, and the computation time satisfies the manufacturing constraints of ten inspections per second. Finally, 95% of the computation time is dedicated to the image segmentation and features extraction, and only 5% for the decision tree combined with the SVM classification.

One on the main contribution of the study which can be generalized to other quality control system by artificial vision is that the combination of a single rule base decision tree with a complex SVM method allows to improve the classification performances and the readability of the results, which is important for the end user.

In the near future, we plan to improve the performance of the system by combining two views of the part and detecting more anomalies.

6 BIBLIOGRAPHY

[1] C.M.Bishop, Neural network for pattern recognition, Oxford university press, 1995.

[2] M.M.Ratnam and Chee Peng Lim, "Egg classification using shadow moiré and neural network" in Proc. of QCAV 2001 - Vol. 2, pp. 355-358, 2001.

[3] R.O.Duda, P.E.Hart and D.G.Stork, Pattern classification, John Wiley & Son, INC, 2001.

[4] C.Firmin, D.Hamad, J.-G.Postaire and R.D.Zhang, "Gaussian neural network for glass bottle production : a learning procedure" in International journal on Neural systems - Vol. 8, pp. 77 - 86, 1996.

[5] V.N.Vapnik, The nature of statistical learning theory, Springer-Verlag, New-York, 1995.

[6] B.Schölkopf, A.Smola, K.R.Müller, C.J.C.Burges and V.N.Vapnik, "Support vector methods in learning and feature extraction" in Australian Journal of Intelligent Information Processing Systems - Vol. 1(:1), pp. 3 - 9, 1998.

[7] Jonsson K., Smola A., Mullër K.R. and Burges C., "Support Vector Machines for Face Authentication" in Proc. of British Machine Vision Conference, pp. 543-553, 1999.

[8] M.A Hearst, "Support vector machines" in IEEE Intelligent systems , pp. 18 - 32, 2003.

[9] Y.Freund, *Yova Freund's AdaBoost applet*, http://www.cs.huji.ac.il/~yoavf/adaboost/index.html - 1999.

[10] J.Miteran, P.Gorria and M.Robert, "Classification géométrique par polytopes de contraintes. Performances et intégration" in Traitement du Signal - Vol. 11(:5), pp. 393 - 408, 1994.

[11] D.Wettschereck and T.Dietterich, "An Experimental Comparison of the Nearest-Neighbor and Nearest-Hyperrectangle Algorithms" in Machine Learning - Vol. 19(:6), pp. 5 - 27, 1995.

[12] P.Geveaux, J.Miteran, S.Kohler and F.Truchetet., "A lighting characterization by a reliable method. Application to defect detection by artificial vision in industrial field" in Proc. of IECON 98, pp. 1242-1245, 1-9-1998.

[13] W.H.Tsai, "Moment preserving thresolding : a new approach" in Computer vision, graphics and image processing - Vol. 29, pp. 377 - 393, 1-12-1985.

[14] J.Miteran, J-P.Zimmer, F.Yang and M.Paindavoine, "Access control : adaptation and real time implantation of a face recognition method" in Optical Engineering - Vol. 40(:4), pp. 586 - 593, 1-4-2001.

[15] P.Geveaux, S.Kohler, J.Miteran, H.Jender and F.Truchetet., "A method of searching segmentation parameters applied to quality control of industrial pieces" in Proc. of QCAV 98, pp. 197-201, 1-11-1998.

[16] J.Miteran, R.Bailly and P.Gorria, "Classification board for real time image segmentation" in Proc. of ICASSP 97, pp. 4069-4072, 1-4-1997.

[17] J.Miteran, S.Bouillant, M.Paindavoine and E.Bourennane, "SVM approximation for real-time FPGA based image segmentation" in Proc. of Journée AAA - Vol. 1, pp. 20-23, 2003.

[18] Y.Freund and R.E.Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting" in Journal of Computer and System Sciences - Vol. 55(:1), pp. 119 - 139, 1997.

[19] R.E.Schapire, "A brief introduction to boosting" in Proc. of 16th International Joint Conference on Artificial Intelligence, pp. 1401-1405, 1999.

[20] V.N.Vapnik, Statistical learning theory, John Wiley & sons, INC, 1998.

[21] G.J.McLachlan, Discriminant Analysis and Statistical Pattern Recognition, John Wiley & Sons, New York, 1992.

[22] R.A.Fisher, "The use of multiple measurements in taxonimic problem" in Annales of Eugenics - Vol. 7, Part II, pp. 179 - 188, 1936.

[23] J.Kittler, "Feature set algorithms" in Pattern recognition and signal processing , pp. 41 - 60, 1978.

[24] P.Pudil, J.Novovicova and J.Kittler, "floating Search methods in feature selection" in Pattern recognition letters - Vol. 15, pp. 1119 - 1125, 1994.

[25] M.Dash and H.Liu, "Feature selection for classification" in Intelligent data analysis - Vol. 1(:3), pp. 131 - 156, 1997.

[26] J.Kittler, M.Hatef and R.P.Duin, "On Combining classifiers" in IEEE Transactions on pattern analysis and machine intelligence - Vol. 20(:3), pp. 226 - 239, 2003.

[27] J.R.Quinlan and R.L.Rivest, "Inferring decision trees using the Minimum Description Length Principle" in Information and Computation - Vol. 80(:3), pp. 227 - 248, 1-3-1989.

Figure captions

Figure 1: Part to be controlled

Figure 2: Anomalies to be detected

Figure 3: Top view of the acquisition system.

Figure 4: Image segmentation.

Figure 5: Arms labeling.

Figure 6: Dimensional features (characteristics of the white lines).

Figure 7: Learning tree based on boolean parameters

Figure 8: Decision tree using a model $\mathcal{M}$.

Figure 9: Different examples of classified images: flawless piece labelled as good (a), flaw piece labelled faulty (b), false alarm piece (c) and faulty piece labelled as good (non detection)(d).