Multi-class Classification in Image Analysis Via Error-Correcting Output Codes

Sergio Escalera, David M. J. Tax, Oriol Pujol, Petia Radeva, and Robert P. W. Duin

Abstract A common way to model multi-class classification problems is by means of Error-Correcting Output Codes (ECOC). Given a multi-class problem, the ECOC technique designs a codeword for each class, where each position of the code identifies the membership of the class for a given binary problem. A classification decision is obtained by assigning the label of the class with the closest code. In this paper, we overview the state-of-the-art on ECOC designs and test them in real applications.

Sergio Escalera
Dept. Matemàtica Aplicada i Anàlisi, Universitat de Barcelona, Gran Via de les Corts Catalanes 585, 08007, Barcelona, Spain
Computer Vision Center, Campus UAB, Edifici O, 08193, Bellaterra, Spain
e-mail: sergio@maia.ub.es

David M. J. Tax
Information and Communication Theory (ICT) Group, Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, PO Box 5031, 2600 GA, Delft, The Netherlands
e-mail: d.m.j.tax@gmail.com

Oriol Pujol
Dept. Matemàtica Aplicada i Anàlisi, Universitat de Barcelona, Gran Via de les Corts Catalanes 585, 08007, Barcelona, Spain
Computer Vision Center, Campus UAB, Edifici O, 08193, Bellaterra, Spain
e-mail: oriol@maia.ub.es

Petia Radeva
Dept. Matemàtica Aplicada i Anàlisi, Universitat de Barcelona, Gran Via de les Corts Catalanes 585, 08007, Barcelona, Spain
Computer Vision Center, Campus UAB, Edifici O, 08193, Bellaterra, Spain
e-mail: petia.ivanova@ub.edu

Robert P. W. Duin
Information and Communication Theory (ICT) Group, Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, PO Box 5031, 2600 GA, Delft, The Netherlands
e-mail: r.duin@ieee.org

Results on different multi-class data sets show the benefits of using the ensemble of classifiers when categorizing objects in images.

1 Introduction

In the literature, one can find several powerful binary classifiers. However, when one needs to deal with multi-class classification problems, many learning techniques fail to manage this information. Instead, it is common to construct the classifiers to distinguish between just two classes, and to combine them in some way. In this sense, Error Correcting Output Codes (ECOC) were born as a general framework to combine binary problems to address the multi-class problem. The strategy was introduced by Dietterich and Bakiri [17] in 1995. Based on the error correcting principles [17] and because of its ability to correct the bias and variance errors of the base classifiers [16], ECOC has been successfully applied to a wide range of image analysis applications, such as face recognition [30], face verification [15], text recognition [11] or manuscript digit classification [32].

The ECOC technique can be broken down into two distinct stages: encoding and decoding. Given a set of classes, the coding stage designs a codeword[1] for each class based on different binary problems. The decoding stage makes a classification decision for a given test sample based on the value of the output code.

At the coding step, given a set of N classes to be learnt, n different bi-partitions (groups of classes) are formed, and n binary problems (dichotomizers) are trained. As a result, a codeword of length n is obtained for each class, where each bit of the code corresponds to the response of a given dichotomizer (coded by +1, -1, according to their class set membership). Arranging the codewords as rows of a matrix, we define a *coding matrix* M, where $M \in \{-1,1\}^{N\times n}$ in the binary case.

It was when Allwein et al. [1] introduced a third symbol (the zero symbol) in the coding process when the coding step received special attention. This symbol increases the number of partitions of classes to be considered in a ternary ECOC framework by allowing some classes to be ignored. Then, the ternary coding matrix becomes $M \in \{-1,0,1\}^{N\times n}$. In this case, the symbol zero means that a particular class is not considered by a certain binary classifier.

The decoding step was originally based on error-correcting principles under the assumption that the learning task can be modeled as a communication problem, in which class information is transmitted over a channel [17]. During the decoding process, applying the n binary classifiers, a code is obtained for each data point in the test set. This code is compared to the base codewords of each class defined in the matrix M, and the data point is assigned to the class with the *closest* codeword.

In this paper, we overview the state-of-the-art ECOC designs. We describe the different ECOC strategies available from both the binary and the ternary ECOC frameworks. We analyze the complexity in terms of the number of classifiers and

[1] The codeword is a sequence of bits of a code representing each class, where each bit identifies the membership of the class for a given binary classifier.

test the designs on different multi-class data sets, showing the benefits of using the ensemble of classifiers when categorizing objects in images.

The paper is organized as follows: Section 2 overview the state-of-the-art ECOC designs. Section 3 shows the experimental results on two challenging image categorization problems. Finally, section 4 concludes the paper.

2 ECOC Designs

The most well-known binary coding strategies are the one-versus-all strategy [19], where each class is discriminated against the rest of classes, and the dense random strategy [1], where a random matrix M is generated maximizing the rows and columns separability in terms of the Hamming distance [17]. In Fig. 1(a), the one-versus-all ECOC design for a 4-class problem is shown. The white regions of the coding matrix M correspond to the positions coded by 1, and the black regions to -1. Thus, the codeword for class C_1 is $\{1,-1,-1,-1\}$. Each column i of the coding matrix codifies a binary problem learned by its corresponding dichotomizer h_i. For instance, dichotomizer h_1 learns C_1 against classes C_2, C_3 and C_4, dichotomizer h_2 learns C_2 against classes C_1, C_3 and C_4, etc. An example of a dense random matrix for a 4-class problem is shown in Fig. 1(c).

The classical coding designs in the ternary ECOC framework are the one-versus-one [13] and the random sparse coding [1]. Fig. 1(b) shows the one-versus-one ECOC configuration for a 4-class problem. In this case, the grey positions correspond to the zero symbol. A possible sparse random matrix for a 4-class problem is shown in Fig. 1(d). Note that the previous coding designs are predefined. Thus, the training data is not considered until the coding matrix M is constructed. Then, each dichotomizer uses the coded positions of M to train the different binary problems.

The most frequently applied decoding strategies are the Hamming (HD) [19] and the Euclidean (ED) decoding distances [13]. With the introduction of the zero symbol, Allwein et al. [1] showed the advantage of using a loss based function of the margin of the output of the base classifier. Recently, the authors of [9] proposed a Loss-Weighted strategy to decode, where a set of probabilities based on the performances of the base classifiers are used to weight the final classification decision. In Fig. 1, each ECOC codification is used to classify an input object X. The input data sample X is tested with each dichotomizer h_i, obtaining an output X_i. The final code $\{X_1,...,X_n\}$ of the test input X is used by a given decoding strategy to obtain the final classification decision. Note that in both, the binary and the ternary ECOC framework, the value of each position X_j of the test codeword can not take the value zero since the output of each dichotomizer $h_j \in \{-1,+1\}$.

Recently, new improvements in the ternary ECOC coding demonstrate the suitability of the ECOC methodology to deal with multi-class classification problems [27][26][10][29][5]. These recent designs use the knowledge of the problem-domain to learn relevant binary problems from ternary codes. The basic idea of these methods is to use the training data to guide the training process, and thus, to construct

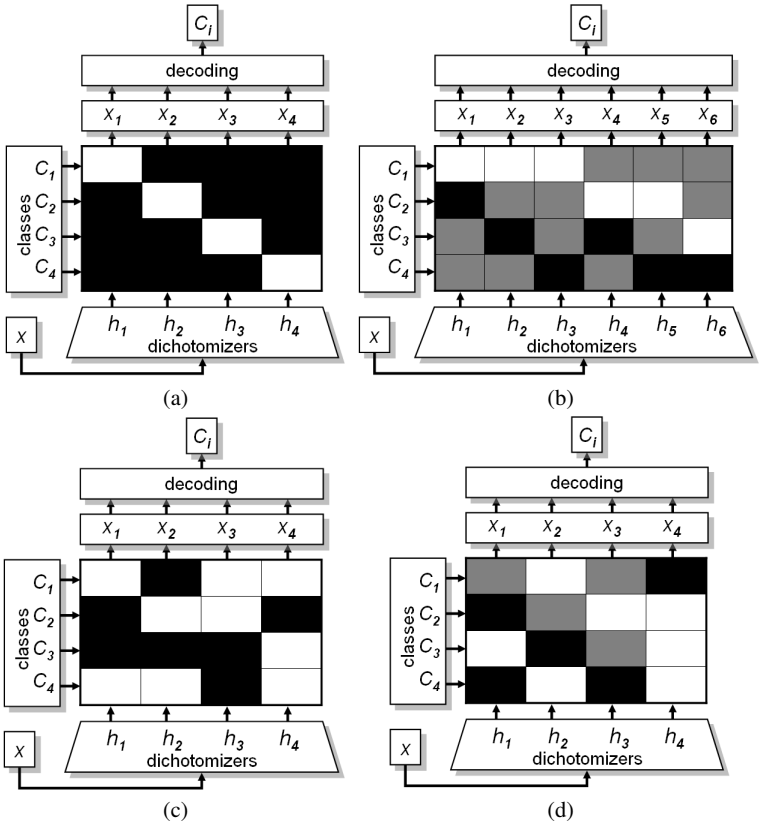


Fig. 1 One-versus-all (a), one-versus-one (b), dense random (c), and (d) sparse random ECOC designs.

the coding matrix M focusing on the binary problems that better fits the decision boundaries of a given data set. The definition of new coding designs also motivated the design of novel decoding methodologies [9]. Next, we describe some of these recent designs: ECOC-ONE, DECOC, Sub-class ECOC coding, and Loss-Weighted decoding.

2.1 ECOC-ONE coding

ECOC-Optimal Node Embedding defines a general procedure capable of extending any coding matrix by adding dichotomizers based on a discriminability criterion. In the case of a multiclass recognition problem, the procedure starts with a given ECOC coding matrix. Then, this ECOC matrix is increased in an iterative way, adding dichotomizers that correspond to different sub-partitions of classes. These

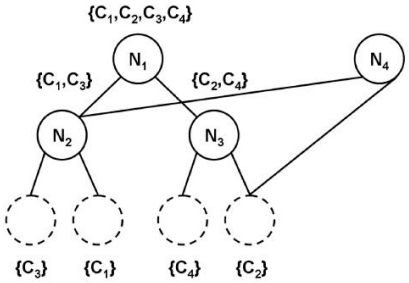
partitions are found using greedy optimization based on the confusion matrices so that the ECOC accuracy improves on both training and validation sets. The training set guides the convergence process, and the validation set is used to avoid overfitting and to select a configuration of the learning procedure that maximizes the generalization performance [17]. Since not all problems require the same dichotomizers structure -in form of sub-partitions-, the optimal node embedding approach generates an optimal ECOC-ONE matrix dependent on the hypothesis performance in a specific problem domain.

Given N_c classes and a coding matrix M:
while $error > \varepsilon$ or $error_t < error_{t-1}, t \in [1,T]$:
Compute the optimal node t:
1) Test accuracy on the training and validation sets S_t and S_v.
2) Select the pair of classes with the highest error analyzing the confusion matrices from S_t and S_v.
3) Find the partition that minimizes the error rate in S_t and S_v.
4) Compute the weight for the dichotomy of the partition based on its classification score.
Update the matrix M.

Table 1 ECOC-ONE general algorithm

Table 1 shows the summarized steps for the ECOC-ONE approach. Note that, the process described is iterated while the error on the training subsets is greater than ε or the number of iterations $i \leq T$. An example of an ECOC-ONE strategy applied to a four-class classification example can be found in Fig. 2. The initial optimal tree corresponds to the dichotomizers of optimal sub-partition of the classes. This tree has been generated using accuracy as a sub-partition splitting criterion. After testing the performance of the ensemble tree (composed by the columns $\{h_1, h_2, h_3\}$ of the ECOC matrix M of Fig. 2(b)), let assume that classes $\{C_2, C_3\}$ get maximal error in the confusion matrices v_t and v_v. We search for the sub-partition of classes using the training and validation subsets so that the error between $\{C_2, C_3\}$ and all previous misclassified samples is minimized. Suppose now that this sub-partition is $\{C_1, C_3\}$ versus $\{C_2\}$. As a result, a new node N_4 corresponding to dichotomy h_4 is created. We can observe in Fig. 2 that N_4 uses a class partition that is present in the tree. In this sense, this new node connects two different nodes of the tree. Note that using the previously included dichotomizers, the partition $\{C_1, C_3\}$ is solved by N_2. In this way, the Hamming distance between C_2 and C_3 is increased by adding

the new dichotomy to the whole structure. At the same time, the distance among the rest of the classes is usually maintained or slightly modified.



(a)

	h1	h2	h3	h4
C1	1	-1	0	1
C2	-1	0	-1	-1
C3	1	1	0	1
C4	-1	0	1	0

(b)

Fig. 2 (a) Optimal tree and first optimal node embedded, (b) ECOC-ONE code matrix M for four dichotomizers.

One of the desirable properties of the ECOC matrix is to have maximal distance between rows. The ECOC-ONE procedure focuses on the relevant difficult partitions, increasing the distance between "close" classes. This fact improves the robustness of the method since difficult classes are likely to have a greater number of dichotomizers centered on them. In this sense, it creates different geometrical arrangements of decision boundaries, and leads the dichotomizers to make different bias errors.

2.2 DECOC and Sub-class ECOC coding

One of the main reasons why the recent problem-dependent designs [27][26][10] attains a good performance is because of the high number of possible sub-groups of classes that is possible in the ternary ECOC framework. On the other hand, using the training data in the process of the ECOC design allows to obtain compact codewords with high classification performance. However, the final accuracy is still based on the ability of the base classifier to learn each individual problem. Difficult problems, those which the base classifier is not able to find a solution for, require the use of complex classifiers, such as Support Vector Machines with Radial Basis Function kernel [23], and expensive parameter optimizations. Look at the example of Fig. 3(a). A linear classifier is used to split two classes. In this case, the base classifier is not able to find a convex solution. On the other hand, in Fig. 3(b), one of the previous classes has been split into two sub-sets, that we call *sub-classes*. Then, the original problem is solved using two linear classifiers, and the two new sub-classes have the same original class label. Some studies in the literature tried to form subclasses using the labels information, which is called Supervised Clustering [31][7]. In these types of systems, clusters are usually formed without taking into account

the behavior of the base classifier that learns the data. In a recent work [33], the authors use the class labels to form the sub-classes that improve the performance of particular Discriminant Analysis algorithms.

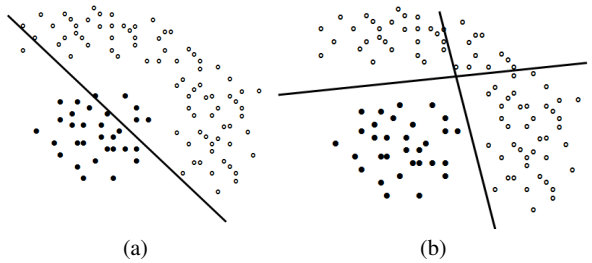


Fig. 3 (a) Decision boundary of a linear classifier of a 2-class problem. (b) Decision boundaries of a linear classifier splitting the problem of (a) into two more simple tasks.

From an initial set of classes C of a given multi-class problem, the objective of the Sub-class ECOC strategy is to define a new set of classes C', where $|C'| > |C|$, so that the new set of binary problems is easier to learn for a given base classifier. For this purpose, the approach uses a guided procedure that, in a problem-dependent way, groups classes and splits them into sub-sets if necessary.

Recently, the authors of [27] proposed a ternary problem-dependent design of ECOC, called DECOC, where given N classes, a high classification performance is achieved with only $N-1$ binary problems. The method is based on the embedding of discriminant tree structures derived from the problem domain. The binary trees are built by looking for the partition that maximizes the mutual information (*MI*) between the data and their respective class labels. Look at the 3-class problem shown on the top of Fig. 4(a). The standard DECOC algorithm considers the whole set of classes to split it into two sub-sets of classes $\wp^+$ and $\wp^-$ maximizing the *MI* criterion on a sequential forward floating search procedure (*SFFS*). In the example, the first sub-sets found correspond to $\wp^+ = \{C_1, C_2\}$ and $\wp^- = \{C_3\}$. Then, a base classifier is used to train its corresponding dichotomizer h_1. This classifier is shown in the node h_1 of the tree structure shown in Fig. 4(d). The procedure is repeated until all classes are split into separate sub-sets $\wp$. In the example, the second classifier is trained to split the sub-sets of classes $\wp^+ = C_1$ from $\wp^- = C_2$ because the classes C_1 and C_2 were still contained in a single sub-set after the first step. This second classifier is codified by the node h_2 of Fig. 4(d). When the tree is constructed, the coding matrix M is obtained by codifying each internal node of the tree as a column of the coding matrix (see Fig. 4(c)).

In the case of Sub-class ECOC, sequential forward floating search (*SFFS*) is also applied to look for the sub-sets $\wp^+$ and $\wp^-$ that maximizes the mutual information between the data and their respective class labels [27].

Given a N-class problem, the whole set of classes is used to initialize the set L containing the sets of labels for the classes to be learned. At the beginning of each

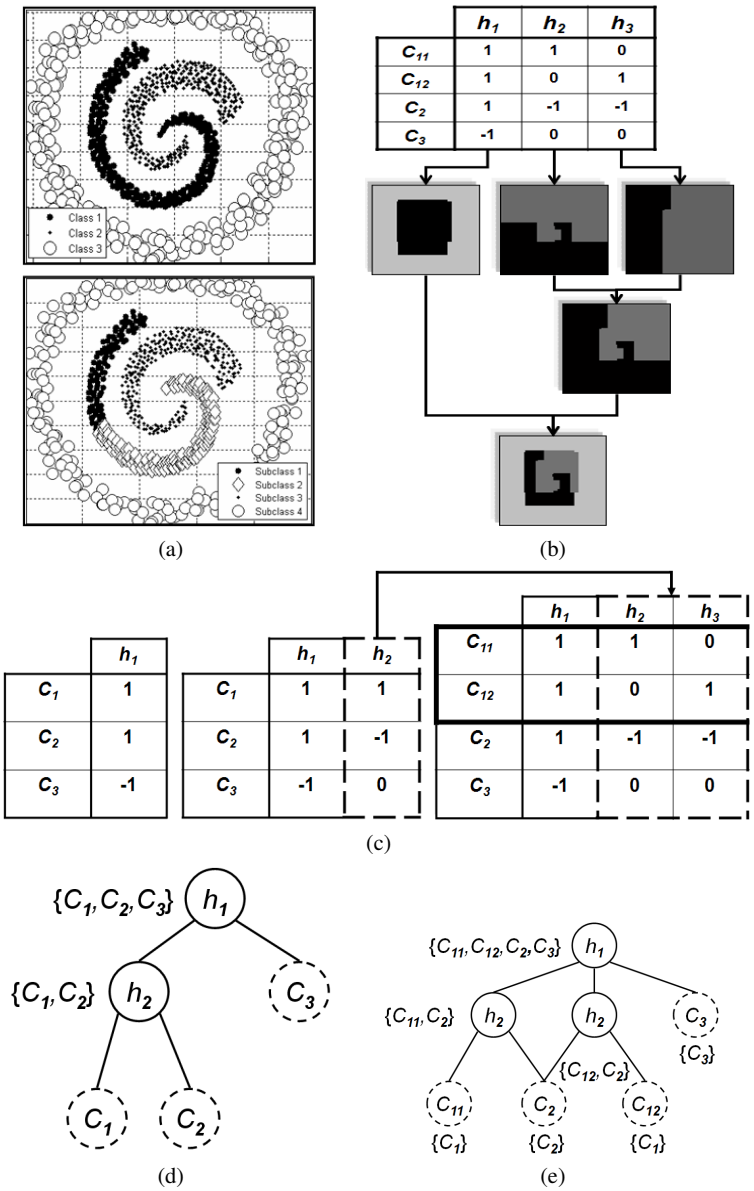


Fig. 4 (a) Top: Original 3-class problem. Bottom: 4 sub-classes found. (b) Sub-class ECOC encoding using the four sub-classes using Discrete Adaboost with 40 runs of Decision Stumps. (c) Learning evolution of the sub-class matrix M. (d) Original tree structure without applying sub-class. (e) New tree-based configuration using sub-classes.

iteration k of the algorithm, the first element of L is assigned to S_k in the first step of the algorithm. Next, $SFFS$ [25] is used to find the optimal binary partition BP of S_k

that maximizes the mutual information I between the data and their respective class labels.

To illustrate the procedure, let us return to the example of the top of Fig. 4(a). On the first iteration of the sub-class ECOC algorithm, $SFFS$ finds the sub-set $\wp^+ = \{C_1, C_2\}$ against $\wp^- = \{C_3\}$. The encoding of this problem is shown in the first matrix of Fig. 4(c). The positions of the column corresponding to the classes of the first partition are coded by +1 and the classes corresponding to the second partition to -1, respectively. In the Sub-class procedure, the base classifier is used to test if the performance obtained by the trained dichotomizers is sufficient. Observe the decision boundaries of the picture next to the first column of the matrix in Fig. 4(b). One can see that the base classifier finds a good solution for this first problem.

Then, the second classifier is trained to split $\wp^+ = C_1$ against $\wp^- = C_2$, and its performance is computed. To separate the current sub-sets is not a trivial problem, and the classification performance is poor. Therefore, the procedure tries to split the data $J_{\wp^+}$ and $J_{\wp^-}$ from the current sub-sets $\wp^+$ and $\wp^-$ into more simple sub-sets. Then, the splitting criteria SC takes as input a data set $J_{\wp^+}$ or $J_{\wp^-}$ from a sub-set $\wp^+$ or $\wp^-$, and splits it into two sub-sets $J^+_{\wp^+}$ and $J^-_{\wp^+}$ or $J^+_{\wp^-}$ and $J^-_{\wp^-}$.

When two data sub-sets $\{J^+_{\wp^+}, J^-_{\wp^+}\}$ and $\{J^+_{\wp^-}, J^-_{\wp^-}\}$ are obtained, only one of both split sub-sets is used. The selected sub-sets are those that have the highest distance between the means of each cluster. Suppose that the distance between $J^+_{\wp^-}$ and $J^-_{\wp^-}$ is larger than between $J^+_{\wp^+}$ and $J^-_{\wp^+}$. Then, only $J_{\wp^+}$, $J^+_{\wp^-}$, and $J^-_{\wp^-}$ are used. If the new sub-sets improve the classification performance, new sub-classes are formed, and the process is repeated.

In the example of Fig. 4, applying the splitting criteria SC over the two sub-sets, two clusters are found for $\wp^+ = C_1$ and for $\wp^- = C_2$. Then, the original encoding of the problem C_1 vs C_2 (corresponding to the second column of the matrix in the center of Fig. 4(c)) is split into two columns marked with the dashed lines in the matrix on the right. In this way, the original C_1 vs C_2 problem is transformed to two more simple problems $\{C_{11}\}$ against $\{C_2\}$ and $\{C_{12}\}$ against $\{C_2\}$. Here the first subindex of the class corresponds to the original class, and the second subindex to the number of sub-class. It implies that the class C_1 is split into two sub-classes (look at the bottom of Fig. 4(a)), and the original 3-class problem $C = \{C_1, C_2, C_3\}$ becomes the 4-sub-class problem $C' = \{C_{11}, C_{12}, C_2, C_3\}$. As the class C_1 has been decomposed by the splitting of the second problem, we need to save the information of the current sub-sets and the previous sub-sets affected by the new splitting. For this purpose, we use the object labels to define the set of sub-classes of the current partition $\wp_c$. If new sub-classes are created, the set of sub-classes C' and the data for sub-classes J' have to be updated. Note that when a class or a sub-class previously considered for a given binary problem is split in a future iteration of the procedure, the labels from the previous sub-sets $\{\wp^+, \wp^-\}$ need to be updated with the new information. Finally, the set of labels for the binary problems $\wp'$ is updated with the labels of the current sub-set $\wp' = \wp' \cup \wp_c$. In the example of Fig. 4, the dichotomizer h_1 considers the sub-sets $\wp^+_1 = \{C_1, C_2\}$ and $\wp^-_1 = \{C_3\}$. Then, those positions

containing class C_1 are replaced with C_{11} and C_{12}. The process is repeated until the desired performance is achieved or the stopping conditions are full-filled.

The conditions that guide the learning and splitting process are defined by the set of parameters $\theta = \{\theta_{size}, \theta_{perf}, \theta_{impr}\}$, where θ_{size} corresponds to the minimum size of a sub-set to be clustered, θ_{perf} contains the minimum error desired for each binary problem, and θ_{impr} looks for the improvement of the split sub-sets regarding the previous ones. In the example of Fig. 4, the three dichotomizers h_1, h_2, and h_3 find a solution for the problem (look the trained boundaries shown in Fig. 4(b)), obtaining a classification error under θ_{perf}, so, the process stops. Now, the original tree encoding of the DECOC design shown in Fig. 4(d) can be represented by the tree structure of Fig. 4(e), where the original class associated to each sub-class is shown in the leaves. The algorithms summarizing the subclass approach and the splitting methodology are shown in tables 2 and 3, respectively.

Table 2 Problem-dependent Sub-class ECOC algorithm.

Inputs: $J, C, \theta = \{\theta_{size}, \theta_{perf}, \theta_{impr}\}$ //Thresholds for the number of samples, performance, and improvement between iterations

Outputs: $C', J', \wp', M$

[Initialization:]

Create the trivial partition $\{\wp_0^+, \wp_0^-\}$ of the set of classes $\{C_i\}$: $\{\wp_0^+, \wp_0^-\} = \{\{\emptyset\}, \{C_1, C_2, ..., C_N\}\}$

$L_0 = \{\wp_0^-\}; J' = J; C' = C; \wp' = \emptyset; M = \emptyset; k = 1$

Step 1 S_k is the first element of L_{k-1}

$L'_k = L_{k-1} \backslash \{S_k\}$

Step 2 Find the optimal binary partition $BP(S_k)$:

$\{\wp_k^+, \wp_k^-\} = argmax_{BP(S_k)}(I(\mathbf{x}, d(BP(S_k))))$

where I is the mutual information criterion, $\mathbf{x}$ is the random variable associated to the features and d is the discrete random variable of the dichotomy labels[a], defined in the following terms,

$$d = d(\mathbf{x}, BP(S_k)) = \begin{cases} 1 & \text{if } \mathbf{x} \in C_i | C_i \in \wp_k^+ \\ -1 & \text{if } \mathbf{x} \in C_i | C_i \in \wp_k^- \end{cases}$$

Step 3 // Look for sub-classes

$\{C', J', \wp'\} = SPLIT(J_{p_k^+}, J_{p_k^-}, C', J', J, \wp', \theta)$[b]

Step 4 $L_k = \{L'_k \cup \wp_k^i\}$ if $|\wp_k^i| > 1 \; \forall i \in \{+, -\}$

Step 5 If $|L_k| \neq 0$

$k = k + 1$ **go to Step 1**

Step 6 Codify the coding matrix M using each partition $\{\wp_i^+, \wp_i^-\}$ of $\wp', i \in [1, .., |\wp'|]$ and each class $C_r \in \wp_i = \{\wp_i^+ \cup \wp_i^-\}$ as follows:

$$M(C_r, i) = \begin{cases} 0 & \text{if } C_r \notin \wp_i \\ +1 & \text{if } C_r \in \wp_i^+ \\ -1 & \text{if } C_r \in \wp_i^- \end{cases} \quad (1)$$

[a] Use *SFFS* of [25] as the maximization procedure and *MI* of [27] to estimate I

[b] Using the splitting algorithm of table 3.

Summarizing, when a set of objects belonging to different classes is split, object labels are not taken into account. It can be seen as a clustering in the sense that the sub-sets are split into more simple ones while the splitting constraints are satisfied.

Table 3 Sub-class *SPLIT* algorithm.

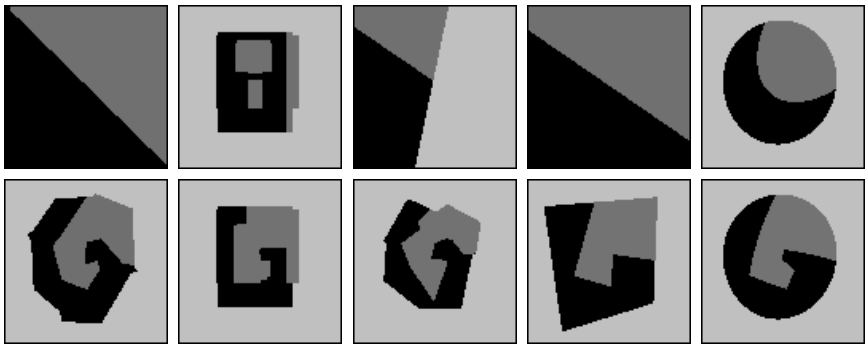
Inputs: $J_{\wp^1}, J_{\wp^2}, C', J', J, \wp', \theta$ // C' is the final set of classes, J' the data for the final set of classes, and $\wp'$ is the labels for all the partitions of classes of the final set.
Outputs: $C', J', \wp'$
Step 1 Split problems:
$\{J^+_{\wp^+}, J^-_{\wp^+}\} = SC(J_{\wp^+})^a$
$\{J^+_{\wp^-}, J^-_{\wp^-}\} = SC(J_{\wp^-})$
Step 2 Select sub-classes:
if $|\overline{J^+_{\wp^+}}, \overline{J^-_{\wp^+}}| > |\overline{J^+_{\wp^-}}, \overline{J^-_{\wp^-}}|$ // find the largest distance between the means of each sub-set.
  $\{J^+_+, J^-_+\} = \{J^+_{\wp^+}, J_{\wp^-}\}$; $\{J^+_-, J^-_-\} = \{J^-_{\wp^+}, J_{\wp^-}\}$
else
  $\{J^+_+, J^-_+\} = \{J^+_{\wp^-}, J_{\wp^+}\}$; $\{J^+_-, J^-_-\} = \{J^-_{\wp^-}, J_{\wp^+}\}$
end
Step 3 Test parameters to continue splitting:
if $TEST\_PARAMETERS(J_{\wp^1}, J_{\wp^2}, J^1_1, J^2_1, J^1_2, J^2_2, \theta)$// call the function with the new sub-sets
  $\{C', J', \wp'\} = SPLIT(J^1_1, J^2_1, C', J', J, \wp', \theta)$
  $\{C', J', \wp'\} = SPLIT(J^1_2, J^2_2, C', J', J, \wp', \theta)$
end
Step 4 Save the current partition:
Update the data for the new sub-classes and previous sub-classes if intersections exists J'.
Update the final number of sub-classes C'.
Create $\wp_c = \{\wp_{c^1}, \wp_{c^2}\}$ the set of labels of the current partition.
Update the labels of the previous partitions $\wp$.
Update the set of partitions labels with the new partition $\wp' = \wp' \cup \wp_c$.

[a] *SC* corresponds to the splitting method of the input data into two main clusters.

It is important to note that when one uses different base classifiers, the sub-class splitting is probably applied to different classes or sub-classes, and therefore, the final number of sub-classes and binary problems differs.

Finally, to decode the new sub-class problem-dependent design of ECOC, the authors use the recently proposed Loss-Weighted decoding design described in the next section.

To show the effect of the Sub-class ECOC strategy for different base classifiers, we used the previous toy problem of the top of Fig. 4(a). Five different base classifiers are applied: Fisher Linear Discriminant Analysis (*FLDA*), Discrete Adaboost, Nearest Mean Classifier, Linear *SVM*, and *SVM* with Radial Basis Function kernel. Using these base classifiers on the toy problem, the original DECOC strategy with the Loss-Weighted algorithm obtains the decision boundaries shown on the top row of Fig. 5. The new learned boundaries are shown on the bottom row of Fig. 5 for fixed parameters θ. Depending on the flexibility of the base classifier more sub-classes are required, and thus, more binary problems. Observe that all base classifiers are able to find a solution for the problem, although with different types of decision boundaries.



(a) (b) (c) (d) (e)

Fig. 5 Sub-class ECOC without sub-classes (top) and including sub-classes (bottom): for *FLDA* (a), Discrete Adaboost (b), *NMC* (c), Linear *SVM* (d), and *RBF SVM* (e).

2.3 Loss-Weighted decoding

The Loss-Weighted decoding is defined as a combination of normalized probabilities to adapt the decoding to both binary and ternary ECOC frameworks. The properties of the decoding strategy are encoded in a matrix that is used to weight the decoding process. Moreover, as not all the hypotheses have the same performance on learning the data samples, the accuracy of each binary problem is used to adjust the final classification decision.

A weight matrix M_W is defined by assigning to each position of the codeword codified by $\{-1,+1\}$ a weight of $\frac{1}{n-z}$, where z is the number of positions codified by zero. We assign to each position (i,j) of a performance matrix H a continuous value that corresponds to the performance of the dichotomizer h_j classifying the samples of class C_i as follows:

$$H(i,j) = \frac{1}{m_i}\sum_{k=1}^{m_i} \varphi(h^j(\rho_k^i),i,j), \quad \text{based on} \quad \varphi(x^j,i,j) = \begin{cases} 1, & \text{if} \quad x^j = y_i^j, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

where m_i are the number of samples of class C_i, ρ is a test sample, and x^i and y^i are the j-th position of a test and a class codeword, respectively. Note that eq.(2) makes H to have zero probability at those positions corresponding to unconsidered classes.

We normalize each row of the matrix H so that M_W can be considered as a discrete probability density function:

$$M_W(i,j) = \frac{H(i,j)}{\sum_{j=1}^{n} H(i,j)}, \quad \forall i \in [1,...,N], \quad \forall j \in [1,...,n] \tag{3}$$

In Fig. 6, a weight matrix M_W for a 3-multi-class problem of four hypotheses is estimated. Figure 6(a) shows the coding matrix M. The matrix H of Fig. 6(b)

represents the accuracy of the hypotheses classifying the instances of the training set. The normalization of H results in a weight matrix M_W shown in Fig. 6(c).

$$M=\begin{bmatrix}1 & 1 & -1 & 0\\ 1 & -1 & 0 & 0\\ 1 & 1 & 1 & -1\end{bmatrix} \quad H=\begin{bmatrix}0.955 & 0.955 & 1.000 & 0.000\\ 0.900 & 0.800 & 0.000 & 0.000\\ 1.000 & 0.905 & 0.805 & 0.805\end{bmatrix} \quad M_W=\begin{bmatrix}0.328 & 0.328 & 0.344 & 0.000\\ 0.529 & 0.471 & 0.000 & 0.000\\ 0.285 & 0.257 & 0.229 & 0.229\end{bmatrix}$$

(a) (b) (c)

Fig. 6 (a) Coding matrix M of four hypotheses for a 3-class problem. (b) Performance matrix H. (c) Weight matrix M_W.

Once we obtain the weight matrix M_W, we introduce the weight matrix in the Loss-based decoding. The decoding estimation is obtained by means of an ELB decoding model $L(\partial) = \mathbf{e}^{-\partial}$, where ∂ corresponds to $y_i^j \cdot f(\rho, j)$, weighted using M_W:

$$LW(\rho, i) = \sum_{j=1}^{n} M_W(i, j) L(y_i^j \cdot f(\rho, j)) \tag{4}$$

The summarized algorithm is shown in table 4.

Table 4 Loss-Weighted algorithm.

Loss-Weighted strategy: Given a coding matrix M,

1) Calculate the performance matrix H:

$$H(i, j) = \frac{1}{m_i} \sum_{k=1}^{m_i} \varphi(h^j(\rho_k^i), i, j) \quad \text{based on} \quad \varphi(x^j, i, j) = \begin{cases} 1, & \text{if } x^j = y_i^j, \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

2) Normalize H: $\sum_{j=1}^{n} M_W(i, j) = 1, \quad \forall i = 1, ..., N$:

$$M_W(i, j) = \frac{H(i, j)}{\sum_{j=1}^{n} H(i, j)}, \quad \forall i \in [1, ..., N], \quad \forall j \in [1, ..., n] \tag{6}$$

3) Given a test data sample ρ, decode based on:

$$LW(\rho, i) = \sum_{j=1}^{n} M_W(i, j) L(y_i^j \cdot f(\rho, j)) \tag{7}$$

3 Experimental results

In this section, we test the state-of-the-art ECOC configurations on two challenging image analysis applications: a real 9-class traffic sign classification problem from the Geomobil project of [4] and Intravascular Ultrasound Tissue Characterization.

3.1 Traffic sign categorization

For this experiment, we use the video sequences obtained from the Mobile Mapping System [4] to test a real traffic sign categorization problem. We choose the speed data set since the low resolution of the image, the non-controlled conditions, and the high similarity among classes make the image categorization a difficult task. In this system, the position and orientation of the different traffic signs are measured with fixed video cameras in a moving vehicle. The system has a stereo pair of calibrated cameras, which are synchronized with a GPS/INS system. The result of the acquisition step is a set of stereo-pairs of images with their position and orientation information. Fig. 7 shows several samples of the speed data set used for the experiments. The data set contains a total of 2500 samples divided into nine classes. Each sample is composed by 1200 pixel-based features after smoothing the image and applying a histogram equalization. From this original feature space, about 150 features are derived using a *PCA* that retained 90% of the total variance.

Fig. 7 Speed data set samples.

The performance and the estimated ranks using the different ECOC strategies for the different base classifiers are shown in table 5. These results are also illustrated in the graphics of Fig. 8. The random matrices were selected from a set of 20000 randomly generated matrices, with $P(1) = P(-1) = 0.5$ for the dense random matrix and $P(1) = P(-1) = P(0) = 1/3$ for the sparse random matrix. The number of binary problems was fixed to the number of classes. Therefore, a direct comparison to the one-versus-all and DECOC designs is possible. Each strategy uses the previously mentioned Linear Loss-weighted decoding to evaluate their performances at identical conditions. To evaluate the performance of the different experiments, we apply stratified ten-fold cross-validation and test for the confidence interval at 95% with a two-tailed t-test [8].

In this particular problem, the sub-class is only required for Discrete Adaboost and *NMC*, while the rest of base classifiers are able to find a solution for the training set without the need for sub-classes. Finally, though the results do not significantly differ between the strategies, the Sub-class ECOC approach attains a better position in the global rank of table 5.

Table 5 Rank positions of the classification strategies for the Speed data set.

	one-versus-one	one-versus-all	dense	sparse	DECOC	Sub-class ECOC
D. Adaboost	66.1(3.1)	56.6(3.1)	55.2(2.8)	52.3(3.6)	58.6(3.2)	60.8(3.1)
NMC	60.7(3.2)	50.65(3.7)	47.4(3.8)	45.1(3.8)	51.9(3.2)	62.8(3.1)
FLDA	74.7(2.8)	71.4(2.9)	74.9(2.6)	72.7(2.5)	72.6(2.8)	76.2(3.0)
Linear SVM	74.9(2.7)	72.3(2.1)	71.8(2.1)	68.2(2.9)	78.9(2.1)	78.9(1.9)
RBF SVM	45.0(0.9)	45.0(0.9)	45.0(0.9)	44.0(0.9)	45.0(0.9)	45.0(0.9)
Global rank	1.8	3.6	3.4	4.6	2.6	1.2

3.2 Intravascular Ultrasound Tissue Characterization

Cardiovascular diseases represented the first cause of sudden death in the occidental world [22]. Plaque rupture is one of the most frequent antecedent of coronary pathologies. Depending on the propensity to collapse, coronary plaque can be divided into stable and vulnerable plaque [3]. According to pathological studies, the main features of a stable plaque are characterized by the presence of a large lipid core with a thin fibrous cap. This last type of plaque can rupture generating thrombi followed by an intimal hyperplasia. Therefore, an accurate detection and quantification of plaque types represents an important subject in the diagnosis in order to study the nature and the plaque evolution to predict its final effect.

One of the most widely used diagnostic procedures consists of screening the coronary vessels employing Intravascular Ultrasound Imaging (IVUS). This technique yields a detailed cross-sectional image of the vessel allowing coronary arteries and their morphology to be extensively explored. This image modality has become one of the principal tools to detect coronary plaque. An IVUS study consists of introducing a catheter which shots a given number of ultrasound beams and collect their echoes to form an image. According with these echoes, three distinguishable plaques are considered in this type of images: calcified tissue (characterized by a very high echo-reflectivity and absorbtion of the ultrasound signal), fibrous plaque (medium echo-reflectivity and good transmission coefficient), and lipidic or soft plaque (characterized with very low reflectance of the ultrasound signal).

Despite the high importance of studying the whole coronary vessel, in clinical practice, this plaque characterization is performed manually in isolated images. Moreover, due to the variability among different observers, a precise manual characterization becomes very difficult to perform. Therefore, automatic analysis of IVUS

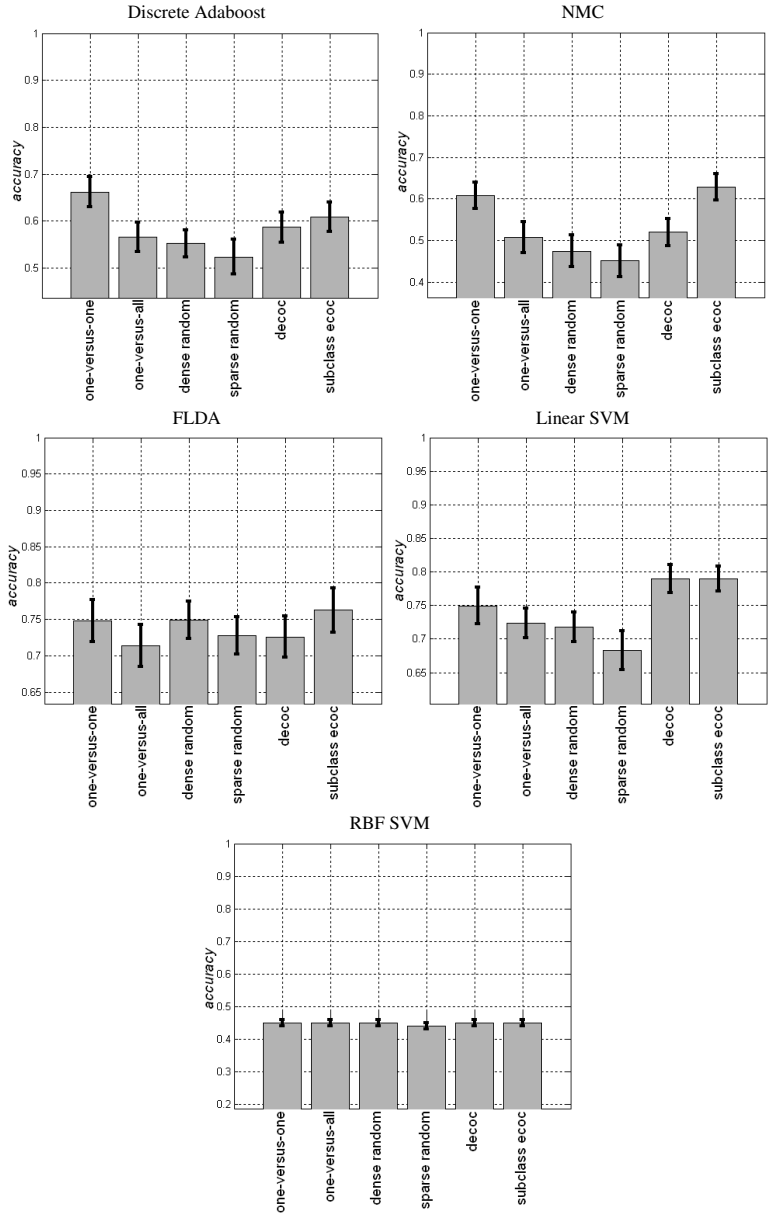


Fig. 8 Speed data set performances.

images represents a feasible way to predict and quantify the plaque composition, avoiding the subjectivity of manual region classification and diminishing the characterization time in large sequences of images. Given its clinical importance, auto-

matic plaque classification in IVUS images has been considered in several research studies. The process can be divided into two stages: plaque characterization step which consists of extracting characteristic features in order to describe each tissue, and a classification step where a learning technique is used to train a classifier.

In this section, we present an intravascular data set based on texture-based features, RF signals, combined features, and slope-based features to characterize the different types of tissues.

Feature Extraction

We consider three types of features, the first ones obtained from RF signals, the second ones based on texture-based features from reconstructed images, and finally, the slope-based features proposed in [18].

RF Features

In order to analyze ultrasound images, the RF signals are acquired from the IVUS equipment with a sampling rate of at least two times the transducer frequency, and filtered using a band-pass filter with 50% gain centered at the transducer frequency [14]. Then, an exponential Time Gain Compensation (TGC) is applied [14]. Once the RF signals have been acquired, filtered and exponentially compensated by the TGC, the power spectrum is obtained. Nair et al. in [18] show the modelling of the power spectrum using Autoregressive Models (ARM) as one of the most suitable and stable methods to analyze ultrasound signals [18]. It also represents an alternative to the Fourier Transform since the ARM have been proved to be more stable when small signal windows are considered.

The ARM are defined as a linear prediction equation where the output x at a certain point t for each A-line is equal to a linear combination of its p previous outputs weighted by a set of parameters a_p [24]:

$$x(t) = \sum_{k=1}^{p} a_p(k)x(t-k),$$

where p is the ARM degree and the coefficients a_p are calculated minimizing the error of the modelled spectrum with respect to the original using the Akaike's error prediction criterium [24].

A sliding window is formed by n samples and m contiguous A-lines with a displacement of $n/4$ samples and $m/3$ A-lines in order to obtain an average AR model of a region. Only one side of the obtained spectrum is used because of its symmetrical properties. This spectrum is composed of h sampled frequencies ranging from 0 to $f_s/2$ [24].

In addition to the spectrum, two global measures are computed: the energy of the A-line and the energy of the window spectrum. All these features are compiled into a unique vector of $h+2$ dimensions which is used as a feature vector in the classification process.

Texture Features Extraction

Given that different plaques can be discriminated as regions with different grey-level distributions, it is a natural decision to use texture descriptors. In the bibliography, one can find a wide set of texture descriptors and up to our knowledge there are no optimal texture descriptors for image analysis in the general case. Our strategy is

instead of trying to find out the optimal texture descriptor for our problem to gather several families of descriptors and apply multiple classifiers able to learn and extract the optimal features for the concrete problem.

Therefore, we employ three different texture descriptors: co-occurrence Matrix [20], local binary patterns [21] and Gabor filters [6, 2]. Additionally, taking into account that highly non-echogenic plaques produce significant shade in the radial direction of the vessel, we include in the feature set the presence of shading in the image as a complementary feature.

The co-occurrence matrix is defined as the estimation of the joint probability density function of gray level pairs in an image [20]. The sum of all element values is:

$$P(i,j,D,\theta) = P(I(l,m) = i \otimes I(l + Dcos(\theta), m + Dsin(\theta)) = j),$$

where $I(l,m)$ is the gray value at pixel (l,m), D is the distance among pixels and θ is the angle between neighbors. We have established the orientation θ to be $[0^o, 45^o, 90^o, 135^o]$ [28, 20]. After computing this matrix, Energy, Entropy, Inverse Difference Moment, Shade, Inertia and Promenance measures are extracted [20].

Local Binary Patterns (LBP) are used to detect uniform texture patterns in circular neighborhoods with any quantization of angular space and spatial resolution [21]. LBP are based on a circular symmetric neighborhood of P members with radius R. To achieve gray level invariance, the central pixel g_c is subtracted to each neighbor g_p, assigning the value 1 to the result if the difference is positive and 0, otherwise. LBPs are defined as follows:

$$LBP_{R,P} = \sum_{p=0}^{P} a(g_p - g_c) \cdot 2^p$$

A Gabor filter is a special case of wavelets [6] which is essentially a Gaussian modulated by a complex sinusoid s. In 2D, it has the following form in the spatial domain:

$$h(x,y) = \frac{1}{2\pi\sigma^2} \exp\{-\frac{1}{2}[(\frac{x^2+y^2}{\sigma^2})]\} \cdot s(x,y)$$
$$s(x,y) = \exp[-i2\pi(Ux + Vy)] \qquad \phi = \arctan V/U$$

where σ is the standard deviation, U and V represent the 2D frequency of the complex sinusoid, and ϕ is the angle of the frequency.

According to [12], one of the main differences in the appearance of calcified tissue compared to the rest of tissue types is the shadow which is appreciated behind it. In order to detect this shadow, we perform an accumulative mean of the pixels gray values on the polar image from a pixel to the end of the column (the maximal depth considered). As a result of extracting the texture descriptors, we construct an n-dimensional feature vector where $n = k + l + m + 1$, k is the number of co-occurrence matrix measurements, l is the number of Gabor filters, m is the number of LPB and the last feature is the measure of the "shadow" in the image.

Intravascular data set

In order to generate the data sets, we used the RF signals and their reconstructed images from a set of 10 different patients with Left Descent Artery pullbacks acquired in Hospital "German Trias i Pujol" from Barcelona, Spain. All these pullbacks contain the three classes of plaque. For each one, 10 to 15 different vessel

sections were selected to be analyzed. Two physicians independently segmented 50 areas of interest per pullback. From these segmentations we took 15 regions of interest (ROI) of tissue per study randomly making a total of 5000 evaluation ROIs. To build the data set, these selections were mapped in both RF signals and reconstructed images. In order to reduce the variability among different observers, the regions where both cardiologist agreed have been taken under consideration. Some samples from the data set are shown on the left of Fig. 9.

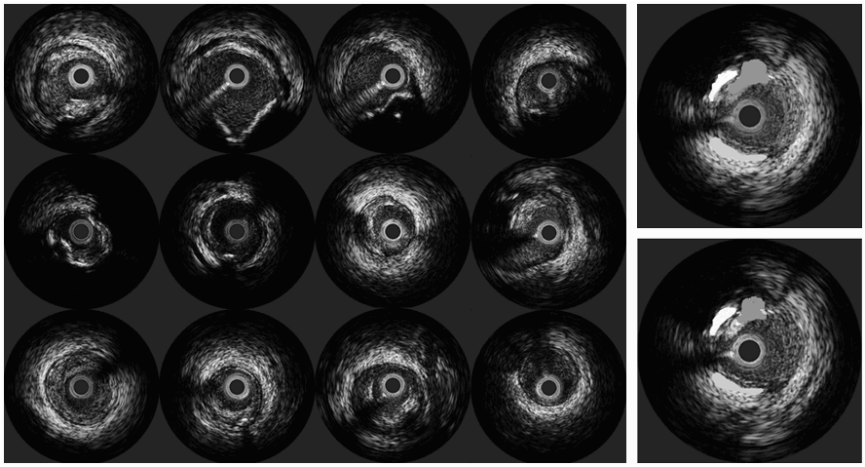


Fig. 9 Left: IVUS data set samples. Right: (top) segmentation by a physician and (down) Automatic classification with Texture-Based Features. The white area corresponds to calcium, the light gray area to fibrosis, and the dark gray area to soft plaque.

To generate the data set on texture features, the intersection between segmented images is mapped into a feature vector. Then, all the features collected are categorized by patient and each of the three possible plaques type. The image features are extracted by using the previous texture descriptors: Co-ocurrence Matrix, Local Binary Patterns, and Gabor Filters. Those features are calculated for each pixel and gathered in a feature vector of 68 dimensions. An example of a manual and automatic texture-based segmentation for the same sample is shown on the right of Fig. 9.

To generate the data set of RF features, the RF signals have been acquired using a 12-bit acquisition card with a sampling rate of $f_s = 200MHz$. The IVUS equipment used is Galaxy II from Boston Scientific with a catheter transducer frequency of $f = 40Mhz$, and it is assumed a sound speed in tissue of $1565m/s$. Each IVUS image consists of a total of 256 A-lines (ultrasound beams), with a radial distance of $r = 0.65cm$. The attenuation in tissue factor used is $\alpha = 1Db/Mhz \times cm$. To analyze the RF signals, the sliding window is composed of $n = 64$ samples of depth and $m = 12$ radial A-lines, and the displacement is fixed in 16 samples and four A-lines. The power spectrum of the window ranges from 0 to $100MHz$ and it is sampled

by 100 points. Then, it is complemented with two energy measures yielding a 102 feature vector.

We also consider a third data set that concatenates the descriptors from the previous RF and texture-based features, obtaining a feature vector of length 170 features.

Slope-based features

Finally, the fourth data set considers the slope-based features proposed by [18]. In particular, each sample is characterized by means of 14 slope-based features corresponding to: maximum power in DB from 20 to 60 MHz, frequency at the maximum power, negative slope in db/MHz between maximum and 60, minimum power in that slope, frequency corresponding to this negative slope, the estimated y intercept of this slope, the positive slope in db/Mhz between 20 and maximum, minimum power in that slope, frequency corresponding to this negative slope, the estimated y intercept of this slope, the mean power, the power at 0 MHz, power Db at 100 Mhz, and the power at the midband frequency (40 MHz) in DB [18].

To solve the problem of Intravascular tissue characterization we apply the Sub-class ECOC strategy over the four previous data sets.

IVUS characterization with sub-classes

For this experiment, we use the four previous IVUS data sets. To measure the performances, we apply leave-one-patient-out evaluation.

Applying *NMC*, Adaboost, and *FLDA* over a set of ECOC configurations, the performance results for RF features, texture-based features, combined RF and texture-based features, and slope-based features are shown in Fig. 10. Comparing the results among the different data sets, one can see that the worst performances are obtained by the RF and slope-based features, which obtain very similar results for all the base classifiers and ECOC configurations. The texture-based features obtain in most cases results upon 90%. Finally, the data set of combined RF and texture-based features slightly outperform the results obtained by the texture-based feature, though the results do not significantly differ.

Concerning the classification strategies, observing the obtained performances in Fig. 10, one can see that independently of the data set and the ECOC design applied, the Sub-class ECOC approach always attains the best results. To compare these performances, the mean rank of each ECOC design considering the twelve different experiments is shown in table 6. In this case, the rankings are obtained estimating each particular ranking r_i^j for each problem i and each ECOC configuration j, and computing the mean ranking R for each ECOC design as $R_j = \frac{1}{N}\sum_i r_i^j$, where N is the total number of problems (3 base classifiers $\times$ 4 data sets). One can see that the Sub-class ECOC attains the best position for all experiments. To analyze if the difference between methods ranks are statistically significant, we apply the Friedman and Nemenyi tests. In order to reject the null hypothesis that the measured ranks differ from the mean rank, and that the ranks are affected by randomness in the results, we use the Friedman test. The Friedman statistic value is computed as follows:

$$X_F^2 = \frac{12N}{k(k+1)}[\sum_j R_j^2 - \frac{k(k+1)^2}{4}] \tag{8}$$

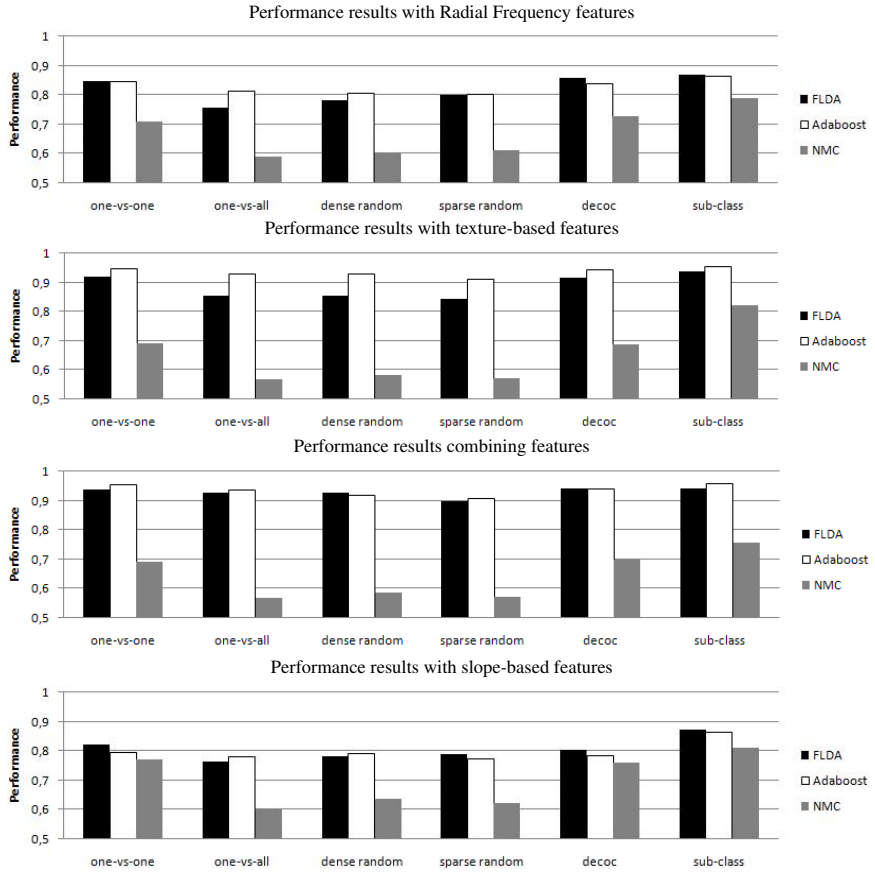


Fig. 10 Performance results for different sets of features, ECOC designs and base classifiers on the IVUS data set.

In our case, with $k = 6$ ECOC designs to compare, $X_F^2 = 30.71$. Since this value is undesirable conservative, Iman and Davenport proposed a corrected statistic:

$$F_F = \frac{(N-1)X_F^2}{N(k-1) - X_F^2} \tag{9}$$

Applying this correction we obtain $F_F = 11.53$. With six methods and twelve experiments, F_F is distributed according to the F distribution with 5 and 55 degrees of freedom. The critical value of $F(5,55)$ for 0.05 is 2.40. As the value of F_F is higher than 2.45 we can reject the null hypothesis. One we have checked for the for the non-randomness of the results, we can perform a post hoc test to check if one of the techniques can be singled out. For this purpose we use the Nemenyi test - two techniques are significantly different if the corresponding average ranks differ by at least the critical difference value (CD):

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}} \tag{10}$$

where q_α is based on the Studentized range statistic divided by $\sqrt{2}$. In our case, when comparing six methods with a confidence value $\alpha = 0.10$, $q_{0.10} = 1.44$. Substituting in eq.10, we obtain a critical difference value of 1.09. Since the difference of any technique rank with the Sub-class rank is higher than the *CD*, we can infer that the Sub-class approach is significantly better than the rest with a confidence of 90% in the present experiments.

Table 6 Mean rank for each ECOC design over all the experiments.

ECOC design	one-versus-one	one-versus-all	dense random
Mean rank	2.33	5.08	4.25
ECOC design	sparse random	decoc	sub-class
Mean rank	5.00	2.67	1.00

4 Conclusions

In this paper, we reviewed the state-of-the-art on coding and decoding designs of Error-Correcting Output Codes. We analyzed the most recent ensemble strategies, showing their benefit to deal with multi-class classification in image analysis. Moreover, the different ECOC configurations were used to solve two challenging computer vision applications: traffic sign classification and intravascular ultrasound tissue characterization, with high success.

Acknowledgements This work has been partially supported by the projects TIN2006-15694-C02 and CONSOLIDER-INGENIO 2010 (CSD2007-00018).

References

1. E. Allwein, R. Schapire, and Y. Singer. Reducing multiclass to binary: A unifying approach for margin classifiers. *JMLR*, 1:113–141, 2002.
2. A. Bovik, M. Clark, and W. Geisler. Multichannel texture analysis using localized spatial filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):55–73, 1990.
3. A. P. Burke, A. Farb, G. T. Malcom, J. Smialek, and R. Virmani. Coronary risk factors and plaque morphology inmen with coronary disease who died suddently. *The New England Journal of Medicine*, 336(18):1276–1281, 1997.
4. J. Casacuberta, J. Miranda, M. Pla, S. Sanchez, A.Serra, and J.Talaya. On the accuracy and performance of the geomobil system. In *International Society for Photogrammetry and Remote Sensing*, 2004.

5. K. Crammer and Y. Singer. On the learnability and design of output codes for multi-class problems. In *Machine Learning*, volume 47, pages 201–233, 2002.
6. J. Daugman. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *Journal of the Optical Society of America*, 2(A):1160–1169, 1985.
7. H. Daume and D. Marcu. A bayesian model for supervised clustering with the dirichlet process prior. *Journal of Machine Learning Research*, pages 1551–1577, 2005.
8. J. Demsar. Statistical comparisons of classifiers over multiple data sets. *JMLR*, pages 1–30, 2006.
9. S. Escalera, O. Pujol, and P. Radeva. Loss-weighted decoding for error-correcting output codes. *VISAPP, to appear*.
10. S. Escalera, O. Pujol, and P. Radeva. Boosted landmarks of contextual descriptors and forest-ecoc: A novel framework to detect and classify objects in clutter scenes. *Pattern Recognition Letters*, 28(13):1759–1768, 2007.
11. R. Ghani. Combining labeled and unlabeled data for text classification with a large number of categories. *Int. conf. Data Mining*, pages 597–598, 2001.
12. D. Gil, A. Hernandez, O. Rodriguez, F. Mauri, and P. Radeva. Statistical strategy for anisotropic adventitia modelling in ivus. *IEEE Trans. Medical Imaging*, 27:1022–1030, 2006.
13. T. Hastie and R. Tibshirani. Classification by pairwise grouping. *NIPS*, 26:451–471, 1998.
14. C. Karla, B. Joel, P. Oriol, N. Salvatella, and P. Radeva. In-vivo ivus tissue classification: a comparison between rf signal analysis and reconstructed images. In *Progress in Pattern Recognition*, pages 137–146. Springer Berlin / Heidelberg, 2006.
15. J. Kittler, R. Ghaderi, T. Windeatt, and J. Matas. Face verification using error correcting output codes. *CVPR*, 1:755–760, 2001.
16. E. B. Kong and T. G. Dietterich. Error-correcting output coding corrects bias and variance. *ICML*, pages 313–321, 1995.
17. H. Madala and A. Ivakhnenko. *Inductive Learning Algorithm for Complex Systems Modelling*. CRC Press Inc, 1994.
18. A. Nair, B. K. nd N. Obuchowski, and G. Vince. Assesing spectral algorithms to predict atherosclerotic plaque composition with normalized and raw intravascular ultrasound data. *Ultrasound in Medicine & Biology*, 27:1319–1331, 2001.
19. N. J. Nilsson. Learning machines. *McGraw-Hill*, 1965.
20. P. Ohanian and R. Dubes. Performance evaluation for four classes of textural features. *Pattern Recognition*, 25:819–833, 1992.
21. T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:971–987, 2002.
22. W. H. Organization. World health organization statistics. *http://www.who.int/entity/healthinfo/statistics/*, 2006.
23. h. OSU-SVM-TOOLBOX.
24. J. Proakis, C. Rader, F. Ling, and C. Nikias. Advanced digital signal processing. *Mc Millan*, 1992.
25. P. Pudil, F. Ferri, J. Novovicova, and J. Kittler. Floating search methods for feature selection with nonmonotonic criterion functions. *ICPR*, pages 279–283, 1994.
26. O. Pujol, S. Escalera, , and P. Radeva. An incremental node embedding technique for error correcting output codes. *Pattern Recognition, to appear*.
27. O. Pujol, P. Radeva, , and J. Vitrià. Discriminant ecoc: A heuristic method for application dependent design of error correcting output codes. *PAMI*, 28:1001–1007, 2006.
28. T. Randen and J. H. Husoy. Filtering for texture classification: A comparative study. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 4:291–310, 1999.
29. W. Utschick and W. Weichselberger. Stochastic organization of output codes in multiclass learning problems. *Neural Computation*, 13(5):1065–1102, 2001.
30. T. Windeatt and G. Ardeshir. Boosted ecoc ensembles for face recognition. *International Conference on Visual Information Engineering*, pages 165–168, 2003.
31. Q. Zgu. Minimum cross-entropy approximation for modeling of highly intertwining data sets at subclass levels. *Journal of Intelligent Information Systems*, pages 139–152, 1998.
32. J. Zhou and C. Suen. Unconstrained numeral pair recognition using enhanced error correcting output coding: a holistic approach. *Proc. in Conf. on Doc. Anal. and Rec.*, 1:484–488, 2005.

33. M. Zhu and A. M. Martinez. Subclass discriminant analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1274–1286, 2006.