Diversity-aware classifier ensemble selection via f-score

Ingrid Visentini, Lauro Snidaro, and Gian Luca Foresti

Department of Mathematics and Computer Science, University of Udine, Italy, via delle Scienze 206, 33100 Udine

Abstract

The primary effect of using a reduced number of classifiers is a reduction in the computational requirements during learning and classification time. In addition to this obvious result, research shows that the fusion of all available classifiers is not a guarantee of best performance but good results on the average. The much researched issue of whether it is more convenient to fuse or to select has become even more of interest in recent years with the development of the Online Boosting theory, where a limited set of classifiers is continuously updated as new inputs are observed and classifications performed. The concept of online classification has recently received significant interest in the computer vision community. Classifiers can be trained on the visual features of a target, casting the tracking problem into a binary classification one: distinguishing the target from the background.

Here we discuss how to optimize the performance of a classifier ensemble employed for target tracking in video sequences. In particular, we propose the F-score measure as a novel means to select the members of the ensemble in a dynamic fashion. For each frame, the ensemble is built as a subset of a larger pool of classifiers selecting its members according to their F-score. We observed an overall increase in classification accuracy and a general tendency in redundancy reduction among the members of an f-score optimized ensemble. We carried out our experiments both on benchmark binary datasets and standard video sequences.

Key words: F-score, Classifiers Selection, Classifiers Fusion, Tracking via classification, Online tracking

1. Introduction

It is well known that the fusion of an ensemble of "weak" independent classifiers can lead to substantial performance improvements with respect to a single monolithic classifier [21]. The term "weak" is used to indicate a classifier that is not particularly specialized or trained for the problem at hand (i.e. it is sufficient that classification performance be slightly better than random guessing). These ensembles can be employed in a broad variety of applications, from medical imaging [48] to network security [16], from biometric person identification [24] to remote sensing [59], in a large range of real-world domains [40].

To fuse classifiers a large number of possible rules can be used [47]: for instance, sum and product [25], Bagging [5] and Boosting [14], Random Subspaces [22], or oracles [33]. Considering couples of classifiers, mutual information [44], Q statistic [60], diversity-based criteria [32, 56] or correlation, for instance, can represent valid pairwise measures that consider their independence to merge their outputs.

To save computational time, an option is to employ only a selection of classifiers instead of the entire set [57]. The selection procedure is aimed at forming a reduced ensemble by choosing within a pool the subset of classifiers that maximizes the performance [30] or, alternatively, reduce the error. This approach is often applied to features [20] to decrease, for instance, the dimensionality of the input space or to choose a more robust subset, but it is also used for classifiers [1], to achieve better performance or to satisfy real-time constraints. In this context, a classifier combination strategy that links together selection and fusion includes switching between fusion and selection [30, 50, 12].

The recent development of online learning methods [39, 43, 34] has opened the possibility to build on-the-fly a classifier ensemble and to train it with incoming samples in an unsupervised manner and without any prior knowledge of data distribution. These techniques are based on a evolution of the original Boosting [55] algorithm and rely on a fixed size ensemble of classifiers, whose weights are continuously updated according to some statistical information on observed samples. However, for instance, the Online Boosting technique can present an optimistic view of the classifiers behaviour, scoring only the distinction between correctly and wrongly labelled (classified) samples without considering the skewness of the training set (see [15] for a discussion on ensembles for the class imbalance problem); assessing the performance of the classifiers in presence of an unbalanced number of training samples can be misleading.

For this reason, Pham and Cham [46] proposed an asymmetric online boosting algorithm, where both a parameter k, that takes into account the asymmetry of the class labels presented to the classifiers, and the number of false/true positives and false/true negatives are considered in the tuning of the coefficients of the linear combination of the classifiers. Even if the problem of unbalanced classes is handled, the application of the entire pool of classifiers can be still computationally expensive.

Email address: lauro.snidaro@uniud.it (Ingrid Visentini, Lauro Snidaro, and Gian Luca Foresti)

1.1. Online classification for video tracking

An interesting application of online learning methods is target tracking in video sequences, that recently has received a new boost thanks to the tracking via classification concept [8, 18, 45, 53, 38]. The idea is that classifiers can be trained on the visual features of a target, casting the tracking problem into a binary classification one: distinguishing the target from the background. In the vast majority of tracking applications, the target changes its appearance as it moves within the field of of view of a video sensor due to rotations (of the target and/or camera) and perspective distortions. For this reason the model learned by the classifiers should be updated at every new frame in a continuous detect ↔ update cycle. The recent availability of methods for online training classifier ensembles on incoming data, like Online Boosting [18], has thus stoked the interest for this type of tracking instrument. The advantages over existing tracking methods are clear:

- the ensemble can be trained on heterogeneous features (e.g. colour features, texture, motion, etc.) thus improving the robustness of the detector
- being trained on a specific object, it works as a detector of the particular instance. In the case of multiple objects in the scene, each of them is tracked by a dedicated ensemble (i.e. trained on the target's features).

Recent works include Avidan's Adaboost-based tracker [3], that exploits features associated to every pixel. However, the work uses the classic Adaboost algorithm and does not learn online the appearance of the target. In [10] the most discriminative color features to separate the target from the background are chosen by applying a two-class variance ratio to log likelihood distributions computed from samples of object and background pixels. In a later work, heterogeneous features have been combined adopting the same fusion method [42]. In these two works the features are ranked and selected afresh for each frame without considering past history (i.e. how features performed in the previous frames).

In [18] the Online Boosting technique devised by Oza [39] is adapted for visual target tracking. Albeit this idea is effective, since it uses the online ensemble learning paradigm, it employs an architecture that relies on a fixed cardinality ensemble. No selection is applied and this can be detrimental for real-time constraints.

1.2. Algorithm outlook

In this work, we propose a new criterion based on the F-score measure to select classifiers from a set of constantly updated ensemble members (Figure 1). This criterion has been used in [9] applied to SVM, but its application in online learning is still unexplored to the best of our knowledge.

1. INITIALIZATION: The full ensemble members are supervisedly trained with a set of labelled samples. This initialization is done once at the startup.

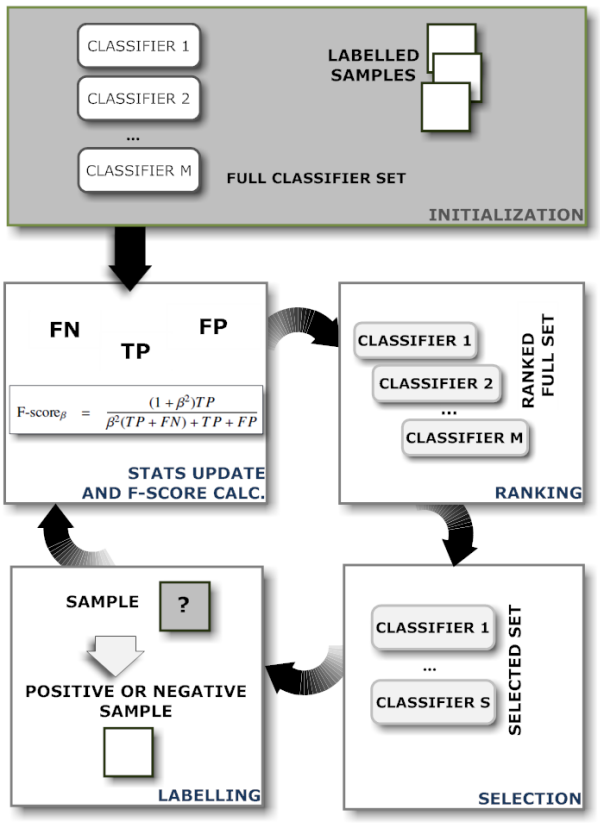

Figure 1: Architecture of the proposed approach for selecting classifiers online, based on their F-score measure. The loop is described in detail in Section 1.2.

2. STATS UPDATE: The statistics (TN and TP, FN and FP, precision, recall and F-score) of each classifier of the full pool are individually updated. Since this step is a matter of storing a few variables the computation for this step is fast.
3. RANKING: The members are ranked in descending order using their revisited F-score value.
4. SELECTION: The classifiers for forming the reduced pool can be selected, as presented here in the paper, for being within the first S classifiers in the ranking.
5. LABELLING: The selected ensemble classifies a new unlabelled testing/validation sample. The labelling of the sample is performed by the selected set only, while the other ensemble members are not considered in this phase.
6. LOOPING: While there are test samples available, repeat all the steps from (2).

This general proposed approach can be used in online and offline datasets. We will test it in both cases:

- in the (offline) case of UCI datasets, we train the ensemble members with a minimal training set of randomly picked samples (1/3 the size of the dataset). We then re-compute the F-score based ranking for each new validation sample

(2/3 of the dataset). The validation samples are processed one-by-one.

- In the (online) case of a video sequence, where data is continuously streaming in, the ensemble is trained on a small amount of initial frames, where the positive samples are manually located as image patches (corresponding to the target) in a semi-supervised fashion. The validation samples are then found and labelled on-the-fly by the selection classifiers, which analyse the video stream frame-by-frame and picking the most likely image patch containing the target. In this case, the found image patch is labelled as +1 (positive), while random patches from the background are used as negative samples (class -1) to recompute the F-score values for all the classifiers in the pool.

1.3. Novelty of this work

The proposed approach provides the following advantages:

- it provides a way to rank the performance of each member of the ensemble;
- it maintains the history of the performance of each classifier (thus allowing better occlusion handling in video tracking applications);
- it evaluates classifiers instead of features thus allowing the transparent integration of heterogeneous features;
- explicit handling of asymmetric samples distributions;
- when applied to video tracking it greatly speeds up the search phase by applying only a reduced number of selected classifiers. This allows fast tracking without a prior model and without an off-line training for real-time applications.

Fast tracking without a prior model and without offline training is achieved by considering the ability of the classifiers to discriminate between the training samples. The F-ratio is used to sort the predictors pool and to form the best subset. The selection task is particularly useful in a preprocessing step to reduce the number of ensemble members and then to reduce the computational burden, removing at the same time redundant or erroneous classifiers. Further extending developing our preliminary work [58], we bring here additional experimental evidence to our findings along with a digression on the much researched issue of whether it is more convenient to fuse or to select. In particular, the concepts of accuracy and diversity of a classifier ensemble are analysed, in light of the performances obtained by ensembles selected via the f-score measure on standard datasets, with some interesting experimental results.

1.4. Organization of the paper

The paper is organized as follows: Section 2 provides the required background notions, briefly covering the concepts underpinning classifier ensembles along with the definitions of the *precision* and *recall* performance metrics. Section 3 describes the proposed approach starting with a criterion to optimize the f-score performance measure, its extension to the multi-class case, and its application as a classifier selection method. Section 4 provides experimental evidence of the classification performance obtained by the proposed approach on standard UCI datasets with a discussion on classification accuracy and diversity. Section 5 shows the application as tracking via classification method on standard and real-world video sequences, while conclusions and final remarks are given in Section 6.

2. Background on classifier ensembles and Precision-Recall metrics

2.1. Classifier ensemble

Combining classifiers is the first step to be taken in order to create an ensemble. Starting from several classifiers $h_1, h_2, \ldots, h_M$, an ensemble H of predictors can be built organizing the members with several linear fusion rules (e.g. sum, average, product, etc. [25]) or considering a non-linear combination technique (e.g. Dempster-shafer, neural network combiners, etc. [47]).

In this work, we decided to employ the mean rule to build a linear combination of experts, so that the final ensemble takes the form

$$H(x) = \frac{1}{M}\left(\sum_{m=1}^{M} h_m(x)\right) \tag{1}$$

To classify a new sample x, belonging to the sample set $\mathbf{X}$, a (weak) classifier $h : \mathbf{X} \rightarrow \{+1, -1\}$ assigns it to the most probable class ω, that is

$$\begin{aligned} \omega &= \arg\max_{\omega_c} P(\omega_c | h(x)) \\ &\propto \arg\max_{\omega_c} P(\omega_c) P(h(x) | \omega_c) \end{aligned} \tag{2}$$

To evaluate the performance of a classifier, or an ensemble, on the training set, many measures can be used. We propose to use precision and recall, as they are fast to compute and suited for online computation, and offer a quite robust indication of how accurately the classifier is labelling the training set.

2.2. Precision and recall

Precision and recall are widely used to evaluate an algorithm's performance in Information Retrieval (IR) [4] or, more generically, to measure the quality of a classification process [11]. With respect to ROC curves, PR curves are more meaningful when the number of negative samples greatly exceeds the number of positive ones since they take into account the skewness between classes [11].

Definition Considering a training set constituted of a set of N couples $(x_1, \omega_1), (x_2, \omega_2), \ldots, (x_N, \omega_N)$ where $x_n \in \mathbf{X}$ are training samples and $\omega_n \in \{+1, -1\}$ their labels, the *precision* π of a classifier h is defined as the probability that K items

$\{x_1, \ldots, x_k\}$ in the training set, that are labelled as belonging to class $\omega = +1$, actually belong to that class

$$\begin{aligned} \pi \equiv & \quad \frac{1}{K} \sum_k P(\omega = +1 | h(x_k) = +1) \\ = & \quad \frac{1}{K} \sum_k \frac{P(\omega=+1, h(x_k)=+1)}{P(h(x_k)=+1)} \end{aligned} \tag{3}$$

Recall is defined as the probability that the items belonging to class $\omega = +1$ are labelled by the classifier h as belonging to that class

$$\begin{aligned} \rho \equiv & \quad \frac{1}{K} \sum_k P(h(x_k) = +1 | \omega = +1) \\ = & \quad \frac{1}{K} \sum_k \frac{P(h(x_k)=+1, \omega=+1)}{P(\omega=+1)} \end{aligned} \tag{4}$$

As known in the literature, a trade-off between Precision and Recall is intrinsic, as increasing one means reducing the other [7]. The relation between precision and recall is given by

$$\pi = \rho \frac{P(\omega = +1)}{P(h(x) = +1)} \tag{5}$$

since, without losing in generality, if we assume that for each $x \in \mathbf{X}$

$$\begin{aligned} \pi = & \quad P(\omega = +1 | h(x) = +1) \\ = & \quad \frac{P(h(x)=+1|\omega=+1)P(\omega=+1)}{P(h(x)=+1)} \end{aligned} \tag{6}$$

then, from (4) we know that the first term of the denominator on the second line is the recall ρ by definition. Thus, we can write

$$P(h(x) = +1) = \frac{\rho}{\pi} P(\omega = +1) \tag{7}$$

Being $P(\omega = +1)$ known a priori, the interesting term is the ratio ρ/π that, as proved, constitutes the balance between precision and recall.

We can also define precision and recall in terms of hits and missed classification.

Definition Considering the training set constituted of a set of N couples, the *true positives* are defined as the number of positive samples correctly classified by the classifier h and counted by the indicator function I

$$TP = \sum_{n=1}^{N} I(h(x_n) = +1, \omega_n = +1) \tag{8}$$

The *false positives* and *false negatives* respectively are the amount of negative samples classified as positives, and the number of misclassified positive samples

$$FP = \sum_{i=1}^{N} I(h(x_n) = +1, \omega_n = -1) \tag{9}$$

$$FN = \sum_{i=1}^{N} I(h(x_n) = -1, \omega_n = +1) \tag{10}$$

From these definitions, the following equations hold

$$\begin{aligned} \pi \equiv & \quad \frac{TP}{TP+FP} \\ \rho \equiv & \quad \frac{TP}{TP+FN} \end{aligned} \tag{11}$$

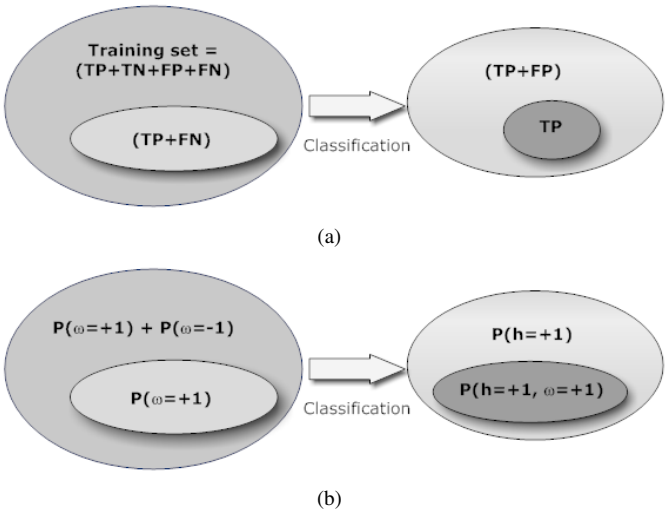

Figure 2: Illustration of a classification procedure in terms of confusion matrix (a) and Bayesian probabilities (b). In both cases, from the training set, where the positive samples are a subset, the result of the classification process is a subset of patterns labelled as positives. Among these, only a portion are the true positives, that is the assigned label matches with the true label. From a graphical perspective, *recall* is related to the ability of the classifier to make the inner circles on both sides (training set and classification result) overlap, while *precision* refers to the ability to make the circles on the right side (classification results) overlap.

Following the above definition, the relation between precision and recall can be defined as

$$\pi = \rho \left[N^+ (TP + FP)^{-1} \right] \tag{12}$$

which comes straightforwardly from (5) and from $N^+ = TP + FN$.

As depicted in Figure 2, supposing to have a training set of size N, the precision is related to the number of "hits" over the total positive-labelled samples, while the recall refers to the ability of the classifier to correctly extract the largest number of relevant (in our case, belonging to the positive class) samples from the training set. From a graphical perspective, recall is the ability of the classifier to make the inner circles on both training set and classification side overlap, while precision refers to the ability to make the inner and outer circles only on the classification (right) side overlap.

Moreover, considering the parallel proposed in Figure 2, $P(\omega = +1)$ can be imagined as related to the number of all positive samples out of all possible training samples, that is $(TP + FN)/N$, and $P(h(x) = +1)$ as the number of samples in the training set that are labelled as positive by the classifier h, that is $(TP + FP)/N$.

3. Proposed approach

The approach presented in this paper relies on the idea of forming an ensemble of classifiers by selecting them from a wider set according to a performance measure (selection by pruning [37] is also an alternative). The full set, comprising a fixed number of classifiers, is initially trained with a few samples; then, as illustrated in Figure 1 at each round the experts

are ranked and selected using their F-score value. The selected set will assign to few samples a label, to be used as the ground truth to determine the F-score of all the classifiers at the next round. The selection process will be repeated in every round in an unsupervised fashion.

When considering the tracking task, the difference is that the the classifier ensemble discriminates between the object and several random background patches in a frame. The target is iteratively extracted from the foreground in each frame of the video sequence, and used as a new sample to repeat the selection-fusion loop.

3.1. F-score and its optimization

Usually precision and recall are compared considering a fixed value for both, or combined into a single formula, such as the F-score, which is a weighted one-dimensional indicator of the two. The F-score, firstly proposed in [54], is defined as their weighted harmonic mean,

$$\text{F-score}_\beta \equiv (1+\beta^2)\frac{\rho\pi}{\beta^2\pi+\rho} \tag{13}$$

When $\beta = 1$ the F-score evenly balances the two components, as it becomes

$$\text{F-score}_1 = 2\frac{\rho\pi}{\pi+\rho} \tag{14}$$

On the other hand, when $\beta < 1$ it favours recall, while precision is preferred otherwise. The F-score spans in the interval (0,1) and high values correspond to good classification quality. When this measure is maximum, all the data is classified correctly.

Arguably, other (even earlier) metrics can be seen as particular cases of F-score, even assuming that the selection exploits some a priori knowledge, while others, named *ranking-based* (i.e. ROC and RP curves, nP and nR, AP, MAP, iMAP, etc.) sort the results and provide a ranking of the outputs. The first case is not desirable, while the second is not significant in our case. For instance, the derivation of E-measure is trivial

$$E_\alpha = 1 - \frac{1}{(\frac{\alpha}{\pi}) + (\frac{(1-\alpha)}{\rho})} = 1 - F_\beta \tag{15}$$

when $\alpha = \frac{1}{\beta^2+1}$ [54].

Unfortunately, to the best of our knowledge there are no classifiers that directly optimize the F-score or the precision–recall balance, as already noted in [52]; for this reason, we need to find a common criterion to optimize the measure for ranking the classifiers.

Using (11), we can express the F-score as

$$\begin{aligned}\text{F-score}_\beta &= \frac{(1+\beta^2)TP}{\beta^2(TP+FN)+TP+FP} \\ &= \frac{(1+\beta^2)}{\beta^2+1+\frac{\beta^2FN+FP}{TP}}\end{aligned} \tag{16}$$

Since $\beta^2 \geq 0$, in order to maximize the F-score, and thus have the best balance between precision and recall, the ratio

$$\frac{\beta^2FN+FP}{TP} \tag{17}$$

in the denominator of (16) should be minimized. The smaller this ratio, the more discriminative the classifier.

3.2. Extension to multi-class problem

The overall F-score value of the entire classification problem can be computed by two different types of average, micro-average and macro-average [41].

Having C classes, in micro-averaging (MI), π and ρ are redefined to consider an average of all individual decisions on all the classes

$$\begin{aligned}\pi_{MI} &\equiv \frac{\sum_{c=1}^{C} TP_c}{\sum_{c=1}^{C}(TP_c+FP_c)} \\ \rho_{MI} &\equiv \frac{\sum_{c=1}^{C} TP_c}{\sum_{c=1}^{C}(TP_c+FN_c)}\end{aligned} \tag{18}$$

F-measure is computed globally as standard F-measure over all classes

$$\text{F-score}_{MI} = (1+\beta^2)\frac{\rho_{MI}\pi_{MI}}{\beta^2\pi_{MI}+\rho_{MI}} \tag{19}$$

If dealing with single-label classification, the micro-average F-score is the same as accuracy.

This formulation allows us to give equal weight to each sample and is therefore considered as an average over all the (sample,class) pairs . However, when the classifier behaviour is similar on common classes, it can lead to unbalanced performance analysis [36].

Another way to calculate a multi-class F-score is the so called macro-averaging (MA) procedure; in this case, F-score is computed locally over each class first and then the (weighted by class size) average over all C classes is taken.

$$\text{F-score}_{MA} = \sum_{c=1}^{C} \text{F-score}_c \tag{20}$$

Macro-averaged F-score gives equal weight to each class, regardless of its frequency, and for this reason is is highly influenced by the classifier's behaviour on anomalous classes.

3.3. F-score as selection method

In this work, we explore the use of the F-score as a means to rank classifiers. Instead of considering the fusion of all the classifiers, we aim at combining only a part of the whole set. Looking at Figure 3, the direct combination approach directly provides a classifier output, while, if the selection step is present the combination is preceded by a ranking and discarding phase that reduces the cardinality of the ensemble. This new set, said $\hat{H}$, is composed by S classifiers that are fused as

$$\hat{H}(x) = \frac{1}{S}\left(\sum_{s=1}^{S} h_s(x)\right) \tag{21}$$

and is formed by choosing the best S predictors according to the F-score ranking. The formulation in (16) allows one to use the F-score as a fast classifier selection criterion that can be applied to every round of computation. The measure is individually computed for each classifier and provides a performance measure to rank all the members of the ensemble quickly.

An intrinsic advantage of the proposed solution is that this allows one to form a flexible selection of classifiers. The members can be potentially discarded and replaced when their performance starts to decrease, or new ones can be added if the

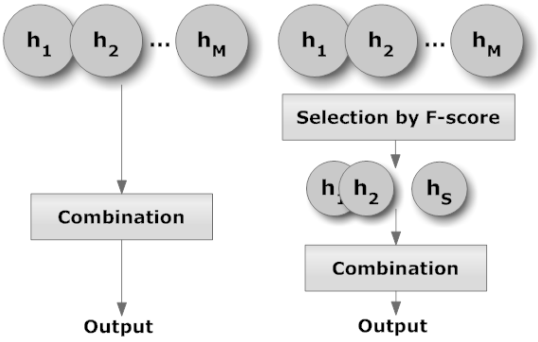

Figure 3: Comparison of a direct combination method (a) and a selection step, like in our case, that precedes the combination phase (b).

system encounters a critical situation. In this paper, at each step of computation we will employ a selection set of fixed cardinality, generated from a larger set of classifiers that will be not discarded. Keeping fixed this general pool and constructing a dynamic selection set will help us to understand how the proposed measure works. However, a dynamic technique to regulate the number of classifiers in the selection ensemble and to replace the classifiers in the main set will be the subject of future investigation.

Algorithm 1 describes in detail the ranking and the selection steps. The selection set $\hat{H}$ assigns a label to the sample (*sample(s) labelling*). All the classifiers in the extended set are tested on the sample, and their predictions are compared with the "truth" label assigned in the previous phase (*performance evaluation* step). The historical values of misclassified and correctly classified samples contribute to calculate the F-score denominator (*ranking* phase) for every classifier in the set. This value is then used to pick $S << M$ classifiers from the original set H and to form the selection set to be used in the next round. Notice that the selection procedure has linear complexity in the number of classifiers in the pool, that is $O(M)$.

Algorithm 1: F-score based selection

Require: Sample x
Require: Classifiers selection $\hat{H}$
Require: Classifiers pool H
Require: Skewness weight β
// For every classifier in H
// The selected set assigns a label to the sample
$\omega \leftarrow \arg\max_{\omega_k} P(\omega_k|\hat{H}(x))$
// Each classifier is tested on the pattern x
// and performance metrics are calculated
for $m \leftarrow 1$ *to* M **do**
  $h_{out} \leftarrow \arg\max_{\omega_k} P(\omega_k|h_m(x))$ as in (2)
  if $\omega = +1$ and $h_{out} = +1$ **then**
    $TP_m \leftarrow TP_m + 1$
  end if
  if $\omega = +1$ and $h_{out} = -1$ **then**
    $FN_m \leftarrow FN_m + 1$
  end if
  if $\omega = -1$ and $h_{out} = +1$ **then**
    $FP_m \leftarrow FP_m + 1$
  end if
  // Calculate the denominator of Eq. (16)
  $Fdenom_m \leftarrow \frac{\beta^2 FN_m + FP_m}{TP_m}$
end for
// Sort the denominators in ascending order
$Fdenom \leftarrow \text{sort}(Fdenom)$
// Fill the ensemble $\hat{H}$ to use at the next round
for $i \leftarrow 1$ *to* S **do**
  $s \leftarrow \text{index}(Fdenom(i))$
  $\hat{H} \leftarrow \hat{H} \bigcup h_s$
end for
Return $\hat{H}$

4. F-score and diversity: a discussion on benchmark binary datasets

Classifier fusion is known to provide increased accuracy by combining the opinions of multiple experts, thus improving reliability by reducing uncertainty [31]. At the same time, eliminating redundant classifiers provides better accuracy and reduces the computational time required. It is important then to have non redundant classifiers taking part to the ensemble in order to reduce the error, thus maintaining the level of *diversity* in the ensemble. One way to achieve diversity is forcing it while training, that is through different inputs or training sets. For instance, Bagging can be employed to partition the training data and use it to train several classifiers. Another method consists in varying the classifier parameters, if any, introducing variations and noise. Therefore, monitoring the level of disagreement among the classifier can be an index of how much the ensemble is varied and comprises independent members. In this respect, diversity measures can be pairwise, when they consider pairs of classifiers and average over results, and non–pairwise, when the diversity refers to the whole ensemble and its performance, and all the members are measured together. Yule's Q statistics [60], the correlation coefficient, the disagreement measure, and the double fault measure [17], for instance, belong to the pairwise set, while Kohavi–Wolpert's variance [26] or Kuncheva's entropy [31] are in the second group. To the best of our knowledge, up to now no unique definition or formalization of diversity has been given [28].

In our experiments we will consider and compare eight different measures of diversity (taken from [29]), with the purpose of showing how the classifier selection performed by the F-score criterion can improve the accuracy of the results, but at the same time without sensibly affecting the diversity of the ensemble.

4.1. Experimental setup

To test how the proposed technique promotes a robust, accurate, diverse ensemble, we have chosen several benchmark binary datasets from the UCI Machine Learning repository [13]. In this paper, we will present the results on

- **German credit data** (numeric), to classify people described by a set of attributes as good or bad credit risks. The dataset consists of 1000 instances with 20 features each.
- **Ionosphere**, to classify as "good" or "bad" the 351 values returned by radars from the ionosphere.
- **Pima Indians Diabetes**, to detect a possible diabetes disease from 768 patterns with 8 features each.
- **Mammographic mass**, to discriminate between benign and malignant mammographic masses, given 961 instances and 6 attributes each.
- **Heart (statlog)**, composed of 370 instances with 13 features to predict the presence or absence of a heart disease.
- **Weaning**, to decide if 302 instances of patients are ready or not for weaning [1].
- **Wisconsin breast cancer** to classify benign or malign cancer, observing 699 clinical records of breast cancer, described by 10 multivariate features.
- **Respiratory data set**, to categorize two respiratory distress syndromes in 85 cases with 17 attributes each.

We tested 50 linear binary classifiers, implemented as "l1" with Matlab Spider libraries [23], trained with a Bagging procedure with up to 80 samples randomly drawn from one third of each dataset, and fused by mean rule. The remaining two thirds are used for validation. One round of cross-validation (randomly) partitions the data set into two (complementary) subsets, and allows performing the training on 1/3 of the data and the validation (= testing) on the remaining 2/3. Traditionally, 2/3 of the dataset is used for training, and the remaining 1/3 for testing. We have chosen the inverse 1/3 2/3 proportions, which give us more room for starting with a relatively small number of training samples and test extensively the proposed approach, which is based on incremental F-score computation and re-ranking of the classifiers. We have chosen bagged linear classifiers to have them the more different as possible, and to allow them to commit some errors in order to study the diversity impact on the ensembles.

To reduce variability, multiple rounds of cross-validation are performed using different partitions, and the validation results are averaged over the rounds. We then repeated the process ten times (10-fold).

To compare the fusion results with the proposed technique, we applied the F-score based ranking to the previous set in order to select 40, 30, 20 and 10 experts, we fused them with the mean rule (Eq. (1)) and we analyzed the outcomes. The accuracy is measured as the number of correct predictions with respect to the the true value out of the total samples number. Precision, recall and F-score are measured as per (11) and (13), averaging their value on the ensemble to obtain a scalar.

[1] *http : //www.bangor.ac.uk/ ~mas00a/activities/real_data.htm*

We calculated the accuracy and the diversity of the different classifiers sets, comparing eight different diversity measures and repeating the experiment in a 10-fold cross-validation to average the results across different test patterns. The diversity measures used are Q statistics (later in the text referred as "Q"), disagreement, double fault measure (DF), Kohavi-Wolpert variance (KW), Interrated Agreement (IA), Kuncheva's entropy (K-Entropy), Generalized Diversity (GD) and Coincident Failure Diversity (CFD). Q varies in $\{-1, +1\}$ and is zero for independent classifiers, while K-Entropy, GD, CFD vary between $0, 1$ and the maximum diversity occurs when the value is one. In the results, we highlighted in boldface the most significant diversity value, that is the highest in the case of Disagreement, KW, K-Entropy, GD and CFD, while it is the lowest for Q, Df and IA.

4.2. Discussion

We summarized the results in several tables, which describe the diversity and the accuracy values and present the average precision, recall and F-score measures for the whole set (fusion) of 50 classifiers, and for the selection, obtained by taking 40, 30, 20 and 10 classifiers from the full pool by F-score ranking, as described in the previous sections.

As a general trend, we can observe that the selection is always improving the classification accuracy with respect to the full pool fusion; using the F-score as a means to carefully pick classifiers resulted in a boost to the classification performances, with peaks up to 25% (8).

Another interesting fact is the relationship between performance and diversity, which suggests that accuracy and diversity are related, as already declared in the literature, emerges from the experiments, since a slight tradeoff exists in most of the cases. However, it is not always the case in our experiments that when the accuracy rises the diversity drops; on the contrary, as an important achievement, we may say that the selection by F-score measure does not always impact on the heterogeneity of the pool (see, for instance, Tables 1,2,5,6,13,14). In many datasets, high F-score values, associated with high accuracy measures, do show positive values of the most of the diversity indicators. This can be seen analyzing the data in Tables 5,7,13, where the random selection sets (indicated as "Random") achieve slightly different diversity values with respect to the selection sets formed by F-score. These, however, yield much higher accuracies with respect to the random choice, as shown in Tables 6,8,14, and even with respect to the fusion of the entire classifier set.

Another observation we can draw is that the measures do not often agree; they assume different values on the three datasets and among the different ensembles. At the same time, the highest accuracy (Tables 6,8,14) is not always obtained by the ensemble that is indicated by most of them as the most diverse. This can be explained by two reasons: the first is that there is a tradeoff between accuracy and diversity, empirically demonstrated for classification [31] and formally defined for regression [6], that implies that there could be no simultaneous maximization of the two. Second, as said before, the diversity measures catch two different aspects of the ensemble: some of them

	Classifiers	Q	Disagreement	Double-fault	KW	IA	K-Entropy	GD	CDF
Fusion	50	0.6338	**0.4523**	0.2620	**0.2216**	**0.0934**	**0.7165**	0.4644	0.5223
Selection	40	**0.4278**	0.3803	0.1682	0.1863	0.1675	0.5652	0.5333	0.6550
Selection	30	0.4956	0.3271	0.1506	0.1603	0.2358	0.4834	0.5208	0.6837
Selection	20	0.5158	0.2853	0.1142	0.1398	0.2449	0.4161	0.5548	0.7174
Selection	10	0.5655	0.2589	**0.0990**	0.1269	0.2462	0.3835	**0.5661**	**0.7208**
Random	40	0.6676	**0.4523**	0.2882	**0.2216**	**0.0897**	**0.7238**	0.4396	0.4954
Random	30	0.6813	0.4430	0.2874	0.2171	0.0978	0.6957	0.4369	0.5008
Random	20	0.6366	0.3874	0.2239	0.1898	0.1471	0.5928	**0.4780**	0.5992
Random	10	**0.5585**	0.3779	**0.2136**	0.1852	0.1640	0.5602	0.4696	**0.6215**

Table 1: Diversity values for the German-numeric dataset. The highest diversity values are highlighted in boldface. As the reader can observe, the different diversity indicators do not agree, promoting two very different classifier sets (fusion of 50 members vs selection of 10), and suggesting that there is no unique and standard definition.

Combination Rule	Nr Classifiers	Precision	Recall	F-measure	Accuracy
Fusion	50	0.8503	0.6624	0.7425	0.6787
Selection	40	0.7962	0.8301	0.8109	0.7297
Selection	30	0.7962	0.8244	0.8100	0.7290
Selection	20	0.7510	0.8607	0.8020	0.7087
Selection	10	0.8094	0.8800	0.8432	**0.7665**
Random	40	0.7119	0.4882	0.5792	0.5015
Random	30	0.7186	0.4989	0.5863	0.5120
Random	20	0.6681	0.5636	0.5993	0.5128
Random	10	0.7107	0.5568	0.6242	**0.5225**

Table 2: German numeric dataset results for single classifier on average (first row), fusion and selection of 40, 30, 20 and 10 classifiers. In the last column, the accuracy is presented. The selection of a number of classifiers gives a boost to the classification.

	Classifiers	Q	Disagreement	Double-fault	KW	IA	K-Entropy	GD	CDF
Fusion	50	0.8898	**0.2673**	0.1569	**0.1310**	**0.3351**	**0.3499**	**0.4688**	**0.7240**
Selection	40	**0.8798**	0.1531	**0.1337**	0.0750	0.5341	0.2091	0.3662	0.6542
Selection	30	0.9027	0.1431	0.1502	0.0701	0.5835	0.2019	0.3229	0.5797
Selection	20	0.9063	0.1480	0.1668	0.0725	0.5941	0.2054	0.3037	0.5573
Selection	10	0.9369	0.1137	0.1480	0.0557	0.6427	0.1710	0.2777	0.4671
Random	40	0.8941	**0.2791**	0.1573	**0.1368**	0.3131	0.3646	**0.4799**	0.7196
Random	30	0.8980	0.2310	0.1543	0.1132	0.4121	0.3049	0.4261	**0.7516**
Random	20	**0.8471**	0.2268	**0.1526**	0.1111	0.4246	0.3227	0.4064	0.6301
Random	10	0.9074	0.2767	0.1685	0.1356	**0.3071**	**0.3943**	0.4394	0.6237

Table 3: Diversity values for the Ionosphere dataset. Here are compared fusion and selection results with a decreasing number of classifiers. We can observe that the trade off between accuracy and diversity is here depicted; when increasing one, the other drops.

Combination Rule	Nr Classifiers	Precision	Recall	F-measure	Accuracy
Fusion	50	0.8279	0.9698	0.8917	0.8438
Selection	40	0.8174	0.9828	0.8914	0.8409
Selection	30	0.8614	1.0000	0.9256	0.8949
Selection	20	0.7868	1.0000	0.8806	0.8381
Selection	10	0.8940	0.9916	0.9402	**0.9148**
Random	40	0.6579	0.7802	0.7128	**0.5881**
Random	30	0.6507	0.7696	0.7051	0.5795
Random	20	0.5631	0.6952	0.6219	0.4972
Random	10	0.6580	0.7269	0.6897	0.5597

Table 4: Accuracy and precision/recall results for the Ionosphere dataset. The accuracy of the single classifier (on average) is presented on the first row, while fusion is following together with the selection results with a decreasing number of classifiers. A smaller set of classifiers seems to work better in this case.

calculate the relationship between couples of classifiers, while others focus on describing the agreement of the ensemble from

	Classifiers	Q	Disagreement	Double-fault	KW	IA	K-Entropy	GD	CDF
Fusion	50	0.6453	0.4359	0.4324	0.2136	**0.0386**	0.6810	0.3368	0.3569
Selection	40	**0.4458**	**0.4580**	0.3310	**0.2244**	0.0595	**0.7383**	0.4121	0.4484
Selection	30	0.5786	0.4180	0.2848	0.2048	0.1428	0.6406	0.4279	0.5154
Selection	20	0.4934	0.3800	**0.1885**	0.1862	0.1416	0.5712	**0.5291**	**0.6321**
Selection	10	0.8355	0.2453	0.2386	0.1202	0.4325	0.3560	0.3452	0.5567
Random	40	0.6310	0.4156	0.4688	0.2036	0.0363	0.6292	0.3090	0.3282
Random	30	0.6514	0.3870	0.5011	0.1896	0.0522	0.5821	0.2812	0.3060
Random	20	**0.6186**	**0.4670**	**0.3447**	**0.2288**	**-0.0079**	**0.7546**	**0.4064**	0.4248
Random	10	0.7427	0.3793	0.3573	0.1859	0.1384	0.5687	0.3489	**0.4494**

Table 5: Diversity values for Pima Indians Diabetes dataset. The diversity values for the 40-elements ensemble suggest that there is a low redundancy among members selected by F-score. In this case, being accurate does not imply being not diverse.

Combination Rule	Nr Classifiers	Precision	Recall	F-measure	Accuracy
Fusion	50	0.8559	0.2138	0.3421	0.7044
Selection	40	0.6910	0.6087	0.6432	**0.7565**
Selection	30	0.5913	0.7338	0.6476	0.7109
Selection	20	0.4696	0.9173	0.6105	0.5951
Selection	10	0.4913	0.8407	0.6179	0.6328
Random	40	0.4869	0.0725	0.1261	**0.6393**
Random	30	0.3444	0.0360	0.0647	0.6276
Random	20	0.3895	0.1457	0.1868	0.6211
Random	10	0.3579	0.3111	0.3279	0.5612

Table 6: Pima Indians diabetes dataset results for single classifier on average (first row), fusion and selection of 40, 30, 20 and 10 classifiers. In the last column, the accuracy is presented. For this dataset a larger classifier ensemble seems to work better, having the highest accuracy with 40 elements.

	Classifiers	Q	Disagreement	Double-fault	KW	IA	K-Entropy	GD	CDF
Fusion	50	0.7249	0.4242	0.4533	0.2079	**0.0473**	0.6689	0.3188	0.3414
Selection	40	**0.5932**	**0.4494**	0.3541	**0.2202**	0.0716	**0.7135**	0.3883	0.4289
Selection	30	0.5938	0.3981	0.2405	0.1951	0.1652	0.6232	**0.4525**	**0.5502**
Selection	20	0.8611	0.2836	0.2525	0.1390	0.3927	0.4249	0.3609	0.4893
Selection	10	0.8686	0.2764	**0.2344**	0.1354	0.3683	0.4127	0.3770	0.5279
Random	40	**0.7370**	0.4286	0.4323	0.2100	0.0499	0.6809	0.3324	0.3590
Random	30	0.7806	0.4229	0.4208	0.2072	0.0451	0.6402	0.3402	0.3714
Random	20	0.7662	0.3936	0.4718	0.1929	0.0233	0.5748	0.3011	0.3266
Random	10	0.7758	**0.4730**	**0.2579**	**0.2318**	**-0.0564**	**0.7877**	**0.4892**	**0.5129**

Table 7: Diversity values for different selection sets on the Mammographical mass dataset. The highest diversity values are highlighted in boldface.

Combination Rule	Nr Classifiers	Precision	Recall	F-measure	Accuracy
Fusion	50	0.8333	0.4602	0.5929	0.6478
Selection	40	0.8732	0.6307	0.7324	0.7911
Selection	30	0.7359	0.8243	0.7669	0.7661
Selection	20	0.7540	0.8813	0.8126	**0.8150**
Selection	10	0.7129	0.8645	0.7762	0.7755
Random	40	0.7254	0.1307	0.1709	**0.5364**
Random	30	0.2184	0.1554	0.1816	0.5177
Random	20	0.2203	0.1187	0.1543	0.5301
Random	10	0.4719	0.3435	0.3489	0.5281

Table 8: Accuracy results on the Mammographical mass dataset for ensembles of decreasing number of classifiers. Combined with Table 7, it reinforces the idea of a tradeoff between accuracy and diversity.

an overall perspective. In any case, a rigorous demonstration of how the F-score promotes the diversity in classifier ensembles is out of the scope of this paper.

In some cases, only 10-20 selected classifiers score the best performance in f-measure and accuracy. This can be explained considering two facts:

- if many classifiers are redundant (same classifier type trained with similar data) or mistrained (e.g., the training data is biased toward one class), a selection will perform

	Classifiers	Q	Disagreement	Double-fault	KW	IA	K-Entropy	GD	CDF
Fusion	50	0.6650	**0.3491**	0.2650	**0.1711**	**0.2914**	**0.5112**	0.3969	0.5719
Selection	40	0.6777	0.2956	0.2399	0.1449	0.3738	0.4311	0.3829	0.5882
Selection	30	**0.5936**	0.3210	**0.2306**	0.1573	0.3200	0.4708	**0.4104**	**0.5900**
Selection	20	0.7849	0.2459	0.2882	0.1205	0.4768	0.3483	0.3019	0.5261
Selection	10	0.9359	0.1525	0.3463	0.0747	0.6633	0.2153	0.1833	0.3314
Random	40	0.6759	0.3375	0.2562	0.1654	0.3067	0.4946	0.3973	**0.5876**
Random	30	**0.5122**	**0.3858**	**0.2555**	**0.1890**	**0.2093**	**0.5749**	**0.4303**	0.5620
Random	20	0.6664	0.3337	0.2672	0.1635	0.3016	0.4851	0.3867	0.5655
Random	10	0.7965	0.2402	0.3216	0.1177	0.4760	0.3480	0.2714	0.4612

Table 9: Also for the weaning dataset, the highest diversity belongs to the less accurate ensemble.

Combination Rule	Nr Classifiers	Precision	Recall	F-measure	Accuracy
Fusion	50	0.7353	0.8413	0.7848	0.7624
Selection	40	0.7301	0.8846	0.7998	0.7723
Selection	30	0.7270	0.9330	0.8171	0.7995
Selection	20	0.7611	0.8918	0.8200	**0.8119**
Selection	10	0.8192	0.8077	0.8119	0.8069
Random	40	0.5032	0.5769	0.5372	0.4876
Random	30	0.5144	0.5928	0.5501	**0.5347**
Random	20	0.4910	0.5619	0.5229	0.5074
Random	10	0.4902	0.4567	0.4728	0.4752

Table 10: As seen in Table 9, also for the weaning dataset there is a tradeoff between diversity and accuracy.

	Classifiers	Q	Disagreement	Double-fault	KW	IA	K-Entropy	GD	CDF
Fusion	50	**0.9935**	**0.1280**	0.5865	**0.0627**	**0.7174**	**0.1596**	**0.0983**	**0.3566**
Selection	40	0.9936	0.0573	0.5772	0.0281	0.8786	0.0760	0.0473	0.1618
Selection	30	0.9950	0.0482	0.6069	0.0236	0.8941	0.0675	0.0382	0.1234
Selection	20	0.9961	0.0455	0.5821	0.0223	0.8999	0.0619	0.0375	0.1237
Selection	10	0.9960	0.0388	**0.5682**	0.0190	0.9082	0.0551	0.0329	0.0844
Random	40	0.9926	0.1092	0.5793	0.0535	0.7606	0.1366	0.0857	**0.3722**
Random	30	0.9923	0.1258	0.6221	0.0617	0.6888	**0.1697**	**0.0903**	0.3161
Random	20	0.9906	0.0636	0.5745	0.0311	0.8599	0.0877	0.0524	0.1730
Random	10	0.9900	0.0888	0.5596	0.0435	0.7812	0.1161	0.0725	0.2366

Table 11: Diversity values for Wisconsin breast cancer dataset.

Combination Rule	Nr Classifiers	Precision	Recall	F-measure	Accuracy
Fusion	50	0.9270	0.9220	0.9239	0.9421
Selection	40	0.9283	0.9404	0.9337	0.9491
Selection	30	0.9461	0.9104	0.9279	0.9474
Selection	20	0.9163	0.9375	0.9265	0.9456
Selection	10	0.9748	0.9019	0.9368	**0.9544**
Random	40	0.3676	0.3670	0.3670	0.5158
Random	30	0.3532	0.3208	0.3362	0.5298
Random	20	0.3396	0.3462	0.3428	0.5158
Random	10	0.3667	0.3318	0.3481	**0.5333**

Table 12: Breast cancer dataset results for single classifier on average (first row), fusion and selection of 40, 30, 20 and 10 classifiers. In the last column, the accuracy is presented.

better than the full ensemble.

- on the other hand, the cardinality of the domain has also an impact: the bagging technique is proved not to be effective in sets with low number of elements, since it produces very small and similar subsets to train the classifiers. Classifiers learned with similar training sets are redundant, and if the training set is too small, the classifier can be undertrained. Consequently, only a small number of classifiers is effective.

	Classifiers	Q	Disagreement	Double-fault	KW	IA	K-Entropy	GD	CDF
Fusion	50	0.8173	0.2359	0.3293	0.1156	0.5226	0.3330	0.2633	0.5087
Selection	40	0.8496	0.2151	0.3237	0.1054	0.5598	0.3006	0.2488	0.4904
Selection	30	0.7779	0.2532	**0.2861**	0.1241	0.4708	0.3565	**0.3066**	**0.5258**
Selection	20	**0.7538**	**0.2592**	0.2945	**0.1270**	**0.4563**	**0.3722**	0.3049	0.5092
Selection	10	0.9163	0.1568	0.2971	0.0768	0.6437	0.2244	0.2124	0.4158
Random	40	0.8139	0.2358	0.3285	0.1156	0.5207	0.3306	0.2635	0.5064
Random	30	0.6897	0.3018	0.2868	0.1479	0.3789	0.4365	**0.3446**	**0.5565**
Random	20	**0.6772**	**0.3278**	0.3247	**0.1606**	**0.3164**	**0.4881**	0.3360	0.4988
Random	10	0.7439	0.2522	**0.2685**	0.1236	0.4370	0.3728	0.3227	0.5041

Table 13: Diversity values from the heart dataset for fusion vs selection of classifiers drawn from the whole pool.

Combination Rule	Nr Classifiers	Precision	Recall	F-measure	Accuracy
Fusion	50	0.7830	0.8316	0.8063	0.7889
Selection	40	0.7763	0.8368	0.8052	0.7861
Selection	30	0.7932	0.8814	0.8330	0.8083
Selection	20	0.8045	0.8281	0.8158	0.8000
Selection	10	0.8176	0.8711	0.8424	**0.8250**
Random	40	0.5277	0.5579	0.5422	0.5028
Random	30	0.5304	0.5670	0.5464	0.4972
Random	20	0.5619	0.4948	0.5253	0.5278
Random	10	0.5672	0.5619	0.5643	**0.5333**

Table 14: Accuracy and precision, recall and F-score results from the "heart" dataset. In this case, the highest accuracy does not imply lowest values of diversity (Table 13).

	Classifiers	Q	Disagreement	Double-fault	KW	IA	K-Entropy	GD	CDF
Fusion	50	**0.2765**	**0.4043**	0.3977	0.1881	**0.1376**	**0.6139**	0.3470	0.4083
Selection	40	0.3827	0.3831	0.3650	**0.1977**	0.1979	0.5732	**0.3553**	0.4519
Selection	30	0.8817	0.1798	0.4006	0.0881	0.6183	0.2435	0.1925	0.4219
Selection	20	0.7392	0.2485	**0.3038**	0.1218	0.4781	0.3759	0.2860	0.4678
Selection	10	0.7760	0.2564	0.3154	0.1256	0.4409	0.3554	0.2891	**0.4977**
Random	40	**0.3106**	**0.3943**	0.4027	**0.1932**	**0.1535**	**0.5942**	0.3390	0.4073
Random	30	0.8006	0.2438	0.3816	0.1195	0.4641	0.3304	0.2727	0.4917
Random	20	0.5475	0.2949	0.3991	0.1445	0.3587	0.4455	0.2598	0.3881
Random	10	0.5862	0.3393	**0.3076**	0.1663	0.2236	0.5143	**0.3920**	**0.5139**

Table 15: Also in the case of RDS dataset, the diversity values are higher for the less accurate datasets (Table 16).

Combination Rule	Nr Classifiers	Precision	Recall	F-measure	Accuracy
Fusion	50	0.7538	0.4828	0.5613	0.6607
Selection	40	0.8036	0.5862	0.6593	0.7143
Selection	30	0.5792	0.7083	0.6269	0.6518
Selection	20	0.7264	0.7963	0.7592	0.7589
Selection	10	0.8038	0.8654	0.8324	**0.8393**
Random	40	0.4300	0.2931	0.3362	0.4732
Random	30	0.5020	0.6042	0.5389	**0.5804**
Random	20	0.5179	0.3519	0.3870	0.5357
Random	10	0.5165	0.5000	0.4964	0.5625

Table 16: Accuracy for several classifier ensembles on the RDS dataset. In general, the selection by F-score has better performance than the fusion of the full classifiers pool, and than the randomly formed ensembles.

We decided to split the results into individual dedicated tables, to present and comment the results in detail, highlighting that the method works significantly better for improving accuracy, but is not always guaranteed to preserve the diversity. We can observe that in some of the datasets the diversity is not diminished, but in some others the compromise between accuracy and diversity can be perceived.

Table 17 and Table 18 provide instead a summary of the findings averaged over all datasets. We normalized each value by a factor that considers the number of total samples, thus weight-

	Classifiers	Q	Disagreement	Double-fault	KW	IA	K-Entropy	GD	CDF
Fusion	50	0.7229324	0.364998302	0.358010273	0.1786713	0.2177219	0.557193	0.3492194	0.4603597
Selection	40	0.6230744	0.336695437	0.293208884	0.1651576	0.2824598	0.5142377	0.3773269	0.5046003
Selection	30	0.6539402	0.310079894	0.254807716	0.15195	0.32608	0.4650826	0.3949907	0.5446884
Selection	20	0.7101704	0.266458311	0.236832892	0.1305718	0.3852162	0.3912897	0.3913218	0.5503507
Selection	10	0.821047	0.209955754	0.241106812	0.1028703	0.4772937	0.3079584	0.3398989	0.5072013
Random	40	0.7318032	0.361005864	0.364304365	0.17689	0.2194946	0.5500875	0.3418601	0.4549045
Random	30	0.7244863	0.361579078	0.362899317	0.1771755	0.2018598	0.5428053	0.3483006	0.4609183
Random	20	0.7078846	0.352907253	0.337046693	0.1729102	0.2062155	0.5341722	0.3611044	0.4637577
Random	10	0.7399543	0.339880291	0.285892967	0.1665662	0.233773	0.5203848	0.3842826	0.5107142

Table 17: Summary of diversity outcomes averaged on all datasets.

Combination Rule	Nr Classifiers	Precision	Recall	F-measure	Accuracy
Fusion	50	0.82946	0.6219801	0.6765221	0.730289
Selection	40	0.7968174	0.7704365	0.7756557	0.782335
Selection	30	0.7535848	0.8438196	0.7908597	0.777698
Selection	20	0.7263741	0.8868461	0.7889864	0.761201
Selection	10	0.7546138	0.8726134	0.8017182	0.78343
Random	40	0.5988719	0.3709227	0.3985109	0.538415
Random	30	0.4699716	0.3769675	0.3947987	0.540041
Random	20	0.4624363	0.379589	0.3982385	0.536094
Random	10	0.5296547	0.4597489	0.4784186	0.532392

Table 18: Summary of precision, recall, F-score and accuracy values averaged on all datasets.

ing more the larger datasets. From the comparison with the random set results, we can see that the selection brings a significant improvement in performances, both due to increased accuracy, and also because less classifiers means that the ensemble is faster. Analysing Table 18 more in depth, after a certain point the improvement is shown to be not significant; this point can be a candidate for an automatic thresholding the number of classifiers, and it will be subject of our future investigations. Table 17 shows another interesting fact: the diversity of a large selection set is not very far from the disagreement among the random set members. The diversity starts to deteriorate when the F-score has no significant improvement; until then, its values are comparable with those of the random set.

The sum-up table seems to suggest a trend in which recall is favoured at the expense of precision as the number of classifiers decreases; this trend is not always confirmed in the experiments, and it will be subject of future analysis. In general, this behaviour can be explained with the fact that separability issues in the data arise at some points, when the classifiers are too few. Precision and recall are in fact related to -respectively- False Positive and False Negative errors, which are a trade off when classifying a non-separable dataset: privileging the classification rate for the positive class often implies decreasing the False Negatives and augmenting the True Positives, but also raising the False Positives, since more negative samples are classified as positive. The lesser classifiers we have in the ensemble, the more they seem to face separability problems between classes; empirically, a bigger ensemble provides richer and sharper boundaries to divide false and negative samples. However, no matter the size, any ensemble can comprise redundant members, thus it is important to select them accurately by diversity means.

To draw a conclusion, we can say that the results show that the F-score ranking-based is an valid approach to select classifiers, since it presents higher accuracy in classifying the test samples from the UCI datasets if compared to the full pool (fusion), and also to randomly selected sets. At the same time, it maintains a good level of diversity among the members of the ensemble which is generally associated to better generalization capability [27].

5. Application to video tracking

Video tracking is the process of locating an object in a scene through time. A robust and accurate tracking process is the fundamental step that precedes high-level reasoning for scene understanding and situation assessment, as following an object constitute a piece of information from which semantics and events can be inferred. To track an object that freely moves in a scene different techniques, developed in the last few years, can be applied [35]. However, no definitive solution has been proposed, and tracking remains a challenging task, especially when illumination and appearance variation occur in unconstrained video sequences.

5.1. Tracking as a classification problem

When treating tracking as a classification problem, a single classifier or classifier ensemble aims at separating the target from the background in the frames of a video sequence. The problem is reduced to a binary task, and involves the target, labelled as positive, and the remaining part of the image, that is

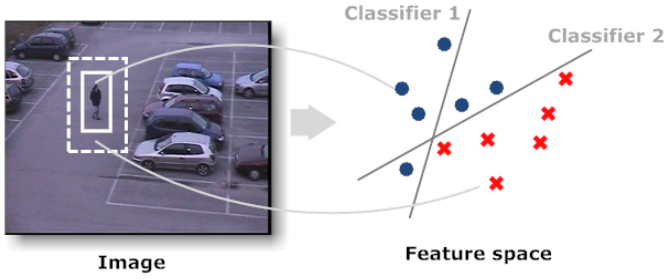

Figure 4: Tracking via classifiers: the components of the object and the background are projected in the features space, where the classifiers operate.

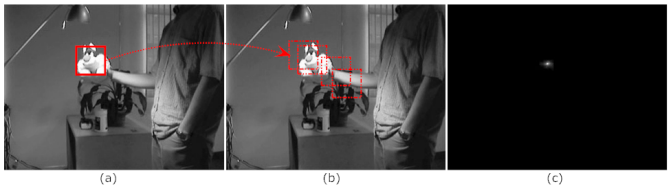

Figure 5: Tracking via classifiers: (a) At time t the model of the object is built (b) At time $t+1$ the template is applied to a dense region around the previous object location (c) The confidence map is drawn from every classifier output. The source images are taken from [49]

labelled as negative. Each low-level feature (color histograms, Haar wavelets, LBPs, HOGs, etc.) can be associated with a simple Bayes classifier, which chooses in a Bayesian fashion the right boundary for labelling a sample positive or negative according to the feature's outcome and previously learned margins. The idea, pioneered by Avidan in [2], relies on training a binary classifier, in his case a support vector machine (SVM), to distinguish a given object in the scene from the background and follow it into subsequent frames. More recently, the tracking via classification concept has received a boost [3, 8, 18, 45, 53, 38], as it is considered more robust to occlusions and illumination changes [19]. Moreover, greater system robustness and performance is achievable with an ensemble of classifiers through data fusion techniques [8, 45].

The idea is the following: an ensemble of classifiers, or a single strong classifier, is trained with a few samples representing the object (positive samples), and several random background patches (negative samples). While tracking, at each frame the classifier locates the most probable position of the object in the image. The target is, thus, extracted from the foreground and used as a new training sample and the process is repeated for the next frame. In real-time video streams, the frames are processed one-by-one; for each each frame, a considerable number of image patches are tested by the selection ensemble (see Figures 5 and 6), and the one with the highest confidence is chosen as the target (see [2]). It must be noted that the final decision (label) is taken by the selection ensemble, but the so-labelled positive sample, together with a randomly chosen negative background patch, is used for re-ranking ALL the classifiers. The selected ensemble thus influences the ranking of the all the classifiers in the pool, since it provides the next training samples.

The approach can be considered unsupervised, when no human operators intervene in the dynamic labelling process, or, semi-supervised, if we consider that a minimal initial training of the classifiers is provided by human knowledge (or by some other algorithm, for example motion detection [51]).

To each feature is associated a Bayesian classifier, that "interprets" the output of the descriptor and assigns a label to it, as shown in Figure 4. Each Bayesian classifier is trained to discriminate between the same feature extracted both from the foreground and the background. Several *weak* Bayesian classifiers are, then, aggregated and their union provides the classifier ensemble. To find the best combination to separate the data, the F-score is calculated to select the best K classifiers and form a new ensemble. This selection set will search for the object in the next frame, as shown in Figure 5, speeding up the computation. During the search for the target, a confidence map is maintained to indicate the zones of high probability for the object's occurrence.

The *confidence* of the ensemble H on a pattern x can be defined as

$$\mathrm{conf}(H(x)) = \frac{P(\omega = +1)P(H(x)|\omega = +1)}{P(H(x))} \tag{22}$$

If the object is found, the target is used to update the ensemble. In fact, from the labels assigned to the target and to the background patches, it is possible to update π and ρ as per (11). The loop of the process is pictured in Figure 6, adapting to the visual-tracking case the general architecture proposed in Figure 1. With respect to standard trackers, where at each time instant the system needs to recursively estimate and predict the object's state including positions and velocities, and thus the transition from one state to another, the tracking via classification approach differs in the sense that there is no filtering as known in the traditional sense. No predicted position at the next time instant is provided, but only detection and localization (via classification) of the target, frame by frame.

In our case, the object is firstly projected into the features space, to reduce the dimensionality of the data and to speed up the learning process. Several heterogeneous features, as Haar wavelets, LBP and color features, are extracted from the region of the image where the target is present.

Algorithm 2 describes the application of the classifier ensemble to a video tracking task. The selection ensemble $\hat{H}$ (Eq. (21)) is used to search for the object in the frames (*samples selection* module) scanning each frame in a template-matching fashion: each subregion of the video frame F_t is processed by the selection ensemble $\hat{H}$, which returns a confidence on each subregion. This is the most time consuming step, as each subregion of the image has to be processed; with the selection set, only S classifiers are used during this phase, thus saving computational time. As it can be seen in the "search" phase of Figure 6, the source frame is scanned searching for the target in subregions of the same dimension of the target. The remaining part of the loop is the same as described in (1), with the image subregion representing the object used as the ground truth, and a random patch from the background constituting the negative sample. At each frame, the localization of the object strictly depends on the output (confidence) of the ensemble $\hat{H}$. The

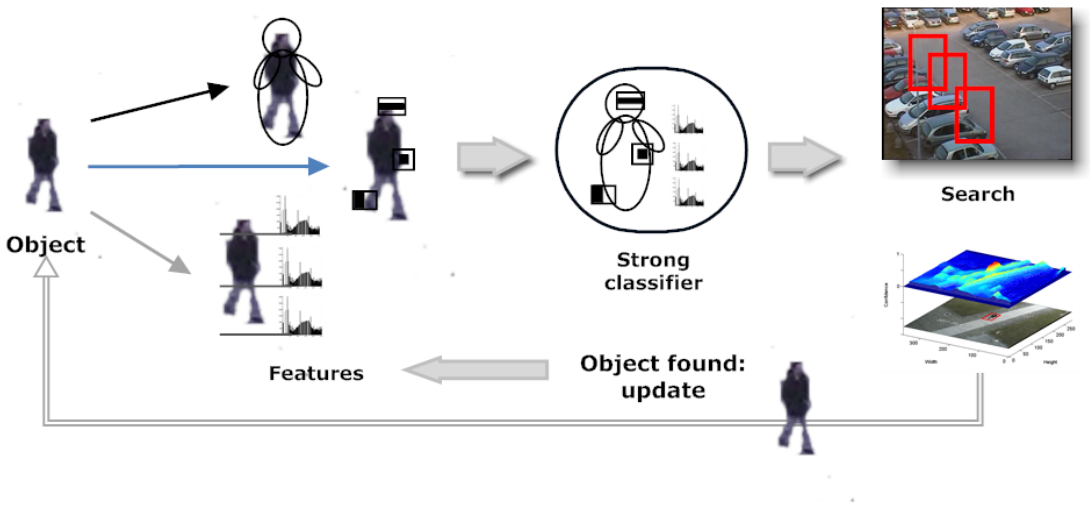

Figure 6: Architecture of the tracking via classification process: the object is decomposed into features, that are used by a classifier to maintain the representation of the target appearance. The classifier searches for the object into a new frame, and, if the object is found, the target model is updated.

Algorithm 2: Tracking via classifier selection

Require: Fixed classifier set H
Require: Randomly initialized selection set $\hat{H}$
Require: Frame F of size $I \times J$
while $x \leftarrow \text{subregion}(F, i, j)$ **do**
  // Test the ensemble $\hat{H}$ on x
  // and save results in a temporary map
  $map(i, j) \leftarrow \text{conf}(\hat{H}(x))$
  // Get the positive sample x_+
  $x_+ \leftarrow \arg\max(map)$
  // Get the negative sample x_-
  $x_- \leftarrow \text{subregion}(map, randX, randY)$
  // Apply Algorithm1 to create
  // a new selection classifier ensemble $\hat{H}$
  $\hat{H} = Algorithm1(x_+, \hat{H}, H, \beta)$
end while

sample x, that has been classified as positive with the maximum confidence by the ensemble $\hat{H}$ at time t, becomes the new positive training sample to test the whole classifiers set. The tracking loop, in fact, behaves like an unsupervised system that searches for a positive sample into a set of possible candidates (patches of the image). At each computation round (at least) one positive sample and one negative counterpart are unsupervisedly chosen from the video stream.

It is important to notice that, as before, the performance of each single classifier will lead to a ranking that will influence all the other ensemble members, and only the selection ensemble will search for the object in the next frame. The selection rule is, thus, required to be fast and accurate to allow a robust real-time tracking, and the F-score measure seems to fit the purpose.

5.2. Experimental setup

In this section we want to study the effect of the F-score ranking on the performance of tracking via classification.

The hardware employed was an AMD Athlon64 3500+ with 1GB of RAM. All the algorithms have been implemented in C++ using optimized structures, i.e. integral images and integral histograms, to reduce the computational requirements. The set-up time for initialization, occuring before the actual tracking is performed, is not included in the computation.

As classifiers, we employed M Naive Bayes classifiers that use maximum a posteriori to decide which class to assign to the pattern x. Every classifier maintains two distributions on the training data, regarding positive and negative samples. Thus

$$\begin{aligned} h(x) &= \arg\max_{\omega_c} P(\omega_c|h(x) \\ &= \arg\max_{\omega_c} \frac{P(\omega_c)P(h(x)|\omega_c)}{P(h(x))} \end{aligned} \tag{23}$$

with $\omega_c \in \{-1, +1\}$. The classes' priors are assumed equiprobable.

5.2.1. Choice of β

Here we discuss how the β parameter in (13) was chosen. After several tests on the CAVIAR[2] sequences, where we bootstrapped a pool of 500 classifiers with only 20 positive hand-labelled samples, the trackers were compared.

In our experiments the pool of classifiers consisted of 500 elements out of which we chose to select 100. In particular, we decided to maintain a static pool of experts without replacing the worst performing classifiers. The rationale is to keep the experimental protocol as simple as possible to show the effectiveness of our solution compared with other methods, but the possibility to remove and substitute the classifiers in the global

[2] http://homepages.inf.ed.ac.uk/rbf/CAVIAR/

Figure 7: Centre image: Trajectories of the F-score tracker when varying β. Yellow: $\beta = 0.5$, red: $\beta = 1$, blue: $\beta = 1.5$, black: $\beta = 2$. Left image: error in pixels on the X coordinate compared with the ground truth. Right: error (in pixels) on the Y coordinate.

pool is a concrete opportunity and it is fully supported by the proposed framework.

In Figure 7 the trajectories of the proposed tracker on the video *Browse1.mpg* when varying β are shown. In the sequence, a man approaches the information point, walks toward the bottom of the scene, and goes back to the leftmost side. As shown in the left and right graphs of Figure 7, compared with the ground truth the most accurate tracker was the one with $\beta = 1.5$. This setting obtained overall good performance on numerous clips of the same dataset. This observation brought us to set $\beta = 1.5$ for the rest of the experiments.

5.2.2. *Tracking algorithms*

In our experiments we tried to consider other similar feature selection/fusion methods that work both in the (online) learning field and the tracking area. The approach can be compared with different tracking algorithms (kernel or model based, particle filters, etc.), but we concentrate to show how our criterion outperforms similar methods.

We decided to compare the proposed method with different fusion and selection approaches. In particular, we tested three different algorithms, referred as

- **PR**: **P**recision/**R**ecall based tracker (proposed solution)
- **OB**: **O**nline **B**oosting based tracker [39]
- **COL**: **COL**our tracker [10]

The Online Boosting algorithm was selected because it is a weighted fusion strategy and can be exploited to linearly combine learning classifiers to track an object. It is a learning-based approach, but has the drawback that it can not swap in and out classifiers, and it is not a selection method but a (weighted) fusion one. To compare the Online Boosting and our technique, that both combine or select members from a pool of classifiers, we kept the number of the pool members fixed at 500; the number of classifiers in the selection set was limited to 100. To guarantee fairness, we performed the experiments using a fixed cardinality ensemble even though our approach could have adapted the number dynamically. Moreover, this fixed threshold facilitates the understanding of the behaviour of the selected classifiers when analysing the swap in/swap out trend.

We employed four different types of features to describe moving objects: Haar features, Local Binary Patterns (LBP), Histograms of Gradients (HOG), and colour histograms. To speed up the search step, we limited the search area to a 50% in excess of the target's dimensions.

The colour tracker [10] selects the best discriminative colour features and uses them to track the target; we have chosen the first 15 (out of 49) most precise features to form the selection. This method (COL) used a selection criterion (variance ratio) to discriminate between features. We used classifiers instead of features, that means that we fused together several heterogeneous features or classifiers at high level, and a fast selection rule that is aimed to save time keeping the performances comparable to similar approaches.

5.3. *CAVIAR sequences*

Tracker	Mean X	Mean Y
PR	3.241	4.445
OB	5.596	2.506
COL	24.126	15.614

Table 19: Average error (in pixels) on the CAVIAR sequence for the proposed approach (PR), the Online Boosting (OB) and the Color tracker (COL).

Tracker	50 feat.	100 feat.	200 feat.	500 feat.
PR	21.03	29.37	38.45	76.94
OB	28.80	32.44	43.05	75.67

Table 20: Application time (in milliseconds) *per frame* on the CAVIAR sequence (Fig.8) for the proposed approach and the Online Boosting tracker.

We used the CAVIAR dataset and the video sequence proposed in [49] to prove the effectiveness of our approach on standard data. Figure 8 shows some frames from the *Fight_RunAway1.mpg* CAVIAR video sequence. In the video, two men meet inside a building, have a brief fight and leave separately. This video represents an interesting case study due to the ambiguity caused by the two men with similar appearance. The video comprises 552 frames at 384×288 pixels resolution. The target was initialized at frame 267 with a change detection

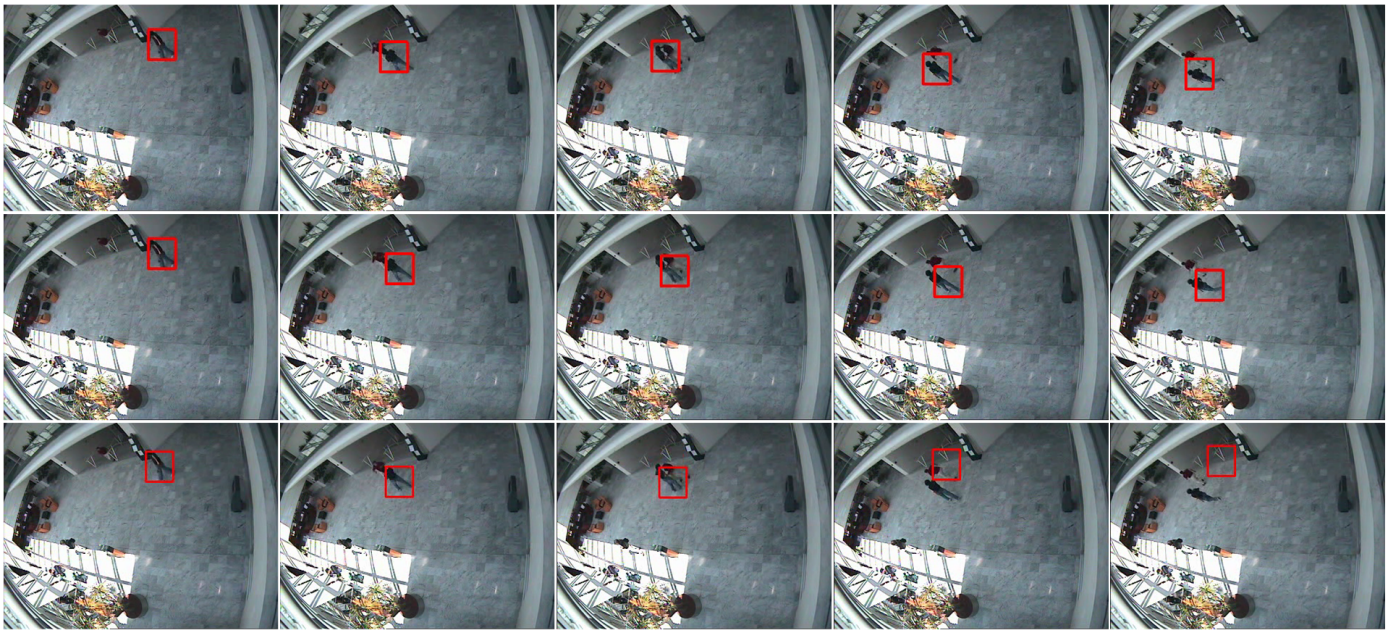

Figure 8: Comparison of output frames taken from PR (top row), Online Boosting (second row), and colour (bottom row) trackers run on the same video sequence.

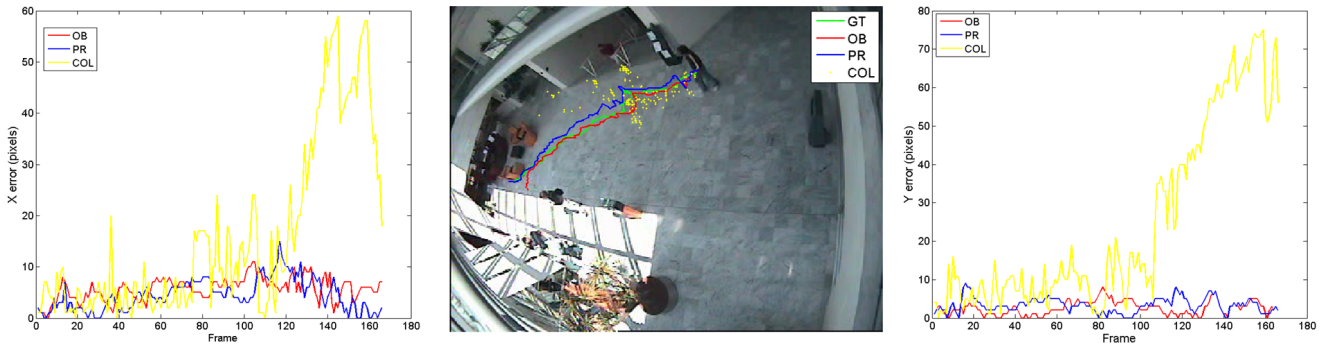

Figure 9: Centre image: Trajectories of the F-score tracker (said PR, blue), Online Boosting tracker (OB, red), and Colour tracker [10] (Col, yellow) compared with ground truth (GT, light green). Left image: error in pixels on the X coordinate for the previous trackers compared with the ground truth. Right: error (in pixels) on the Y coordinate.

algorithm. In this case no bootstrapping was required: in the first frame where the target appears, a model of the foreground is built using random features. As already discussed in Section 5.1, the training procedure at time t uses unsupervised samples coming from the search phase performed at time $t-1$.

The output of the proposed technique is shown in the first row of Figure 8 where the target is correctly tracked even when the two men are very close, without drifting. In this sequence, the difference in the illumination conditions and in the target's appearance can be critical conditions for colour histograms. In fact, the Colour tracker drifted after the men's collision (bottom row), while the Online Boosting detector (second row of Figure 8) correctly followed the target, exploiting other features as shape and texture. Figure 9 shows the errors with respect to the ground truth. The average shift in pixels from the ground truth for the PR tracker, the OB method and the Color tracker is presented in Table 19. The PR and OB trackers obtained similar results with a slight advantage of the former, while the Color tracker's drift resulted in higher average error with respect to the ground truth. Table 20 presents the average application time for the aforementioned trackers on the CAVIAR video sequence. The Colour tracker took an average of 136.67 msecs to process a frame. In the case of the PR and OB trackers, as we can see from Table 20, the time of computation strictly depends on the number of classifiers considered; when the classifiers amount included in the selection set is strictly less than the pool cardinality, the proposed approach outperforms the others. Remember from Section 5.2.2 that the number of classifiers used for tracking was 100 out of a pool of 500.

Figure 10 shows the number of classifiers used per each feature type along the frames of the CAVIAR video sequence. Since the ensemble is randomly initialized, in the very first frames the ensemble undergoes significant changes. The subsequent behaviour (i.e. after frame 10) shows a preference for

Figure 11: Centre image: trajectories of the target followed by the F-score tracker ('PR', white) on the parking lot sequence and by the Online Boosting tracker ('OB', red) compared with the ground truth (GT, light green). Left image: error in pixels on the X coordinate. Right: error on the Y coordinate.

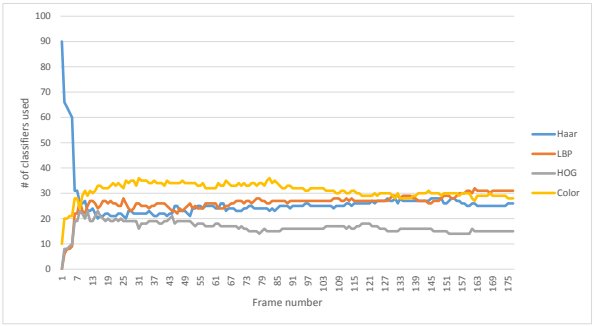

Figure 10: Number of classifers used (per feature type) on the CAVIAR sequence.

Tracker	50 feat.	100 feat.	200 feat.	500 feat.
PR	17.52	26.34	33.52	68.14
OB	22.46	30.11	38.18	67.59

Table 21: Comparison of computation time (in milliseconds *per frame*) of the F-score based tracker (PR) and the Online Boosting algorithm (OB) on the Sylvester sequence.

Tracker	Mean X	Var X	Mean Y	Var Y
PR	0.947	2.262	0.427	0.423
OB	1.948	5.288	0.574	0.601

Table 22: Error mean and variance (in pixels) on the Sylvester sequence for both Online Boosting (OB) and the proposed approach (PR).

the classifiers based on colour features in the first half of the sequence, while this preference loses ground in the second half in favour of non colour-based classifiers, in particular LBP ones. This turns to be supporting the actual content in the video, since the second half of the sequence shows the fight and thus the occlusions of the tracked person with the attacker. While in the first half colour was determinant for discriminating the target from the background, in the following frames this less significant.

5.4. Sylvester sequence

In the second experiment, we tested three trackers on the Sylvester black and white video sequence [49]. The video comprises 1340 frames at 320×240 pixels resolution; the content of the sequence is a puppet that is moved under a light bulb, suddenly changing pose and illumination. The target was manually initialized in the first frame on the puppet's muzzle with a 40×40 pixels square.

Our approach has been compared with the OB tracker only since in this case Haar, LBP and HOG features only could be used. The statistics on the error consider the shift in pixels with respect to the provided ground truth; from the first and third graphs of Figure 12, we can see that the most troublesome part of the sequence is from the half till the end of it, where the puppet starts to rotate and changes very rapidly in its pose and the error increases. The error is expressed as the distance in pixels between the center of the detected target and the ground truth. Ground truth location for the center of the bounding box is considered to be the puppet's nose. Example trajectories of the center of the bounding box are shown in the central images of Figure 12. The error means with their variance (in pixels) for both the OB and PR trackers are presented in Table 22.Considering the computation time for both algorithms (Table 21) and their performance (Table 22), we see that the proposed solution is faster and slightly more accurate on this sequence. It depends on the fact that our method cuts the classifier number, while the Online Boosting fuses the classifiers without discarding any, but simply reducing their weight in the linear combination. The selection instead picks a small subset with the most performing classifiers, maintaining the accuracy as high as possible, saving at the same time a lot of computation time.

Figure 13 shows several significant frames of the video sequence (bottom row) along with a trace of the classifiers used by the proposed PR tracker (top row). The trace of the has the classifiers ordered by feature type: Haar from 1 to 166, LBP

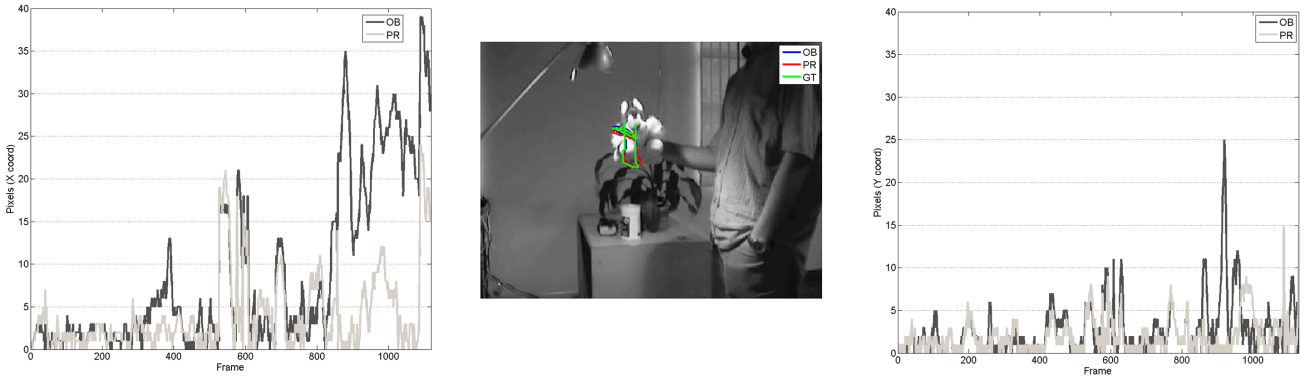

Figure 12: Centre image: example of trajectories of the F-score tracker ('PR', blue) and Online Boosting tracker ('OB', red) compared with the ground truth (GT, light green). Left image: error in pixels on the X coordinate. Right: error on the Y coordinate.

from 167 to 333 while remaining is composed of HOG features.The chart in middle row of the Figure shows the Hamming distance between subsequent columns of the trace graph. This means that the Hamming distance is calculated on two binary vectors indicating which classifiers in the pool are in use in two subsequent frames. The greater the Hamming distance, the greater the number of classifiers changed between two subsequent frames. A change in the composition of the ensemble indicates that (as per Algorithm 1) the selection process has found more promising classifiers to better detect the target, and is discarding some of those used previously. This can be shown also in Figure 14 where the percentage of swapped classifiers is shown per feature type. Starting from a random selection, the highest swapping activity is at the beginning until a suitable enough configuration is found. It is interesting to note the activity of the graphs in Figure 13 and Figure 14 along with those of Figure 12: the algorithm seems to adapt to the changes in the target modifying the ensemble accordingly.

5.5. Parking lot

We acquired a 251 frames video sequence from the top of a building in our university campus at 360×288 pixels resolution. The difficulty of this sequence is constituted by the ambiguity generated by the appearance of a pedestrian wearing black clothes that walks occluded by cars of similar colour.

We can say that in this case only colour histograms are weak decision makers, because the foreground (target) and the background (car) have a very similar colour. Other features, like contours, edges or shape based, could perform better, and their fusion could improve the classification.

As we can see in Figure 15, the Mean Shift and the colour features based trackers get stuck on a region of the background (and for this reason their accuracy results are not numerically shown in the following). On the contrary, the proposed technique and the Online Boosting successfully follow the target while the critical occlusion occurs. This is also confirmed in Figure 11 where the F-score tracker ('PR') based on a selection of 100 classifiers out of 500 and the Online Boosting tracker ('OB') formed by 500 fused classifiers are compared with the

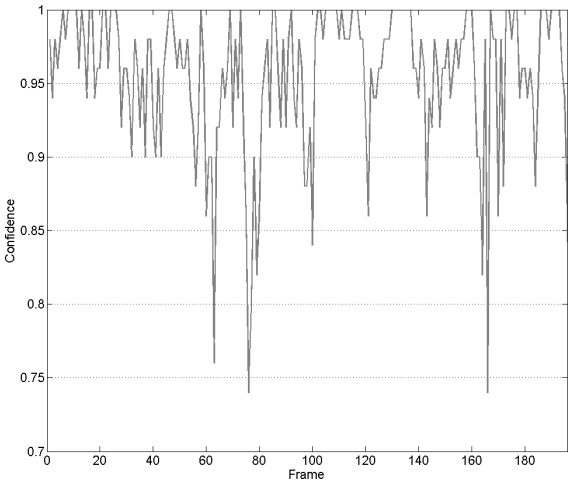

Figure 16: Confidence of the selection ensemble in the parking lot sequence. The salient points are when the pedestrian approaches the black cars (around frame 60), and crosses the dark area. The system recovers quickly, forcing the classifiers to learn the new situation.

ground truth (GT) coordinates manually labelled. The errors in pixels on the X and Y coordinates are shown in the two charts. Their statistics can be found in Table 24 where the mean and variance of both algorithms are provided, confirming that the two approaches are comparable and similar in performance, even though the proposed PR-tracker is using $1/5$ of the classifiers used by the OB tracker. Of course, this has an impact on classification speed as we can see from Table 23 indicating better results for our approach. We tested the application time for the OB algorithm on a progressively augmenting set of classifiers, and for the F-score based ensemble with an increasing number of selected elements, drawn from a fixed size set of 500 classifiers. When the two algorithms work on the same set size (that is the cardinality of the selection set matches the pool's size, therefore no selection is performed), PR is clearly slower due to the overhead given by the selection step. The

Tracker	50 feat.	100 feat.	200 feat.	500 feat.
PR	21.09	29.44	38.52	77.01
OB	29.96	33.60	44.22	76.84

Table 23: Application time (in milliseconds *per frame*) of the proposed technique (PR) compared with the Online Boosting (OB) on the sequence of Fig.15.

Tracker	Mean X	Var X	Mean Y	Var Y
PR	1.088	0.520	3.251	2.459
OB	2.018	1.532	1.795	2.032

Table 24: Error means and variance (in pixels) on the parking lot sequence (Figure 11) for the F-score based tracker (PR) and the Online Boosting algoritm (OB).

result is presented only to show the computational overhead of the selection procedure. The colour tracker and Mean Shift took 131.011 and 47.57 milliseconds per frame respectively to process the sequence of Figure 15.

In Figure 18, we see the percentage of swapped classifiers between pairs of frames. After the initialization phase and corresponding high activity, ensemble changes can be seen while the target is undergoing partial occlusion.

In Figure 17 the trace of the selection of the 100 classifiers from a pool of 500 is displayed for this sequence. The salient points are marked with the correspondent frame and a red line that indicates the time. Comparing this graph with the confidence one (Figure 16), we notice that in the frames where the pedestrian approaches the black cars (around frame 60), the confidence suddenly decreases. In the subsequent instants the system recovers quickly, forcing the classifiers to learn the new appearance and picking the most suitable features for the classification.

6. Conclusions

The novelty of the paper is focused on the use of the F-score measure as a means to select classifiers of an online trained ensemble. The F-score served as a selection rule to discriminate, without weights adjustments, between several classifiers that employ heterogeneous features. On standard datasets, we have observed a general improvement in classification accuracy and a general tendency in redundancy reduction among the members of an f-score optimized ensemble, this could hint at a way to obtain both accurate and diverse ensembles.

Also, this new technique allows the fast application of a small number of selected classifiers for real-time applications such as target tracking for video surveillance, where the proposed approach achieves an improvement in terms of speed and accuracy with respect to similar state-of-the art algorithms on both standard and real-world video sequences.

As future work, the study of an expanded version of the accuracy-diversity balance, which is a three-way tradeoff involving bias, variance and covariance is a natural follow-up to provide another means for describing the ensemble performance.

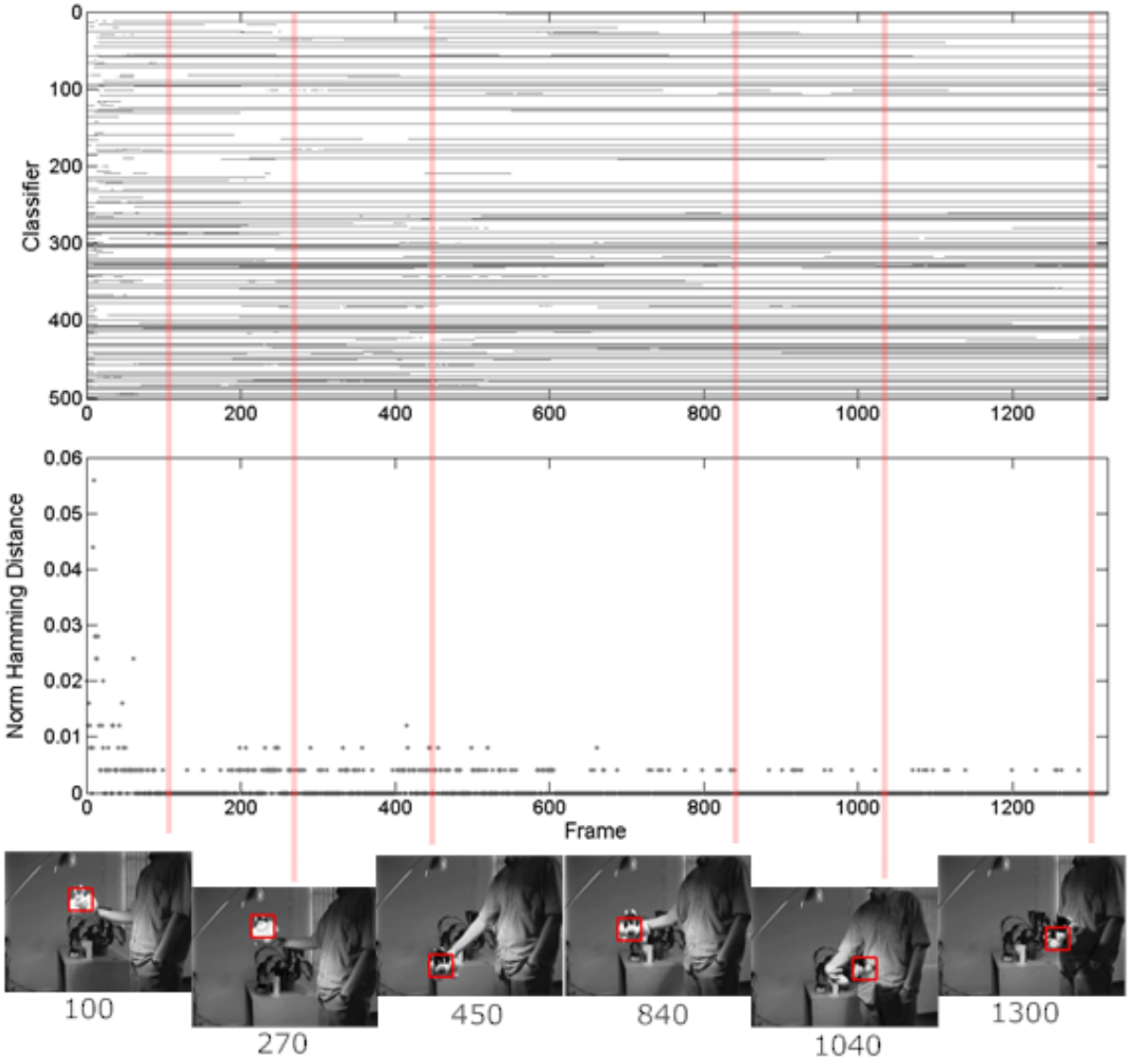

Figure 13: The top row shows the trace of the classifiers used on the Sylvester video sequence [49]. The middle row shows the normalized Hamming distance between subsequent frames, while in the bottom row some salient points are described by the correspondent video shot.

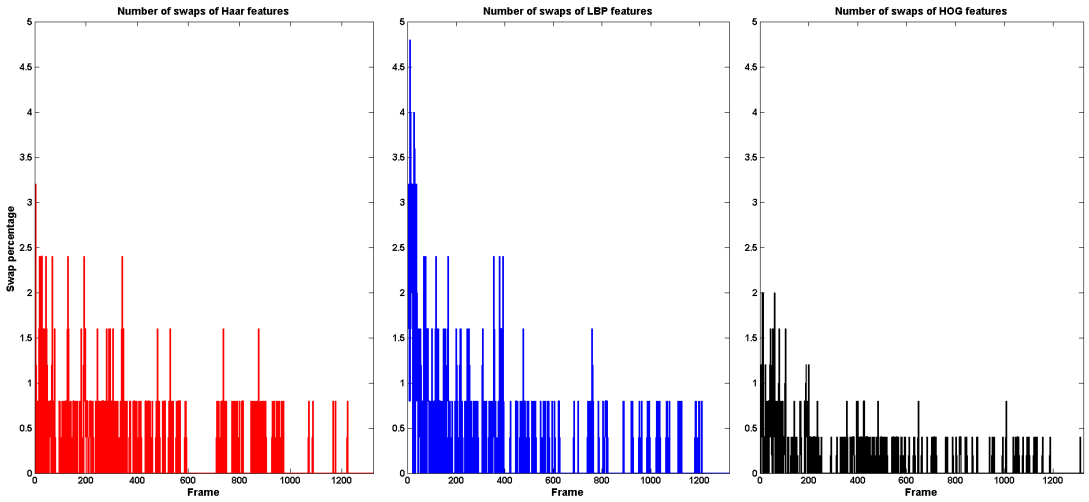

Figure 14: Sylvester sequence. Percentage of Haar (left), LBP (center), and HOG (right) features swapped per frame. The activity in the graphs can be put in relation with the errors shown in Figure 12.

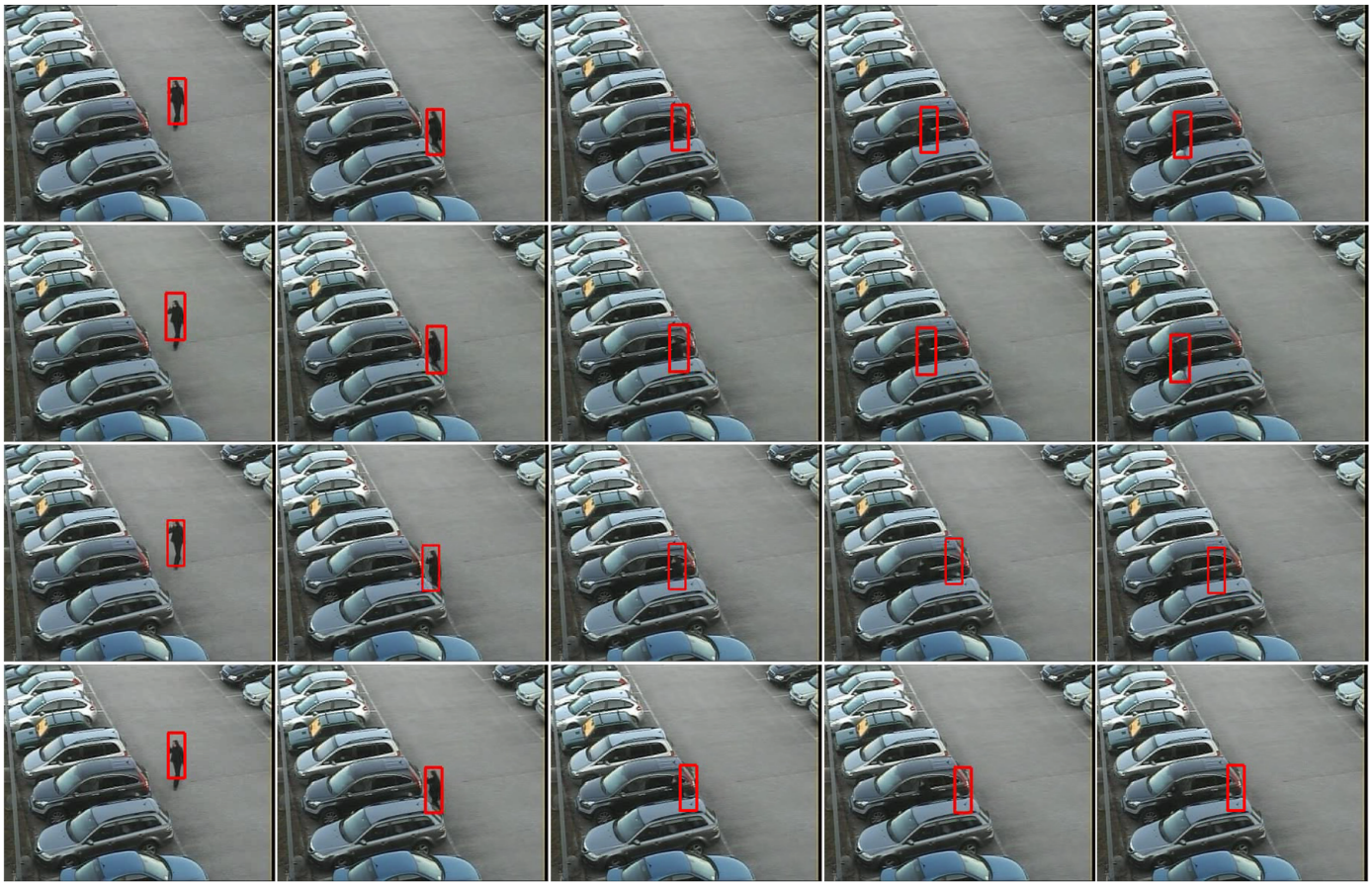

Figure 15: Comparison of the accuracy of the proposed tracker (first row), Online Boosting (second row), Colour based tracker (third row), and Mean Shift (bottom row).

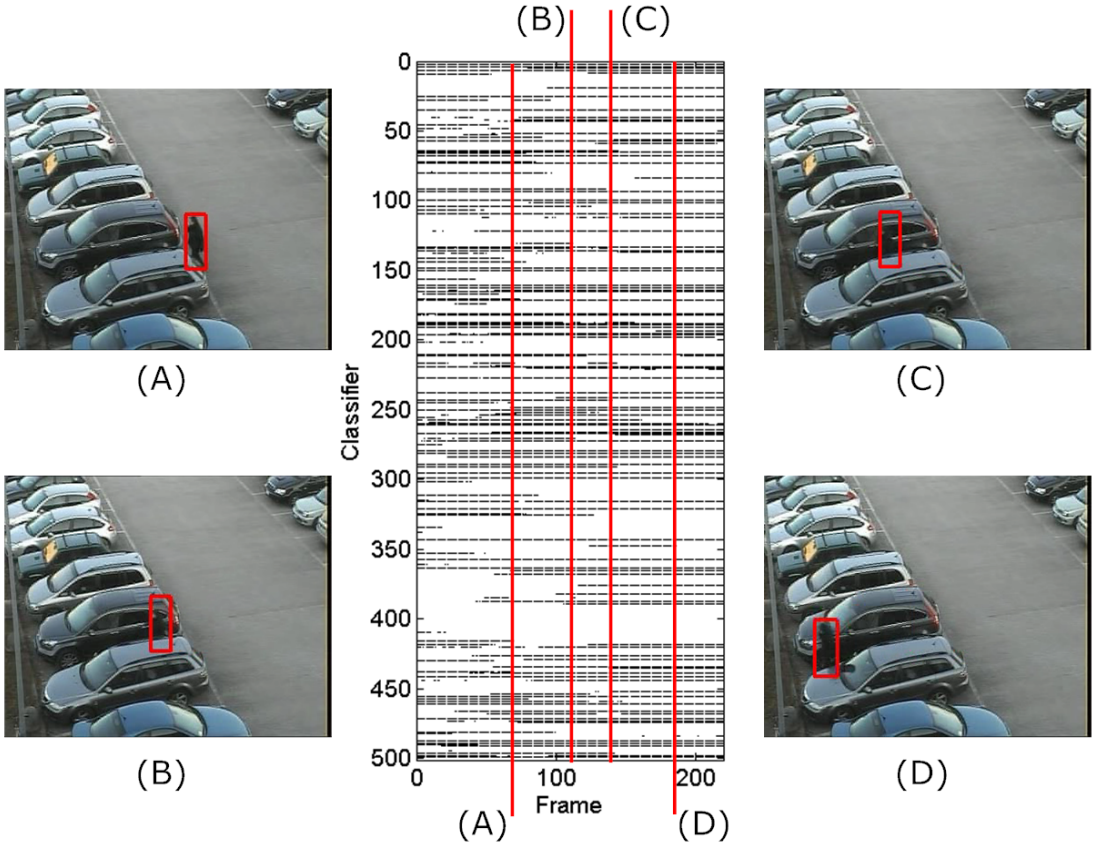

Figure 17: Classifiers trace on the video sequence showing a pedestrian occluded by cars with similar colour. The salient points are highlighted by the correspondent frames.

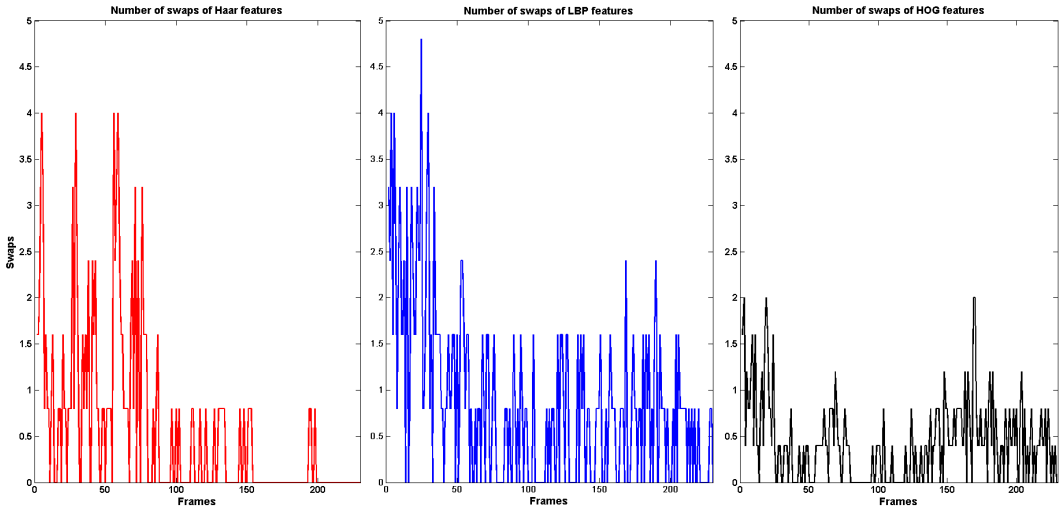

Figure 18: Percentage of swapped classifiers per feature type in the parking lot sequence divided by type.

References

[1] M. Aksela. Comparison of classifier selection methods for improving committee performance. In *International Workshop on Multiple Classifiers Systems*, pages 84–93, 2003.

[2] Shai Avidan. Support vector tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(8):1064–1072, August 2004.

[3] Shai Avidan. Ensemble tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(2):261–271, 2007.

[4] Ricardo A. Baeza-Yates and Berthier A. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press / Addison-Wesley, 1999.

[5] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[6] Gavin Brown, Jeremy L. Wyatt, and Peter Tiňo. Managing diversity in regression ensembles. *J. Mach. Learn. Res.*, 6:1621–1650, December 2005.

[7] M. Buckland and F. Gey. The relationship between recall and precision. *Journal of the American Society for Information Science*, 45(1):12–19, Jan 1999.

[8] Thierry Chateau, Vincent Gay-Belille, Frederic Chausse, and Jean-Thierry Laprest. Real-time tracking with classifiers. In *European Conference on Computer Vision*, 2006.

[9] Y. W. Chen and C. J. Lin. *Studies in Fuzziness and Soft Computing*, volume 207/2006, chapter Combining SVMs with various feature selection strategies, pages 315–324. Springer, 2006.

[10] Robert T. Collins, Yanxi Liu, and Marius Leordeanu. Online selection of discriminative tracking features. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(10):1631–1643, October 2005.

[11] J. Davis and M. Goadrich. The relationship between precision-recall and roc curves. In *International Conference on Machine learning*, pages 233–240, 2006.

[12] Csaba Főző and Csaba Gáspár-Papanek. 3-level confidence voting strategy for dynamic fusion-selection of classifier ensembles. *Acta Cybernetica*, 19(1):41–60, 2009.

[13] A. Frank and A. Asuncion. UCI machine learning repository, 2010.

[14] Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In *Thirteen International Conference on Machine Learning*, pages 148–156, 1996.

[15] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera. A review on ensembles for class imbalance problem: Bagging, boosting and hybrid based approaches. *IEEE Transactions on System, Man and Cybernetics - Part C: Applications and Reviews*, 42(4):463–484, 2012.

[16] Giorgio Giacinto, Roberto Perdisci, Mauro Del Rio, and Fabio Roli. Intrusion detection in computer networks by a modular ensemble of one-class classifiers. *Information Fusion*, 9(1):69–82, 2008. Special Issue on Applications of Ensemble Methods.

[17] Giorgio Giacinto and Fabio Roli. Design of effective neural network ensembles for image classification purposes. *Image and Vision Computing*, 19(9-10):699–707, 2001.

[18] H. Grabner, J. Sochman, H. Bischof, and J. Matas. Training sequential on-line boosting classifier for visual tracking. In *International Conference on Pattern Recognition*, 2008.

[19] Helmut Grabner, Michael Grabner, and Horst Bischof. Real-time tracking via on-line boosting. In *Proceedings of the British Machine Vision Conference (BMVC)*, volume 1, page 4756, September 2006.

[20] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[21] L.K. Hansen and P. Salamon. Neural networks ensembles. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12:993–1001, 1990.

[22] Tin Kam Ho. The random subspace method for constructing decision forests. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(8):832–844, 1998.

[23] http://people.kyb.tuebingen.mpg.de/spider/.

[24] Tae-Kyun Kim and Josef Kittler. Combining classifier for face identification at unknown views with a single model image. In *SSPR/SPR*, pages 565–573, 2004.

[25] J. Kittler, M. Hatef, R. P.W. Duin, and J. Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239, 1998.

[26] R. Kohavi and D. H. Wolpert. Bias plus variance decomposition for zero-one loss functions. In Lorenza Saitta, editor, *International Conference on Machine Learning*, pages 275–283. Morgan Kaufmann, 1996.

[27] Anders Krogh, Jesper Vedelsby, et al. Neural network ensembles, cross validation, and active learning. *Advances in neural information processing systems*, pages 231–238, 1995.

[28] L. I. Kuncheva. That elusive diversity in classifier ensembles. *Lecture Notes in Computer Science*, 2652:1126–1138, 2003.

[29] L. I. Kuncheva and C. J. Whitaker. Measures of diversity in classifier ensembles. *Machine Learning*, 51:181–207, 2003.

[30] L.I. Kuncheva. Switching between selection and fusion in combining classifiers: an experiment. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 32(2):146–156, 2002.

[31] L.I. Kuncheva and J.J. Rodriguez. Classifier ensembles with a random linear oracle. *IEEE Transactions on Knowledge and Data Engineering*, 19(4):500–508, 2007.

[32] Ludmila I. Kuncheva. Diversity in multiple classifier systems. *Information Fusion*, 6(1):3–4, 2005. Diversity in Multiple Classifier Systems.

[33] Ludmila I. Kuncheva and Juan J. Rodriguez. Classifier ensembles with a random linear oracle. *IEEE Trans. on Knowl. and Data Eng.*, 19(4):500–508, 2007.

[34] K. W. Lau and Q. H. Wu. Online training of support vector classifier. *Pattern Recognition*, 36(8):1913–1920, August 2003.

[35] Emilio Maggio and Andrea Cavallaro. *Video tracking: theory and practice*. John Wiley & Sons, 2011.

[36] Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.

[37] G Martinez-Muñoz, Daniel Hernández-Lobato, and Alberto Suárez. An analysis of ensemble pruning techniques based on ordered aggregation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(2):245–259, 2009.

[38] Hieu T. Nguyen and Arnold W. Smeulders. Robust tracking using foreground-background texture discrimination. *International Journal of Computer Vision*, 69(3):277–293, 2006.

[39] Nikunj C. Oza. Online bagging and boosting. In *2005 IEEE International Conference on Systems, Man and Cybernetics*, volume 3, pages 2340–2345, Oct. 2005.

[40] Nikunj C. Oza and Kagan Tumer. Classifier ensembles: Select real-world applications. *Information Fusion*, 9(1):4–20, 2008. Special Issue on Applications of Ensemble Methods.

[41] A. Özgür, L. Özgür, and Güngör T. Text categorization with class-based and corpus-based keyword selection. In Springer Verlag, editor, *Proceedings of ISCIS'05. Lecture Notes in Computer Science*, pages 607–616, 2005.

[42] T. Parag, F. Porikli, and A. Elgammal. Boosting adaptive linear weak classifiers for online learning and tracking. In *International Conference on Computer Vision and Pattern Recognition*, 2008.

[43] R. Pelossof, M. Jones, I. Vovsha, and C. Rudin. Online coordinate boosting. *ArXiv e-prints*, Oct 2008.

[44] Hanchuan Peng, Fuhui Long, and Chris Ding. Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(8):1226–1238, 2005.

[45] Nemanja Petrović, Ljubomir Jovanov, Aleksandra Pižurica, and Wilfried Philips. Object tracking using naive bayesian classifiers. In *ACIVS '08: Proceedings of the 10th International Conference on Advanced Concepts for Intelligent Vision Systems*, pages 775–784, 2008.

[46] Minh-Tri Pham and Tat-Jen Cham. Online learning asymmetric boosted classifiers for object detection. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[47] R. Polikar. Ensemble based systems in decision making. *IEEE Circuits and Systems Magazine*, 6(3):21–45, Third Quarter 2006.

[48] R. Polikar, A. Topalis, D. Parikh, D. Green, J. Frymiare, J. Kounios, and C. M. Clark. An ensemble based data fusion approach for early diagnosis of alzheimer's disease. *Information Fusion*, 9(1):83–95, 2008. Special Issue on Applications of Ensemble Methods.

[49] David A. Ross, Jongwoo Lim, Ruei-Sung Lin, and Ming-Hsuan Yang. Incremental learning for robust visual tracking. *International Journal of Computer Vision*, 77(1-3):125–141, 2007.

[50] D. Ruta. Multilayer selection-fusion model for pattern classification. In *Int. Conference of Artificial Intelligence & Applications (AIA)*, pages 403–173, Innsbruck, Austria, 2004.

[51] L. Snidaro and G.L. Foresti. Real-time thresholding with Euler numbers. *Pattern Recognition Letters*, 24(9-10):1533–1544, June 2003.

[52] D.M.J. Tax, M. Loog, and R.P.W. Duin. Optimal mean-precision classifier. In F. Roli J.A. Benediktsson, J. Kittler, editor, *Multiple Classifier Systems*, volume 5519, pages 72–81, Berlin, 2009. Lecture Notes in Computer Science, Springer.

[53] Carlo Tomasi, Slav Petrov, and Arvind Sastry. 3d tracking = classification + interpolation. In *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, page 1441, 2003.

[54] C. J. van Rijsbergen. *Information retrieval, Second edition*. Butterworths, 1979.

[55] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 511–518, Kauai, Hawaii, December 2001.

[56] I. Visentini, J. Kittler, and G. L. Foresti. Diversity-based classifier selection for adaptive object tracking. In *Int. Workshop on Multiple Classifiers Systems*, pages 438–447, Iceland, June 2009.

[57] Ingrid Visentini, Lauro Snidaro, and Gian Luca Foresti. Cascaded online boosting. *Journal of Real-Time Image Processing*, 5(4):245–257, 2010.

[58] Ingrid Visentini, Lauro Snidaro, and Gian Luca Foresti. Selecting classifiers by f-score for real-time video tracking. In *Proceedings of the 13th Conference on Information Fusion*, Endinburgh, Scotland, July 26-29th 2010.

[59] Björn Waske, Jon Atli Benediktsson, and Johannes R. Sveinsson. Classifying remote sensing data with support vector machines and imbalanced training data. In *Proceedings of the 8th International Workshop on Multiple Classifier Systems*, MCS '09, pages 375–384, 2009.

[60] G. Udny Yule. On the association of attributes in statistics. *Philosophical Transactions of the Royal Society of London*, 194:257–319, 1900. Ser. A.