Selecting classifiers by F-score for real-time video tracking

Ingrid Visentini, Lauro Snidaro and Gian Luca Foresti
Department of Mathematics and Computer Science
University of Udine, Italy
ingrid.visentini@uniud.it

Abstract – *In this work we propose the F-score measure as a novel means to perform online selection of the members of a classifier ensemble. This allows the fast application of a small number of selected classifiers for real-time applications such as target tracking for video surveillance. The proposed selection criterion relies on a performance evaluation to assess the ability of individual classifiers to predict the class membership, that is to discriminate between foreground and background in the context of video tracking. Preliminary experiments have shown encouraging results on real-world sequences.*

Keywords: Classifiers Fusion, Classifiers Selection, Object Detection, Video tracking

1 Introduction

To fuse classifiers a large number of possible rules can be used [26]: for instance, sum and product [15], Bagging [4] and Boosting [9], Random Subspaces [14], or oracles [18]. Considering couples of classifiers, mutual information [23], Q statistic [33], diversity-based criteria [17, 32] or correlation, for instance, can represent valid pairwise measures to join classifiers. In general, it is well known that the fusion of "weak" classifiers making independent errors can lead to dramatic improvements in classification performance [13]. These ensembles can be employed in a broad variety of applications, from medical imaging [27] to network security [10], in a large range of real-world domains [20].

Recently, the target tracking field in video applications has received a new boost thanks to the tracking via classification concept [11, 24]. Recent works include Avidan's Adaboost-based tracker [2], that exploits features associated to every pixel, and Collins' one, which is able to select the most discriminative color features to separate the target from the background by applying a two-class variance ratio to log likelihood distributions computed from samples of object and background pixels [7]. In this last case the features are selected at each epoch without considering past history. In a later work, heterogeneous features have been combined adopting the same fusion method [21]. Also in [11] the distinction between object and background is exploited to track the target constantly updating the model.

The new rise of online learning methods [19, 22] has opened the possibility to build on-the-fly a classifier ensemble and to train it with fresh samples in an unsupervised manner without any prior knowledge of the data distribution. All these methods are based on the evolution of Boosting, and rely on a fixed dimension ensemble of classifiers, whose weights are updated maintaining 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; assessing the performance of the classifiers in presence of an unbalanced number of training samples can be misleading. Pham and Cham [25] 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. But in this case the application of the entire pool of classifiers can be still computationally expensive.

An alternative to fusion is selection that is aimed at forming a reduced ensemble by choosing within a pool the subset of classifiers that maximizes the performance [16] or, alternatively, reduce the error. This approach is often applied to features [12] 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 include switching between fusion and selection [16].

In this work, we propose a new criterion based on the F-score to select classifiers from a set of constantly updated ensemble members. This criterion has been used in [6] applied to SVM, but its application in online learning is still unexplored to the best of our knowledge. The measure is based on the precision and recall scores of each individual classifier and 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 then simple single-frame selection approaches;
- it evaluates classifiers instead of features thus allowing the transparent integration of heterogeneous features;
- explicit handling of asymmetric samples distributions;
- 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.

2 Background and related work

2.1 Ensemble of classifiers

Consider two possible classes ω_+, ω_- so that $\omega_+ = +1$ represents the target, and $\omega_- = -1$ the background. Starting from several classifiers $h_1, h_2, \ldots, h_M$, an ensemble of predictors H can be built organizing the members with several linear fusion rules or with a non-linear combination. 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

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

Considering a sample x belonging to the sample set $\mathbf{X}$ and a classifier $h : \mathbf{X} \rightarrow \{+1, -1\}$, the pattern x is assigned to class ω if

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

where $w_c \in \{-1, +1\}$. The *confidence* of the classifier can be defined as the posterior probability

$$conf(h(x)) = \frac{P(\omega)P(h(x)|\omega)}{P(h(x))} \tag{3}$$

2.2 Precision and recall

Precision and recall are widely used to evaluate an algorithm's performance in Information Retrieval (IR) [3] or, generically, to measure the quality of a classification process [8]. 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 [8].

The *precision* π of a classifier h is defined as the probability that all the items $\{x_1, \ldots, x_k\}$ in the training set, that are labelled as belonging to class $\omega = +1$, actually belong to that class

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

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

$$\begin{aligned} \rho \equiv & \tfrac{1}{K}\textstyle\sum_k P(h(x_k) = +1|\omega = +1) \\ = & \tfrac{1}{K}\textstyle\sum_k \frac{P(h(x_k)=+1, \omega=+1)}{P(\omega=+1)} \end{aligned} \tag{5}$$

2.3 F–score measure

Keeping the error fixed, a tradeoff between Precision and Recall is intrinsic, as increasing one means reducing the other [5]. Usually the two measures 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 [31], is defined as their weighted harmonic mean,

$$\text{F-score}_\beta \equiv (1 + \beta^2)\frac{\rho\pi}{\beta^2\pi + \rho} \tag{6}$$

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

$$\text{F-score}_1 = 2\frac{\rho\pi}{\pi + \rho} \tag{7}$$

On the other hand, when $\beta < 1$ it favours recall, while precision is preferred otherwise.

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.

3 Proposed solution

In this work, we explore the use of the F-score as a means to rank classifiers and provide a robust on-the-fly selection of classifiers in a video tracking application.

First of all, all the classifiers are trained with some training samples. Then, the classifiers are ranked considering

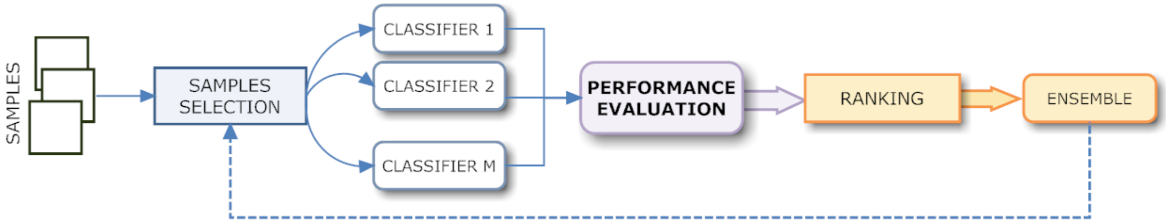


Figure 1: Architecture of the proposed approach. The selected ensemble, formed exploiting the F-score measure, is employed to track the target in the next step.

their ability to discriminate between the training patterns. In our case, the performance measure is represented by the F-score value that is calculated for each expert.

The ranking of the classifiers is a preliminary step to be performed before the selection phase; the selection set is in fact formed by choosing the best S predictors according to the F-score ranking. Alternatively, one can start with a random classifier and add more learners while the F-score increases.

The selection ensemble $\hat{H}$ so obtained

$$\hat{H}(x) = \frac{1}{S}\left(\sum_{s=1}^{S} h_s(x)\right) \tag{8}$$

is employed to detect the target in the image. Since the selection rule is required to be fast and accurate to allow a robust real-time tracking, we have chosen to optimize both precision and recall through the F-score measure and to rank the classifiers consequently.

3.1 Optimizing the F-score

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 [30]; for this reason, we need to find a common criterion to optimize the measure for ranking the classifiers.

Considering a training set constituted of a set of couples $(x_1, \omega_1), (x_2, \omega_2), \ldots, (x_N, \omega_N)$ where $x_n \in \mathbf{X}$ are image patches, and $\omega_n \in \{\omega_+ = +1, \omega_- = -1\}$ their labels, we can define the true positives as the number of positive samples correctly classified by the classifier h and counted by the indicator function I

$$TP = \sum_{n=1}^{N} I(h(x_n) = +1, \omega_n = +1) \tag{9}$$

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

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

$$FN = \sum_{i=1}^{N} I(h(x_n) = -1, \omega_n = +1) \tag{11}$$

Then, we can express the F-score as

$$\begin{aligned} \text{F-score}_\beta &= \frac{(1+\beta^2)TP}{\beta^2(TP+FN)+TP+FP} \\ &= \frac{(1+\beta^2)}{\beta^2+1+\frac{\beta^2 FN+FP}{TP}} \end{aligned} \tag{12}$$

or, when $\beta = 1$,

$$\text{F-score}_1 = 2\frac{TP}{2TP+FP+FN} \tag{13}$$

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

$$\frac{\beta^2 FN + FP}{TP} \tag{14}$$

in the denominator of (12) should be minimized. The smaller this ratio, the more discriminative the classifier.

This new formulation allows one to use the F-score as a fast classifier selection criterion that can be applied to every new 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. This is a critical factor when online learners are involved, because many other aspects, like training time, should be considered.

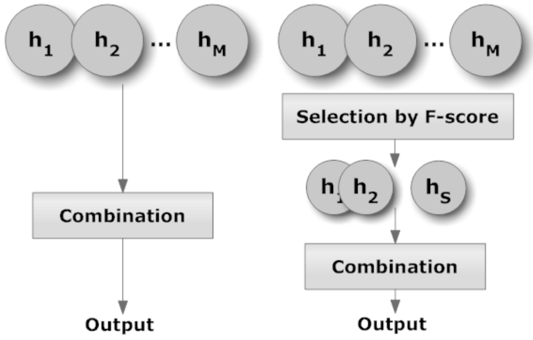
An intrinsic advantage of the proposed solution is that this allows one to form a flexible selection of classifiers. The members can be potentially replaced when their performance starts to decrease, or new ones can be added if the system encounters a critical situation. In this paper, the use of a fixed pool and selection ensemble cardinality allows one to understand how the proposed measure works; however, a dynamic way of regulating the number of classifiers in the ensemble will be the subject of future investigation.

3.2 Application to tracking

In Figure 1 a flowchart of the proposed approach is presented, while Algorithm 1 describes in detail the ranking and the selection steps, Algorithm 2 proposes the application of the classifier ensemble to a video tracking task. First of all, the classifiers in the pool are tested on positive and negative samples and their predictions are compared with the true labels (Figure 1, *performance evaluation* step). The values of misclassified and correctly classified samples contribute to

```
Algorithm 1: F-score based selection
Require: Positive sample(s) $x_+$ and negative one(s) $x_-$
Require: Classifiers selection $\hat{H} = \perp$
Require: Classifiers pool $H$
Require: Skewness weight $\beta$
  // For every classifier in $H$
  for $m = 1, 2, \dots, M$ do
    // Test the classifier on the positive pattern
    $\omega \leftarrow \arg\max_{\omega_k} P(\omega_k | h_m(f(x_+)))$ as in (2)
    if $\omega == +1$ then
      $TP_m \leftarrow TP_m + 1$
    else
      $FN_m \leftarrow FN_m + 1$
    end if
    // Test the classifier on the negative pattern
    $\omega \leftarrow \arg\max_{\omega_k} P(\omega_k | h_m(f(x_-)))$ as in (2)
    if $\omega == +1$ then
      $FP_m \leftarrow FP_m + 1$
    end if
    // Calculate the denominator of Eq. (12)
    $Fdenom_m \leftarrow \frac{\beta^2 FN_m + FP_m}{TP_m}$
  end for
  // Sort the denominators in ascending order
  $\mathrm{sort}(Fdenom)$
  // Fill the ensemble
  $counter \leftarrow 0$
  while $counter < S$ do
    $s \leftarrow \mathrm{index}(Fdenom(counter))$
    $\hat{H} \leftarrow \hat{H} \bigcup h_s$
    $counter \leftarrow counter + 1$
  end while
  // Update the classifiers parameters
  for $m = 1, 2, \dots, M$ do
    $\mathrm{update}(\mu_{m,\omega_+}, \mu_{m,\omega_-})$ as in Eq. (17)
    $\mathrm{update}(\sigma_{m,\omega_+}, \sigma_{m,\omega_-})$ as in Eq. (18)
  end for
  Return $\hat{H}$
```

```
Algorithm 2: Selection for tracking application
Require: Selection classifier ensemble $\hat{H}$
Require: Frame $F$ of size $I \times J$
  while $x \leftarrow \mathrm{subregion}(F, i, j)$ do
    // Test the ensemble on $x$
    // and save results in a temporary map
    // $map(i, j) \leftarrow conf(\hat{H}(x))$
  end while
  // Set the positive sample $x_+$
  $x_+ \leftarrow \arg\max(map)$
  // Set the negative sample $x_-$
  $x_- \leftarrow \mathrm{subregion}(map, randX, randY)$
  return $x_+$ as the target
```



(a) (b)

Figure 2: Comparison of a direct combination method (a) and a selection step, like in our case, that precedes the combination phase (b).

calculate the F-score denominator (*ranking* phase) for every classifier in the pool. This value is used to pick the $S << M$ classifiers from the original set H (*ensemble* generation).

Looking at Figure 2, the direct combination approach directly provides a classifier output, while, if the selection step is present as in our case, the combination is preceded by a ranking and discarding phase that reduces the cardinality of the ensemble.

The new selection ensemble $\hat{H}$ is employed to search for the object in the subsequent frame (*samples selection* module). This is probably 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.

At each frame the training set composition strictly depends on the output of the ensemble $\hat{H}$ in the previous frame. Algorithm 2 summarizes this approach: each subregion of the video frame F_t is processed by the selection ensemble $\hat{H}$. The 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 at time $t+1$

$$X_{t+1} \leftarrow x_+ = \arg\max_x(\hat{H}(x)), x \in F_t \tag{15}$$

The same can be said for the samples classified as negatives, that is belonging to the class ω_-

$$X_{t+1} \leftarrow X_{t+1} \cup x_- \in F_t \tag{16}$$

The tracker, in fact, behaves like an unsupervised system that searches for a positive sample into a set of possible candidates (patches of the image). Thus, there are no pre-set samples, but the classifier ensemble finds a single positive sample and a single random negative pattern in each frame to be learnt in the next step. When the ensemble classifier is used for tracking [11, 19], usually at each computation round (at least) one positive sample and one negative counterpart are unsupervisedly chosen from the video stream.

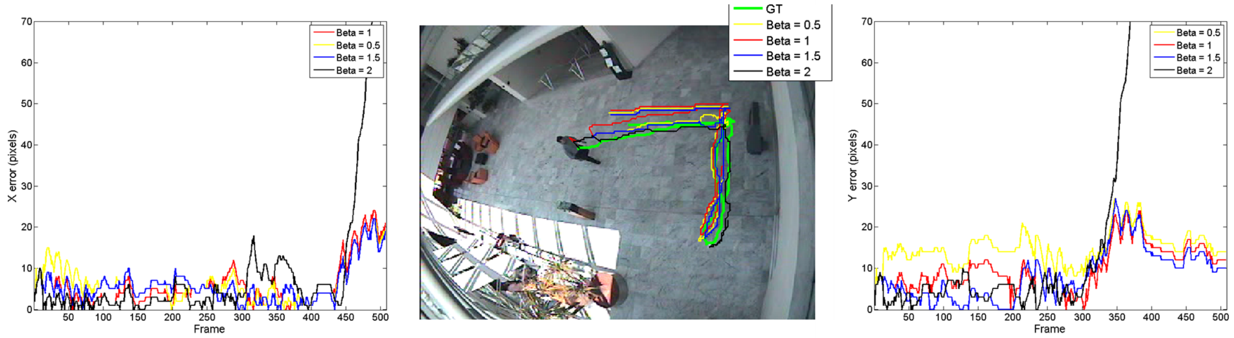


Figure 3: Centre image: Trajectories of the F-score tracker when varying β. Yellow: $\beta = 0.5$, red: $\beta = 1$, blue: $\beta = 1.5$, black: $\beta = 2$. Left image: error in pixels on the X coordinate compared with the ground truth. Right: error (in pixels) on the Y coordinate.

3.3 Parameters update

After a new sample is processed, the parameters of all classifiers must be updated to maintain the model coherence with the appearance of the target.

A single Bayesian classifier h_m maintains two distributions on the training data, regarding positive x^+ and negative x^- samples. When learning a specific pattern x, the parameters of the corresponding distribution at time t are updated considering the new observation $h(x)$. Thus, at time t the mean $\mu_{m,\omega}$ and the variance $\sigma^2_{m,\omega}$ of each class become

$$\mu_{m,\omega,t} = (1-\alpha)\mu_{m,\omega,t-1} + \alpha h(x) \tag{17}$$

$$\sigma^2_{m,\omega,t} = (1-\alpha)\sigma^2_{m,\omega,t-1} + \alpha(h(x) - \mu_{m,\omega,t})^2 \tag{18}$$

where α is a learning rate parameter. In our experiments, α was set to 0.25 taking into account the frame rate and the typical movement speed of observed objects; other previous experiments proved that small variations in this value produced no significant effects [29].

4 Experiments

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.

4.1 Choice of β

First of all, we describe how the β parameter of (6) was chosen. After several tests on the CAVIAR[1] sequences, where we bootstrapped a pool of 500 classifiers with only 20 positive hand-labelled samples, the trackers were compared.

The number of the classifiers used in our experiments was fixed at 500, while we choose only 100 to be selected. 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 pool is a concrete opportunity and it is fully supported by the proposed framework.

In Figure 3 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 3, 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 motivation brought us to set $\beta = 1.5$ for the rest of the experiments.

4.2 Settings

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. Of course our approach can be compared with different tracking algorithms (kernel or model based, particle filters, etc.), but we aimed 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 [19]
- **COL**: **COL**our tracker [7]

The Online Boosting has been selected because it is a weighted fusion strategy and can be exploited to linearly combine learning classifiers to track an object. It is a heavily learning-based approach, but has the drawback that 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 these

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

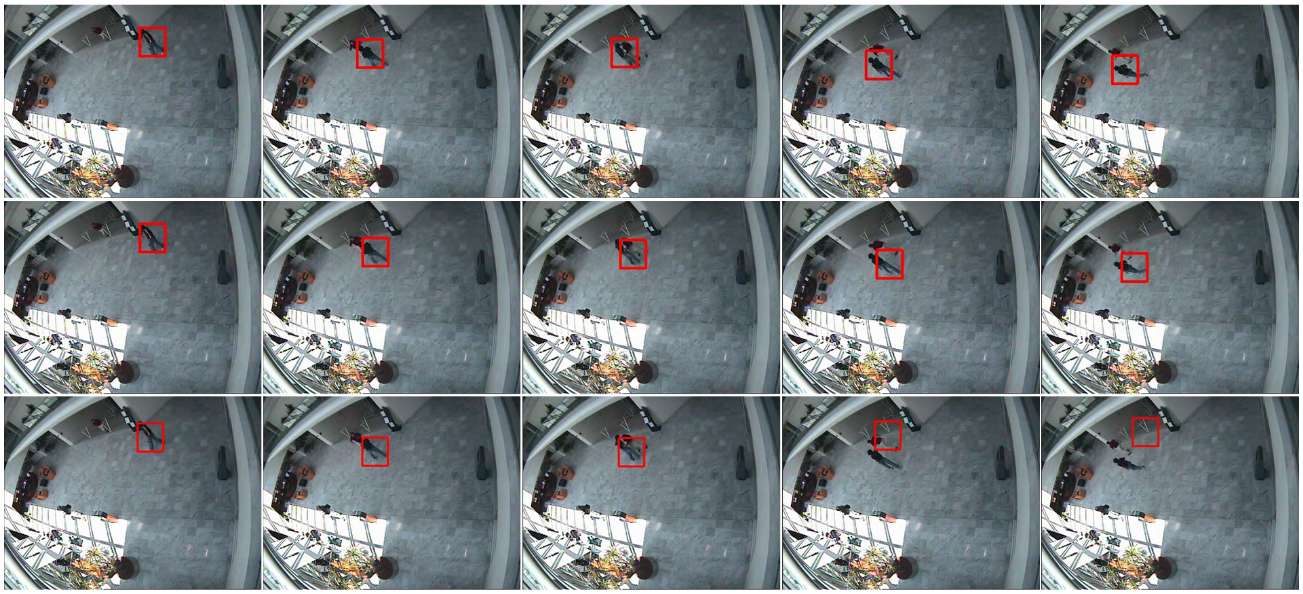


Figure 4: Comparison of output frames taken from PR (top row), Online Boosting (second row), and colour (bottom row) trackers run on the same video sequence.

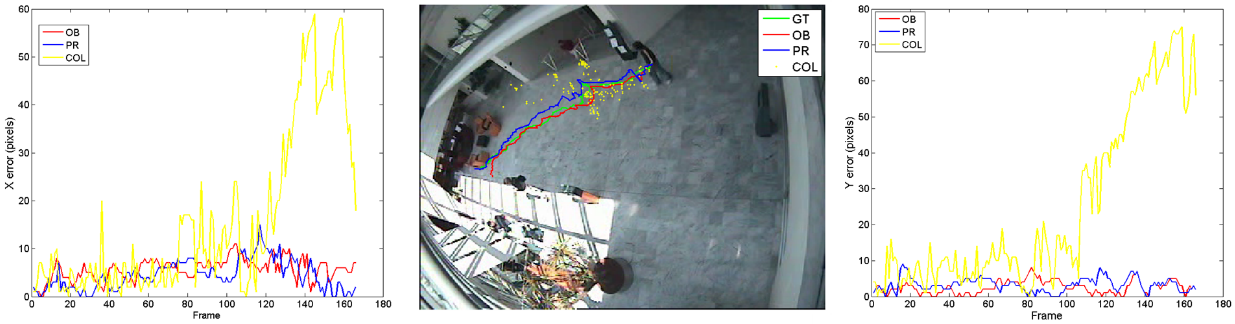


Figure 5: Centre image: Trajectories of the F-score tracker (said PR, blue), Online Boosting tracker (OB, red), and Colour tracker [7] (Col, yellow) compared with ground truth (GT, light green). Left image: error in pixels on the X coordinate for the previous trackers compared with the ground truth. Right: error (in pixels) on the Y coordinate.

predictors 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 [7] 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.

4.3 CAVIAR sequences

We used the CAVIAR dataset and the video sequence proposed in [28] to prove the effectiveness of our approach on standard sequences.

In Figure 4 are presented some frames from the *Fight_RunAway1.mpg* CAVIAR video sequences. 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 of 552 frames at 384 × 288 pixels

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

Table 1: 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 2: Application time (in milliseconds) *per frame* on the CAVIAR sequence (Fig.4) for the proposed approach and the Online Boosting tracker.

resolution. The target was initialized at frame 267 with a change detection algorithm. In this case no bootstrapping was required: in the first frame where the object appears, a model of the foreground is built using random features. As already said in Section 3.2, the training procedure at time t uses unsupervised samples coming from the search phase performed by the selection set at time $t-1$.

In the first row of Figure 4 the proposed technique output is shown; it correctly tracked the target even when the two men were 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, while the Online Boosting detector (second row of Figure 4) correctly followed the object, exploiting other features as shape and texture. In Figure 5 the errors with respect to the ground truth are presented.

The average shift in pixel from the ground truth for the PR tracker, the OB method and the Color tracker is presented in Table 1. The precision-recall based approach resulted more robust on the CAVIAR sequence, while the Color tracker's drift resulted in higher error with respect to the ground truth.

Table 2 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 tracker and the Online Boosting, as we can see from Table 2 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.

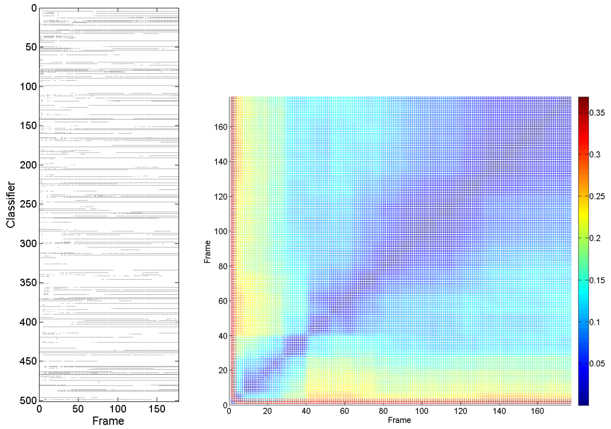


Figure 6: Left: Trace of the features used by the PR tracker in the CAVIAR sequence. Right: Hamming distance between selections of classifiers ensembles.

The selection process is highlighted in Figure 6, where the trace of the selected classifiers is shown in the left part of the figure. The frames on the x-axis are numbered from the instant in which the target appears in the scene. The 500 features are subdivided in four groups. In the first part, Haar features go from 1 to 125, the LBPs from 126 to 250, and HOG and colour respectively fill the remaining portion of the graph. Reflecting the sudden change of the target pose in this sequence, a high number of classifiers are swapped in the ensemble. This can be seen in the graph showing the number of changed classifiers (represented with the Hamming distance) between pairs of frames, where a high value denotes an activity in subsequent (with peaks of 38%) or contiguous frames. From frame 40 to frame 80 the uncertainty of the classification (due to the occlusion) results in a peak of about 30% swapped classifiers.

To conclude, preliminary experiments showed promising results suggesting that the selection process can be used to reduce the computational requirements in a video tracking application while at the same time retaining an acceptable level of accuracy when compared to similar approaches.

5 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. 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. Preliminary results suggested that the proposed approach achieved an improvement in terms of speed and accuracy with respect to similar state-of-the art algorithms on real-world video sequences.

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

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

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

[5] 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.

[6] 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.

[7] 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.

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

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

[10] 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.

[11] 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.

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

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

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

[15] J. Kittler, M. Hatef, R. P.W. Duin, and J. Matas. On combining classifiers. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(3):226–239, 1998.

[16] 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.

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

[18] 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.

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

[20] 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.

[21] 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.

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

[23] 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.

[24] 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.

[25] 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.

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

[27] 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.

[28] 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.

[29] L. Snidaro and I. Visentini. Fusion of heterogeneous features via cascaded on-line boosting. In *Proceedings of the Eleventh International Conference on Information Fusion*, pages 1340–1345, Cologne, Germany, June 30th-July 3rd 2008.

[30] 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.

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

[32] 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.

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