Fast Object Detection with Entropy-Driven Evaluation

Raphael Sznitman[1], Carlos Becker[1], François Fleuret[1,2], and Pascal Fua[1]

[1]École Polytechnique Fédérale de Lausanne (EPFL), Switzerland
[2]IDIAP Research Institute, Switzerland
{firstname.lastname@epfl.ch}, francois.fleuret@idiap.ch

Abstract

Cascade-style approaches to implementing ensemble classifiers can deliver significant speed-ups at test time. While highly effective, they remain challenging to tune and their overall performance depends on the availability of large validation sets to estimate rejection thresholds. These characteristics are often prohibitive and thus limit their applicability.

We introduce an alternative approach to speeding-up classifier evaluation which overcomes these limitations. It involves maintaining a probability estimate of the class label at each intermediary response and stopping when the corresponding uncertainty becomes small enough. As a result, the evaluation terminates early based on the sequence of responses observed. Furthermore, it does so independently of the type of ensemble classifier used or the way it was trained. We show through extensive experimentation that our method provides 2 to 10 fold speed-ups, over existing state-of-the-art methods, at almost no loss in accuracy on a number of object classification tasks.

1. Introduction

Ensemble classifiers, such as Tree-based Random Forests (RF) [5], Boosted Stumps (BS) [9] and Boosted Trees (BT) [6] have proven effective at solving tasks such as object detection [26, 10, 2], image retrieval in large datasets [27, 30] and object categorization [3, 14, 21, 18].

Typically, using an ensemble classifier requires evaluating the many individual classifiers they are made of. Due to the ever growing amount of data that must be predicted, and in spite of their efficient prediction procedure, the resulting computational requirements can remain prohibitive, especially for applications with limited computational power.

Speeding up object classifiers is a well researched problem and it is well known that at its heart lies a trade-off between accuracy and speed [26, 23, 4, 29, 8]. Perhaps the most famous demonstration of this trade-off is found in the seminal face detection paper [26], where a "hard" cascade of classifiers was used to filter out and reject non-faces while only mildly decreasing the overall classifier accuracy.

Since then, a number of new ways to improve this trade-off have been proposed, mainly along the lines of rejecting non-object candidates quickly. For example, a relaxation of the hard cascade to a *soft* one was introduced to more effectively reject non-target candidates and to alleviate many of the difficulties encountered when training hard cascades [4]. Similarly, some have used rejection criteria based on Walds' Sequential Probability Ratio Test [23] or using more empirical observations [29].

While these methods have been successful, they require either strong independence assumptions on the output of the weak learners to guarantee optimality [23], or precise estimation of a set of rejection thresholds, whose values are computed from samples of a sufficiently large *validation set* [4, 29]. In the first case, performance suffers when the assumptions are violated, which is often. In the second, the methods are at a disadvantage when training and validation data are expensive or difficult to acquire.

In this paper we propose a method that classifies candidates quickly on the basis of a sequence of results, or *stages*, computed during prediction. We introduce a generic Bayesian framework that keeps track of the uncertainty of a candidate's class label as prediction proceeds and stops when it falls below a chosen level. Unlike previous methods that need multiple stage-specific thresholds, the class label uncertainty in our method only requires a *single* threshold, common to all stages, to specify when the uncertainty is small enough. We show experimentally that our method provides significant gains in speed or accuracy, and often both, over state-of-the-art early stopping methods on a number of object classification tasks when using different ensemble classifiers.

The remainder of this paper is organized as follows: In Sec. 2 we briefly survey the related literature. Sec. 3 intro-

duces our general framework and method, while in Sec. 4, we demonstrate the performance of our approach on different tasks.

2. Related Work

There has been much interest in speeding up classifiers based on context. For the purpose of finding objects in images, some successful methods have exploited image-based features that are either global, local or additive within complex optimization schemes to prune large areas of the search space and speed up detection [17, 24]. More recently, Multiple Instance Learning [29, 28, 8] has also been used to this end, by simultaneously evaluating multiple candidates based on their local neighborhoods. This has been shown to be particularly appropriate when a single positive sample is surrounded by many negative ones, as is the case of faces in typical scenes.

While effective, these methods are not intended for cases that require independent classification of each instance. Among methods designed for this purpose and that attempt to reject negatives early, one of the earliest and most influential works is the hard cascade of classifiers proposed in [26] for face detection. In that work, rejection thresholds on a set of distinct classifiers were used to conservatively filter out non-faces during classification. Since then, a number of strategies have been proposed to optimize how cascades are built [15, 16, 20].

Perhaps most relevant to this work is that of [4], where the hard cascade was replaced by a single *soft cascade* classifier with stage-wise rejection thresholds that were computed using a cascade calibration procedure (**CCAL**). More specifically, thresholds on the sum of stage scores were found by adjusting a performance vector such that each stage of the soft cascade rejected at most a fixed proportion of positive targets from a validation set. While effective, this approach requires the user to adjust a performance vector and target error rate. Furthermore, the final classifier accuracy is closely linked to the quantity and variability of positive samples in the validation set used to calibrate the cascade. Along the same lines, a simple Direct Backward Pruning algorithm (**DBP**) was introduced in [29] which sets rejection thresholds on the sum of stage scores. This was achieved by taking the threshold at each stage to be the minimum sum of stage scores observed over a validation set or subset. While effectively removing the need for the performance vector of [4], this strategy can still only perform well for validation subsets large enough so that thresholds generalize to the test set.

A different approach to setting rejection thresholds is that of [23], where stage-wise thresholds are based on the Sequential Probability Ratio Test (**SPRT**). This test is shown to be optimal when individual observations at each stage are i.i.d. However, in practice, the ratio test used relies on the sum of stage scores, hence strongly correlating observations and thus violating a number of assumptions.

In short, all the above-mentioned approaches to early termination of a classifier evaluation critically depend either on large validation sets or on several strong assumptions on the behavior of weak classifiers. Our approach avoids these requirements by tracking the uncertainty of the candidate class in a Bayesian way.

3. Method

As in cascade approaches [26, 23, 4, 29], given an ensemble classifier that sequentially evaluates stages, our goal is to reduce the computational burden by using as few resources as possible on easy-to-classify cases. We differ from earlier approaches in that we track the class label in a Bayesian way by modeling the individual stage computations, and dynamically infer when additional classification stages are necessary at run-time. In some sense, our method *normalizes each stage individually* and accumulates their normalized evidence. The resulting procedure is able to cope with non-reliable stages by properly quantifying the evidence they provide, instead of using their shear scores as an indicator.

In the remainder of this section, we begin by specifying our notations and using them to describe in a unified manner the **CCAL**, **DBP** and **SPRT** algorithms discussed in Section 2. We then introduce our own Entropy Driven Evaluation (**EDE**) approach to making early decisions and the algorithm that implements it.

3.1. Notation

For a given sample, let $x \in \mathbb{R}^D$ be its feature vector and $Y \in \{-1, 1\}$ its label. We take $F : \mathbb{R}^D \rightarrow \{-1, 1\}$ to be an ensemble classifier of the form

$$F(x) = sign\left(\sum_{k=1}^{K} g_k(x)\right), \tag{1}$$

where

$$g_k : \mathbb{R}^D \rightarrow \mathbb{R}, \tag{2}$$

is a *stage score* at stage index k. In BS classifiers [9, 13], a stage is defined as $g_k(x) = \alpha_k h_k(x)$ where h_k is a weak learner and α_k its weight. Similarly, in BT and RF classifiers [5], g_k is a decision tree. In the remainder of this paper, we will assume that F has been trained using a dedicated training set and that $\{g_k\}_{k=1}^{K}$ are known and fixed. Let $G_k(x) = (g_1(x), \ldots, g_k(x))$ and

$$f^k(x) = \sum_{m=1}^{k} g_m(x), \tag{3}$$

be the sum of the first k stage scores. Fig. 1 depicts this sum for a few examples from a validation set $\mathcal{V} =$

Table 1. Summary of Notation

$x \in \mathbb{R}^D$	Sample feature vector
$Y \in \{-1, 1\}$	Sample class label
$\mathcal{V}$	Validation set
N	Number of validation samples
ϕ	Stopping criterion
k_x^*	Stopping stage for sample x
$g_k(x)$	Stage score k
$G_k(x)$	k first stage scores
$f^k(x)$	Sum of k first stage scores
$F(x)$	Ensemble classifier
ϵ_k	k^{th} posterior probability of positive class
δ	Block offset
γ	Entropy threshold
ϵ^*	Probability threshold

$\{(x_n, y_n)\}_{n=1}^N$. In this figure, each curve shows the sum of stage scores of a BS face classifier for an example face (green) or non-face (red). Note that we consider the validation set to be disjoint from the classifier training set.

3.2. Framework

For a given test sample x, we want to evaluate as few stages as possible for a classifier of the form given in Eq. (1). Let the number of stages to evaluate be

$$k_x^* = \arg\min_{k=1,\ldots,K} \{\phi(g_1(x), \ldots, g_k(x)) \geq 0\}, \quad (4)$$

where the ϕ function is a *stopping criterion*. As discussed in Sec. 2, what differentiates earlier approaches is the way ϕ is defined. In fact, in both [4] and [29], the stopping criteria are of the form

$$\phi(g_1(x), \ldots, g_k(x)) = f^k(x) - \theta_k, \quad (5)$$

where the $\theta_k \in \mathbb{R}, k = 1, \ldots, K$ are thresholds estimated from the validation set $\mathcal{V}$.

The **DBP** algorithm [29] selects each threshold by computing

$$\theta_k = \min_{\{n: y_n = 1, f^K(x_n) > \tau\}} f^k(x_n), \quad (6)$$

where τ is a user specified threshold on the final sum $f^K(x)$.

In the **CCAL** algorithm of [4], the thresholds are taken to be $\theta_k = \max_{r \in \mathbb{R}} r$ such that

$$\sum_n pred(f^k(x_n) \leq r) y_n \leq p_k |P|, \quad (7)$$

where $|P|$ is the total number of positive examples in the validation set, p_k is a user specified proportion and $pred$ is a function that returns one if $f^k(x_n) \leq r$ and zero otherwise.

Similarly, the **SPRT** stopping criterion in [23], is of the form

$$\phi(g_1(x), \ldots, g_k(x)) = \max\{f^k(x) - \overline{\theta}_k, \underline{\theta}_k - f^k(x)\}, \quad (8)$$

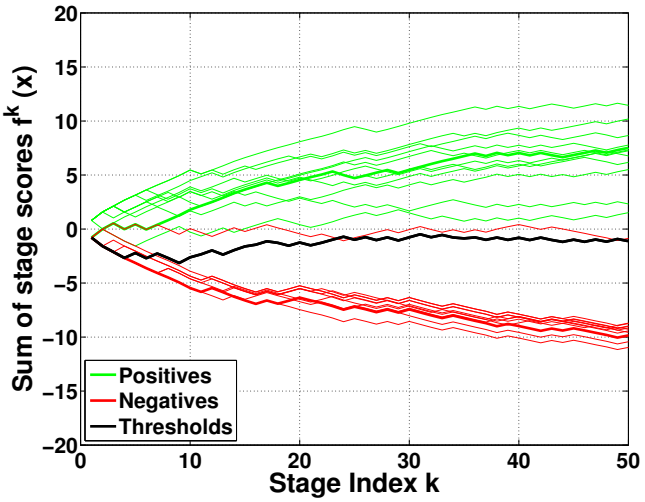

Figure 1. Example of the sum of stage scores produced by a face classifier as function of the stage index. Each green and red line corresponds to a face and non-face sample from a validation set. The jittery black line shows a typical set of rejection thresholds produced by the **CCAL** procedure. Examples with sum of scores that fall below the black line at any stage k are rejected early.

where $(\overline{\theta}_k, \underline{\theta}_k)$ are upper and lower thresholds estimated using the SPRT. However, in practice $\overline{\theta}_k$ is always set to be larger than $f^k(x)$, reducing Eq. (8) to the same form as Eq. (5).

Clearly, the performances of these stopping criteria are strongly dependent on the quality and representativity of the validation set used, as their threshold values are explicitly selected from examples.

3.3. Entropy Driven Evaluation

We define the stopping criterion ϕ in a significantly different way. First we consider that the sample to evaluate is randomly selected, and hence we model the class label Y as a discrete random variable with probability distribution $P(Y)$, *i.e.* a Bernouilli random variable. After each stage, we observe the value, $g_k(x)$, which we also treat as a random variable and for which we can evaluate $P(g_k(x)|Y = 1)$ and $P(g_k(x)|Y = -1)$. In addition, we assume that $g_k(x)$ is conditionally independent from $g_j(x), j < k$ given the class label, which leads to

$$P(G_k(x)|Y = y) = \prod_{m=1}^{k} P(g_m(x)|Y = y). \quad (9)$$

Unlike [23] which also assumes conditional independence given the class label, this is our only assumption on the behavior of the stage scores. Given this, we take our stopping criteria to be

$$\phi(g_1(x), \ldots, g_k(x)) = \gamma - H(Y|G_k(x)), \quad (10)$$

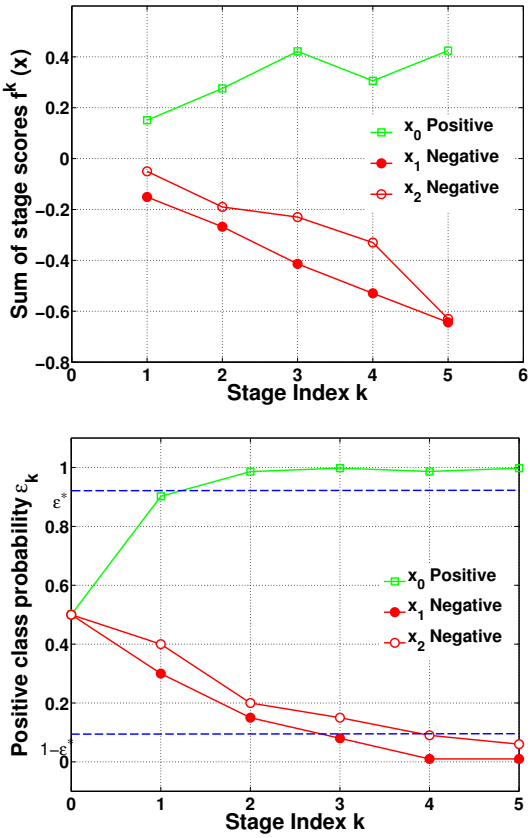

Figure 2. Example of **EDE** process. (*top*) Sum of stage scores, $f^k(x)$ for a positive sample x_0 (green) and negative samples x_1 and x_2 (red), as function of the stage index k. (*bottom*) Evolution of posterior distribution as a function of the index k for each sample. The dotted line depicts a chosen entropy threshold γ, converted to its corresponding probability thresholds ϵ^* and $1-\epsilon^*$.

where $H(Y|G_k(x))$ is the conditional entropy of Y when $G_k(x)$ has been observed [7] and $\gamma \in (0,1)$ is a user specified threshold. In this context, the conditional entropy provides a measure of uncertainty on the class label, and hence we look for k_x^* such that it reduces the uncertainty below a specified level γ.

We now show how to compute this conditional entropy. First, note that for any value of k, we can compute the posterior distribution of Y after evaluating k stages. To do this, we observe that

$$\begin{aligned} P(Y|G_k(x)) &= \frac{1}{Z} P(g_k(x)|Y, G_{k-1}(x)) P(Y|G_{k-1}(x)) \\ &= \frac{1}{Z} P(g_k(x)|Y) P(Y|G_{k-1}(x)), \end{aligned} \quad (11)$$

where we have used Eq. (9) to derive Eq. (11). This recursive form of the posterior is convenient as it allows for an easy update rule for sequential observations. As such, if we let $\epsilon_k = P(Y=1|G_k(x))$ then, for any stage k

$$\epsilon_{k+1} = \frac{1}{Z} P(g_{k+1}(x)|Y=1)\epsilon_k, \quad (12)$$

where

$$Z = \epsilon_k P(g_{k+1}(x)|Y=1) + (1-\epsilon_k) P(g_{k+1}(x)|Y=-1).$$

As in [23], we can estimate the conditional likelihoods, $P(g_k(x)|Y=1)$ and $P(g_k(x)|Y=-1)$ using the validation set. To do this, we represent each distribution using a histogram and use a Parzen window technique with a Gaussian kernel to smooth the estimation. We choose our kernel width to be: $h_{os} = 1.44\sigma n^{-1/5}$ where n is the number of examples used to estimate the density and σ is the sample standard deviation. Using such a kernel width has been shown to be fairly stable for density estimation with relatively few samples [23, 22].

Hence, the conditional entropy becomes

$$\begin{aligned} H(Y|G_k(x)) &= H(\epsilon_k) \\ &= -\epsilon_k \log_2(\epsilon_k) - (1-\epsilon_k)\log_2(1-\epsilon_k). \end{aligned} \quad (13)$$

Note, that since the entropy $H(\epsilon_k) \in (0,1)$ is a concave and symmetric function (*i.e.* $H(\epsilon) = H(1-\epsilon), \epsilon > 1/2$), we may use a threshold on the probability ϵ_k instead of on the entropy. This can be achieved by solving: $\gamma = -\epsilon\log(\epsilon) - (1-\epsilon)\log(1-\epsilon)$ for $\epsilon \in (0,1)$, which provides two probability thresholds: ϵ^* and $1-\epsilon^*$. In fact, using the probability threshold is more efficient than using the entropy threshold as the former only requires checking an inequality between two scalars, while the latter also requires computing Eq. (13).

To highlight the difference between our approach with the methods described in Sec. 3.2, consider the following example. As depicted in Fig. 2, let two samples, x_1 and x_2 have sum of scores $f^{k^*}(x_1) = f^{k^*}(x_2)$, where $k^* = 5$ is the first stage where rejection takes place for both x_1 and x_2, when using a method from Sec. 3.2. In this case, both x_1 and x_2 are rejected since rejection solely depends on $f^{k^*}(\cdot)$. In **EDE** however, early stopping depends on the posterior distribution, which depends on the sequence $G_k(x)$ and not $f^k(\cdot)$. That is, early stopping of our method depends on the progression of values at each stage, and not on the sum of stage scores. As such, x_1 could be rejected earlier than x_2.

Furthermore, our method normalizes each stage individually, and estimates the reliability of individual stages. To build some intuition about this fact, consider an extreme example where a given stage computes a random response that is completely unrelated to x. Such a stage would strongly impact previous methods in a negative way, as it would force the threshold at that stage to be unreliable. **EDE** on the other hand, would effectively ignore the stage from the overall estimation, as it would weigh the information at this stage very poorly, since $P(g_k(x)|Y=1)$ and $P(g_k(x)|Y=-1)$ would be similar.

3.4. Block Evaluation

While in [23, 4, 29] the stopping criteria are evaluated at each stage of the prediction procedure, this may not be necessary in some cases. For this reason, we propose to evaluate the stopping criterion at specific intervals of stages.

Let $\delta \in \{1, \ldots, K-1\}$ be a *block offset*, then we can update the posterior distribution and evaluate the stopping criterion at every δ stages. In this case, the posterior can be written as

$$\epsilon_{k+\delta} = \frac{1}{Z} P(g_{k'}(x)|Y=1)\epsilon_k, \tag{14}$$

where $g_{k'}(x) = \sum_{m=k+1}^{k+\delta} g_m(x)$ and note that when $\delta = 1$, Eq. (12) is recovered.

As will be shown in our experiments, this block evaluation of the stopping criterion is not only beneficial as it reduces the need to update the posterior at each stage, but also makes **EDE** less sensitive to noise contained in each stage observation. While this may make evaluation slower (*i.e.* more stages are evaluated), this typically improves accuracy.

3.5. Algorithm

Our run-time algorithm is summarized by Alg 1. The user provides a test example x with a prior on the label, ϵ_1, as well as the conditional distributions $P(g_{k'}(x)|Y=y)$, the stopping threshold γ and the block offset δ.

Algorithm 1 Entropy Driven Evaluation (**EDE**)

Require: $x, \epsilon_1, \gamma, \delta, \{P(g_{k'}(\cdot)|Y=y)\}_{k=1}^{K/\delta}$

1: $k \leftarrow 1$
2: **while** $k \leq K$ and $H(\epsilon_k) > \gamma$ **do**
3: Compute: $g_{k'}(x) = \sum_{m=k+1}^{k+\delta} g_m(x)$
4: $\epsilon_{k+\delta} = \frac{1}{Z} P(g_{k'}(x)|Y=1)\epsilon_k$
5: $k \leftarrow k + \delta$
6: **end while**
7: **return** $y = sign(\epsilon_k - 0.5)$

In general, the only difference between our algorithm and the evaluation of a typical ensemble classifier is the update of the posterior distribution (line 4). This involves executing two lookups to compute the likelihoods, three multiplications, one division, addition and subtraction. For most ensemble classifiers, the cost of line 4 is therefore far smaller than that of evaluating a particular stage.

4. Experiments

We now demonstrate the efficiency of our proposed **EDE** stopping criterion for three different tasks: face detection, image classification and structure recognition.

As noted in [23, 29], comparing published results of competing early stopping methods is difficult because they are produced by pipelines that depend on training data, specific features being used, approach to non-maximal suppression, and parameter settings among many other things. Therefore, to compare our early stopping approach against others as fairly as possible, we reimplemented the **CCAL**, **DBP** and **SPRT** stopping criteria and evaluate each approach using the same classifier and validation set for each task mentioned above.

In each of the following experiments, we estimated the parameters of **EDE** by cross-validation on the validation set. That is, to determine $\gamma \in (0, 0.1), \epsilon_1 \in (0, 0.5)$ and $\delta \in \{1, ..., K-1\}$, we performed a brute-force search of the parameter space, and selected the parameters that required the smallest number of stage evaluations, and for which at most 1% of the classification accuracy was incorrect when compared to non-early-stopping prediction. For each triple, this process was repeated 5 times over the validation set and the best triple was selected.

In general, we are interested in observing how a method performs in terms of the number of stage evaluations and prediction accuracy. To compare performances between methods, we will therefore specify and tune user parameters to achieve either similar prediction accuracy or evaluate a similar number of stages. By doing so, we may observe if one method performs better on one performance criteria while the other is fixed.

4.1. Face Detection

We begin by evaluating each approach on a face detection task. Here, we used 4000 positive and 5 million negative examples to train a BS classifier with 500 weak learners, as described in [2]. The validation set was comprised of 4000 positives and 6000 negatives. This set was used to compute both the **CCAL**, **DBP** and **SPRT** rejection thresholds and the conditional probabilities $P(g_k(x)|Y=y)$ of Section 3.3 that **EDE** requires.

For this experiment, we adjusted the number of stage evaluations, or stumps, required by each early stopping method to be approximately the same. We then compared the accuracy performance of the four stopping criteria on the MIT+CMU dataset [19], which consists of 130 images containing a total of 507 labelled faces. To this end, we used a detector window size of 30×30 pixels, a sliding window of 1 pixel, a scaling factor of 1.25, and the same non-maximum suppression parameters in all cases.

In Fig. 3, we report the accuracy and the distribution of stage evaluations required by each method. Note that our **EDE** method uses on average slightly under 10 stages, to achieve near equal accuracy levels to traditional BS with non-early stopping. For approximately the same average number of stump evaluations, **CCAL**, **DBP** and **SPRT** per-

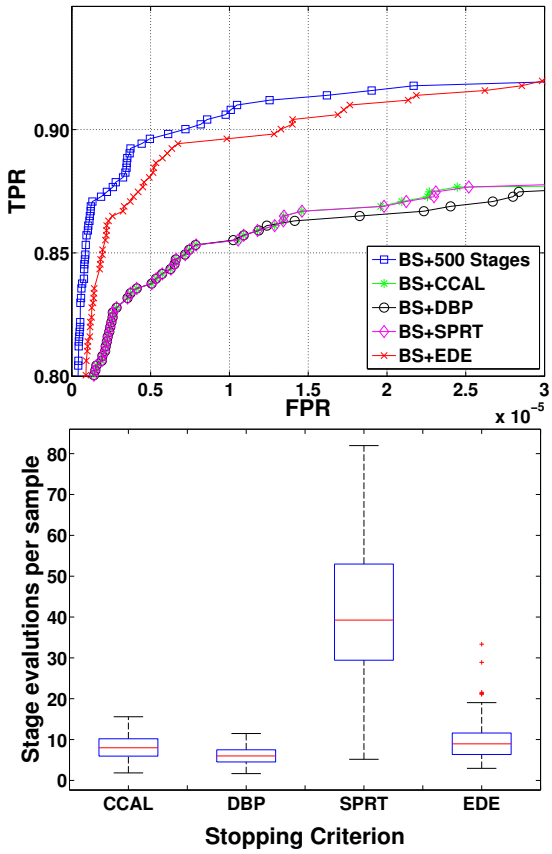

Figure 3. (*top*) Face detection ROC curves for each early stopping method evaluated and for non-early stopping on the MIT+CMU face dataset. Each stopping criterion was re-implemented and evaluated using the same Boosted Stumps (BS) classifier, training set and validation set. (*bottom*) Distribution of the number of stage evaluations of test sample for each method. In this case, each stage consists in the evaluation of a single stump.

form significantly worse than **EDE**.

4.2. Caltech Dataset

We also validated our approach on the Caltech-256 object classification dataset [12], and chose the task of binary classification of the form "object versus clutter" for different object categories. Since a validation set is required to estimate the rejection thresholds, we only evaluate object categories for which enough positive samples are available to form large enough training and validation sets. In this case airplanes, motor-bikes, and easy faces.

We trained BS, BT, and RF classifiers for each task using a publicly available feature set [11][1], which consists of several types of features such as SIFT, PHOG and Linear Binary Patterns.

In each case, we used 250 positive and 250 negative examples for training and validation, and used the remaining examples for testing. In addition, for each classification task and classifier, we also performed 10 independent training rounds, where we randomly chose 150 positive and 150 negative samples, and used the remaining 200 samples for validation. The BS, BT and RF classifiers were constructed using 500 stumps and trees, respectively.

[1]Caltech-256 features available at:
http://www.idiap.ch/%7Ettommasi/source_code_CVPR10.htm

Classifier & Stopping Criteria		Object Category: Motorcycle	Airplane	Faces
Boosted Stumps	*None*	*500 (.993)*	*500 (.964)*	500 (1)
	EDE	**47 (.992)**	**77 (.960)**	**31 (.999)**
	CCAL	232 (.981)	220 (.946)	110 (.992)
	DBP	232 (.977)	223 (.938)	117 (.992)
	SPRT	257 (.988)	281 (.959)	132 (.997)
Boosted Trees	*None*	*500 (.993)*	*500 (.960)*	500 (1)
	EDE	**48 (.991)**	**100 (.957)**	**32 (.998)**
	CCAL	239 (.983)	215 (.935)	116 (.989)
	DBP	237 (.978)	219 (.931)	120 (.990)
	SPRT	253 (.979)	274 (.949)	143 (.997)
Random Forest	*None*	*500 (.990)*	*500 (.962)*	500 (1)
	EDE	**52 (.988)**	**85 (.952)**	**32 (1)**
	CCAL	292 (.986)	259 (.933)	136 (.992)
	DBP	282 (.982)	249 (.926)	142 (.992)
	SPRT	316 (.986)	329 (.941)	157(.997)

Table 2. Average number of stage evaluations and best F-score (in brackets) on object classification tasks for three Caltech-256 categories either without an early stopping criterion or with one of the four discussed in this paper. Our **EDE** criterion, in bold, not only provides the least number of stage evaluations but also the best F-score when compared to other early stopping methods.

For each classifier and stopping criterion pair, we report in Table 2 both the average number of stage evaluations required and the best F-score (in brackets) for each classification task. The F-score [1] is computed as $Fscore = 2\frac{PrRe}{Pr+Re}$, where Pr and Re are the precision and recall levels for a given classifier threshold. The best F-score is chosen over all classifier threshold values and gives a general notion of classification performance.

In all evaluated cases, the **EDE** stopping criterion outperforms the other three methods on both speed and accuracy. Furthermore by tolerating a small reduction in accuracy when compared to non-early stopping, **EDE** allows for up to 15 fold speed increases.

To illustrate the effect of different parameter settings for our method, as well as for others, Fig. 4 depicts the best F-scores of each early stopping method as a function of the average number of evaluations when using a BT classifier on the Airplane classification task. Here, we have evaluated each approach with different parameters. That is for **EDE**, each blue circle represents the best F-score and number of

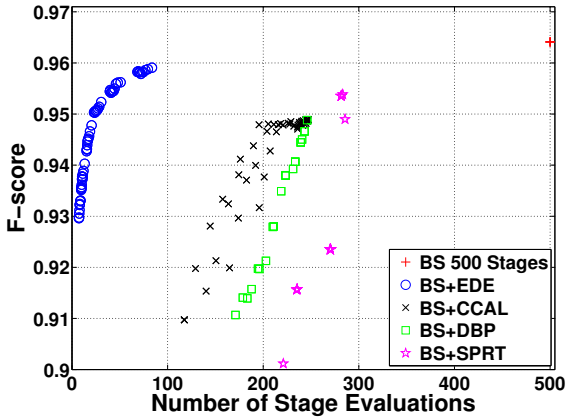

Figure 4. Best F-score as function of number of stage evaluations required for a Boosted Tree (BT) classifier with different stopping criteria for Caltech-256 Airplane classification. Each method uses the same classifier and validation set and each point shows the average performance for a stopping method when used with a specific set of parameters. For **EDE**, in blue, we show the effect of selecting different values of γ and δ.

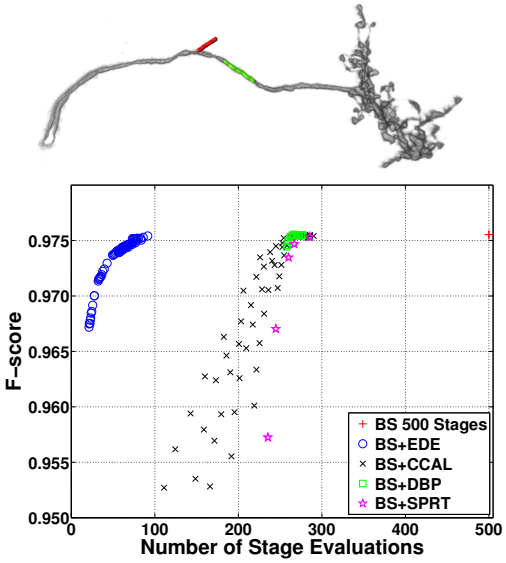

Figure 5. (*top*) Classification of positive (green) and negative (red) tubular linear structures. (*bottom*) Best F-score as a function of the number of stage evaluations required for a Boosted Stump (BS) classifier on the above classification task. Each point shows the average performance for a stopping method when used with different sets of parameters values.

stages evaluated when using a specific value of γ and δ. Similarly, each point for the other methods represents their performances using different values for their respective parameters.

4.3. Path Classification

Lastly, we evaluate our method on a biomedical classification task. In [25], the authors proposed an approach to classifying tubular paths, such as the one depicted in Fig. 5(*top*), as truly corresponding to linear structures or not. These paths are obtained by selecting pairs of 3D points that appear to be on the centerlines of linear structures and connecting them to form paths. Consequently, certain paths generated by this process belong to linear structures while others do not. To assess which is which, their algorithm computes a feature vector based on gradient histograms for each path and classifies it as a path or not.

We constructed our training and validation set by computing the gradient histogram features described in [25] for 5000 positive and 5000 negative randomly selected samples from two different volumes[2]. Similarly, a 30'000 path test set was generated.

From the training and validation set, half the samples were randomly selected to train a BS classifier with 500 stumps and the other half to learn the early stopping criteria. We repeated this 10 times and evaluated the resulting classifiers and corresponding stopping criteria on the test set. Fig. 5(*bottom*) depicts our results. In general, **EDE** delivers virtually the same accuracy as the non-early-stopping approach in a fifth of the time, and is more than twice as fast as **CCAL**, **DBP**, and **SPRT**.

Finally, in Fig. 6 we demonstrate that **EDE** performs well even for small validation sets, whereas other approaches require much larger ones. To this end, we trained a BS classifiers with 500 stumps and re-learned the thresholds and probability distributions using ever smaller validation sets. Even by the time the number of positives in the validation set has dropped from 2500 to 200, the performance of **EDE** is barely affected, while that of the others have degraded substantially.

5. Conclusion

This paper addressed the problem of speeding up the prediction of binary classification when using ensemble classifiers. Our proposed solution uses Bayesian inference to establish and track the probability of a samples class label as stage evaluations are computed. Our early stopping criteria is to terminate the prediction process when the uncertainty of the class label, measured by its Shannon entropy, falls below a chosen level. We showed through extensive experimentation on several classification tasks that our approach provides significant accuracy and speed improvements over state-of-the-art early stopping methods. In our future work, we plan to investigate the feasibility of using our approach in the context of SVM classifiers and the many kernels they may make use of.

[2]Volumes are available at: http://cvlab.epfl.ch/data/delin/

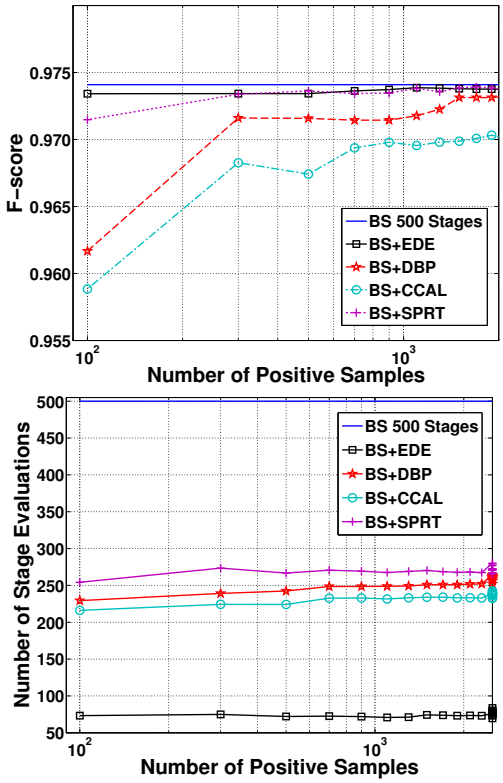

Figure 6. Size of the validation set. (*top*) Best F-score and (*bottom*) number of stage evaluations as a function of the number of positives used in the validation set on a logarithmic scale. **EDE** quickly reaches its peak performance whereas the other approaches require far more to do so.

Acknowledgements

This work was supported in part by the EU ERC Grant MicroNano. Francois Fleuret was supported in part by the European Community's Seventh Framework Programme under grant agreement No 247022 - MASH.

References

[1] S. Agarwal, A. Awan, and D. Roth. Learning to Detect Objects in Images via a Sparse, Part-Based Representation. *PAMI*, 26(11):1475–1490, 2004. 6

[2] K. Ali, F. Fleuret, D. Hasler, and P. Fua. A Real-Time Deformable Detector. *PAMI*, 34(2):225–239, 2012. 1, 5

[3] A. Bosch, A. Zisserman, and X. Munoz. Image Classification Using Random Forests and Ferns. In *ICCV*, 2007. 1

[4] L. Bourdev and J. Brandt. Robust Object Detection via Soft Cascade. In *CVPR*, pages 236–243, 2005. 1, 2, 3, 5

[5] L. Breiman. Random Forests. *Machine Learning*, 2001. 1, 2

[6] R. Caruana and A. Niculescu-Mizil. An Empirical Comparison of Supervised Learning Algorithms. In *ICML*, 2006. 1

[7] T. M. Cover and J. Thomas. *Elements of Information Theory*. Wiley Interscience Press, 1991. 4

[8] P. Dollár, R. Appel, and W. Kienzle. Crosstalk Cascades for Frame-Rate Pedestrian Detection. In *ECCV*, 2012. 1, 2

[9] Y. Freund and R. Schapire. A Short Introduction to Boosting, 1999. Journal of Japanese Society for Artificial Intelligence, 14(5):771-780. 1, 2

[10] J. Gall, A. Yao, N. Razavi, L. V. Gool, and V. Lempitsky. Hough Forests for Object Detection, Tracking, and Action Recognition. *PAMI*, 2011. 1

[11] P. Gehler and S. Nowozin. On Feature Combination for Multiclass Object Classification. In *CVPR*, 2009. 6

[12] G. Griffin, A. Holub, and P. Perona. Caltech-256 Object Category Dataset. Technical report, California Institute of Technology, 2007. 6

[13] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2001. 2

[14] J. Liu, J. Luo, and M. Shah. Recognizing Realistic Actions from Videos in the Wild. In *CVPR*, 2009. 1

[15] H. Luo. Optimization Design of Cascaded Classifiers. In *CVPR*, 2005. 2

[16] P. Minh-Tri, V.-D. Hoang, and C. Tat-Jen. Detection with multi-exit asymmetric boosting. In *CVPR*, 2008. 2

[17] F. Moosmann, E. Nowak, and F. Jurie. Randomized clustering forests for image classification. *PAMI*, 30(9):1632 – 1646, 2008. 2

[18] S. Paisitkriangkrai, C. Shen, and A. Henge. Sharing Features in Multi-Class Boosting via Group Sparsity. In *CVPR*, 2012. 1

[19] H. A. Rowley, S. Baluja, and T. Kanade. Rotation Invariant Neural Network-Based Face Detection. *JMLR*, page 963, 1998. 5

[20] M. J. Saberian and N. Vasconcelos. Learning optimal embedded cascades. *PAMI*, 34:2005–2018, 2012. 2

[21] A. Saffari, C. Leistner, J. Santner, M. Godec, and H. Bischof. On-line random forests. In *CVPR*, 2009. 1

[22] D. W. Scott. *Multivariate Density Estimation: Theory, Practice and Visualization*. John Wiley and Sons, 1992. 4

[23] J. Šochman and J. Matas. Waldboost - Learning for Time Constrained Sequential Detection. In *CVPR*, pages 150–157, 2005. 1, 2, 3, 4, 5

[24] R. Sznitman and B. Jedynak. Active Testing for Face Detection and Localization. *PAMI*, 32(10):1914–1920, 2010. 2

[25] E. Turetken, F. Benmansour, and P. Fua. Automated Reconstruction of Tree Structures Using Path Classifiers and Mixed Integer Programming. In *CVPR*, 2012. 7

[26] P. Viola and M. Jones. Robust Real-Time Face Detection. *IJCV*, 57(2):137–154, 2004. 1, 2

[27] P. Wang, J. Wang, G. Zeng, J. Feng, H. Zha, and S. Li. Salient Object Detection for Searched Web Images via Global Saliency. In *CVPR*, pages 3194–3201, 2012. 1

[28] B. Zeisl, C. Leistner, A. Saffari, and H. Bischof. On-Line Semi-Supervised Multiple-Instance Boosting. In *CVPR*, pages 1879–1888, 2010. 2

[29] C. Zhang and P. A. Viola. Multiple-Instance Pruning for Learning Efficient Cascade Detectors. In *NIPS*, 2007. 1, 2, 3, 5

[30] D. Zikic, B. Glocker, E. Konukoglu, J. Shotton, A. Criminisi, D. Ye, C. Demiralp, O. Thomas, T. Das, R. Jena, and S. Price. Context-Sensitive Classification Forests for Segmentation of Brain Tumor Tissues. In *MICCAI*, 2012. 1