Evaluation of the AdaBoost IPM

Jan Šochman

jan.sochman@cmp.felk.cvut.cz
tel.: +420 2 2435 5731

Center for Machine Perception
Prague, Czech Republic

TN-eTRIMS-CMP-01-2007
version 1.1

April 25, 2007

This document describes the evaluation of the AdaBoost image processing module (IPM) described in more detail in [5]. The structure of the evaluation is motivated by the project deliverable D2.5 description [1] and Wolfgang Förstner's presentation on the Performance Characteristics for Classification and Learning [2] from Hamburg Project Meeting in March 2007.

The AdaBoost IPM is evaluated, both in learning and classification phase, in terms of its (i) performance on the ground truth data provided by the teacher and (ii) in terms of the self-awareness of its abilities. The AdaBoost IPM is tested on two types of objects – T-style windows and triangular cornices.

In the following, the AdaBoost IPM is described briefly in Section 1 and the experimental evaluation is given in Section 2. Conclusion remarks and future plans are presented in Section 3.

1 Brief IPM description

The AdaBoost image processing module (IPM) was designed to work as a lower-level image interpretation module, working directly with images. Its outputs are confidence-rated hypotheses of positions of objects of interest in the image. A higher-level reasoning module (e.g. SCENIC) is expected to run the module, use its outputs for further reasoning and send "down" feedback on both learning and classification results of the IPM. This process can be repeated (reasoning loop) until satisfactory scene interpretation is obtained. A more detailed description of the design and the provided functionality of the IPM can be found in [5].

The module allows to train and apply to images a discrete AdaBoost classifier [3]. Currently, only gray-scale images are used for training and classification. Only rectified images of building facades are considered. The classifier uses Haar-like features in a manner similar to [7] except that the cascade is not build. To train a full cascaded classifier, very large training sets are required which is prohibitive in the eTRIMS project where the emphasis in put on the reasoning loop which may start from very small evidence, rather than training from a large datasets.

2 Experimental evaluation

The AdaBoost IPM provides both learning and classification operators to the higher-level reasoning modules (e.g. SCENIC) [5]. The evaluation thus needs to consider both learning and classification tasks. Quality of solution of both tasks is measured in terms of the module's performance on the ground truth data and the self-awareness of its performance.

	win	**T-win**	**+-win**	**†-win**	**corn**	**⌒-corn**	**△-corn**
train set	146	482	0	14	97	41	44
test set	556	268	25	82	106	49	51

Table 1: Summary of annotated datasets. For examples of annotated object types see Figure 1.

Figure 1: Examples of annotated objects. From left to right, top row: T-style window, †-style window, +-style window, example of "other" window, bottom row: △-style cornice, ⌒-style cornice, "other" cornice.

The **learning** part of the AdaBoost IPM needs labelled data (ground truth) as its input. To evaluate the ground truth performance of the learning process, the *training error* is thus a natural choice.

The self-awareness can be measured by the *upper bound on the training error* which is minimised by the AdaBoost learning, and the *error of the weak classifiers* added to the AdaBoost ensemble. If the error is close to 0.5, or the upper bound does not converge to zero, the training starts to be inefficient. In such case, the problem is too difficult for the IPM. Another measure which is of interest for the learning self-awareness evaluation is the *training samples margin*. Although, the margin is not directly measurable, the AdaBoost has been shown [4] to maximise the margin even after the training error drops to zero (i.e. it is worth to continue training even if the training error is zero).

For the **classification** evaluation of the AdaBoost IPM, the standard way is to use the *receiver operation curve* (ROC) as a ground truth performance measure. The AdaBoost classifier returns a confidence value of the object being of the given type. For the self-awareness evaluation, the *confidence value* is directly related to the classifier's trust in the returned hypothesis. The classifier should return higher confidence values on positive examples than on negative ones.

Two other measures for the classifiers robustness are presented. First, in the world of facades where the AdaBoost IPM is being applied, the confusion matrix is a measure showing the expected behaviour of the classifier on the objects of other types than the classified one. Second, the algorithm assumes rectified images. These images are currently obtained by semi-manual rectification tool with relatively high precision. However, an automatic rectification tool is being developed where the precision is expected to vary. A test on classifier sensitivity to image rotation is performed to test the robustness of the classifier to image transformations.

The following experiments have been done on two datasets of rectified classical facades (mostly baroque and pseudo-historical). The training dataset consists of 27 and test set of 45 annotated images. Table 1 summarises the number of annotated objects and an example of each type is depicted in Figure 1.

Classifiers for two object types – a T-style window and a triangular cornice – have been trained to evaluate the applicability of the AdaBoost learning approach. The first object type, T-style window, has clear structure and the AdaBoost working on Haar-like features should perform well for it. The triangular cornice object type shows the limits of the approach. Both

Figure 2: Positive examples from the training set for the T-style window detector.

object types have been chosen as important facade elements with interesting relations among objects of the same type (windows appear usually in rows) and also among objects of different types (a cornice is typically located above a window). Having detectors for such objects allows higher-level reasoning algorithms to study spatial relations of hypothesised entities.

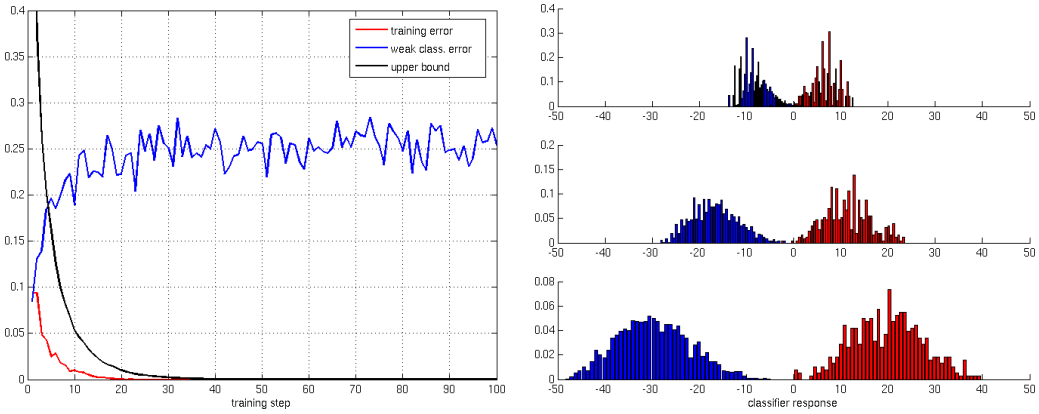
2.1 T-style windows

A training set consisting of 482 T-style windows from classical facades was used to train an AdaBoost classifier. The positive samples in the training set are depicted in Figure 2. Only the gray scale versions of the samples are used. To generate the positive training samples, a border of 20 % of the annotated window size has been added to the original annotation so that the samples contain small surrounding of the window. The negative part of the training set consists of 2000 random samples from the same set of facade images generated from inside of the facade (facade annotation was available) such that their size approximately corresponds to the true (annotated) windows and they do not overlay the ground truth windows more than by half of its area.

One hundred weak classifiers were combined into an AdaBoost T-style window classifier. The statistics of the training process are shown in Figure 3a. The training error drops to zero around the 20th training step, which indicates the task is relatively easy. Due to easiness of the learning task, useful weak classifiers (with error significantly below 0.5) are found even after the training error drops to zero. The effect of further training is depicted in Figure 3b. Although the positive and negative samples are already separated, the margin is further widened by adding more weak classifiers to the ensemble.

The resulting classifier have been tested on the test set described above (see Table 1). The ROC plot is shown in Figure 5a. Some of the detection results are shown in Figure 6. Note that the average confidence value is higher for the images with T-style windows than for the images without T-style windows. The ROC plot shows that about 95 % of the windows are correctly detected. This is important for initialising the reasoning loop. The better is the initialisation, the easier is to continue with further reasoning. Important is that the positions of the false positives in the detection examples in Figure 6 are mostly random as opposed to the correct detections. Knowing more about the scene at the higher reasoning levels where more cues are combined together, the false positives could be very likely removed. The left image in the middle row of Figure 6 also demonstrates robustness of the method to unremoved radial distortion.

Table 2 summarises the detection results on different window types (only for the maximum false positive ROC point). The successful detection rate on the general windows is probably due to high false positive rate and due to similar structures appearing more likely in window-like regions. Very high confusion percentage is obtained for †-style windows where the main (and only) difference is the crossbar dividing the top windowpane which is not captured by the classifier.



(a) (b)

Figure 3: Training process for the T-style window classifier. Left plot shows the training error, the upper bound and the error of lastly added weak classifier to the ensemble. The right plot shows the distribution of the classifier responses on positive (red) and negative (blue) samples for the training steps 20, 50 and 100 (from top to bottom).

win	**+-win**	**†-win**
56.3 %	56.0 %	76.5 %

Table 2: Percentage of different window types detected by the T-style window detector on the test set. For total number of examples in the test set see Table 1.

Another test has been performed to prove the classifier robustness to image rotation which may occasionally happen due to inaccurate rectification. One hundred annotated T-style windows from the test set have been rotated (the annotation stayed the same – non-rotated) and evaluated by the classifier. The result of the experiment is depicted in Figure 4a. The T-style window classifier can be reliably applied in the range -8 to $+8$ degrees which is sufficient for manual rectification. Nevertheless, further experiments will be needed when the automatic rectification tool is available (the tool is being developed currently).

Another example of incomplete rectification can be seen in the left image of middle row of Figure 6. Even though the radial distortion has not been removed, the detection results are comparable to those without radial distortion.

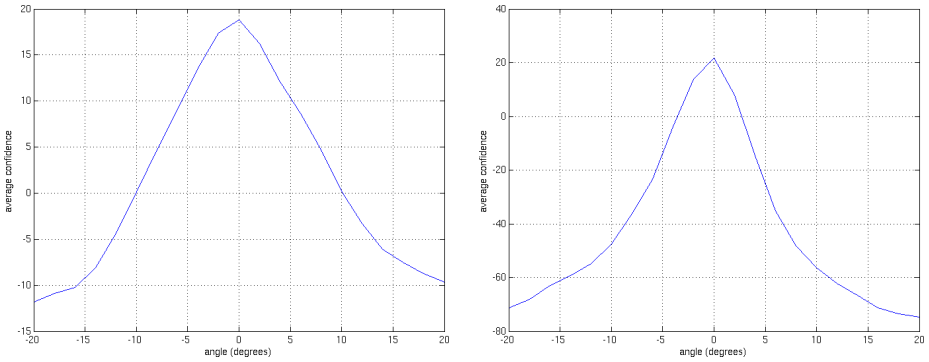
The speed of the classifier is about 10s on full resolution image (aprox. 1200×1500). This may be limiting while running the reasoning loop for more than a single iteration. Since the offline cascaded classifier [7] or time-optimised WaldBoost [6] need very large training sets, there is a need for *online time-optimised* version of AdaBoost algorithm.

2.2 Cornices

The training set for the △-style cornices consists of only 44 positive examples while 2000 negative examples are generated randomly from the facade images as for the T-style windows. One hundred weak classifiers are trained and combined into an ensemble.

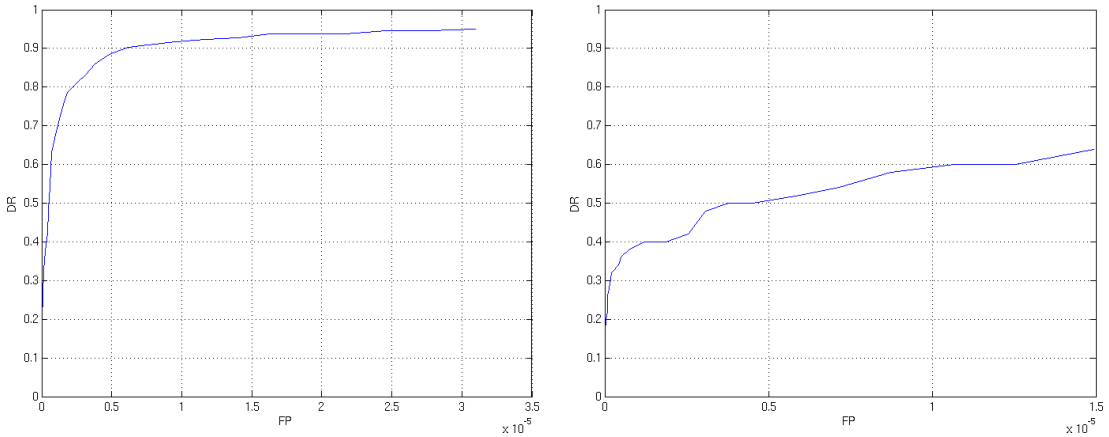
The training process statistics are shown in Figure 7. Again the learning task is very easy. Training error drops to zero in less than 10 training steps, the error of the lastly added weak classifier is even lower than in the case of T-style windows, and the margin is widened after the training error is drops to zero.

Nevertheless, after evaluating the classifier on the test set, see Figure 5b, the performance is significantly worse than for T-style windows. The reason for this discrepancy in the confidence of the classifier and its performance on the test dataset can be used by higher-level reasoning algorithm as a tool for learning good enough classifiers. If there is a strong high-level evidence



(a) (b)

Figure 4: Sensitivity of the (a) T-style window and (b) △-style cornice detector to rotation.



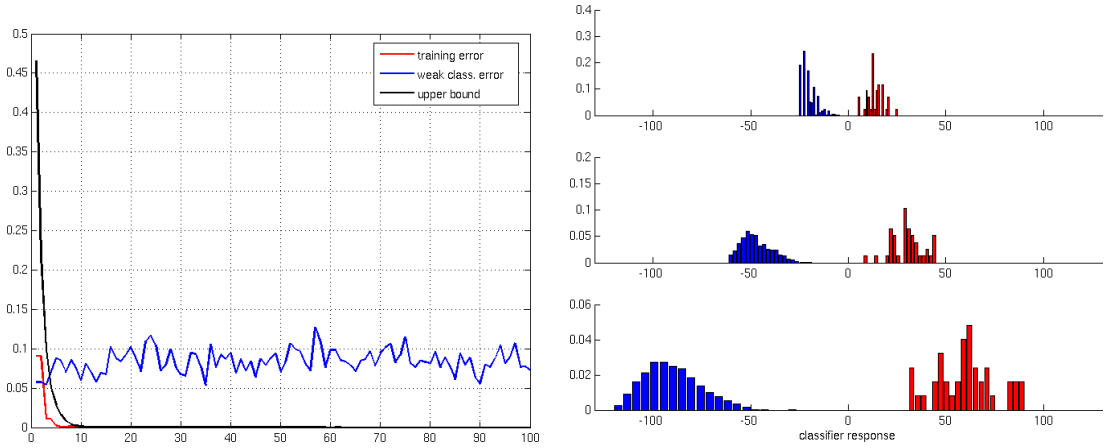
(a) (b)

Figure 5: ROC of (a) T-style window and (b) triangular cornice detector on the test dataset.

Figure 6: Examples of the T-style window detector output on images from the test set. The bottom row show the result when there are no T-style windows in the facade. Average confidence value for the images (left to right, top to bottom), T-style windows present: 16.4, 14.9, 16.4, 15.7, no T-style windows: 12.6, 14.1. Note that the radial distortion is not removed from the left image in the middle row and still good results are obtained.

general cornice	⌒-style cornice
3.9 %	14.3 %

Table 3: Percentage of different cornice types detected by the △-style cornice detector on the test set. For total number of examples in the test set see Table 1.



(a) (b)

Figure 7: Training process for the △-style cornice classifier. Left plot shows the training error, the upper bound and the error of lastly added weak classifier to the ensemble. The right plot shows the distribution of the classifier responses on positive (red) and negative (blue) samples for the training steps 20, 50 and 100 (from top to bottom).

for an object in an image and the IPM does not "see" this object and still the learning statistics were rather confident, the "unseen" object is a good example for further training. On the other hand, if the learning algorithm statistics show that the problem is already difficult, no further training will help and the higher-level reasoning algorithm has to use another IPM to support his hypothesis.

Table 3 summarises the confusion percentage for another types of cornices detected by the △-style cornice detector. The table shows that the △-style cornices can be reasonably well distinguished from all the other cornice types.

The △-style classifier is slightly less robust to the image rotation (see Figure 4b). The range where the results are not degraded too significantly is approximately -3 to $+3$ degrees. This range is very likely at the edge of usability for manual rectification.

The speed of the classifier is again in order of seconds. For improvement see the discussion in preceding section.

3 Conclusions

The AdaBoost IPM has been tested on two types of objects – T-style windows and triangular cornices. The evaluation examined the module's performance in terms of (i) precision on ground-truth data, and (ii) self-awareness of its own abilities for both, learning and classification. These measures can be used by the higher-level reasoning algorithms to control the quality of IPM's outputs and to test its usability for a given task.

The experiments shows that a reliable detector can be trained if sufficient amount of training data is available. The experiment with △-style cornices outlines the way, the higher reasoning level can use the measures provided by the AdaBoost IPM to check its expected performance and to improve its real performance. First experiments on combination of the AdaBoost IPM with higher-level reasoning modules are described in deliverable D1.2.

References

[1] eTRIMS IST 027113 Annex 1 Final Version EC approved on 23 September 2005.

[2] Wolfgang Förstner. Performance characteristics for classification and learning. Hamburg meeting talk, March 2007.

[3] Y. Freund and Schapire R.E. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, August 1997.

[4] Robert E. Schapire, Yoav Freund, Peter Bartlett, and Wee Sun Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *The Annals of Statistics*, 26(5):1651–1686, 1998.

[5] Jan Šochman. Specification of the AdaBoost IPM for use in SCENIC. Technical Report TN-eTRIMS-CMP-01-2006, Center for Machine Perception, Prague, Czech Republic, January 11 2007.

[6] Jan Šochman and Jiří Matas. WaldBoost - Learning for Time Constrained Sequential Detection. In *CVPR*, volume 2, pages 150–157, Los Alamitos, USA, June 2005.

[7] P. Viola and M.J. Jones. Robust real time object detection. In *SCTV*, Vancouver, Canada, 2001.