Absolute contrasts in face detection with AdaBoost cascade

Marcin Wojnarski

Warsaw University, Faculty of Mathematics, Informatics and Mechanics
ul. Banacha 2, 02-097 Warszawa, Poland
mwojnars@ns.onet.pl

Abstract. Object detection using AdaBoost cascade classifier was introduced by Viola and Jones in December 2001. This paper presents a modification of their method which allows to obtain even 4-fold decrease in false rejection rate, keeping false acceptance rate – as well as the classifier size and training time – at the same level. Such an improvement is achieved by extending original family of weak classifiers, which is searched through in every step of AdaBoost algorithm, with classifiers calculating *absolute* value of contrast.
Test results given in the paper come from a face localization problem, but the idea of absolute contrasts can be applied to detection of other types of objects, as well.

1 Introduction

The original AdaBoost method is known from the late 1980s as a multiclassifier and a training procedure for a collection of *weak classifiers* (called also *features*), having success rate of about 0.5, to boost them by suitable voting process to very high level of performance [2, 3]. Although this training scheme gives a classifier of very good accuracy, the number of simple features used is far too big, making real-time applications impossible [1].

The AdaBoost cascade method proposed by Viola and Jones [1] solves this problem. Viola and Jones connected a number of strong classifiers built with standard AdaBoost algorithm in a sequence, to form a *cascade* of classifiers of increasing complexity. Every stage of the cascade either rejects the analyzed window or passes it to the next stage. Only the last stage may finally accept the window. Thus, to be accepted, a window must pass through the whole cascade, but rejection may happen at any stage.

During detection, most sub-windows of the analyzed image are very easy to reject, so they are rejected at very early stage and do not have to pass the whole cascade. In this way, the average processing time of a single sub-window can be even thousands times lower than in the case of a standard AdaBoost classifier, particularly because the first stages can be very small and fast, and only the last ones have to be large.

Viola and Jones [1] proposed also a set of features for use particularly in face detection (but easily applicable to detection of other objects). This paper presents an extension of that family of features, after which the AdaBoost algorithm – both the original and the cascade one – yields a classifier of much better accuracy. Original features given by Viola and Jones [1] are described in section 2.1, and the extension is presented in section 3.

2 AdaBoost cascade algorithm

2.1 The weak classifier

For each detection window o of the image being processed, a weak classifier gives a decision $\delta_\omega(o) \in \{-1, +1\}$ indicating membership of the image o to one of two classes, labelled by -1 (negative, e.g. a non-face) and +1 (positive, e.g. a face).

The classifier first calculates a *region contrast* $c(R)$ of the window o:

$$c(R) = \sum_{(x,y)\in R^+} o(x,y) - \sum_{(x,y)\in R^-} o(x,y) \ , \tag{1}$$

where R is a sub-window of the window o, composed of a *positive sub-region* R^+ and a *negative sub-region* R^-. After computing the contrast, the classifier gives a response of $+1$ if $c(R) >= \theta$ or -1 otherwise. Here, R and θ are parameters of the weak classifier.

There are four types of regions R, presented in Figure 1. Positive sub-region R^+ is drawn in white and negative sub-region R^- is drawn in black. The main advantage of these sub-regions is that they have rectangular shape, so they can be computed very rapidly (in constant time) using an *integral image* representation of the original image. The integral image is computed only once, at the beginning of object detection in a specified image.

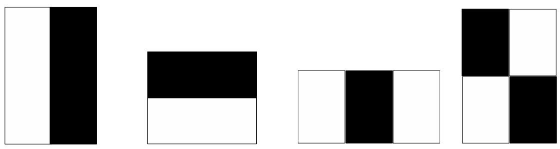

Fig. 1. Types of regions used in weak classifiers. White indicates positive sub-region and black negative sub-region

The contrast region R is parameterized not only by its type t ($t \in \{A, B, C, D\}$), but also by four integral parameters: position (x, y) of its upper-left corner relative to the detection window, width a and height b of sub-regions (all sub-regions of a given region have the same size). Note that the family of all possible features is very large, e.g. in the experiments described further in the paper this family was composed of about 160000 elements. Certainly, only small subset of this family is used in a final strong classifier – AdaBoost algorithm is used to choose the best subset.

2.2 The strong classifier

A strong classifier is composed of a number of weak classifiers. Its decision is made by weighted voting: decision of a t-th weak classifier is multiplied by a weight α_t:

$$\gamma_t(o) = \delta_t(o) * \alpha_t$$

and all values $\gamma_t(o)$ are summed up and compared with a threshold Θ to form a final decision. Usually $\Theta = 0$, but when strong classifiers are further connected into a cascade, their thresholds can be different, in order to obtain required rate of false acceptance/rejection at every stage.

The AdaBoost algorithm is used to choose the most discriminative subset of all possible features and to set values of α_t. The algorithm works in an incremental manner, finding consecutive weak classifiers one by one. In every step, the algorithm looks through the whole family of contrast regions R to choose the best one – this is a simple exhaustive search method. However, to evaluate a weak classifier, its threshold θ has to be set, as well, not only the region R. Thus, for *every* possible R the best θ has to be found, which takes $N \log N$ time, where N is the number of training images. Certainly, this procedure is very time-consuming, usually it takes several minutes to choose a next weak classifier.

Every image in the training set has an associated weight, which is a positive real number. The weights are used during evaluation of weak classifiers: images with bigger weights are more important and have more influence on which classifier will be chosen next. The weights get changed after every step of AdaBoost procedure – in this way the successive classifiers found by the algorithm can be different (but not necessarily have to be).

Initially the weights $w_{i,t}$ are equal and sums up to 1:

$$w_{i,1} = \frac{1}{N}$$

for every image $i = 1, \ldots, N$. When the next weak classifier is found, the weights are modified and normalized to sum up to 1:

$$v_{i,t+1} = w_{i,t} e^{-\gamma_t(o_i) y_i} \ ,$$

$$w_{i,t+1} = \frac{v_{i,t+1}}{\sum_{i=1}^{N} v_{i,t+1}} \ ,$$

where $\gamma_t(o_i)$ is a decision value of the recently found classifier and y_i is a true classification ($+1$ or -1) of the i-th image.

2.3 The cascade

Every stage of a cascade is built using the simple AdaBoost algorithm. When the next stage is created, its threshold Θ is set to the biggest value which still guarantees that the false rejection rate is below a predefined level – this is evaluated on a separate data set (evaluation set), not the training one.

Before the next stage can be created, both the training and the evaluation set must be filtered: images rejected by the new stage have to be removed. In consequence, new negative images have to be generated, as their number would drop roughly by half at every stage.

3 New type of weak classifiers

The family of weak classifiers used in the algorithm proposed by Viola and Jones has a disadvantage. Namely, the contrast computed by a weak classifier, eq. (1), depends on *which* exactly sub-region (R^+ or R^-) is darker, so the weak classifier discriminates between windows with R^+ darker than R^- and R^- darker than R^+.

However, in many cases we would like to discriminate between windows with *the same* or *different* intensity in R^+ and R^-, ignoring information of *which* exactly sub-region is darker. That is because in real images what is important is the existence or lack of an intensity difference, and not the exact sign of the difference. For example, in face localization one may encounter a dark face on a bright background as well as a bright face on a dark background, so a classifier detecting whether there is a difference in intensity would be more discriminative than the one detecting a sign of the difference.

Table 1. Test error rates of face classifiers. Every classifier is characterized by its size (number of weak classifiers at each stage of a cascade) and number of training images (positive+negative examples). The first four classifiers are strong ones, the latter are cascades. FA – false acceptance rate, FR - false rejection rate. ABS – a classifier was built using extended family of weak classifiers, comprising both standard classifiers and the ones computing absolute contrast. In this case, the number of weak classifiers chosen from the absolute-contrast family is given in parentheses.

Classifier	FA	FR
20 weak, 500+500 images	1.4%	9.0%
20 weak, 500+500 images, ABS (4)	0.6%	8.2%
100 weak, 1000+1000 images	0.3%	3.2%
100 weak, 1000+1000 images, ABS (49)	0.4%	0.9%
4+4+10+20, 1000+1000	0.086%	13.1%
4+4+10+20, 1000+1000, ABS (1+2+5+12)	0.024%	14.2%
5+15+30+50+100+200, 1500+1000	0.00120%	4.4%
5+15+30+50+100+200, 1500+1000, ABS (1+6+11+25+50+120)	0.00028%	4.5%

For this reason, we extended the family of weak classifiers with the ones computing *absolute* values of contrasts:

$$c(R) = \left| \sum_{(x,y)\in R^+} o(x,y) - \sum_{(x,y)\in R^-} o(x,y) \right| \tag{2}$$

3.1 Test results

The results of strong and cascaded classifiers of different size built of the extended family of weak classifiers, compared to the ones using original family alone, are shown in Table 1. The results come from a face localization problem, but the idea of absolute contrasts can be applied in detection of other types of objects, as well. Two types of errors – false acceptance and false rejection – are considered separately, as in practical applications the former should be hundreds times lower than the latter.

Images of faces came from MPEG-7 and Altkom databases. MPEG-7 contained images of 635 persons, 5 for each subject (3175 in total). Altkom was composed of images of 80 persons, 15 for each subject (1200 in total). The images had 46×56 pixels and contained frontal or slightly rotated faces with fixed eye positions, with varying facial expressions and under different lighting conditions.

The number of positive training examples in each experiment is given in Table 1. Test sets contained the same number of faces as training ones. A validation set was used to find thresholds of strong classifiers in a cascade. All these sets were disjoint in each experiment (i.e. contained images of different persons) and included the same proportion of images from MPEG-7 and Altkom databases.

Negative examples were generated randomly from 10000 large images not containing faces, in on-line fashion. They had to be generated after creation of every new stage of a cascade, so as to compensate for examples correctly rejected by the last stage.

Exemplary face images used in the experiments are shown in Figure 2.

In order to speed up the training, only weak classifiers of even positions and sizes were considered (this applied both to standard and absolute classifiers). This is almost equivalent to scaling down the images by a factor of 2, but has the advantage of not introducing rounding errors.

Results from Table 1 show that using the extended family allows to achieve over 4-fold decrease in one type of error rate (e.g., false rejection), keeping the other one at a similar level, and without a need of increasing classifier size.

It is worth mentioning that using the absolute-contrast classifiers alone, without the standard ones, gives worse results than using the original family alone, so it is good to *extend* the original family, but not to *replace* it with the absolute-contrast one.

Table 1 (first column, in parentheses) contains also information about the number of weak classifiers which were chosen by AdaBoost algorithm from the

Fig. 2. Positive examples (faces) used in the training process

absolute-contrast family, at every stage of a cascade. Comparison of this information with the total size of each stage shows that the contribution of absolute-contrast classifiers rises with the size of a stage, from 20 or 25% at the first stage to 60% at the last one. This suggests that absolute contrasts are more useful when the problem becomes more difficult.

3.2 Efficiency of the cascade and the training process

Calculation of a decision of an absolute-contrast weak classifier takes the same amount of time as in the case of a standard classifier, since the only additional operation is a computation of an absolute value. Therefore, the final strong or cascaded classifier is as fast as the one built of original weak classifiers alone.

The use of the extended family does not slow down significantly the training algorithm either. Although the family is twice bigger, searching through it takes (if properly implemented) at most 10% longer. That is because the most time-consuming part of the search through original family is a quest for the optimal threshold θ, given a type, a size and a position of a sub-window (t, a, b, x, y). This requires calculating and sorting of contrasts of a given sub-window on all training images (say N), which takes time of the order of $N \log N$.

When absolute contrasts are also considered, additionally a sorted sequence of absolute values of contrasts is needed. However, this does not require computations of $N \log N$ complexity again, because a sorted sequence of contrasts is already available, and after transformation by absolute-value function this sequence turns into two sorted sequences, which can be merged in linear time. It should be noted here that construction of a cascade is a very time-consuming process, which takes from several hours to several days when executed on a personal computer (CPU 2.0 GHz), so time efficiency of the presented modification is an important feature.

4 Conclusion

The paper presented a modification of Viola and Jones' object detection algorithm. The modified algorithm utilizes an extended family of features which is searched through during construction of strong classifiers. This extension enables 4-fold decrease in false rejection rate without increase in false acceptance rate or classifier size. Moreover, the extension does not influence significantly training time, despite the fact that the family of features is twice bigger. Obviously, resolution of training images does not have to be increased either.

Acknowledgements

The research has been supported by the grant 3T11C00226 from Ministry of Scientific Research and Information Technology of the Republic of Poland.

References

1. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: IEEE Computer Vision and Pattern Recognition. Volume 1. (2001) 511–518
2. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences **55**(1) (1997) 119–139
3. Schapire, R.E., Freund, Y., Bartlett, P., Lee, W.S.: Boosting the margin: a new explanation for the effectiveness of voting methods. In: Proc. 14th International Conference on Machine Learning, Morgan Kaufmann (1997) 322–330
4. Xiao, R., Li, M.J., Zhang, H.J.: Robust multipose face detection in images. IEEE Trans. Circuits and Systems for Video Technology **14**(1) (2004) 31–41
5. Skarbek, W., Kucharski, K.: Image object localization by adaboost classifier. In Campilho, A.C., Kamel, M.S., eds.: ICIAR (2). Volume 3212 of Lecture Notes in Computer Science., Springer (2004) 511–518
6. Papageorgiou, C.P., Oren, M., Poggio, T.: A general framework for object detection. In: International Conference on Computer Vision. (1998) 555–562