

Symmetry Enhanced Adaboost

Florian Baumann, Katharina Ernst, Arne Ehlers, Bodo Rosenhahn

Institut für Informationsverarbeitung
Leibniz Universität Hannover
Appelstraße 9a, 30167 Hannover, Germany

Abstract. This paper describes a method to minimize the immense training time of the conventional Adaboost learning algorithm in object detection by reducing the sampling area. A new algorithm with respect to the geometric and accordingly the symmetric relations of the analyzed object is presented. Symmetry enhanced Adaboost (SEAdaboost) can limit the scanning area enormously, depending on the degree of the objects symmetry, while it maintains the detection rate. SEAdaboost allows to take advantage of the symmetric characteristics of an object by concentrating on corresponding symmetry features during the detection of weak classifiers. In our experiments we gain 39% reduced training time (in average) with slightly increasing detection rates (up to 2.4% and up to 6% depending on the object class) compared to the conventional Adaboost algorithm.

1 Introduction

This paper introduces an extension of the object detection framework proposed by Viola and Jones [1] in which the well established Adaboost algorithm [2, 3] is utilized to form a strong classifier.

Adaboost in one of its various forms is widely used in many tasks of object detection. Applied by the framework of Viola and Jones it allows high detection rates in real-time classification. Because of its economic consumption of processing power it is a good choice for applications, such as face detection, on mobile devices like digital cameras and camera phones. One drawback of the algorithm is the extensive training phase, due to the huge number of weak classifiers and the iterative training.

Our contribution is to reduce this problem by exploiting symmetric object characteristics, since symmetries can be observed all over nature [4], see also Figure 1. Not only constructed objects (e.g. buildings, cars, etc.) have axes of symmetry but also humans, animals and plants. We want to take advantage of this phenomenon by exploiting this fact in the learning and detection phase of the conventional Adaboost algorithm. In the experiments we demonstrate the applicability of our method on two example scenarios. The first scenario is the optical inspection of SMD-components while our second scenario approaches face detection.

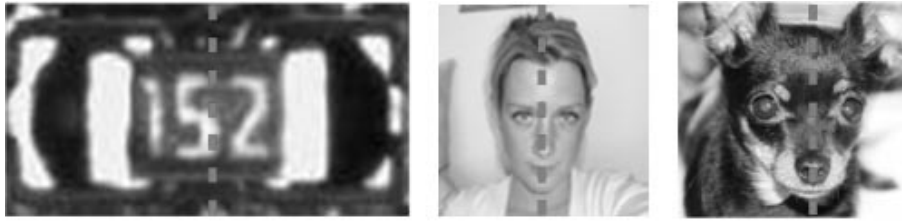


Fig. 1. Objects with symmetric characteristics: left: SMD-component symmetry, middle: face symmetry, right: animal symmetry

This paper is structured as follows: In the next section all required tools for boosting are briefly described. In section 3 the modified symmetry-enhanced Adaboost algorithm will be defined in detail. The final section presents the experimental results on SMD-components and face detection and a comparison to the conventional Adaboost algorithm for object detection as proposed by Viola and Jones [1].

2 Adaboost Algorithm

Adaptive Boosting (short: Adaboost), a machine learning algorithm proposed by Freund and Schapire [2, 3] is the foundation for our work. This algorithm is based upon the boosting theory of Kearns developed in 1988 [5]. In collaboration with Vazirani [6] the theory was enhanced and finally elaborated by Freund and Schapire in 1995 to the popular Adaboost algorithm.

2.1 Adaboost by Freund & Schapire

Adaboost combines multiple weak classifiers to one strong classifier in the fashion of a weighted sum. These weights are determined during a round-based training phase. In each round a given weak classifier is evaluated on the training set resulting in higher weights for better performing classifiers. An important aspect of Adaboost is that also weights are assigned to the training examples. These weights are adapted after each round to increase the influence of incorrectly classified examples. The only required inputs besides the number of rounds in the training phase are a collection of positive and negative image examples as well as the set of weak classifiers.

Adaboost yields good results in various object detection applications but has the shortcoming of an extensive training phase.

Several variants of boosting algorithms have been developed in the last years, e.g. LPBoost, SoftBoost, MILBoost, FloatBoost or S-Adaboost [7–11]. LPBoost focuses on minimizing the generalization error by maximizing the soft margin. Its drawback is to determine a good upper bound for the number of iterations.

SoftBoost [8] overcomes this problem by applying a relative entropy regularization yielding in an upper bound that is logarithmic in the number of examples. But the generalization error of SoftBoost decreases only slowly at the beginning of the training. Entropy Regularized LPBoost [7] adds the same relative entropy regularization to LPBoost and thus combines its high performance with a good upper bound. In contrast to Adaboost MILBoost [9] introduces a cost function which is combined with the feature selection. A backtrack mechanism is provided by FloatBoost [10] that allows to remove weak classifiers having low performance after each iteration step of Adaboost. In S-Adaboost [11] the Divide and Conquer Principle is utilized to allow Adaboost to divide the input space into subspaces of e.g different classification difficulty and to classify them using dedicated strong classifiers. A non-linear combination of these classifiers joins the advantages of simpler but more general and stronger but more specific classifiers.

Whereas these methods concentrate on the weighting functions iteration [8, 7] and feature analysis [9, 10], we propose to integrate geometric prior information in the feature selection to speed up the training phase. Assuming a symmetry property in the training of object classes is a simple but effective way to reduce the enormous sampling size and to exploit properties of natural objects. Therefore, our method is completely different to earlier approached modifications. But our contribution can easily be applied to the above mentioned adaptations of Adaboost to further enhance the performance.

2.2 Framework by Viola & Jones

Based on the algorithm of Freund and Schapire [2], Viola and Jones introduced Adaboost into the field of face detection [1]. They boosted weak classifiers using Haar-like features and the representation by integral images. Viola and Jones achieved excellent results, furthermore their framework is working in real-time. The name Haar-like feature comes from their similarity to Wavelet transformations with Haar basis functions. Haar-like features are simple rectangle features, which are a good choice to represent weak classifiers in boosting. Each feature is described by a position (offset) and a +1/-1-profile. Figure 2 shows some typical features for object detection.

The integral image offers a fast computation of rectangle feature values. For each image location (x,y) the sum of pixels above and to the left of (x,y) is pre-calculated and stored. Using integral images the Haar-like features can be computed in an very efficient way.

To deal with the huge amount of weak classifiers in the training phase they proposed an algorithm that searches in each training round over the set of possible weak classifiers and selects the one with the lowest classification error by passing only a single time through a sorted list. For more information about the feature selection mechanism see [1].

3 Symmetry enhanced Adaboost

3.1 Comprising of symmetric object characteristics by inserting symmetry features

Many natural objects have symmetric characteristics. Faces can be divided into two halves by inserting one symmetry axis. Other objects from nature for example flowers, butterflies or snowflakes can be divided into several segments. Even viruses are showing a very high symmetry in several axes. Regarding SMD-Components we can divide each picture into four quadrants by inserting two symmetry axes. We want to benefit from this symmetry to reduce the training time and to stabilize the detection rate of Adaboost. We make the assumption that each selected classifier in an image section implicates at least one symmetric classifier. Reflected along one of the symmetry axes this symmetric classifier should hold a similar low classification error. In that way the best classifier selected in each round is used to detect a symmetric classifier in a reduced, mirrored search area. Furthermore to refuse classifiers providing higher error rates we apply a threshold to assure that the symmetric classifier produces a classification error nearby the original feature. By setting the threshold in our experiments to 0.025 the classification error of the symmetric classifier is only allowed to be 0.025 higher than the error of the original classifier. The classification error is computed as the sum of the weights of the misclassified examples images. As these weights are normalized, the classification error is within the range $[0, 1]$. It must be taken into consideration that the position of the object differs from

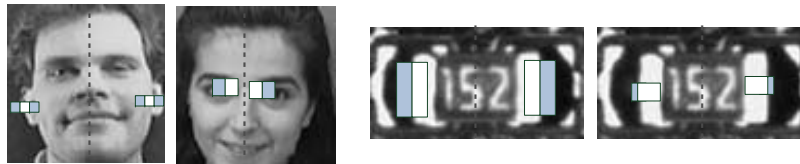


Fig. 2. Examples of a detected feature and its corresponding symmetry feature.

picture to picture in little variabilities. Thus a variance in x- and y-direction is inserted. This variance extends the sampling area of the symmetric object area to a few pixel in both directions.

Hence a possible deviation in the position of the object can be adjusted, and the algorithm has more flexibility in choosing the best corresponding classifier. By this idea the symmetric characteristics of objects are utilized to reduce the sampling area. The sampling of only half or quarter of the image suffices in our application, while features are placed in the other image area by computing the symmetries.

The reflected features are based on features found in the reduced sampling area. In our application we use one symmetric axis in the vertical position for SMD- and face detection. In order to get the corresponding symmetric features we

must reflect only the vertical features. Line- and horizontal features can stay in their configuration. But in the case of two symmetric axes we need to apply a reflection to vertical and horizontal features. Figure 2 shows four examples of a feature and its corresponding symmetry feature detected by SEAdaboost in faces and SMD-Components.

3.2 Modified Algorithm

Based on the notation in [1] the SEAdaboost algorithm can be summarized as follows:

- **Given:** Labeled images $(x_1, y_1), \dots, (x_m, y_m)$, with $y_j \in \{1, 0\}$
- **Select** optimal symmetry axis of images $(x_1, y_1), \dots, (x_m, y_m)$
- **Initialize** weights $w_{1,j} = \frac{1}{2n}, \frac{1}{2l}$ if $y_j = 0, 1$ respectively, with n denoting the number of negative and l of positive images,
- For round number $r = 1, \dots, R$:
 1. **Normalize** weights $w_{r,j}$:

$$w_{r,j} = \frac{w_{r,j}}{\sum_{k=1}^m w_{r,k}} \quad (1)$$

2. **Select** best weak classifier in the image region limited by the symmetric axis with respect to the error:

$$\kappa_r = \min \sum_k w_k |h(x_k, f, p, \tau) - y_k| \quad (2)$$

3. **Define** $h_r(x)$ with minimal error κ_r :

$$h_r(x) = h(x, f_r, p_r, \tau_r) \quad (3)$$

4. **Select** best symmetric weak classifier with respect to the error and the variance ψ in x - and y -direction:

$$\kappa_{r,sym} = \min \sum_k w_k |h(x_k, f, p, \tau, \psi) - y_k| \quad (4)$$

5. **Define** $h_{r,sym}(x)$ with minimal error $\kappa_{r,sym}$ and respect to a threshold ρ :

$$h_{r,sym}(x) = h(x, f_{r,sym}, p_{r,sym}, \tau_{r,sym}, \psi, \rho) \quad (5)$$

6. **Update** the weights:

$$w_{r+1,j} = w_{r,j} \cdot \chi_r^{1-\lambda_j} \cdot \chi_{r,sym}^{(1-\lambda_j)\beta_r}, \quad (6)$$

$$\chi_r = \frac{\kappa_r}{1 - \kappa_r} \quad \text{and} \quad \chi_{r,sym} = \frac{\kappa_{r,sym}}{1 - \kappa_{r,sym}}$$

with

$$\lambda_j = \begin{cases} 0 & , \text{ if } x_j \text{ classified correctly by the weak classifier} \\ 1 & , \text{ else.} \end{cases}$$

and with

$$\lambda_{j,sym} = \begin{cases} 0 & , \text{ if } x_j \text{ classified correctly by the symmetric weak classifier} \\ 1 & , \text{ else.} \end{cases}$$

and with

$$\beta_r = \begin{cases} 1 & , \text{ if } \kappa_{r,sym} < \kappa_r + \rho \\ 0 & , \text{ else.} \end{cases}$$

– The strong classifier is:

$$Z(x) = \begin{cases} 1 & , \text{ if } \sum_{r=1}^R \delta_r h_r(x) + \delta_{r,sym} h_{r,sym}(x) \geq \frac{1}{2} \sum_{r=1}^R \delta_r + \delta_{r,sym} \\ 0 & , \text{ else.} \end{cases} \quad (7)$$

$$\text{with } \delta_r = \log \frac{1}{\chi_r}, \quad \delta_{r,sym} = \beta_r \log \frac{1}{\chi_{r,sym}}$$

After preprocessing and selecting the optimal symmetry axes the labeled image examples are initialized, i.e. the weights of each positive and negative image example are set with respect to the respective training set size. In the beginning of each round the image weights are normalized (1).

Afterwards the algorithm searches in every round for the best weak classifier, in an area limited by at least one symmetric axis (2). Searching means to select the best classifier that splits the images in two labeled groups and thereby produces the lowest error. One great advantage of this mechanism is that we only need to pass a single time through a sorted list by calculating an error based on the image weights. More information about the feature selection mechanism is given in [1].

Once a weak classifier is detected and defined (3), the regarding feature is reflected and moved to the symmetric area (4). In this area we search with respect to a variance for the corresponding symmetric classifier. As shown in Figure 2 the symmetric classifiers can be moved from the reflected position by several pixels.

Optimally, the algorithm selects two weak classifiers each round, whereas the selection of the symmetry classifier is subject to restrictions. This classifier should only be selected and defined (4,5), if the produced classification error is at maximum given by the threshold ρ added to the error of the already selected classifier. If the classifier cannot fulfill this condition, it is rejected. In that way the optimal weak classifiers with respect to the weights of the example images are selected. In the next step the weights of the example images are updated regarding its classification success by the newly selected weak classifiers. This provides the opportunity to prefer incorrect classified images in the following rounds. After the step of updating the weights the next round starts.

Finally after the expiration of round R the algorithm forms a strong classifier

by a linear combination of the selected (conventional and symmetric) weak classifiers. Each weak classifier is weighted by its own produced error in (2) and (4). The strong classifier can be applied to unseen images.

4 Experimental Results

We evaluate SEAdaboost in two fields of application, face detection and the optical inspection of SMD-components. Whereas for faces, the training data are obviously faces vs. non-faces, for SMD-components it is different, since we want to detect wrong positioned or missing components, as well as components with defect solder joints, see Figure 3.



Fig. 3. Detection of different faces (credits to AT&T Laboratories, Cambridge) and non-faces / SMD-components and SMD-components with different kinds of failures.

One challenge in the classification of SMD-components is given by image examples, which contain reflections of other components nearby. The difficulty in the training of these examples lies in classifying defects of solder joints correctly, while detecting reflections as those. With regard to the face detection we need to locate faces in different poses or with different expressions. Furthermore our method should be robust to changing backgrounds or illumination/lighting disparities. For the face detection training we use the AT&T database of faces. Therefore we give credits to AT&T Laboratories Cambridge. The database contains ten different images each of 40 people varying in the lighting, facial expressions (open / closed eyes, smiling / not smiling) and facial details (glasses / no glasses). The images were taken against a dark homogeneous background with the people in an upright, frontal position.

We divided our image bases into a training and validation set using a 67/33 ratio. Crowther and Cox [12] illustrated that especially for small bases like our set of SMD-components a split containing only a small part for validation is not recommendable. They suggested to select a ratio between 50/50 and 70/30.

We demonstrate the difference between the classification results of the conventional Adaboost and SEAdaboost, i.e. we show the minimization of training time in Table 1.

Table 1. Experimental results of face- and SMD-detection in comparison. The SEAdaboost achieves the highest detection rate while the training time reduces significantly.

Application	Algorithm	Training time	Hit-rate	Rounds
Face	Adaboost	34.0 hours	89.56%	18
Face	SEAdaboost	21.0 hours	90.72%	18
SMD	Adaboost	36.0 hours	91.6%	10
SMD	SEAdaboost	23.0 hours	97.5%	10

By incorporating the idea of symmetry in the conventional Adaboost algorithm and the thereby reduced sampling area, Symmetry enhanced Adaboost effectively stabilizes learning results. In the scenario of face detection our method increases the detection rate slightly between 0.2% and 2.5% and in application on SMD-components between 2.4% and 6.0% depending on the number of training rounds.

The dependency of the detection rate from the number of training rounds respectively training time is shown for both object classes in Figure 4 and 5. The test results indicate significantly, that the detection rate increases with a higher number of training rounds for all implemented algorithms.

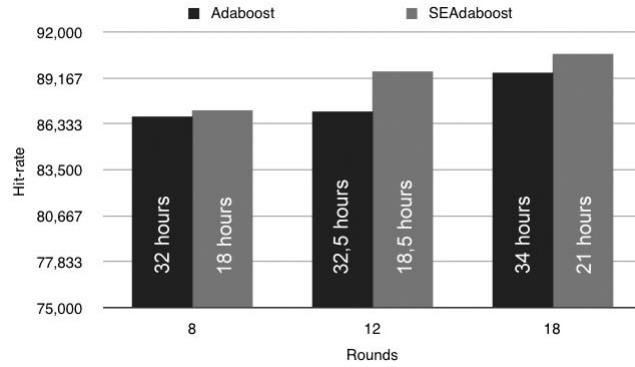


Fig. 4. Rounds/training time vs. detection rate applied to faces. With a higher number of rounds, the detection rate increases.

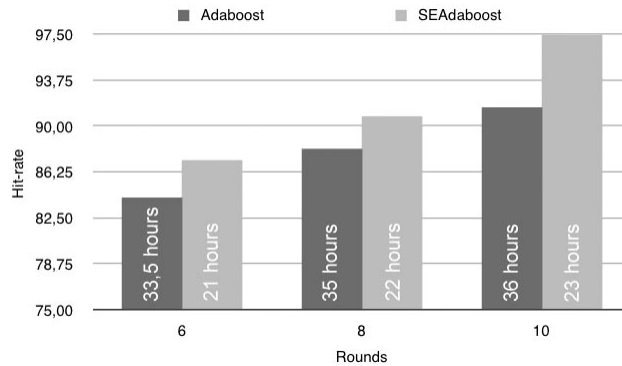


Fig. 5. Rounds/training time vs. detection rate applied to SMD-components. With a higher number of rounds, the detection rate increases.

The sets of corresponding symmetry classifiers allow the detection of objects by smaller features which can be fitted tighter to the significant regions in images. Figure 6 demonstrates that the region around the eyes is represented by the symmetric features more precisely than by a conventional three-rectangle feature. Also small rotations of the object can be better compensated.

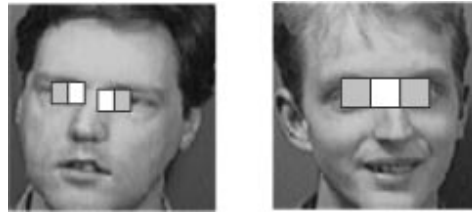


Fig. 6. Regions around the eye detected by SEAdaboost exploiting a pair of symmetric two-rectangle features and by Adaboost using a conventional three-rectangle feature.

The training time was lowered significantly by almost 40%. In spite of the reduction of the object sampling area, SEAdaboost further improves the classification results. Obviously the conventional Adaboost algorithm would achieve the same detection accuracy by increasing the number of training rounds, but with the drawback of an extended training time.

5 Conclusions

In this paper Symmetry enhanced Adaboost (SEAdaboost) was described, a method to minimize the immense training time of the conventional Adaboost al-

gorithm, while minor increasing the detection rate. The basic idea behind SEAdaboost is to exploit symmetries of objects, which are often present in certain object classes. We applied our algorithm to the field of face detection and classification of SMD-components. Our experimental results show that SEAdaboost reduces the learning phase drastically. The training time is reduced by almost 40%, while the detection rate could be slightly improved. Thus SEAdaboost qualifies to detect objects with symmetrical characteristics (e.g. components, faces). An application of SEAdaboost to objects with asymmetric characteristics would lead to a higher training time and high error rates of the symmetry features. However this effect could be used to make a statement concerning the object symmetry exploiting the weights of the symmetry features. Thus an application of SEAdaboost would also be of great convenience to estimate an object symmetry.

Our approach can easily be combined with other extensions, such as LPBoost, SoftBoost, MILBoost, FloatBoost or S-Adaboost [7–11].

References

1. Viola, P., Jones, M.J.: Robust real-time face detection. *International Journal of Computer Vision* **57** (2004) 137–154
2. Freund, Y., Schapire, R.E.: A short introduction to boosting. *Journal of Japanese Society for Artificial Intelligence* **14** (1999) 771–780
3. Freund, Y., Schapire, R.E.: Experiments with a new boosting algorithm. *Proceedings of the Thirteenth International Conference* (1996) 148–156
4. Loy, G., Eklundh, J.O.: Detecting symmetry and symmetric constellations of features. *ECCV* (2006) 508–521
5. Kearns, M.: Thoughts on hypothesis boosting. Unpublished manuscript (1988)
6. Kearns, M., Vazirani, U.V.: *An introduction to computational learning theory*. MIT Press (1994)
7. Warmuth, M.K., Glocer, K.A., Vishwanathan, S.: Entropy regularized lpboost. *Proc. Intl. Conf. Algorithmic Learning Theory in Lecture Notes in Artificial Intelligence*, Springer-Verlag (2008) 256–271
8. Warmuth, M.K., Glocer, K., Raetsch, G.: Boosting algorithms for maximizing the soft margin. *Advances in Neural Information Processing Systems* MIT Press **20** (2008) 1585–1592
9. Viola, P., Platt, J.C., Zhang, C.: Multiple instance boosting for object detection. *Advances in Neural Information Processing* **18** (2007) 1417–1426
10. Li, S.Z., Zhang, Z., Shum, H.Y., Zhang, H.: Floatboost learning for classification. *Advances in Neural Information Processing Systems*, Microsoft Research Asia (2002)
11. Jiang, J.L., Loe, K.F.: S-adaboost and pattern detection in complex environment. *Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 03)* (2003) 413–418
12. Crowther, P.S., Cox, R.J.: A method for optimal division of data sets for use in neural networks. Springer Berlin / Heidelberg, *Lecture Notes in Computer Science* **3684** (2005) 1–7