PCA Enhanced Training Data for Adaboost *

Arne Ehlers[1], Florian Baumann[1], Ralf Spindler[2], Birgit Glasmacher[2], and Bodo Rosenhahn[1]

[1] Institut für Informationsverarbeitung,
Leibniz Universität Hannover,
{ehlers,baumann,rosenhahn}@tnt.uni-hannover.de
[2] Institut für Mehrphasenprozesse,
Leibniz Universität Hannover

Abstract. In this paper we propose to enhance the training data of boosting-based object detection frameworks by the use of principal component analysis (PCA). The quality of boosted classifiers highly depends on the image databases exploited in training. We observed that negative training images projected into the objects PCA space are often far away from the object class. This broad boundary between the object classes in training can yield to a high classification error of the boosted classifier in the testing phase. We show that transforming the negative training database close to the positive object class can increase the detection performance. In experiments on face detection and the analysis of microscopic cell images, our method decreases the amount of false positives while maintaining a high detection rate. We implemented our approach in a Viola & Jones object detection framework using AdaBoost to combine Haar-like features. But as a preprocessing step our method can easily be integrated in all boosting-based frameworks without additional overhead.

1 Introduction

Several well known algorithms exist to perform object detection and recognition in images. For an overview on existing approaches and databases in the field of face detection, we recommend the web page [7] or the overview articles [17] and [18]. In the vast amount of available techniques, two complementary strategies are very common for object detection, namely boosting [6, 13] and PCA-based methods [12, 4]. Simply speaking and more detailed in Section 2.1, the strategy behind AdaBoost is to linearly combine several weak classifiers to gain a strong classifier. By using integral images, the classifiers are based on the difference of local (rectangle shaped) image patches. Object detection using AdaBoost is known to be slow in the training phase, but real-time capable in detection, even on resource limited systems. AdaBoost is further known to be sensitive in generating false-positive mistakes, which is sometimes compensated for by using

* This work has been partially funded by the DFG within the excellence cluster REBIRTH.

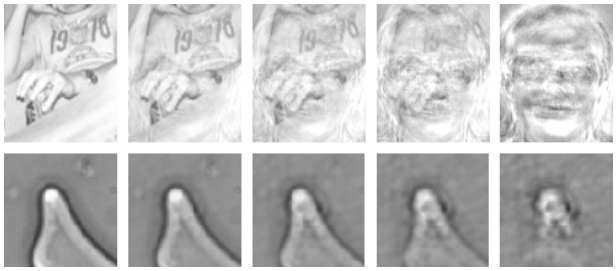


Fig. 1. Morphing a non-object towards the a PCA-trained object space. Top : Example morph for a face space. Bottom: Example morph for a cell space.

additional post-processing steps, such as Canny-pruning or histogram analysis. In contrast to a linear combination of local classifiers, the idea behind a PCA-space of objects is to learn a global subspace from training data which span the object variations as principal components in a high-dimensional vector space [12]. Additional to the PCA, several variants, e.g. independent component analysis (ICA) [2] or Kernel PCA (KPCA)[11] have also been proposed for face detection and recognition. The method for (K)PCA-based object detection is explained in more detail in Section 2.2. PCA based methods are more robust to noise since a subspace is learned from the training data, but much slower in the detection phase, because image patches need to be projected in the object space. For this reason we decided not to boost features evaluated in PCA-space as proposed in [19] and [1].

Since boosting- and PCA-based approaches rely on completely different principles, our main interest is to combine both methods without introducing additional limitations. So the key question is how to integrate a learned subspace of objects (e.g. faces, cells) in the boosting approach. Our observation is, that training data for e.g. non-objects are commonly very far away from the object space, once a PCA-space has been learned from the training data.

So the key contribution described in Section 3 is to modify the training-data of the non-objects in such a way, that they are closer to the PCA-space of objects and therefore to cause a much smaller margin at the start of training. This is in some way contrary to approaches in semi-supervised learning [8] in which the negative training class is raised at the boundary being opposite to the object class.

Figure 1 shows two non-object examples which are morphed towards their object-space, namely a non-face towards face-space in the top row and a defective cell towards cell-space in the bottom row. The middle images are non-objects which are much better suited to learn a boundary between the positive and negative classes in the boosting framework. Overall, it allows to train a much more selective classifier, especially if only a sparse amount of training data is available. Additionally, since the amount of training data remains unchanged

(images of non-objects are replaced with synthesized new images closer to the object space), we do not introduce any additional overhead in the training or testing phase in conventional AdaBoost. Another advantage is, that many existing modifications and improvements to the classical AdaBoost algorithm such as Entropy Regularized LPBoost, SoftBoost, MILBoost or SEAdaboost [16, 15, 14, 3] can still be used, since we only perform a kind of pre-processing with the training data (in this case for the non-objects). Therefore, it is sufficient to use the conventional AdaBoost algorithm to evaluate the impact of our approach.

2 Foundations

This section introduces the foundations for our work, namely the AdaBoost algorithm and the basic idea behind PCA-learning.

2.1 AdaBoost

AdaBoost is a popular machine learning algorithm proposed by Freund and Schapire [6] that can be applied to achieve good results in different areas in object detection. As a boosting algorithm it forms a strong classifier while trained on labeled classes of training data. The boosted classifier is a linear combination of single classifiers selected from a given set in each training round due to their minimal classification error. This error is calculated depending on weights assigned to the training images. By adapting these weights AdaBoost concentrates in later training rounds on the examples that are hard to classify. Suitable as weak classifiers are Haar-like features because of their fast and simple computation as proposed by Viola and Jones in [13]. Because of AdaBoost's autonomous learning the quality of the boosted classifier depends to a high degree on the characteristics of the training classes.

2.2 PCA-learning

A principal component analysis (PCA) transforms (possibly) correlated variables into uncorrelated variables called principal components by applying an orthogonal transformation. The transformation is designed in such a fashion that the first axis (principal component) represents the highest amount of data variation, whereas the other following (orthogonal) axes are sorted in a decreasing order, depending on the amount of variance. A PCA is simply computed as a singular value decomposition of a data matrix or by an eigendecomposition of a covariance matrix generated from a data set. In our implementation we derived the eigenvectors from a singular value decomposition of the covariance matrix. As that covariance matrix is symmetric and positive semidefinite the obtained eigenvectors are identical to those provided by a PCA and also the order of the corresponding eigenvalues is the same.

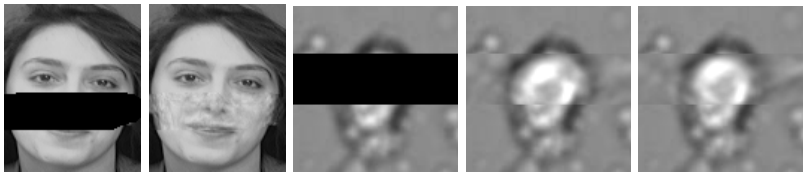


Fig. 2. Missing data estimation of face image (on the left) and a cell image (on the right). No blending was performed and it is shown, that the mouse and nose as well as the cell structure is well approximated. The reconstruction of the cell to the right is based on a Kernel-PCA. For this kind of data, it seems to approximate the cell boundaries slightly better.

Following the notations in [12], we assume n-dimensional data points $\Gamma_1 \dots \Gamma_m$ and compute the average of the data points as

$$\Psi = \frac{1}{m} \sum_{i=1}^{m} \Gamma_i \tag{1}$$

We further define $\Phi_i = \Gamma_i - \Psi$ as the difference to the average vector (and shift the data in this way towards the origin). The matrix $A = [\Phi_1, \dots, \Phi_m]$ contains the difference vectors of the data, so that the covariance matrix of the data points is given as

$$C = \frac{1}{m} \sum_{i=1}^{m} \Phi_i \Phi_i^T \tag{2}$$

$$= AA^T \tag{3}$$

A singular value decomposition $C = UDV^T$ of the covariance matrix C allows to compute the principal components of the data. Note, that the computation of $A^T A$ can be much more efficient, if less data points are available than the dimension of the data (see [12] for details). This happens, e.g. when face images are encoded as vector and only a few face images (e.g. a couple hundred) are available. Then U needs to be multiplied by A and rescaled to get the eigenvectors. Then the first s Eigenvectors can be used for approximation of the face space. Figure 2 shows two simple examples in which the missing parts of a corrupted face and cell image (not contained in the training data) has been reconstructed with the help of a database by performing a subspace projection. As can be seen, the position of the nose and mouth as well as the cell boundaries have been reconstructed fairly well.

There exist many extensions and modifications about PCA-methods for data clustering. In our experiments on Cell-data we will use a Kernel-PCA (KPCA) [11]. Here, we want to demonstrate that the method on PCA-enhancement of training data is not restricted to a specific method for subspace learning. The idea for KPCA is to employ a mapping $\phi(x)$ on the data to lift the input to a higher dimensional space, in which the subspace can be approximated more

easily. Since the higher dimensional space can become very large, they key idea behind Kernel-PCA is to avoid the explicit computation of ϕ and to work with a kernel $K_{i,j} = k(x_i, x_j) =< \phi(x_i), \phi(x_j) >= \phi(x_i)^T \phi(x_j)$. The covariance matrix C then becomes

$$C = \frac{1}{m} \sum_{i=1}^{m} \phi(x_i)\phi(x_i)^T \tag{4}$$

which is again splitted and normalized after a SVD, $C = UDV^T$. In our experiments we used two standard kernels, namely $K(x, y) = -\exp((x - y)^2/(\sigma)^2)$ with $\sigma = 1$ and $K(x, y) = (x^T y)^2$.

3 Training data enhancement

Obviously one possibility to combine PCA-based face detection and boosting is to use both algorithms for classification separately and to join (in a smart way) the outcome. The disadvantage of this method would be that the superior time performance of the boosting approach would be lost since the computation of the PCA-mapping takes significantly longer time for a test image. Therefore, the idea is to modify and enhance the training data. The impact of some characteristics of the training data on the test error is subject to the margin theory, that is described briefly in the following.

3.1 Margin analysis

Shortly after publication of the AdaBoost algorithm research has been started to examine its detection performance. As the consideration of only the training error is not sufficient to estimate the test error of AdaBoost, Schapire et al. [10] proposed the margin as a measure of the confidence in the algorithms classification. They defined the margin as the difference between the sum of the weights of the weak classifiers voting for the correct object class and the maximal sum of weights assigned to an incorrect class. As these weights are normalized to sum up to one, the margin is defined in the range [-1,1] and a positive value implies a correct decision. Hence a large positive margin represents a confident correct classification.

Evaluated on the complete training set a margin distribution can be derived. Schapire et al. observed that, due to its adaption of the training example weights, AdaBoost proceeds very aggressive in reducing the amount of training examples having a small margin. For this reason the AdaBoost learning algorithm can reduce the test error even after the training error has reached zero.

In the last decade much research has been done in estimating the test error subject to the margin and find a boundary based on the minimum margin and other training parameter. But recent research [9] indicates as well that considering the minimum margin is not sufficient and also the complete margin distribution has to be taken into account.

The margin theory should not be discussed in this work in detail. As we followed a more empirical approach in enhancing our detection framework, an experiment on the impact of our method on the margin distribution will be presented in Section 4.1.

3.2 Training data adaption

In our experience, negative training examples are not always well chosen to differ between objects and non-objects. This is mainly due to the fact, that the non-object space is significantly larger and more complex than the positive examples. Basically, the non-objects can be seen as the complementary space to the learned PCA object space. Therefore, its variability is hard to reflect in the training data. The idea is to bring the training data close to the PCA-space. This can simply be done by projecting the negative training examples onto the trained PCA-space and then shifting it back towards the non-object space with a scale $\lambda \in [0 \dots 1]$: Let $U_s = U(1:n, 1:s), s \leq n$ be the upper left matrix of U stemming from $C = UDV^T$ and let T be an example of the non-object class. The shift of T towards T_s being closer to the object space can simply be done by computing

$$Proj = U_s^T \cdot (T - \Psi) \tag{5}$$

$$Rec = U_s \cdot Proj + \Psi \tag{6}$$

$$T_s = Rec + \lambda(T - Rec) \tag{7}$$

Note, that U is an unitary matrix and thus U_s describes the inverse projection of U_s^T. In case of the more efficient computation mentioned in Section 2.2 this property is not given and instead a pseudoinverse has to be used. Obviously, $\lambda = 0$ yields the projection on the PCA-space, whereas $\lambda = 1$ leads to the training example itself. So λ steers the amount on how much the example is shifted towards the object space. In our experiments we used $\lambda = 0.3, 0.5$ and 0.7, respectively. Some example morphs of negative training data for different weighting factors are shown in Figure 1 and the second row of Figure 3.

4 Experiments

We decided for two kinds of experiments. The first experiments are on face detection. Here we use the well known 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 and facial details (glasses / no glasses). The images were taken against a dark homogeneous background with the people in an upright, frontal position.
The second set of experiments is performed on microscopic images of cells. The cells are recorded during cryo-conservation, and therefore ice fronts are forming around the cells. The goal is to detect (and track) the cells in the videos. Here we collected 250 images of cells and 350 images of non-cells, so that we gain a reasonable database. We divided our image bases into a training and validation

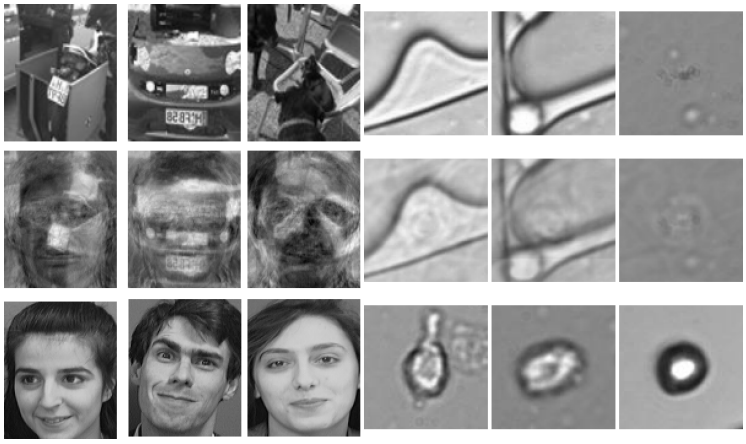


Fig. 3. Top row: Example images of non-faces and non-cells. Middle row: Morphed images towards the trained PCA space. These images are either used to find a more selective classifier. Bottom row: Example faces and cells of the used databases.

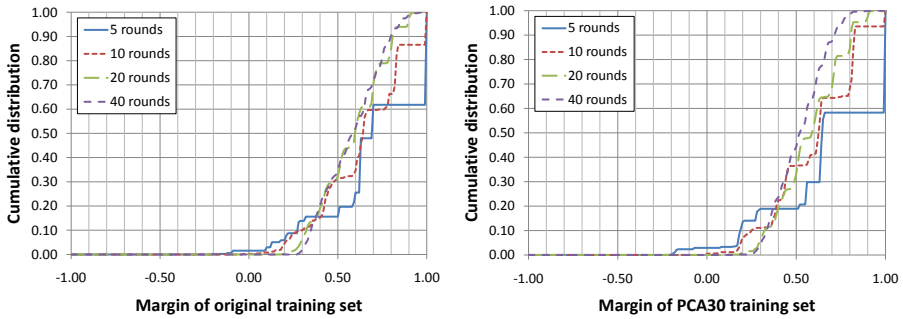
set using a 67/33 ratio. Crowther and Cox [5] illustrated that especially for small bases a split containing only a small part for validation is not recommendable. They suggested to select a ratio between 50/50 and 70/30.

Figure 3 shows in the top row example images of non-faces and non-cells. The middle row shows morphed images towards the trained PCA space. These images are then used for training to find a more selective classifier. The bottom row shows positive example images of faces and cells of the used databases.

4.1 Experiments with the AT&T database

Using the PCA from Section 3 we generated four different training sets containing the good/bad examples and morphed bad examples with $\lambda = 0.3, 0.5$ and 0.7. For all four data sets we performed an AdaBoost learning as described in Section 2.1. Here we used simple rectangular features for training.

Margin distribution Figure 4(a) presents the cumulative margin distributions of the original face training set after different training rounds. In Figure 4(b) the margins boosting the AT&T database using PCA-enhanced non-faces with $\lambda = 0.3$ is shown. The impact of the boosting process on the margin distribution is clearly noticeable in both figures. The amount of training examples having a small margin is in both cases strongly reduced during training. Roughly after 10 rounds the training error reaches zero as all examples images have a positive margin and hence are correctly classified. Then the AdaBoost algorithm further concentrates on the training examples that are hard to classify and continues to reduce the number of narrow decisions.



(a) (b)

Fig. 4. (a): Cumulative margin distributions of the original face training set after 5, 10, 20 and 40 rounds. (b): Cumulative margin distributions of the morphed training set. The negative object class has been morphed using $\lambda = 0.3$.

But it is also observable that it is more difficult to classify the morphed training set. After 5 training rounds the boosted classifier for the morphed set makes almost twice as much wrong decisions compared to the classifier boosted on the original training set. Also about 15% and 30% of the training examples on the morphed set have a margin smaller than 0.2 and 0.56, respectively. In comparison, for the original set the smallest 15% have a margin below 0.32 and the margins of the smallest 30% do not exceed 0.62.

After 40 training rounds the boosted classifier for the morphed training set has caught up in the lower region of the margin distribution. The minimum margin amounts roughly to 0.26 in both cases and the progress of cumulative distributions is similar showing only a slightly steeper slope for the morphed training set.

As discussed in Section 3.1 the margin distribution in training has been found to be an indicator for the quality of a classifier in terms of its test error. Hence the result of the PCA enhanced training to achieve a similar margin distribution starting from an adverse one can be interpreted as a higher gain during training. Therefore we expect classifiers boosted using PCA enhanced training data to achieve superior performance in the test phase.

Face detection In the following the results of experiments on the test set are presented. Figure 5(a) shows the ROC-curves of multiple classifiers varying the classification threshold in detecting faces. The curve in red is the classifier based on the original data set, whereas the other curves show the performance of the classifiers using PCA-images for training. Overall, the curves show that the classifiers which have been trained with the PCA-images are more selective in detecting faces so that good detection rates are achieved while maintaining a lower false alarm rate.

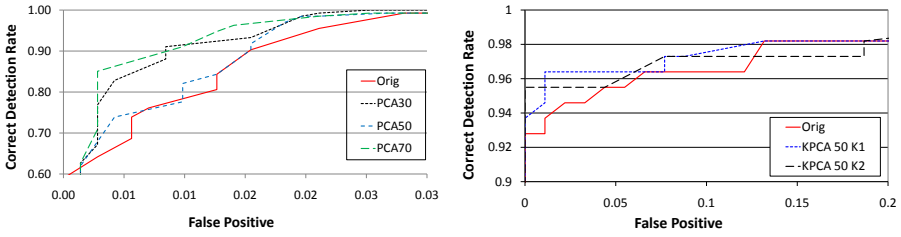


Fig. 5. (a): ROC curve for the face database using different thresholds of boosting with the original data (red) and using PCA-enhanced non-faces with different λ-values (0.3, 0.5 and 0.7). The PCA-enhanced data reveals a much more selective performance. (b): ROC-curve for the cell database.

4.2 Experiments with the Cell database

For the cell database we decided on using a Kernel-PCA-method for modifying the training data. The main reason is to demonstrate that variants of PCA-learning can be used in a similar fashion. Especially for image data with larger image gradients (due to edges), KPCA-methods can be better suited, since the overall smoothing effect using PCA can be reduced. Since the KPCA-enhanced training is dependent on the selection of the kernel function we further decided to compare two different (standard) kernels, namely $K_1(x,y) = -\exp((x-y)^2/(\sigma)^2)$ and $K_2(x,y) = (x^T y)^2$. The KPCA-enhanced training data leads for both kernels to an increased performance of the detection rate, which is shown in the ROC-curve in Figure 5(b). E.g. for a detection rate of 96.3%, the PCA-enhanced training data with K_1 yields a classifier which produces a false-positive detection rate of 1%, whereas the original data produces a false-positive detection rate of 6%. The PCA-enhanced training data with K_2 yields similar performance, in producing a detection rate of 95.5% with no false positives.

5 Conclusion

We introduced an approach to enhance the training data of boosting-based object detection frameworks to achieve a higher detection performance. Using principal component analysis we shift the negative training examples in PCA space near to the positive training class. The trained classifier achieves a lower classification error being more selective in detection. Our experiments on face detection and microscopic cell images showed that our method decreases the false positive rate of the boosted classifier. The variable strength of our transformation allows for a trade-off between true positive and false alarm rates. But in all experiments our approach managed to significantly lower the amount of false alarms without reducing the detection rate. As a preprocessing of the training data our method can be integrated in nearly all boosted detection frameworks without any computational costs in the learning and detection process.

References

1. Ali, S., Shah, M.: An integrated approach for generic object detection using kernel pca and boosting. In: ICME. pp. 1030–1033 (2005)
2. Bartlett, M., Movellan, J., Sejnowski, T.: Face recognition by independent component analysis. Neural Networks, IEEE Transactions on 13(6), 1450 – 1464 (2002)
3. Baumann, F., Ernst, K., Ehlers, A., Rosenhahn, B.: Symmetry enhanced adaboost. In: Advances in Visual Computing - 6th International Symposium. vol. 6453, pp. 286–295 (nov 2010)
4. Blanz, V., Vetter, T.: A morphable model for the synthesis of 3d faces. In: SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques. pp. 187–194. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA (1999)
5. 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, 1–7 (2005)
6. Freund, Y., Schapire, R.E.: A short introduction to boosting. Journal of Japanese Society for Artificial Intelligence 14(5), 771–780 (1999)
7. Homepage, F.D.: . `http://www.facedetection.com/` (2010)
8. Leistner, C., Grabner, H., Bischof, H.: Semi-supervised boosting using visual similarity learning. In: CVPR (2008)
9. Li, H., Shen, C.: Boosting the minimum margin: Lpboost vs. adaboost. In: Proceedings of the International Conference on Digital Image Computing: Techniques and Applications, DICTA. pp. 533–539 (2008)
10. Schapire, R.E., Freund, Y., Barlett, P., Lee, W.S.: Boosting the margin: A new explanation for the effectiveness of voting methods. In: Proceedings of the Fourteenth International Conference on Machine Learning (ICML). pp. 322–330 (1997)
11. Schölkopf, B., Mika, S., Smola, A., Rätsch, G., Müller, K.R.: Kernel pca pattern reconstruction via approximate pre-images. Proceedings of the 8th International Conference on Artificial Neural Networks, Perspectives in Neural Computing, Springer Verlag pp. 147–152 (1998)
12. Turk, M., Pentland, A.: Face recognition using eigenfaces. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 586–591. IEEE Computer Society (1991)
13. Viola, P., Jones, M.J.: Robust real-time face detection. International Journal of Computer Vision 57(2), 137–154 (2004)
14. Viola, P., Platt, J.C., Zhang, C.: Multiple instance boosting for object detection. Advances in Neural Information Processing 18, 1417–1426 (2007)
15. Warmuth, M.K., Glocer, K., Raetsch, G.: Boosting algorithms for maximizing the soft margin. Advances in Neural Information Processing Systems MIT Press 20, 1585–1592 (2008)
16. 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 (5254), 256–271 (2008)
17. Yang, M.H., Kriegman, D.J., Ahuja, N.: Detecting faces in images: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence 24, 34–58 (2002)
18. Zhang, C., Zhang, Z.: A survey of recent advances in face detection. Microsoft Research Technical Report, MSR-TR-2010-66 (2010)
19. Zhang, D., Li, S.Z., Gatica-Perez, D.: Real-time face detection using boosting in hierarchical feature spaces. In: ICPR (2). pp. 411–414 (2004)