# Fast Face Detector Training Using Tailored Views

Kristina Scherbaum
Cluster of Excellence MMCI at Saarland University
Saarbrücken, Germany
scherbaum@mmci.uni-saarland.de

James Petterson
Commonwealth Bank of Australia
Sydney, Australia
james.petterson@cba.com.au

Rogerio S. Feris
IBM Watson Research Center
Hawthorne, New York
rsferis@us.ibm.com

Volker Blanz
Universität Siegen
Siegen, Germany
blanz@informatik.uni-siegen.de

Hans-Peter Seidel
MPI for Informatics
Saarbrücken, Germany
hpseidel@mpi-inf.mpg.de

## Abstract

*Face detection is an important task in computer vision and often serves as the first step for a variety of applications. State-of-the-art approaches use efficient learning algorithms and train on large amounts of manually labeled imagery. Acquiring appropriate training images, however, is very time-consuming and does not guarantee that the collected training data is representative in terms of data variability. Moreover, available data sets are often acquired under controlled settings, restricting, for example, scene illumination or 3D head pose to a narrow range. This paper takes a look into the automated generation of adaptive training samples from a 3D morphable face model. Using statistical insights, the tailored training data guarantees full data variability and is enriched by arbitrary facial attributes such as age or body weight. Moreover, it can automatically adapt to environmental constraints, such as illumination or viewing angle of recorded video footage from surveillance cameras. We use the tailored imagery to train a new many-core implementation of Viola Jones' AdaBoost object detection framework. The new implementation is not only faster but also enables the use of multiple feature channels such as color features at training time. In our experiments we trained seven view-dependent face detectors and evaluate these on the Face Detection Data Set and Benchmark (FDDB). Our experiments show that the use of tailored training imagery outperforms state-of-the-art approaches on this challenging dataset.*

## 1. Introduction

Face detection is an important task for a wide range of applications in computer vision. Thus, a variety of face detection algorithms have been presented in recent years, many of them involving supervised or unsupervised machine learning methods. Their goal is to learn a face classification function by training on a set of annotated sample pictures which is then applied to new, unseen pictures in order to detect faces. This, however, requires large amounts of manually labeled face pictures during the training phase, which is not only very time-consuming but also difficult to obtain. Moreover, available data sets are often acquired under controlled settings, restricting, for example, scene illumination or 3D head pose to a narrow range. Or, in contrast, they may scatter widely but without a guarantee that they sample all relevant dimensions sufficiently densely. Also, manually labeled data often involves mistakes. When training multiview classifiers for example, the manually labeled viewing angles might be inaccurate and thus lead to biased results. This paper addresses these shortcomings and takes a look into the automated generation of tailored training samples from a 3D morphable face model.

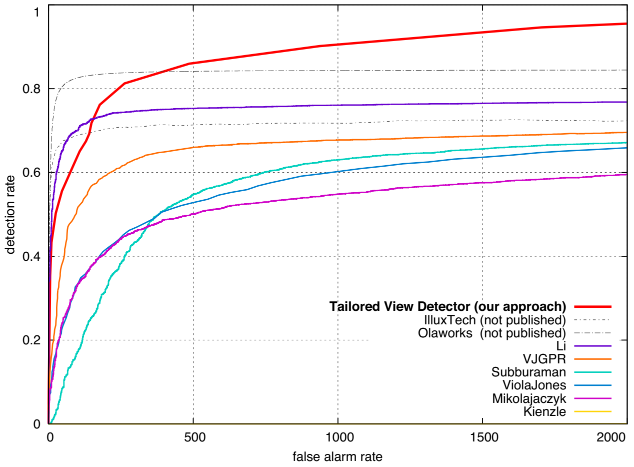

Figure 2. Comparison of the detections at equal error rate. Training on the synthetically tailored views of 75 distinct subjects, is sufficient to outperform previously presented face detectors on the challenging Face Detection Data Set and Benchmark (FDDB) [1].

To automatically compute training samples, we start from

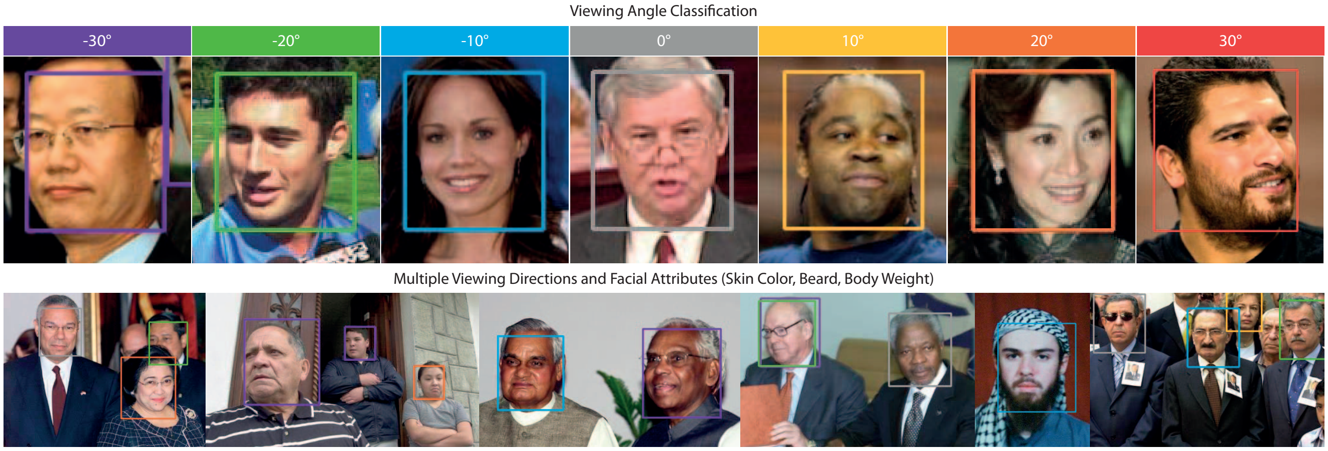
Viewing Angle Classification
-30°
-20°
-10°
0°
10°
20°
30°
Multiple Viewing Directions and Facial Attributes (Skin Color, Beard, Body Weight)

Figure 1. **Top row:** Comparison of detection results at distinct viewing angles. The respective bounding box color denotes the automatically classified viewing angle $\phi \in [-30°; +30°]$. **Bottom row:** Exemplary detection results on challenging 'FDDB' imagery [1]. The algorithm robustly detects faces despite apparent variation of facial attributes such as increased body weight, beards, bright or dark skin color.

a few randomized facial samples which we gain from a 3D morphable face model [2, 3]. Each 3D random face is then modulated by automatically changing a set of facial attributes such as gender, weight or age. For each modulated face, we either freely choose the settings for rendering, such as the parameters for 3D pose, position, size and illumination, or extract these parameters from given particular video sequences recorded by surveillance cameras. The latter method in particular can be helpful when training classifiers for cameras that are positioned at unusual viewing angles or in a very dark or artificially lit environment. Using this procedure, we generate seven distinct training sets, each set corresponding to a specific range of face orientations. On each training set we then train a view-specific classifier using an adaptive boosting (AdaBoost) approach which we derived from the object detection method Viola and Jones initially presented in 2001 [4]. In contrast to their approach, we implemented the algorithm to run in a many-core setup that also extends the feature plane by an arbitrary number of additional layers, which may be used either for color representation or further feature sets.

Finally, we merge our seven view-dependent classifiers to a single classifier and compare its performance to state of the art methods on the FDDB benchmark dataset [1].

## 2. Related Work

Face detection is often the first step in complex image processing applications, like face recognition, visual surveillance, or human-machine interaction. This explains the high level of interest of the research community in this topic. Many solutions to detecting faces in images have been presented in the last decade. Comprehensive surveys may be found in Yang et. al. [5, 6]. Out of the presented approaches we focus on the widely used appearance-based machine learning approach presented by Viola and Jones [4, 7].

**Object Detection Framework** The algorithm is based on the assumption that many coarse Haar feature classifiers, connected in series, are superior to a single classifier built with high-level image descriptors. The coarse classifiers are organized hierarchically, where the number of computed Haar features tends to increase with each stage. While in the first stage only a few Haar features are computed, each following stage has stricter requirements and usually requires more features. At training time, the AdaBoost algorithm determines a constant threshold for each stage, to which the candidates can be compared at detection time. Image candidates passing all stages are considered to contain a face. Negative images exit at earlier stages. The overall number of Haar features varies and depends on the training parameters.

**Detection Performance** Using the integral image structure [4] Haar features can be computed quickly. On mobile devices or when applied to video sequences, however, performance may decay drastically. Consequently, a variety of algorithmic performance improvements has been presented in literature. Noticeable speedups have been achieved when running Viola and Jones' detector on several GPUs [8, 9] or by combining GPU and CPU [10, 11]. Chuang et. al. [12] introduced an enhanced training algorithm considering sampling optima for video material. Others explored the possibilities of parallelism using many-core architectures, improved memory behavior or investigated how to optimally compute the integral image [13, 14, 15, 16, 17, 18]. All above methods reported considerable computational speedup, but only at runtime. We, in contrast, parallelize the AdaBoost at training time. Though we additionally introduce new layers (more features) for the use of color channels, the many-core architecture allows for fast large-scale training.

**Acquired Training Data** Data-driven face classifiers require appropriate training data. There are many supervised or unsupervised solutions to image annotation, such as collaborative annotation projects [19, 20, 21, 22], or algorithmic approaches for Google's image search [23], web content [24, 25, 26, 27] or labeled social media content [28, 29]. As a result, a large variety of manually labeled datasets has been published. Comprehensive surveys of facial datasets can be found in [30, 31, 32]. Other challenging datasets can be found in [33, 34, 1]. Out of these, the face dataset of the MPI for Biological Cybernetics is the most related to our work. However, their dataset does not cover the full spectrum of statistical data variability and does not include facial attributes such as age or body weight.

**Synthetic Training Data** In general, manually collected facial datasets show insufficient data variability. To increase variability, people usually collect large amounts of images. Automatically synthesized training data, by contrast, involves high-level face models to increase variability: In 2004 Yue-Min et. al. [35] relit faces in training images using harmonic images they derived from a 3D face model. Dianle et. al. [36] use an active shape model to synthesize training data. The data is then used to find facial landmarks in different views. A variety of face recognition systems [37, 38, 39, 40, 41] use 3D models to synthesize intermediate views or viewpoint invariant reference frames for the purpose of face recognition. A more comprehensive survey on similar methods can be found in [42]. Pishchulin et. al. [43] use a morphable body model to generate training data for pedestrian detection. They could show that even a low number of synthetic training samples — with increased data variability — can outperform detectors trained on large manually collected data sets. Similar to our work, Weyrauch et. al. [44] use a morphable model for pose invariant face recognition. From three input images of each subject in the training database, a 3D model [2] is extracted. The model is then rendered under varying pose and illumination conditions to build a set of synthetic images, used for training a component-based face recognition system. In contrast to their method, we focus on face detection, and show that state-of-the-art results can be obtained by leveraging tailored training data from a 3D morphable model. We do not require initial facial input images, but randomly generate artificial faces while controlling the data variability. Moreover, we introduce facial attributes such as body weight or skin tone and make use of an advanced face model [3] to render the subjects' ages. We also show that our approach is suitable to adjust rendering parameters to particular illumination and pose constraints of given surveillance cameras (Figure 8).

## 3. Synthetic Training Images

When manually labeling and selecting training images of faces, there is no guarantee that the collected data includes all possible shape and texture variations of faces. Consequently, people usually collect large amounts of training data to at least sample as much variation as possible. Examples can be found in [30, 31, 32]. When generating synthetic training data, in contrast, this workaround is not necessary. By deploying a statistically driven face model for data generation, one can be sure to incorporate the full data variance (with respect to the database of the face model). We use this technique in the following section to first generate randomized 3D faces using a 3D morphable face model [2, 3]. In a second step, we modulate these three-dimensional random faces by applying facial attributes, which we then render for defined viewing angles and illumination parameters. Finally, we compose the face renderings with natural-looking background images.

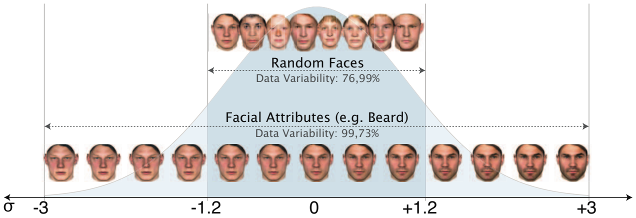

Figure 3. First we generate random faces by modulating existing faces from the database permitting 76.99% of all possible variations ($\sigma \in [-1,2;+1,2]$) which we secondly modulate applying attributes at full data variability ($\sigma \in [-3;+3]$).

**Modeling** To generate artificial training data we employ a 3D morphable face model [2, 3]. The model's database contains $m = 512$ faces ranging from the age of 3 months to $\approx 40$ years with an approximately equal number of female and male individuals (200 adults, 236 children aged between 7 and 16 years and 76 very young children aged between 3 and 12 months). The 3D shape of each face $F_i$ is stored in terms of the $x, y, z$ coordinates of all surface vertices $k \in \{1, ..., n\}, n = 75972$ in a vector $S_i$. Analogously, we store the color values (red, green and blue) of the surface vertices in a texture vector $T_i$:

$$\mathbf{S_i} = (x_1, y_1, z_1, \dots, x_n, y_n, z_n)^T \tag{1}$$

$$\mathbf{T_i} = (R_1, G_1, B_1, \dots, R_n, G_n, B_n)^T \tag{2}$$

Performing a Principal Component Analysis on all shape and texture vectors we estimate the probability distributions of faces around their averages $\overline{\mathbf{s}}$ and $\overline{\mathbf{t}}$. The result is a small set of $(m-1) = 511$ orthogonal principal axes (eigenvectors) $s_j, t_j$ which vary around the averages $\overline{\mathbf{s}}$ and $\overline{\mathbf{t}}$ :

$$\mathbf{S} = \overline{\mathbf{s}} + \sum_{j=1}^{m-1} \alpha_j \cdot \mathbf{s_j}, \qquad \mathbf{T} = \overline{\mathbf{t}} + \sum_{j=1}^{m-1} \beta_j \cdot \mathbf{t_j} \tag{3}$$

The eigenvectors of the PCA represent the variation across all faces in the database. Most eigenvectors do not explicitly represent semantically meaningful facial features. By definition of the principal component analysis, however, the eigenvectors are sorted according to their corresponding

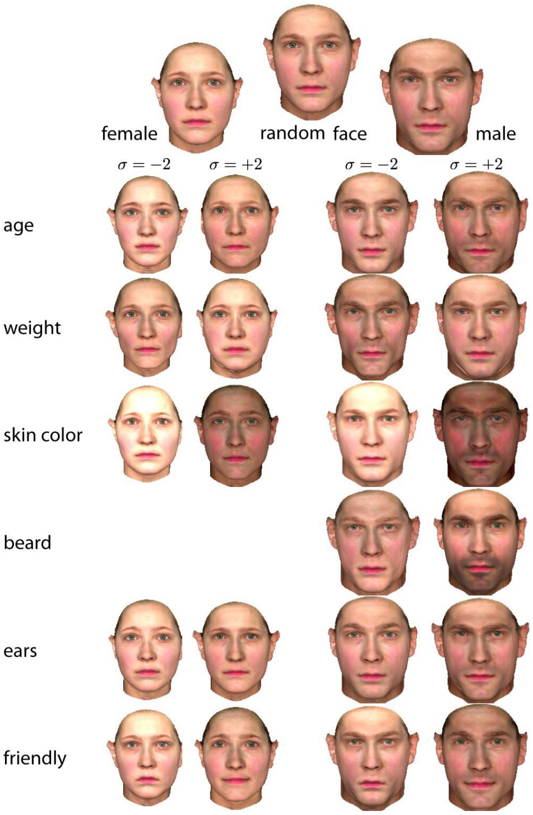

Figure 4. Generation of synthetic face data: from an initially generated random face (top middle), we first derive a female and male version (second row) and then apply attribute-vectors to those to modulate the age, weight, skin color, beard shadow, ear size and the facial expression. The above figure shows the modulation results for $\sigma = \pm 2$.

eigenvalues in decreasing order. Thus the highest variation between all faces in the database is represented by the frontmost eigenvectors. In the following analysis, we therefore only consider the first eigenvectors ($j <= 50$) for shape $s_j$ and texture $t_j$ to control the variance of the computed training data. The rear eigenvectors ($j > 50$) contain highly individual details which are not relevant for our purpose and thus may be ignored. In a first step, we generate a set of randomized 3D faces by manipulating available 3D faces from the 3D morphable model face database. We then apply shape and texture variation to the selected samples by deploying a varying factor $\sigma$ to the first 50 eigenvectors $s$ and $t$ ($\sigma \in [-1.2; 1.2]$). We initially keep the scaling factor $\sigma$ relatively low with respect to the available variability to prevent producing unwanted artifacts which would require a manual quality check (cf. Figure 3). Modifying the first 50 eigenvectors only serves as regularization and prevents enhancing highly individual details. The results are randomly generated faces (or 'random faces') which serve as the basis for all following steps. We additionally require a large angle (Mahalanobis Distance) between computed sample face vectors to ensure low similarity in-between the random faces.

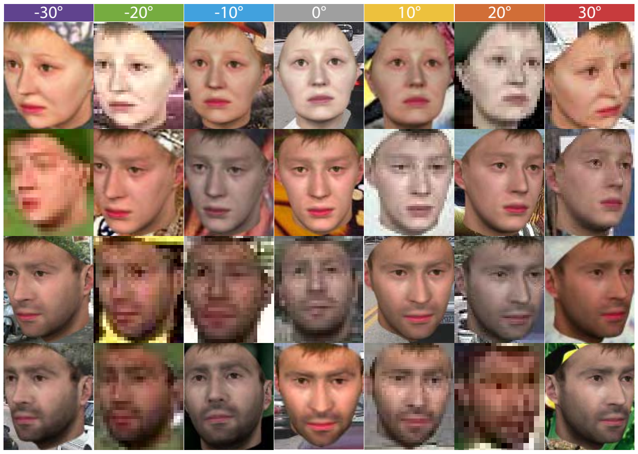

Figure 5. Each row shows the various renderings of a single manipulated random face. In a final compositing step we randomly scale the size of the rendered faces to simulate typical surveillance recordings and blend the rendered views with background scenes.

**Modulation** In a first step we generate a female and male version for each random face. To these we subsequently apply a set of facial attributes such as increased or decreased body weight or, for example, light or dark skin color. We learn these attributes in a step prior to the stimulus generation procedure, so they can be applied automatically to each face. The learning procedure ([2]) involves manual labeling of each database face according to the perceived strength of the attribute in each face. Each face $F_i$ is attributed a scalar value $\mu_i$. Then, we fit a linear function $f$ to these data that reproduces the labels $\mu_i$. Following the gradient of $f$ in PCA space will then produce a perceived change of the attribute strength in a given face, while all other individual characteristics of the face remain unchanged. Please note, that this has to be done only once for each attribute, regardless of how many data sets for training are required.

We performed this method for the following facial attributes: age (young / old); body weight (obese / skinny); beard shadow (dark shadow / no shadow); skin color (light / dark); ears (big / small) and also for one facial expression (friendly / unfriendly). We apply these attributes to all random faces using a scaling factor $\sigma \in [-3.0; 3.0]$, corresponding to $99.73\%$ of all possible variations (cf. Figure 3). The strength of all attributes can be precisely determined in terms of variance. For example, for female faces we avoid bearded female faces by setting $\sigma = 0$. Results of the modulation are shown in Figure 4, where we rendered all facial attributes for the values $\sigma = \pm 2$.

**Rendering** In the next step, we transform all previously generated 3D faces into 2D representations. To obtain an image $I_i(x, y)$ from a given 3D face $F_i$, we apply standard computer graphics procedures:

$$R_\rho(F_i) = I_i(x, y) \qquad (4)$$

Figure 6. The background scenes used for image compositing are randomly sampled from the PASCAL2 Visual Object Classes Challenge 2012. The scenes typically show outdoor scenes, urban scenes or any other human environment.

By backprojection from PCA space we first determine object coordinate vectors for shape and texture of each face. Applying rigid transformation and scaling we then map each coordinate to world coordinates. Next, a perspective projection maps each world coordinate to a point in image space. In a final step we compute the surface normals. The resulting images depend on a set of rendering parameters $\rho$, where the respective number of parameters is given in brackets: 3D rotation (3), 3D translation (3), focal length of the camera (1), angle of directed light (2), intensity of directed light (3) and ambient light (3), color contrast (1), gain (3) and offset (3) in each color channel. All faces are rendered at seven predefined viewing angles. We thus generate a total of seven distinct training sets that all differ in the chosen face viewing angle $\phi$, where $\phi \in \{-30°; -20°; -10°; 0°; +10°; +20°; +30°\}$. In addition, to achieve naturally varying results, we apply slight random variations while rendering. Within each set, we modulate for example the left-right viewing direction of the face ($[-3°; 3°]$), the up-down angle ("‘nodding‘", $\theta \in [-15°; 15°]$) and the in-plane rotation ($\gamma \in [-5; 5]$). All above values are motivated by heuristics. To simulate various environments, we additionally modulate color and intensity of the ambient light and apply random variations to color contrast, color gain and offset. Finally, we render a segmentation mask for each face. For each of the resulting training sets we later train a single face classifier which we combine to a multi-view face detector in the end.

**Compositing** In a final step we blend the rendered faces with background scenes that do not contain any faces (‘negatives’). We use background images that are randomly sampled from the ‘PASCAL2 Visual Object Classes Challenge 2012 (VOC2012)’[1] image collection. To exclude apparent faces from the selected images, we initially remove all images that are labeled to contain humans or human faces. The remaining images typically show outdoor scenes, urban scenes or any other human environment (cf. Figure 6). We blend each rendered face with a random background and apply a smooth contour blend (Gaussian blur) using the rendered contour mask at the facial contour. In addition, we randomly vary the size and position of the face in the background image and then store its rectangular coordinates as ground truth. Each composed image contains exactly one individual face in the end. Images containing at least one face are also referred to as ‘positives’. Figure 5 illustrates examples of the composed positive training images.

## 4. Enhanced AdaBoost Training

Using the above-described synthetic training data, we train a face classification system. Given an image patch, the system should determine whether the patch contains faces or not and should locate potential faces in the image. Moreover, the system should work across varying image sizes, depths and resolutions.

The above requirements are satisfied by the OpenCV[2] implementation of Viola and Jones' AdaBoost framework [4, 7]. While the algorithm performs very well during testing phase (real-time), the training phase can quickly turn slow and tedious, especially when training on numerous images (>3000). One reason is that per stage, a very high number of features has to be computed, evaluated, selected or finally discarded. This makes the training procedure very slow and can become very impractical when it comes to determine proper training parameters. One can easily spend days to weeks tweaking parameters. Another drawback of Viola and Jones' method is that only greyscale images are processed as opposed to recent approaches that have shown that color information may improve face detection results ([45]). To overcome these drawbacks we present two major adaptions to the OpenCV system: Firstly, we introduce new feature layers that can be used either for color channels or arbitrary descriptors and secondly, we parallelize the complete training procedure to run on many-core architectures. Despite using more feature layers (or colors) we could thus tremendously speed up the training procedure.

**Parallelization** Sharing the load among multiple CPUs allows for very fast training procedures and for the training on thousands of images, rather than a few hundred, in a very short time. In our tests this decreased the training time by a factor of 5.3 using eight cores versus a single core. Parallelizing the AdaBoost training procedure [4, 7] is not straightforward, however, since the algorithm is sequential by nature. Among the non-sequential parts, the most expensive step is the computation of features on every image patch. At each step a massive number of features has to be

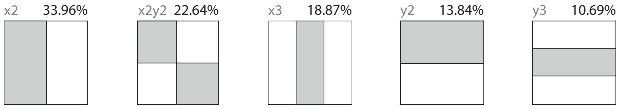

Figure 7. Histogram of the selected Haar features at training time. Haar feature $x2$ is selected in $\frac{1}{3}$ of all cases.

[1] http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2012/

[2] http://opencv.willowgarage.com/wiki/

computed and among these, the most descriptive ones have to be selected. We speed up these processes by introducing the following improvements: a) We precompute a subset of the features in each CPU b) We parallelize the feature selection at each AdaBoost iteration (involving a lightweight synchronization with the master CPU). c) We parallelize the negative patch selection (for which the classifier fails) as this procedure massively slows the algorithm down at the later stages (on the order of hours). All modifications have been implemented using the OpenMP[3] API.

**New Features Layers** OpenCV's Viola-Jones implementation was initially designed to train on intensity images only. We, however, expect the color channels to contain relevant information for the detection of faces. Evaluation showed that by introducing additional layers the algorithm now finds the most descriptive features in the $red$ $(57.86\%)$ channel, followed by the $blue$ $(25.16\%)$ and $green$ $(16.98\%)$ channels. Out of these, Haar feature $x2$ is the most frequently selected feature (cf. Figure 7). In future work, we are planning to exploit other color spaces suitable for skin-color modeling, as for example proposed by [46].

**View-Classification** We moreover alter the training procedure: In contrast to common practice, where people use manually labeled and probably also biased data, we know the exact viewing angle of each face in our training sets, allowing us to train view-dependent detectors rather than a generalized single classifier. However, during the detection phase one might want to detect all faces in an image, regardless of their viewing angles. For this purpose we later recombine the view-dependent cascades to a single multi-view classifier that is capable of finding faces at any viewing direction in the image (cf. Figure 1).

## 5. Evaluation and Results

Overall we trained seven distinct classifiers, one for each of our seven synthetic training sets. For each classifier we trained on 5000 positive samples (thus 35000 in total) and 5000 negative samples. After training, we evaluated the resulting classifiers according to the Face Detection Data Set and Benchmark (FDDB) as recommended by Jain and Learned-Miller in 2010 [1]. The FDDB benchmark dataset comprises 2845 images with a total of 5171 annotated faces. Within the dataset a considerable number of challenging pictures can be found. Examples are challenges such as low resolution faces, out-of-focus faces, occlusions or difficult and unusual face poses. Please also note that the FDDB-benchmark framework requires evaluation in terms of a tenfold cross validation per definition. Before evaluating our classifiers according to the FDDB standard, we first align all seven classifiers. We therefore apply a stage threshold bias shift to all seven detectors such that they all start with zero false alarms at the same point in the receiver operator characteristic (ROC). After threshold adjustment, we build a cascade of all seven classifiers, which — for each image patch — searches for faces at the respective viewing angles. We currently combine the view-based detectors naively, more sophisticated methods, however, (e. g., vector boosting [47]) could lead to improved accuracy. The resulting cross-validated ROC curve is shown in Figure 2.

However, the results indicate that training on statistically well distributed synthetic training data seems to be a promising concept: Though the method of Li et al. still appears partially superior to ours, our classifier could outperform most previously published methods, such as Kienzle, Mikolajaczyk, Subburaman, VJGPR and the standard Viola-Jones approach reported on the FDDB benchmark homepage [4].

## 6. Applications

Most presented face detection algorithms so far do involve a time-comsuming initial step: the collection and labeling of training images. This step is inevitable to achieve optimal detection results for a specific camera type or environmental setting. Detection accuracy is directly related to the quality of training data. Synthetic training data, in contrast, might overcome these drawbacks, and additionally offers a wide range of new applications:

**Self-Learning Surveillance Cameras** Surveillance cameras, as for example in large cities, are usually installed over the course of years and thus vary widely in terms of their intrinsic parameters (such as focal length or resolution). They are located all over the city, at varying positions, viewing perspectives, illuminations and environments. Regardless of this fact, surveillance systems often use the same detector for all cameras. Camera-specific properties are ignored and detection might fail, for example at unusual viewing perspectives like a birds-eye view. Camera-specific detectors could be a solution to this. However, generalized detectors are still standard, since it would be too time-consuming to train hundreds of camera-specific face detectors.

Using our system, taking a few snapshots per surveillance camera is sufficient to train camera-specific detectors. With the help of little manual interaction (about seven clicks per sample face) we can extract the parameters $\rho$ from the camera snapshots. All parameters are estimated automatically in an analysis-by-synthesis loop which finds the parameters $\alpha, \beta, \rho$ that make the synthetic image $\mathbf{I}_{model}$ as similar as possible to the original image $\mathbf{I}_{input}$ in terms of pixelwise difference

$$E_I = \sum_x \sum_y \sum_{c \in \{r,g,b\}} (I_{c,input}(x,y) - I_{c,model}(x,y))^2$$

[3] http://openmp.org/wp/

[4] http://vis-www.cs.umass.edu/fddb/results.html

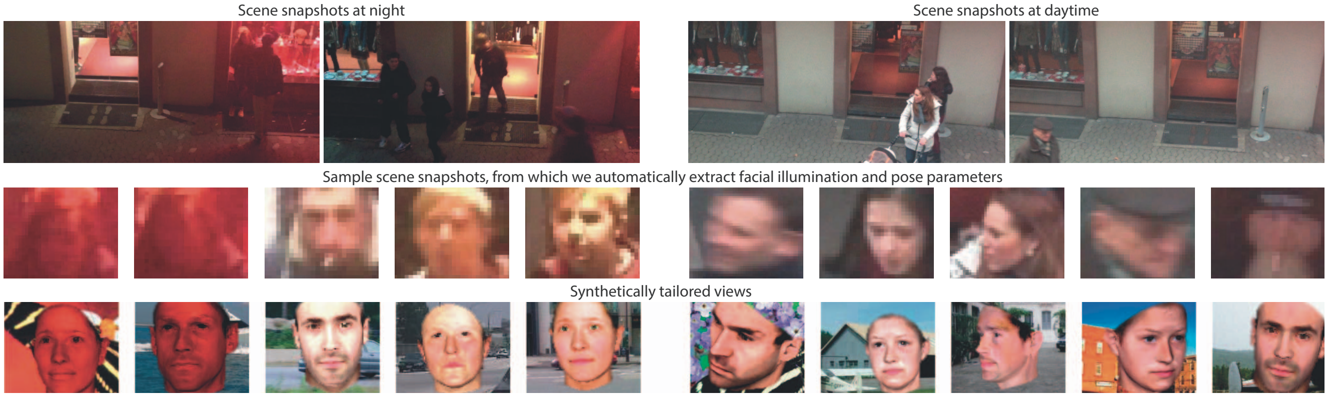
Scene snapshots at night
Scene snapshots at daytime
Sample scene snapshots, from which we automatically extract facial illumination and pose parameters
Synthetically tailored views

Figure 8. **Top row:** Shown are four still images recorded by a single surveillance camera. While the scenes on the left were recorded at night, the scenes on the right show recordings at daytime. **Middle row:** From the still images we manually extract a few cropped face images which we then use to automatically estimate facial illumination and pose parameters by fitting the 3D morphable face model them. **Bottom row:** Using the estimated light and pose parameters we render tailored training imagery showing arbitrarily modulated random faces.

Inferring from the extracted parameters $\rho$, it is straightforward to automatically generate thousands of labeled training images. The generated files guarantee coverage of the full spectrum of statistical data variability and may be equipped with tailored facial attributes. Using the parallelized training procedure on top, one may quickly compute effective detectors for each single camera, at the cost of a few clicks.

**Selective Training Data** Regarding surveillance systems, it might also be the case that one wants to train a detector for specific target groups. Examples could be a surveillance camera at a primary school. For these cases, specific training data might be helpful but difficult to obtain. To overcome these difficulties we can use our system to generate training samples following predefined constraints. We can do this by either taking a few samples and generating many variants (bootstrapping) of them or by labeling our data with respect to the wanted attribute (young vs. old) and manipulate existing faces from our datasets.

**Full Control of Arbitrary Attributes** Depending on the use of the detector, it might be useful to train detectors for specific accessories or attributes, such as glasses, beards or hats. While common methods would require observing enough sample data in the real world, our system allows us to design arbitrary attributes in 3D and to place them on any rendered face. This way, one may produce a large amount of training images for any specific purpose.

## 7. Summary, Discussion and Future Aspects

We presented a face detection system that is trained only on synthetic training data. The results indicate that using synthetic training data is meaningful and offers a variety of useful applications. The time consuming process of manually labeling faces can be replaced by a fully automated procedure. However, the generation of synthetic training data highly depends on the availability of a suitable 3D face model (such as morphable 3D face model or active shape model). Constructing a morphable model from scratch is very time consuming and also requires available 3D data of faces. But once a model is all set, training data comes at almost no computational cost and scales easily to larger quantities. In addition, it is easy to adapt the artificial training data to any specific requirement. Facial attributes may be designed, modified or extracted in arbitrary ways.

Though the presented results are already very promising, one could explore whether combining real-world data with synthetic training data could further improve the reported results. Also, when combining many view-dependent classifiers, it would be suitable to perform an additional post-processing step as, for example, vector boosting. For the detection of facial skin, advanced color models could be helpful to enhance our results. These improvements will be part of future work.

## References

[1] V. Jain and E. Learned-Miller, "FDDB: A benchmark for face detection in unconstrained settings," Univ. of Massachusetts, Amherst, Tech. Rep., 2010.

[2] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3D faces," *ACM Transactions on Graphics (SIGGRAPH'99)*, pp. 187–194, 1999.

[3] K. Scherbaum, M. Sunkel, H.-P. Seidel, and V. Blanz, "Prediction of Individual Non-Linear Aging Trajectories of Faces," *Comput. Graphics Forum (EUROGRAPHICS'07)*.

[4] P. Viola and M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," *IEEE Conf. on Comput. Vision and Pattern Recog. (CVPR'01)*, p. 511, 2001.

[5] M.-H. Yang, D. J. Kriegman, and N. Ahuja, "Detecting faces in images: a survey," *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI'02)*, pp. 34–58, 2002.

[6] C. Zhang and Z. Zhengyou, "A survey of recent advances in face detection," *Microsoft Research, Tech. Report MSR-TR-2010-66*, 2010.

[7] P. Viola and M. Jones, "Robust real-time face detection," *Intl. J. Comput. Vision (IJCV'04)*, pp. 137–154, 2004.

[8] J. Kong and Y. Deng, "GPU accelerated face detection," *Intl. Conf. on Intell. Control and Inf. Process. (ICICIP'10)*, '10.

[9] “A software-based dynamic-warp scheduling approach for load-balancing the Viola-Jones face detection algorithm on GPUs,” *J. Parallel and Distrib. Comput.*, 2013.

[10] B. Sharma, R. Thota, N. Vydyanathan, and A. Kale, “Towards a robust, real-time face processing system using CUDA-enabled GPUs,” *Intl. Conf. on High Performance Comput. (HiPC’09)*, pp. 368–377, 2009.

[11] G. Wei and C. Ming, “The face detection system based on GPU+CPU desktop cluster,” *Intl. Conf. on Multimedia Technol. (ICMT’11)*, pp. 3735–3738, 2011.

[12] S. L. H. Chuang Jan Chang, “LSO-AdaBoost Based Face Detection for IP-CAM Video,” *Applied Mechanics and Mater.*, pp. 3543–3548, 2013.

[13] J. Cho, B. Benson, S. Mirzaei, and R. Kastner, “Parallelized Architecture of Multiple Classifiers for Face Detection,” *IEEE Intl. Conf. on Application-specific Syst., Architectures and Processors (ASAP’09)*, pp. 75–82, 2009.

[14] N. Zhang, “Working towards efficient parallel computing of integral images on multi-core processors,” *Intl. Conf. on Comput. Eng. and Technol. (ICCET’10)*, pp. 30–34, 2010.

[15] M.-T. Pham, Y. Gao, V. Hoang, and T.-J. Cham, “Fast polygonal integration and its application in extending haar-like features to improve object detection,” *IEEE Conf. on Comput. Vision and Pattern Recog. (CVPR’10)*, pp. 942–949, 2010.

[16] C.-H. Chiang, C.-H. Kao, G.-R. Li, and B.-C. Lai, “Multi-level parallelism analysis of face detection on a shared memory multi-core system,” *Intl. Symp. on VLSI Design, Automation and Test (VLSI-DAT’11)*, pp. 1–4, 2011.

[17] Y.-T. Wu, Y.-T. Wu, C.-Y. Cho, S.-Y. Tseng, C.-N. Liu, and C.-T. King, “Parallel Integral Image Generation Algorithm on Multi-core System,” *IEEE Intl. Symp. on Parallel and Distrib. Process. with Applications (ISPA’11)*, pp. 31–35, 2011.

[18] B.-C. C. Lai, C.-H. Chiang, and G.-R. Li, “Data locality optimization for a parallel object detection on embedded multi-core systems,” *IEEE Intl. Conf. on Software Eng. and Service Science*, pp. 576–579, 2011.

[19] M. Everingham, A. Zisserman, C. Williams, L. V. Gool, M. Allan, C. Bishop, O. Chapelle, N. Dalal, T. Deselaers, G. Dorko, S. Duffner, J. Eichhorn, J. Farquhar, M. Fritz, C. Garcia, T. Griffiths, F. Jurie, D. Keysers, M. Koskela, J. Laaksonen, D. Larlus, B. Leibe, H. Meng, H. Ney, B. Schiele, C. Schmid, E. Seemann, J. Shawe-Taylor, A. Storkey, S. Szedmak, B. Triggs, I. Ulusoy, V. Viitaniemi, and J. Zhang, “The’05 pascal visual object classes challenge,” *in 1st PASCAL Challenges Workshop*, 2005.

[20] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, “LabelMe: A Database and Web-Based Tool for Image Annotation,” pp. 157–173, 2008.

[21] A. Torralba, R. Fergus, and W. Freeman, “80 Million Tiny Images: A Large Data Set for Nonparametric Object and Scene Recognition,” *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI’08)*, pp. 1958–1970, 2008.

[22] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” *Conf. on Comput. Vision and Pattern Recog. (CVPR’09)*, 2009.

[23] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman, “Learning Object Categories from Google’s Image Search,” *IEEE Intl. Conf. on Comput. Vision (ICCV’05)*, pp. 1816–1823, 2005.

[24] J. Jeon, V. Lavrenko, R. Manmatha, “Automatic image annotation & retrieval using cross-media relevance models” 2003.

[25] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman, “Discovering objects and their location in images,” *IEEE Intl. Conf. on Comput. Vision (ICCV’05)*.

[26] A. Makadia, V. Pavlovic, and S. Kumar, “Baselines for Image Annotation,” *Intl. J. Comput. Vision*, pp. 88–105, 2010.

[27] D. Tsai, Y. Jing, Y. Liu, H. Rowley, S. Ioffe, and J. Rehg, “Large-scale image annotation using visual synset,” *IEEE Intl. Conf. on Comput. Vision (ICCV’11)*, pp. 611–618, 2011.

[28] L. Denoyer and P. Gallinari, “A Ranking Based Model for Automatic Image Annotation in a Social Network,” *Intl. AAAI Conf. on Weblogs and Social Media (ICWSM’10)*, 2010.

[29] Y.-Y. Chen, W. H. Hsu, and H.-Y. M. Liao, “Learning facial attributes by crowdsourcing in social media,” *Intl. Conf. Companion on World Wide Web (WWW’11)*, 2011.

[30] R. Gross, “Face Databases,” *Handbook of Face Recogn. 2005.*

[31] M. Grgic, http://www.face-rec.org/databases/, last checked: 04/2013, Face Recognition Homepage.

[32] R. Frischholz, http://www.facedetection.com/facedetection/datasets.htm, last checked: 04/2013, The Face Detection Homepage.

[33] P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek, “Overview of the Face Recognition Grand Challenge,” *Proc. of the IEEE Conf. on Comput. Vision and Pattern Recog. (CVPR’05)*, pp. 947–954, 2005.

[34] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, “Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments,” Univ. of Massachusetts, Amherst, Tech. Rep., 2007.

[35] Y.-M. Li, J. Chen, L.-Y. Qing, B.-C. Yin, and W. Gao, “Face detection under variable lighting based on resample by face relighting,” *Intl. Conf. on Machine Learn. and Cybern. (ICMLC’04)*, pp. 3775–3780, 2004.

[36] D. Zhou, D. Petrovska-Delacrétaz, and B. Dorizzi, “3D Active Shape Model for Automatic Facial Landmark Location Trained with Automatically Generated Landmark Points,” *Intl. Conf. on Pattern Recog. (ICPR’10)*, pp. 3801–3805, 2010.

[37] V. Blanz, P. Grother, P. Phillips, and T. Vetter, “Face recognition based on frontal views generated from non-frontal images,” *IEEE Conf. on Comput. Vision and Pattern Recog. (CVPR’05)*, 2005.

[38] A. Rama and F. Tarres, “P2CA: a new face recognition scheme combining 2D and 3D information,” *IEEE Intl. Conf. on Image Process. (ICIP’05)*, pp. 776–9, 2005.

[39] M. Toews and T. Arbel, “Detecting and Localizing 3D Object Classes using Viewpoint Invariant Reference Frames,” *IEEE Intl. Conf. on Comput. Vision (ICCV’07)*, pp. 1–8, 2007.

[40] L. Wang, L. Ding, X. Ding, and C. Fang, “Improved 3D assisted pose-invariant face recognition,” *IEEE Intl. Conf. on Acoustics, Speech and Signal Process. (ICASSP’09).*

[41] A. Ansari, M. Mahoor, and M. Abdel-Mottaleb, “Normalized 3D to 2D model-based facial image synthesis for 2D model-based face recognition,” *IEEE GCC Conf. and Exhibition (GCC’11),*, pp. 178–181, 2011.

[42] K. Bowyer, K. Chang, and P. Flynn, “A survey of approaches and challenges in 3D and multi-modal 3D+2D face recognition,” *Comput. Vision and Img. Understanding*, 2006.

[43] L. Pishchulin, T. Thormählen, and C. Wojek, “Learning People Detection Models from Few Training Samples,” *IEEE Conf. on Comput. Vision and Pattern Recog. (CVPR’10).*

[44] B. Weyrauch, B. Heisele, J. Huang, and V. Blanz, “Component-Based Face Recognition with 3D Morphable Models,” *Comput. Vision and Pattern Recog. Workshop (CVPRW’04)*, p. 85, 2004.

[45] A. Frome, G. Cheung, A. Abdulkader, M. Zennaro, B. Wu, A. Bissacco, H. Adam, H. Neven, and L. Vincent, “Large-scale privacy protection in google street view,” *IEEE Intl. Conf. on Comput. Vision (ICCV’09)*, pp. 2373–2380, 2009.

[46] J.-C. Terrillon, H. Fukamachi, S. Akamatsu, and M. N. Shirazi, “Comparative performance of different skin chrominance models and chrominance spaces for the automatic detection of human faces in color images,” *IEEE Intl. Conf. on Autom. Face and Gesture Recog. (FG’00)*, pp. 54–63, 2000.

[47] C. Huang, H. Ai, Y. Li, and S. Lao, “Vector boosting for rotation invariant multi-view face detection,” *IEEE Intl. Conf. on Comput. Vision (ICCV’05).*