Real-time High Performance Deformable Model for Face Detection in the Wild

Junjie Yan Xucong Zhang Zhen Lei Stan Z. Li*
Center for Biometrics and Security Research & National Laboratory of Pattern Recognition
Institute of Automation, Chinese Academy of Sciences, China
{jjyan,xczhang,zlei,szli}@nlpr.ia.ac.cn

Abstract

We present an effective deformable part model for face detection in the wild. Compared with previous systems on face detection, there are mainly three contributions. The first is an efficient method for calculating histogram of oriented gradients by pre-calculated lookup tables, which only has read and write memory operations and the feature pyramid can be calculated in real-time. The second is a Sparse Constrained Latent Bilinear Model to simultaneously learn the discriminative deformable part model, and reduce the feature dimension by sparse transformations for efficient inference. The third contribution is a deformable part based cascade, where every stage is a deformable part in the discriminatively learned model. By integrating the three techniques, we demonstrate noticeable improvements over previous state-of-the-art on FDDB with real-time speed, under widely comparisons with both academic and commercial detectors.

1. Introduction

Face detection is a foundation stone in face based applications and is one of the most important problems in biometric. Frontal face detection systems have been proposed in early years, such as [10, 12, 16, 19, 22]. Among these methods, the framework proposed by Viola and Jones (V-J) [22] is the most popular one for its advantage in speed, and holds the dominant position in face detection during the recent decade. The V-J detector and its subsequences have achieved great successes. However, their performances are far from satisfactory (e.g. on FDDB benchmark [8]), due to the large appearance variations caused by the unconstrained illumination, occlusion, expression and so on.

Although be different in feature representation and learning algorithm, previous face detectors tend to feed a fixed feature representation to a fixed classifier. It would result in the ambiguousness in practice where the large appearance variations can exist. For example, the relative positions of two eyes for different individuals, and the pose and expression variations for the same individual. Instead of taking these appearance variations as blacking boxes, we conduct the deformable part based structural model originally proposed in [2], where every part can have deformation to capture the real world face variations. Considering the efficiency, we make the following three contributions.

The first is an efficient method to compute histogram of oriented gradients (HOG) [1]. HOG is utilized for the advantage in tolerating local geometric and photometric transformation for faces in the wild. However, the original HOG feature is with high computation cost, mainly due to the division and inverse trigonometric operations in calculating the orientation partition. In this paper, we propose a method to avoid the complex operations with pre-calculated lookup tables, where only read and write memory operations are involved. Our HOG feature pyramid can be calculated in about 20ms on a standard PC for VGA image, thus can be ready for real-time applications.

The second is a sparse constrained latent bilinear model for discriminative parameter learning. In the detection phase, the most time consuming operation is the convolution between the HOG feature and the learned template. To reduce the computation cost while keep the discrimination, we propose to learn a sparse transformation to project the original feature to a low dimensional subspace, in which the convolution is conducted. The sparse constrain of transformation is to reduce the computation cost in projection. Particularly, the transformation is defined on the cells of HOG feature. We present a novel sparse constrained latent bilinear algorithm to optimize the sparse transformation and the classifier simultaneously.

The third is a deformable part based cascade in detection. We use the root template as the first stage of the cascade, and the consequent stages are set to be the learned deformable parts. To achieve occlusion invariance in any face region, the score of every part stage is the score of the deformable parts in current stage plus its parent stage. We propose a greedy strategy to convert the learned detector to

*Stan Z. Li is the corresponding author.

be the cascade structure. When the score of a stage does not satisfied a learned threshold, the candidate is rejected immediately. With the cascade structure, our detector keeps the advantage of model flexibility, while avoids a lot of unnecessary computation.

By conducting the DPM to face detection and integrating the three techniques above, we dramatically improve the face detection performance with the real-time speed, according to the experimental comparisons on FDDB [8]. Our method achieves 85.5% detection rate at 0.5 false-positive-per-image, while the previous best published result [13] is 72.4% and the best commercial system is 83.2%.

The rest of the paper is organized as follows. In section 2, we review the related work. The feature, classifier, and cascade are discussed in Section 3, 4, 5, respectively. The experimental comparisons are discussed in section 6, and finally in section 7, we conclude the paper.

2. Related Work

The deformable part based representation is related to recent proposed object detection and pose estimation systems, including [2, 4, 25, 24, 3]. But these models are not suitable for face based applications for the high computation cost. To save the computation cost while still enjoy the flexibility, we propose a sparse bilinear model to jointly learn a sparse transformations, and a classifier in the compact subspace. The bilinear model is motivated by [18, 17], but we constrain the sparsity, which results in quite different formulation and optimization.

Part based face representation has been explored in early years [15, 5, 20]. But how to effectively use these models in real world face detection is still unclear. The most related recent work is [26, 23], which used local parts around landmarks to represent face, and proposed a tree structure deformable model for joint face detection, landmark location and pose estimation with promising performance. Besides the differences in model learning and cascade based detection, our model have three advantages over [26]. Firstly, [26] needs landmark annotations, while our model can be automatically learned from coarse bounding boxes annotations. Secondly, our method performs much better than [26] on face detection experiments. Finally, our method runs at real-time, while [26] takes about 10s for an image.

3. Feature

The mainstream face detectors always utilized the simple features (e.g. Wavelet, Haar, LBP) with nonlinear classifiers (e.g. boosting, kernel SVM). Differently, we explore the advantages of complex feature in face detection, such as gradient of oriented histograms (HOG) for the following reasons. Firstly, it's robust in capturing local geometric and photometric transformations, which is very important for face detection in the wild, where possible large appearance variations exist. Secondly, it's of low dimension (e.g. some thousands dimension for a face), for which the machine learning technique is very mature. Finally, the histogram feature itself provides high nonlinearity, and we can use simple linear classifiers in model learning. However, HOG is often with high computation cost. We first review the standard HOG computation procedure, and then show how to speed it up.

The standard procedure of calculating HOG pyramid is described as follows. Given an input image I, we resize it into different scales to build an image pyramid. For each scale, we compute the gradient in (x, y) as (dx, dy) with the convolution kernel [-1 0 1] and its transpose. Then the orientation of gradient at each pixel is calculated and discretized into different partitions. As discussed in [1], the discretization can be divided into contrast sensitive and insensitive, where the orientation range belongs to $[-180°, 180°]$ and $[0°, 180°]$, respectively. If the orientation of gradient at the pixel belongs to the i-th partition, its magnitude is added to the corresponding bin of the histogram in the cell. In our experiment, we set the number of contrast sensitive bins to be 18, and the number of contrast insensitive bins to be 9. The 4 different types of energy as defined in [1] is also added to reflect the energy of the cell. Finally, the contrast sensitive and insensitive features are normalized by the energy.

The gradient computation step only involves sum operations, which are very efficient. With a detailed analysis of the standard HOG implementation, we found that most of the time are spent on calculating the orientation and the "bilinear" interpolation, where division and inverse trigonometric operations involved.

Fortunately, we find that these complex operations can be avoided for nearly free under the assumption that the image gray value is integral number and ranges in $[0, 255]$. Note that this assumption can always hold, and the images of other types can be normalized easily. Since x and y are in $[0, 255]$, the range of dx, dy are $[-255, 255]$. We build two 511×511 matrices L_1 and L_2, where $L_1(i, j)$ and $L_2(i, j)$ store the partition index of contrast sensitive and insensitive discretization when the $dx = i - 255$ and $dy = j - 255$. The L_1 and L_2 can be computed in the model initialization phase or stored in advance, thus take no time in the runtime. With the help of L_1 and L_2, for each pixel (x, y) and its gradients (dx, dy), the orientation discretization index can be got immediately by lookup $L_1(dx + 255, dy + 255)$ and $L_2(dx + 255, dy + 255)$, which is very efficient compared with the complex division and inverse trigonometric operations. Similar idea can also be used in computing the weight of "bilinear" interpolation in soft aggregation.

To detect faces in multiple scales, we need to build a feature pyramid, which includes image resizing operations.

Although thought to be complex, it can be very efficient with the optimized code on modern hardware. We set the number of intervals between two octaves as 5, and the corresponding pyramid has 23 scales for VGA image. In this configurations, the total computation time for HOG pyramid construction is about 20ms.

4. Classifier

The deformable part based model (DPM) is originally proposed in [2] for general object detection. In this paper, we conduct it to represent face, and propose novel method for model learning. The detail of the model in representation is referred to [2] for the space limitation here.

4.1. Sparse Constrained Bilinear Learning

The model defined in DPM is of the linear form, so that can be learned by mature linear classifiers. Since the annotations of part locations are often not available, they are taken as latent variable in the training phase, and optimized by the following objective function:

$$\arg\min_{w} \frac{1}{2}\|w\|^2 + C\sum_{n}\max(0, 1 - y_n w^T \phi(I_n, \Theta_n^*)) \quad (1)$$
$$s.t. \quad \Theta^* = \arg\max_{\Theta_n} w^T \phi(I_n, \Theta_n)$$

where the first term is used for regularization, and the second term is the Hinge loss. $\phi(I_n, \Theta_n^*)$ is the feature vector of the image I_n with the face configuration Θ_n^*, by concatenating the global appearance feature, part appearance feature and the deformation feature. w is the concatenation of global template, part templates, and part deformation parameters with the same order. The configuration parameters Θ^* are taken as latent variable, and inferred in the runtime by maximizing $w^T\phi(I_n, \Theta_n^*)$ on all the possible deformations. y_n is 1 for positive samples, and -1 for negative samples. The Eq. 1 can be solved with the Latent-SVM algorithm proposed in [2], where a coordinate descent procedure is conducted to iteratively optimize the latent part deformation parameters Θ_n^*, and the model parameters w.

To simplify the notation, we use the an equivalent matrix based notation of Eq. 1:

$$\arg\min_{W_a, w_s} \frac{1}{2}\|W_a\|_F + \frac{1}{2}\|w_s\|_2 \quad (2)$$
$$+C\sum_{n}\max[0, 1 - y_n(Tr(W_a^T \Phi_a(I_n, \Theta_n^*) + w_s^T \phi_s(\Theta_n^*)))]$$

where $\Phi_a(I_n, \Theta_n^*)$ is a $n_f \times n_c$ dimensional feature matrix generated by concatenating the appearance feature of root and each part. Every column of $\Phi_a(I_n, \Theta_n^*)$ is a n_f dimensional HOG feature vector of a cell, and n_c is the number of cells in root and parts. W_a is a $n_f \times n_c$ dimensional feature matrix generated with the same way as $\Phi_a(I_n, \Theta_n^*)$. w_s is a vector generated by concatenating all the spatial terms w_{s_i}. $Tr(\cdot)$ is the trace operation. $\|\cdot\|_F$ is the Frobenius norm, and $\|W_a\|_F = Tr(W_a^T W_a)$. The Eq. 2 equals to Eq. 1, but it can give some insights of the learning, which will be explored below.

The dimension of appearance feature determines the efficiency in the detection phase. What's more, different dimensions in the feature $\phi_a(I_n, \Theta_n)$ may have the redundance. For the two reasons, we propose a bilinear model to learn a compact subspace of the original feature space, where the convolution is conducted, so that the computation cost in detection can be reduced, and the redundance can be avoided. To this end, the learning algorithm is divided into learning the two parts jointly, the transformation matrix to project the original feature to a compact subspace, and the classifier on the learned subspace. Particularly for our problem, the transformation matrix is defined on the n_f dimensional feature of cell in HOG. Here we use a bilinear formulation to solve the chicken and egg problem. Note that the number of convolution operations is reduced with the cost of projection operations. In order to reduce the cost in projection, we also add a sparse regularization to prefer the sparse transformation matrix, where the zero items can be avoided in the projection. The object function of our proposed sparse constrained latent bilinear model is defined as:

$$\arg\min_{W_a, w_s, P} \frac{1}{2}\|P^T W_a\|_F + \frac{1}{2}\|w_s\|_2 + \|Vec(P)\|_1 \quad (3)$$
$$+C\sum_{n}\max_{\Theta_n^*}[0, 1 - y_n(Tr(W_a^T P \Phi_a(I_n, \Theta_n^*) + w_s^T \phi_s(\Theta_n^*)))]$$

where $Vec(\cdot)$ is the operator to reshape a matrix to be a vector. P is a $n_f \times n_d$ dimensional transformation matrix, and the $\|Vec(P)\|_1$ is used as a relaxation of l^0 norm. Note that here we constrain that $n_d < n_f$. It projects the original high feature space to a low dimensional space, and then applys the convolution. Consequently, the regularization is conducted on $P^T W$ globally. To solve the non-convex problem with latent variable, we divide it into the following subproblems, and solve them iteratively.

4.1.1 Fix P to solve W_a and w_s

When the transformation matrix P is fixed, the regularization term $\|Vec(p)\|_1$ in Eq. 3 can be removed. By denoting PP^T as A, $A^{\frac{1}{2}}W_a$ as $\widetilde{W_a}$, and $A^{-\frac{1}{2}}P\Phi_a(I_n, L_n^*)$ as $\widetilde{\Phi_a}(I_n, L_n^*)$. It can be shown that object function can be reformulated as:

$$\arg\min_{\widetilde{W_a}, w_s} \frac{1}{2}\|\widetilde{W_a}\|_F^2 + \frac{1}{2}w_s^T w_s \quad (4)$$
$$+C\sum_{n}\max[0, 1 - y_n(Tr(\widetilde{W_a}^T \widetilde{\Phi_a}(I_n, L_n^*)) + w_s^T \phi_s(L_n^*))]$$

which has the same form with the optimization problem in Eq. 2, and the Latent-SVM solver can be used here. Note that the dimension of $\widetilde{W_a}$ is $n_d \times n_c$, which is smaller than than original $n_f \times n_c$ dimensional classifier W_a. Once the solution to Eq. 4 is achieved, W_a is computed by $(PP^T)^{-\frac{1}{2}}\widetilde{W_a}$.

4.1.2 Fix W_a and w_s to solve P

When the W_a and w_s are fixed, we iteratively relax the F norm of $P^T W_a$ and the l^1 norm of P, while keeps the detection loss the same, which results in the following two subproblems.

Relax the F Norm. When the term $\frac{1}{2}\|P^T W_a\|_F$ is ignored, the problem becomes to be:

$$\arg\min_{P} \|Vec(P)\|_1 \tag{5}$$
$$+C\sum_n \max_{\Theta_n^*}[0, 1 - y_n(Tr(W_a^T P \Phi_a(I_n, \Theta_n^*) + w_s^T \phi_s(\Theta_n^*)))]$$

which can be transformed to be standard linear programming problem, and solved by efficient simplex method.

Relax the l^1 Norm. When the term $\|Vec(P)\|_1$ is ignored, and W_a and w_s are fixed, we first inference the part location of every training samples Θ_n^* by finding the part configurations to maximize Eq. 3. Denoting $W_a W_a^T$ as A, $A^{\frac{1}{2}}P$ as $\widetilde{P}$, and $A^{-\frac{1}{2}}W_a\Phi_a(I_n, \Theta_n^*)^T$ as $\widetilde{\Phi_a}(I_{n_n}, \Theta_n^*)$, the object function equals to:

$$\arg\min_{\widetilde{P}} \frac{1}{2}\|\widetilde{P}\|_F^2 \tag{6}$$
$$+C\sum_n \max[0, 1 - y_n(Tr(\widetilde{P}^T\widetilde{\Phi_a}(I_n, \Theta_n^*)) + w_s^T \phi_s(\Theta_n^*))]$$

The only difference between Eq. 6 and standard SVM is an additional term $w_s^T\phi_s(\Theta_n^*)$. Since $w_s^T\phi_s(\Theta_n^*)$ is a constant in the optimization, it can be taken as an additional dimension of $Vec(\widetilde{\Phi_a}(I_n, \Theta_n^*))$. In this way, the Eq. 6 can be solved by the standard SVM solver. After we get $\widetilde{P}$, the P can then be computed by $(W_a W_a^T)^{-\frac{1}{2}}\widetilde{P}$.

In our implementation, we calculate the PCA of HOG features extracted from randomly generated patches, and the first n_d eigenvectors are combined as the initial value of P. After that, we iteratively fix P to optimize W_a and w_s, and fix W_a and w_s to optimize P. In optimizing P, we further iteratively relax the l^1 and F norm to wrap the problem to be standard SVM and linear programming problem. Once we get the optimized P, W_a and w_s, we can project the original HOG feature to be a low dimensional compact subspace, and conduct the convolution with the learned W_a on it, so that a lot of computations are avoided.

5. Cascade

The learned detector consists of a root template and a set of part templates. Although we have learned a compact subspace to reduce the computation cost, it still can not run in real-time on standard PC, mainly due to the calculation of the appearance scores for all parts. Here we present a method to convert the learned non-cascade model to be a part based cascade, which finally makes our detector real-time.

The detector is applied to all the possible candidate face regions in the detection procedure. For a candidate region b in image I, we use the following additive cascade structure:

$$S_n(I_b) = S_{n-1}(I_b) + F_n(I_b) \tag{7}$$

where $F_n(I_b)$ is the response in the n-th stage, and $S_n(I_b)$ is the detection score. There is a threshold t_n in every stage. If $S_n(I_b) > t_n$, the candidate is passed to the next stage for further consideration, otherwise is taken as negative sample and rejected immediately. Every stage is set to be a deformable part, includes both the appearance score and spatial score. With the cascade structure, the detector only pays attention to promising regions and the overwhelming majority of negative samples can be rejected in early stages. Then the problem becomes to be how to convert the learned non-cascade model to be a cascade model.

Here we conduct a greedy algorithm to select the order of the parts in the cascade. Motivated by [3], we use the learned model to detect the positive and negative samples, and cache the appearance score and spatial score of each part and root. To achieve occlusion invariance, the first stage is set to be the root template. Since the score of the root can be passed to the sequent stages, the effects of occlusion in a special part can be avoid naturally. For each consequent stage, we greedily select a part with the procedure described in Algorithm 1.

A candidate part list is kept in Algorithm 1. At each loop, we greedily select a part that rejects most of negative samples with the pre-defined true positive rate. Once the part is selected, we pop it from the candidate list, and select its succeed parts from the remaining candidate part set.

6. Experiments

In this part, we show the details in training the proposed face detection model, and compare our method with other state-of-the-art methods on challenging FDDB [8].

6.1. Practical Training

Our face model is trained on a subset of AFLW database [11], which is a newly released database collected from Flickr. For the unconstrained nature of Flickr, the images exhibit large variations in pose, illumination, expression, ethnicity, and imaging conditions. There are 25933 faces in

Cache the detection score of root template and parts for training positive and negative samples;
Select the root template as the 0-th stage, and push the part set into the candidate list;
for $i \leftarrow 1$ **to** N **do**
  for $j \leftarrow 1$ **to** $N - i$ **do**
    Calculate S_i for positive and negative samples if the j-th part in the candidate list is taken as the i-th stage model;
    Set the threshold to satisfy that the true positive rate is ρ^i, and calculate the corresponding false positive rate;
  end
  Select the part with the lowest false positive rate to be the ith stage model, and pop it from the candidate list;
end

Algorithm 1: The greedy algorithm to convert the model to be a cascade.

21997 images. Besides the face bounding boxes, the 21 keypoint annotations are also provided, thus the 3D face pose can be estimated.

Our face model consists of eight different views: $[-90°, -60)$,$[-60°, -30°)$,$[-30°, -15°)$,$[-15°, 0)$,$[0°, 15°)$, $[15°, 30°)$, $[30°, 60°)$,$[60°, 90°]$, according to the yaw angle. We select 2092 faces for the $[0°, 15°)$ model, 2104 faces for the $[15°, 30°)$ model, 2104 faces for the $[30°, 60°)$ model, and 2224 faces for the $[60°, 90°]$ model. The learned model of each face view consists of a global template with 9×9 HOG cells, and eight part templates with 6×6 HOG cells. The cell size of HOG is set to be 8×8 pixel. To suppress the repetitive detections, we use NMS (Non Maximum Suppression) [1] as a post-process step, and the overlap threshold is set to be 0.5. The initial true positive rate ρ in cascade learning is set to be 0.999.

6.2. Experiment on FDDB

FDDB [8] is one of the most widely used benchmark for face detection in unconstrained setting. It contains 2845 images and 5171 faces collected from the news photograph. Following the standard protocol in [8], we report the average discrete and continuous ROC of the ten subfolders.

We compare our result with the top 6 academic methods, and 2 commercial systems listed on the FDDB webpage, including: (1) Olaworks face detector[1], which is a commercial system, and achieved previous best result on FDDB; (2) IlluxTech[2] frontal face detector, which is another commercial system; (3) SURF cascade face detector [13], which is based on SURF feature and a modified boosting algorithm; (4) Jain's face detector [9], where the image context is used to improve boosting based algorithm; (5) OpenCV V-J face detector [3], which is one of the most popular open source face detector; (6) Subburaman's face detector [21], which improves V-J detector in sliding window phase; (7) Mikolajczyk's face detector [14], which used additional body information. (8) Tree structure face model (TSM) [26]. Note that all the results except OpenCV are reported by their authors thus can guarantee the best parameters used. For TSM and our detector, we resize the original image two times.

The ROC curves and some of the detection examples are shown in Fig. 1. Our proposed method can achieve 85.5% true positive rate with 142 false positives (there are 284 faces in average for each subfolder in FDDB, so that this corresponding to 0.5FPPI). The best commercial system Olaworks face detector gets 83.2% true positive rate and the best academic system TSM [26] can only get 72.4% true positive at the same FPPI.

6.3. Discussion

Here we compare the proposed detector with the dominant V-J detector and recently proposed tree structure model [26] for face detection in the following three aspects.

Effectiveness From the experimental comparisons on challenging FDDB, we can find that our model achieves a large performance margin over published V-J based methods and tree structure method for unconstrained face detection.

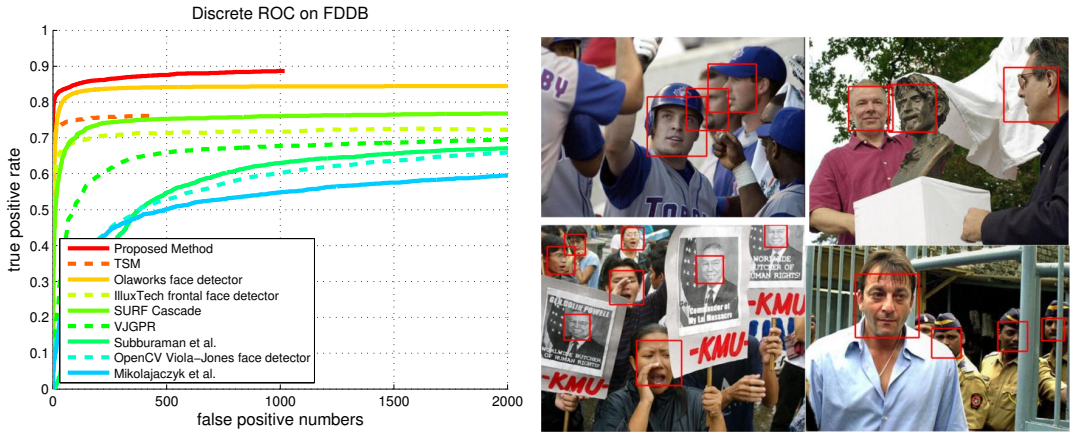
Training Samples Traditional V-J based methods often need millions of faces or more to get the desired performance, which makes data collection a difficult problem. Benefiting from the low dimensional histogram feature and the generalization of bilinear model, the proposed method can achieve good performance by thousands of training instances. Recently proposed tree structured model [26] are with the same property, but additional landmark annotations are needed.

Efficiency The V-J detector is very efficient for real applications. The tree structure model [26] can get better accuracy but with high computation cost (0.025-0.1FPS). Benefiting from the methods discussed above, our eight view face detector runs about 10 FPS and frontal face detector runs about 22 FPS on a PC laptop with Intel Core i5 CPU for VGA image. Considering the large performance improvement and the coming better hardware, we believe that our method would be more competitive.

7. Conclusion

In this paper, an effective deformable part model is built for unconstrained face detection. We argue the advantages

[1] http://eng.olaworks.com/olaworks/main/
[2] http://illuxtech.com
[3] http://sourceforge.net/projects/opencvlibrary/



(a) Discrete score ROC on FDDB

(b) Examples on FDDB

Figure 1. Quantitative and qualitative results on FDDB.

of gradient histogram feature in face detection and propose a method to compute it efficiently. Furthermore, We develop a sparse constrained latent bilinear model for jointly learning the model parameters and reducing the model dimension. Finally, we show how to convert the learned model to be a occlusion invariant deformable part based cascade for further speedup. State-of-the-art detection performance is achieved on FDDB, compared with both academic and commercial systems.

Acknowledgement

This work was supported by the Chinese National Natural Science Foundation Project #61070146, #61105023, #61103156, #61105037, #61203267, National IoT R&D Project #2150510, National Science and Technology Support Program Project #2013BAK02B01, Chinese Academy of Sciences Project No. KGZD-EW-102-2, European Union FP7 Project #257289 (TABULA RASA), and Authen-Metric R&D Funds.

References

[1] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*. IEEE, 2005.

[2] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *TPAMI*, 2010.

[3] P. Felzenszwalb, R. Girshick, and D. a. McAllester. Cascade object detection with deformablepresumably part models. In *CVPR*. IEEE, 2010.

[4] R. B. Girshick, P. Felzenszwalb, and D. Mcallester. Object detection with grammar models. In *NIPS*. 2011.

[5] B. Heiselet, T. Serre, M. Pontil, and T. Poggio. Component-based face detection. In *CVPR*. IEEE, 2001.

[6] O. Inc. http://eng.olaworks.com.

[7] Y. Inc. http://www.flickr.com/.

[8] V. Jain and E. Learned-Miller. Fddb: A benchmark for face detection in unconstrained settings. Technical report, University of Massachusetts, Amherst, 2010.

[9] V. Jain and E. Learned-Miller. Online domain adaptation of a pre-trained cascade of classifiers. In *CVPR*. IEEE, 2011.

[10] T. Kanade. *Picture processing system by computer complex and recognition of human faces*. Department of Science, Kyoto University, 1973.

[11] M. Kostinger, P. Wohlhart, P. Roth, and H. Bischof. Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization. In *ICCV Workshop*. IEEE, 2011.

[12] C. Kotropoulos and I. Pitas. Rule-based face detection in frontal views. In *ICASSP*. IEEE, 1997.

[13] J. Li, T. Wang, and Y. Zhang. Face detection using surf cascade. In *ICCV Workshops*. IEEE, 2011.

[14] K. Mikolajczyk, C. Schmid, and A. Zisserman. Human detection based on a probabilistic assembly of robust part detectors. *ECCV*, 2004.

[15] B. Moghaddam and A. Pentland. Probabilistic visual learning for object representation. *PAMI*, 1997.

[16] E. Osuna, R. Freund, and F. Girosit. Training support vector machines: an application to face detection. In *CVPR*. IEEE, 1997.

[17] H. Pirsiavash and D. Ramanan. Steerable part models. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3226–3233. IEEE, 2012.

[18] H. Pirsiavash, D. Ramanan, and C. Fowlkes. Bilinear classifiers for visual recognition. In *NIPS*, 2009.

[19] H. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *TPAMI*, 1998.

[20] H. Schneiderman and T. Kanade. Object detection using the statistics of parts. *IJCV*, 2004.

[21] V. Subburaman and S. Marcel. Fast bounding box estimation based face detection. In *ECCV Workshop*, 2010.

[22] P. Viola and M. Jones. Robust real-time face detection. *IJCV*, 2004.

[23] J. Yan, X. Zhang, Z. Lei, D. Yi, and S. Li. Structural models for face detection. In *FG*. IEEE, 2013.

[24] Y. Yang and D. Ramanan. Articulated human detection with flexible mixtures of parts. *TPAMI*, To appear.

[25] S. Zhu and D. Mumford. *A stochastic grammar of images*. 2007.

[26] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *CVPR*. IEEE, 2012.