FACE DETECTION USING LOCAL HYBRID PATTERNS

Chulhee Yun, Donghoon Lee, and Chang D. Yoo

Department of Electrical Engineering, Korea Advanced Institute of Science and Technology
chyun90@kaist.ac.kr, iamdh@kaist.ac.kr, cdyoo@ee.kaist.ac.kr

ABSTRACT

This paper examines a novel binary feature referred to as the Local Hybrid Patterns (LHP) that is generated by mixing highly discriminative bits of the binary local pattern features (BLPFs) such as the Local Binary Patterns (LBP), Local Gradient Patterns (LGP), and Mean LBP (MLBP). Starting with the most discriminative BLPF selected, the LHP generating algorithm iteratively updates the bits of the selected BLPF by replacing the least discriminative bit with the most discriminative bit of all the candidate BLPFs. At the expense of a small increase in computation, the LHP is guaranteed to give smaller or equal empirical error compared to any BLPFs considered in the pool. Experimental comparison of different sets of features consistently shows that the LHP leads to better performance than previously proposed methods under the AdaBoost face detection framework on MIT+CMU and FDDB benchmark datasets.

Index Terms— Local hybrid pattern, feature combination, mean local binary pattern, AdaBoost, face detection

1. INTRODUCTION

The most popular face detectors today are based on the Viola and Jones [1] detector, which is constructed in a cascaded manner using the AdaBoost [2] algorithm with Haar-like features. Unfortunately, this detector and its variants [3, 4] lack robustness to different lighting conditions.

One feature that has enhanced the detector's robustness to global illumination variation is the Local Binary Patterns (LBP) [5], which is a binary local pattern feature (BLPF) generated from an image patch by comparing the intensity of the center pixel with its neighboring pixels and concatenating the binary comparison results. The use of relative intensity makes it more robust against global illumination variation without incurring much computational overhead. As a result, the LBP and its variants have been extensively utilized for face detection [6, 7, 8].

Another BLPF which has received some attention is the Local Gradient Patterns (LGP) [9]. It is considered to be robust against local illumination variation. After computing the absolute gradient between the center pixel and its neighbors, the LGP is obtained by comparing the gradients with their mean.

The BLPFs can be extended to multi-block BLPFs. The multi-block LBP (MB-LBP) [7] and multi-block LGP (MB-LGP) [10] are obtained by extending pixel-based operation to block-based operations: instead of pixel values, mean values of blocks are used during the feature extraction to better describe local structures of an image. It has been shown that multi-block features perform better than pixel-based BLPFs [7] for face detection.

The Mean LBP (MLBP) [11] was first proposed for the face recognition task without drawing much attention in face detection literature. In this paper, MLBP and its multi-block extension, Multi-block MLBP (MB-MLBP) are considered as feature candidates.

Recent object detection research draws upon the strengths of a wide variety of features [12, 13, 14]. In [14], the AdaBoost algorithm combines the most discriminative BLPF from a pool of candidate features and image patch locations such that at each round the feature and patch location generating the smallest error is chosen among all candidates.

Previously, a BLPF encoded as a fixed length bit-string was selected at each round of the AdaBoost face detection framework, and all the bits in the string were generated under the same operation. In this paper, the fixed length bit-string is likewise selected at each round but the bits can originate from different types of BLPFs which means each bit in the bit-string can be generated under different operations. In other words, we extend feature combination from "word-wise" to "bit-wise".

The remaining sections of this paper are organized as follows. The details of the LHP generating algorithm is described in Section 2. Experimental results are described in Section 3. Finally, Section 4 summarizes the paper.

2. METHOD

In this paper, let us define $F(I; a, b, x, y)$ to denote an L-bit BLPF F extracted from an input image I at position (x, y) with a block size $a \times b$. The blocks are generated from $3a \times 3b$ image patch by dividing it into 3×3 grid, and a $w \times h$ image

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIP)(No.NRF-2010-0028680).

contains $(w-3a+1) \times (h-3b+1)$ patches. Note that LBP and LGP are computed using 1×1 size blocks (or pixels).

When training a face detector using word-wise methods, AdaBoost algorithm calculates errors of all BLPFs and patch positions $F_i(\cdot; a, b, x, y)$ using the training images and chooses the best feature $\hat{F}$ and patch $(\hat{a}, \hat{b}, \hat{x}, \hat{y})$ which shows the smallest empirical error. The error is calculated as follows. For each feature and patch, we construct two histograms H_p and H_n for positive and negative datasets, respectively. The histograms have 2^L bins, each corresponding to a feature value. Given m_p positive training images $I_{p1}, \cdots, I_{pm_p}$ and their weights $w_{p1}, \cdots, w_{pm_p}$, the height of the histogram H_p at feature value v is $H_p(v) = \sum_{j=1}^{m_p} \mathbf{1}\{F_i(I_{pj}; a, b, x, y) = v\} w_{pj}$, where $\mathbf{1}\{\cdot\}$ is the indicator function. H_n is constructed in a similar way, and the error of $F_i(\cdot; a, b, x, y)$ is defined as $\epsilon(F_i, a, b, x, y) = \sum_{v=1}^{2^L} \min\{H_p(v), H_n(v)\}$. The details of the word-wise approaches can be found in [15]. Error calculation requires a significant amount of computation, which makes the number of error calculation a dominant factor in time complexity.

2.1. Local Hybrid Patterns

Suppose we are given N types of BLPFs $F_1, F_2, \ldots, F_N$, each having P candidate image patches indexed by (a, b, x, y). Also, we assume that all features have bit length L. The word-wise methods need to perform PN error calculations.

The intuitive idea of this paper is to find the most discriminative L bits from N types of BLPFs. However, exhaustively examining all possible combinations of bits is almost computationally impossible. Each image patch produces NL bits, and L bits should be chosen; in total, there are $P \times \binom{NL}{L}$ candidates compared to PN candidates in the previous methods. For instance, in our biggest experiment setting ($P = 1,517$, $N = 3$, and $L = 8$), the number of candidates explosively increases from 4,551 to 1,115,709,507. This makes conventional feature selection algorithms such as AdaBoost and random forests [16] infeasible to use; we have to test billions of candidates during each round.

Our method remedies this problem by effectively and efficiently choosing the discriminative bits in a greedy manner as described in Algorithm 1. First, we now define the N BLPFs as sets consisting of their bits: $F_i = \{f_i^1, \cdots, f_i^L\}$. We also define an error calculation function $\epsilon(F, a, b, x, y)$ which takes a set of BLPF bits F and an image patch (a, b, x, y), and returns the empirical error through the error calculation procedure described at the beginning of this section.

We firstly choose the $N+1$ candidate patches from P patches based on errors of original features: N with the smallest errors on individual features (lines 1–4), and one with the smallest average error (lines 5–6). Then we perform a greedy local search on each patch. Starting from the L bits of the best feature on the patch, we iteratively update the bits by: (1) choosing and removing the least conducive bit among the

Algorithm 1 Extraction process of Local Hybrid Patterns

// Obtain patch locations
1: **for** $i = 1$ to N **do**
2: $\quad (\hat{a}_i, \hat{b}_i, \hat{x}_i, \hat{y}_i) \leftarrow \mathrm{argmin}_{(a,b,x,y)}\, \epsilon(F_i, a, b, x, y)$
3: $\quad \hat{F}_i \leftarrow F_i$
4: **end for**
5: $(\hat{a}_0, \hat{b}_0, \hat{x}_0, \hat{y}_0) \leftarrow \mathrm{argmin}_{(a,b,x,y)} \sum_{i=1}^{N} \epsilon(F_i, a, b, x, y)$
6: $\hat{F}_0 \leftarrow \mathrm{argmin}_{F \in \{F_1, \cdots, F_N\}}\, \epsilon(F, \hat{a}_0, \hat{b}_0, \hat{x}_0, \hat{y}_0)$

// Local greedy method to obtain LHP
7: **for** $i = 0$ to N **do**
$\quad$ *// Initizlizing*
8: $\quad F_{\text{sel}}^i \leftarrow \hat{F}_i;\ F_{\text{unsel}}^i \leftarrow (\cup_{j=1}^{N} F_j) \backslash \hat{F}_i$
9: $\quad e_{\text{cur}}^i \leftarrow \epsilon(\hat{F}_i, \hat{a}_i, \hat{b}_i, \hat{x}_i, \hat{y}_i);\ c \leftarrow 0$
10: $\quad$ **while** $c < L/2$ **do**
$\qquad$ *// Select the least discriminative bit*
11: $\qquad f_{\text{worst}} \leftarrow \mathrm{argmin}_{f \in F_{\text{sel}}^i}\, \epsilon(F_{\text{sel}}^i \backslash f, \hat{a}_i, \hat{b}_i, \hat{x}_i, \hat{y}_i)$
$\qquad$ *// Remove the least discriminative bit*
12: $\qquad F_{\text{sel}}^i \leftarrow F_{\text{sel}}^i \backslash f_{\text{worst}}$
$\qquad$ *// Select the most discriminative bit*
13: $\qquad f_{\text{best}} \leftarrow \mathrm{argmin}_{f \in F_{\text{unsel}}^i}\, \epsilon(F_{\text{sel}}^i \cup f, \hat{a}_i, \hat{b}_i, \hat{x}_i, \hat{y}_i)$
$\qquad$ *// Update LHP if error is improved*
14: $\qquad e_{\text{best}} \leftarrow \epsilon(F_{\text{sel}}^i \cup f_{\text{best}}, \hat{a}_i, \hat{b}_i, \hat{x}_i, \hat{y}_i)$
15: $\qquad$ **if** $e_{\text{cur}}^i > e_{\text{best}}$ **then**
16: $\qquad\quad e_{\text{cur}}^i \leftarrow e_{\text{best}};\ F_{\text{sel}}^i \leftarrow F_{\text{sel}}^i \cup f_{\text{best}}$
17: $\qquad\quad F_{\text{unsel}}^i \leftarrow (F_{\text{unsel}}^i \backslash f_{\text{best}}) \cup f_{\text{worst}};\ c \leftarrow c+1$
$\qquad$ *// If not, terminate the search*
18: $\qquad$ **else**
19: $\qquad\quad F_{\text{sel}}^i \leftarrow F_{\text{sel}}^i \cup f_{\text{worst}};$ break
20: $\qquad$ **end if**
21: $\quad$ **end while**
22: **end for**
23: $s \leftarrow \mathrm{argmin}_i\, e_{\text{cur}}^i;\ \text{LHP} \leftarrow F_{\text{sel}}^s$

L bits (lines 11–12), (2) calculating errors for $NL - L$ unselected bits by considering a new pattern consisting of $L-1$ remaining bits and one of $NL - L$ unselected bits (line 13), and (3) replacing the removed bit by most discriminative bit if the training error decreases by replacement (lines 14–17).

Note that in the iterative procedure, the actual exchange is made only if the training error decreases. If not, the local search terminates with the current selection (lines 18–19). We also restrict the maximum number of replacements to $L/2$, to facilitate fast training while allowing an adequate number of replacements (line 10). The assumption behind the restriction in the number of replacements is that dominant feature type constructing L bits should be the original feature which made the patch to be chosen, and more than $L/2$ replacements can change the dominant feature type. After the search algorithm terminates on all $N+1$ patches, the feature with the smallest training error is selected to form the new LHP (line 23).

The number of error calculation in our algorithm has no

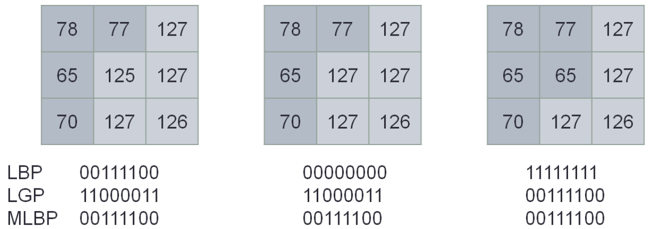

Fig. 1. Values of LBP, LGP, and MLBP for three similar patches.

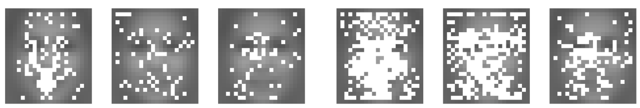

Fig. 2. Selected position of LBP, LGP, and MLBP up to third and fourth stages.

large difference compared to previous methods. At first, PN error calculations are needed to select the $N+1$ patches. At each iteration of the local search, L calculations are done for selecting the worst selected bit, and $NL-L$ for selecting the best unselected bit. The search is performed for at most $L/2$ steps. Thus, there are at most $PN+(N+1)\times(L+NL-L)\times L/2 = PN+L^2N(N+1)/2$ error calculations throughout our algorithm. In our example, this number evaluates to 4,935, which is an 8.4% increase compared to 4,551.

This small increase in training time can be justified by smaller empirical errors. We initialized the local search with the best original features and made replacements only when there is improvement on training error. This enables us to guarantee that generated LHP always has smaller or equal training error compared to word-wise methods. In AdaBoost algorithm, smaller training error leads to higher probability of training examples having large margins, which again leads to smaller upper bound on test error with high probability [17]. Thus, detectors trained using our algorithm are highly likely to have lower test error than word-wise methods.

2.2. Binary Local Pattern Features for LHP

In this paper, we generate the LHP by combining bits of LBP, LGP, MLBP, and their multi-block extensions. The MLBP is identical to the original LBP except that it uses the mean intensity value of neighboring pixels instead of the center pixel value. In fact, the MLBP does not involve center pixel values.

The MLBP is more robust to illumination and small location variation than both LBP and LGP. Consider the three image patches shown in Figure 1. The patches with different center values can be thought as parts of face edges captured from different facial images. Both the LBPs and LGPs obtained from the patches differ completely, while the MLBP is the same for all three images.

Table 1. Experiments and detection rates at 0.1 FPPI.

Features Used	Previous	LHP
LBP+LGP	0.780893	0.784955
LBP+LGP+MLBP	0.783214	0.791143
MB-LBP+MB-LGP	0.795011	0.800232
MB-LBP+MB-LGP+MB-MLBP	0.800812	0.804487

Figure 2 supports the above argument. Based on mean image of the training set, we plotted the positions of pixel-based LBP, LGP, and MLBP selected by a word-wise algorithm. The left three images show the selected positions of LBP, LGP, and MLBP, respectively, up to the third stage of the classifier. The right three show the positions up to the fourth stage. We can notice that positions of MLBP are concentrated around eyes and mouth and on the boundary of face while others are distributed more uniformly over the face.

As in the other features such as MB-LBP and MB-LGP, MLBP can be extended to a novel feature named Multi-block Mean Local Binary Patterns (MB-MLBP).

3. EXPERIMENTS

3.1. Experimental Settings

Faces from AFLW dataset [18] were utilized for positive examples. We picked faces that have estimated yaw angle from -30°to 30°. We computed the mean face from landmarks and performed similarity transform to align the faces in the same position and direction. As a result, 10,031 faces were obtained from the dataset. Among them, 7,531 faces were randomly chosen for training dataset, and the rest were used for validation. We also augmented the dataset with modification to the original images such as scaling, rotation, translation, and reflection. At last, the images were resized and cropped into 22×24 size gray-scale images. In this way, we obtained 75,310 training images and 25,000 validation images.

For negative examples, we gathered from the Internet 68,073 images that do not contain faces. Among them, we used 58,073 for training dataset pool and 10,000 for validation. We sampled 100,000 images with size 22×24 for negative training dataset, which was replaced with false positives of the cascade after each stage. For negative validation dataset we used 1,000,000 images drawn from validation dataset pool.

When training the stages of the cascade, we generated 25, 60, 120, and 400 weak classifiers for first, second, third, and later stages, respectively. We trained face detectors using different selections of features and algorithms. For those utilizing multi-block BLPFs, we limited block sizes to be either one of 1×1, 1×2, 2×1, and 2×2.

For evaluation of face detectors, ROC curves were obtained using FDDB dataset [19], in both discrete and continu-

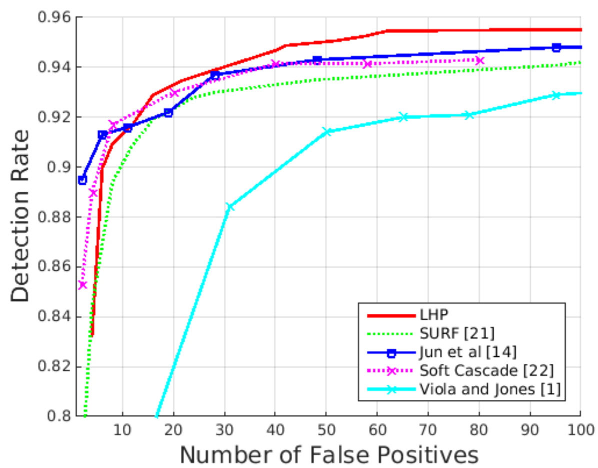

Fig. 3. Comparison results of LHP with the state-of-the-art algorithms on MIT+CMU dataset.

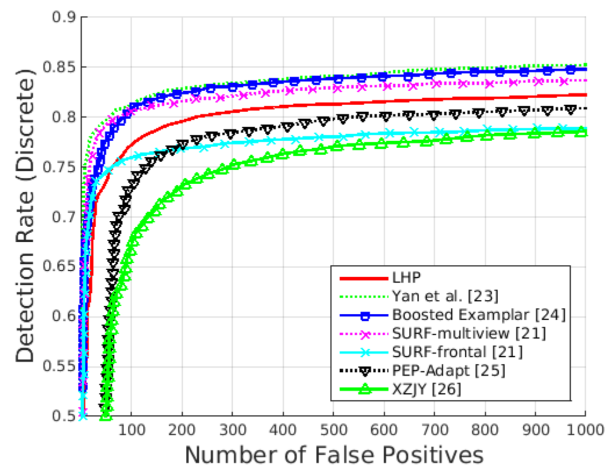

Fig. 4. Comparison results of LHP with the state-of-the-art algorithms on FDDB dataset.

ous scores. The detection rate at 0.1 false positives per image (FPPI) was obtained for quantitative comparison. Due to limited space, we only present results measured in discrete score. Continuous score results showed largely the same trends.

3.2. Experimental Results

Comparison with Single Features. We trained face detectors using single type of BLPFs; LBP, LGP and MLBP. Their detection rates on FDDB at 0.1 FPPI were 0.766196, 0.682073, and 0.774512, respectively. We believe that this result is due to the robustness MLBP in detecting edges, as previously mentioned.

Comparison with Multiple Features. We trained face detectors using different sets of features for both previous word-wise algorithm [14] and our bit-wise approach. The feature sets and detection rates at 0.1 FPPI are illustrated in Table 1. The first two rows show the results with pixel-based features, and the last two show multi-block ($a, b \in \{1, 2\}$) results. The comparison results reveal that LHP consistently shows higher detection rates than the word-wise algorithm for all feature combinations.

Performance on Test Benchmarks. To compare the LHP with the other state-of-the-art algorithms, we conducted the experiment on two widely used datasets. For LHP, the results from the configuration in the last row of Table 1 are reported.

The first benchmark was MIT+CMU dataset [20]. The MIT+CMU dataset contains 130 gray-scale images with 507 frontal faces. ROC curves of our detector and other classifiers [21, 14, 22, 1] are depicted in Figure 3. Our result was comparable to other state-of-the-art classifiers, and it even outperformed all of them above 0.125 FPPI range.

Comparison results of LHP with the state-of-the-art algorithms on FDDB dataset are shown in Figure 4. Even though some state-of-the-art results [23, 24, 21] performed slightly better than LHP, it should be emphasized that they considered several face detectors for multi-view face detection [21] or trained detectors on multi-view face examples [23, 24], while we only considered a single face detector for frontal faces. Our method showed the best performance among frontal detectors, and even outperformed other recent results using multi-view face detectors [25, 26].

4. CONCLUSION

In this paper, we have presented the proposed LHP and the local greedy algorithm for extracting LHP. The LHP is designed by a combination of the most discriminative bits from a candidate BLPF pool consisting of LBP, LGP, and MLBP. In the LHP generation process, the bits of LHP are initialized into one of the BLPFs and are iteratively updated by replacing the least discriminative bit with the most discriminative bit from all the candidate BLPFs. The proposed LHP is guaranteed to give smaller empirical error than any previously proposed combination of BLPFs considered in the pool, with only a small increase in computation. Experimental comparison on MIT+CMU and FDDB benchmark datasets for different sets of features consistently show that the LHP performs better than previously proposed features and combination methods under the AdaBoost face detection framework.

5. REFERENCES

[1] P. Viola and M. J. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[2] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," in *Proc. of Computational Learning Theory*. Springer, 1995, pp. 23–37.

[3] R. Lienhart and J. Maydt, "An extended set of Haar-like features for rapid object detection," in *Proc. of International Conference on Image Processing*. IEEE, 2002, vol. 1, pp. I–900.

[4] T. Mita, T. Kaneko, and O. Hori, "Joint Haar-like features for face detection," in *Proc. of International Conference on Computer Vision*. IEEE, 2005, vol. 2, pp. 1619–1626.

[5] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern Recognition*, vol. 29, no. 1, pp. 51–59, 1996.

[6] A. Hadid, M. Pietikainen, and T. Ahonen, "A discriminative feature space for detecting and recognizing faces," in *Proc. of Computer Vision and Pattern Recognition*. IEEE, 2004, vol. 2, pp. II–797.

[7] L. Zhang, R. Chu, S. Xiang, S. Liao, and S. Z. Li, "Face detection based on multi-block LBP representation," in *Advances in Biometrics*, pp. 11–18. Springer, 2007.

[8] H. Jin, Q. Liu, H. Lu, and X. Tong, "Face detection using improved LBP under bayesian framework," in *Proc. of Multi-Agent Security and Survivability*. IEEE, 2004, pp. 306–309.

[9] B. Jun and D. Kim, "Robust face detection using local gradient patterns and evidence accumulation," *Pattern Recognition*, vol. 45, no. 9, pp. 3304–3316, 2012.

[10] S. Zhou and J. Yin, "Robust face detection using multi-block local gradient patterns and extreme learning machine," in *Extreme Learning Machines 2013: Algorithms and Applications*, pp. 81–94. Springer, 2014.

[11] G. Bai, Y. Zhu, and Z. Ding, "A hierarchical face recognition method based on local binary pattern," *Proc. of Congr. Image Signal Process*, pp. 610–614, 2008.

[12] X. Wang, T. X. Han, and S. Yan, "An HOG-LBP human detector with partial occlusion handling," in *Proc. of International Conference on Computer Vision*. IEEE, 2009, pp. 32–39.

[13] H. Cevikalp and B. Triggs, "Efficient object detection using cascades of nearest convex model classifiers," in *Proc. of Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 3138–3145.

[14] B. Jun, I. Choi, and D. Kim, "Local transform features and hybridization for accurate face and human detection," *Pattern Analysis and Machine Intelligence*, vol. 35, no. 6, pp. 1423–1436, 2013.

[15] B. Froba and A. Ernst, "Face detection with the modified census transform," in *Proc. of Automatic Face and Gesture Recognition*. IEEE, 2004, pp. 91–96.

[16] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[17] R. E. Schapire, Y. Freund, P. Bartlett, and W. S. Lee, "Boosting the margin: A new explanation for the effectiveness of voting methods," *Annals of Satistics*, pp. 1651–1686, 1998.

[18] M. Kostinger, P. Wohlhart, P. M. Roth, and H. Bischof, "Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization," in *Proc. of International Conference on Computer Vision Workshops*. IEEE, 2011, pp. 2144–2151.

[19] V. Jain and E. G. Learned-Miller, "FDDB: A benchmark for face detection in unconstrained settings," *UMass Amherst Technical Report*, 2010.

[20] H. A. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," *Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 23–38, 1998.

[21] J. Li and Y. Zhang, "Learning SURF cascade for fast and accurate object detection," in *Proc. of Computer Vision and Pattern Recognition*. IEEE, 2013, pp. 3468–3475.

[22] L. Bourdev and J. Brandt, "Robust object detection via soft cascade," in *Proc. of Computer Vision and Pattern Recognition*. IEEE, 2005, vol. 2, pp. 236–243.

[23] J. Yan, Z. Lei, L. Wen, and S. Z. Li, "The fastest deformable part model for object detection," in *Proc. of Computer Vision and Pattern Recognition*. IEEE, 2014.

[24] H. Li, Z. Lin, J. Brandt, X. Shen, and G. Hua, "Efficient boosted exemplar-based face detection," in *Proc. of Computer Vision and Pattern Recognition*. IEEE, 2014.

[25] H. Li, G. Hua, Z. Lin, J. Brandt, and J. Yang, "Probabilistic elastic part model for unsupervised face detector adaptation," in *Proc. of International Conference on Computer Vision*. IEEE, 2013, pp. 793–800.

[26] X. Shen, Z. Lin, J. Brandt, and Y. Wu, "Detecting and aligning faces by image retrieval," in *Proc. of Computer Vision and Pattern Recognition*. IEEE, 2013, pp. 3460–3467.