A Bottom-up Framework for Robust Facial Feature Detection

Victor Erukhimov
Intel Corporation
30 Turgenev str., N.Novgorod, 603155, Russia
victor.eruhimov@intel.com

Kuang-chih Lee
Digital Persona
720 Bay Road, Redwood City, CA 94063 USA
leekc307@gmail.com

Abstract

Registration of facial features is a significant step towards a complete solution of the face recognition problem. We have built a general framework for detecting a set of individual facial features such as eyes, nose and lips using a bottom-up approach. A joint model of discriminative and generative learners is employed providing unprecedented results in terms of both detection rate and false positives rate. An Adaboost cascade learner is used to find candidates for facial features and a graphical model selects the most likely combination of features based on their individual likelihoods as well as relative positions and infers the missing components. We show good detection results on different large image datasets under challenging imaging conditions.

1. Introduction

An important step towards building a robust and efficient face recognition system is the localization of key facial features such as eyes, nose and mouth. However, human faces provide quite a few challenges to vision researchers to solve this problem including appearance (non-frontal views, occlusions), illumination changes (lack of/changes in color information, specular reflection) and complexity of the object (changes in facial expression, sunglasses and beard). For a complete literature review of human face process, please see [12].

Face and facial features detection has received a lot of attention in the context of face recognition [1]. The celebrated Adaboost/Gentleboost cascade learner [5,8] has been successfully applied to the face detection and eye detection problem. Beymer et al. [3] presents an appearance model-based eye detector. Others [2, 4] build a joint facial features model capable of detecting several features based on their individual appearances and relative position. [2,11,14] show that the weakness of local evidence or individual detectors can be compensated by shape and/or relative position of facial features.

The goal of this work is to build a robust feature detector that works under different illumination and pose changes using low resolution images as opposed to statistical shape models, such as Active Shape/Appearance Models (ASM/AAM) [2, 13], which require higher resolution input images to perform pixel-based registration.

2. Bottom-up framework

The focus of this work is on building a robust and fast detector of parts of human faces such as eyes, nose and lips. A good joint model of facial features should be able to solve very challenging computer vision problems such as face recognition. We approach the problem of detecting facial features by combining the discriminative and generative frameworks. The discriminative learners such as boosted cascade classifier [8] have gained much attention by showing extremely high recognition rates on a face detection problem. Such a learner can be trained to find specific features in images with very high detection rates. However it also shows considerably high false positive rates so the detection should be followed by a postprocessing algorithm that filters out bad samples.

Generative models in its turn have manifested themselves as efficient methods for describing structured objects [9]. [11] made use of this by building a detector for partially occluded faces with a markov random field imposed on facial features. [6,10] used discriminative learners as a part of model likelihood. We follow a similar path combining Adaboost detectors together with closed form pairwise potentials in a graphical model. The potentials defined on pairs of features play the role of filters sorting out the false positive examples. Individual detectors find samples of eyes, noses and lip corners in a face image and then a joint model that takes into account both individual features likelihoods and their relative positions to build the final combination of features.

However Adaboost cascade detector likelihood might be quite noisy and their detection rates on facial features may be much lower than detection rates for faces. One of the reasons for low performance is that features such as lip corners

978-1-4244-2154-1/08/$25.00 ©2008 IEEE

do not possess much structure and thus can't be detected reliably alone. Another reason is that often pixel patches representing individual features are quite small and do not possess enough data for detection. As a result of noisy cascade likelihood we can obtain pretty high likelihood values for a false combination of features. In order to address this issue we are using prior factors that zero in thick tails of gaussian distributions and filter out false combinations of features.

In order to combine all these techniques together we use a bottom-up approach. We start from individual detectors that find multiple hypotheses of facial features locations and scales. First we run a face detector on an input image. Inside all found faces we search for facial features. Then we iterate through all possible combinations of features that constitute a subset of 2 eyes, one nose and two (left and right) lip corners, use a graphical model to infer missing components. The final combination is selected by maximizing its likelihood.

3. Individual features detection

In order to detect individual facial features we train Adaboost cascade detectors. We use 5 detectors with 20 cascade stages each for faces, eyes, nose, left and right lip corners. All of them are trained on datasets with about 1000 positive samples and 100000 negative samples. Positive samples with resolution 30x30 were manually cropped from a consumer photo dataset, negative examples were automatically sampled from images without faces taken from personal collections. Individual classifiers use Haar features as predictors. We used OpenCV implementation of Adaboost cascade learning that closely follows Viola and Jones's work [8]. For each positive prediction returned by a detector we calculate confidence by summing up confidence values from all stages. Figure 1 shows ROC curves for individual detectors obtained by thresholding confidence values.

Detection of individual facial features provides quite a few challenges because the features, especially lip corners, do not possess much structure and have quite low resolution. If a feature scale is smaller than in the training dataset, it will not be detected. Adding small scale features to the training dataset will increase false positive rate. We solve the low resolution problem by applying cascade detectors to small faces zoomed up 2 and 4 times. In other words, if the face width or height is lower than a certain threshold of 200 pixels and the detector doesn't find any instances of features, then we zoom the face image 2 times up and run the detection again. If still no features are detected, we zoom the face image again 2 times up and run the detection.

The lack of structure possessed by facial features is a serious problem causing many false positive detections even when we run detection only inside face rectangles. The following two sections deal with filtering false positives and selecting the final combination of facial features.

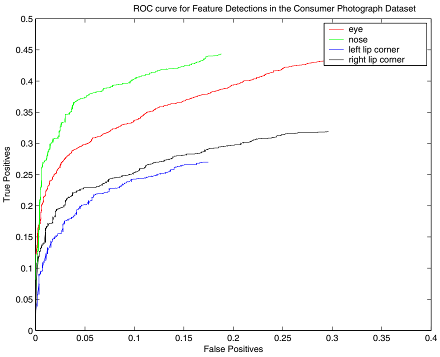


Figure 1. ROC Curves of Individual Detectors. All individual detectors perform much worse than a general adaboost face detector because each facial component contains less structure than a complete face.

4. Combination of features

We will denote a facial feature instance with f_i where i indexes feature types $\{nose, lefteye, righteye, leftlip, rightlip\}$ from 0 to 4 correspondingly. The goal of the algorithm described in this section is to find the most likely combination of facial features $\{f_i^j\}$ detected by Adaboost cascades. Here $j = 1..k_i$ is the index of the feature of a particular type i. A "combination of features" here means a set of features $F = \{f_0^{j_0}, f_1^{j_1}, f_2^{j_2}, f_3^{j_3}, f_4^{j_4}\}$ or its subset $\tilde{F} \subset F$. Each feature f_i^j is given by a rectangle characterized by its center coordinates (x, y) as well as width w and height h normalized by width and height of the face image so that $0 \leq x \leq 1$ and $0 \leq y \leq 1$.

We introduce a graphical model to infer missing features in each combination and compute the likelihood of the combination that is used as the score. The graphical model uses both individual feature likelihoods and pairwise potentials that account for relative positions of features.

4.1. Generative Model

We build a Bayesian Network with the likelihood given in the form

$$L = \left(\prod_{(i,j) \in E} \Psi_{ij}(f_i, f_j) \right) \Psi_{nose}(f_{nose}) \left(\prod_i \Phi_i(f_i, I) \right). \tag{1}$$

Here i, j are indices of facial features. Each feature f_i has a corresponding node in the Bayesian network and is described by its coordinates (x_i, y_i) normalized to width and height of the face image correspondingly. The graph structure of the model is shown in Figure 2. E is a set of edges

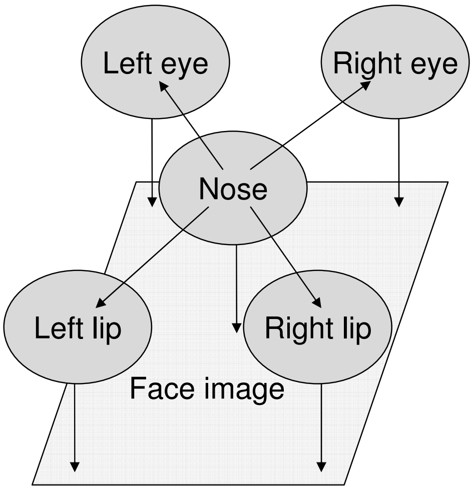


Figure 2. Bayesian network model of a face.

in a Bayesian network, and I denotes the image of the face. The likelihood in Eq.(1) consists of two parts. Potentials encoded by Ψ_{ij} are conditional probability distributions characterizing the relative positions of features. Ψ_{nose} describes the expected position of the nose inside the face rectangle. Both Ψ_{ij} and Ψ_{nose} are given in closed form. Φ_i is the likelihood of the input face image given the coordinates of the facial feature i. It is calculated using the Adaboost cascade.

In order to calculate $\Phi_i(f_i, I)$ we select the image rectangle with the center in x_i, y_i and width and height equal to 20% from the width and height of the face image. This rectangle is an input to the Adaboost cascade corresponding to the feature i. The rectangle is moved through all cascade steps as opposed to regular detection when some (in fact the most of) rectangles are filtered out before they reach the final step. Then we obtain the resulting score $s_i(f_i, I)$ from the cascade and

$$\Phi(f_i, I) = \alpha s_i(f_i, I). \tag{2}$$

We used values of α in the region of $0.01 - 0.1$. We will discuss the choice of α below.

The pairwise potential Ψ_{ij} is a conditional probability distribution. One can think of the first and second products in (1) as of a Bayesian network where all distributions are conditioned on the position of nose. Ψ_{ij} is a conditional Gaussian distribution multiplied by the prior factor:

$$\begin{aligned}&\Psi(f_i, f_{nose}) = P(f_i|f_{nose}) = \\ &= N\left(\mathbf{p}_i, \mathbf{m}_i + \hat{R}_i \mathbf{p}_{nose}, \hat{C}_i\right) F_i\left(f_i, f_{nose}\right).\end{aligned} \tag{3}$$

Here $\mathbf{p_i} = \{x_i, y_i\}$ is a vector characterizing feature position in the image, $N\left(\mathbf{p}, \mathbf{m}, \hat{C}\right)$ is a normal distribution of $\mathbf{p}$ with mean $\mathbf{m}$ and covariance $\hat{C}$, $\hat{R}_i$ is a regression 2x2 matrix. In addition we have a Gaussian distribution on nose $\Psi_{nose} = N(\mathbf{p}_{nose}, \mathbf{m}_{nose}, \hat{C}_{nose})$. Prior factors F_i have a form of a step function and are introduced to cut off long tails of gaussian distributions using prior information about relative positions of facial features:

$$\begin{cases} F_{leye} = 1(x_{nose} - x_{leye})1(y_{nose} - y_{leye}) \\ F_{reye} = 1(x_{reye} - x_{nose})1(y_{nose} - y_{reye}) \\ F_{llip} = 1(x_{nose} - x_{llip})1(y_{llip} - y_{nose}) \\ F_{rlip} = 1(x_{rlip} - x_{nose})1(y_{rlip} - y_{nose}) \end{cases} \tag{4}$$

Here $1(x) = 1$ if $x \geq 0$, otherwise $1(x) = 0$. We use abbreviation ”llip” instead of ”left lip”, ”rlip” instead of ”right lip”, ”leye” instead of ”left eye” and ”reye” instead of ”right eye”. The vertical axis is directed from eyes towards mouth.

Representation of potentials in the form of conditional probability distribution allows us to learn them directly from data. Obviously prior factors in Eq.(4) are equal to 1 on training samples so the problem of learning potentials Eq.(3) from data is equivalent to learning conditional Gaussian distributions.

4.2. Inferring the Most Probable Feature Combination.

The graphical model described in the previous section contains unobserved continuous variables with potentials defined as adaboost confidence funtcions. The most general inference algorithm capable of handling any network/potentials type, is Gibbs sampling. Also, nonparametric belief propagation [11] was used to infer unobserved continuous variables. However both approaches, if applied in a straightforward manner, are computationally expensive and do not guarantee convergence. In order to make the problem tractable, we significantly reduce the feature combination search space. We consider only combinations where 3 or more features have been detected with the adaboost cascade, i.e., they were not filtered by any cascade stage. If nose is not detected, we require other 4 features to be detected. Note that individual detectors can return positive likelihoods for image regions that are filtered out by one of cascade stages. However, given relatively high detection rate of individual classifiers (see Fig.1), this requirement is met most of the time – at least 3 features from the correct combination are detected by the detectors. Allowing more than 2 missing features significantly increases false positive rate given imprecise detector likelihood function.

Inference algorithm now is tractable and straightforward. We run individual detectors, iterate through all possible combinations of features, filter out combinations that do not meet the requirements above, infer missing features and select the combination with the highest likelihood. Inference of missing features is also straightforward. All the nodes from the markov blanket of an unobserved feature are observed for any combination that was not filtered by the requirements.

The likelihood of the unobserved feature is as follows:

$$P(\mathbf{p}_i) \propto \hat{\Psi}_i(\mathbf{p}_i)\Phi(\mathbf{p}_i). \tag{5}$$

$\hat{\Psi}_i(\mathbf{p}_i)$ is a gaussian distribution obtained from multiplying Ψ_{ij} and Ψ_{nose} with observed feature coordinates. The average location of feature $\langle \mathbf{p} \rangle$ is equal to

$$\langle \mathbf{p} \rangle = \frac{\int \mathbf{p}\hat{\Psi}_i(\mathbf{p}_i)\Phi(\mathbf{p}_i)d\mathbf{p}}{\int \hat{\Psi}_i(\mathbf{p}_i)\Phi(\mathbf{p}_i)d\mathbf{p}}. \tag{6}$$

We calculate both integrals using MCMC. We generate G samples $\{p_i^{(g)}\}$ from Ψ_i and for arbitrary function $W(\mathbf{p})$

$$\int W(\mathbf{p})\hat{\Psi}_i(\mathbf{p})d\mathbf{p} = \frac{1}{G}\sum_g W(\mathbf{p_g}). \tag{7}$$

Since there are only two dimensions MCMC converges quickly and we need only about 100 samples.

When all missing components are inferred we calculate loglikelihood Eq.(1) of the combination. The final combination corresponds to the maximum loglikelihood. Parameter α establishes the balance between the importance of individual cascade likelihoods and relative positions likelihood. In the most of the experiments we used the value of $\alpha = 0.01$.

Here is the outline of the overall facial features detection algorithm.

4.3. Facial features detection algorithm

1. Run face detector on the input image
2. For each detected face
    (a) Run eyes, nose and lips detectors on the facial image cropped from the input image
    (b) For each combination of facial features
        i. Check if the combination meets the requirements. If it doesn't, go to the next combination.
        ii. Infer all missing components using a Bayesian network Eq.(1).
        iii. Calculate the loglikelihood of the 5-feature combination using loglikelihood Eq.(1).
3. Choose the combination with the maximum loglikelihood.

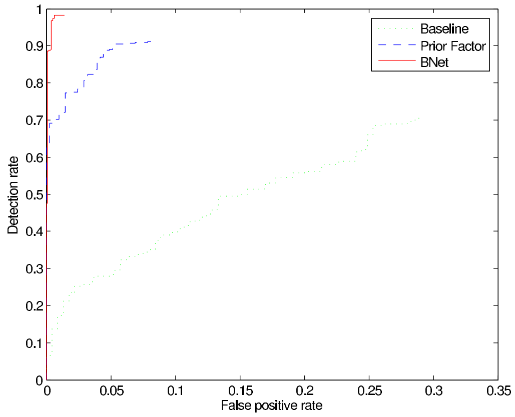


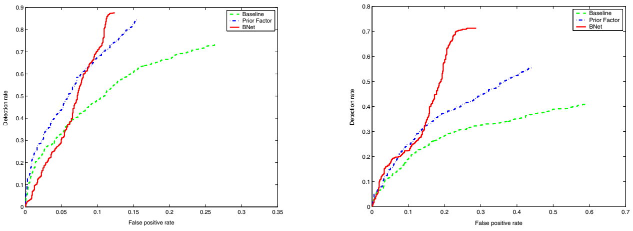
Figure 3. ROC Curve of our consumer photograph testset. The red curve shows the performance of our algorithm. The green curve shows the performance of the baseline algorithm that simply combines the individual detection results. The blue curve shows the performance of the baseline results filtered by the prior knowledge of the relative positions of the facial features.

5. Experimental results

To evaluate the robustness and efficacy of our approach, we have tested the facial component detection technique on the standard Feret dataset [7] and a consumer photograph testset. The Feret dataset contains several views of 1208 individuals. Since our current face detector currently can detect human faces in the range of viewing angle ±67.5 degree, we only reported our results from the detected face images in frontal, quarter profile, and half profile views. Another testset of consumer photographs contains more than 15000 faces of 200 individuals. The consumer photographs in this testset were taken by several different digital cameras under completely uncontrolled environments.

Figure 5 displays the results of facial component detection in some difficult and representative images. The results demonstrate that our system is capable of delivering precise location of facial components under many challenging conditions involving low resolution (small detected faces), varying pose, illumination, shadowing, expression, wearing sunglasses, and so on.

Figure 3 shows the receiver operating characteristic curve (ROC) obtained by thresholding likelihood scores for consumer photo dataset, and Figure 4 – for Feret dataset. The detection rate is defined as the number of of the correct detections divided by the number of the detected faces. The false positive rate is defined as the number of the incorrect detections divided by the number of the detection attempts. A detection is considered to be correct if the distance between detected feature location and the ground truth position is smaller than some threshold. This threshold is currently picked by the relative length that is 20% of face rectangle size along each axis. *Bnet* corresponds to the likelihood given by Eq.(1), *Prior Factor* takes into account only individual detector likelihoods and prior factor, and *Baseline* uses the likelihood equal to the product of individual detector likelihoods. By comparing three curves, we can estimate the importance of each factor in the likelihood function Eq.(1). The detection rate in the *Baseline* is lower than 70% because lots of the facial components are mis-detected by the individual component detectors. Our Bayesian network approach can effectively infer the locations of mis-detected facial components and thereby it helps to improve the detection rate on the consumer photograph dataset (Fig.3) to more than 97% with less than 1% false positive rate.



(a) (b)

Figure 4. ROC curve for Feret frontal (a) and quarter left (b) photograph testsets. The red curve shows the performance of our algorithm. The green curve shows the performance of the baseline algorithm that simply combines the individual detection results.

We used OpenCV and OpenPNL to implement our system. Our system can detect facial components at 0.5-2 FPS comfortably with a 1024 × 768 input image without any code optimization on a laptop with PM 1.7GHz. The most the computational time is consumed by individual Adaboost detectors. We can speculate that bringing the system to realtime is a matter of choosing the right scales and locations for Adaboost to search in.

6. Conclusion

We have built a robust facial feature detection system. Individual Adaboost detectors have been put in a context of a Bayesian network that combined cascade likelihood together with relative position information. This combination provided unprecedent results both in terms of detection rate and false positives rate. We have shown that the algorithm is working in very challenging conditions such as low resolution and changes in pose up to half profile. At the same time because of the algorithm simplicity we stay within reasonable bounds of computational time. Future steps would be to generalize the algorithm for profile views and use the facial features detection algorithm in the face recognition system.

Acknowledgments

We want to thank Alexander Bovyrin for his enthusiasm and help with Adaboost detectors.

References

[1] R. Brunelli and T. Poggio. Face recognition: Features versus templates. *IEEE Trans. Pattern Anal. Machine Intell.*, 15:1042–1052, 1993.

[2] T. Cootes and C. Taylor. Statistical models of appearance for computer vision, 1999.

[3] D.Beymer and M.Flickner. Eye gaze tracking using an active stereo head. In *IEEE Conf. on Computer Vision and Pattern Recognition*, volume 2, pages 451–458, 2003.

[4] B. Heisele, T. Serre, M. Pontil, T. Vetter, and T. Poggio. Categorization by learning and combining object parts. 2:1239–1245, 2002.

[5] I.Fasel, B.Fortenberry, and J.Movellan. A generative framework for real time object detection and classification. In *Computer Vision and Image Understanding*, 2005.

[6] J.Lafferty, A.McCallum, and F.Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Intl. Conference on Machine Learning*, 2001.

[7] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss. The FERET evaluation methodology for face-recognition algorithms. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(10):1090–1034, 2000.

[8] P.Viola and M.Jones. Robust real-time object detection. In *Int'l. J. Computer Vision*, 2004.

[9] R.G.Cowell, A.P.Dawid, S.L.Lauritzen, and D.J.Spiegelhalter. *Probabilistic Networks and Expert Systems*. Springer, 1999.

[10] S.Kumar and M.Hebert. Discriminative random fields: A discriminative framework for contextual interaction in classification. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2003.

[11] E. Sudderth, A. Ihler, W. Freeman, and A. Willsky. Nonparametric belief propagation, 2002.

[12] S.Z.Li and A.K.Jain. *Handbook of Face Recognition*. Springer, 2004.

[13] T.Cootes, G.Edwards, and C.Taylor. Active appearance models. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 23(6):681–685, 2001.

[14] Z.Peng, L.Tao, G.Xu, and H.Zhang. Detecting facial features on images with multiple faces. In *Third International Conference on Advances in Multimodal Interfaces*, pages 191–198, 2000.

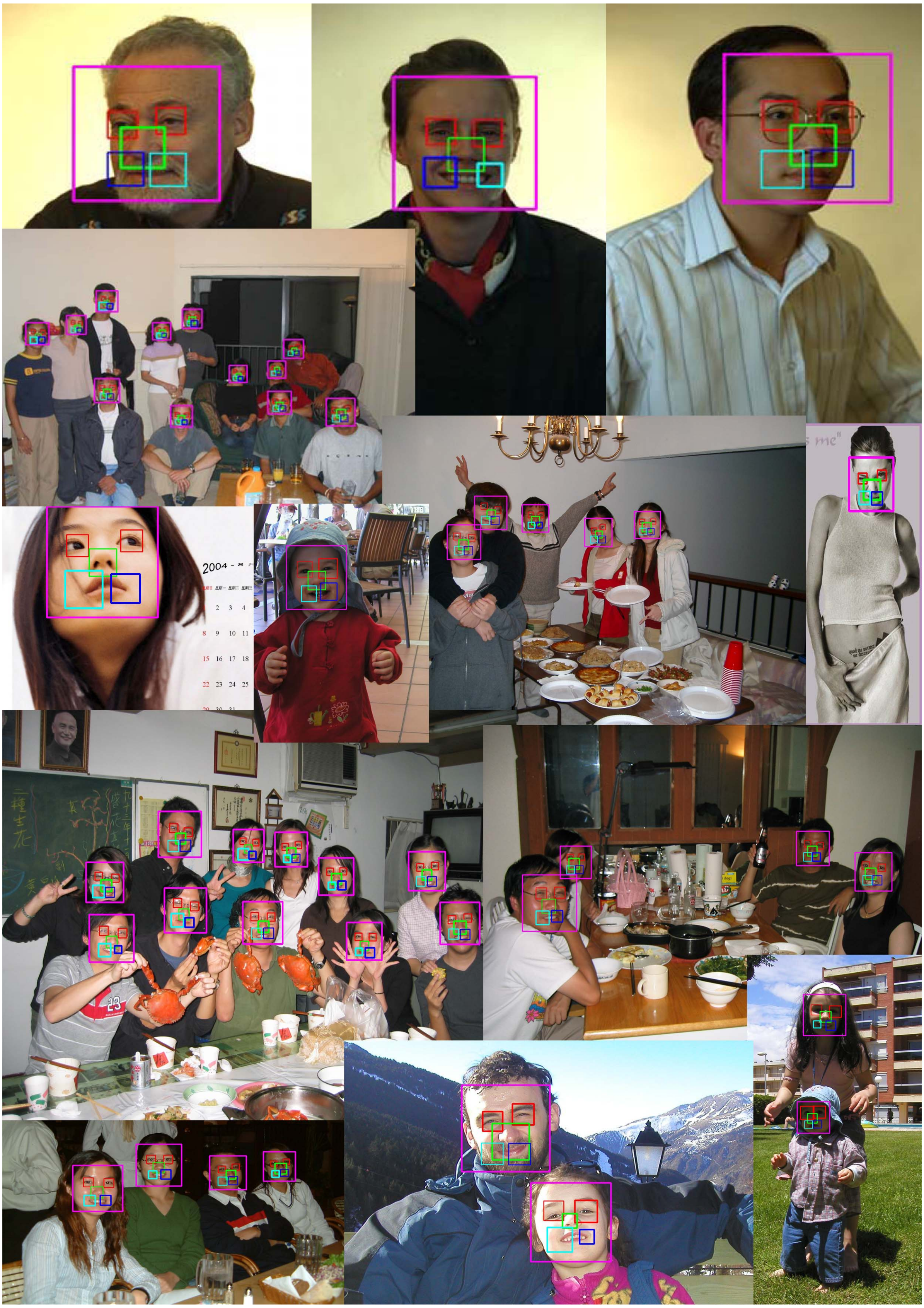


Figure 5. Visual output of our facial component detector from Feret and our consumer photograph testset. Each face is marked by a pink rectangle. Eyes, nose top, left lip corner, and right lip corner inside each face are marked by red, green, light blue and dark blue rectangles respectively. The challenges for the detection include varying pose, outdoor illumination (directional lights and shadowing, expression, wearing sunglasses, and low resolution (small) faces.