Head-pose estimation in-the-wild using a Random Forest

Roberto Valle[1], José Miguel Buenaposada[2], Antonio Valdés[3], and Luis Baumela[1]

[1] Univ. Politécnica Madrid, Spain. {rvalle,lbaumela}@fi.upm.es
[2] Univ. Rey Juan Carlos, Spain. josemiguel.buenaposada@urjc.es
[3] Univ. Complutense Madrid, Spain. avaldes@ucm.es

Abstract. Human head-pose estimation has attracted a lot of interest because it is the first step of most face analysis tasks. However, many of the existing approaches address this problem in laboratory conditions. In this paper, we present a real-time algorithm that estimates the head-pose from unrestricted 2D gray-scale images. We propose a classification scheme, based on a Random Forest, where patches extracted randomly from the image cast votes for the corresponding discrete head-pose angle. In the experiments, the algorithm performs similar and better than the state-of-the-art in controlled and in-the-wild databases respectively.

Keywords: Head-pose estimation, random forest, real-time, in-the-wild

1 Introduction

Head-pose estimation is an essential preprocessing step for accurately inferring many facial attributes, such as age, gender, race, identity or facial expression. Additionally, head-pose is also used in other contexts, such as identifying social interactions [9, 14], focus of attention [1, 17], or gaze estimation [19].

By estimating the head-pose, we mean predicting the relative orientation between the viewer and the target head. It is usually parametrized by the head's yaw, pitch and roll angles [15]. Yaw and pitch rotations are the most informative for interpersonal communication and cause the largest appearance changes in the expressive parts of the face. For this reason, most approaches only estimate one of them or both. In this paper, we consider the problem of inferring the discretized yaw angle from a face image.

Facial pose estimation methods may be broadly organized into four groups. *Subspace* approaches assume that facial appearance changes, originated by pose variations, lie on a low-dimensional manifold embedded in a high-dimensional feature space [2, 3, 18]. Approaches based on *flexible models* fit a face deformable model and estimate pose from the location of a set of landmarks [21]. Methods based on *classification* discretize the range of poses in a group of classes and solve the problem using a classification algorithm [20]. *Regression* approaches estimate a continuous function that maps facial features to the space of poses [12, 13, 10, 16, 8].

Depending on the input data, methods may also be grouped into those using 2D images [2, 3, 18, 21, 20, 12, 13, 10, 16] or 3D range data [8]. Range data images provide direct shape information which facilitates head-pose estimation. On the other hand, RGB or gray-scale 2D images are more ubiquitous, but make pose estimation harder because of the lack of texture in some facial regions.

Traditionally, pose estimation algorithms have been evaluated in laboratory conditions, using databases such as Pointing-04 or CMU Multi-PIE [11, 16, 12, 13, 10]. Nowadays, the interest has shifted towards evaluations involving more realistic and challenging situations using databases such as AFLW or AFW with images acquired "in-the-wild" [21, 18].

In this paper, we present a classification approach to estimate head-pose in-the-wild, based from 2D images, on a regression forest. Our algorithm obtains discrete orientation data from the predictions of a Random Forest. It achieves results close to the state-of-the-art in laboratory conditions evaluated using the Pointing-04 database, and better than Zhu *et al.* [21] and Sundararajan *et al.* [18] on the challenging AFLW and AFW databases. Additionally, it performs in real-time at 80 FPS.

2 Head-pose classification based on a Random Forest

We propose Random Forest in order to obtain a discrete head-pose estimation. The Random Forest is a well-known machine learning algorithm formed by an ensemble of T decision trees, whose prediction is determined by combining the outputs from all the trees. This technique has been successfully used in a variety of computer vision problems, such as classification, regression and probability density estimation [5]. Moreover, it is a widely used machine learning algorithm because it may be trained with a moderately low amount of information and the resultant ensemble can perform in real-time.

2.1 Patch-based channel features

We use visual features as Dantone *et al.* [6]:

- From each training image, we randomly choose a set of square patches, $\mathcal{P}_i = \{(\mathcal{I}_i, h_i)\}$, where h_i is the pose and $\mathcal{I}_i$ is the appearance of the patch, described by a set of channels $\mathcal{I}_i = (\mathtt{I}^1, \ldots, \mathtt{I}^k)$ [7]. $\mathtt{I}^\alpha$ are the values of channel α in image $\mathtt{I}$. The channels are gray-scale values, Sobel borders and 35 Gabor filters, some of which are shown in Fig. 1.
- Our features are the difference between the average values in two rectangles, R_1 and R_2, in a channel. We describe each of them with the pair of rectangle coordinates within the patch boundaries in channel α, $\theta = (R_1, R_2, \alpha)$. So, given patch p and parameters θ, the feature value is:

$$f(p, \theta) = \frac{1}{|R_1|} \sum_{\mathbf{q} \in R_1} \mathtt{I}^\alpha(\mathbf{q}) - \frac{1}{|R_2|} \sum_{\mathbf{q} \in R_2} \mathtt{I}^\alpha(\mathbf{q}), \tag{1}$$

  where $\mathbf{q} \in \mathbb{R}^2$ are pixel coordinates.

The splitting nodes (weak learners) of the decision trees in the Random Forest use these features to select the best channels and face subregions to regress the head-pose. Since we address the problem of head-pose in-the-wild, this kind of local feature will be more robust than an holistic approach.

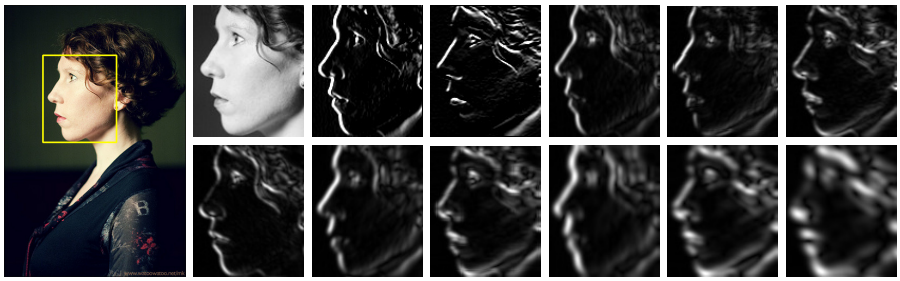

Fig. 1: Sample channels used in our approach.

2.2 Training regression forest

Following the standard Random Forest approach [4], we train each decision tree using a randomly selected set of patches from a random subset of the training faces. We optimize each weak learner by selecting the $\theta = (R_1, R_2, \alpha)$, from a random pool of candidates $\phi = (\theta, \tau)$, that maximizes the information gain

$$IG(\phi) = \mathcal{H}(\mathcal{P}) - \sum_{S \in \{L,R\}} \frac{|\mathcal{P}_S(\phi)|}{|\mathcal{P}|} \mathcal{H}(\mathcal{P}_S(\phi)), \tag{2}$$

where τ details the threshold over the feature value, $\mathcal{P}_L(\phi) = \{\mathcal{P} | f(P, \theta) < \tau\}$, $\mathcal{P}_R(\phi) = \mathcal{P} \setminus \mathcal{P}_L(\phi)$, and $\mathcal{H}(\mathcal{P}_S(\phi))$ is the class uncertainty measure. In our case, $\mathcal{H}(\mathcal{P}) = log(\sigma\sqrt{2\pi e})$ is the Gaussian differential entropy of the continuous patch labels.

2.3 Pose estimation

Once we have trained the Random Forest for image patches, and given an input image $\mathtt{I}$, we estimate the head-pose orientation as follows:

1. Detect face bounding box in $\mathtt{I}$.
2. Resize bounding box to $W \times H$ pixels, denoted $\mathtt{I}_r$.
3. Compute α channels from $\mathtt{I}_r$.
4. Extract from $\mathtt{I}_r$ patches of size $N \times N$ with a stride of S pixels, denoted $\mathcal{P}$, the set of input patches.
5. For each patch $p_i \in \mathcal{P}$:

5.1. For each tree t_j in the Forest (see Fig. 2):
  5.1.1. Input p_i to t_j.
  5.1.2. The leaf node of t_j reached by p_i provides a discrete distribution of the face orientation, $p(yaw|p_i, t_j)$.
5.2. Compute the patch face pose distribution, $p(yaw|p_i) = \sum_j p(yaw|p_i, t_j)$.
6. Compute the final face pose distribution, $p(yaw|\mathtt{I}_r) = \sum_i p(yaw|p_i)$.
7. The final classification is the most probable discrete orientation in $p(yaw|\mathtt{I}_r)$ (see Fig. 3).

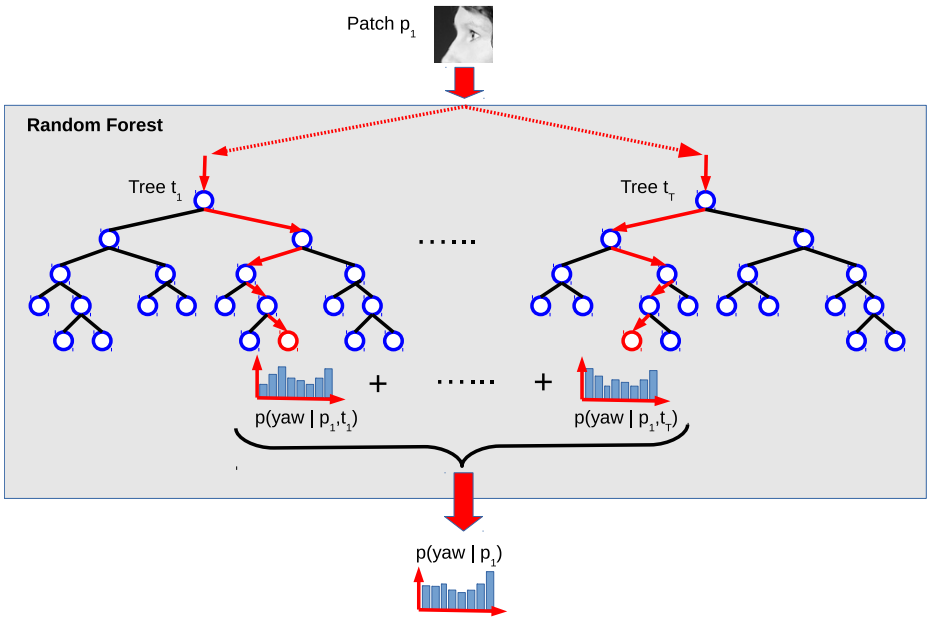

Fig. 2: Random Forest classification of an individual image patch. The result is a discrete probability distribution of the yaw angle.

3 Experiments

In this section, we compare different state-of-the-art approaches with our method on both controlled and in-the-wild databases.

3.1 Databases

To evaluate and train our algorithm we need a set of images labelled with ground truth head-pose data. The accurate estimation of this data is not easy. For this reason most public face databases do not provide it. We have chosen Pointing-04

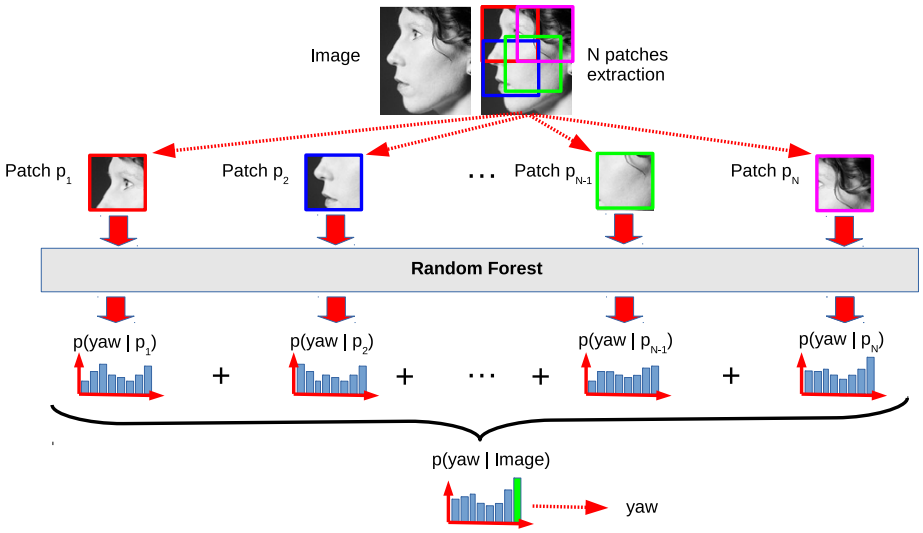

Fig. 3: Estimation of head-pose orientation using different face image patches.

to compare our approach with the traditional algorithms evaluated in laboratory conditions. Another popular database of this type is Multi-PIE, that we do not use because it shows saturated results. We also employ AFLW and AFW. These databases were acquired for face detection in an unrestricted setting. Their faces present extreme poses, partial occlusions, etc.

- **Pointing-04**. Included as a part of the Pointing 2004 Workshop on Visual Observation of Deictic Gestures to allow an uniform evaluation of head-pose estimation. The database contains 2790 images of 15 subjects captured in a controlled scenario spanning discrete yaw and pitch poses from -90° to 90° with 15° interval. It provides a coarse ground truth obtained by asking subjects to direct their heads toward a set of markers placed around them in a room.
- **AFLW**. Provides an extensive collection of 25993 in-the-wild faces, with 21 facial landmarks annotated depending on their visibility. To the best of our knowledge, this is the largest public database providing face pose labels in an uncontrolled scenario. AFLW uses manually annotated landmarks positions to approximate face bounding box and coarse yaw, pitch and roll angles by fitting a mean 3D face using the POSIT algorithm.
- **AFW**. Consist of 250 images with 468 challenging in-the-wild faces. It is commonly used as a test set because of the low number of images. It provides large variations in scales and discrete poses annotated for angular yaw from -90° to 90° with 15° interval plus facial bounding box.

3.2 Evaluation

As in related work, we employ the mean absolute error (MAE) metric in order to evaluate and compare the algorithms. In addition, we also display results using a cumulative error distribution, representing the percentage of test faces with absolute error lower than some degrees of tolerance. Finally, since our approach provides discrete classification results, we also show the confusion matrix. In our implementation we discretize angular yaw in steps of 15° {-90°, -75°, -60°, -45°, -30°, -15°, 0°, +15°, +30°, +45°, +60°, +75°, +90°}, which let us to compare our results with other state-of-the-art approaches.

We follow a 90%/10% hold-out evaluation scheme to deduce the performance on Pointing-04 and AFLW databases. Given the small size of AFW, we only use it for testing. In this case we train the algorithm with a balanced data set from AFLW with 700 images per class.

3.3 Configuration of Random Forest parameters

We use the same configuration of parameters for our algorithm in all experiments. We resize the face bounding box provided by each database to 105×125 pixels and assume the head to be the prominent object in the rectangle. The forest has $T = 20$ trees each of them trained from a randomly selected set of images equally distributed by yaw angle (9100 for AFLW and 2457 for Pointing-04). From each bounding box we randomly extract 20 patches of 61×61 pixels. The performance of the algorithm is quite sensitive to this parameter. A smaller patch would not capture enough information to predict the poses. On the other hand, a larger patch would provide an implementation more sensitive to occlusions.

Tree growing stops when the depth reaches 15, or if there are less than 20 patches in a leaf. We train each tree node by selecting the best parameters from a pool of $\phi = 50000$ samples obtained from $\theta = 2000$ different combinations of $[\alpha, \text{R1}, \text{R2}]$ and $\tau = 25$ thresholds. The maximum random size of the subpatches defining the asymmetric areas R1 and R2 is set to be lower than a 75% of the patch size.

For efficiency and accuracy reasons, we also filter out leaves with a maximum variance threshold set to 400. This limits the impact in the final prediction of non-informative leaves. A pair of crucial test-time parameters are the number of trees in the forest and the stride controlling the sampling of patches. We process only 1 out of 10 possible patches. Test values can be empirically tuned to find the desired trade-off between accuracy and temporal efficiency of the estimation process, making the algorithm adaptive to the constraints of different applications.

In Fig. 4 we show a set of sample results. Green and blue lines represent the estimated angular yaw and the ground truth respectively.

3.4 Results

In the first experiment, we evaluate the performance of the proposed algorithm in Pointing-04, a database acquired in laboratory conditions. The results in Table 1

Fig. 4: Sample results for Pointing-04 (top), AFLW (middle) and AFW (bottom) databases. Green and blue lines indicate respectively pose estimation and ground truth yaw angle.

show that our proposal has a MAE close to the state-of-the-art in this database. Also, our classification accuracy, i.e. specific discrete head-pose angle properly labelled with the correct class, is behind the best. Nevertheless, in a 93.54% of the cases the error in our approach is lower than 15°. All three approaches with better results use holistic HOG-based face features [12, 13, 10]. In this constrained context, this global feature is slightly more informative for estimating face pose than the set of local patches that we use in our approach. However, as we show in the sequel, local representations will have better performance in unconstrained situations.

Method	Pointing-04	
	MAE	Accuracy (0°)
Stiefelhagen [16]	9.5°	52.0%
Haj [12]	6.56°	67.36%
Hara [13]	5.29°	-
Geng [10]	4.24°	73.30%
Our method	**7.84°**	**55.19%**

Table 1: Head-pose estimation results in a constrained database.

In the second experiment, we consider the unconstrained situations appearing in real-world situations. In Table 2 we present the results for AFLW and AFW databases. Here our approach achieves the best performance, both in terms of

MAE and classification accuracy with an error less than 15°. These results proves the powerful representational ability of local features with a nonlinear regression algorithm. This approach can deal with challenging in-the-wild conditions, such as the presence of occlusions, illumination changes or facial expressions.

Our algorithm also outperforms its competitors in terms of computational requirements. It submits a frame rate of 80 FPS (12 ms per image) on an Intel Core i7 CPU processor at 3.60GHz with 8 cores multi-threaded, 300 times faster than the second best approach, Zhu *et al.* [21]. Sundararajan *et al.* [18] provides similar runtime performance, but with a clearly worse head-pose accuracy.

Method	AFLW		AFW	
	MAE	Accuracy ($\leq$15°)	MAE	Accuracy ($\leq$15°)
Haj [12]	-	-	-	78.7%
Zhu [21]	-	-	-	81.0%
Sundararajan [18]	17.48°	58.05%	17.20°	58.33%
Our method	**12.26°**	**72.57%**	**12.50°**	**83.54%**

Table 2: Head-pose estimation results for in-the-wild databases.

Finally, in Fig. 5 and Fig. 6 we compare the cumulative head-pose error of our approach against Sundararajan *et al.* [18]. We also present the confusion matrix of the yaw classification label. The colour intensity in it represent the percentage of success for each class (see bar on the right side). As can be seen, most incorrect predictions are adjacent to the proper ground truth angle. The largest errors are between ±90° and ±45° classes. This is reasonable, given the lower appearance variation between them.

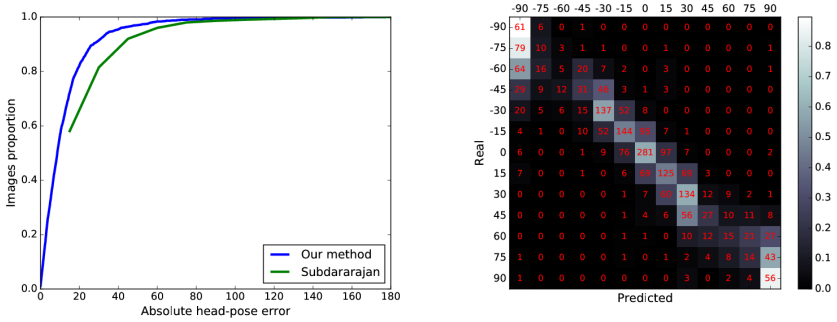

Fig. 5: Cumulative head-pose error distribution and confusion matrix for AFLW.

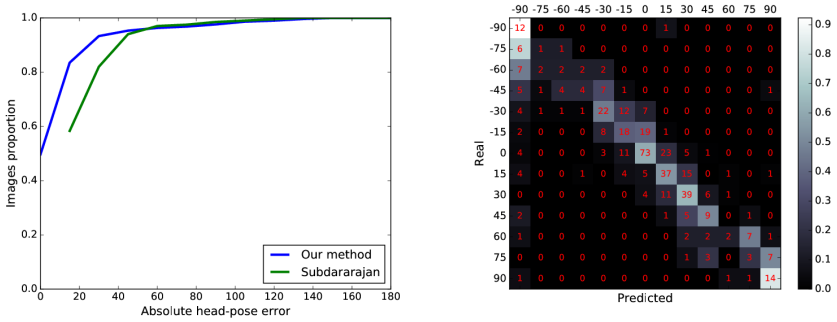

Fig. 6: Cumulative head-pose error distribution and confusion matrix for AFW.

We developed our own open-source code of the previously described Random Forest classifier algorithm. All implementations could be made publicly available after submission.

4 Conclusions

In this paper, we have presented an algorithm to estimate head-pose yaw angle in unrestricted situations. To this end, we learn a regression forest from random face patches. We obtain the optimal splitting in each tree node according to the entropy computed from continuous yaw angle. The experimental evaluation shows that our algorithm performs best in the tests on unrestricted databases, proving the superior robustness of this local representation with the presence of occlusions, illumination changes, motion blur and exaggerated facial expressions.

Acknowledgements: The authors gratefully acknowledge funding from the Spanish Ministry of Economy and Competitiveness under project SPACES-UPM (TIN2013-47630-C2-2R).

References

1. Ba, S.O., Odobez, J.M.: Multiperson visual focus of attention from head pose and meeting contextual cues. IEEE Trans. on Pattern Analysis and Machine Intelligence 33(1), 101–116 (2011)
2. Balasubramanian, V., Ye, J., Panchanathan, S.: Biased manifold embedding: A framework for person-independent head pose estimation (2007)
3. BenAbdelkader, C.: Robust head pose estimation using supervised manifold learning. In: Proc. European Conference on Computer Vision (ECCV) (2010)
4. Breiman, L.: Random forests. Machine Learning 45(1), 5–32 (2001)
5. Criminisi, A., Shotton, J., Konukoglu, E.: Decision forests for classification, regression, density estimation, manifold learning and semi-supervised learning. Tech. Rep. MSR-TR-2011-114, Microsoft Research (2011)

6. Dantone, M., Gall, J., Fanelli, G., Gool, L.V.: Real-time facial feature detection using conditional regression forests. In: Proc. Conference on Computer Vision and Pattern Recognition (CVPR) (2012)
7. Dollar, P., Tu, Z., Perona, P., Belongie, S.: Integral channel features. In: Proc. British Machine Vision Conference (BMVC) (2009)
8. Fanelli, G., Dantone, M., Gall, J., Fossati, A., Van Gool, L.: Random forests for real time 3D face analysis. International Journal of Computer Vision 101(3), 437–458 (2013)
9. Gaschler, A., Jentzsch, S., Giuliani, M., Huth, K., de Ruiter, J., Knoll, A.: Social behavior recognition using body posture and head pose for human-robot interaction. In: Proc. International Conference on Intelligent Robots and Systems (IROS) (2012)
10. Geng, X., Xia, Y.: Head pose estimation based on multivariate label distribution. In: Proc. Conference on Computer Vision and Pattern Recognition (CVPR) (2014)
11. Gross, R., Matthews, I., Cohn, J., Kanade, T., Baker, S.: Multi-PIE (2008)
12. Haj, M.A., González, J., Davis, L.S.: On partial least squares in head pose estimation: How to simultaneously deal with misalignment. In: Proc. Conference on Computer Vision and Pattern Recognition (CVPR) (2012)
13. Hara, K., Chellappa, R.: Growing regression forests by classification: Applications to object pose estimation. In: Proc. European Conference on Computer Vision (ECCV) (2014)
14. Marín-Jiménez, M.J., Ferrari, V., Zisserman, A.: "Here's looking at you, kid." Detecting people looking at each other in videos. In: Proc. British Machine Vision Conference (BMVC) (2011)
15. Murphy-Chutorian, E., Trivedi, M.M.: Head pose estimation in computer vision: A survey. IEEE Trans. on Pattern Analysis and Machine Intelligence 31(4), 607–626 (2009)
16. Stiefelhagen, R.: Estimating head pose with neural networks. In: Proc. International Conference on Pattern Recognition Workshops (ICPRW) (2004)
17. Subramanian, R., Yan, R.Y., Staiano, J., Lanz, O., Sebe, N.: On the relationship between head pose, social attention and personality prediction for unstructured and dynamic group interactions. In: Proc. International Conference on Multimodal Interaction (2013)
18. Sundararajan, K., Woodard, D.L.: Head pose estimation in the wild using approximate view manifolds. In: Proc. Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) (2015)
19. Valenti, R., Sebe, N., Gevers, T.: Combining head pose and eye location information for gaze estimation. IEEE Trans. on Image Processing 21(2), 802–815 (2012)
20. Wu, J., Trivedi, M.M.: A two-stage head pose estimation framework and evaluation. Pattern Recognition 41(3), 1138–1158 (2008)
21. Zhu, X., Ramanan, D.: Face detection, pose estimation, and landmark localization in the wild. In: Proc. Conference on Computer Vision and Pattern Recognition (CVPR) (2012)