Benchmarking Head Pose Estimation In-The-Wild

Elvira Amador[1], Roberto Valle[1], José Miguel Buenaposada[2], and Luis Baumela[1]

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

Abstract. Head pose estimation systems have quickly evolved from simple classifiers estimating a few yaw angles, to the most recent regression approaches that provide precise 3D face orientations in images acquired "in-the-wild". Accurate evaluation of these algorithms is an open issue. Although the most recent approaches are tested using a few challenging annotated data bases, their published results are not comparable. In this paper we review these works, define a common evaluation methodology, and establish a new state-of-the-art for this problem.

Keywords: Head pose estimation, Convolutional neural networks

1 Introduction

We define head pose as the yaw, pitch and roll angles which determine the orientation of the head in the camera reference system [12]. It has attracted much research due to its relevance as a preprocessing step of many face analysis tasks such as alignment of facial landmarks [2, 19] or facial expressions recognition [3]. It is also used in video-surveillance [10] and intrinsically linked with human-computer interaction in social communication [11], gaze [17] and focus of attention [1] estimation.

There are many approaches for image-based head-pose estimation. Some of them use very low resolution images [10] or 3D range data [4]. In this paper we only consider methods that use 2D images of average or high resolution. Among these, *manifold embedding* and *non-linear regression* techniques are possibly the most popular ones. The former assume that separated continuous head-pose subspaces exist according to appearance [13]. Non-linear regression methods learn a mapping from image features to pose angles. Random Forests [4, 18] and Convolutional Neural Networks (CNNs) [5, 9, 14] are some of the most prevailing.

At present, the best performing approaches are based on CNNs. Yang *et al.* [19] use a small CNN for regression of yaw, pitch and roll angles with 3 convolutional layers, 3 pooling layers and 2 full connected layers. Ranjan *et al.* [14] fuse intermediate feature layers at different resolutions, and use a multi-task approach to detect faces, estimate facial landmarks, head pose and gender. The H-CNN architecture [9] uses an inception module [16] that pools and concatenate features from intermediate layers and is jointly trained on the visibility, facial landmarks and head pose estimation parameters. In Table 1 we show the performance of these approaches. Although they use the same data bases, their results cannot be inmediately compared. This will be further discussed in Section 2.

In this paper we review the problem of estimating head pose by regressing the yaw, pitch and roll head angles from medium/high resolution images acquired "in-the-wild", i.e. in realistic unrestricted conditions. Our contributions are:

- A brief survey of the best head pose estimation algorithms.
- Definition of an evaluation methodology and publicly available benchmark to precisely compare the performance of head pose estimation algorithms.
- The establishment of the state-of-the-art on this benchmark.

2 Benchmarking head pose

There are many public data bases with face labeled data. However very few of them provide ground truth head pose, because of the difficulty in accurately estimating these angles. Traditionally, pose estimation algorithms have been evaluated with data bases acquired in laboratory conditions and with imprecise angular information [12]. Later, more realistic and accurate data sets such as AFLW [7] emerged. They have images in challenging real-world situations acquired without any position, illumination or quality restriction.

Here we propose the use of three data bases:

- **AFLW** [7]. It contains a collection of 25993 faces acquired in an uncontrolled scenario with head poses ranging between ±120° for yaw and ±90° for pitch and roll angles. It provides manual annotations of 21 face landmarks. The yaw, pitch and roll angles were obtained automatically from the labeled landmarks using the POSIT algorithm assuming the structure of the mean 3D face.
  We have found several annotations errors and, consequently, removed these faces from our benchmark. From the remaining faces we randomly choose 21074, 2068 and 1000 instances for training, validation and testing respectively. These images will be available after publication.
- **AFW** [20]. This small data base has been traditionally used only for testing purposes. It has 250 images with 468 faces in quite challenging settings. It provides discrete yaw labels ranging from -90° to 90° with 15° intervals, plus the facial bounding box. These labels were manually annotated, hence often they are not very accurate.
- **300W**[3]. It includes 689 challenging faces obtained from the testing subsets of other data bases (HELEN, LFPW and IBUG). This the most popular face alignment benchmark. It provides face bounding boxes and 68 manually annotated landmarks. It does not provide any pose information.
  Following the same procedure as in AFLW, we have used the POSIT algorithm to estimate the three pose angles for each face instance. This data set will also be publicly available.

[3] https://ibug.doc.ic.ac.uk/resources/300-W/

Method	AFLW (MAE)			AFW	300W (MAE)		
	yaw	pitch	roll	yaw	yaw	pitch	roll
Peng *et al.* [13]	-	-	-	86.3%	-	-	-
Valle *et al.* [18]	12.26°	-	-	83.54%	-	-	-
Gao *et al.* [5]	6.60°	5.75°	-	-	-	-	-
Yang *et al.* [19]	-	-	-	-	4.20°	5.19°	2.42°
Ranjan *et al.* [14]	7.61°	6.13°	3.92°	97.7%	-	-	-
Kumar *et al.* [9]	6.45°	5.85°	8.75°	96.67%	-	-	-

Table 1: Head-pose estimation published results. For AFLW and 300W we show the Mean Absolute Error (MAE) in degrees. For AFW we show the classification success rate.

In Table 1 we show the published results of the best head pose estimation algorithms. AFLW figures are not comparable among any of the cited works. Some select 1000 test images at random and use the rest for training [14, 9]. Valle *et al.* [18] chose 10% of the images for testing and the rest for training. Gao *et al.* [5] use 15561 randomly chosen image faces for training and the remaining 7848 for testing. Moreover, none of these AFLW subsets are publicly available, hence it is impossible to make a fair comparison among any of these approaches.

Similarly, the results for AFW are not comparable. Some approaches test on the whole data base [14, 18]. However, each was trained on a different subset of AFLW. Moreover, Kumar *et al.* [9] test on the 341 images whose height is larger than 150 pixels. Peng *et al.* [13] test on a different set of 459 faces.

Finally, in 300W the head pose labels are not available. Yang [19] computes them from an average face composed of 49 3D points. Unfortunately, these pose labels are not publicly available. Additionally, they train on the 300W benchmark training images.

In summary, to have comparable results all algorithms should use the same train, validation and test data sets. For our benchmark we propose to use a single train and validation data set composed respectively by 21074 and 2068 face images randomly chosen from AFLW. For testing we have three data sets: the AFLW test is performed on the remaining 1000 images; when testing with AFW and 300W we use respectively all 468 and 689 faces from AFW and 300W test sets.

3 Experiments

Once we have defined our benchmark, our goal in this section is to establish the base-line results for it.

3.1 Methodology

Following the models used by the best published results [5, 9, 14, 19], we use a distributed face representation extracted from a deep CNN. Training such a model from scratch requires a large amount of data and computing power. The usual approach in computer vision is to use a general architecture already trained on a related problem and fine-tune it for the task at hand (see Fig. 1).

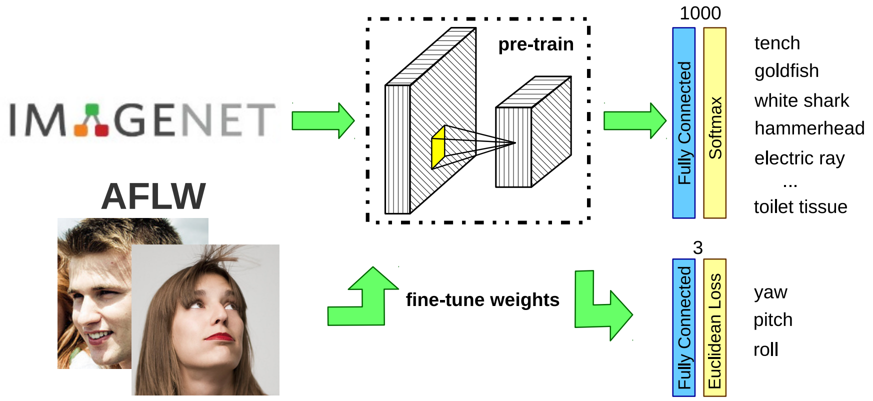

Fig. 1: Transfer learning methodology to fine-tune ImageNet generic weights.

To build our base line regressors we use AlexNet [8], GoogLeNet [16], VGG [15] and ResNet [6] trained architectures, top performers in the image classification task of the ILSVRC competition. AlexNet was also used by Ranjan *et al.* [14], GoogLeNet by Kumar *et al.* [9], and VGG-Net[4] by Gao *et al.* [5]. In each architecture we change the last 1000 units Softmax classification layer with an Euclidean Loss layer with three units for modeling the yaw, pitch and roll angles.

For fine-tuning and evaluation we use the Caffe framework with a GeForce GTX 1080 (8GB) graphics processor. We followed the same procedure for each model. We use Nesterov Accelerated Gradient Descent (NVG) method, initialize the learning rate to $\alpha = 10^{-5}$ and reduce it with $\gamma = 0.1$ factor after "step size" iterations (see Table 2). Momentum was set to $\mu = 0.9$. Table 2 reports the remaining optimization of parameters for each architecture. We optimize the GPU memory occupation by setting the batch length and number of iterations on the basis of the network size. So, large networks use a small batch and larger number of iterations (see Table 2). The network weights used for tests are those at the last iteration. They will be publicly available after publication.

It takes 12 hours for fine tuning the parameters of the largest net, ResNet-152, and process test images on average at a rate of 4 FPS. In Fig. 2 we show a pair of learning curves for VGG-19 and ResNet-152 architectures. Validation curves

[4] They used VGG-Face, a VGG-16 architecture trained on the VGG face data base.

Model	image size	iterations	weight decay	step size	batch
AlexNet [8]	227x227	25000	0.0005	10000	24
GoogLeNet [16]	224x224	25000	0.005	10000	24
VGG-16 [15]	224x224	25000	0.0005	10000	24
VGG-19 [15]	224x224	25000	0.005	10000	24
ResNet-50 [6]	224x224	63000	0.000005	21000	10
ResNet-101 [6]	224x224	126000	0.000005	42000	5
ResNet-152 [6]	224x224	252000	0.005	84000	2

Table 2: Parameters configuration according to each architecture.

are more stable because we always process all test images. However, depending on the batch, the training performance has a larger variance. Vertical dashed red lines mark the number of iterations required to complete an epoch.

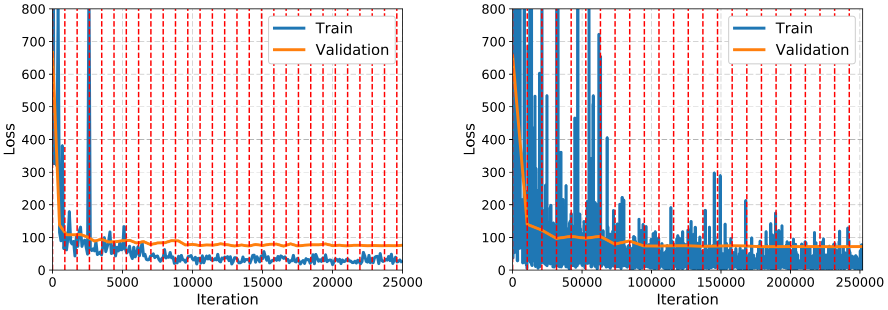

Fig. 2: Learning curves examples for VGG-19 and ResNet-152 architectures.

In Table 3 we present the results of the base line classifiers for each network architecture. In general, these results confirm that the deeper the representation, the better the performance. This is a well-known fact in the deep learning literature [6].

In AFLW we use the Mean Absolute Error (MAE) of each angle as evaluation metric. Hence, the base line model using AlexNet achieves better performance than Ranjan *et al.* [14]. Similarly, GoogLeNet results improve those by Kumar *et al.* [9]. For VGG-16, results are only marginally better thank those by Gao *et al.* [5], although our net was trained on the more general ImageNet data set.

In AFW, since it provides discrete labels, we use as metric the classification success rate. Here, although again the results are also not strictly comparable, the models by Kumar *et al.* [9] and Ranjan *et al.* [14] improve those achieved by our base line classifiers. This is surprising since in the more precise AFLW regression case, the result is the opposite. Perhaps in this case the discretization played against our models or, since AFW was manually labeled, the annotation error is higher. Hence, the MAE differences are less significant. This may also be the reason why the best model, ResNet-152, has a marginally worse performance than ResNet-50.

The MAEs of Yang *et al.* [19] in 300W, although not strictly comparable, are better than those of our base line classifiers. This may be caused by the fact that they train their CNN on the 300W training data set and, perhaps, overfit to it.

Method	AFLW (MAE)			AFW	300W (MAE)		
	yaw	pitch	roll	yaw	yaw	pitch	roll
AlexNet [8]	6.40°	5.21°	3.47°	86.32%	6.86°	6.61°	5.82°
GoogLeNet [16]	6.27°	5.49°	4.05°	95.51%	5.71°	7.99°	6.85°
VGG-16 [15]	6.45°	5.24°	3.61°	88.03%	7.65°	6.69°	6.88°
VGG-19 [15]	5.99°	4.93°	3.15°	94.23%	5.56°	6.35°	4.65°
ResNet-50 [6]	6.03°	5.02°	3.22°	94.44%	5.13°	5.91°	3.23°
ResNet-101 [6]	5.69°	4.97°	3.06°	94.44%	5.71°	5.87°	3.03°
ResNet-152 [6]	5.94°	4.89°	2.98°	94.01%	5.52°	6.16°	3.18°

Table 3: Head pose base line estimation results for different architectures.

Finally, in Fig. 3 we present some representative face images with head pose estimation errors greater than 15° obtained using ResNet-152 architecture. As can be noticed, sometimes the estimation seems to be more accurate than the annotation. This may be caused by the manual annotation error.

4 Conclusions

We have surveyed the state-of-the-art on face pose estimation “in-the-wild”. Although some of the best performing approaches use the same test data bases, their published results are not comparable.

In this paper we have defined an evaluation procedure and benchmark data sets with images captured in unrestricted settings. We have also trained a set of CNN-based base line estimation results supported by this methodology. The model based on the deepest network architecture, ResNet-152, provides the best overall performance. Hence, confirming that deeper representations have better generalization capabilities. When contrasted with the best published results in the literature, although not strictly comparable, the base line model based on ResNet-152 achieves better performance in the challenging and precise AFLW data set. By making publicly available the parameters of the base line classifiers and the benchmark data sets, we expect that future algorithms will be compared on fairer grounds.

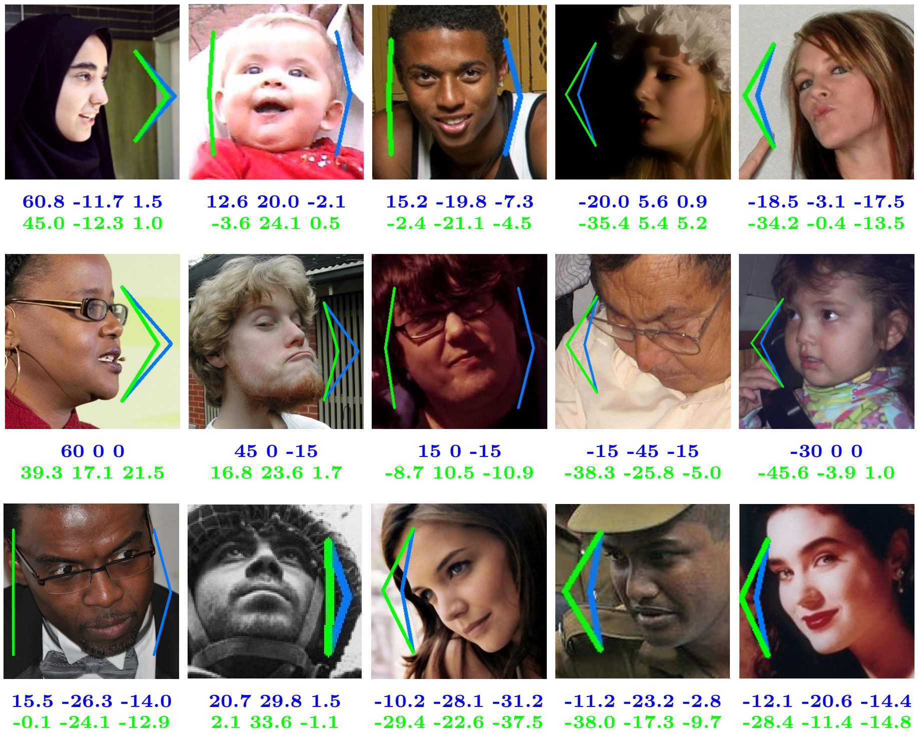

Fig. 3: Representative results with yaw errors greater than 15° for AFLW (top), AFW (middle) and 300W (bottom) data bases. Green and blue colors point out respectively pose estimation and ground truth yaw angle.

Acknowledgements: The authors gratefully acknowledge computer resources provided by the Supercomputing and Visualization Center of Madrid (CeSViMa) and 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 (TPAMI) 33(1), 101–116 (2011)
2. 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)
3. Demirkus, M., Precup, D., Clark, J.J., Arbel, T.: Soft biometric trait classification from real-world face videos conditioned on head pose estimation. In: Proc. Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) (2012)

4. Fanelli, G., Dantone, M., Gall, J., Fossati, A., Van Gool, L.: Random forests for real time 3D face analysis. International Journal of Computer Vision (IJCV) 101(3), 437–458 (2013)
5. Gao, B.B., Xing, C., Xie, C.W., Wu, J., Geng, X.: Deep label distribution learning with label ambiguity. IEEE Trans. on Image Processing (TIP) 26(6), 2825–2838 (2016)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proc. Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
7. Koestinger, M., Wohlhart, P., Roth, P.M., Bischof, H.: Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization. In: IEEE International Workshop on Benchmarking Facial Image Analysis Technologies (2011)
8. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Proc. Neural Information Processing Systems (NIPS) (2012)
9. Kumar, A., Alavi, A., Chellappa, R.: Kepler: Keypoint and pose estimation of unconstrained faces by learning efficient H-CNN regressors. In: Proc. International Conference on Automatic Face and Gesture Recognition (FG) (2017)
10. Lee, D., Yang, M., Oh, S.: Fast and accurate head pose estimation via random projection forests. In: Proc. Conference on Computer Vision and Pattern Recognition (CVPR) (2015)
11. Marín-Jiménez, M.J., Zisserman, A., Eichner, M., Ferrari, V.: Detecting people looking at each other in videos. International Journal of Computer Vision (IJCV) 106(3), 282–296 (2014)
12. Murphy-Chutorian, E., Trivedi, M.M.: Head pose estimation in computer vision: A survey. IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI) 31(4), 607–626 (2009)
13. Peng, X., Huang, J., Hu, Q., Zhang, S., Metaxas, D.N.: Three-dimensional head pose estimation in-the-wild. In: Proc. International Conference on Automatic Face and Gesture Recognition (FG) (2015)
14. Ranjan, R., Patel, V.M., Chellappa, R.: Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. CoRR abs/1603.01249 (2016)
15. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR abs/1409.1556 (2014)
16. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S.E., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proc. Conference on Computer Vision and Pattern Recognition (CVPR) (2015)
17. Valenti, R., Sebe, N., Gevers, T.: Combining head pose and eye location information for gaze estimation. IEEE Trans. on Image Processing (TIP) 21(2), 802–815 (2012)
18. Valle, R., Buenaposada, J.M., Valdés, A., Baumela, L.: Head-pose estimation in-the-wild using a random forest. In: Proc. Articulated Motion and Deformable Objects (AMDO) (2016)
19. Yang, H., Mou, W., Zhang, Y., Patras, I., Gunes, H., Robinson, P.: Face alignment assisted by head pose estimation. In: Proc. British Machine Vision Conference (BMVC) (2015)
20. 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)