Computer Vision for Head Pose Estimation: Review of a Competition

Heikki Huttunen[1], Ke Chen[1], Abhishek Thakur[2], Artus Krohn-Grimberghe[2], Oguzhan Gencoglu[1], Xingyang Ni[1], Mohammed Al-Musawi[1], Lei Xu[1], and Hendrik Jacob van Veen[3]

[1] Tampere University of Technology, Finland
[2] University of Paderborn, Germany
[3] Zorgon, The Netherlands

Abstract. This paper studies the prediction of head pose from still images, and summarizes the outcome of a recently organized competition, where the task was to predict the yaw and pitch angles of an image dataset with 2790 samples with known angles. The competition received 292 entries from 52 participants, the best ones clearly exceeding the state-of-the-art accuracy. In this paper, we present the key methodologies behind selected top methods, summarize their prediction accuracy and compare with the current state of the art.

Keywords: Head pose estimation, computer vision, competition

1 Introduction

Head pose estimation [5, 6, 9–11, 15] attempts to predict the viewing direction of human head given a facial image. More specifically, the output of a head pose estimator consists of the yaw and pitch angles in 2D space and optionally the roll angle in 3D space. The estimation of head orientation is difficult due to variations in illumination, sparsity of data and ambiguity of labels.

On one hand, collecting data for head pose estimation is difficult although there exists large facial databases such as *Labeled Faces in the Wild* [12] and *Youtube Faces Dataset* [22]. However, it is almost impossible to manually annotate these collected images with an exact head orientation. The available solution adopted by the public benchmarking datasets is to ask the participants to look at a set of markers that are located in predefined direction in the measurement room (*e.g.*, 93 direction marks of Pointing'04 dataset [8]). Therefore, the data of the benchmark sets is sparse, both in terms of subjects and angles. For example, there are only 30 images for each head pose angle acquired from 15 subjects in the Pointing'04 dataset. The data are then further divided into training and testing set, which makes the data for training even more sparse.

On the other hand, the annotated labels obtained with the pose direction markers are noisy because they in fact define the direction of *gaze* instead of the *head pose* direction. In other words, even the within-subject head pose direction

can have a large variation while looking at the same marker. As a result, the images with the same label can actually have different true poses.

In addition to the ambiguity caused by the varying appearance of different persons, the mentioned two challenges lead to a complicated observation-label relation, which requires a model that is truly robust. Considering the labels for head pose estimation as sparse discrete integers, such a problem can be formulated into the following three types of frameworks: 1) regression-based approaches [3, 5, 10, 19]; 2) classification-based approaches [6, 11]; and 3) hybrid of the two [9]. In this paper we mostly concentrate on the classification approach, but will first briefly review the principles behind all approaches.

In *regression* frameworks for head pose estimation, a regression mapping is learned from low-level image features to continuous scalar-valued label space. Reference [5] introduced a two-layer regression framework in a coarse-to-fine fashion, which first approximately determines the range of predicted labels and then learns a regression function to discover the exact label values. Alternatively, regression forests have shown superior efficiency in head pose estimation [3] compared to other regression methods. After the introduction of the tree-based approach [3], alternating regression forests were proposed [19] to incorporate the global loss across all trees during training instead of independently growing trees in the original random forests. Recently, a K-clusters regression forest was proposed with a more flexible multi-branch splitting algorithm instead of the standard binary function, thus integrating the locality in randomized label subspace into the whole regression framework [10].

When using the *classification* approach, the labels are treated as independent class labels [11], which discards the ordered dependency across labels. Geng *et al.* [6] introduced the concept of soft labeling to capture the correlation of adjacent labels around true pose and also model the noise in the labels. Guo *et al.* [9] investigated both advantages and disadvantages of regression based and classification based algorithms, and then introduced a hybrid approach by adding an extra classification step to locally adjust the regression prediction.

In this paper we consider a large variety of prediction methods crowdsourced from the research community in a form of a competition. The *TUT Head Pose Estimation Challenge* was organized in the Kaggle.com competition hosting platform[4] during the Fall 2014, and attracted altogether 292 submissions from 52 teams around the world. In the sequel we describe selected methods of the top participants and compare them with recently proposed state-of-the-art methods.

Apart from the learning based approach considered in this paper, there has been increasing interest in geometry-based approaches that fit a geometric face model into the measurements using algorithms such as Iterative Closest Point (ICP). State of the art methods in this field typically use a 3D sensor with applications in the transportation and driver monitoring [17, 21]. The model based approach and the depth measurements can reach significant accuracy gain in comparison to plain 2D data. Nevertheless, plain RGB cameras are extremely widespread and purely data driven methods have surpassed human accuracy in

[4] `https://inclass.kaggle.com/c/tut-head-pose-estimation-challenge/`

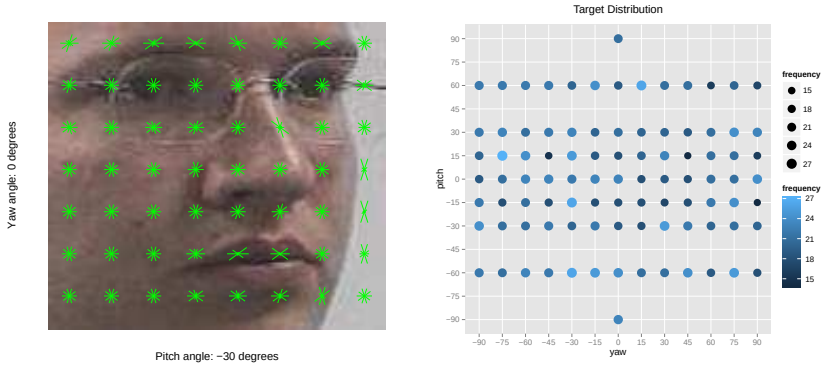

Fig. 1. Left: An example image with green lines illustrating the HOG features. Right: All combinations of yaw and pitch angles in the training set. The size and color of each point represents the number of images in each category.

many areas [20], so we will limit our attention to this line of research. We also hope that the manuscript will serve other researchers in the field as a collection of benchmark methods. All data together with the ground truth and benchmark code are publicly available at the supplementary site of this paper[5].

2 Material and Methods

The material used in the experiments is derived from the widely used Pointing'04 dataset [8]. The original data was collected by requesting test subjects to look at markers located at different viewing directions in the measurement room, and an example of the original images is shown in Figure 1 (left). The locations of the total of 93 angles (markers) are then the basis of the annotations. The 93 directions are the product of 13 different pitch angles and 9 different yaw angles, as illustrated in Figure 1 (right). The dataset in our experiments is slightly modified by a tight cropping and resizing the head area from the original 384×288 resolution to images of size 150×150 pixels, thus forcing the methods to estimate the angles based on relative location of face features instead of the absolute location. The database consists of pictures of 15 subjects, each looking at the 93 angles twice. In total this results in $15 \times 93 \times 2 = 2790$ samples.

The cropped and resized images were transformed to feature vectors using dense Histogram of Oriented Gradient (HOG) features as defined by Felzenszwalb *et al.* [4] with a 9×9 grid. The HOG features are the most common feature set for head pose estimation representing the state of the art in the field [6, 10]. The Felzenszwalb variant differs from the original HOG features [2] in that it uses both directed and undirected histogram bins as well as additional energy features. In our case, the image was split to 9×9 blocks, and the HOG

[5] `http://sites.google.com/TUTheadpose/`

features with 9 undirected bins, 18 directed bins and 4 energy features were calculated to result in a feature vector of dimension $9 \times 9 \times (9 + 18 + 4) = 2511$.

For the competition, the data was split to three parts: The training set with 1953 randomly selected images, validation set with 279 and test set with 558 samples. In other words, the proportions of the three subsets are 70 %, 10% and 20% of all samples. The role of separate validation and test sets is that the competition participants can probe the accuracy of their algorithm on the validation set, while the final standings are determined based on the test set. This discourages overfitting to the test set and results in better generalization.

In the following, we will consider two criteria for prediction accuracy. The main accuracy metric is the Mean Absolute Error (MAE) defined as $\text{MAE} = \frac{1}{2N} \sum_{n=1}^{N} \left(|\hat{\theta}_n - \theta_n| + |\hat{\phi}_n - \phi_n| \right)$, where θ_n and ϕ_n denote the true yaw and pitch angles of the n'th sample and $\hat{\theta}_n$ and $\hat{\phi}_n$ their estimates, respectively. For classification based methods, we will also consider the mean accuracy, *i.e.,* the proportion of cases when the two angles are predicted exactly correct.

2.1 State of the Art

This section reviews two recent algorithms for head pose estimation, which were proposed in 2014 top conferences: Multivariate Label Distribution (MLD) [6] and K-clusters Regression Forests (KRF) [10]. As mentioned in the introductory section, MLD and KRF methods represent the state of the art among classification and regression based approaches, respectively.

K-clusters Regression Forests Based on the standard random forests for regression a *K-cluster Regression Forest* was recently proposed [10] by introducing more flexible node split algorithm instead of binary split. The splitting rule of K-cluster Regression Forests at each node consists of three steps: 1) Cluster the training samples into multiple groups according to the distribution of the label space; 2) Learn the decision function to distinguish the samples in the same cluster from others as a classification problem; 3) Split the data using the predicted cluster label by the trained classifier.

As a result, the novel splitting scheme gives more freedom of choosing partitioning rule and increases the accuracy in comparison to a standard regression forest. It is worth noting that the size of clusters can be determined by either adaptive selection or cross-validation. In the experiments, we adopt the adaptive K-clusters Regression Forests (AKRF) with the same parameters as in [10] as the baseline state-of-the-art regression method for comparing the results generated by the participants of the competition.

Multivariate Label Distribution Multivariate Label Distribution (MLD) is a recently proposed classification method [6] aimed at capturing the correlation between neighboring poses in the label space. Based on standard Label Distribution Learning (LDL), Multivariate Label Distribution is extended to model the

two-dimensional output of head pose estimation (i.e., yaw and pitch angles of head viewing direction), which can mitigate the data sparsity and imbalancedness. By mining the correlation across labels, MLD can intuitively be treated as multi-label learning with correlated labels.

2.2 Top Methods of the TUT Head Pose Estimation Challenge

The TUT Head Pose Estimation Challenge was organized in Fall 2014, and provided the participants readily calculated HOG feature vectors together with the ground truth yaw and pitch angles for the 1953 training samples. The participants were requested to predict the corresponding angles for the validation and test sets. The participating teams were allowed to submit the predicted angles four times each day for assessment. The Kaggle.com platform automatically calculates the accuracy of both subsets but reveals only the validation set accuracy (called public leaderboard score), while the test set accuracy (called private leaderboard score) is visible to organizers only until the end of the competition.

The top scoring participants all use a classification based approach. The exact methods that can be divided into three broad categories: 1) Support Vector Machines (SVM), 2) Neural Networks, and 3) Ensemble methods.

Support Vector Machines The support vector machine is a widely used classifier due to its maximum margin property, which separates the classes with a largest possible distance between them [18]. Typically the SVM is used together with the kernel trick that implicitly maps the data into a high dimensional kernel space, but also the linear kernel is widely used, especially with large data sets.

The basic linear two-class SVM has later been extended to nonlinear decision boundaries via the kernel trick (substituting each dot product by a higher order mapping), and to multiclass classification problems via the one-vs-all (each class is compared against the rest) and one-vs-one (each pair of classes is compared) heuristics. Probably the most famous implementation is the LIBSVM [1], which was also used by the participating methods described below. The LIBSVM implementation is also the optimization engine of many machine learning packages, including the *Scikit-learn* [16] also used by some of the teams.

There were two SVM-based submissions to the TUT Head Pose Estimation Challenge ending up as 2nd (team *Abhishek*) and 4th best (team *Aurora*) in the final results (Table 2). Team *Abhishek* standardizes the features by removing the mean and scaling to unit variance. Without standardization, a feature with large variance may dominate the objective function. The team uses the SVM with Radial Basis Function (RBF) kernel defined as $K(\mathbf{x}, \mathbf{x}') = \exp\left(-\gamma\|\mathbf{x} - \mathbf{x}'\|^2\right)$, with $\gamma \in \mathbb{R}$ a free parameter selected by cross-validation. The method also separates the yaw and pitch angles and trains a separate SVM model for each. So, for both pitch and yaw, a prediction for a given test sample is always among the angles found in the training set. The separation of the full 93 class problem into two problems with 13 and 9 classes simplifies the estimation and may be particularly helpful with the SVM, whose extension to multiclass problems is non-trivial.

Team *Abhishek* fine-tunes the SVM model parameters using an extensive grid-search on a parameter grid consisting of different values of penalty and gamma parameters. The grid search was performed on a 5-fold Cross Validation set and optimized the Mean Absolute Error (MAE) which was set as the evaluation metric for the task. Interestingly, the optimal parameters for both the pitch and yaw model are the same with $C = 10$ and $\gamma = 0.0$.

Team *Aurora* was another team whose solution relies on the SVM. The method is a straightforward application of a single SVM classifier to the data with the original 93 class encoding. The score is however, significantly improves by averaging the predictions of an ensemble of SVM classifiers obtained by randomly subsampling the data. More specifically, random samples of 80% of the training data are used to train a large number of models. Each of these is used for prediction and the resulting predicted angles are then combined together. The team experimented with different fusion strategies, and ended up taking the median of the SVM predictions as the most accurate method.

Neural Networks Artificial neural networks (ANN) are powerful, nonlinear models that can learn complex relationships between variables. They have been studied already for over six decades and have been shown to be successful in various machine learning problems including image recognition [13] and optical character recognition [14]. Due to their nature, the ANN treats the multi-label encoding of the classes in a straightforward manner and does not require any multi-category heuristics like inherently binary classifiers such as the SVM.

Team *ogencoglu* uses ANN in the TUT Head Pose Estimation Challenge placing in the third best position. The method first standardizes the features to zero mean and unit variance, and treats the data as a single estimation problem simultaneously for both angles. However, the encoding of the classes is nontrivial: Instead of the straightforward 93-class encoding, the multi-label target vector is obtained by concatenating the yaw and pitch into 22-element indicator vectors (first 13 elements indicate the yaw angle, and the remaining 9 elements indicate the pitch angle). The target always contains exactly two nonzero elements (one among the first 13 and one among the 9 last ones). The final classification for yaw angle is completed by selecting the angle that gives the maximum output probability among the first 13 outputs. Similarly, classification of pitch angle is performed by examining the remaining 9 elements of the output vector.

The neural network topology consists of 2 hidden layers having 200 and 70 neural units respectively with sigmoid activation functions. The output is a softmax layer of size 22. The neural network is trained with the backpropagation algorithm with minibatch stochastic gradient descent optimization to minimize the negative log-likelihood. The batch size and learning rate are selected to be 50 and 0.01 respectively. The training is run for total of 750 iterations. The solution is implemented using pylearn2 [7] library on an NVIDIA graphics processing unit (GPU) for faster computations.

Table 1. Effect of stacking the classifiers. The first three rows tabulate the public (validation) and private (test) MAE of straightforward use of a 500-tree random forest, 5-nearest neighbor and logistic regression classifiers, respectively. The bottom row shows the decreased MAE when augmenting the original features with the outputs of the first three classifiers.

Model	*Public MAE*	*Private MAE*
500-tree random forest	6.156	6.546
5-nearest neighbor	6.828	7.460
Logistic regression	6.694	6.949
Extremely randomized trees (with stacking)	**4.772**	**4.718**

Ensemble Methods and Stacked Generalization Stacked Generalization was proposed already in 1992 as a tool for improved generalization using a pool of classifiers. The seminal paper by Wolpert [23] has inspired later work on averaging the predictions of a collection of classifiers in various ways. The basic principle is to train a pool of first level classifiers and feed their outputs to a second layer predictor, possibly together with the original features.

In the TUT Head Pose Estimation Challenge, team *Triskelion* used the stacked generalization framework ending up on the 6th place. The first layer of classifiers consists of a pool of logistic regression, random forest and nearest neighbor classifiers. The predicted class membership probabilities of the three are appended to the 2511-dimensional feature vector as three additional higher level features. Note that the three classifiers are first trained on the training set, after which their outputs are calculated for the training, validation and test sets. At first sight one could imagine that the augmented features are highly overfitted to the training set, but practice has shown this not to be the case. As the final second layer predictor, an extremely randomized trees classifier is trained on the training data with augmented features. The problem is encoded as a multi-class classification task. In other words, separate models are trained for yaw and pitch angles. Table 1 shows the effect of stacking in terms of Public MAE and Private MAE for the individual models and the stacked ensemble. One can see that adding the three high-level features decreases the error about 30 %.

3 Results

The TUT Head Pose Estimation Challenge was open for submissions approximately one month. During that period, altogether 292 entries were submitted by 52 players in 37 teams. As a baseline, the competitors were given the result of a ridge regression model, whose MAE score for the test set equals 9.06. The MAE can be lowered in a straightforward manner to approximately 6.0 using, *e.g.*, random forest classifier with enough trees. In this section we concentrate the top-6 teams, whose entries clearly outperform this level.

Table 2. Competition results. All numbers denote the Mean Absolute Error between the true and predicted yaw and pitch angles for the test set (20 % of all data). The best score in each column is highlighted in boldface font.

Method	*Pitch MAE*	*Yaw MAE*	*Overall MAE*	*Pitch Accuracy*	*Yaw Accuracy*	*Overall Accuracy*
Team f623	3.47	5.30	4.38	**0.81**	0.66	0.54
Team Abhishek	3.55	5.30	4.42	0.80	0.67	0.53
Team ogencoglu	4.17	**4.78**	4.48	0.79	**0.71**	**0.55**
Team Aurora	3.84	5.54	4.69	0.79	0.66	0.52
Team RainStorm	3.92	5.24	4.58	0.80	0.69	0.54
Team Triskelion	4.23	5.31	4.77	0.73	0.60	0.43
TOP-6-mean	3.76	5.17	4.46	0.60	0.44	0.26
TOP-6-median	**3.35**	5.04	**4.20**	0.80	0.64	0.51
KRF [10]	5.33	6.03	5.68	0.29	0.17	0.05
MLD [6]	4.49	5.43	4.96	0.76	0.65	0.48

The results of the six top performing teams for the test data (private MAE score) are summarized at the top of Table 2. The columns of the table correspond to the MAE of the pitch and yaw angles separately, and the third column is the average of the two. The three rightmost columns show the classification accuracy of the methods, for pitch and yaw angles and their average, respectively. More specifically, the accuracy refers to the proportion of cases where the angle was predicted exactly as annotated. Note that this measure is not reliable with regression based methods, as the exact prediction seldom occurs in a continuous-valued output (same applies to averaged output of a classifier). Nevertheless, we include this accuracy criterion as it gives valuable insight as to why a particular method works well.

From the table, one can clearly see that the differences between the top performing teams is relatively small. In terms of the pitch angle, the SVM based approaches (teams *Abhishek* and *Aurora*) seem to dominate, while the yaw angle is most accurately predicted by the neural network (team *ogencoglu*).

In addition to the prediction errors of individual submissions, the table also shows the accuracy of committee predictors. More specifically, the rows *TOP-6-mean* and *TOP-6-median* are the scores of combining the TOP-6 teams by averaging and taking the median of the 6 predictions, respectively. Table 2 shows that averaging the predictions does not improve the prediction accuracy compared to individual submissions. Instead, the median of individual submissions clearly improves the accuracy compared to any individual submission.

The two bottom rows of the table show the accuracy of two recent reference methods for this data. The KRF method [10] is a recent regression tree based method, and MLD [6] is a classifier method, both developed for the same feature extraction approach as with our data. However, one can clearly see that the methods of the competitors clearly outperform the state of the art.

4 Discussion

In this paper we summarized a collection of well performing methods for head pose estimation from still images. Moreover, the approach illustrates the importance of collaboration between different players in developing accurate solutions to real world problems. In the case of the TUT Head Pose Estimation Challenge, the competition was originally opened as an exercise for the participants of a graduate level course, but soon gathered submissions from an international audience. The discussion on the competition forum was quite lively, discussing various approaches and proposing novel ideas.

The paper describes a collection of community machine learning methods for head pose estimation. Data driven machine learning is becoming more and more mainstream as the advances in software and hardware allows easier adoption of recent methods. The paper gives a partial answer on how well a generic data driven machine learning methods implemented by non-experts in the field (of head pose estimation) compare against tailored state of the art methods.

The results of the top-scoring teams are clearly exceeding the state of the art. One should bear in mind that the submissions are somewhat optimized against this particular dataset and its HOG representation. Although the test data was hidden from the participants until the end, the results probably are slightly optimistic and favourable for the participating teams, because the data split was random and not a more systematic "leave-one-person-out" type of split. On the other hand, the competitors were not given the origin of the dataset, so they were not aware of the exact feature extraction procedure.

With thorough optimization the performance of the comparison methods could probably be improved because the image size and feature extraction parameters of the two are different from the test data (and from each other). Nevertheless, the top competitors used *general* machine learning tools without domain specific knowledge (the approach of a non-academic pattern recognition engineer implementing a real application) and proved that they can reach the accuracy of highly sophisticated *tailored* algorithms.

The number of submissions to the competition was relatively large, and gained worldwide attention. The feedback from the students of the course was also positive proving the significance of gamification as a tool for motivating students to put forth their best effort and to combine research aspects with classroom education.

References

1. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. ACM Trans. on Intell. Syst. and Technology 2(3), 27 (2011)
2. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE Conf. Comp. Vis. and Patt. Recogn. vol. 1, pp. 886–893 (2005)
3. Fanelli, G., Gall, J., Van Gool, L.: Real time head pose estimation with random regression forests. In: IEEE Conf. Comp. Vis. and Patt. Recogn. pp. 617–624 (2011)

4. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE Trans. Pattern Anal. Machine Intell. 32(9), 1627–1645 (2010)
5. Foytik, J., Asari, V.K.: A two-layer framework for piecewise linear manifold-based head pose estimation. Int. J. of computer vision pp. 270–287 (2013)
6. Geng, X., Xia, Y.: Head pose estimation based on multivariate label distribution. In: IEEE Conf. Comp. Vis. and Patt. Recogn. pp. 1837–1842 (June 2014)
7. Goodfellow, I.J., Warde-Farley, D., Lamblin, P., Dumoulin, V., Mirza, M., Pascanu, R., Bergstra, J., Bastien, F., Bengio, Y.: Pylearn2: a machine learning research library. arXiv preprint arXiv:1308.4214 (2013)
8. Gourier, N., Hall, D., Crowley, J.L.: Estimating face orientation from robust detection of salient facial structures. In: FG Net Workshop on Visual Observation of Deictic Gestures. pp. 1–9. FGnet (IST–2000–26434) Cambridge, UK (2004)
9. Guo, G., Fu, Y., Dyer, C., Huang, T.: Head pose estimation: Classification or regression? In: Int. Conf. Pattern Recognition. pp. 1–4 (Dec 2008)
10. Hara, K., Chellappa, R.: Growing regression forests by classification: Applications to object pose estimation. In: Eur. Conf. Comp. Vis. pp. 552–567. Springer (2014)
11. Huang, C., Ding, X., Fang, C.: Head pose estimation based on random forests for multiclass classification. In: Int. Conf. Pattern Recognition. pp. 934–937 (2010)
12. Huang, G.B., Ramesh, M., Berg, T., Learned-Miller, E.: Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Tech. Rep. 07-49, University of Massachusetts, Amherst (October 2007)
13. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Adv. in neural inf. proc. syst. pp. 1097–1105 (2012)
14. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE 86(11), 2278–2324 (1998)
15. Murphy-Chutorian, E., Trivedi, M.M.: Head pose estimation in computer vision: A survey. IEEE Trans. Pattern Anal. Machine Intell. pp. 607–626 (2009)
16. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. J. Machine Learning Res. 12, 2825–2830 (2011)
17. Pelaez C, G., Garcia, F., de la Escalera, A., Armingol, J.: Driver monitoring based on low-cost 3-d sensors. IEEE Trans. on Intelligent Transportation Syst. 15(4), 1855–1860 (Aug 2014)
18. Scholkopf, B., Smola, A.J.: Learning with kernels: support vector machines, regularization, optimization, and beyond. MIT press (2001)
19. Schulter, S., Leistner, C., Wohlhart, P., Roth, P.M., Bischof, H.: Alternating regression forests for object detection and pose estimation. In: IEEE Conf. Computer Vision. pp. 417–424 (2013)
20. Taigman, Y., Yang, M., Ranzato, M., Wolf, L.: Deepface: Closing the gap to human-level performance in face verification. In: IEEE Int. Conf. Computer Vision. pp. 1701–1708 (2014)
21. Tulyakov, S., Vieriu, R.L., Semeniuta, S., Sebe, N.: Robust real-time extreme head pose estimation. In: Int. Conf. Pattern Recogn. pp. 2263–2268 (Aug 2014)
22. Wolf, L., Hassner, T., Maoz, I.: Face recognition in unconstrained videos with matched background similarity. In: IEEE Int. Conf. Computer Vision. pp. 529–534 (2011)
23. Wolpert, D.H.: Stacked generalization. Neural networks 5(2), 241–259 (1992)