Real-Time Head Pose Estimation Using Multi-Variate RVM on Faces in the Wild

Mohamed Selim, Alain Pagani, Didier Stricker

Augmented Vision Research Group,
German Research Center for Artificial Intelligence (DFKI),
Tripstaddterstr. 122, 67663 Kaiserslautern, Germany
Technical University of Kaiserslautern
{mohamed.selim,alain.pagani,didier.stricker}@dfki.de
http://www.av.dfki.de

Abstract. Various computer vision problems and applications rely on an accurate, fast head pose estimator. We model head pose estimation as a regression problem. We show that it is possible to use the appearance of the facial image as a feature which depicts the pose variations. We use a parametrized Multi-Variate Relevance Vector Machine (MVRVM) to learn the three rotation angles of the face (yaw, pitch, and roll). The input of the MVRVM is normalized mean pixel intensities of the face patches, and the output is the three head rotation angles. We evaluated our approach on the challenging YouTube faces dataset. We achieved a head pose estimation with an average error tolerance of $\pm 6.5^\circ$ in the yaw rotation angle, and less than $\pm 2.5^\circ$ in both the pitch and roll angles. The time taken in one prediction is 2-3 milliseconds, hence suitable for real-time applications.

Keywords: Head Pose Estimation, Real-Time, MVRVM, YouTube Faces

1 Introduction

Head pose estimation is an important computer vision problem. It can be looked at as an individual problem, or as an important module in other problems. It can be the goal of a system, like in detecting user's gaze in Human-Computer Interaction systems (for example, digital signage displays). Thus, head pose estimation has a variety of uses in real world applications.

Moreover, head pose estimation is an important pre-processing step in solving various computer vision problems. Many computer vision tasks and image understanding techniques rely on a reliable, fast, and accurate head pose estimator. Examples of such computer vision problems are: gaze direction estimation, pose-invariant gender or age classification. In [17], the authors combined head pose with eye localization for solving the problem of gaze estimation.

The problem depicted in literature [12] as the head pose is an important factor when solving problems that deal with faces, or with facial analysis. For example, pose estimation can be an important pre-processing step in implementing a pose-invariant age or gender classifier. One can have different classifiers for gender

that are trained on different poses of the face. Another example of using the pose estimation is facial expression recognition.

Fig. 1. Sample detections from the YouTube faces dataset. The red rectangle depicts the detected face, and the estimated *yaw* angle is indicated inside the green circle at the top-left corner. The circle represents a top view of the face, and the green line inside it shows the detected *yaw* angle. A detected angle of of 0 °is indicated by a line pointing downward. Despite the images having different backgrounds, presence of eye glasses or not, or some occlusion on the face, our method can predict the head pose correctly

Due to the importance of the head pose estimation problem, either as a goal itself or as part of other more complex systems, considerable attraction appeared in literature [12], and considerable effort has been put in solving the head pose estimation problem. Another important aspect is that in some situations, a fast algorithm is required to allow the integration of the head pose estimator module in other systems without adding a considerable overhead.

Solving the problem of the head pose estimation can be carried out in one of several ways. When using a classifier, a rough head rotation estimation can be carried out, and the output of such system is either, the head has a frontal pose, right profile or left profile. However, the problem can be viewed a multi-class classification problem, where the data can be classified according to the main head rotation angles. For example, the classes can be according to the yaw rotation angle $+90, +45, 0, -45, -90$. The problem can be solved by for example using a sufficient amount of training dataset, and a SVM. One of the datasets that provide different poses is the FERET dataset[14].

In case we need to allow the detection of more rotation angles, modeling the problem as a classification one would result in many classes that might not be suitable for separation in prediction. Thus, we model the problem as a regression problem, where the output of the trained regressor is a value in a probabilistic range of values that are detected by the range of motion of the head. Moreover, we detect the three rotation angles of the head in prediction on a Multi Variate Relevance Vector Machine.

The paper is organized as follows, the idea behind Relevance Vector Machines and their differences from SVMs are discussed in the next section. In section 3 the theory of our approach is discussed in details. In section4 we present the results and the evaluations carried on the challenging YouTube faces dataset[18] and also on a sample images that we collected. Finally, we conclude and summarize the paper in section 5, and present some future work ideas.

2 Related Work

This section discusses in details the Relevance Vector Machine. Later, various pose estimation techniques are presented.

2.1 Relevance Vector Machine

The RVM, short for Relevance Vector Machine, proposed by Tipping [16], adapts the main ideas of Support Vector Machines (SVM) to a Bayesian context. Results appeared to be as precise and sparse as the SVMs, moreover, yielded a full probability distribution as output of the prediction unlike the SVM which yields non probabilistic predictions [16]. The RVMs fit in our approach as the required output is the three angles of the head, which are floating point values in a probabilistic range. RVMs learn a mapping between input vector $\mathbf{y}$ and output vector $\mathbf{x}$ of the form:

$$x = W\phi(y) + \xi, \tag{1}$$

where ξ is a Gaussian noise vector with 0 mean and diagonal covariance matrix. ϕ is a vector of basis function of the form $\phi(y) = (1, k(y, y_1), k(y, y_2), ..., k(y, y_n))^T$, where k is the kernel function. and y_1 to y_n are the input vectors from the training dataset. The weights of the basis functions are written in the matrix W. In the RVM framework, the weights of each input example are governed by a set of hyperparameters, which describe the posterior distribution of the weights.

During training, a set of input-output pairs (x_i, y_i) are used to learn the optimal function from equation 1. To achieve this, the hyperparameters are estimated iteratively. Most hyperparameters go to infinity, causing the posterior distributions to effectively set the corresponding weights to zero. This means that the matrix W only has few non-zero columns. The remaining examples with non-zero weights are called *relevance vectors*.

Tipping's original formulation only allows regression from multivariate input to uninvariate output. In our approach, the input vector is generated from the

face image. However, the output learnt vector is the three rotation angles of the head. Therefor, we use an extension of the RVM, called MVRVM, short for Multi Variate Relevance Vector Machine proposed by Thayananthan *et al* [15], and using an EM type algorithm for the training.

2.2 Pose estimation

Head pose estimation had much interest during the recent years in literature [12]. We can look at the existing approaches according to the input data they use. Some approaches used 2D images and some uses 3D depth data. Looking into the 2D approaches, we can differentiate between them as some rely on appearances, and some rely on features that are detected on the face.

Most methods use 2D view-based models [6, 13, 10] or 3D models [5, 8]. Regarding the approaches that need facial features detection, they rely on the visibility of the features in the different poses they need to estimate. Many work was done that use Active Appearance Models (AAMs)[6]. They rely on feature detection, tracking, and model fitting, which can lose the tracking or be error prone if the detection of the points or landmarks was not correct.

The work done by [7] uses 3D data that can be captured from depth cameras. This approach cannot be easily applied on video streams, because they use depth data which require special hardware.

In comparison to our approach, we do not rely on depth data, but our approach uses the 2D facial image as input, therefore, can be easily applied in various applications without the need to having 3D cameras.

The problem of head pose estimation can be solved in different resulting spaces. Either discrete angles, or a continuous range of motion. Work done by [19], accepts results with error tolerance of $\pm 15^{\circ}$. The work done by [2] estimates the head pose by detecting and tracking facial landmarks. Relying on the tracking facial landmarks limits the head pose estimation to the visibility of the landmarks. Based on this limitation, the detected head pose in the yaw angle is limited to roughly angles between -60 and +60 degrees. Our approach does not rely on landmark localization tracking, the pose is estimated using the facial image.

3 Approach

The problem considered in this paper is head pose estimation in real-time. One of the advantages of our method is the low computational complexity. Instead of using a hand crafted descriptor, simply normalized mean pixel values are being used as features in our approach, which is proven to acheive high accuracy as discussed later in section 4 in the paper. We use the Multi-Variate Relevance Vector Machines (MVRVM) as it treats the estimation of the head rotation angles as a regression problem. The approach does not rely on high quality images, but it is supposed to work on facial images taken "in the wild", where no conditions apply while capturing the facial image. An overview of the system is

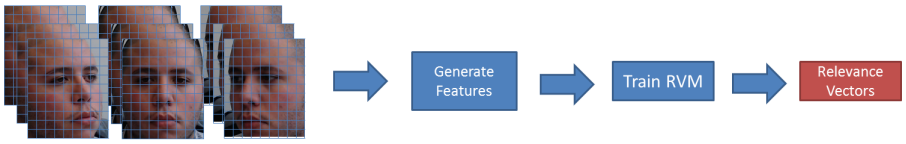

Fig. 2. Input images (for clarification only) are divided into patches. Feature vectors are generated for each image. The RVM is trained with the input feature vectors. The results of the training are the relevance vectors that can be used later in prediction

shown in figure 2. The input image is partitioned into patches, followed by feature extraction. The training images are passed to the Relevance Vector Machine. In iterative learning, the relevance vectors are learnt by the RVM. The following subsections describe the approach in more details.

3.1 Features

The face image is divided into patches by a grid of size $a \times b$ blocks, where a is the number of columns in the X direction and b is the number of rows in the Y direction. For each patch, the mean value of the pixel intensity is calculated. All the mean values are concatenated together, resulting in the feature vector for the input image. The feature vector of the image is normalized as follows.

3.2 Normalization

In order to prepare the data from regression training by the RVM. The feature vector is normalized such that the vector has a zero mean and unit standard deviation. The normalization step adds robustness to light changes that might occur among different input images. First, σ is calculated for the feature vector.

$$\sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \overline{x})^2} \tag{2}$$

, where N is the number of elements in the vector, and $\overline{x}$ is the mean value of the feature vector.

Later, for each element in the feature vector, a normalized value is calculated as in equation 3. The simple features that are used in the training of the Relevance Vector Machine do not require any facial landmark localizations [19], or any complex tracking algorithms. This makes the computation of the pose extremely fast.

$$\forall x_i \in X, \quad v_i = \frac{x_i - \mu}{\sigma} \tag{3}$$

3.3 Parameters Optimization

In order to optimize the Relevance Vector Machine for head pose estimation problem, the parameters included in the process need to be optimized. One of the parameters is the kernel width of the relevance vector machine. It controls the sparsity of the RVM. Varying the kernel width, affects the number of relevance vectors, hence, it has be to optimized so that we avoid the over-fitting problem.

Also, the size of the grid used in feature generation has to be investigated. The partitioning of the face incorporates the pose varying information based on the face appearance. We would like to find the optimal grid size in both horizontal and vertical directions, such that we get the least error possible by the RVM.

In the next section, the datasets used in our evaluation of the approach are discussed.and we show parameter optimization results. We show the result of cross validation training and prediction on the dataset used in our study.

4 Evaluation and results

4.1 Training dataset

In order to evaluate our approach on real data, we need a dataset that has a set of images with continuous degrees varying in the head pose. The standard datasets like FERET [14] has discrete specific values for head pose. We want to evaluate our approach on a dataset that was not captured in a controlled environment, in other words, captured in the wild. Most importantly, we need to evaluate our approach on a dataset that has continuous angles.

The Labeled faces in the wild [9] is a challenging dataset in terms of occlusion, image quality, varying poses, different illumination, etc. However, it does not provide sufficient samples for each subject in different poses. The best candidate to the best of our knowledge is the YouTube faces dataset [18]. The dataset consists of videos of different subjects, and such meets the main requirement of having faces with head rotation angles for different subjects. Also, the range of rotation of most of the subjects in the dataset is wide, for example some videos have yaw rotation from -88 degress up to 80 degrees. Moreover, it is a challenging dataset that was not captured in a controlled environment, nor was it captured using high quality cameras.

First, we tuned the parameters in our approach on the YouTube faces dataset to find the parameters that will yield results with the least error in the rotation angles. Followed by that, we ran 4-fold cross validation on the dataset and reported the results. Following in the section are more details about the datasets and the carried evaluations and optimization.

Youtube Faces Dataset In order to have a regressor that can estimate the pose in high accuracy, a training dataset of faces is required to have different samples at different angles. The FERET dataset [14] is one of the standard

datasets for face analysis. However, it only has an image for a specific pose. Each subject of the FERET dataset has a frontal image(yaw=0) and two profile images(yaw=90,-90), and some other specific angles(45,-45,-67.5,67.5). The dataset was taken with discrete angles, thus, not suitable for our application.

One important property of the dataset is to have continuous angles of the head, this can be found in videos where the subject's head is moving freely. One very challenging dataset is the YouTube faces dataset [18]. It is a dataset of face videos that was designed for studying the problem of unconstrained face recognition in videos. The dataset contains 3425 videos of 1595 different people. All videos were downloaded from YouTube [1]. Each subject in the dataset has an average of 2.15 videos. The shortest video sequence contains 48 frames, the longest one contains 6070 frames, and the average number of frames for the videos is nearly 181 frames. The authors of the dataset followed the example of the Labeled Faces in the Wild image collection [9], which resulted in a large collection of videos.

The dataset was used by [3] in video to video comparisons. Also, it was used by [11] in face verification in the wild from videos. To the best of our knowledge, it was not used for head pose estimation in the wild.

An important feature of the YouTube faces dataset that made us use it in our work is that the three rotation angles (our main interest) of the head are available for each frame in the dataset. The authors of the dataset report that they used the state of the art methods to obtain the rotation angles values. They used the face.com API. This allowed us to perform various evaluations where we can train on one subject only, or train on many subjects.

Parameters Optimization Results The parameter σ_k controls the kernel width, and the sparsity of the RVM, and as mentioned before, it has to be taken care of in order to avoid over-fitting. As we increase the kernel width, the number of relevance vectors decreases and the RVM can predict for new input image in a probabilistic manner. If the kernel parameter is small, the RVM will use all the input feature vectors as relevance vectors, and that means it is not learning anything from the data and cannot differentiate between them.

Figure 3 shows the average error in the three head rotation angles while varying the kernel width σ_k from 1 to 55 with a different value of increments in the iterations. In each iteration we train with 75% of the data and test with the remaining 25%, assuring that the test set is not included in the training set. This is to give us an estimate of the optimal value for the kernel parameter.

We can notice that the average error is decreasing as we increase the kernel width. Also, the number of relevance vectors is decreasing too. The error roughly stayed nearly constant starting from the kernel width 7.5. We did not want to minimize the number of relevance vectors while maintaining the error as low as possible. As shown in the figure, the error starts increasing again at high kernel parameter values. The decided to proceed with kernel width of size 13 as it yields low error in the rotation angles and also not too small number of relevance vectors.

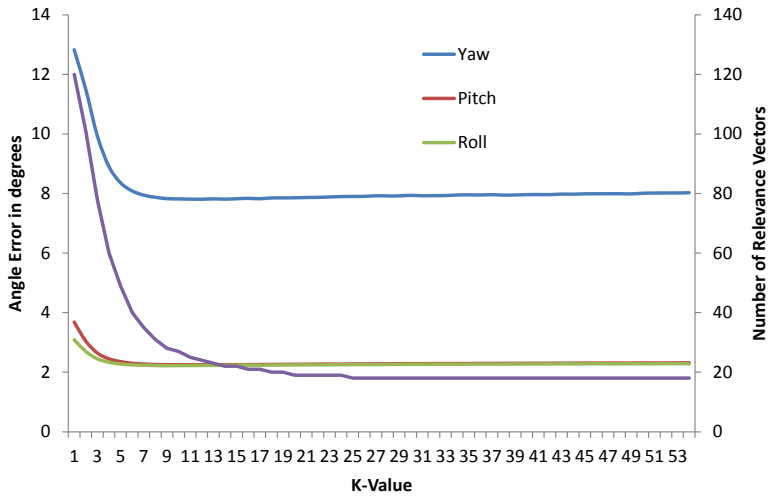

Fig. 3. The effect of varying the kernel width σ_k on the average mean error in the three head rotation angles (yaw, pitch, and roll). On the secondary axis, the number of relevance vector is shown.

We now discuss the optimization of grid size that is used in feature generation as described before. The grid size controls the number of patches on the input image. Using small number of patches (divisions) on the image of the face, reduces the size of the feature vector, which increases the prediction speed. Nevertheless, using small feature vector size reduces the regressor's precision, because the input feature vector doesn't enclose enough information for the head pose among different samples.

In order, to detect precisely, the number of divisions that yields the minimum error, we evaluated the YouTube faces dataset on using different grid sizes. We varied the size of the grid, from 5×5, up to 20×20. We maintained the kernel width σ_k at the optimized value 13. The results of that evaluation are shown in figure 4. We can notice that the size of the grid that yielded the least error in the three rotation angles was 15×15. After this value, the number of relevance vectors kept increasing.

We optimized both the kernel width that controls the sparsity of the RVM and the grid divisions that controls the feature vector size used in the training process. By the experimnetal evaluation shown above, the optimal value of the kernel width is 13, and the optimal value for the grid size is 15.

After performing the optimization on the YouTube faces dataset, we evaluated the whole dataset using the tuned parameters. We ran a 4-fold cross validation tests on all the subjects in the datasets using all the videos provided for each subject.

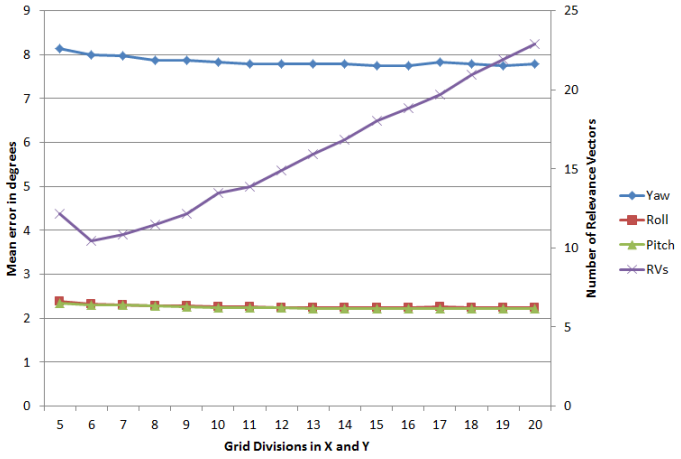

Fig. 4. The effect of varying the grid size used in feature generation, on the average mean error in the three head rotation angles (yaw, pitch, and roll). The grid of size 15×15 yields the best results.

The results as shown in figure 5, show that for more than 75% of the dataset the mean error was less than 10°in the main rotation angle of the head, the yaw angle. Also, for about 20% of the dataset, the error is below 6° in the same angle. So, our approach can achieve good performance on a very challenging dataset, by correctly detecting the head pose for more than 75% of the dataset with error tolerance of $\pm 3 - 4°$ using simple features that doesn't require complex features detection on the face of the subject.

Finally, we tested the proposed approach on unseen videos of the same subject. We used the subjects in the dataset that had more than 2 videos. We trained with two videos, and tested on an unseen video of the subject. The results of that approach are as expected showed less accuracy. The number of subjects included in that test were 533 subjects. The average mean error in the rotation angles were 21, 6, and 5 degrees in the yaw, pitch, and roll rotation respectively. Keeping into consideration we didn't limit the range of the yaw angles in the training, we used the full range provided by the videos, from left profile to right profile appearances. Results are promising for an error of 20 degrees in that challenging test.

The architecture of the machine used in the evaluations is a 6-core Intel Xeon CPU with hyper-threading technology, and 64 GB of RAM. Our evaluation application runs in parallel using the 12 threads provided by the CPU.

Finally, our methods is suitable for real-time applications as the time taken by the computation of one single prediction of the three head rotation angles is only 2-3 milliseconds, with no need of complex landmark detection or model fitting or tracking.

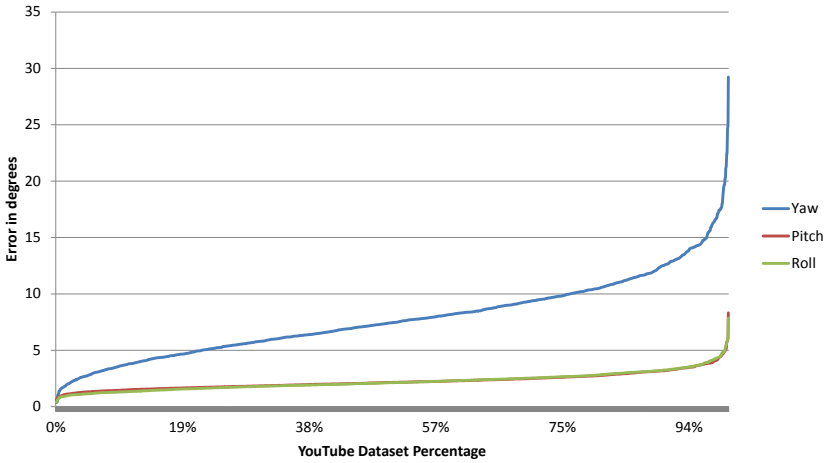

Fig. 5. Results on the YouTube faces dataset. The $\sigma_k = 13$, and the grid size for feature generation is 15×15. The X-axis represents a percentage of the dataset which consisits of 1595 subjects. The Y-axis represents the mean error of the 4-fold cross validation evaluation.

5 Conclusion and future work

5.1 Conclusions

We present a regression scheme for head pose estimation using the appearance of the facial image. The output of our approach is an estimation of the three rotation angles in the full range of the angles, with floating point values. Our approach neither relies on complex features generation, nor does it rely on special landmark localization, but rather relies on the appearance of the facial image. The facial image is divided into patches using a grid of size $a \times b$.

We optimized the division parameters in the features generation. Also, we optimized the kernel parameter σ_k that controls the sparsity of the RVM and the number of relevance vectors learnt during the training. The online prediction of the three head rotation angles is very fast, it takes around 2-3 milliseconds, hence it is suitable for real-time use. This allows the use of the proposed method as a pre-processing step in other applications that rely on the head pose.

We evaluated our approach on a challenging dataset, the YouTube faces dataset. It has images from videos that were taken in uncontrolled environments, with varying face sizes, illumination, some occlusions, etc. We showed that our approach can learn the three head rotation angles using simple features. This approach doesn't rely on depth images, nor 3D information beside the 2D image. Our approach doesn't need landmark detections on the face and can predict full range of motion of the face. We showed that it can learn faces with extreme rotation angles.

The results of our evaluation on the YouTube faces dataset show that we achieve an estimation with error tolerance of $\pm 6.5^\circ$ in the yaw rotation angle, and less than $\pm 2.5^\circ$ on the pitch and roll angles on the whole dataset. For more than 80% of the dataset, our approach estimates the angles with tolerance error of $\pm 10^\circ$. Our final evaluation on the YouTube faces dataset was run in 4-fold cross validation. We also presented promising results on unseen videos of the subjects, taken into different conditions from the training videos.

5.2 Future Work

In order to improve the results and get more fine head pose estimations, a cascade of RVMs can be built in a way that the first RVM can give a rough estimate on the head rotation angles. Following the first regressor, a set of RVMs are to be trained on a smaller range of angles. The Set of RVMs can be one level after the main one, or different number of levels of RVMs can be trained on data with smaller range of angles. The number of levels in the cascade tree and the number of RVMs in each level must be investigated.

We also would like to compare the results of the single RVM and the cascade with more complex pose estimators which rely on facial landmarks.

Moreover, we would like to investigate the user of other light normalization techniques as a pre-processing step in the preparation of the training dataset.

Regarding the evaluations on the YouTube faces datasets, we would like to improve our results on the very challenging 20% of the dataset by running more evaluations in a scheme where the training faces and the testing faces are completely different. We also plan to test the approach on other challenging datasets like the PaSC dataset [4].

References

1. Youtube.
2. Akshay Asthana, Stefanos Zafeiriou, Shiyang Cheng, and Maja Pantic. Incremental face alignment in the wild. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1859–1866. IEEE, 2014.
3. Lacey Best-Rowden, Brendan Klare, Joshua Klontz, and Anil K Jain. Video-to-video face matching: Establishing a baseline for unconstrained face recognition. In *Biometrics: Theory, Applications and Systems (BTAS), 2013 IEEE Sixth International Conference on*, pages 1–8. IEEE, 2013.
4. J.R. Beveridge, P.J. Phillips, D.S. Bolme, B.A. Draper, G.H. Givens, Yui Man Lui, M.N. Teli, Hao Zhang, W.T. Scruggs, K.W. Bowyer, P.J. Flynn, and Su Cheng. The challenge of face recognition from digital point-and-shoot cameras. In *Biometrics: Theory, Applications and Systems (BTAS), 2013 IEEE Sixth International Conference on*, pages 1–8, Sept 2013.
5. Volker Blanz and Thomas Vetter. Face recognition based on fitting a 3d morphable model. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(9):1063–1074, 2003.

6. Timothy F Cootes, Gavin V Wheeler, Kevin N Walker, and Christopher J Taylor. View-based active appearance models. *Image and vision computing*, 20(9):657–664, 2002.
7. Gabriele Fanelli, Juergen Gall, and Luc Van Gool. Real time head pose estimation with random regression forests. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 617–624. IEEE, 2011.
8. Lie Gu and Takeo Kanade. 3d alignment of face in a single image. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 1305–1312. IEEE, 2006.
9. Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.
10. Michael Jones and Paul Viola. Fast multi-view face detection. *Mitsubishi Electric Research Lab TR-20003-96*, 3:14, 2003.
11. Meina Kan, Dong Xu, Shiguang Shan, Wen Li, and Xilin Chen. Learning prototype hyperplanes for face verification in the wild. *Image Processing, IEEE Transactions on*, 22(8):3310–3316, 2013.
12. Erik Murphy-Chutorian and Mohan M Trivedi. Head pose estimation in computer vision: A survey. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(4):607–626, 2009.
13. Alex Pentland, Baback Moghaddam, and Thad Starner. View-based and modular eigenspaces for face recognition. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*, pages 84–91. IEEE, 1994.
14. P. Jonathon Phillips, Hyeonjoon Moon, Syed A. Rizvi, and Patrick J. Rauss. The feret evaluation methodology for face-recognition algorithms. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(10):1090–1104, October 2000.
15. Arasanathan Thayananthan, Ramanan Navaratnam, Bjrn Stenger, PhilipH.S. Torr, and Roberto Cipolla. Multivariate relevance vector machines for tracking. In *Computer Vision ECCV 2006*, volume 3953, pages 124–138. Springer Berlin Heidelberg, 2006.
16. Michael E Tipping. Sparse bayesian learning and the relevance vector machine. *The journal of machine learning research*, 1:211–244, 2001.
17. R. Valenti, N. Sebe, and T. Gevers. Combining head pose and eye location information for gaze estimation. *Image Processing, IEEE Transactions on*, 21(2):802–815, Feb 2012.
18. Lior Wolf, Tal Hassner, and Itay Maoz. Face recognition in unconstrained videos with matched background similarity. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 529–534. IEEE, 2011.
19. Xiangxin Zhu and Deva Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2879–2886. IEEE, 2012.