The CSI Journal on
Computer Science and Engineering
Vol. 13, No. 2, 2016
Pages 42-47
Regular Paper

A Deep Learning Method to Estimate 3D Point of Regard by Joint Head and Eye Information

Rahim Entezari Mohammad Mahdi Arzani Mahmood Fathy
Amir Hossein Bayat

Department of Computer Engineering, Iran University of Science and Technology, Tehran, Iran

Abstract

The development of systems that can characterize the state of the human is now important for many applications. In particular, as an indicator of attention and interest, the human gaze is an important cue in people behaviors, personality, intentions, and activities. Gaze also play a crucial role in the communication process. However, in spite of great advances during last three decades, current gaze estimation methods cannot addresses required conditions in this field, e.g. user head movements and minimum user calibration. There have been some works to resolve such problems but those methods lack good precision. In this work, we have used a method for appearance-based gaze estimation using convolutional neural networks, which is multimodal. This method in our implemented setting significantly outperforms state-of-the-art methods.

Keywords: Gaze Estimation, Convolutional Neural Networks, Head Movement, Attantion, Calibration.

1. Introduction

The most important ways of non-verbal communication consist of facial expressions, hands position, gestures, and gaze. The last one plays a crucial role when people interact, as it is used to regulate the flow of communication, monitor feedback, reflect cognitive activity, express emotions, and communicate the nature of the interpersonal relationship [1]. In general, gaze is a very strong indicator for subject attention process.

In the recent years, as there is much rich information in non-verbal cues, there has been a growing interest from diverse domain on tools, which are able to retrieve the state of human.

Appearance-based gaze estimation which does not need specific tools is a hot topic research in computer vision as it can be used for several application domains, including gaze-based human-computer interaction and visual behavior analysis [2].

We now know that the desired gaze consists of two factors [3] i.e. the head pose and the eye locations. The estimation of these mentioned factors is often achieved using expensive or limiting hardware.Therefore, the problem is often simplified by either considering the head pose or the eye center locations as the only feature to understand the interest of a subject [4].

Here, Appearance-based 3D gaze estimation is a supervised regression problem in which 3D gaze direction is predicted from input features, i.e. a set of an eye image and a 3D head pose. In general, the performance of appearance-based methods depend on the quality and diversity of training data and also generalization ability of the regression algorithm [5].

In most of the previous studies, evaluation has been conducted using the test and training data of the same person, and this leads to less generalization for the different conditions [6].

This paper is based on previous work [6] done by Xucong Zhang, et al .They have employed the LeNet for the deep

network. They have used cropped eyes and head pose information directly integrated with EYEDIAP dataset for the features, i.e. there is no explicit feature extraction phase.

We have deployed the same method except for the following differences in the setting:

- Extracted head pose information
- Cropped eye images
- Number of training data
- Training parameters

Zhang et al. at Max Plank Institute (MPI) have used vectors of 2D angle for the head pose, while we have used vectors of 12, consisting of 9 float numbers of head pose rotation matrix and 3 of translation matrix.

They also cropped both eyes in one image (Figure 1(a)) whereas we have used both eyes in separate images (Figure 1(b)).

They have only used the screen target sequences and have not covered floating target data, which are more challenging and contain many extreme gaze directions, but we have used all dataset videos, divided for the train, test and validation. Figure 2. Shows some frames which are not covered by [6].

(a) Both eyes in one image (b) Left and Right eyes

Figure 1. Example eye images (a) MPI, (b) ours

Figure 2. Example frames not covered by [6]

In this work, we also have used different training parameters for the deep network, which obviously helped us to converge better and get better results. The goal here is to predict the point of regard. i.e. where the participant is looking at (Figure 3).

Below, we review the related works on gaze estimation (Section 2) and then introduce used method (Section 3). In section 4, we talk about details of used method and experiments. Section 5 consists of conclusion and future works.

2. Related Work

The survey by Hansen [1] provides a comprehensive overview of computer vision methods for gaze estimation. In general, gaze estimation methods can be further divided to model-based or appearance-based [1]. Model-based methods use a geometric eye model and are divided into corneal-reflection and shape-based methods, depending on whether they require external light sources to detect eye features.

Some related works on corneal reflection-based methods were focused on stationary setting and some more complicated with arbitrary head poses [6].

On the other hand, the focus of shape-based methods is on pupil center and iris edges [1]. Zhu et al. [7] used thres holding for pupil center estimation. Yamazoe et al. [17] also proposed to fit a geometric model from segmentations of the eye images. These segments were obtained through simple thres holding. During the test phase, they used the iris center derived from a fitted ellipse to infer gaze. Voting-based methods are also used in edge detection process [8]. The more complex shape models [9] were proposed to compensate problems in simple shape-based models, but this leads to more computation and most of these models need high-quality images [1]. Some later works in shape-based methods [10] used the ensemble of Random Regression Trees for pupil center localization. Strupczewski et al. [11] proposed different geometric model for webcam eye gaze tracking.

The most important advantage of appearance-based gaze estimation methods, which are also known as holistic methods [1], is that they use eye images as input, therefore there is potential to work with lower resolution eye images. Some early works assumed a fixed head pose while newer works deal with 3D head pose estimation [6].

It is worth mentioning that there is another category for hybrid models [1]. Some works [12] use methods of combining shape and appearance or shape and color [13].

Appearance-based methods require larger amounts of user-specific training data than model-based methods [1] but it can be compensated with last released datasets, e.g. EYEDIAP which consists many frames.

Williams et al. relied on semi-supervised Gaussian Process Regression (GPR) for visual mapping [14]. More recently, Lu et al. [15] proposed adaptive linear regression which is based on sparse image reconstruction. Their method required a fixed head pose. To remove fixed head pose constraint Lu et al. proposed a Gaussian Process Regression based pose correcting scheme on top of fixed head pose model [16]. Funes et al. [17] used RGBD cameras to directly handle eye appearance variation by generating frontal looking eye images used as input to adaptive linear regression.

Some recent works [18]uses egocentric videos and based on activity, head and hand locations, eye gaze is estimated. In [19] authors used conditional random field as a graphical model to map relation between head, human body pose and face for eye gaze estimation. Mansouryar et al. directly maps 2D pupil positions to 3D gaze directions in scene camera coordinate space [20]. Feng Lu et al. have employed the advantages of recent sparse auto-encoding techniques. They partition any eye image into small patches .Using these patches they learn a codebook comprising a set of bases,

which can reconstruct any eye image patch with sparse coefficients. By examining these coefficients, they can analyze the eye shape more effectively [21]. Krafkaet. al. [22] released the dataset of gaze tracking on the mobile device, called Gaze Tracker. They also have used the convolutional neural network to estimate gaze of mobile device user on the screen. They employed eyes, face and face grid, that is a binary mask used to indicate the location and size of the head within the frame. They reported a prediction error of 1.71cm on mobile phone screen according to the location of the camera.

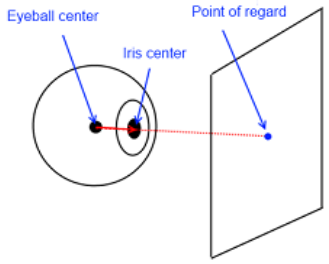

Figure 3. Illustration of point of regard

3. The Proposed Method

We first convert each video to frames, then eyes are extracted from frames. Extracted eyes are fed to a convolutional network. This multimodal convolutional network is the same one used in [6]. They have used LeNet network architecture that consists of one convolutional layer followed by a max-pooling layer. The third layer is a second convolution layer, which is followed by another max-pooling layer, and a final fully connected layer.

Similar to [6] we train a linear regression layer on top of the fully connected layer to predict point of regard. This CNN has two inputs, cropped eyes and head pose information. The head pose information here is nine values of the rotation matrix and three values for translation. A rotation of α radians about the x-axis, β radians about y-axis and γ radians about z-axis are defined as following, respectively [23]:

$$R_x(\alpha)=\begin{bmatrix}1 & 0 & 0\\ 0 & \cos\alpha & -\sin\alpha\\ 0 & \sin\alpha & \cos\alpha\end{bmatrix}$$

$$R_y(\beta)=\begin{bmatrix}\cos\beta & 0 & \sin\beta\\ 0 & 1 & 0\\ -\sin\beta & 0 & \cos\beta\end{bmatrix}$$

$$R_z(\gamma)=\begin{bmatrix}\cos\gamma & -\sin\gamma & 0\\ \sin\gamma & \cos\gamma & 0\\ 0 & 0 & 1\end{bmatrix}$$

So if we rotate first about the x-axis, then the y-axis and finally the z-axis, this can be represented as the matrix product of$R= R_x(\alpha)\ R_y(\beta)\ R_z(\gamma)$ which is equivalent to following Rotation matrix:

$$\begin{bmatrix}\cos\beta\cos\gamma & \sin\alpha\sin\beta\cos\gamma-\cos\alpha\sin\gamma & \cos\alpha\sin\beta\cos\gamma+\sin\alpha\sin\gamma\\ \cos\beta\sin\gamma & \sin\alpha\sin\beta\sin\gamma+\cos\alpha\cos\gamma & \cos\alpha\sin\beta\sin\gamma-\sin\alpha\cos\gamma\\ -\sin\beta & \sin\alpha\cos\beta & \cos\alpha\cos\alpha\end{bmatrix}$$

We tried to extract three values of angles in addition to the translation matrix. As it has not been mentioned in the dataset that what are the right orders of rotations, i.e. x-axis, y-axis then z-axis, x-axis, z-axis then y-axis, etc. we have deployed these six possible permutations to find the corresponding order. We have found that mentioned order "A rotation of α radians about the x-axis, β radians about the y-axis, and γ radians about the z-axis" is the right one. So we tested two scenarios, first the extracted three rotation angles were concatenated to translation matrix; secondly, we have used integrated twelve values. Our results showed the later scenario gives better results. This information used to learn mapping from eye images and head pose vectors to point of regard.

3.1. Preparing Data

We have used EYEDIAP dataset[24]. In total, there are 94 recorded sessions in EYEDIAP. Each session will be denoted by the string "P-C-T-H" which refers to the participant id (1-16), the recording conditions C= (A or B), the used target T= (DS, CS or FT), i.e. Discrete Screen, Continuous Screen and floating target moving in the space, respectively. The head pose also consists of H=(S or M), static or moving. Each session in conditions "A" correspond to 2.5 minutes of recording time, whereas the sessions recorded in conditions "B" last approximately 3 minutes each. This corresponds to more than 4 hours of data. Table 1 summarize all recorded data.

Table 1. Summary of the recorded sessions

Participants	Recorded sessions
1-11	A-DS-S; A-DS-M; A-CS-S; A-CS-M; A-FT-S; A-FT-M
12-13	B-FT-S; B-FT-M
14-16	A-DS-S; A-DS-M; A-CS-S; A-CS-M; A-FT-S; A-FT-M; B-FT-S; B-FT-M

To standardize the definition of all 3D variables in the data, Funes et al. have defined a common world coordinate system (WCS), in which the variables refers to meters. In this definition, if $p_k \epsilon R^3$ is a point defined w.r.t. the coordinate system of Kinect RGB camera, then defined WCS is the equivalent p_w, w.r.t. the WCS is given by $p_w = R_w p_k + t_w$ where:

$$R_w=\begin{bmatrix}1 & 0 & 0\\ 0 & -1 & 0\\ 0 & 0 & -1\end{bmatrix},\ t_w=\begin{bmatrix}0\\ 0\\ 1\end{bmatrix}$$

In which R_w and t_w are rotation and translation matrices respectively.

The very first stage is to convert videos into frames using OPENCV. We tried to use MATLAB for framing, but the problem was that framed images were not synchronized with the corresponding frame in the video. It seems MATLAB has some problems with framing videos, after framing, we have about 500K frames. It is mentioned in EYEDIAP dataset that some frames are not valid. This is due to two factors, the participant is blinking at the given frame or the participant is not looking at the visual target at the given frame, e.g. whenever the participant is distracted. So non-OK frames are removed from extracted frames using given data within

dataset. For cropping eyes we used the integrated data within the dataset, i.e. Kinect RGB eyeball center left and right (x_l,y_l,x_r,y_r).

3.2. POR Estimation Using CNN

Here we have used Multimodal CNN [6] to learn how to map inputs to gaze coordinates in world coordinate space. Inputs are eye images and 12 value head vectors. This model is multimodal as it uses both eye images and head pose float numbers.

In our model, we have used the Le Net network architecture that consists of one convolutional layer followed by a max-pooling layer, a second convolution layer followed by a max-pooling layer, and a final fully connected layer. We train a linear regression layer on top of the fully connected layer to predict gaze angle vectors. The input to the network is RGB eye images with a fixed size of 40 x 40 pixels, as available eyeball center is at the exact center of this window (Figure 4).

The output of the network is a 3D gaze point where the observer is looking at. The point of regard, which is given within dataset, is compared to this output coordinate. As a loss function, we used the sum of individual L_2 losses that measures the difference between predicted coordinate and the actual point of regard.

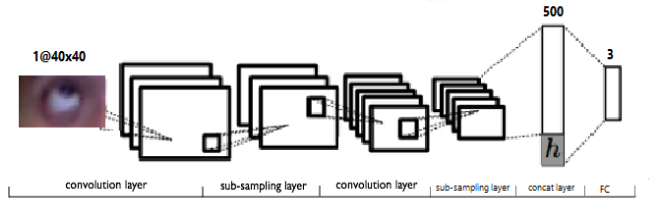

Figure 4. Architecture of used multimodal CNN. Head vectors are added to the output of fully connected layer.Here we have used [6] but with 3D gaze point

4. Experiments

In this section, we discuss gaze estimation task and validate the effectiveness of used multimodal CNN approach.

We compare our method with state-of-the-art methods on the EYEDIAP dataset. We have divided the dataset into training, validation, and test sets.

4.1. Training

We have used the first 10 people of EYEDIAP for training. This means training set consists of about 200K frames. We trained deep network using caffe [25]. Caffe is a deep learning framework made with expression, speed, and modularity in mind. It is developed by the Berkeley Vision and Learning Center (BVLC) and by community contributors. We have used the batch size of 500, so every epoch is 400. We have trained this network from scratch, with the base learning rate of 0.001, the momentum of 0.95 and weight decay of 0.0005. We also have used ad a delta to get better convergence instead of SGD. Delta parameter is set to 1e-6. With respect to validation loss, we have trained our CNN for 80K iterations.

4.2. Validation

The next 2 person are used for validation, and this was about 40K frames. With batch size of 500, we had 80 test iterations with interval of 10K. This last value means we have carried out validating every 10K training iterations.

4.3. Test

We have used remaining dataset videos for the phase of the test, i.e. we have 70K frames. In this phase we have fed each frame of test to the trained model, in addition to the head pose data. The output which is a 3D float number refers to the point the person is looking at. For getting Mean Error Degree, for each testing frame, we have connected the center of eyes to the predicted and actual gaze point which results in 2 vectors. The degree between these two vectors are calculated with the following formula:

$$\Theta = \cos^{-1}\left(\frac{\vec{x}.\vec{y}}{||x||||y||}\right)$$

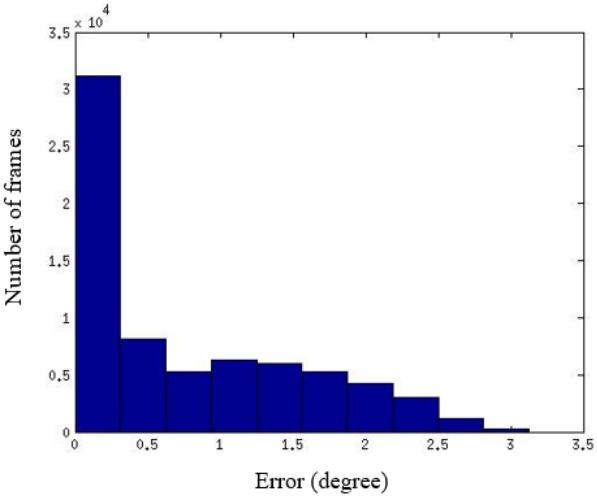

Figure 5. histogram of error for test frames

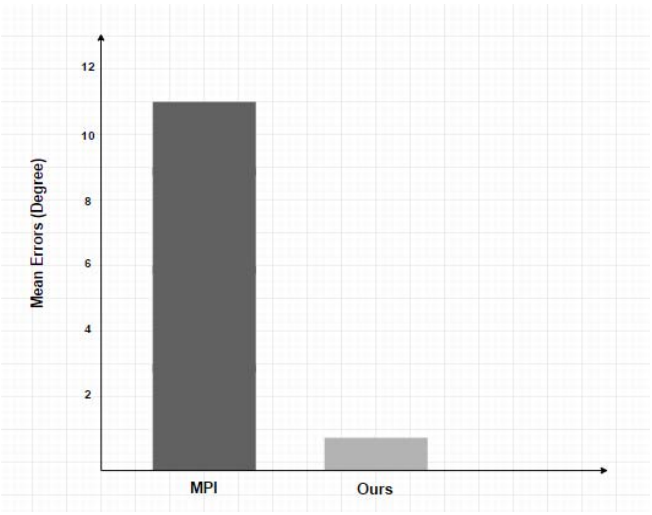

Figure 6. performance of our method compared to [6]

Figure 5 show the histogram of error for all test frames. As you can see in figure 6. We have much better result than state-of-the-art [6] method for EYEDIAP dataset. The mean

error degree is less than one degree, meaning the difference between actual gaze vector and estimated one is about 0.79 degree.

Figure 7. Also shows the output of some layers in this CNN.

In this figure, (a) corresponds to an example of input RGB frame. This input is cropped eye images. (b) Shows the filters of first convolution layer, (c) shows the output of first convolution layer. Another layer shown in this figure is max-polling, which is a form of non-linear down-sampling. Max-pooling partitions the input image into a set of non-overlapping rectangles and, for each such sub-region, outputs the maximum value:

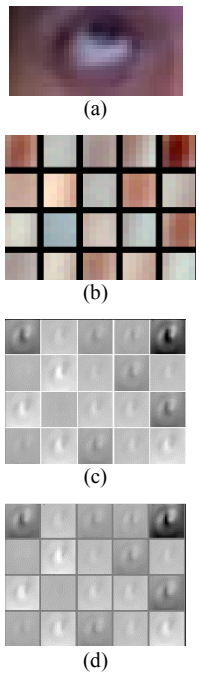

Figure 7. Figure 7 (a) Cropped eye, (b) conv1 filters, (c) conv1 output (rectified responses of the filters), (d) pool1 output

5. Conclusion and Future Work

Despite lots of works mentioned in literature, there have been few works for appearance-based gaze estimation with low-quality images. Those with this condition evaluated exclusively under controlled conditions with low range head poses. In this work we have employed [6] method with different setting for EYEDIAP dataset which varies in different head pose and illumination conditions. This dataset is very challenging as floating target sequences contain many extreme gaze directions. Our setting for CNN-based estimation model significantly outperforms [6] which is state-of-the-art. Our future work will consist in using different deep network. We also will try to get eye images and head pose automatically.

References

[1] Hansen, D. Witzner, and Q. Ji, "In the eye of the beholder: A survey of models for eyes and gaze," *Pattern Analysis and Machine Intelligence, IEEE Transactions on* pp. 478-500, 2010.

[2] C. H. Morimoto, and M. R. M. Mimica, "Eye gaze tracking techniques for interactive applications," *Computer Vision and Image Understanding,* vol. 98, pp. 4-24, 2005.

[3] S. R. Langton, H. Honeyman, and E. Tessler, "The influence of head contour and nose angle on the perception of eye-gaze direction," *Perception & psychophysics,* vol. 66, pp. 752-771, 2004.

[4] N. Robertson, and I. Reid, "Estimating gaze direction from low-resolution faces in video," in *European Conference on Computer Vision*, pp. 402-415, 2006.

[5] Y. Sugano, Y. Matsushita, and Y. Sato, "Learning-by-synthesis for appearance-based 3d gaze estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1821-1828, 2014.

[6] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, "Appearance-based gaze estimation in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4511-4520, 2015.

[7] Z. Zhu, K. Fujimura, and Q. Ji, "Real-time eye detection and tracking under various light conditions," in *Proceedings of the 2002 symposium on Eye tracking research & applications*, pp. 139-144, 2002.

[8] R. Valenti, and T. Gevers, "Accurate eye center location and tracking using isophote curvature," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1-8, 2008.

[9] I. F. Ince, and J. W. Kim, "A 2D eye gaze estimation system with low-resolution webcam images," *EURASIP Journal on Advances in Signal Processing,* vol. 2011, pp. 1-11, 2011.

[10] N. Markuš, M. Frljak, I. S. Pandžić, J. Ahlberg, and R. Forchheimer, "Eye pupil localization with an ensemble of randomized trees," *Pattern recognition,* vol. 47, pp. 578-587, 2014.

[11] A. Strupczewski, B. Czuprynski, J. Naruniec, and K. Mucha, "Geometric Eye Gaze Tracking," in *International Conference on Computer Vision Theory and Applications*, pp. 446-457, 2016.

[12] D. W. Hansen, J. P. Hansen, M. Nielsen, A. S. Johansen, and M. B. Stegmann, "Eye typing using Markov and active appearance models," in *Applications of Computer Vision, 2002.(WACV 2002). Proceedings. Sixth IEEE Workshop on*, pp. 132-136, 2002.

[13] D. W. Hansen, "Using colors for eye tracking," *Color Image Processing: Methods and Applications,* pp. 309-327, CRC Press, 2006.

[14] O. Williams, A. Blake, and R. Cipolla, "Sparse and Semi-supervised Visual Mapping with the S^ 3GP," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06),* pp. 230-237, 2006.

[15] F. Lu, Y. Sugano, T. Okabe, and Y. Sato, "Adaptive linear regression for appearance-based gaze estimation," *IEEE transactions on pattern analysis and machine intelligence,* vol. 36, pp. 2033-2046, 2014.

[16] F. Lu, T. Okabe, Y. Sugano, and Y. Sato, "A Head Pose-free Approach for Appearance-based Gaze Estimation," in *BMVC*, pp. 1-11, 2011.

[17] K. A. F. Mora, "3D Gaze Estimation from Remote RGB-D Sensors," PhD Thesis, École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, 2015.

[18] Y. Li, A. Fathi, and J. M. Rehg, "Learning to predict gaze in egocentric video," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3216-3223, 2013.

[19] B. Benfold, and I. Reid, "Unsupervised learning of a scene-specific coarse gaze estimator," in *2011 International Conference on Computer Vision*, pp. 2344-2351, 2011.

[20] M. Mansouryar, J. Steil, Y. Sugano, and A. Bulling, "3D gaze estimation from 2D pupil positions on monocular head-mounted eye trackers," in *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*, pp. 197-200, 2016.

[21] F. Lu, and X. Chen, "Person-independent eye gaze prediction from eye images using patch-based features," *Neurocomputing,* vol. 182, pp. 10-17, 2016.

[22] K. Krafka, A. Khosla, P. Kellnhofer, H. Kannan, S. Bhandarkar, W. Matusik, and et al., "Eye tracking for everyone," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2176-2184, 2016.

[23] G. G. Slabaugh, "Computing Euler angles from a rotation matrix," accessed on June 2016, at http://thomasbeatty.com/MATH%20PAGES/ARCHIVES%20-%20NOTES/Applied%20Math/euler%20angles.pdf.

[24] K. A. F. Mora, F. Monay, and J.-M. Odobez, "EYEDIAP: a database for the development and evaluation of gaze estimation algorithms from RGB and RGB-D cameras," in *Proceedings of the Symposium on Eye Tracking Research and Applications*, pp. 255-258, 2014.

[25] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, and et al., "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia*, pp. 675-678, 2014.

Rahim Entezarireceived the B.S. degree in Computer Engineering from Amirkabir University (Tehran Polytechnic), Iran, 2013. He is currently pursuing the MSc. program of Artificial Intelligence at Iran University of Science and Technology. His current research interests include deep learning, probabilistic graphical models, and computational neuroscience.
E-mail: r_entezari@comp.iust.ac.ir

Mohammad Mahdi Arzani received the B.S. degree in Computer Engineering from Shahed University, Iran ,2008, and the M.S. degree in Artificial Intelligence from Sharif University of Technology, Iran , 2010. He is currently pursuing the Ph.D. program at Iran University of Science and Technology. His current research interests include deep learning, probabilistic graphical models and human behavior analysis.
E-mail: marzani@iust.ac.ir

Amir Hossein Bayat recievied his BSc in Computer Engineering from K.N.Toosi University of Technology in 2014 He is currently pursuing the MSc. program of Artificial Intelligence at Iran University of Science and Technology. His research interests include computer vision, machine learning, probabilistic graphical models, deep learning and big data mining.
E-mail: a_bayat@comp.iust.ac.ir

Mahmood Fathyreceived his BSc in electronics from Iran University of Science and Technology in 1984, MSc in computer architecture in 1987 from Bradford University, United Kingdom, and PhD in image processing computer architecture in 1991 from University of Manchester, United Kingdom. Since 1991, he has been an academic member in the Computer Engineering School of IUST. His research interests include image and video processing, parallel and distributed processing machine learning, vehicular social network, and big data mining.
E-mail: mahfathy@iust.ac.ir

Paper Handling Data:

Submitted: 11.11.2016
Received in revised form: 15.12.2016
Accepted: 27.12.2016
Corresponding author: Dr. Mahmood Fathy,
Department of Computer Engineering, Iran University of Science and Technology, Tehran, Iran.