Knowledge-Based Systems 000 (2016) 1–9

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

# Appearance-based gaze estimation using deep features and random forest regression

Yafei Wang[a,b], Tianyi Shen[b], Guoliang Yuan[b], Jiming Bian[a], Xianping Fu[b,*]

[a] School of Physics and Optoelectronic Engineering, Dalian University of Technology, Dalian 116024, China
[b] Information Science and Technology College, Dalian Maritime University, Dalian 116026, China

ARTICLE INFO

*Article history:*
Received 22 March 2016
Revised 26 July 2016
Accepted 29 July 2016
Available online xxx

*Keywords:*
Appearance
Gaze estimation
Deep features
Random forest
CNN

ABSTRACT

Conventional appearance-based gaze estimation methods employ local or global features as eye gaze appearance descriptor. But these methods don't work well under natural light with free head movement. To solve this problem, we present an appearance-based gaze estimation method using deep feature representation and feature forest regression. The deep feature is learned through hierarchical extraction of deep Convolutional Neural Network (CNN). And random forest regression with cluster-to-classify node splitting rules is used to take advantage of data distribution in sparse feature space. Experimental results demonstrate that the deep feature has a better performance than local features on calibrated gaze regression. The combination of deep features and random forest regression provides an effective solution for gaze estimation in a natural environment.



## 1. Introduction

The eye gaze plays a significant role in understanding human attention, feeling and mind. It is essential for many multimedia applications such as cognitive processes analysis and human-computer interaction [1]. Although many non-intrusive gaze tracking systems are proposed [2], it is still hard to get accurate gaze estimation when using one single web camera under natural light with free head movement.

The gaze estimation methods are divided into two categories, feature-based methods and appearance-based methods. Feature-based methods typically depend on pupil detection with the light sources' reflections on the cornea. A map from pupil center of geometry model to the gaze calibration points is established by calculating eye movement features. These methods could achieve very high accuracy (error less than 1°), but the function of calibrated regression fluctuate strongly due to free head movement [3]. Instead of focusing on geometric mapping, appearance-based methods treat the cropped eye image as a point representation in high dimensional space and learn the mapping relation from this point in given feature space to screen coordinates. Thus, the gaze estimation of new input eye images can be acquired by regression model trained by all exited image data, therefore, some latent gaze features can be implicitly modeled.

Traditional appearance-based gaze estimation methods were proposed using intensity feature, histogram information or gradient information of the whole eye image as input data. Recently, there have been a lot of novel approaches to appearance description. Lu et al. [4] introduced Grid feature, which is a 15-dimension feature calculating the sum ratio of pixel intensity in separated blocks. It has a high computational complexity combining with adaptive linear regression. But the method requires a fixed head pose. Building on Lu's work, Mora and Odobez [5] took both left and right eye images into consideration and put forward couple adaptive linear regression to eliminate gaze error between both eyes. Wang et al. [6] proposed a coarse-to-fine gaze estimation with TOP (Topology-preserving) feature, which is a sparse feature selection learning based on dense local feature. The well topological structure is kept in manifold space,which improves the features representative ability of eye image. However, these proposals are not effective with free head movement.

To solve this problem, Mora and Odobez [7] estimated gaze angle with head pose estimation using RGB-D data. In their method, a 3D face model was built upon the multimodal Kinect data, and gaze direction in the head coordinate system was determined by [4] method on the eye appearance. Although their method provides robust and accurate head pose tracking, the overall gaze estimation accuracy is relatively low (around 10°).] Lai et al. [8] took advantage of approximate random forest to build a

* Corresponding author.
*E-mail addresses:* wangyafei@dlmu.edu.cn (Y. Wang), fxp@dlmu.edu.cn (X. Fu).

http://dx.doi.org/10.1016/j.knosys.2016.07.038
0950-7051/

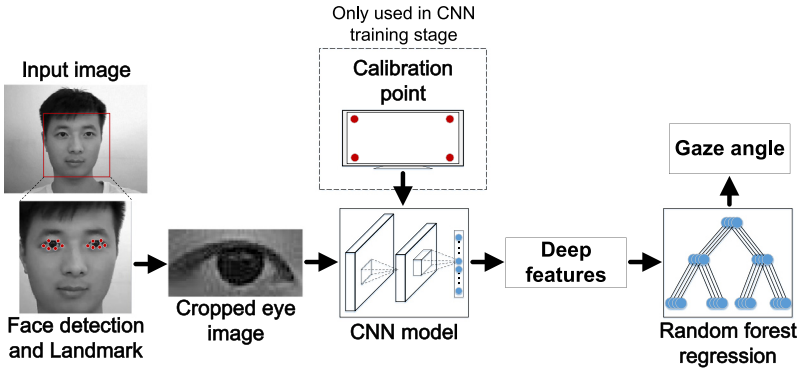


**Fig. 1.** Overview of proposed gaze estimation method.

5-dimension joint head pose and eye gaze space model, which performed well in frontal head motion. But this method relies too much on the accuracy of head pose estimation and needs fine configuration of head pose and eye gaze. Lu and Chen [9] employed auto-encoder to learn a sparse codebook of eye images and used spatial pyramid pooling to get eye features which are sparse coefficients reconstructing on the basis of codebook. This patch-based features improve the gaze estimation accuracy without training by the same person during the different sessions. Zhang et al. [10] took no account of head pose in gaze estimation and realized the appearance-based gaze estimation using CNN immediately to learn the mapping between eye image and gaze coordinate. They concatenated three-dimensional head pose in the last hidden layer with the output of fully connected layer. The change of original LeNet slightly improves the performance of CNN on gaze estimation with free head movement.

We focus on the gaze tracking system using one single web camera under natural light and allowing free head movement. To address this issue, we present a novel gaze estimation method combining deep feature extraction and feature forest regression. Unlike previous appearance-based gaze estimation methods, the proposed method extracts deep features from deep CNN to predict eye gaze direction. The CNN is used to classify eye images into calibrated fixations index through multi-scale convolutions and pooling with last hidden layer as deep features. The deep features have spare representation in characterize image and shown significant improvement on classification method.

Having extracted the deep features, regression forest is used to find the direct correlation between feature space and gaze direction. During forest node splitting, we firstly minimize the squared error loss by cluster and then split the node by binary classification for better partitions in deep feature space. The applying of fine-tuned cluster algorithm and SVM (Support Vector Machine) classification has a positive impact on the error reduction of the child nodes in gaze regression since it averages the predicted gaze cluster. The proposed method works well on eye images dataset under the condition of natural light and free head motion. Deep features regression forest model shows certain robustness to occlusion.

The paper is structured as follows: Section 2 gives the deep feature learning method and random forest regression for deep feature. Experimental results are given in Section 3, while traditional methods are used as a contrast. Additionally, the test result within our images dataset are also analyzed. Finally, Section 4 draws conclusions and gives some directions for future work.

## 2. Proposed method

### 2.1. Overall model

The overall framework proposed is shown in Fig. 1. In order to verify our method on individual images, an offline training and testing eye images dataset is built by cropping eye region from its localization of facial landmarks. First, the subject's face which always appears fully in the field of view is detected using modified version of Viola-Jones method [11]. Once the face region which takes the bounding box (Fig. 1) is localized, next step is to obtain the eye region. In our pipeline, a SDM (Supervised Descent Method) [12] facial landmark detector is used to find the eye corners and other fiducial points. The selected landmarks designating the eye region are shown as red dots in Fig. 1. This algorithm satisfies the eye region localization due to its two important characteristics which have been evaluated on "face in the wild" datasets [12]: (1) it is robust under natural light with different illumination conditions and (2) its outcome is significantly fast and accurate for face align in the wild with large head rotary movement.

Afterwards, all images in which the cropped eye images contain background region are discarded, which happened in about 4% of all cases. In this manner, the images dataset delivers eye images without background noise for the evaluation of appearance-based estimation function.

In this paper, the appearance variation of appearance-based estimation function to inferring gaze is handled in two degrees of freedom, in other words, our gaze estimator outputs two dimension gaze angle vector. General framework of our method contains cascaded stages, deep feature representation and feature regression.

Firstly, learning deep feature for eye gaze appearance is to set up a CNN deep feature space. Based on sparse calibration [6] which means using less training points during gaze calibration, the eye images can be divided into n classes which predict same value with calibration points. The CNN with several convolution layers and max-pooling layers is built to indicate classes by last soft-max layer. Deep feature is learned as the last hidden layer constitute the sparse feature space after extraction cascade of network.

Secondly, training regression forest model for deep feature is utilized to predict gaze direction. While regression forest refers to utilize many decision trees with respective random drawn training samples at leaf node, regression forest benefits by its ensemble voting output. The exert of regression forest seems more like partitions in the given feature space. In order to reduce the error loss, at the split of each node, it is necessary to find clusters of training data at current node and determine its classification of the found clusters. The cluster algorithm preserves that the found clusters are

Please cite this article as: Y. Wang et al., Appearance-based gaze estimation using deep features and random forest regression, Knowledge-Based Systems (2016), http://dx.doi.org/10.1016/j.knosys.2016.07.038

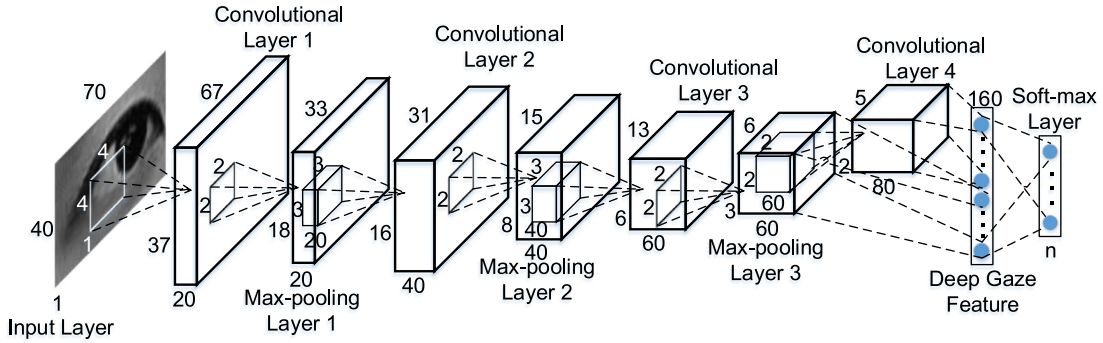


**Fig. 2.** The structure for deep feature extraction.

used before node splitting. Then the employ of SVM on the existing clusters cast the node splitting problem into a classification problem. In this case, new eye gaze feature will be transferred to its trees leaf node which is assigned gaze label by classification.

### 2.2. Deep feature learning

The task of the CNN is to learn deep feature from the input eye images. Inspired by DeepID [13], our model uses the similar architecture containing three convolutional layers with max-pooling layer, one single max-pooling layer, one hidden layer and one soft-max layer for training classification. Fig. 2 shows the detailed structure of CNN. The input of the network is gray eye image with a fixed size of 40 × 70 (*height* × *width*). The last soft-max layer indicates numbers of classes equal to numbers of calibration points. The feature filter size for the convolutional layer is 4 × 4, 3 × 3, 3 × 3, 2 × 2 pixels respectively, while at the same time the number of feature filters is 20, 40, 60, 80. The convolutional layer is defined as $out^j = \max(0, b^j + \sum_i k^{ij} * in^i)$, where $*$ is the convolution operator, $in^i$ is the $i$th input feature maps, $out^j$ is the $j$th output feature maps, $b^j$ is the $j$th bias and $k^{ij}$ is the convolution kernel. This layer combines features linearly followed by ReLU (rectified linear units) for neurons. For all max-pooling layers, the feature size is 2 × 2. The max-pooling layer can be denoted as $out^j_{p,q} = \max_{0 \le m,n < s} \{in^j_{p \times s+m, q \times s+n}\}$, where $in^j$ is the $j$th input feature map and $out^j$ is its output feature map pools over each non-overlap $s \times s$ region, $p$ and $q$ are the index of pooling region which also indicate the row and column of $out^j$, $m$ and $n$ indicate the numbers belonging to the pooling region.

In the last hidden layer which is fully connected to the fourth convolutional layer and the third max-pooling layer, a 160-dimension deep feature for gaze appearance is extracted. The deep feature learns discriminative ability with supervised task of calibration classification by minimize the cross-entropy loss.

Fig. 3 shows the output of layers in CNN when the testing image goes through the trained CNN. In the output of first and second max-pooling layer, outline of eye image can be seen. But the next two layers just exhibit some active network units. The CNN model active different neuron unit at the same layer for different eye image input. In this paper, the fourth convolutional layer and the third max-pooling layer are used to present eye features.

### 2.3. Random forest regression

Random forest is an ensemble of decision trees [14] and has been widely used in many applications, such as head pose estimation [15–17], feature selection [18] and recommender system [19]. Recently, it was reported to outperform other appearance-based gaze estimation methods in [20]. They clustered training eye

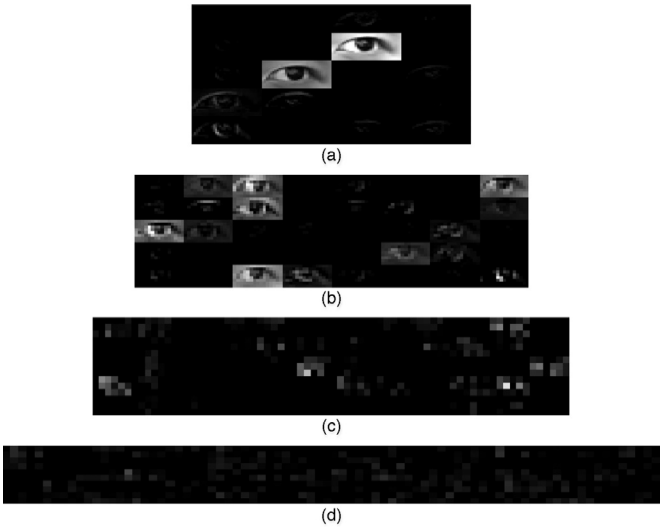


**Fig. 3.** Output of different layers in CNN. (a): output of the convolutional layer 1 and max-pooling layer 1 (20 × 33 × 18), 4 × 5 feature maps size of 33 × 18, (b): output of the convolutional layer 2 and max-pooling layer 2 (40 × 15 × 8), 8 × 5 feature maps size of 15 × 8, (c): output of the convolutional layer 3 and max-pooling layer 3 (60 × 6 × 3), 12 × 5 feature maps size of 6 × 3, (d): output of the convolutional layer 4 (80 × 5 × 2), 16 × 5 feature maps size of 5 × 2.

images dataset according to head pose angles, and queried testing eye images to their nearest clusters. In addition, it shows that random forest has a good stability and generalization on person-independent gaze estimation by averaging these predicted gaze cluster. It is crucial to the performance of the forest that the diversity between the trees is caused by randomness, in which sampling on the aggregate training set and generating the node splitting candidates are random.

A cluster-to-classify random forest is a variance of the ordinary decision trees with the modified splitting criterion that minimizes empirical loss at each node. Typical examples can be seen in [21]. Single decision tree is grown in a recursive fashion by partitioning the training samples into subsets successively [18]. At each recursive step, the samples in a node are split into several subsets (child nodes) based on the feature classification that could minimize the square errors of the child nodes by cluster. The splitting process will come to an end when the impurity of the node is lower than a threshold, after which the leaf node will be assigned a regression label. Random forest operates outputting regression label by a multitude of decision trees voting to share collaborative decision-making [22].

The whole training data set is denoted as $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$, where $N$ is the calibration points number, $\mathbf{x}_i$ is the $i$th deep feature vector in feature space, and $\mathbf{y}_i$ is the $i$th two-dimensional gaze vector in Euler angle space. The forest contains $M$ tree $\mathbf{T} =$

$\{\mathbf{T}_1, \mathbf{T}_2, \ldots, \mathbf{T}_M\}$. And each tree $\mathbf{T}_m$ is trained by different bootstrap samples [23] $\{\mathbf{X}_m, \mathbf{Y}_m\}$ from the original training data set.

The task of forest regression is to build a random forest learning a mapping for regression (Algorithm 1). Child node split in standard random forest operators at single dimension threshold which cannot get function with minimizing squared error loss. Thus, the rules of node splitting is modified as [21]. But the cluster algorithm in this method is [24] instead of $k$-means clustering. Because this cluster algorithm finds cluster centers with higher density than their neighbors during clustering procedure in which the outliers are assignated according to distance measure in embedded space. The number of clusters arises significantly in sparse feature space.

**Algorithm 1** Cluster-to-classify random forest

**Input:** $N$ samples of feature and label pair $\{\mathbf{X}_N, \mathbf{Y}_N\} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$
**Output:** Regression forest
**INITIALIZATION**
set *rdrawn*, *mtree*
**PROCESS**
**for** $j = 1$ to *mtree* **do**
  random drawn $rdrawn \times N$ training samples $\{\mathbf{X}_m, \mathbf{Y}_m\} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^m$
  $node = 1$
  **while** the node is impure **do**
    **CLUSTER**
    compute feature local density $\rho_m$ and distance measure $\delta_m$
    recognize cluster centers by sort $\rho_m > \min(\rho)$ and $\delta_m > \min(\delta)$
    get clusters $\{\mathbf{C}_1, \mathbf{C}_2, \ldots, \mathbf{C}_K\}$ by remaining feature points assignation
    **CLASSIFY**
    compute $f = \min_{\mathbf{w}_k} \|\mathbf{w}_k\|_2 + Z \sum_{i=1}^{m} (\max(0, 1 - d_i^k \mathbf{w}_k^T \mathbf{x}_i))^2$, $\mathbf{x}_i \in \mathbf{C}_k$
    node splits using $f$
    $node++$
  **end while**
**end for**

Before each node splitting, local density of each feature vector is calculated by $\rho_m = \Sigma_n \chi(d_{mn} - d_c)$, where $\chi(x) = 1$ if $x < 0$, otherwise $x = 0$, and $d_c$ denote a cut off distance with Gaussian kernel. Minimum distance between the point $m$ and any other point with higher density is measured by $\delta_m = \min_{n:\rho_m > \rho_n} (d_{mn})$. Then, the cluster centers is identified based on fast search by sorting candidates whose $\rho_m > \min(\rho)$ and $\delta_m > \min(\delta)$. After the cluster centers have been found, cluster $\{\mathbf{C}_1, \mathbf{C}_2, \ldots, \mathbf{C}_K\}$ is completed by remaining feature points assignation to the same cluster as its nearest neighbor of higher density. The clusters minimize the error loss function $L = \min_{\mathbf{C}} \sum_{k=1}^{K} \sum_{\mathbf{x}_i \in \mathbf{C}_K} \|\mathbf{y}_i - \mathbf{a}_k\|_2^2$, where $\mathbf{a}_k$ is the constant estimate computed as the mean of training samples $\mathbf{a}_k = \frac{1}{|\mathbf{C}_K|} \sum_{i \in \mathbf{C}_K} \mathbf{y}_i$. where $\mathbf{a}_k$ is the constant estimates computed as the mean of training samples $\mathbf{a}_k = \frac{1}{|\mathbf{C}_K|} \sum_{i \in \mathbf{C}_K} \mathbf{y}_i$.

At the node splitting stage, the task casts into a $K$-class classification problem to preserve the existing clusters. Based on the training samples, the index of clusters is determined. The node weight vector which determines whether the data belongs to left child node or right child node is solved by SVM optimization, $\min_{\mathbf{w}_k} \|\mathbf{w}_k\|_2 + Z \sum_{i=1}^{m} (\max(0, 1 - d_i^k \mathbf{w}_k^T \mathbf{x}_i))^2$. where, $\mathbf{w}_k$ is the $k$th cluster weight vector, $Z$ is the penalty parameter. If $\mathbf{x}_i \in \mathbf{C}_k$, $d_i^k = 1$, otherwise, $d_i^k = -1$. With the weight vector, the child node to forward is the one with maximum product value on training samples (Algorithm 2).

**Algorithm 2** predict based on random forest

**Input:** testing sample of feature $\mathbf{x}_{test}$, regression forest $\mathbf{T} = \{\mathbf{T}_1, \mathbf{T}_2, \ldots, \mathbf{T}_m\}$
**Output:** predict label $\mathbf{y}_{pre}$
**for** $i = 1$ to $m$ **do**
  load parameters of $\mathbf{T}_i$
  $node = 1$
  **while** current node is not leaf node **do**
    read splitting attribute of current node $\mathbf{w}_k$
    $K = \max_k (\mathbf{w}_k^T \mathbf{x}_{test})^2$
    forward to its $K$th child node
  **end while**
  get regression value $\mathbf{y}_{pre}^i$ of current tree $\mathbf{T}_i$
**end for**
$\mathbf{y}_{pre} = \sum_{i=1}^{m} \mathbf{y}_{pre}^i / m$

This kind of node splitting strategy makes non-leaf node have more than two child nodes when $K$ is larger than two. Thus the number of clusters is more than two. But in our work, the value of $K$ is two since the larger $K$ value does not improve the estimation results significantly.

To create leaf node for each tree when reaching maximum number of training samples belonging to that node. During the prediction of random forest regression, the outputs from each tree contribute to the final estimation result by voting.

## 3. Experimental results

### 3.1. Dataset and experimental setup

Our experimental images are captured from a normal USB camera fixed on a desktop computer. The image resolution is 640 by 480. The dataset contains the images collected from different times, most of which are under the normal natural light, while a small number of which are under the condition of straight strong sun light at noon or weak screen light at night. Any other auxiliary light sources are not added. Totally 22 groups of videos from 6 subjects (see Fig. 4) under different light illumination conditions were collected, including 16 groups with subjects wearing glasses. Each group contains data from twice calibration steps, one without head movement and the other one with free head movement. In the 41 calibration points on the screen, the former 25 points are for training and the later 16 points are for testing. The calibration points and corresponding gaze angles are shown in Fig. 5(a). There are 107,681 available images after the images without eyes or with closing eyes have been removed.

Fig. 5(b) illustrates the distribution of head pose angle in the second calibration with free head movement. The head pose was calculated using Pose from Orthography and Scaling with ITerations (POSIT) where a statistical anthropometric 3D rigid model is used as an approximation of the human head, combined with SDM for facial features extraction. The values of pitch angles are almost positive which concentrate on the interval between 0° and 15°, whereas the values of yaw angles evenly varies from −10° to 10°. The red area demonstrates that range of yaw angle $[-2°, 1°]$ and pitch angle [12°, 14°] with higher frequency of occurrence. While other values of angles do not appear much in the rest of area except the yellow areas.

Similar to Ref. [10], mean grayscale intensity of images is used to represent the light conditions and calculate the histogram of mean grayscale intensity as shown in Fig. 5(c). The eye images under a darker light condition has a lower mean grayscale intensity, on the contrary, the eye images under a brighter light condition has a higher one. The largest percentage of total number of

Please cite this article as: Y. Wang et al., Appearance-based gaze estimation using deep features and random forest regression, Knowledge-Based Systems (2016), http://dx.doi.org/10.1016/j.knosys.2016.07.038


**Fig. 4.** The face images of gaze calibration under different light illumination conditions.

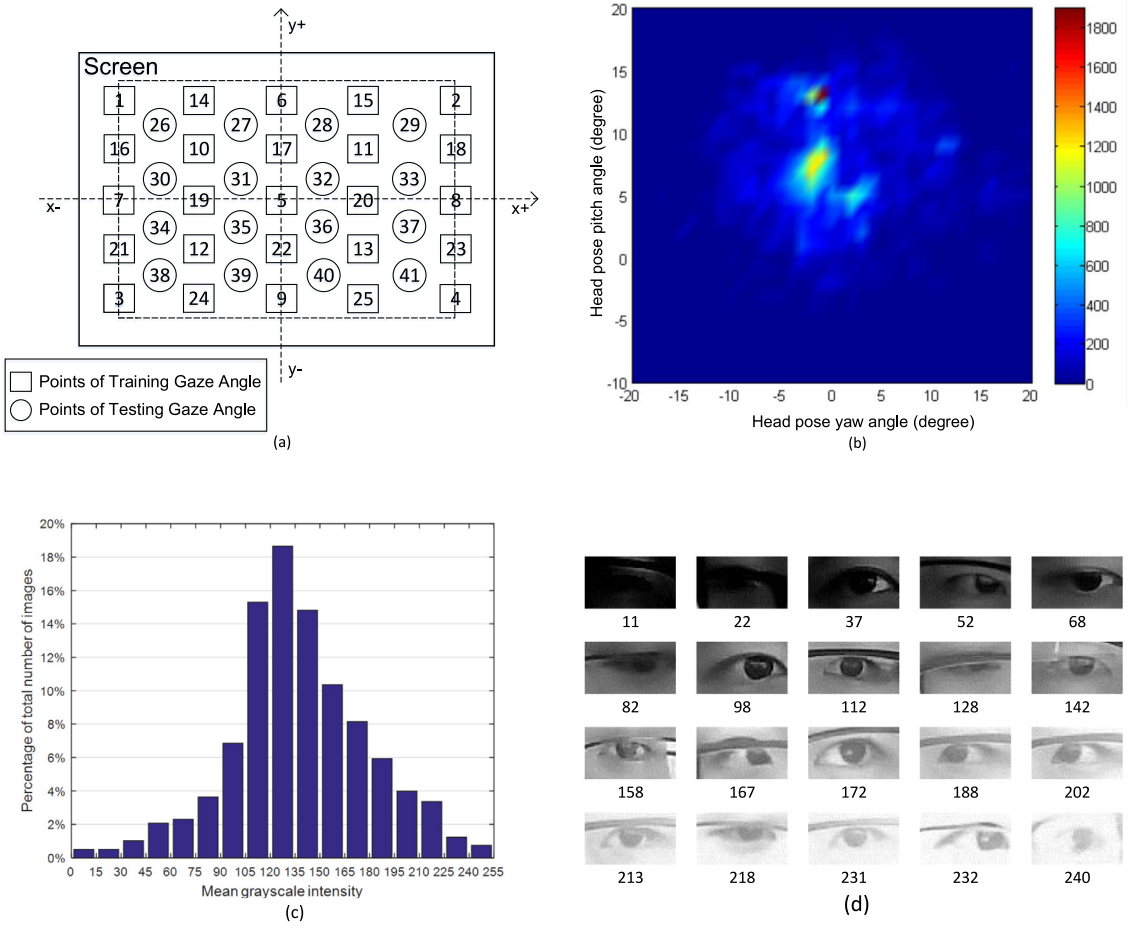


**Fig. 5.** (a): Calibration points of training and testing gaze angle in coordinates on the screen. (b): Distribution of head pose angle (yaw and pitch) for the eye images dataset. (c): Percentages on image mean grayscale intensity of the eye images dataset. (d): Example images with their mean grayscale intensity values in the bottom.

eye images reaches to 18.646% with mean grayscale intensity between 120 and 135. In total, the percentage of the top three mean grayscale intensity sums up to 48.753%. These results underline the distribution of our dataset in terms of appearance variations. Fig. 5(d) gives some examples of eye images, which can be easily seen that more than half of the eye images in the dataset are in the cases of subjects wearing eye glasses.

### 3.2. Deep gaze feature

The deep CNN was implemented based on theano [25]. Corresponding code was running on a PC with Intel Core i7-4790 3.60 GHz CPU, 16 GB RAM and GTX980Ti graphics card. The CNN deep feature layer data can be shown as Fig. 6 after the training error convergence. As can be seen, these deep features are sparse with

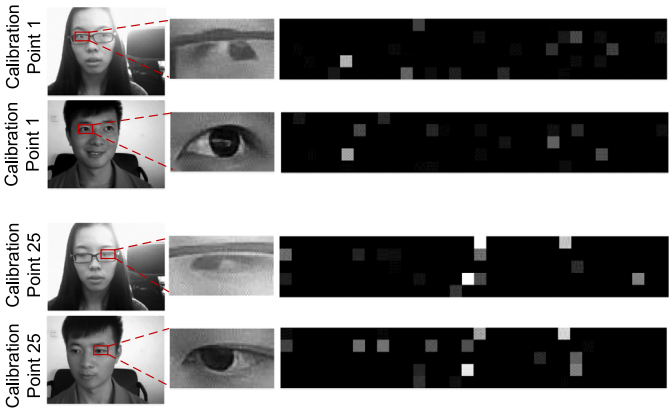


**Fig. 6.** Examples of learned deep features. The left column shows two different subject face images and eye detection results for calibration point 1 and calibration point 25. The mid column shows the corresponding eye images. The right column shows their corresponding deep features. For the convenience of illustration, the feature is reshaping to 5 by 32, in which brighter squares indicate higher values, while the black squares indicate zero values.

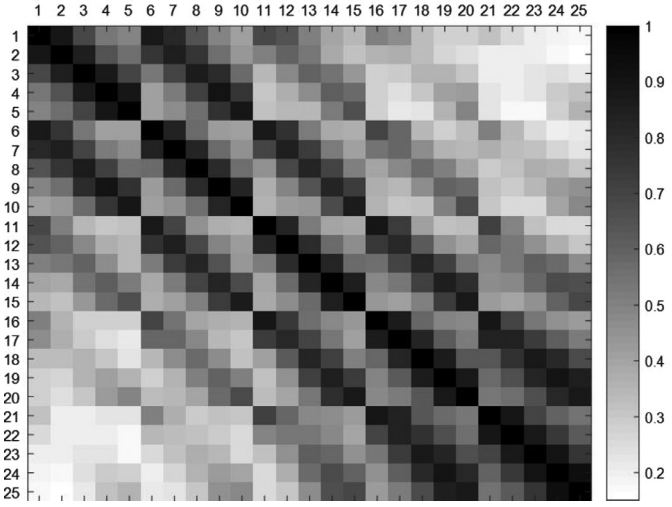


**Fig. 7.** The cosine similarity confusion matrix of mean deep features of 25 calibration points.

non-negative neurons in CNN. In order to measure the similarity between two deep features, cosine similarity is used to evaluate them in positive space. Cosine similarity is denoted as $\cos(\theta) = \frac{\mathbf{a}\cdot\mathbf{b}}{|\mathbf{a}||\mathbf{b}|}$, where **a** and **b** are 160-dimension features. The resulting similarity ranges from 0 to 1. Its outcome reflects the cosine value of the angle between feature vectors. Two feature vectors with the almost same orientation have a cosine similarity close to 1.

In Fig. 6 similarities between deep features from same calibration point (calibration point 1 and calibration point 25) are 0.91 and 0.93. Fig. 7 shows the cosine similarity of mean deep features of 25 calibration points, color-coded from white (minimum) to black (maximum). Refer to Fig. 5(a), the mean feature value of calibration point is quite similar with the mean feature values of its neighbor calibration points. For example, the cosine similarity of the calibration point 1 and its neighbor points 2,6,7 are 0.89, 0.87 and 0.80 respectively. Then, we make a comparison between the results of this deep feature and other features on regression approaches.

**Table 1**
Mean absolute error (degree) of gaze angle on different regression approaches and different features.

| | PLS | RF | SVR [28] | RVR [28] |
|---|---|---|---|---|
| Grid | 6.2059 | 6.2592 | 7.3809 | 6.4781 |
| Raw data | 5.8275 | 5.4804 | 6.4867 | 5.978 |
| Gabor | 4.7985 | 4.6567 | 5.0077 | 4.954 |
| HOG | 3.6182 | 3.483 | 4.1341 | 3.6845 |
| LBP | 5.0491 | 5.1308 | 5.2998 | 5.1915 |
| LBP+HOG | 4.6304 | 4.1888 | 4.3078 | 4.2777 |
| Deep feature 160 | 2.8301 | 2.5526 | 3.1346 | 2.9383 |

**Table 2**
Single image testing computational time cost (ms) of feature extraction methods and feature regression methods for appearance-based gaze estimation.

| Feature extraction | Time cost | Feature regression | Time cost |
|---|---|---|---|
| Grid | 0.3 | PLS | 34 |
| Raw data | 1 | RF | 25 |
| Gabor | 38 | SVR [28] | 8 |
| HOG | 6 | RVR [28] | 7 |
| LBP | 59 | ALR [4] | 6 |
| LBP+HOG | 66 | MLD [30] | 5 |
| Deep feature with CPU | 29 | our method | 37 |
| Deep feature with GPU | 1 | | |

### 3.3. Comparison with other methods

In order to testify the performance of deep feature with other local or global features, several features are used, including Grid, raw data, HOG (Histogram of Oriented Gradient), LBP (Local Binary Pattern), Gabor, LBP+HOG. Besides, few regression methods are employed, such as PLS (Partial Least Squares), RF (standard Regression Forest), SVR (Support Vector Regression), and RVR (Relevance Vector Regression). Table 1 compares the mean absolute error of gaze angle on these regression approaches and different features. Grid feature introduced in [4] is a 15-dimension feature. LBP+HOG feature extracts HOG feature on LBP feature map. The PLS regression [26] finds the most relevant principle component of training feature to represent training label. LIBSVM [27] was applied in the SVM classification in our method and the SVR regression method [28].

The training dataset contains eye images of the former 25 gaze points of all subjects, whose corresponding two dimensional gaze angles are defined as training labels. Thus, the eye images data of the later 16 gaze points is tested using feature extraction and feature regression. Estimation error is the common Euler angle difference of prediction gaze angles and labeled gaze angles. The results show that the Grid data feature performs the worst among all regression methods. All features were performed dimensionality reduction using Synchronized Delaunay Submanifold Embedding [29] except Grid feature who has only 15 dimensions and deep feature. Compared to Grid, Gabor, LBP and LBP+HOG, HOG feature has a better performance. While 160-dimension deep feature performs better than others among every regression method. The actual computational time of each method includes the time of feature extraction and feature regression (see Table 2). In this table, the computational time of our method is relatively respectable especially when using GPU. Corresponding codes were running on the same PC implemented deep CNN with Matlab r2015a.

Then, our method is compared to other appearance-based gaze estimation methods. In Section 2.2, the extracted deep features are 160 dimension. To testify the influence of the dimension value to the results, different dimensional deep features are extracted to regress, including 50, 80, 110, 210 and 260 dimensional features. The mean absolute error of gaze angle on these different methods is presented in Table 3. The ALR gaze

Please cite this article as: Y. Wang et al., Appearance-based gaze estimation using deep features and random forest regression, Knowledge-Based Systems (2016), http://dx.doi.org/10.1016/j.knosys.2016.07.038

**Table 3**
Mean absolute error (degree) of gaze angle on appearance-based gaze estimation methods.

| Method | Error |
|---|---|
| ALR [4] | 6.7177 |
| CALR [8] | 6.4624 |
| TOP [6] + ALR [4] | 5.9044 |
| MLD [30] | 4.7528 |
| Our method - Deep feature 50 | 2.5732 |
| Our method - Deep feature 80 | 2.4417 |
| Our method - Deep feature 110 | 1.9608 |
| Our method - Deep feature 210 | 2.3652 |
| Our method - Deep feature 260 | 2.1571 |
| Our method - Deep feature 160 | 1.5311 |

estimation method in [4] was realized via convex programming [31]. Based on ALR, the CALR [8] is solved by Second-Order Cone Programming using both left and right eye images. The TOP+ALR replaced the Grid feature as Topology-preserving in [6].MLD was applied on head pose estimation in [30]. It defined head pose angle as a soft label, built the Gaussian distribution of exist head pose and learned the mapping between image and its distribution. In our work, MLD is ultilized in gaze estimation. The maximum value of the distribution of input eye image is the coarse gaze angle. And the final gaze angle is the linear interpolate of Delaunay triangulation area of the coarse gaze angle.

Table 3 shows that CALR and TOP have improved the accuracy of ALR. But, ALR-based methods are easily affected by large head rotary motion. Besides, eyeglasses frames in images also have great impact on the gaze estimation accuracy. According to Tables 1 and 3, out method with 160-dimension deep features shows the best accuracy on our images dataset.

Below, the test result with-in our images dataset is analyzed detailly from four perspectives, illumination condition, test points, person-independent and occlusion.

### 3.4. Comparison on illumination condition

In order to deal with gaze estimation under natural light, eye images under different illumination conditions are collected. Inspired by Ref. [10], image mean grayscale intensity value can be used to represent the light conditions. The distribution of mean absolute errors of gaze angle on testing eye images dataset under different illumination conditions is shown in Fig. 8. The red points and curves represent the errors of training data, while the blue ones represent the errors of testing data. This clearly illustrates a generally consistent trend of the errors on X-axis and Y-axis. Larger errors mainly distribute in the part with mean grayscale intensity value greater than 225 or smaller than 30 due to the slight differences among grayscale of the images. Considering the statistical result of mean grayscale intensity, which is shown in Fig. 5(c), it can

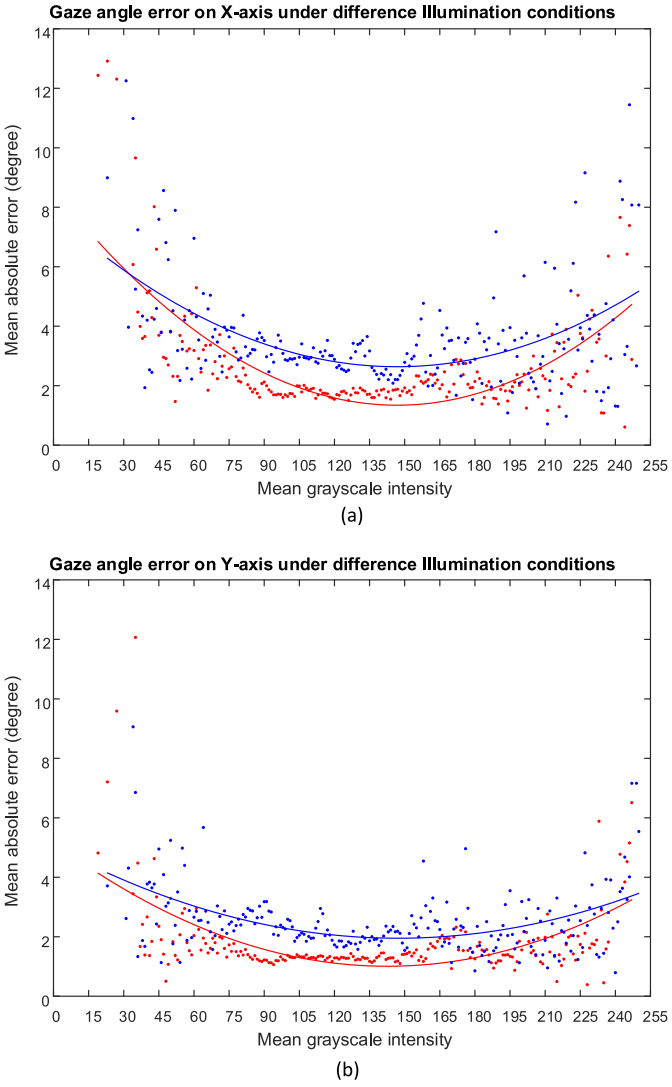


**Fig. 8.** Mean absolute error of gaze angle on different mean grayscale intensity values. (a): Gaze angle errors on X-axis, (b): Gaze angle errors on Y-axis. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

**Table 4**
Mean absolute error (degree) of gaze angle on test points using different calibration points.

| Points | C4 (X-axis, Y-axis) | C5 (X-axis, Y-axis) | C9 (X-axis, Y-axis) | C13 (X-axis, Y-axis) | C25 (X-axis, Y-axis) |
|---|---|---|---|---|---|
| 26 | (4.6809, 2.7276) | (4.2409, 2.2277) | (3.4612, 1.9015) | (2.5322, 1.6191) | (2.0846, 1.4011) |
| 27 | (10.4046, 2.7908) | (6.3677, 2.3641) | (3.1141, 1.8222) | (1.8951, 1.498) | (1.3348, 1.2842) |
| 28 | (11.3960, 2.7336) | (6.4720, 2.3198) | (3.2568, 1.8282) | (1.909, 1.7297) | (1.2915, 1.3231) |
| 29 | (5.8651, 2.4634) | (4.5345, 2.0528) | (3.615, 1.8472) | (2.2915, 1.8500) | (1.7847, 1.3924) |
| 30 | (4.5268, 5.7974) | (5.5180, 3.9696) | (2.6780, 1.6684) | (2.2168, 1.0943) | (2.0181, 0.9394) |
| 31 | (9.4998, 5.7234) | (4.6891, 2.6347) | (2.4385, 1.5922) | (1.6828, 1.0328) | (1.4897, 0.9697) |
| 32 | (11.6149, 5.4943) | (4.8957, 2.0475) | (3.1370, 1.8291) | (1.7545, 1.2850) | (1.2360, 1.0162) |
| 33 | (7.1082, 5.3794) | (6.1302, 3.0323) | (3.5266, 1.8280) | (2.1538, 1.3771) | (1.8672, 1.0558) |
| 34 | (4.6332, 5.8973) | (5.6973, 4.1173) | (2.9306, 1.8211) | (3.7503, 2.0703) | (2.0465, 1.0240) |
| 35 | (9.4257, 5.9042) | (4.528, 2.9106) | (2.5960, 1.6886) | (1.8096, 1.0128) | (1.4134, 1.0088) |
| 36 | (12.3238, 5.9565) | (5.2727, 2.4709) | (3.2757, 1.9662) | (1.9056, 1.2452) | (1.1931, 1.0283) |
| 37 | (8.4449, 6.2382) | (7.1113, 3.5201) | (3.7503, 2.0703) | (2.2311, 1.4825) | (1.8873, 1.052) |
| 38 | (4.8797, 2.9458) | (4.5314, 2.5147) | (3.8198, 2.0974) | (2.9114, 1.6825) | (2.1163, 1.3795) |
| 39 | (10.3998, 3.3364) | (6.4832, 2.9307) | (3.4057, 1.9540) | (2.3068, 1.4396) | (1.3407, 1.2151) |
| 40 | (12.4866, 3.7755) | (7.6761, 3.374) | (3.5524, 2.1944) | (2.0900, 1.5999) | (1.2459, 1.2637) |
| 41 | (7.5903, 3.6625) | (6.1520, 3.0117) | (4.0374, 2.3491) | (2.4371, 1.8942) | (1.8226, 1.3542) |

Please cite this article as: Y. Wang et al., Appearance-based gaze estimation using deep features and random forest regression, Knowledge-Based Systems (2016), http://dx.doi.org/10.1016/j.knosys.2016.07.038

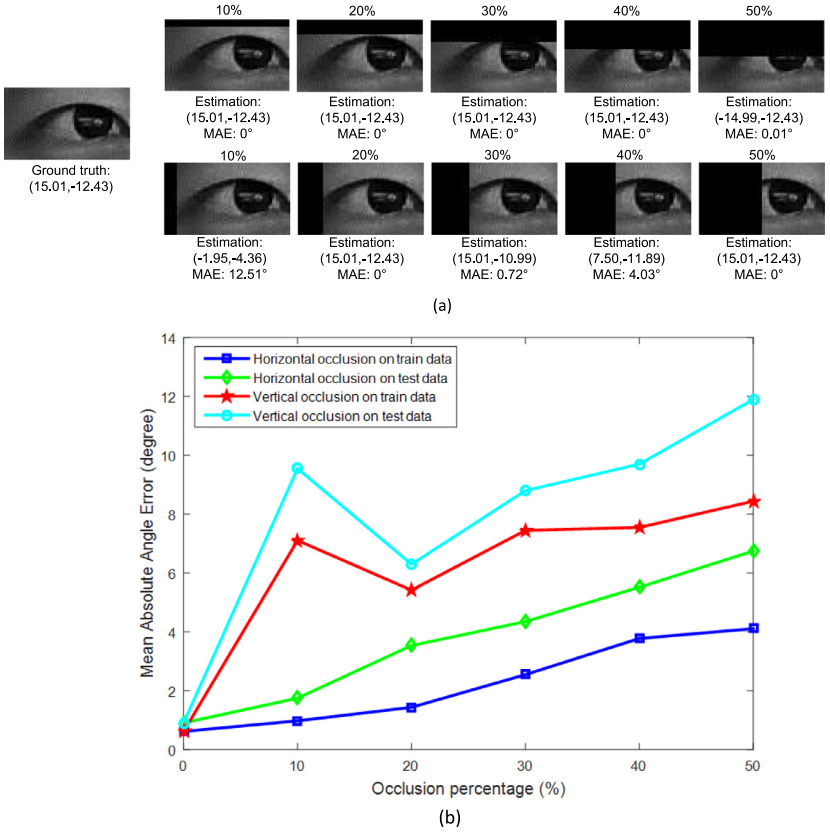


**Fig. 9.** (a): Eye image occlusion examples on the training data. The Ground truth value or estimation value, mean absolute error is marked below the corresponding eye image. The first row is the vertical occlusion, and the other rows are horizontal occlusion. (b): Results of occlusion on training data and testing data.

be concluded that larger amount of data appear in the condition of lower errors. Although the errors on testing set is more than training set, the difference is less than 1°. Thus, our method shows a better performance under the illumination conditions with mean grayscale intensity value between 120 to 165.

### 3.5. Comparison on test points

The data with 25 gaze points is employed for calibration and training while the data with 16 gaze points for test in previous comparison which means that every test gaze point makes sense. To analyze the effect of using gaze calibration points, we compared the mean absolute error of gaze angle with 4 points (C4), 5 points (C5), 9 points (C9) and 13 points (C13). The index of training gaze calibration points and 16 test points is shown in Fig. 5(a). Table 4 lists the mean absolute error on X-axis and Y-axis of gaze angle on 16 test points using different gaze calibration points. With the increase of the numbers of training points, the corresponding error is minimized both on X-axis and Y-axis. There is an obvious decrease between the errors of C4 and C5, especially the results of test point 32 and 36, which show that the mid calibration point has a significant effect to the accuracy of our results. Besides, considering the complexity of calibration, 13-point calibration is better than 25-point since the similar results.

### 3.6. Comparison on person-independent

To discuss our method on person-independent or cross subject test, 25-calibration training eye image data from one subject with free head movement is used to train the CNN model and random forest regression model. And the data of other subjects is tested on this model. Detailed results of the cross subject estimation are presented in Table 5. Table 5 illustrates the confusion matrix of mean absolute errors. A-F represents the subject A-F (see Fig. 4). Due to conditions of low resolution, free head movement and natural light, cross subject error on appearance-based gaze estimation method will be large. Ref. [5] also did the similar person-independent test in the frontal session with error larger than 14°. Compared to their results, our method is more effective with no more than 8° mean error.

**Table 5**
Mean absolute error (degree) of gaze angle for cross subject gaze estimation.

| Test subject | Used subject model | | | | | |
|---|---|---|---|---|---|---|
| | A | B | C | D | E | F |
| A | – | 5.0502 | 5.2659 | 6.7537 | 6.994 | 5.6844 |
| B | 5.6663 | – | 5.5281 | 5.0247 | 7.6889 | 5.2169 |
| C | 5.2234 | 5.3566 | – | 6.2273 | 6.3632 | 4.3095 |
| D | 6.3686 | 5.5607 | 7.3944 | – | 7.2319 | 5.9687 |
| E | 7.101 | 6.8342 | 7.3 | 6.8349 | – | 7.7804 |
| F | 7.0154 | 5.573 | 5.3218 | 6.2588 | 7.3886 | – |

### 3.7. Comparison on occlusion

Finally, as many applications of CNN have indicated that the CNN model has certain robustness to occlusion, the influence of occlusion is evaluated by picking original eye images on a certain percentage occupation using training data and testing data of subjects. Fig. 9 illustrates results that 10% to 50% areas were occluded, respectively. The horizontal occlusion seems to have no influence to estimated results until it reaches 50%. Conversely, the results of vertical occlusion are not stable.

Please cite this article as: Y. Wang et al., Appearance-based gaze estimation using deep features and random forest regression, Knowledge-Based Systems (2016), http://dx.doi.org/10.1016/j.knosys.2016.07.038

## 4. Conclusion

In this work, a CNN-based feature regression method is present for gaze estimation under natural light. The CNN built a sparse active feature space from the training eye images dataset. The gaze estimation model learned mapping between deep feature and gaze coordinate by random forest regression. Unlike the traditional node splitting when training the random forest, cluster algorithm and SVM classification are utilized in node splitting to minimize the squared error loss. The experimental results show that deep feature significantly improves the performance on different regression method among other local features. And our method which is the combination of deep feature and modified random forest regression, also achieved better performance compared with other appearance based gaze estimation methods. We hope to apply our method to real-time driver gaze tracking and have a further study on gaze estimation in real driving environment with background noise.

## Acknowledgments

This research has been supported by the National Natural Science Foundation of China under grant No.61272368, No.61370142, the Fundamental Research Funds for the Central Universities (3132016352) and the Fundamental Research of Ministry of Transport of P.R. China (2015329225300). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

## References

[1] C. Conati, C. Merten, Eye-tracking for user modeling in exploratory learning environments: an empirical evaluation, Knowl.-Based Syst. 20 (6) (2007) 557–574.
[2] D.W. Hansen, Q. Ji, In the eye of the beholder: a survey of models for eyes and gaze, IEEE Trans. Pattern Anal. Mach.Intell. 32 (3) (2010) 478–500.
[3] X. Fu, Y. Zang, H. Liu, A real-time video-based eye tracking approach for driver attention study, Comput. Inf. 31 (4) (2012) 805–825.
[4] F. Lu, Y. Sugano, T. Okabe, Y. Sato, Adaptive linear regression for appearance-based gaze estimation, IEEE Trans. Pattern Anal. Mach. Intell. 36 (10) (2014) 2033–2046.
[5] K.A.F. Mora, J.-M. Odobez, Person independent 3D gaze estimation from remote RGB-D cameras, in: Proceedings of the 2013 20th IEEE International Conference on Image Processing (ICIP), IEEE, 2013, pp. 2787–2791.
[6] X. Wang, K. Xue, D. Nam, J. Han, H. Wang, Hierarchical gaze estimation based on adaptive feature learning, in: Proceedings of the 2014 IEEE International Conference on Image Processing (ICIP), IEEE, 2014, pp. 3347–3351.
[7] K.A.F. Mora, J.-M. Odobez, Gaze estimation from multimodal kinect data, in: Proceedings of the 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), IEEE, 2012, pp. 25–30.
[8] C.-C. Lai, Y.-T. Chen, K.-W. Chen, S.-C. Chen, S.-W. Shih, Y.-P. Hung, Appearance-based gaze tracking with free head movement, in: Proceedings of the 2014 22nd International Conference on Pattern Recognition (ICPR), IEEE, 2014, pp. 1869–1873.
[9] F. Lu, X. Chen, Person-independent eye gaze prediction from eye images using patch-based features, Neurocomputing 182 (2016) 10–17.
[10] X. Zhang, Y. Sugano, M. Fritz, A. Bulling, Appearance-based gaze estimation in the wild, in: Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 4511–4520.
[11] P. Viola, M. Jones, Rapid object detection using a boosted cascade of simple features, in: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), vol. 1, IEEE, 2001, pp. I–511.
[12] X. Xiong, F. Torre, Supervised descent method and its applications to face alignment, in: Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013, pp. 532–539.
[13] Y. Sun, Y. Chen, X. Wang, X. Tang, Deep learning face representation by joint identification-verification, in: Advances in Neural Information Processing Systems (NIPS), 2014, pp. 1988–1996.
[14] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.
[15] G. Fanelli, J. Gall, L. Van Gool, Real time head pose estimation with random regression forests, in: Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2011, pp. 617–624.
[16] Y. Li, S. Wang, X. Ding, Person-independent head pose estimation based on random forest regression, in: Proceedings of the 2010 17th IEEE International Conference on Image Processing (ICIP), IEEE, 2010, pp. 1521–1524.
[17] M. Dantone, J. Gall, G. Fanelli, L. Van Gool, Real-time facial feature detection using conditional regression forests, in: Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2012, pp. 2578–2585.
[18] Q. Zhou, H. Zhou, T. Li, Cost-sensitive feature selection using random forest: selecting low-cost subsets of informative features, Knowl.-Based Syst. 95 (2016) 1–11.
[19] H.-R. Zhang, F. Min, Three-way recommender systems based on random forests, Knowl.-Based Syst. 91 (2016) 275–286.
[20] Y. Sugano, Y. Matsushita, Y. Sato, Learning-by-synthesis for appearance-based 3D gaze estimation, in: Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 1821–1828.
[21] K. Hara, R. Chellappa, Growing regression forests by classification: applications to object pose estimation, in: Computer Vision–ECCV 2014, Springer, 2014, pp. 552–567.
[22] Q. Long, A flow-based three-dimensional collaborative decision-making model for supply-chain networks, Knowl.-Based Syst. 97 (2016) 101–110.
[23] A. Liaw, M. Wiener, Classification and regression by random forest, R News 2 (3) (2002) 18–22.
[24] A. Rodriguez, A. Laio, Clustering by fast search and find of density peaks, Science 344 (6191) (2014) 1492–1496.
[25] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. Goodfellow, A. Bergeron, N. Bouchard, D. Warde-Farley, Y. Bengio, Theano: new features and speed improvements, Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop, 2012.
[26] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer, 2009 Second Edition.
[27] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, C.-J. Lin, Liblinear: a library for large linear classification, J. Mach. Learn. Res. 9 (2008) 1871–1874.
[28] F. Martinez, A. Carbone, E. Pissaloux, Gaze estimation using local features and non-linear regression, in: Proceedings of the 2012 19th IEEE International Conference on Image Processing (ICIP), IEEE, 2012, pp. 1961–1964.
[29] T. Schneider, B. Schauerte, R. Stiefelhagen, Manifold alignment for person independent appearance-based gaze estimation., in: Proceedings of the 2014 22nd International Conference on Pattern Recognition (ICPR), 2014, pp. 1167–1172.
[30] X. Geng, Y. Xia, Head pose estimation based on multivariate label distribution, in: Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 1837–1842.
[31] E. Candes, J. Romberg, l1-magic: recovery of sparse signals via convex programming, URL: www.acm.caltech.edu/l1magic/downloads/l1magic.pdf 4(2005) 14.

Please cite this article as: Y. Wang et al., Appearance-based gaze estimation using deep features and random forest regression, Knowledge-Based Systems (2016), http://dx.doi.org/10.1016/j.knosys.2016.07.038