Frontiers of Information Technology & Electronic Engineering
www.zju.edu.cn/jzus; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Multiclass classification based on a deep convolutional network for head pose estimation*

Ying CAI[1,2], Meng-long YANG‡[3], Jun LI[2]

([1]*School of Computer Science, Sichuan University, Chengdu 610065, China*)

([2]*College of Information Engineering, Sichuan Agricultural University, Yaan 625014, China*)

([3]*School of Aeronautics and Astronautics, Sichuan University, Chengdu 610065, China*)

E-mail: caiying34@qq.com; steinbeck@163.com; ljun402@163.com

Received Apr. 20, 2015; Revision accepted May 15, 2015; Crosschecked Oct. 16, 2015

Abstract: Head pose estimation has been considered an important and challenging task in computer vision. In this paper we propose a novel method to estimate head pose based on a deep convolutional neural network (DCNN) for 2D face images. We design an effective and simple method to roughly crop the face from the input image, maintaining the individual-relative facial features ratio. The method can be used in various poses. Then two convolutional neural networks are set up to train the head pose classifier and then compared with each other. The simpler one has six layers. It performs well on seven yaw poses but is somewhat unsatisfactory when mixed in two pitch poses. The other has eight layers and more pixels in input layers. It has better performance on more poses and more training samples. Before training the network, two reasonable strategies including shift and zoom are executed to prepare training samples. Finally, feature extraction filters are optimized together with the weight of the classification component through training, to minimize the classification error. Our method has been evaluated on the CAS-PEAL-R1, CMU PIE, and CUBIC FacePix databases. It has better performance than state-of-the-art methods for head pose estimation.

Key words: Head pose estimation, Deep convolutional neural network, Multiclass classification
doi:10.1631/FITEE.1500125 **Document code:** A **CLC number:** TP391

1 Introduction

The problem of head pose estimation has enjoyed substantial attention in the computer vision community. Robust algorithms of head pose estimation could be beneficial for many applications, such as video surveillance, human computer interaction, video conferencing, and face recognition. However, it is still an intrinsically challenging task because of the appearance variation between identities, complex illumination, varied background, and other factors. Many methods use classification or regression to solve the problem of pose estimation. In this paper, we treat the problem of head pose estimation as a classification question, because we believe that there are invariant essential features in the images with the same pose and these features are suitable for pose classification. Furthermore, we find that the deep convolutional neural network (DCNN) performs well on many visual tasks, because spatial topology and shift-invariant local features are well captured (LeCun *et al.*, 1998). We consider that appropriate DCNN architecture and an effective image preprocess will produce good performance on head

‡ Corresponding author

* Project supported by the National Key Scientific Instrument and Equipment Development Project of China (No. 2013YQ49087903), the National Natural Science Foundation of China (No. 61402307), and the Educational Commission of Sichuan Province, China (No. 15ZA0007)

ORCID: Ying CAI, http://orcid.org/0000-0002-5096-6175

pose estimation.

Murphy-Chutorian and Trivedi (2009) summarized the problem of head pose estimation and Tang (2014) reviewed different algorithms for head pose estimation. Previous methods on head pose estimation could roughly be categorized into model-based approaches and appearance-based approaches.

Model-based approaches usually use geometric features. Wang and Sung (2007) used six key feature points (two outer eye corners, two inner eye corners, and two mouth corners) to estimate pose, and they assumed that the two eye corners and mouth corners are approximately in the same plane. Some other model-based approaches suffer from efficiency concerns when the template set is large. For example, Lanitis *et al.* (1995) extracted face features using the active shape model (ASM) and adopted a greedy search to match the feature points. These techniques usually need a lot of feature landmarks.

Appearance-based approaches use features which are modeled or learned from the training data. Distance metric learning (Wang *et al.*, 2008), subspace analysis (Fu and Huang, 2006), and manifold embedding (Raytchev *et al.*, 2004) are popular methods used to extract appearance features. Huang *et al.* (2010) used Gabor feature-based random forests and assistant linear discriminative analysis (LDA) to obtain a classification accuracy of 97.23%.

There are some hybrid methods. Storer *et al.* (2009) used a 3D morphable model (3DMM) for head pose estimation. A morphable face model represents the face by a vector space based on the statistics of sample faces. It is effective but slightly time consuming.

From another perspective, there are two steps in most head pose estimation algorithms. The first step is feature extraction. The second step estimates the head pose according to the features obtained. Ma *et al.* (2006) proposed the local Gabor binary patterns (LGBP) features and used a radial basis function (RBF) kernel support vector machine (SVM) classifier to estimate the pose. They have achieved a classification accuracy of 97.14%.

Research on neural networks showed that the standard fully connected multi-layer networks can be used as outstanding classifiers if there are good features. Nevertheless, DCNN could complete feature extraction and classification in an integrated architecture and performs excellently, as DCNN learns the types of shift-invariant local features and increases robustness to irrelevant variability of the inputs. The network takes the raw pixels of the image as input to make best use of texture context information, and learns low-level features and high-level representation in an integrated fashion. The global high-level features at higher layers of the deep structures could effectively work on challenging images, for example, when low-level features from local regions are ambiguous or corrupted (Sun *et al.*, 2013). Thus, convolutional networks will be a better way to estimate the head pose. Through training the network, the feature extraction filters are optimized together with the weights of the classification components, to obtain a satisfactory classification accuracy.

2 Convolutional neural networks

The first implementation of a convolutional neural network can be considered by Fukushima's Neocognitron (Fukushima, 1980). However, only recently has the potential of convolutional networks been really recognized.

The convolutional neural network has been used successfully in a number of vision applications such as handwriting recognition (LeCun *et al.*, 1989; 1998), visual document analysis (Simard *et al.*, 2003), car detection (Matsugu and Cardon, 2004), face parsing (Luo *et al.*, 2012), image classification (Krizhevsky *et al.*, 2012), and scene parsing (Farabet *et al.*, 2013). Using deep convolutional networks, Ciresan *et al.* (2012) significantly improved the state of the art on some standard classification datasets.

Convolutional neural networks set up local receptive fields, weight sharing, and spatial subsampling (LeCun and Bengio, 1995). Unlike the shallow neural network, small local receptive fields of convolutional winner-take-all neurons yield large network depth, resulting in many sparsely connected neural layers, as found in macaque monkeys between retina and visual cortex. Each layer in a convolutional net is composed of units organized in planes, called feature maps. All units within a feature map share the same weight. Weight sharing could reduce the number of parameters and helps generalization. Another characteristic of convolutional neural networks is the spatial sub-sampling layer. The purpose is to achieve robustness to slight distortions, playing the same role as the complex cells in visual

perception models. Jarrett *et al.* (2009) compared different combinations of nonlinearities and pooling strategies and introduced strong nonlinearities after convolution, including absolute value rectification and local contrast normalization.

The purpose of convolutional network training is to minimize the mean squared error (MSE) in a set of target outputs. We can compute the MSE using Eq. (3). Therefore, the back-propagation (BP) algorithm is used in the learning process to adjust the weights parameter to achieve the goal. The following formula updates the neuron's weights:

$$\omega(t+1) = \omega(t) + \eta\delta(t)x(t), \tag{1}$$

where η is the learning rate, $x(t)$ is the input to the neuron, and $\delta(t)$ is the error term for the neuron.

3 Proposed method

3.1 Architecture of DCNN

The convolutional network architecture used for training is reported here (Fig. 1). It is similar to the well-known LeNet5 (LeCun *et al.*, 1998), but more feature maps, different pooling, and part-connection are used, because our input face image is more complex than LeNet5's input digit and character image. The six layers are named C1, S2, C3, S4, C5, and F6, respectively. The character 'C' indicates a convolutional layer, the S layer is a sub-sampling layer, and the F layer is a fully connected layer. The input of the network is a 32×32 pixel gray-scale image. The first layer C1 has 10 feature maps of size 28×28 and uses 10 convolution kernels with 5×5 size. Each unit in each feature map is connected to a 5×5 neighborhood in the input. Contiguous units in C1 take input from the neighborhood on the input that overlaps by 4 pixels. The next layer, S2, is a 2×2 sub-sampling layer with 10 feature maps of size 14×14. Each unit in each map is a weighted maximum value of a 2×2 neighborhood in the corresponding feature map in C1. Contiguous units in S2 take input from contiguous, non-overlapping 2×2 neighborhoods in the corresponding map in C1. It is a so-called pooling layer. C3 is convolutional with 20 feature maps of size 10×10. Each unit in each feature map is connected to several 5×5 neighborhoods at identical locations in a subset of S2's feature maps. Different C3 maps choose different subsets of S2 according to the matrix shown in Fig. 2, to break the symmetry and force the maps to extract different and complementary features. Forcing information through fewer connections should result in the connections becoming more meaningful. So, this is a part-connection between S2 and C3, and there are 136 enabled convolution kernels with a 5×5 size to learn. S4 is a sub-sampling layer with 2×2 sub-sample ratios containing 20 feature maps of size 5×5. C5 is a convolutional layer with 120 feature maps of size 1×1 and uses 2400 convolution kernels of size 5×5. Each C5 map takes input from all 20 of S4's feature maps and uses 20 different convolution kernels. F6 is the output layer. It has 7 outputs (since there are 7 class labels) and is fully connected to C5 as a classification layer.

Fig. 2 shows the set of S2 feature maps combined by each C3 feature map, where columns indicate the feature maps in C3, rows indicate the feature maps in S2, and the symbol '*' means an existing connection.

We choose max-pooling in our network for sub-sampling. Although building local invariance to shift can be performed with any symmetric pooling operation, Scherer *et al.* (2010) found that max-pooling can lead to faster convergence, select

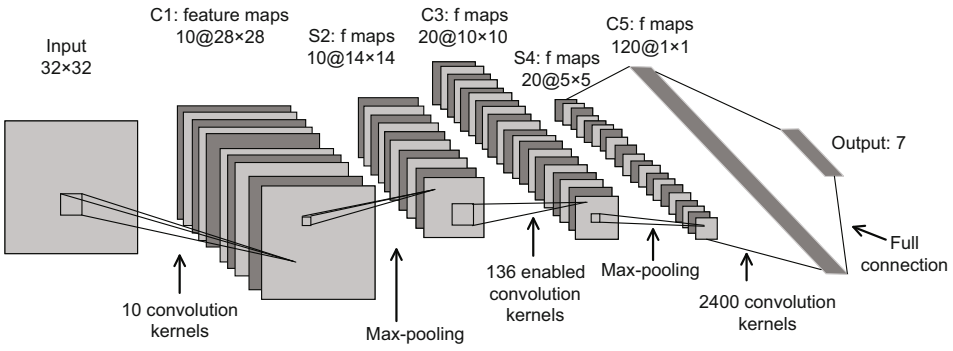

Fig. 1 Architecture of our six-layer convolutional neural network for head pose estimation

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	*		*					*		*	*		*	*	*	*	*			*
2		*	*	*				*	*		*	*	*	*	*	*	*	*		*
3	*		*	*	*				*	*		*	*	*	*	*	*	*	*	*
4		*	*	*	*	*				*	*		*	*	*	*	*	*	*	*
5	*		*	*	*	*	*	*			*	*		*	*	*	*	*	*	*
6		*	*	*	*	*	*	*	*			*			*	*	*	*	*	*
7	*			*	*	*	*		*	*			*			*	*	*	*	*
8		*			*	*	*	*		*	*		*	*			*	*	*	*
9	*					*	*	*	*		*	*	*	*	*			*	*	*
10		*					*		*	*		*	*	*	*	*			*	*

Fig. 2 The matrix for part-connection

superior invariant features, and improve generalization. The difference between max-pooling and average pooling is that the average operation is replaced by a max operation. We use a classical gradient BP algorithm and a stochastic version of the Levenberg-Marquardt algorithm with diagonal approximation of the Hessian (LeCun and Bengio, 1995) to complete optimization.

3.2 Image preprocessing

As is well known, appropriate image preprocessing can improve the accuracy of classification. Our algorithm has robustness to temperate scale and location change, because of the powerful learning ability of DCNN. So, we do not need precise face crop as long as we ensure that the face is included with little background. We design a simple image preprocessing method. The steps are as follows:

First, we employ a recently developed DCNN facial point detector (Sun *et al.*, 2013) to extract five facial feature points, including left eye center, right eye center, nose tip, left mouth corner, and right mouth corner, from the input image. Serious pose (yaw pose greater than $\pm 40°$ or pitch pose greater than $\pm 30°$) will result in a slight inaccuracy. Fortunately, the accuracy is enough for us to roughly calculate the rectangular box used to crop a face.

Second, the nose tip is treated as the center of the box. We denote the maximum value of the two distances from left and right mouth corners to nose tip in the vertical direction as Y_{down}, the maximum value of the two distances from left and right eye centers to nose tip in the vertical direction as Y_{up}, the distance in the horizontal direction from the left eye center to nose tip as X_{left}, and the distance from the right eye center to nose tip as X_{right}. So, the left top point and the right bottom point of the face box can be calculated according to the following formulae:

$$\begin{cases} X_{\text{lt}} = X_{\text{nt}} - X_{\text{right}} N_{\text{l}}, \\ Y_{\text{lt}} = Y_{\text{nt}} - Y_{\text{up}} N_{\text{u}}, \\ X_{\text{rb}} = X_{\text{nt}} + X_{\text{left}} N_{\text{r}}, \\ Y_{\text{rb}} = Y_{\text{nt}} + Y_{\text{down}} N_{\text{d}}, \end{cases} \tag{2}$$

where X_{nt} and Y_{nt} are the coordinates of the nose tip, X_{lt} and Y_{lt} are the coordinates of the box's left top point, X_{rb} and Y_{rb} are the coordinates of the box's right bottom point, and N_{u}, N_{d}, N_{l}, and N_{r} are the proportional coefficients. We can adjust the size of the box through changing these coefficients. However, when serious poses cause the smaller one of X_{right} and X_{left} to be less than 1/6 of the larger one or zero, the smaller one will be too small to use. In this case, we replace the smaller one of $X_{\text{right}} N_{\text{r}}$ and $X_{\text{left}} N_{\text{l}}$ with $M \cdot \text{distance\_eyes}$ in the above formulae, where distance_eyes is the horizontal distance of two eye centers in the frontal face image, and M is the proportional coefficient. Using the same proportional coefficients (N_{u}, N_{d}, N_{l}, N_{r}, and M) between different resolution databases, we can keep a uniform crop style. Now we can crop the face from the original input images based on the face box. Via this method we can maintain the individual-relative facial features ratio. This is helpful for extracting excellent features. The graphics of this method are shown in Fig. 3.

Finally, we unify the size of images to 32×32 and normalize all pixels of the image to zero mean and unit variance, which can accelerate learning and reduce the influence of illumination (LeCun *et al.*, 1991).

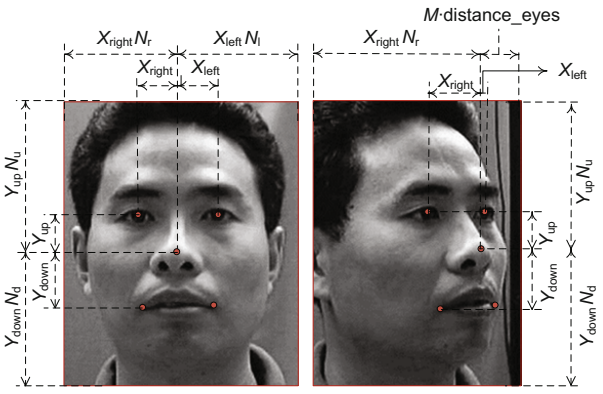

Fig. 3 Crop parameters. The red frame is the crop face box. References to color refer to the online version of this figure

3.3 Training tactics

Preparing a training image set is a very important part of deep learning. We can refer to some tactics used in training LeNet5 (LeCun *et al.*, 1998) to improve the generalization of neural networks. The most effective one is distortion, which includes scale, shift, rotation, and elastic distortion. Distortion forces the neural network into looking more closely at the important and essential aspects of the training patterns. Another advantage is the enlargement of the training set. By adjusting the line between two eye centers to be horizontal, we can remove the pose of the head in the roll direction. Image rotation is inappropriate for our method, but shift and scale are still useful. We shift the face box (calculated in Section 3.2) on the original image in four directions (up, down, left, and right) using four shift values (7, 10, 12, and 15 pixels) on the CAS-PEAL-R1 database whose resolution is 360×480. So, we can enlarge the size of the training set by a factor of 16. We also zoom the scale of the face box. Specifically, we shrink and magnify the face box using four values (3, 5, 7, 10 pixels). Through changing scale we can enlarge the training set by a factor of 8. The scale and shift are implemented only on the training set. Finally the training set is passed through the neural network in a different randomized order in each epoch to avoid over fitting.

4 Experimental results

4.1 Experimental results of seven poses on the CAS database

CAS-PEAL-R1 (Gao *et al.*, 2008) is a subset of the CAS-PEAL face database, which has been released for the purposes of research. It contains 30 871 images of 1040 individuals (595 males and 445 females) with varying pose, expression, accessory, and lighting (PEAL). We are interested only in the pose subset. The platform of the CAS camera system is shown in Fig. 4. To capture images with different pitch poses, the model was asked to look upwards, look steadily at camera C4 (the middle one), and look downwards. For each same pitch pose, they simultaneously capture nine images with different yaw poses through the nine cameras C0–C8. The CAS-PEAL-R1 releases only images taken by C1–C7.

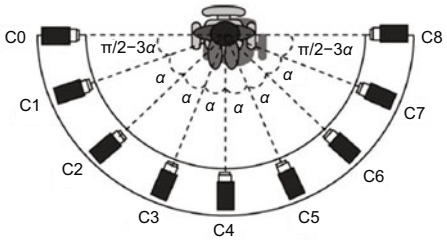

Fig. 4 Cameras configuration of the CAS-PEAL-R1 database (α is 15° or 22.5°)

The reason why we chose CAS-PEAL-R1 is that deep learning needs a lot of training samples and pose estimation needs accurate pose annotation. The way of taking pose pictures can determine the accuracy of pose annotation. Of course, simultaneous snapshot of models by different cameras is a more precise way than asking models to turn their heads. Among all public pose databases, CAS is a good option because sufficient specimens are captured by simultaneous snapshot. We use only the images whose pitch poses are annotated with 0 and yaw poses annotated with $(\pm 45°, \pm 30°, \pm 15°, 0°)$. The seven poses contain 6876 images in all.

We used five factors to evaluate the result. The first is classification accuracy (CA) on the test image set. The second is the summation of MSE (SMSE) of all test images. It helps to analyze the MSE which we want to minimize. We can compute the MSE using Eq. (3). The third is the number of all error classifications (NE). The fourth is the number of error classifications which occur between adjacent categories (NEA). For example, the 0° can be judged to ±15° in seven poses classification $(\pm 45°, \pm 30°, \pm 15°, 0°)$. This is considered a slight error. The fifth is the number of error classifications which occur between non-adjacent categories (NENA); i.e., the 0° is judged to any other pose except ±15°. This is considered a serious error.

$$E = \sqrt{\sum_{i=1}^{n} (x^i - t^i)^2}, \tag{3}$$

where n is the number of categories, x is the actual value of the output, and t is the label. In experiments of seven poses, our label is a seven-dimensional vector. The value of one dimension which corresponds to an assigned pose is +1 and others are −1. The output is also a seven-dimensional vector, and the value range of each dimension is $(-1, +1)$.

The experimental steps are as follows:

Step 1: experiment on the original image. We prepared training and test sets according to the ratio 9:1 in every pose. In other words, we randomly chose 10% images in every pose including 688 images for test and there remains 6188 images for training. In the following experiments we always used the ratio 9:1 between the training set and the test set if there was no special declaration. As shown in Table 1, CA was acceptable. It proved our assumption that the head pose features abstracted through DCNN are suitable for classification. In contrast, the SMSE was large and NENA was increased.

Step 2: We cropped the face area from the original image using the method mentioned in Section 3.2. If we enlarge the proportional coefficients, the box will contain more face texture information. If we reduce them, the box will contain less background. We want to know which proportional coefficients will have good performance. So, we compared the results between $N_{\mathrm{u}} = 4$, $N_{\mathrm{d}} = 2.5$, $N_{\mathrm{l}} = 3$, $N_{\mathrm{r}} = 3$, $M = 0.5$ and $N_{\mathrm{u}} = 2$, $N_{\mathrm{d}} = 2$, $N_{\mathrm{l}} = 2$, $N_{\mathrm{r}} = 2$, $M = 0.3$. The whole face will be cropped using the former coefficients and only the five sense organs will be cropped using the latter. Fig. 5 shows the graphic comparison. For protection of privacy, we show only three poses. The results of the classification are shown in Table 1. Compared with step 1, CA was lower using the five sense organs image than using the original image, because the original image included more changeless background. This is helpful for classification. On the other hand, if the background is uncontrolled, it will be obstructive to classification. We cannot always keep the background unchanged in practical applications. Thus, the original image is not a good option, and we should remove the influence coming from the background to improve the generalization of the algorithm. We next examined the performance of the whole face image; CA was improved slightly, SMSE was reduced observably, and NENA was only one. We can see that more face texture is useful for pose classification. So, in the following experiments we chose the former proportional coefficients.

Step 3: We used the scale tactic mentioned in Section 3.3 on the 6188 training images. Now, we obtained additional eight times the number of training images, a total of 55 692 images for training. The scale tactic introduces a multi-scale face image in the training set, and it is helpful for handling slight scale changes. So, as shown in Table 1, CA was better than in step 2, SMSE was reduced to 15.05, and NENA was zero.

Fig. 5 The two crop results using different proportional coefficients. The first row shows the whole face images and the second row the five sense organs

Step 4: We used the shift tactic mentioned in Section 3.3 on the 6188 training images. Now, we obtained additional 16 times the number of training images, a total of 105 196 images for training. Through this tactic, the influence caused by alignment offset in four directions including up, down, left, and right can be weakened. So, as shown in Table 1, CA was better than in steps 2 and 3, SMSE was reduced to 11.32, and NENA was zero.

Step 5: We used both the shift and scale tactics. Now there were 154 700 images for training. Not only has the size of the training sample been steeply increased, but also the disadvantageous influence of the alignment offset and scale change controlled. A large number of training samples provided protection for parameter optimization and avoided over-fitting. Meanwhile, two tactics improved the robustness of features. So, as shown in Table 1, CA was improved to 98.4, SMSE was reduced to 9.82, and NENA was steady at zero.

We repeated the above five steps five times. Each time we randomly chose the test images. Then, the average results are given in Table 1.

4.2 Experimental results of nine poses on the CAS database

Having achieved good results on the CAS database using seven yaw poses, we now explore the performance of our method on pitch pose.

There are two pitch poses in CAS. We used only the images whose pitch pose was annotated

Table 1 The average results for different kinds of training image sets

Training set	CA (%)	SMSE	NE	NEA	NENA
Original images	96.4	48.78	25	18	7
CFFI	93.2	39.03	47	47	0
CWFI	96.4	26.68	25	23	2
Scaled images	97.4	15.05	18	18	0
Shifted images	98.0	11.32	14	14	0
SSI	98.4	9.82	11	11	0

CA: classification accuracy; SMSE: summation of mean squared error; NE: number of all error classifications; NEA: number of error classifications which occur between adjacent categories; NENA: number of error classifications which occur between nonadjacent categories; CFFI: cropped facial feature images; CWFI: cropped whole face images; SSI: scaled and shifted images

with (PU, PD) and yaw pose was annotated with 0°. Every pose included 939 images. Now we have nine poses including 8754 images. Using the same strategies as in step 5 (Table 1), there were 876 images for test and 196 950 images for training. Then, we changed the outputs of the network to 9. As shown in Table 2, the results were not as good as for seven poses. It is clear that, more categories need more excellent features and the pitch pose images in CAS have a congenital deficiency caused by the artificial looking upward and downward of the models in these photographs.

We explored whether deepening the DCNN layers and magnifying the input image size would improve the results. So, we designed an eight-layer convolutional network and magnified the size of the input image to 39×39. There were 4 C-layers, 3 S-layers, and 1 F-layer. The order of layers was like this C1-S2-C3-S4-C5-S6-C7-F8. There were 10 feature maps of size 36×36 and 10 convolution kernels with 4×4 size in C1, 20 feature maps of size 16×16 in C3. Because front layers extracted local low-level features and back layers processed global high-level features in a deep network, we used the same part-connection as a six-layer network on the front convolution layer C3. There were 136 enabled convolution kernels with 3×3 size between S2 and C3 which need to train, 40 feature maps of size 6×6 and 800 convolution kernels with 3×3 size in C5, 120 feature maps of size 1×1 and 4800 convolution kernels with 3×3 size in C7. All sub-sampling ratios of s-layers were 2×2. F8 was an output layer with nine outputs. The architecture of the eight-layer network is shown in Fig. 6.

Now, we experimented on an eight-layer network using the same strategies as in step 5 in Table 1. As shown in Table 2, the performance was better than that of the six-layer network. We already knew that, more training images are helpful for a deeper network. So, we executed 24 shifts with 3, 5, 7, 10, 12, and 15 pixels, respectively, and 12 scales with 3, 5, 7, 10, 12, and 15 pixels, respectively. Then, we can enlarge the training set by a factor of 36. As shown in Table 2, the improvement was slight because the added training samples came from the same individuals. If there were more training samples from different models, perhaps the improvement will be more. Additionally, this experiment was obviously more time-consuming than the experiment with six-layer network.

Table 2 The results using six and eight-layer convolutional networks on nine poses

Network architecture	NTS	CA (%)	SMSE	NE	NEA	NENA
6-layer network	196 950	97.54	18.4	22	21	1
8-layer network	196 950	98.29	17.5	15	15	0
8-layer network	291 486	98.51	16.3	13	13	0

NTS: number of training samples

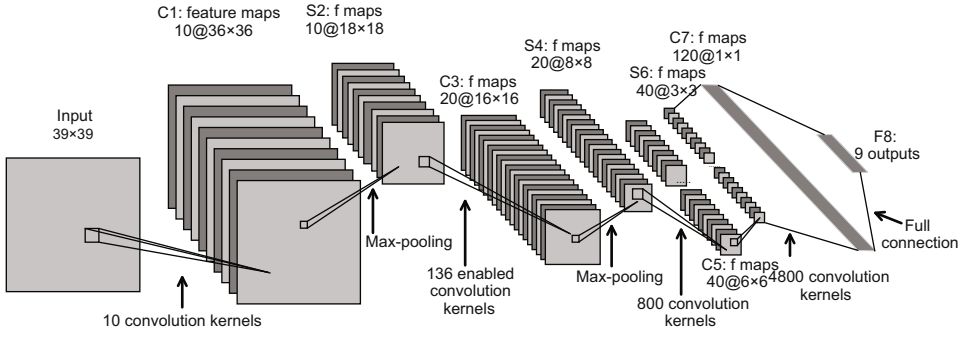

Fig. 6 Architecture of our eight-layer convolutional neural network for head pose estimation

4.3 Experimental results on other databases

The CUBIC FacePix database (Black *et al.*, 2002) consists of 30 individuals. Three sets of images are available for each individual. In the first set, every individual contains 181 images whose yaw pose varies from $+90°$ to $-90°$ with $1°$ interval and pitch pose annotated with $0°$. These images were captured by a moving video camera. The second and third sets are targeted to illumination experiments. We used only the first set and picked out the images whose yaw pose was annotated with $(\pm 45°, \pm 30°, \pm 15°, 0°)$. There are seven poses including 210 images.

As shown in Table 3, our first experiment was on the original images. CA was 81%. The second experiment was conducted on the whole face images, and CA was only 90.5%. It was lower than the CA on CAS database because there were too few individuals. Finally, we used the same tactics as in step 5 in Table 1 to enlarge the training set from 189 to 4725. CA was improved to 100%.

Because there were resemblances between cropped face images from different databases, we merged the 189 whole face training images from FacePix with the 6188 whole face training images from CAS to improve the results for each. The CA was improved from 90.5% to 100% on the CUBIC FacePix database and kept 96.4% on the CAS database. It is possible that with more individuals in FacePix, CA on CAS will be improved. Because the individuals in FacePix have different illuminations and race, all these differences are helpful in advancing the discriminating ability of the classifier. In future work we will enrich the diversity of training samples.

The CMU Pose, Illumination, and Expression (PIE) database (Sim *et al.*, 2002) consists of over 40 000 facial images of 68 individuals. The platform of the CMU camera system is shown in Fig. 7.

Table 3 The results on the CMU and CUBIC FacePix databases using seven poses

Database	Training set	CA (%)	SMSE	NE	NEA	NENA
FacePix	Original images	81	4.2	4	3	1
	CWFI	90.5	5.9	2	2	0
	SSI	100	1.5	0	0	0
	WFI	100	2.7	0	0	0
CMU	Original images	100	2.22	0	0	0
	CWFI	93.9	6.4	3	2	1
	SSI	100	3.05	0	0	0

We picked out the images taken by c37, c05, c27, c29, c11, c09, and c07 cameras with a neutral expression and not wearing glasses whose pitch pose was annotated with $\pm 22.5°$ and whose yaw pose annotated with $(\pm 45°, \pm 22.5°, 0°)$. A total of 476 images were selected. CA was 100% on the original images. Then, it decreased to 93.9% on the cropped whole face images. The reason was the same as with CAS, is that changeless backgrounds are really helpful for classification. To increase CA on cropped images, we used the same tactics as in step 5 in Table 1. Then, the number of training images was enlarged to 10 675. As shown in Table 3, CA was improved to 100%.

4.4 Comparison with other methods

A large number of head pose estimation methods have been mentioned in two head pose estimation surveys: Murphy-Chutorian and Trivedi (2009) and Tang (2014). We compared three state-of-the-art methods which have achieved the best results on the CAS-PEAL-R1 database. They are B. Ma. LGBP (survey by Murphy-Chutorian and Trivedi (2009), Table 2), C. Huang. VRF+LDA (survey by Tang (2014), Table 2), and B. Ma. LBIF+SVM (survey by Tang (2014), Table 2).

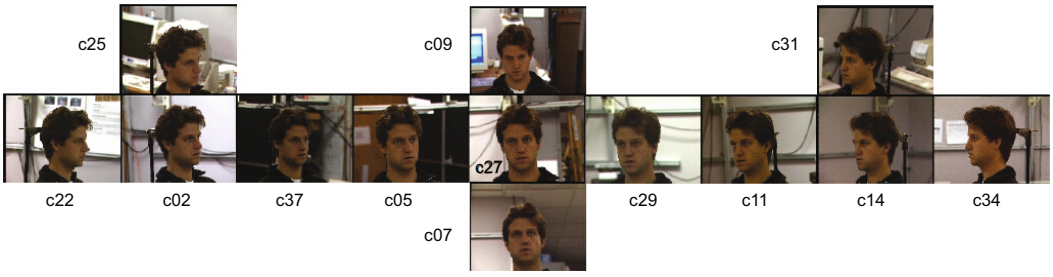

Fig. 7 An illustration of pose variation in the PIE database and the serial number of cameras

The outstanding learning ability of DCNN is based on having many training samples. So, we used tactics to obtain a larger number of different training images. We set the same test environment as for other methods, but we used more training images. In this way we paid more attention to the performance in test and the classification abilities in practical applications.

B. Ma. LGBP represents a multi-view face based on local Gabor binary patterns (LGBP) (Ma *et al.*, 2006). The LGBP is operated on many sub-regions of the images. Then they encoded the local facial characteristics into a compact feature histogram. Finally, an RBF kernel SVM classifier was trained to estimate the pose. In the CAS-PEAL database, they used a subset containing 1400 images of 200 individuals with seven poses $(\pm 45°, \pm 30°, \pm 15°, 0°)$ whose IDs range from 401 through 600. The 1400 images were divided into three subsets. Two subsets were taken as the training set including 934 images, and the other subset was taken as the testing set including 466 images. The best accuracy of pose estimation on the CAS-PEAL-R1 database using this method was 97.14% (Table 4).

C. Huang. VRF+LDA uses Gabor feature based random forests as the classification technique and implements linear discriminative analysis (LDA) (Huang *et al.*, 2010) as the node test to improve the discriminative power of individual trees in the forest, with each node generating both constant and variant numbers of child nodes. They prepared the same training and testing set as B. Ma. LGBP. The best accuracy of pose estimation on the CAS-PEAL-R1 database based on variant node splits random forests plus LDA (VRF+LDA) was 97.23% (Table 4).

B. Ma. LBIF+SVM combines the biologically inspired features (BIF) with the local binary pattern (LBP) (Ma, 2013) feature into a new feature descriptor named 'local biologically inspired features' (LBIF). Then some ensemble-based supervised methods were applied to reduce the dimension of the LBIF features. In their experiments, they took the yaw poses as the class labels. In this sense, the images with the same yaw pose but different pitch poses belong to the same class. So, the images in the CAS-PEAL-R1 database belong to seven different classes. They used a subset containing 4200 images of 200 subjects whose IDs range from 401 through 600. The 4200 images were divided into three subsets. Two subsets were taken as the training set including 2800 images, and the other subset was taken as the testing set including 1400 images. The best accuracy of pose estimation on the CAS-PEAL-R1 database based on the SVM classifier was 94.57% (Table 4).

Experiment 1 uses the same test sets as B. Ma. LGBP and C. Huang. VRF+LDA. We used the same 200 individuals for test and the remaining 840 individuals for training. Then the same tactics as in step 5 in Table 1 have been used on training sets. The 1400 test images were randomly divided into three subsets, each subset including 466 images. We conducted the experiment on all the three subsets and took the average of the three.

Experiment 2 uses the same test sets as B. Ma. LBIF+SVM. We used the same 200 individuals for test and the remaining 840 individuals for training. Again, we used the same tactics as in step 5 in Table 1. The 4200 test images were randomly divided into three subsets, each subset including 1400 images. We conducted the experiment on all the three subsets and took the average of the three.

As shown in Table 4, our method achieved better results than all the other methods.

Table 4 The results of our method compared with three state-of-the-art methods

Method	Classification accuracy (%)	Number of test samples	Number of training samples
B. Ma. LGBP	97.14	466	934
C. Huang. VRF+LDA	97.23	466	934
Our experiment-1	98.47	466	136 900
B. Ma. LBIF+SVM	94.57	1400	2800
Our experiment-2	97.17	1400	410 700

5 Conclusions

In this paper, we propose an effective method for head pose classification based on DCNN. One major point of excellence of DCNN is the abstracted spatial topology and shift-invariant texture features. It is useful for head pose classification on a single 2D image. So, we design two appropriate DCNN architectures to compare their classification ability in all kinds of situations. Before training, an effective way has been proposed to preprocess images. To obtain a powerful classifier, we adopt the shift and scale strategies to complete the preparation of training

images. Our classifier performs well on three different databases. As the results show, our method significantly improves the classification accuracy compared with state-of-the-art methods. According to an analysis of false examples, we find that our classifier tends to make mistakes on the images with serious expression or occlusion, because our training data include very few samples with expression or occlusion. In the future we are going to enrich the diversity of training samples and improve the architecture of the network to make our method robust for more target classes and really uncontrolled test environments.

References

Black, J.A.Jr., Gargesha, M., Kahol, K., *et al.*, 2002. A framework for performance evaluation of face recognition algorithms. SPIE, **4862**:163. [doi:10.1117/12.473032]

Cireşan, D., Meier, U., Schmidhuber, J., 2012. Multi-column deep neural networks for image classification. CVPR, p.3642-3649. [doi:10.1109/CVPR.2012.6248110]

Farabet, C., Couprie, C., Najman, L., *et al.*, 2013. Learning hierarchical features for scene labeling. *IEEE Trans. Patt. Anal. Mach. Intell.*, **35**(8):1915-1929. [doi:10.1109/TPAMI.2012.231]

Fu, Y., Huang, T.S., 2006. Graph embedded analysis for head pose estimation. 7th Int. Conf. on Automatic Face and Gesture Recognition, p.1-6. [doi:10.1109/FGR.2006.60]

Fukushima, K., 1980. Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biol. Cybern.*, **36**(4):193-202. [doi:10.1007/BF00344251]

Gao, W., Cao, B., Shan, S.G., *et al.*, 2008. The CAS-PEAL large-scale Chinese face database and baseline evaluations. *IEEE Trans. Syst. Man. Cybern. A*, **38**(1):149-161. [doi:10.1109/TSMCA.2007.909557]

Huang, C., Ding, X.Q., Fang, C., 2010. Head pose estimation based on random forests for multiclass classification. ICPR, p.934-937. [doi:10.1109/ICPR.2010.234]

Jarrett, K., Kavukcuoglu, K., Ranzato, M., *et al.*, 2009. What is the best multi-stage architecture for object recognition? ICCV, p.2146-2153. [doi:10.1109/ICCV.2009.5459469]

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. ImageNet classification with deep convolutional neural networks. NIPS, p.1097-1105.

Lanitis, A., Taylor, C.J., Cootes, T.F., *et al.*, 1995. Automatic interpretation of human faces and hand gestures using flexible models. Int. Workshop on Automatic Face- and Gesture-Recognition, p.98-103.

LeCun, Y., Bengio, Y., 1995. Convolutional networks for images, speech, and time series. *In*: Arbib, M.A., (Ed.), The Handbook of Brain Theory and Neural Networks. MIT Press, Cambridge.

LeCun, Y., Jackel, L.D., Boser, B., *et al.*, 1989. Handwritten digit recognition: applications of neural network chips and automatic learning. *IEEE Commun. Mag.*, **27**(11):41-46. [doi:10.1109/35.41400]

LeCun, Y., Kanter, I., Solla, S.A., 1991. Eigenvalues of covariance matrices: application to neural-network learning. *Phys. Rev. Lett.*, **66**(18):2396. [doi:10.1103/PhysRevLett.66.2396]

LeCun, Y., Bottou, L., Bengio, Y., *et al.*, 1998. Gradient-based learning applied to document recognition. *Proc. IEEE*, **86**(11):2278-2324. [doi:10.1109/5.726791]

Luo, P., Wang, X.G., Tang, X.O., 2012. Hierarchical face parsing via deep learning. CVPR, p.2480-2487. [doi:10.1109/CVPR.2012.6247963]

Ma, B.P., Zhang, W.C., Shan, S.G., *et al.*, 2006. Robust head pose estimation using LGBP. ICPR, p.512-515. [doi:10.1109/ICPR.2006.1006]

Ma, B.P., Chai, X.J., Wang, T.J., 2013. A novel feature descriptor based on biologically inspired feature for head pose estimation. *Neurocomputing*, **115**(4):1-10. [doi:10.1016/j.neucom.2012.11.005]

Matsugu, M., Cardon, P., 2004. Unsupervised feature selection for multi-class object detection using convolutional neural networks. ISNN, p.864-869. [doi:10.1007/978-3-540-28647-9_142]

Murphy-Chutorian, E., Trivedi, M.M., 2009. Head pose estimation in computer vision: a survey. *IEEE Trans. Patt. Anal. Mach. Intell.*, **31**(4):607-626. [doi:10.1109/TPAMI.2008.106]

Raytchev, B., Yoda, I., Sakaue, K., 2004. Head pose estimation by nonlinear manifold learning. ICPR, p.462-466. [doi:10.1109/ICPR.2004.1333802]

Scherer, D., Müller, A., Behnke, S., 2010. Evaluation of pooling operations in convolutional architectures for object recognition. Proc. 20th Int. Conf. on Artificial Neural Networks, p.92-101. [doi:10.1007/978-3-642-15825-4_10]

Sim, T., Baker, S., Bsat, M., 2002. The CMU pose, illumination, and expression (PIE) database. 5th IEEE Int. Conf. on Automatic Face and Gesture Recognition, p.46-51. [doi:10.1109/AFGR.2002.1004130]

Simard, P.Y., Steinkraus, D., Platt, J.C., 2003. Best practices for convolutional neural networks applied to visual document analysis. 7th Int. Conf. on Document Analysis and Recognition, p.958-963. [doi:10.1109/ICDAR.2003.1227801]

Storer, M., Urschler, M., Bischof, H., 2009. 3D-MAM: 3D morphable appearance model for efficient fine head pose estimation from still images. ICCV, p.192-199. [doi:10.1109/ICCVW.2009.5457701]

Sun, Y., Wang, X.G., Tang, X.O., 2013. Deep convolutional network cascade for facial point detection. CVPR, p.3476-3483. [doi:10.1109/CVPR.2013.446]

Tang, Y.Q., Sun, Z.N., Tan, T.N., 2014. A survey on head pose estimation. *Patt. Recogn. Artif. Intell.*, **27**(3):213-225 (in Chinese).

Wang, J.G., Sung, E., 2007. EM enhancement of 3D head pose estimated by point at infinity. *Image Vis. Comput.*, **25**(12):1864-1874. [doi:10.1016/j.imavis.2005.12.017]

Wang, X.W., Huang, X.Y., Gao, J.Z., *et al.*, 2008. Illumination and person-insensitive head pose estimation using distance metric learning. ECCV, p.624-637. [doi:10.1007/978-3-540-88688-4_46]