BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **16**, No 6
Special issue with selection of extended papers from 6th International Conference on Logistic, Informatics and Service Science LISS'2016

Sofia • 2016 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.1515/cait-2016-0083

Head Pose Estimation Based on Robust Convolutional Neural Network

Jiao Bao, Mao Ye

School of Computer Science and Engineering, University of Electronic Science and Technology of China, Center for Robotics, Key Laboratory for NeuroInformation of Ministry of Education, Chengdu, China
Emails: shazhijv@126.com cvlab.uestc@gmail.com

Abstract: *Head pose estimation plays an important role in face recognition. However, it faces vast challenges on account of the initialization, facial feature points' location accuracy and so on. Inspired by the observation that head pose angles change smoothly and continuously, we present a method based on a robust convolutional neural network for head pose estimation. The proposed network architecture consists of three levels and each level has three convolutional neural networks. The first level is a global one; it predicts the head pose quickly as a preliminary estimation. The following two levels are local ones; they refine the estimation achieved from the previous level step by step. Higher and higher resolution image with different input regions are taken as input in our network. At last, a multi-level regression is employed to combine the estimations from each level. The whole process is conducted in a cascade way to improve the head pose estimation performance directly with three angles together. We perform large experiments on nine challenging benchmark datasets. The experimental results demonstrate that our method performs better than the compared methods.*

Keywords: *Head pose estimation, convolutional neural network, cascade network, multi-level regression, deep learning.*

1. Introduction

Head pose estimation is defined as the process to predict the orientation parameters or the Euler rotation angles of the face in images. Generally speaking, there are two models for head pose, i.e., face orientation model and Euler rotation angle model as demonstrated in Fig. 1. Recently, due to the broad application of head pose

estimation in face recognition, facial feature analysis and human computer interaction, it becomes a hot topic in pattern recognition and computer vision [1].

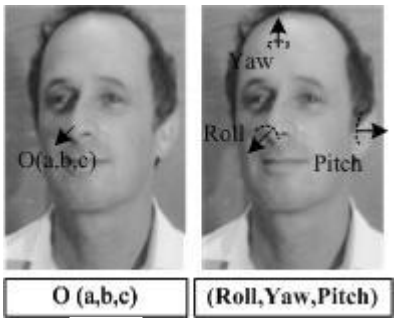


Fig. 1. Two description models of head pose. The left one is face orientation model (a); the right one is Euler rotation angle model (b)

Accordingly, a lot of head pose estimation approaches have been proposed. These approaches can be classified into two categories [1], i.e., one is based on the traditional machine learning method and the other is based on the deep network.

Traditional approaches contain detector array method [2], nonlinear regression method [3], manifold embedding method [4], flexible model method [5], geometric method [6], tracking method [7] and hybrid method [8]. These approaches have ability to dynamically construct model of the human face in image, and generate a new model or adapt the existing model relying on an initialized position and pose.

However, there are many limitations of these approaches [1]. Firstly, they rely on the initialization and facial feature points' location accuracy. But good initialization and high accuracy are still a challenging problem in the real scene, especially with extreme pose, illumination, expressions, or occlusions and so on. Secondly, they mostly estimate head pose with one or two angles, or avoid head pose estimation with extreme angles.

Recently, deep convolutional neural network and other deep model methods have been used in computer vision and machine learning, such as face detection, pose estimation [9], face parsing [10], image classification [11], facial point detection [12], depth map [13] and so on. However, there are few such methods used for head pose estimation. K a n et al. [14] propose a deep progressive auto-encoder network for head pose estimation. It achieves great success by learning the non-linear function from the non-frontal human face images to the frontal ones. However, the estimation result is limited to [–45°, 45°], it works not well when the image with extreme angles. As shown in Fig. 2, we can see that it is still a challenging problem when samples are perturbed with extreme pose, illumination, expressions, or occlusions.

In order to solve the problems mentioned above, we propose a new approach for head pose estimation based on a robust deep convolutional neural network with three carefully designed levels. The head pose estimation is formulated as the multi-level regression problem towards three Euler rotation angles.

The primary contributions of the proposed method are as follow:

1. A new method based on a robust convolutional neural network is proposed for head pose estimation. The designed network estimates head pose step by step with one global level and two local levels.

2. The three angles are estimated together from the face images directly without the initialization and facial feature points' location accuracy.

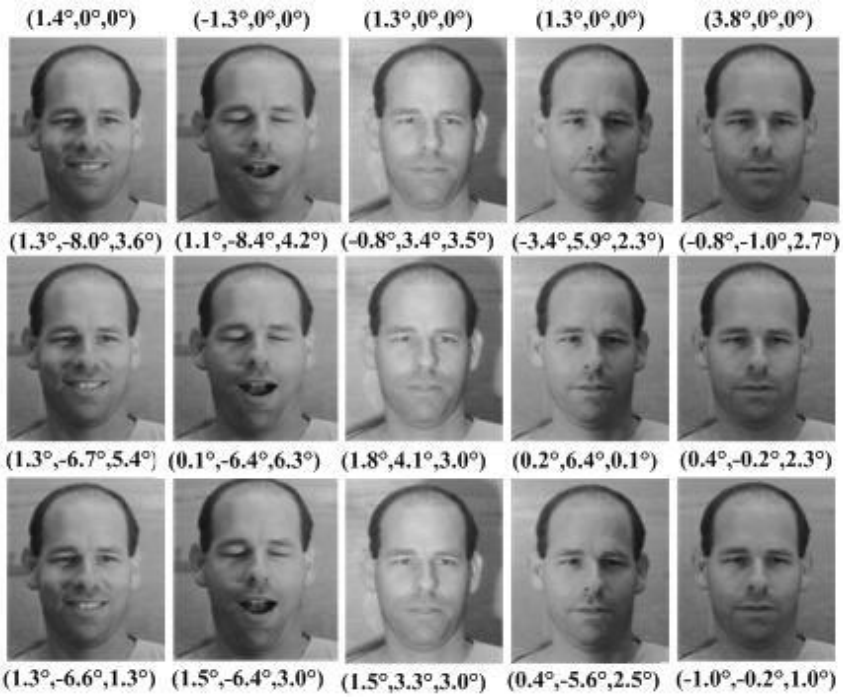


Fig. 2. Examples of head pose estimation. The top text: the labelled angles of images. The first row: roust rough estimations of our first global level networks. The second and the third rows: the tuned estimation results of our second and third local levels of the proposed networks. Obviously, the results are improved step by step. In addition, we can see our method is not sensitive to illumination, expression, etc.

The rest of the work is arranged as follows. Section 2 displays the robust convolutional neural networks in detail. Section 3 shows the implementation details. Section 4 presents the experiment results. Section 5 draws the conclusion.

2. The proposed network

In this section, we present a method for head pose estimation. Firstly, we will give an overview of the proposed network in the first part. Secondly, we will describe the components of the framework in details in the second, third and fourth parts. Finally, we will illustrate the motivation of selecting this deep convolutional neural network and give some discussions in the fifth part.

2.1. Model overview

In this work, we will use Euler rotation angle model to describe head pose. Therefore, three angles will be estimated, i.e., the roll, the pitch and the yaw. As shown in Fig. 3, in our work, three levels are carefully designed, i.e., one global level and two local levels. The estimations of these levels are combined by a multi-level regression. Furthermore, in order to prevent errors to be amplified in the network with deeper and deeper level, discriminant conditions are introduced to control these errors in each level.

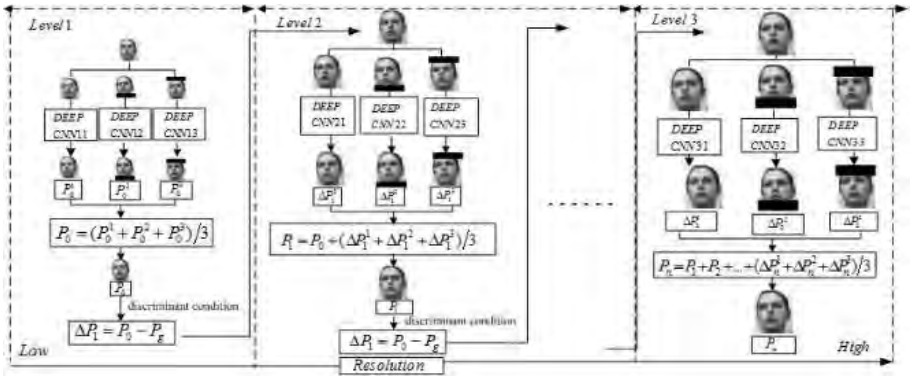


Fig. 3. Overview of the robust network for head pose estimation. The input human face image is 2D

In Fig. 3, the first global level networks are denoted as CNN11, CNN12, and CNN13. Networks in this level predict a preliminary estimation P_0. Networks in the following levels are local, they refine the previous estimation in a cascade way. Networks in the second level are denoted as CNN21, CNN22 and CNN23 which predict the deviations between the current estimation and the ground truth. Networks in the nth level are denoted as CNNn1, CNNn2 and CNNn3 and their actions are similar to networks in the second level. Black shaded areas of face images are the abandon parts. With the deepening of the network, the resolution of the input image is higher and higher.

The first level of the network (denoted as the global level) emulates a roughly approximate head pose from low resolution human face images with different input regions. There are three convolution neural networks in the first global level, and these networks have similar structure, as shown in Fig. 3. All of these networks have an input layer, two convolutional layers, two pooling layers and an output layer, respectively (as shown in Fig. 4). The only difference between them is the input layer, i.e., the input region of human face image. Three input regions of the global level network are the whole human face image (CNN11), the top and middle part of face image (CNN12) and the middle part and the bottom part of face image (CNN13) respectively. Due to these three CNNs with different input regions, our network will be more robust to obtain initial head pose estimation namely P_0.

After getting the robust preliminary head pose estimation P_0 from the global level, successive levels networks (denoted as the local levels) endeavour to refine this preliminary head pose estimation by regressing the deviations ΔP between the

current estimation and the ground truth data step by step. There are also three convolution neural networks with similar structure in each local level. In order to characterize these tiny variations, higher-resolution human face images are taken as input. Furthermore, these three head pose angles are estimated and are refined together rather than separately in each level.

Last, a multi-level nonlinear regression is employed to combine the estimations from the global and local levels. Apparently, this regression has two parts which correspond to the rough and adjustment estimations separately.

2.2. The global level of the network

Let $x \in \mathbf{R}^d$ indicates the human face image of d pixels, $P_g(x) \in \mathbf{R}^3$ denotes the ground truth of the normalized head pose angles respectively. In our work, the goal of the head pose estimation is to learn a mapping nonlinear relationship **F** from the human face image space to the head pose space directly:

$$\mathbf{F}: x \to P_{\mathrm{g}}(x). \tag{1}$$

Generally speaking, to model the complex and nonlinear function **F**, the head pose estimation problem is formulated as minimizing the mean square error of the following objective:

$$\mathbf{F}^* = \arg\min_{\mathbf{F}} \| P_{\mathrm{g}}(x) - \mathbf{F}(x) \|_2^2, \tag{2}$$

where $\mathbf{F} = \{f_1, f_2, ..., f_i, ..., f_k\}$, f_i is the complex mapping function of i-th CNN network. There is also an activation function σ in each network, and σ is a $\tanh$ function in our work, the output range of this function is [–1, 1].

One of the most remarkable things is that the objective of the global level optimizes the difference between $P_{\mathrm{g}}(x)$ and $f_j(x)$, while the objective of the local level optimizes the difference between $\Delta P(x)$ and $f_j(x)$ in the local levels, where $\Delta P_{\mathrm{g}}(x)$ is the difference between $P_{\mathrm{g}}(x)$ and the current head pose which is learned from the previous level, $f_i(x)$ and $f_j(x)$ represent the nonlinear mapping of the network in each local level. In other words, $f_i(x) \in \mathbf{F}$ in the global level learns head pose estimation, while in the local level it learns the deviation.

An average estimation P_0 is calculated from the first level as the initial:

$$P_0 = \frac{(\tilde{P}_1)_1 + (\tilde{P}_1)_2 + ... + (\tilde{P}_1)_{n_1}}{n_1}, \tag{3}$$

where $(\tilde{P}_1)_{n_1}$ is the estimation from the global level, in this work, $n_1 = 3$. After this stage, a rough but robust estimation is obtained. Thus, it is easy to calculate the deviation between the current estimation P_0 and the ground truth, and the deviation is regarded as the ground truth data for the next local level.

2.3. The local level of the network

Once the initial estimation P_0 is obtained, several successive local levels networks are employed to improve P_0. These successive levels iteratively estimate the updates $\Delta\tilde{P}_j(x)$ between the current estimation $P_{j-1}(x)$ and the ground truth $P_{\mathrm{g}}(x)$.

With the high resolution input face image x, the objective of each CNN in the first local level learns a function $\mathbf{L_1}$ from image space to the deviations $\Delta P_1(x)$ as follows:

$$\mathbf{L_1}^* = \arg\min_{\mathbf{L_1}} \| \Delta P_1(x) - \mathbf{L}_1(x) \|_2^2 . \quad (4)$$

where $\Delta P_1(x) = P_g(x) - P_0(x)$.

With the average estimation update $\Delta\widetilde{P}_1$ from the first local level, we obtained the new estimation $P_1 = P_0 + \Delta\widetilde{P}_1$.

Then for the *k*-th successive local level, the goal is to optimize the new deviation $\Delta P_k(x) = P_g(x) - P_{k-1}(x)$ between the predicted current $(k-1)$-th estimation $P_{k-1}(x)$ and the ground truth data $P_g(x)$. The objective of each CNN in the *k*-th local level is shown as follows:

$$\mathbf{L}_k^* = \arg\min_{\mathbf{L}_k} \| \Delta P_k(x) - \mathbf{L}_k(x) \|_2^2 . \quad (5)$$

After getting the last tiny update $\Delta\widetilde{P}_n$ from the *n*-th local level, then we update the new estimation in a cascade way as follows:

$$P_n = P_0 + \Delta\widetilde{P}_1 + ... + \Delta\widetilde{P}_n . \quad (6)$$

2.4. The multi-level regression

The different size of input regions of network in first global level can cover many possible conditions, it not only provides a robust initial estimation but also gives a very useful prior for the following estimations. The initial estimation is robust, but it is not accurate enough, so as an effect, the following local levels networks are proposed to learn the refined tiny steps between the current estimation and the ground truth. But few local levels are required because the steps are not large. Therefor these local levels network are only allowed to refine the initial prediction in a very tiny range (Fig. 4).

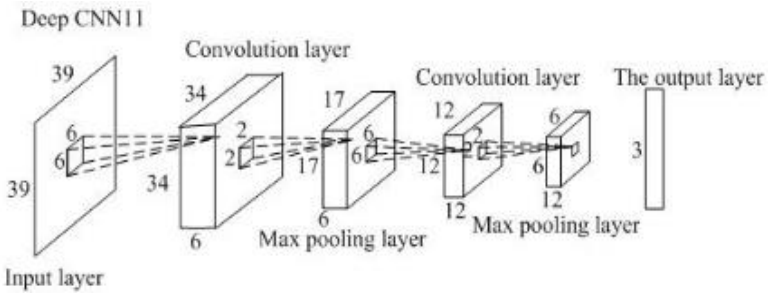


Fig. 4. The structure of the first convolutional neural network in the first level. Sizes of input regions, convolution, max pooling and the vector of the output layers are illustrated by cuboids whose length, width, and height denote the number of maps and the size of each map respectively. Local receptive fields of neurons in different layers are illustrated by small squares in the cuboid

We adopt a multi-level regression to combine the estimations of different levels and it effectively improves the estimation. Finally, the predicted head pose estimation is formulated in a cascade way as follows:

$$P_n = \frac{(\tilde{P}_1)_1 + (\tilde{P}_1)_2 + \ldots + (\tilde{P}_1)_{n_1}}{n_1} + \sum_{k=2}^{n} \frac{(\Delta\tilde{P}_k)_1 + (\Delta\tilde{P}_k)_2 + \ldots + (\Delta\tilde{P}_k)_{n_2}}{n_2}, \tag{7}$$

for a cascade with n_i predicts at level i. In our work, $n = n_i = 3$. Obviously, the first average term of (7) is the value of the first global level, i.e., the absolute head pose estimation, while the second term is the sum of the average steps in each local level, i.e., the refined improvements. Obviously, (6) and (7) are equivalent.

2.5. Network structure selection and discussion

Network structure selection. There are three leading factors about selecting this network for head pose estimation. First, estimating head pose is a difficult task and needs deep level network. The network increases the nonlinearity of the features and represents the relationship between image space and the head pose space. Second, the network is necessary since the estimation of the single level network is rough and inaccurate. Third, the structure of the network in each level based on two considerations, i.e., different input regions and low to high resolution version input image, which can effectively improve the performance.

Differences with traditional methods. Our proposed approach is clearly different from the traditional methods, their two main differences: Firstly, the traditional methods adopt linear function mapping from feature space to head pose space, while our network learns a highly nonlinear multi-level regressor. Secondly, the traditional methods employ the mean value or a random value as the initial estimation, while our proposed network estimates head pose step by step without initialization and facial feature points.

Differences with deep auto-encoder network [14]. Both deep auto-encoder network and our proposed network are used to estimate head pose. The differences between them are mainly on two reasons: Firstly, auto-encoder network for head pose estimation is limited in some angles of yaw. Our network can estimate three angles. Secondly, in deep auto-encoder network, the yaw angle is limited to [–45°, 45°]. The estimation may be not very robust when the angle changes to large, such as in [–90°, 90°]. Our network can predict three angles together even if with extreme angles.

3. Implementation details

The input image of our network is grey, recorded as $x(h, w)$, where h and w are the height and the width respectively. The convolutional layer is denoted by $C(k, m)$, k is the size of the square convolutional kernels and m is the number of map features. Let (h, w, m) represents m maps from the previous layer of size h by w, then the convolutional operation is namely $C(k, m)$,

(8) $$y_{i,j}^{t} = \tanh\left(\sum_{r=0}^{m-1}\sum_{u=0}^{k-1}\sum_{l=0}^{k-1} x_{i+u,j+l}^{r} \cdot \mathbf{W}_{u,l}^{(r,t)} + \mathbf{b}^{t}\right),$$

where x and y are the output of the previous and current layers, respectively, $i = 0, 1, 2, ..., h-k$, $j = 0, 1, 2, ..., h-k$, $\mathbf{W}$ is weight, $\mathbf{b}$ is offset term, and $\tanh$ presents the activation function which is usually nonlinear. Most generally, $\tanh$ is defined as $\tanh(x) = \dfrac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$.

In a general way, m maps in the previous layer are concerned with m square convolutional kernels. An activation function $\tanh$ is used after making sum of the output maps and the bias $\mathbf{b}$. In different convolutional level, the sets of the kernels and the bias are different, respectively.

The pooling layer is denoted as $P(s)$, where s is the size of the square pooling regions. In principle, there are two kinds of pooling operation, i.e., the mean pooling and the max pooling. In our work, the max pooling is employed. The coefficients in this layer are organized in a similar way as weights in the previous convolutional layer. The pooling results are multiplied with a gain coefficient $\mathbf{g}$ and shifted by a bias $\mathbf{b}$, and a $\tanh$ non-linear activation function is used after taking the max pooling operation. The pooling operation $P(s)$ is formulated as following:

(9) $$y_{i,j}^{t} = \tanh(\mathbf{g}^{t} \cdot \max_{0 \le u,l \le s} \{x_{i \cdot s+u, j \cdot s+l}^{i}\} + \mathbf{b}^{t}).$$

The fully connected layer is denoted as $F(n)$, where n and m are the numbers of neurons in the current layer and previous layer, respectively,

(10) $$y_{j} = \tanh(\sum_{i=0}^{m-1} x_{i} \cdot w_{i,j} + b_{j}), \quad j = 0, 1, 2, ..., n-1.$$

Input ranges and parameter setting. The normalized input ranges of CNNs in the first level are shown in Fig. 4. As can be seen in Fig. 3, the input regions and the convolutional operations of the network are demonstrated in Table 1.

Table 1. Settings of the network. $I(.,.)$ demonstrates the region of input image, $C(.,.)$ draws the convolutional operation, $P(.)$ represents the max pooling and $F(.)$ denotes the output vector

Level	Network	Layer0	Layer1	Layer2	Layer3	Layer4	Layer5
$\mathbf{L}_1$	CNN11	$I(39, 39)$	$C(6, 6)$	$P(2)$	$C(6, 12)$	$P(2)$	$F(3)$
	CNN12	$I(31, 39)$	$C(6, 6)$	$P(2)$	$C(6, 12)$	$P(2)$	$F(3)$
	CNN13	$I(31, 39)$	$C(6, 6)$	$P(2)$	$C(6, 12)$	$P(2)$	$F(3)$
$\mathbf{L}_2$	CNN21	$I(60, 60)$	$C(9, 10)$	$P(2)$	$C(9, 20)$	$P(2)$	$F(3)$
	CNN22	$I(48, 60)$	$C(9, 10)$	$P(2)$	$C(9, 20)$	$P(2)$	$F(3)$
	CNN23	$I(48, 60)$	$C(9, 10)$	$P(2)$	$C(9, 20)$	$P(2)$	$F(3)$
$\mathbf{L}_3$	CNN31	$I(80, 80)$	$C(13, 10)$	$P(2)$	$C(13, 20)$	$P(2)$	$F(3)$
	CNN32	$I(64, 80)$	$C(13, 10)$	$P(2)$	$C(13, 20)$	$P(2)$	$F(3)$
	CNN33	$I(64, 80)$	$C(13, 10)$	$P(2)$	$C(13, 20)$	$P(2)$	$F(3)$

With the purpose of training a reliable and robust network, we perturb the train images by changing the translation, rotation and scaling. Head pose estimation need to be learned of the robust network including the weigh $\mathbf{W}$, the gain $\mathbf{g}$ and the

bias **b**. These parameters are initialized by a random function and learned by stochastic gradient descent algorithm. Learning rate η is also an important parameter which need to manually set, we set η=0.0001 according lots of experiments. What is more, numepoches=100 and batchsize=100.

4. Experiments

In this section, face datasets, methods for comparisons and evaluation strategy are introduced in detail primarily, then the performance of each step of the deep model is studied, and finally the experiment results of our method compared to other existing methods are demonstrated.

4.1. Datasets, methods for comparison and evaluation strategy

The set used for our proposed network contains 14144 images, due to some links of the datasets have failed, so some database only part. The training set of our network contains 1120 images of FERET [15], 242 images of Imm_face [16], 974 images of ORL [17], 2800 images of FEI [18], 576 images of INDINA[19], 1092 images of Weizmann, 1167 images of MultiPIE (14%) [20], 2232 images of Pointing'04 [21], 1065 images of UMIST [22] and 282 images of Multifacepose [23]. The testing set contains 837 images of Pointing'04 (20%), 280 images of FERET (20%) and 1027 images of MultiPIE (9%).

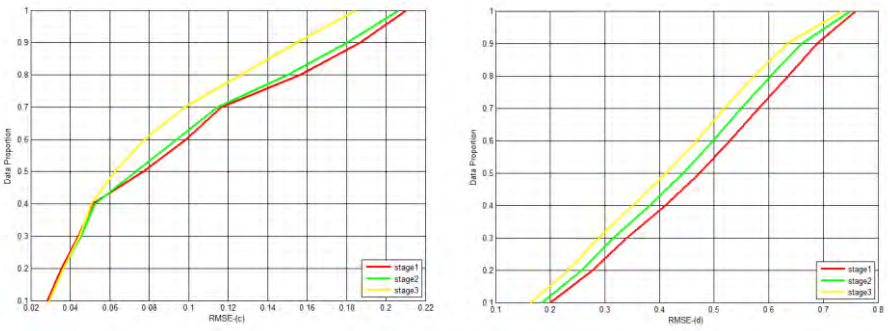
The output vector is expressed as the roll angle (in-plane rotation), the yaw angle (left-right rotation) and the pitch angle (up-bottom rotation). In our experiments, each image in the training set is manually labelled with three angles. The 181 roll angles {–90°, –89°, –88°,..., 0°,..., 88°, 89°, 90°} are labelled by computing the angle between two facial feature points of eyes, 13 yaw angles are re-labelled as {–90°, –75°, –60°, –45°, –30°, –15°, 0°, 15°, 30°, 45°, 60°, 75°, 90°} and 9 pitches angles are re-labelled as {–90°, –60°, –30°, –15°, 0°, 15°, 30°, 60°, 90°} according to the labelled datasets, and all of these angles are normalized to [–1, 1], we set the angle to 0 if this angle is not labelled in the datasets.

We will compare our proposed method to the existing methods as follows. Softmax is a classification method. Neural Networks (NN) [24] estimates head pose by minimizing conditional probability function. SPAE [14] progressively converts the non-frontal face images to the frontal ones. The experimental result of this method is better than many algorithms.

The each level performance of the network is measured with the Root Mean Squared Error (RMSE). The performance of the network is measured the classification accuracy of the estimated head pose. It includes seven measures, one for the yaw angle, one for the pitch angle, one for the roll angle, one for both of the yaw and the pitch angle, one for both of the yaw and the roll angle, one for both of the roll and pitch angle, one for all of them.

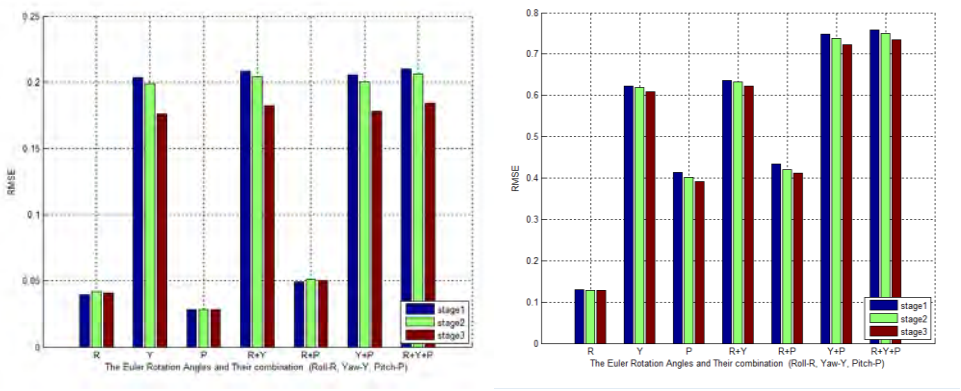
4.2. Investigation of each stage

Our proposed network consists of three levels. Therefore, we investigate how convolution neural networks in each level contribute to the performance improvement for the estimation. The experiments are expressed on two datasets in terms of average RMSE of three Euler rotation angles. The assessments of performance are shown in Figs 5 and 6, where "stage 1, 2, 3" represent the estimation result in each level respectively.



(a) (b)

Fig. 5. The comparison of each step on FERET and Pointing'04 Databases; (a) and (b) demonstrate the RMSE of the three angles in each step



(a) (b)

Fig. 6. The comparison of each step on FERET and Pointing'04 Databases. Both (a) and (b) display the RMSE of each dimension and their combinations of the three angles vector in each step

As shown in (a), (b) of Fig. 5, we compute the average RMSE of the angle vector at each step of the networks which is improved clearly. As can be seen, on FERET datasets, the RMSE improvements of Level 1 and Level 2 are tiny, but result of Level 3 is improved a lot. On Pointing'04 datasets, the improvements are smooth at each stage. Because we use the preliminary estimation rather than a random initial estimation at Stage 1, the RMSE of the estimation is improved up to about 2%.

In Fig. 6, we compute the average RMSE of each angle and their combinations in each level of the networks. Compared with Pointing'04 datasets, it is easy to find that the improvement on FERET is more obvious. However, the RMSE of the roll

angle at Stage 1 on FERET datasets is a little bit larger than at Stage 2, but it does not affect the final results.

This improvement root in two reasons, better global information and feature of head pose. The global level network handles large variation integrally and receives a good estimation compared to the mean value, networks at Level 2 and Level 3 are designed to refine the rough estimation length by length. In order to well capture the subtle variation, higher and higher resolution face images are used in Level 2 and Level 3, respectively.

4.3. Comparisons on FERET and MultiPIE datasets

We compare our method with some methods on both FERET and MultiPIE datasets [1]. The accuracy of the yaw angles of the proposed network are shown in Tables 2 and 3.

Table 2. Comparison with the existing method on FERET datasets

Methods		Softmax	NN	SPAE	Our method
Probe pose	–30°	51.6%	37.5%	98%	94.5%
	–15°	54.8%	47.8%	99%	99.2%
	15°	55.2%	48.3%	100%	100%
	30°	52.8%	38.0%	99%	99.2%
Average		53.6%	42.9%	96.4%	98.5%

In Table 2, Softmax performs the worst because of the limitation of capturing complex nonlinearity. NN performs a little better than Softmax. However, both of them are still worse than SPAE, it is possibly because SPAE are proposed with deep network. Our method outperform the compared SPAE method, it is with an improvement by 1.7%. In Table 3, NN performs still the worse. SPAE performs better than NN but worse than our method. Our method also outperforms the compared methods, and it is with an improvement by 1.8%. In Tables 2 and 3, we can see that the estimation with extreme angle is poor; this may be due to the less training samples and so on.

Table 3. Comparison with the existing methods on MultiPIE datasets

Method		NN	SPAE	Our method
Probe pose	–45°	35.2%	84.9%	85.1%
	–30°	46.1%	92.6%	95.5%
	–15°	51.2%	96.3%	98%
	15°	50.3%	96.3%	98.7%
	30°	48.6%	94.3%	96.5%
	45°	35.6%	84.4%	85.7%
Average		44.5%	91.4%	98.5%

Table 4. Comparison with the existing methods on Pointing'04 datasets.

Method	Accuracy (%)						
	Yaw	Pitch	Roll	Yaw+Pitch	Yaw+Roll	Roll+Pitch	Yaw+Pitch+Roll
Our method	71.4	73.5	99.1	52.8	70.2	72.5	52.2

In addition, we compute the accuracy of the angles and their combinations on Pointing'04 datasets in Table 4. Compared with the methods that predict two or one

head pose angles only, it is easy to find that the accuracy of our method performs a little poorly. This can be understood. This is because the number of estimated head pose angles and the accuracy rate are interacting to each other. However, our proposed approach still has progress. Firstly, it gives a try to predict all the three head pose angles together directly. Secondly, it does not need the initialization and facial feature location accuracy. What demand add is, because it is harsh for the intervals of the roll angle is 1°, we allow the max error is 5°in our experiments.

5. Conclusions

In this paper, we propose a new approach for head pose estimation based on a robust convolutional neural network with three designed levels in a cascade way. There are two main targets in our approach. The first is to propose a method for estimating all the three head pose angles together from images. The second is to present a method for estimating the angles without initialization and facial feature points' location. In this way, the proposed deep network not only alleviates the problem of estimating three head pose angles, but also gives a resolution of predicting head pose without initialization and facial feature points' location. The proposed method is compared with several head pose estimation algorithms on challenging datasets. Experimental results show that the proposed method performs better than the compared methods.

***Acknowledgments*:** This work was supported in part by the National Natural Science Foundation of China (Grant No 61375038) and Applied Basic Research Programs of Sichuan Science and Technology Department (Grant No 2016JY0088).

References

1. Chutorian, M. E., M. Trivedi. Head Pose Estimation in Computer Vision: A Survey. – IEEE Trans Pattern Analysis and Machine Intelligence, Vol. **31**, 2008, No 4, pp. 607-326.
2. Huang, J., X. Shao, H. Wechsler. Face Pose Discrimination Using Support Vector Machines (SVM). – In: International. Conf. Pattern Recognition, Vol. **1**, 1998, No 4, pp. 154-156.
3. Li, Y., S. Gong, J. Sherrah, H. Liddell. Support Vector Machine Based Multi-View Face Detection and Recognition. – Image and Vision Computing, Vol. **22**, 2004, No 5, pp. 413-427.
4. Srinivasan, S., K. L. Boyer. Head Pose Estimation Using View Based Eigenspaces. – In: International. Conf. Pattern Recognition, Vol. **4**, 2002, No 4, pp. 302-305.
5. Kruger, N., M. Potzsch, C. V. D. Malsburg. Determination of Face Position and Pose with a Learned Representation Based on Labeled Graphs. – Image and Vision Computing, Vol. **15**, 1997, No 8, pp. 665-673.
6. Gee, R. C. Determining the Gaze of Faces in Images. – Image and Vision Computing, Vol. **12**, 1994, No 94, pp. 639-647.
7. Yao, P., G. Evans, A. Calway. Using Affine Correspondence to Estimate 3-D Facial Pose. – In: International Conf. Image Processing, Vol. **3**, 2001, pp. 919-922.
8. Sherrah, J., S. Gong. Fusion of Perceptual Cues for Robust Tracking of Head Pose and Position. – Pattern Recognition, Vol. **34**, 2001, No 8, pp. 1565-1572.

9. O s a d c h y, Y. L. C. M., M. L. M i l l e r. Synergistic Face Detection and Pose Estimation with Energy-Based Models. – Journal of Machine Learning Research, Vol. **8**, 2007, No 1, pp. 1017-1024.
10. L u o, P., X. W a n g, X. T a n g. Hierarchical Face Parsing Via Deep Learning. – CVPR, 2012, pp. 2480-2487.
11. K r i z h e v s k y, I. S., G. E. H i n t o n. ImageNet Classification with Deep Convolutional Neural Networks. – Advances in Neural Information Processing Systems, Vol. **25**, 2012, No 2.
12. S u n, Y., X. W a n g, X. T a n g. Deep Convolutional Network Cascade for Facial Point Detection. – In: Conference on Computer Vision and Pattern Recognition, Vol. **9**, 2013, No 3, pp. 3476-3483.
13. E i g e n, D., C. P u h r s c h, R. F e r g u s. Depth Map Prediction from a Single Image Using a Multi-Scale Deep Network. – Eprint Arxiv, 2014, pp. 2366-2374.
14. K a n, M., S. S h a n, H. C h a n g, X. C h e n. Stacked Progressive Auto-Encoder for Face Recognition. – In: IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1883-1890.
15. **http://www.frvt.org/**
16. **http://www2.imm.dtu.dk/~aam/**
17. **http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html**
18. **http://fei.edu.br/~cet/facedatabase.html**
19. **htttp://vis-www.cs.umass.edu/~vidit/IndianFaceDatabase/**
20. **http://www.multipie.org/**
21. **http://www-prima.inrialpes.fr/Pointing04/data~face.html**
22. **http://www.pudn.com/downloads628/sourcecode/graph/detail2553627.html**
23. **http://www.eecs.qmul.ac.uk/~andrea/spevi.html**
24. B e n g i o, Y. Learning Deep Architectures for AI. – Foundations and Trends in Machine Learning, Vol. **2**, 2009, No 1, pp. 1-127.