AMSE JOURNALS-2016-Series: Advances B; Vol. 59; N°1; pp 113-130
Submitted July 2016; Revised Oct. 15, 2016, Accepted Dec. 10, 2016

Scale Invariant Constrained Deep Network for Head Pose Estimation

*Jiao Bao, **Mao Ye

*School of Computer Science and Engineering, Center for Robotics,
Key Laboratory for NeuroInformation of Ministry of Education,
University of Electronic Science and Technology of China,
China, School of Computer Science and Engineering, University of Electronic Science and Technology of China, Western High-Tech Industrial Zone, Chengdu(shazhijv@126.com)
** School of Computer Science and Engineering, Center for Robotics,
Key Laboratory for NeuroInformation of Ministry of Education,
University of Electronic Science and Technology of China,
China, School of Computer Science and Engineering, University of Electronic Science and Technology of China, Western High-Tech Industrial Zone, Chengdu(cvlab.uestc@gmail.com)

Abstract

Head pose estimation plays a significant role in face recognition, facial expression analysis and so on. However, it still faces great challenges due to the initialization and facial feature points location accuracy. Inspired by the observation that head pose variations change tinily and nonlinearly but smoothly and continuously, we present a new approach based on scale invariant constrained deep network for head pose estimation. The proposed network architecture consists of three levels and each level has three 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 layer step by step. At last, a multi-level regression is employed to combine the estimations from each level. The whole process is conducted in a deep structure to learn the three angles together directly. We perform large experiments on nine challenging datasets. The experimental results show that the proposed method perform remarkably better than the compared existing methods.

Key words

Head pose estimation, convolutional neural network, scale invariant constrained, multi-level regression, deep learning

1. Introduction

Head pose estimation has become a hot topic in pattern recognition and computer vision[1,5,6], because of its broad application in face recognition, the line of sight estimation and so on[1]. Accordingly, a lot of head pose estimation approaches have been proposed, and these approaches can be classified into two categories, i.e. one is based on the traditional machine learning method and the other is based on the deep network.

However, there are still many difficulties and challenges for head pose estimation. Firstly, lots of methods 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, many approaches mostly estimate head pose with one or two angles, or avoid head pose estimation with extreme angle.

In order to solve the problems mentioned above, we propose an new approach for head pose estimation based on scale invariant constrained deep network with three carefully designed levels. The head pose estimation is formulated as the multi-level regression problem towards three euler rotation angles. Different from traditional nonlinear regression approaches and deep auto-encoder network. The proposed method can predict three angles from images directly.

In our proposed deep network, head pose is estimated step by step. The primary contributions of this work are as follow:

1. A scale invariant constrained deep network is proposed for head pose estimation, the designed network estimates head pose step by step with one global level and two local levels.

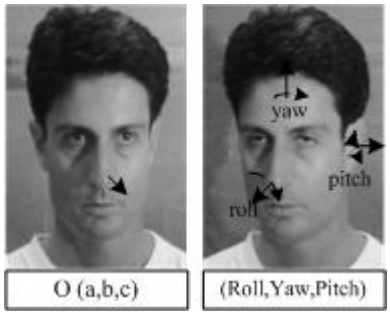
2. The three angles are estimated together from the face images directly by the proposed network without the initialization and facial feature points location accuracy.

The rest of the work is arranged as follows. Section 2 introduces some related works and discusses the advantages and disadvantages of these methods. Section 3 shows the framework of our network in detail. Section 4 demonstrates the implementation details. Section 5 presents the experiment results. Section 6 draws the conclusions.

2. Related work

Over last decade, significant head pose estimation methods have been proposed, and these methods can be classified into two categories as mentioned above.

The traditional methods can be summarized into three categories according to the input data requirement of the algorithm [1]. The first kind approaches is tracking methods and the combination



(a) (b)

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

of varieties algorithms [2]. The methods often rely on the initialization. The second one is flexible model methods and geometric methods [3]. These methods are simple and fast. But they usually need high resolution images for precise detection of facial feature point location. The third kind methods contain appearance template methods, detector array methods, nonlinear regression methods and manifold embedding methods [4,9]. These methods mentioned above are widely focus on for many years because of their suitable for single frame image of low resolution.

The deep network methods have been used in face detection, pose estimation, face parsing, image classification, facial point detection, depth map and so on [7]. However, such methods used for head pose estimation is few. Shan al et [8] proposes a deep progressive auto-encoder network for head pose estimation [8], it achieves great success by learning a non-linear function from the non-frontal face images to the frontal ones. However, the estimated results of this method is limited to, and it works not well with the extreme angle images. Our method belongs to this kind of method. The experiments on several public datasets show that the performance of the proposed method achieves good results.

3. The proposed deep network

In this section, we present a novel method based on deep network for head pose estimation. Firstly, we will give an overview of the proposed method in the first part; Secondly, we will

describe the components of the framework in details in the second, third and forth parts; Finally, we will illustrate the motivation of selecting this deep network and give some discussions in the fifth part.

3.1 Model overview

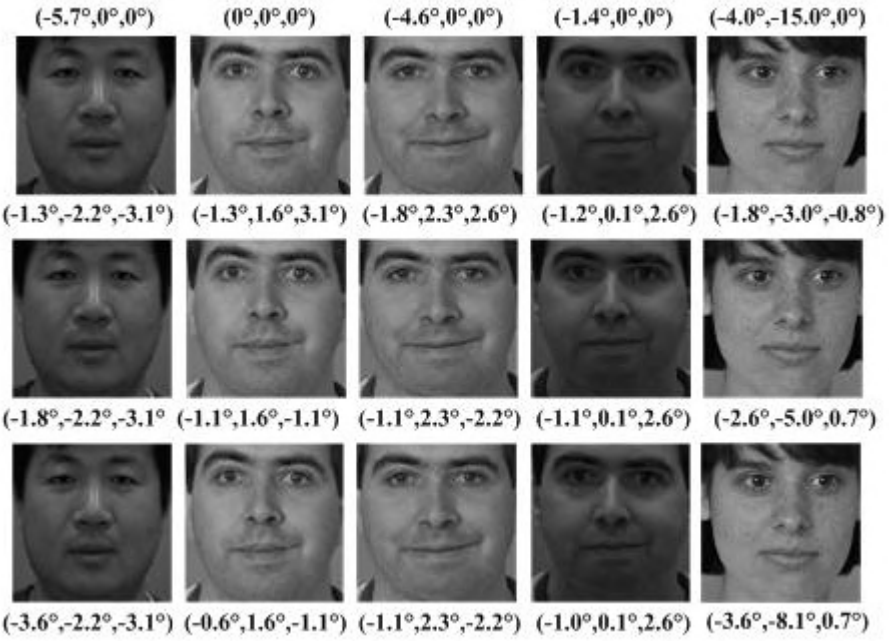


Fig. 2 Examples of head pose estimation. The top text: the labeled angles of images. The first row:roust rough estimations of our first global level networks. The second and the third rows: the tuned estimation results from our second and third local levels of the proposed networks. Obviously, the results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

In this work, we will use euler rotation angle model to describe head pose. Therefor, 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. There are three networks in each level. The whole process is conducted in a deep structure to learn the three angles together step by step. The estimations of these levels are combined by a multi-level regression. Furthermore, in order to prevent the errors to amplify in the network with deeper and deeper level, discriminant conditions are introduced to control these errors in each level.

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 convolution neural networks have a input layer, two convolutional layers, two pooling layers and a 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

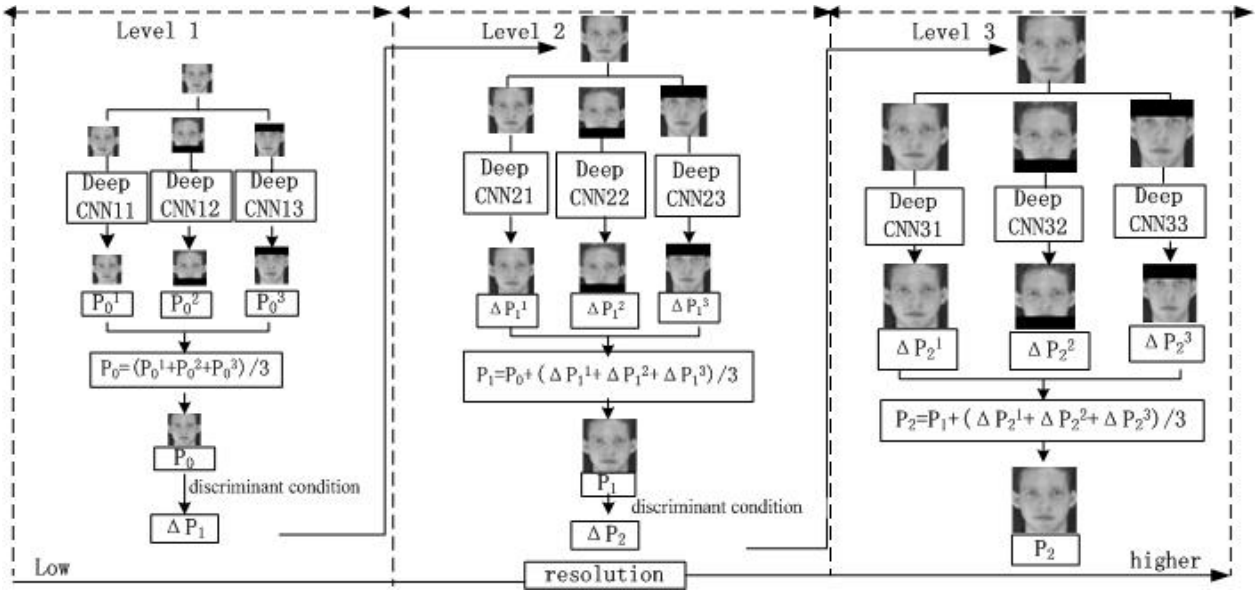


Fig. 3. Overview of the proposed deep network for head pose estimation. The input human face image is 2D. The first global level networks are denoted as CNN11, CNN12, and CNN13. Networks at this level predict a preliminary estimation P_0. Networks in the following levels are local, they refine the previous 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 level, the resolution of the input image is higher and higher.

to this three CNNs with different input regions, our network will be more robust to obtain a 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) endeavor to refine this preliminary 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 with different input regions are taken as the 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 of the global and local levels network. Apparently, this regression has two parts which correspond to the rough and adjustment estimations separately.

3.2 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 three normalized head pose angles respectively. In our work, the goal of the head pose estimation is to learn a mapping nonlinear relationship $\mathbf{F}$ from the human face image space to the head pose space directly:

$$\mathbf{F}: \quad x \to P_g(x). \tag{1}$$

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

$$\mathbf{F}^* = \underset{\mathbf{F}}{\operatorname{argmin}} \left\| P_g(x) - \mathbf{F}(x) \right\|_2^2. \tag{2}$$

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

The mean square error objective measures the relationships of the angles vector global. However, this metrics ignores the inner relationships of the vector. Motivated by this, we use the scale invariant error to measure the relationships among the three angles.

For the input image x and labeled ground truth $P_g(x)$, the estimation problem is formulated as minimizing the scale invariant error objective in the following:

$$\mathbf{F}^* = \underset{\mathbf{F}}{\operatorname{argmin}} \frac{1}{2n^2} \sum_{i,j}^{n} \left\| ((P_g(x))_i - (P_g(x))_j) - ((\mathbf{F}(x))_i - (\mathbf{F}(x))_j) \right\|_2^2. \tag{3}$$

where $(.)_i$ represents the ith element of a vector and n represents the dimension of the output vector, in our work, it is 3.

Exchanging the term of formula (3), then

$$\mathbf{F}^* = \underset{\mathbf{F}}{\operatorname{argmin}} \frac{1}{2n^2} \sum_{i,j}^{n} \left\| ((P_g(x))_i - (\mathbf{F}(x))_i) - ((P_g(x))_j - (\mathbf{F}(x))_j) \right\|_2^2 . \tag{4}$$

set $p(x) = F(x) - P_g(x)$ in formula (4), we get the following objective:

$$\mathbf{F}^* = \underset{\mathbf{F}}{\operatorname{argmin}} \frac{1}{2n} \sum_{i}^{n} p_i^2 - \frac{1}{2n^2} \sum_{i,j}^{n} p_{i,j}^2 . \tag{5}$$

combining like terns if the second part of formula (5), we obtain

$$\mathbf{F}^* = \underset{\mathbf{F}}{\operatorname{argmin}} \frac{1}{2n} \sum_{i}^{n} p_i^2 - \frac{1}{2n^2} (\sum_{i}^{n} p_i)^2 . \tag{6}$$

where function $\mathbf{F}$ contains lots of parameters, it is a collection of functions.

One of the most remarkable things is that the objective of the global level optimizes the difference between $P_g(x)$ and $f_i(x)$, while the objective of the local level optimizes the difference between $\Delta P_g(x)$ and $f_j(x)$, where $\Delta P_g(x)$ is the difference between $P_g(x)$ and the estimation learned from the previous level, $f_i(x)$, $f_j(x)$ represent the nonlinear mapping of the network in each 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 as in formula (7), where $(\widetilde{P}_1)_{n_i}$ is the estimations learned from the global level. 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.

$$P_0 = \frac{(\widetilde{P}_1)_1 + (\widetilde{P}_1)_2 + ... + (\widetilde{P}_1)_{n_1}}{n_1} . \tag{7}$$

in this work, $n_1 = 3$.

3.3 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 $\Delta\widetilde{P}_j(x)$ between the current estimation $P_{j-1}(x)$ and the ground truth $P_g(x)$.

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

$$\mathbf{L}_1^* = \underset{\mathbf{L_1}}{\operatorname{argmin}} \frac{1}{2n^2} \sum_i^n \left\| ((\Delta P_1)_i - (\Delta P_1)_j) - ((L_1)_i - (L_1)_j) \right\|_2^2. \tag{8}$$

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

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

Then for the kth the 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 $(k-1)th$ estimation $P_{k-1}(x)$ and the ground truth data. The objective of each CNN in the kth local level is shown as follows:

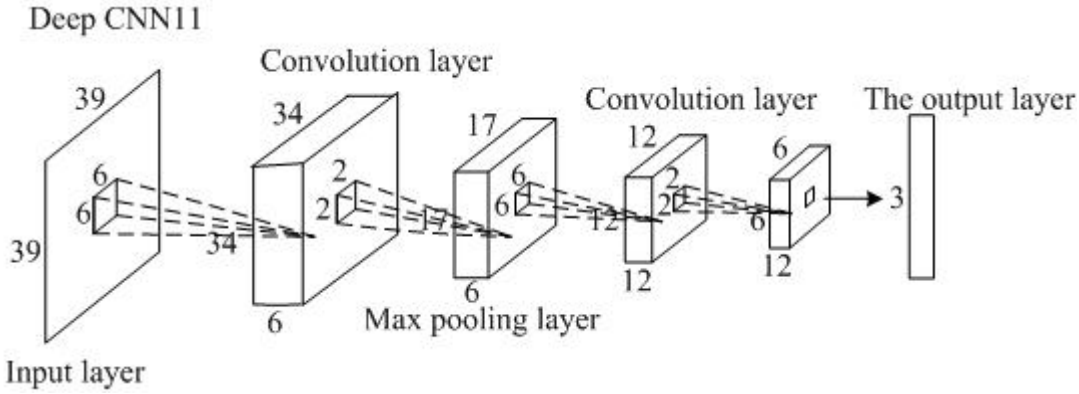


Fig.4. The structure of the first convolutional neural network in the global level. Sizes of input regions, convolution, max pooling and the vector of the output layers are illustrated by cuboids whose length, width, and height denote the number of maps and the size of each map respectively. Local receptive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

$$\mathbf{L}_k^* = \underset{\mathbf{L}_k}{\operatorname{argmin}} \frac{1}{2n^2} \sum_i^n \left\| ((\Delta P_k)_i - (\Delta P_k)_j) - ((L_k)_i - (L_k)_j) \right\|_2^2. \tag{9}$$

For formula (8), set $\Delta p_k(x) = \mathbf{L_k}(x) - \Delta P_k(x)$, for $\forall i, j$ we have the following formula:

$$\mathbf{L_k}^* = \underset{\mathbf{L_k}}{\operatorname{argmin}} \frac{1}{2n} \sum_i^n (\Delta p_k)_i^2 - \frac{1}{2n^2} \sum_{i,j}^n (\Delta p_k)_{i,j}^2. \tag{10}$$

Obviously, $\frac{1}{2n^2} \sum_{i,j} (\Delta p_k)^2_{\ i,j} = \frac{1}{2n^2} (\sum_i (\Delta p_k)_i)^2$ in formula (10), so we achieve:

$$\mathbf{L_k}^{*} = \underset{\mathbf{L_k}}{\operatorname{argmin}} \frac{1}{2n}\sum_{i}^{n}(\Delta p_k)_i^2 - \frac{1}{2n^2}(\sum_{i}^{n}(\Delta p_k)_i)^2. \tag{11}$$

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

$$P_n = P_0 + \Delta\widetilde{P}_1 + ... + \Delta\widetilde{P}_n. \tag{12}$$

3.4 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 the effect of 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.

We adopt a multi-level regression to combine the estimations learned from different levels. Finally, the predicted head pose estimation is formulated in a cascade way as follows

$$P_n = \frac{(\widetilde{P}_1)_1 + (\widetilde{P}_1)_2 + ... + (\widetilde{P}_1)_{n_1}}{n_1} + \sum_{i=2}^{n}\frac{(\Delta\widetilde{P}_k)_1 + (\Delta\widetilde{P}_k)_2 + ... + (\Delta\widetilde{P}_k)_{n_i}}{n_i}. \tag{13}$$

for a cascade with n_i predicts at level i. In this paper, $n_1 = n_i = 3$. Obviously, the first average term of formula (13) 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, formula (12) and formula (13) is equivalent.

3.5 Network structure selection

Network structure selection. There are three leading factors about selecting this deep network for head pose estimation. First, estimating head pose is a difficult task and needs deep network. The deep network increases the nonlinearity of the features and represents the relationship between image space and the head pose space. Second, the proposed network is necessary since the estimation of the single level network is rough and inaccurate. Third, the structure of our 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 proposed 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 deep network estimates head pose step by step without initialization and facial feature points.

Differences with deep auto-encoder network. Both deep auto-encoder network and our proposed deep 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 one angles of yaw. Our deep network can estimate three angles. Secondly, in deep auto-encoder network, the yaw angle is limited to $[-45^0,45^0]$. The estimation may be not very robust when the angle changes to large, such as in. Our deep network can predicts the three angles together even if with extreme angles.

Level	net	Layer0	Layer1	Layer2	Layer3	Layer4	Layer5
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)
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)
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)

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

4. Implementation details

The input image of our proposed 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 $C(s,m)$, s 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 $C(s,m)$

$$y_{i,j}^{t} = \tanh(\sum_{r=0}^{m-1}\sum_{k=0}^{s-1}\sum_{l=0}^{s-1} x_{i+k,j+l}^{r} \cdot \mathbf{W}_{k,l}^{(r,t)} + \mathbf{b}^{t}). \tag{13}$$

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

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 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

$$y_{i,j}^{t} = \tanh(\mathbf{g}^{t} \cdot \max_{0 \le k,l < s}\{x_{i \cdot s+k, j \cdot s+l}^{i}\} + \mathbf{b}^{t}). \tag{14}$$

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.

$$y_{j} = \tanh(\sum_{i=0}^{m-1} x_{i} \cdot w_{i,j} + b_{j}) \quad j = 0,1,2,...,n-1. \tag{15}$$

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

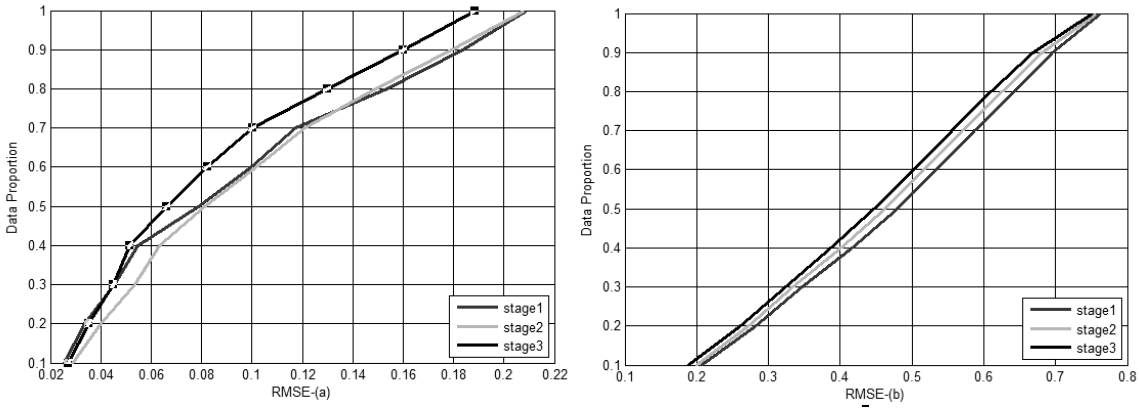
With the purpose of training a reliable and robust deep network, we perturb the train images by changing the translation, rotation and scaling. Head pose estimation need to be learned of the proposed deep network including the weigh $\mathbf{W}$, the gain $\mathbf{g}$ and the bias $\mathbf{b}$. These parameters are initialized by a random function and learned by stochastic gradient descent algorithm. What is more, learning rate η is also a important parameter which need to manually set. In our work, we set $\eta = 0.0002$ according lots of experiments.

5. 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 on three challenging are demonstrated.

5.1 Datasets, methods for comparison and evaluation strategy

The set used for our proposed network has 14144 images, due to some links of the datasets have failed, so some datasets only part. The training set of our network contains 1120 images from FERET [10], 242 images from Imm_face [11], 974 images from ORL [12], 2800 images from FEI [13], 576 images from INDINA [14], 1092 images from Weizmann, 1167 images from MultiPIE (14%) [15], 2232 images from Pointing'04 [16], 1065 images from UMIST [17] and 282 images from Multifacepose [18]. The testing set contains 837 images from Pointing'04 (20%), 280 images from FERET (20%) and 1027 images from MultiPIE (9%).



(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.

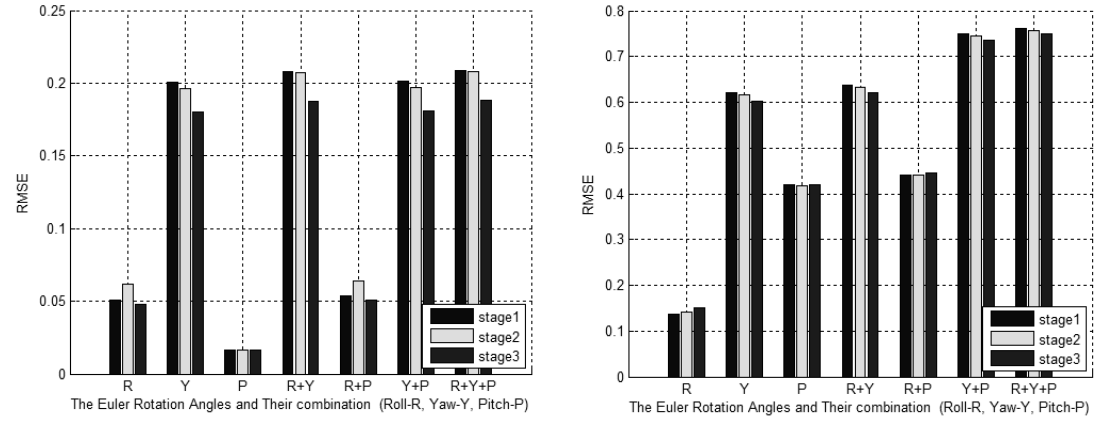
The output vector is expressed as the roll angle(in-plane rotation), the yaw angle(left-right rotation) and the pitch angle(up-bottom rotation). In our experiments, each image in the training set is manually labeled with three angles. The 181 roll angle $\{-90^0, -89^0, -88^0, ..., 0^0, ..., 88^0, 89^0,$

$90^0\}$ are labeled by computing the location between two facial feature points of two eyes in each image. 13 yaw angles are re-labeled as $\{-90^0, -75^0, -60^0, -45^0, -30^0, -15^0\ 0^0, 15^0, 30^0, 45^0, 60^0, 75^0, 90^0\}$ and 9 pitches angles are re-labeled as $\{-90^0, -60^0, -30^0, -15^0, 0^0, 15^0, 30^0, 60^0, 90^0\}$ according to the labeled datasets. All of these angles are normalized to $[-1, 1]$. We set the angle to 0 if this angle is not labeled in the database.

We will compare our proposed method to the existing methods as follows. **Asthana11** [19] proposes an automatic 3D pose normalization method which can synthesize a frontal human face image. **MDF** [20] generates virtual image of the probe human face image through the Morphable Displacement Field. **CCA** [21] maps images at two poses to a common space. **PLS** [22] projects image at two poses to a latent subspace. **GMA** [23] projects the images at all poses to a discriminative common space. **DAE** [24] directly converts the non-frontal face images to the frontal ones. **SPAE** [8] progressively converts the non-frontal face images to the frontal ones. **MLD-WJ** and **MLD-J** [9] estimation head pose angles based on multivariate label distribution.

The performance in each level of the network is measured with the root mean squared error(**RMSE**). The performance of the proposed network is measured with 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 pitch angle, one for both of the yaw and roll angle, one for both of the roll and pitch angle, one for all of them.

5.2 Investigation of each stage



(a) (b)

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

Our proposed deep network has three levels. Therefore, we investigate how 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 Fig.5 and Fig.6. Where "stage 1,2,3" represent the estimation result of the devised deep structure network at each level respectively.

As shown in (a), (b) of Fig.5, we computer the average RMSE of the angle vector in 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 is 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%. This demonstrates that the former global level mainly handles the large variations of head pose and the latter local levels precisely refine the head pose estimation with subtle variations.

In (a), (b) of Fig.6, we computer the average RMSE of each angle and their combinations in each level of the network. 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 estimation and local updates of head pose. The global level 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

Model		3D		2D					
Methods		Asthana	MDF	PLS	CCA	GMA	DAE	SPAE	**Our Method**
Probe Pose	-45^0	74.1%	78.7%	51.1%	53.3%	75.0%	69.9%	84.9%	**85.3%**
	-30^0	91.0%	94.0%	76.9%	74.2%	74.5%	81.2%	92.6%	**95.5%**
	-15^0	95.7%	99.0%	88.3%	90.0%	82.7%	91.0%	96.3%	**98.3%**
	15^0	95.7%	98.7%	88.3%	90.0%	92.6%	91.9%	96.3%	**98.8%**
	30^0	89.5%	92.2%	78.5%	85.5%	87.5%	86.5%	94.3%	**96.7%**
	45^0	74.5%	81.8%	56.5%	48.2%	65.2%	74.3%	84.4%	**85.9%**
	average	86.8%	96.7%	73.3%	73.5%	79.6%	82.5%	91.4%	**93.42%**

Tab. 2. Comparison with the existing methods on the MultiPIE datasets.

length. In order to well capture the subtle variation, higher and higher resolution face images are used in level 2 and level 3 respectively.

5.3 Comparisons on MultiPIE datasets

Firstly, we compare our proposed network with the existing methods on **MultiPIE** datasets [1]. The accuracy of the yaw angles of our network are shown in Table 2 .

As can be seen from Table 2, **CCA** and **PLS** perform the worst mainly because of the limitation of the fully unsupervised method and its limited ability of capturing complex nonlinearity. **GMA** performs better than **CCA** and **PLS**. However, both of them are still worse than **DAE** and **SPAE**, it is possibly because **DAE** and **SPAE** are proposed with deep structure network. Although both **DAE** and **SPAE** are used deep network, **SPAE** performs even better than **DAE**, this is thanks to **DAE** directly converts the non-frontal face images to the frontal ones. **Our method** outperforms better than all other compared methods, it is with an improvement by 1.7% .

Overall, our method achieves better performance on the datasets. There are two main reasons for these improvements. On one hand, the improvements come from the better initial estimation from the global level. On the other hand, a good estimation is obtained from the following two local levels step by step.

Furthermore, the training set of our method is composed of different datasets including large variations from expression, illumination, partial occlusions etc, while the major variation of FERET datasets is from the identity with similar pose, expression illumination and so on. This means that the distribution of training set of our network is different from the testing one. Even trained from a

Method	Accuracy(%)						
	Yaw	Pitch	Roll	Yaw+Pitch	Yaw+Roll	Pitch+Roll	Yaw+Pitch+Roll
Our Method	73.1	75.6	89.2	54.3	73.4	75.3	53.9
MLD-WJ	73.3	86.24	_	64.27	_	_	_
MLD-J	67.96	81.51	_	55.66	_	_	_

Table 3. The accuracy head pose estimation results on the Pointing'04 dataset.

different distribution, our deep network still works well, which demonstrates it is robust to the out-of-database scenario.

5.4 Comparisons on Pointing'04 datasets

Furthermore, we also evaluate our proposed method for head estimation on Pointing'04 datasets. These datasets consists of images with large variations in expression, illumination, partial occlusion. Both the accuracy and MAE results of head pose angles are assessed on this challenging datasets are shown in Table 3 and Table 4.

As shown in Table 3, we obtain the same result only in the yaw angle compared to the existing methods MLD-WJ and MLD-J. Our results is about 10% lower in other angles, and this is poor. However, compared with the methods which predict only one or two angles, our proposed approach can predict all the three angles together directly. That is to say, this is a new method, and this method has much room for improvement.

In Table 4, compared to MLD-J and MLD-WJ methods, the MAE of our method is employed with an average improvement about 0.15^0 of two angles. The performance is good except the yaw angle, and the reason might be that the human subjects often pose differently at yaw angles or lack of such samples in training set. However, it still performs much better than the kernel methods. In addition, because it is harsh for the intervals of the roll angle is 1^0, we allow the max error is 5^0 in our experiments. Then, we get a good result both on accuracy and MAE.

Conclusion

In this paper, we propose a new approach for head pose estimation based on scale invariant constrained deep network with three designed levels. There are two main targets in our approach.

Method	The Mean Absolute Error(^{0})						
	Yaw	Pitch	Roll	Yaw+Pitch	Yaw+Roll	Pitch+Roll	Yaw+Pitch+Roll
Our Method	5.7	2.7	5.96	6.3	5.99	5.97	6.5
MLD-WJ	4.24	2.69	_	6.45	_	_	_
MLD-J	5.02	3.54	_	7.94	_	_	_

Table 4. The MAE of head pose estimation results on the Pointing'04 data set.

The first is to propose a method for estimating all the three 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 alleviate the problem of estimating three head pose angles, but also give a resolution of predicting head pose without initialization and

facial feature points location. The proposed method is compared with several new head pose estimation algorithms on three challenging datasets. Experiments show that the proposed method performs significantly better than many compared methods.

Discussion

In this work, we proposed a new method based on scale invariant constrained deep network for head pose estimation. The method achieves good performance without initialization and facial feature points location. However, it still has disadvantages, such as lots of samples, time-consuming and so on. Therefore, our future work is main focuses on the following two aspects. On the one hand, we should try to solving the problem that needs lots of samples. On the other hand, we will take a chance on designing novel method for head pose estimation.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (61375038) and Applied Basic Research Programs of Sichuan Science and Technology Department (2016JY0088)

References

1. Murphy Chutorian E, Trivedi M., Head pose estimation in computer vision: a survey, 2008, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 31, no. 4, pp. 607-326.
2. A. Gee, R. Cipolla, Determining the gaze of faces in images, 1994, Image and Vision Computing, vol. 12, no. 94, pp. 639-647.
3. M. Ladies, J. C. Vorbruggen, J. Buhmann, J. Lange, V. Christoph, R. P. Wurtz, W. Konen, Distortion invariant object recognition in the dynamic link architecture, 1993, IEEE Trans. Comput.,vol.42, pp.300-311.
4. R. Gonzalez, R. Woods, Digital image processing 2nd ed, 2002, Inc., pp.582–584
5. T. Ling, M. J. Chen, Image denoising method using gradient matching pursuit, Mathematical modeling of engineering problems, 2016, vol.3, no.2,pp.53-56.
6. T. Ling, Super-resolution reconstruction method integrated with image registration, Review of computer engineering studies, 2015, vol.2, no.1, pp. 31-34.
7. Y. Sun, X. Wang, X. Tang, Deep convolutional network cascade for facial point detection, 2013, Conference on Computer Vision and Pattern Recognition, vol. 9, no. 3, pp. 3476-3483.

8. M. Kan, S. Shan, H. Chang, X. Chen, Stacked progressive Auto-encoder for Face Recognition, 2014, IEEE Conference on Computer Vision and Pattern Recognition, pp.1883-1890.
9. G. Xin, X. Yu, Head pose estimation based on multivariate label distribution, 2014, IEEE Con. On Computer Vision and Pattern Recognition , pp. 1837 – 1842.
10. http://www.frvt.org/.
11. http://www2.imm.dtu.dk/~aam/.
12. http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html.
13. http://fei.edu.br/~cet/facedatabase.html.
14. http://vis-www.cs.umass.edu/~vidit/IndianFaceDatabase/.
15. http://www.multipie.org/.
16. http://www-prima.inrialpes.fr/Pointing04/data-face.html.
17. http://www.pudn.com/downloads628/sourcecode/graph/detail2553627.html.
18. http://www.eecs.qmul.ac.uk/~andrea/spevi.html.
19. A. Asthana,T. K. Marks, M. J. Jones, K. H. Tieu, M. Rohith, Fully automatic pose invariant face recognition via 3d pose normalization, 2011, IEEE International Conference on Computer Vision, pp.937-944.
20. S. li, X. Liu, X. Chai, H. Zhang, S. Lao, S. Shan, Morphable displacement field based image matching for face recognition across pose, 2012, pp.102-115.
21. H. Hotelling, Relations between two sets of variates, 1936, Biometrika, vol.28, no.3/4, pp.321-377.
22. A. Sharma, D. W. Jacobs, Bypassing synthesis: pls for face recognition with pose, low resolution and sketch, 2011, IEEE Conference on Computer Vision and Pattern Recognition, pp.593-600.
23. A. Sharma, A. Kumar, D. W. Jacobs, Generalized multiview analysis: a discriminative latent space, 2012, IEE Conference on Computer Vision and Pattern Recognition.
24. Y. Bengio, Learning deep architectures for ai, 2009, Foundations and trends in machine learning, vol.2, no.1, pp.1-127.