Fine-Grained Head Pose Estimation Based on a 6D Rotation Representation with Multiregression Loss

Jin Chen[1], Huahu Xu[1(✉)], Minjie Bian[1,2], Jiangang Shi[3], Yuzhe Huang[1], and Chen Cheng[1]

[1] School of Computer Engineering and Science, Shanghai University, Shanghai, China
huahuxu@163.com
[2] State Grid Shanghai Municipal Electric Power Company, Shanghai, China
[3] Shanghai Shangda Hairun Information System Co., Ltd., Shanghai, China

Abstract. Estimating the head pose is vital in action evaluation since it has extensive applications such as in automobile driver-assistance systems, performance evaluations of athletes and customers' attention in retail stores. It is difficult to predict the head orientation from an RGB image by deep learning more accurately. We propose 6DHPENet, a fine-grained 6D head pose estimation network, to estimate the 3D rotations of the head. First, the model adopts a 6D rotation representation for 3D rotations as training objective to guarantee effective learning. 6D rotation representation is a continuous and one-to-one mapping function for 3D rotations. Second, achieving 3D facial landmarks from real-time activities consumes more time and is subject to frontal views. We drop the 3D facial landmarks to enhance the adaptability and generalization ability in various application scenes. Third, after the last convolution extraction layer, a squeeze-and-excitation module is introduced to construct both the local spatial and global channel-wise facial feature information by explicitly modeling the interdependencies between the feature channels. Finally, a multiregression loss function is presented to improve the accuracy and stability for a full-range view of the head pose estimation. In addition, our method is compact and efficient for mobile devices because of the lightweight CNN backbone. The quantitative experiment results trained on 300W-LP datasets show the superior performance of our 6D rotation representation-based multiregression fine-grained method on the AFLW2000 and BIWI datasets.

Keywords: Head pose estimation · 6D rotation · Fine-grained image analysis · Multiregression loss · Landmark-free method

1 Introduction

Head pose estimation (HPE) estimates the 3D rotations of heads from a single RGB image or videos. It has a wide range of applications, such as the evaluation of athletes' performance [5] in sports such as diving and skiing, head orientation estimations for

H. Gao et al. (Eds.): CollaborateCom 2022, LNICST 461, pp. 231–249, 2022.
https://doi.org/10.1007/978-3-031-24386-8_13

occluded pedestrians for the intention recognition of pedestrians [4], virtual and augmented reality [3] and other human attention modeling. For instance, in an automobile driver-assistance system [3], head pose estimation of the driver is an essential method to monitor whether the driver is engaged in fatigued driving or inattentive behavior. Due to the high practical value of HPE, it is a crucial task in the field of computer vision, especially in the area of action evaluation.

The camera-relative HPE is defined as the computation of the 3D rotation matrix for a head pose in the projection from the 3D space to the image plane. One of the important steps in the work is to compute the rigid Euclidean transformation from the 3D points in the world coordinate system to the corresponding 2D points in the camera coordinate system [1]. Inspired by the advancements in deep learning technology, many approaches have leveraged convolutional neural networks (CNNs) to address the problem of HPE. Overall, the research approaches can be classified into two main strategies: one is with 3D facial landmarks, and the other is without 3D facial landmarks. Some researches [3, 6] use facial landmark detection to establish a 2D-3D correspondence matching regression between the image and the 3D ground-truth landmarks. A standard human head 3D model [7] is used to imitate the real 3D face parameters, and then the 3D rotations of the heads are computed. However, this strategy has difficulty extracting key feature points from large poses such as profile views because of the occlusion of the facial features, which leads to more computation and larger models.

As a consequence, many landmark-free approaches [8–10] have been proposed to solve this issue. The head pose is estimated directly from a single image by the CNN regression models. Instead of predicting the 3×3 rotation matrix directly, they always choose other representations for the 3D rotations, as formulated in the 3D or 4D representations, such as the Euler angles and quaternions. Euler angles [11] are the most widely used because of their intuitiveness and simplicity of expression with only three elements, the angle of yaw, pitch and roll. However, there are some limitations for Euler angles. One is the ambiguity problem in terms of the gimbal lock [1]. Gimbal locking means that when two rotating axes become parallel, one degree of freedom (DOF) will be lost. Second, choosing different rotation orders will lead to different value of angles. Third, non-stationary properties [5] exist because the facial features do not change smoothly with respect to the angle size. The quaternion representation for 3D rotations has the antipodal problem [12]. When defining the representation space, $\left(q \in R^4, ||q||_2 = 1\right)$, q and $-q$ correspond to the same 3D rotation. In addition, the results from [13] proved that the 3D and 4D representations for 3D rotations are unsuitable for regression networks because of discontinuity. As demonstrated in these studies, using Euler angles or quaternion representations to annotate the head orientation is inappropriate for the regression in the CNN model. To solve these problems, we take the continuous representations (the 5D or 6D rotation matrix) to express the 3D rotation matrix to lower the error rate in neural networks. Furthermore, since real-time inferencing is necessary for many applications, including motion capture and generating attention maps for retail stores [14], it is necessary to adopt a more lightweight and mobile-friendly network as the CNN backbone to achieve faster speeds.

In this paper, we present a fine-grained 6D head pose estimation network (6DHPENet) that adopts a 6D rotation representation for 3D rotations as the training objective and make inferences from a single RGB image without 3D facial landmarks. The 6DHPENet predicts the value of the 3D rotation matrix of a head pose. It is trained on 300W-LP datasets and achieves the best result with an average mean absolute error (MAE) of 3.63 on the BIWI and AFLW2000 datasets. Specifically, the total MAE improves by 3.2% in the AFLW2000 datasets. There are four main innovations in the construction of the proposed 6DHPENet. First, the 6D rotation representation is composed by simply dropping the last column vector of the original 3×3 rotation matrix. Then a Gram-Schmidt-like process is used for mapping the 6D rotation representation to the original space. Second, to be friendly to mobile devices, we choose two lightweight CNN models, EfficientNet [15] and RepVGG [16], as the CNN backbone to extract the shallow and deep facial features. Third, to gain the fine-grained facial feature, the squeeze-and-excitation (SE) [17] module is embedded after the last CNN layer. To construct the local spatial and global channel-wise information, the SE module explicitly models the interdependencies between the feature channels. Finally, we use a multiregression loss function that contains the geodesic loss and the orthogonal loss to compute the difference between the predicted value and the ground-truth value in the training process for a gradient descent regression. The geodesic loss minimizes the angular difference between the ground-truth rotation matrix and the predicted one. The orthogonal loss constrains the orthogonality of the predicted rotation matrix caused by the calculation error in the Gram-Schmidt-like process to obtain better stability. In conclusion, our research contributions can be summarized as follows.

- **6D Rotation Representation for 3D Rotations**. The continuity of the 6D representation reduces the misleading neural networks, and dropping 3D facial landmarks enhances the generalization performance of the method.
- **Extraction of Fine-grained Feature Information**. The lightweight CNN backbone is introduced, and the SE module is embedded to construct local spatial and global channel-wise facial fine-grained feature information.
- **A Novel Multi-Regression Loss Function**. It contains the geodesic loss and the orthogonal loss. It improves accuracy and stability for the full-range view of the HPEs.
- **Excellent experiment results**. This shows that our approach based on a 6D rotation representation combined with the SE module and multiregression loss, is effective and suitable for fine-grained head pose estimation.

The remainder of the paper is organized as follows: Sect. 2 gives a brief review of the state-of-the-art head pose estimation methods. Section 3 presents the framework of proposed method and details of each part. Section 4 shows an experiment implementation and analysis on several public datasets. Finally, conclusions from our work are discussed in Sect. 5.

2 Related Work

2.1 Approaches for Discontinuous Representations

Head pose estimation has been actively researched over the past 25 years. For the research on monocular RGB images, there are several kinds of methods including classical methods [37, 38], geometric & deformable landmark-based methods [2] and regression & classification landmark-free methods [8–10, 18, 36, 41]. The traditional classical methods include template matching and cascaded detectors. Their characteristically take the discretized pose as a template compared to the input images. The geometric methods [39, 40] are also called the perspective to point (PnP) problems with 3D facial landmarks. In the existing landmark-free approaches, most of them are used to predict head pose from a discretized set of poses by regression and classification methods or multitasks methods [8–10, 18, 36]. In conclusion, these methods usually choose discontinuous annotations as the training objective of a head pose estimation.

For instance, Euler angles and quaternions are utilized as the training regression objective for most state-of-the-art methods. Hopenet [8] proposed a CNN model combined with a multiloss to predict head pose Euler angles directly from image intensities without key points. The multiloss network is composed of a pose bin classification and a regression component. Based on the same strategy as Hopenet WHENet [18] introduced a wrapped loss to improve the yaw accuracy for anterior views in a full-range HPE. Similarly, QuatNet [9] designs a multiregression loss that combines L2 regression with ordinal regression loss to address the non-stationary property in a HPE. Bin et al. [36] introduce a method using two-stage ensembles with average top-k regression. Despite the intuitiveness of Euler angles, it has been proven that four or fewer dimensional representations are discontinuous representations for 3D rotations.

2.2 Continuous Representations of 3D Rotations

In neural networks, the theoretical [23, 24] results suggest that functions that are smoothly or strongly continuous have a lower approximation error for a given number of neurons. Therefore, many researchers devote themselves to studying the theory of continuous representations of 3D rotations.

Wu et al. [21] studied the problem of restoring the orthonormality of a noisy rotation matrix by finding its nearest correct rotation matrix. Zhou et al. [13] present continuous representations for a general case of the n dimensional rotation group $SO(n)$, which is suitable for neural networks and shows that it needs at least 5 dimensions of information to achieve a continuous representation of rotations in 3D rotation space. Another innovation for Zhou et al. [13] is to propose a geodesic loss to minimize the angle error between two rotations. 6DRepNet [19] follows the approach by Zhou et al. [13] and engages a network to predict the 6D rotation representation for 3D rotations. Furthermore, Zhi et al. [20] proposed a deep network pipeline based on vector representation for a 3D rotation matrix with vector orthogonal constraints. Cao et al. [20] used three-vector annotations and illustrated that the Euler angle annotation has issues of discontinuity.

2.3 Fine-Grained Head Pose Estimation

To obtain better HPE accuracy by deep CNN networks, it is desirable to enhance feature aggregation and extract more effective local and global context information. There are several fine-grained HPE methods. Yang et al. [10] proposed a classification method that learns a fine-grained structure mapping for spatially grouping features before aggregation. Wu et al. [21] studied learning from a synergy process of 3D morphable models (3DMMs) and facial landmarks to predict complete 3D facial geometry, including 3D alignment, face orientation, and 3D face modeling.

For extraction of the fine-grained features, the existing methods need more annotations or more complex construction of networks. To explicitly model the interdependencies between the feature channels simply, Shen et al. [17] proposed a squeeze-and-excitation network (SENet) with an SE block that adaptively recalibrates the channel-wise feature responses. It has a more understandable network structure.

3 Model Framework

3.1 6DHPENet Overview

AS shown in Fig. 1, 6DHPENet is a solution to end-to-end 2D image-to-3D rotation correspondence learning for head pose estimation without 3D landmarks. It mainly consists of five modules.

CNN Backbone for Encoding Feature Space. The input of this network is a single RGB image I, which shows a cropped head. For the input image I, a CNN backbone is utilized to extract the shallow and deep facial features from the image and encode a feature space K, where $K \in R^{H \times W \times C}$. We choose RepVGG-b1g2 [15] and EfficientNet-B0 [16] as the CNN backbone.

Squeeze-and-Excitation Module for Embedding Fine-Grained Features. The feature space $K\left(K \in R^{H \times W \times C}\right)$ is passed into a squeeze-and-excitation module to obtain

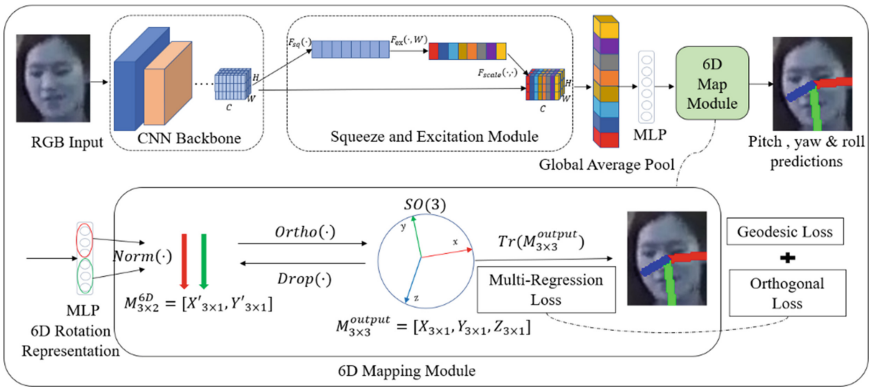

Fig. 1. An overview of 6DHPENet.

the fine-grained and reweighted feature space $\tilde{K}\left(\tilde{K} \in R^{H\times W\times C}\right)$. The squeeze-and-excitation module contains three steps: First, the input feature space K is squeezed into a channel-wise feature descriptor s by the average pooling function $F_{sq}(\cdot)$. Second, the channel-wise feature descriptor s is excited into a channel-wise weight descriptor z by $F_{ex}(\cdot, W)$. Third, the input feature space K is reweighted into the fine-grained feature space $\tilde{K}$ by $F_{scale}(\cdot, \cdot)$. In Sect. 3.2, we will introduce the details of $\tilde{K} = SE(K)$.

Output of a 6D Rotation Representation. After embedding the fine-grained feature space $\tilde{K}$, $\tilde{K}$ is sized into the feature space $F\left(F \in R^{1\times 1\times C}\right)$ by global average pooling (GLP) and reshaped into the fused feature vector f ($f \in R^{C}$). The fused feature vector f is sent into a multilayer perceptron (MLP) called the fully connected layer Z_{linear} that outputs 6 dims of neurons, called m_6. The output tensor m_6 is viewed as a 6D rotation representation matrix $M_{3\times 2}$. In Sect. 3.3, we will give the definition of 6D representation for 3D rotations.

6D Mapping Module. Through a Gram-Schmidt-like process $Ortho(\cdot)$ to map the 6D Matrix $M_{3\times 2}$ into the 3D rotations, $M_{3\times 3}(M_{3\times 3} \in SO(3), M_{3\times 3}M_{3\times 3}^{T} = I, \det(M_{3\times 3}) = 1)$, finally the Euler angles (the angles of yaw, pitch, roll) are computed by $Tr(M_{3\times 3}^{output})$ from the 3D rotations $M_{3\times 3}$. In Sect. 3.4, we introduce the details of the 6D mapping process.

Multiregression Loss Function. The multiregression loss function contains the geodesic loss L_{geo} and the orthogonal loss L_{ortho}. L_{geo} minimizes the angular difference between the ground-truth rotation matrix and the predicted rotation matrix. L_{ortho} constrains the orthogonality of the predicted rotation matrix caused by a calculation error by the Gram–Schmidt-like process to obtain better stability. The multiregression loss function computes the difference between the predicted value and the ground-truth value in the training process for the gradient descent regression.

3.2 SE Module for Embedding Fine-Grained Features

Embedding fine-grained features passes the feature space $K\left(K \in R^{H\times W\times C}\right)$ into a squeeze-and-excitation module to obtain the fine-grained and reweighted feature space $\tilde{K}\left(\tilde{K} \in R^{H\times W\times C}\right)$. The squeeze-and-excitation module contains three steps: First, the input feature space K is squeezed into a channel-wise feature descriptor s by global average pooling $F_{sq}(\cdot)$ in Eq. (6). $F_{sq}(\cdot)$ sums out the global feature information by no parameterization and reduces the characteristic dimensions. Second, the channel-wise descriptor s is excited into a channel-wise weight descriptor z by $F_{ex}(\cdot, W)$ in Eq. (7). Third, the input feature space K is reweighted into the fine-grained feature space $\tilde{K}$ by $F_{scale}(\cdot, \cdot)$ in Eq. (8).

$$s = F_{sq}(K) = \frac{1}{H\times W}\sum_{i=1}^{H}\sum_{j=1}^{W} K(i,j) \tag{6}$$

$$z = F_{ex}(s, W) = \sigma(g(z, W)) = \sigma(W_2(\delta(W_1(z, W)))) \tag{7}$$

$$\tilde{K} = F_{scale}(K, z) \tag{8}$$

In Eq. (7), σ is the sigmoid function, δ is the ReLU function, $W_{\mathrm{i}}(i = 1, 2)$ is the i^{th} fully connected layer, and W is the weight of excitation. $F_{ex}(\cdot, W)$ obtains the channel-wise weight z by reducing the channel dimensions by increasing the channel dimensions to achieve a more nonlinear processing capability. In Eq. (8), z is resized to the same size as K, and then a pixel-wise multiplication is conducted to reweight the feature space K into the fine-grained feature space $\tilde{K}$.

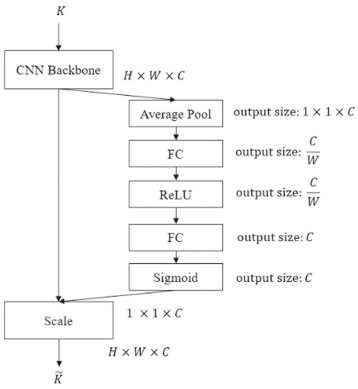

Fig. 2. Flowchart of the squeeze-and-excitation module.

The flowchart of the SE module is shown in Fig. 2. For the inputting feature space K, there are two stages for generating the output feature space $\tilde{K}$. In the first stage, the input feature space K is squeezed into a channel-wise feature descriptor $s \in R^{1\times1\times C}$ by global average pooling $F_{sq}(\cdot)$ in Eq. (6). $F_{sq}(\cdot)$ sums out the global feature information on the $H \times W$ dimensions. Second, the channel-wise descriptor s is excited into a channel-wise weight descriptor $z \in R^{1\times1\times C}$ by $F_{ex}(\cdot, W)$ in Eq. (7). There are four steps in $F_{ex}(\cdot, W)$. The factor W is set to 16 by experiment practice. The first step is an MLP layer that outputs $\frac{C}{W}$ dims of neurons to fuse the channel characteristics. The second step is a ReLU activation function. The third step is an MLP layer that outputs c dims of neurons to extract the channel-wise attention. The fourth step is a sigmoid function to normalize the value between[0, 1]. Finally, the channel-wise weight descriptor z is reshaped into $1 \times 1 \times C$ for the preparation of the pixel-wise multiplication. In the second stage, the input feature space K is reweighted into the fine-grained feature space $\tilde{K}$ by conducting a pixel-wise multiplication $F_{scale}(\cdot, \cdot)$ between the input feature space K and the channel-wise weight z in Eq. (8).

3.3 6D Representation for the 3D Rotations

Zhou et al. [13] presented continuous representations for the general case of the n dimensional rotation group $SO(n)$. We discuss the definition of the 6D rotation representation.

It is a continuous one-to-one mapping relationship between 6D rotation representation $M_{3\times2} = \left[\vec{a}_1, \vec{a}_2\right]$ and the 3D rotations $M_{3\times3} = \left[\vec{b}_1, \vec{b}_2, \vec{b}_3\right]$, where $\vec{a}_i, \vec{b}_i, i = 1, 2, \ldots, n$ are column vectors.

First, let the original space be $X = SO(3)$ and the representation space be $R = R^{3\times2}$. Given the 3D rotation matrix $M_{3\times3}$, we know that it is an orthogonal matrix and meets the conditions: $M \in R^{3\times3}$, $M_{3\times3} \in SO(3)$, $M_{3\times3}M_{3\times3}^T = I$, $\det(M_{3\times3}) = 1$. Therefore, the 6D rotation representation $M_{3\times2}$ should contain an orthogonalization process in the representation itself. Then we can define a mapping $G_{map}(\cdot)$ from the original space to the representation space by simply dropping the last column vector of the input 3D rotation matrix:

$$G_{map}\left(\left[\vec{b}_1, \vec{b}_2, \vec{b}_3\right]\right) = \left[\vec{b}_1, \vec{b}_2\right] \tag{1}$$

The set $G_{map}(\cdot)$ is a Stiefel manifold from the theory [25]. For the mapping f_{map} from the representation space to the original space, we can define the following Gram–Schmidt-like process:

$$f_{map}\left(\left[\vec{a}_1, \vec{a}_2\right]\right) = \left[\vec{b}_1, \vec{b}_2, \vec{b}_3\right] \tag{2}$$

$$\vec{b}_1 = Norm(\vec{a}_1) = \frac{\vec{a}_1}{||\vec{a}_1||} \tag{3}$$

$$\vec{b}_2 = Norm\left(\vec{U}_2\right) = \frac{\vec{U}_2}{\left|\left|\vec{U}_2\right|\right|}, \vec{U}_2 = \vec{a}_2 - \vec{b}_1 \cdot \vec{a}_2\vec{b}_1 \tag{4}$$

$$\vec{b}_3 = \vec{b}_1 \times \vec{b}_2 \tag{5}$$

Norm$(\cdot)$ denotes a normalization function by the L2-Norm. The third vector $\vec{b}_3$ can be calculated by the ordinary cross product $\vec{b}_1 \times \vec{b}_2$ in Eq. (5). The process of the Proof: for every $M_{3\times3} \in SO(3)$, $f_{map}\left(G_{map}(M_{3\times3})\right) = M_{3\times3}$ has been verified by the induction and the properties of the orthonormal basis vectors in [13, 26].

3.4 6D Mapping Module

From Sect. 3.3, we note that the 6D rotation matrix can be transformed into the 3×3 rotation matrix by a one-to-one mapping function $Ortho(\cdot)$ which equals Eqs. (2) and $Drop(\cdot)$ which equals Eq. (1). Thus, we introduce the steps of the 6D mapping process by using the theory of the definition of the 6D Representation for the 3D Rotations.

As shown in Algorithm 1, specifically, we split the output of the MLP layer into two column vectors $M_{3\times2}^{6D} = [X'_{3\times1}, Y'_{3\times1}]$ and then use a Gram–Schmidt-like process $Ortho(\cdot)$ in Eq. (2,3,4,5) to map the 6D rotation Matrix $M_{3\times2}$ into the 3D rotations $M_{3\times3}(M \in R^{3\times3}, M_{3\times3} \in SO(3), M_{3\times3}M_{3\times3}^T = I, \det(M_{3\times3}) = 1)$. The Gram–Schmidt-like process $Ortho(\cdot)$ has several steps. First, $X_{3\times1}$ is normalized by using Eq. (3) with $X'_{3\times1}$. Second, the second vector $Z'_{3\times1}$ is generated by using Eq. (4). Third, $Z_{3\times1}$ is normalized by using Eq. (3) with $Z'_{3\times1}$. Fourth, the third vector $Y_{3\times1}$ is generated by using Eq. (5). Fifth, the third vector $Y_{3\times1}$ is generated by using Eq. (5). Finally, $X_{3\times1}$, $Y_{3\times1}$ and $Z_{3\times1}$ are aggregated into the 3D rotation matrix $M_{3\times3}^{output}$.

Algorithm 1. 6D Map Module

Input: The 6 dims of neurons m_6 (output by the MLP layer).

Output: The 3D rotation matrix $M_{3\times3}^{output}$.

1: Split the m_6 into two column vectors ${X'}_{3\times1}, {Y'}_{3\times1}$.
2: Normalize ${X'}_{3\times1}$ into $X_{3\times1}$ using Equation (3) $X_{3\times1} = norm({X'}_{3\times1})$.
3: Generate the second vector ${Z'}_{3\times1}$ using Equation (4) ${Z'}_{3\times1} = CrossProduct(X_{3\times1}, {Y'}_{3\times1})$
4: Normalize ${Z'}_{3\times1}$ into $Z_{3\times1}$ using Equation (3) $Z_{3\times1} = norm({Z'}_{3\times1})$.
5: Generate the third vector $Y_{3\times1}$ using Equation (5) $Y_{3\times1} = CrossProduct(Z_{3\times1}, X_{3\times1})$
6: Aggregate $X_{3\times1}, Y_{3\times1}, Z_{3\times1}$ into the rotation matrix $M_{3\times3}$
7: **return** the 3D rotation matrix $M_{3\times3}^{output}$

Finally, to compare the result with other methods, the 3D rotation maxtrix $M_{3\times3}^{output}$ is transformed into the Euler angles, the angles of yaw, pitch and roll by $Tr(M_{3\times3}^{output})$. $Tr(M_{3\times3}^{output})$ means computing the Euler angles from a rotation matrix and the details have been shown in Slabaugh et al. [11].

3.5 Multiregression Loss Function

Our method is trained end-to-end. During the training process, we employed a novel multiregression loss function that contains the geodesic loss L_{geo} and the orthogonal loss L_{ortho} for the training objective. As shown in Eq. (9), the total loss is the addition of the L_{geo} and L_{ortho} with a weighted term α that is set to a small number whose range is between [0.1, 0.5].

$$L = L_{geo}\left(M_{predict}, M_{ground}\right) + \alpha \cdot L_{ortho}\left(M_{predict}\right) \tag{9}$$

Geodesic Loss. For a given 3D rotation matrix M, $M \in SO(3)$, can be represented by a rotation axis $\vec{\mu}$ and rotation angle θ according to Rodrigues' rotation formula [27]. Figure 3 shows the relationship between M and θ that equals Eq. (10).

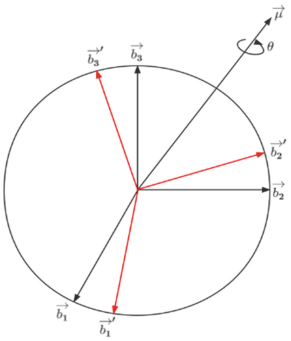

Fig. 3. Geodesic loss geometric meaning.

$$tr(M) = 1 + cos2\theta \tag{10}$$

Then the geodesic loss is defined as the minimal angular difference between two 3×3 rotation matrices, which calculates the angle between the predicted rotation matrix $M_{predict}$ and the ground truth matrix M_{ground}. It can be defined as:

$$L_{geo} = \cos^{-1}\left(\frac{\left(M_{00}^{''} + M_{11}^{''} + M_{22}^{''} - 1\right)}{2}\right) \tag{11}$$

$$M'' = M_{predict} M_{ground}^{T} \tag{12}$$

Orthogonal Loss. L_{ortho} constrains the orthogonality of the predicted rotation matrix caused by the calculation error on the Gram–Schmidt-like process. Given the predicted rotation matrix $M_{predict} = \left[\vec{b}_1, \vec{b}_2, \vec{b}_3\right]$, where $\vec{b}_{\mathrm{i}}, i = 1, 2, \ldots, n$ are column vectors, $M_{predict} \in R^{3\times3}$, $\vec{b}_{\mathrm{i}} \in R^{3\times1}$, the loss term is shown as follows:

$$L_{ortho} = MAE\left(\vec{b}_{\mathrm{i}}\vec{b}_{\mathrm{j}}, 0\right)(i \neq j, i \in [1, 2, 3], j \in [1, 2, 3]) \tag{13}$$

$$M_{predict} = \left[\vec{b}_1, \vec{b}_2, \vec{b}_3\right] \tag{14}$$

In Eq. (13), we adopt the mean absolute error loss function for the orthogonal loss to minimize the difference between the cross product of the column vectors and the zero value. If a calculation error does not exist, the cross product of the column vectors should be zero because the column vectors in the rotation matrix are mutually orthogonal.

3.6 Complete Structure of 6DHPENet

AS shown in Algorithm 2, for a single RGB image, we trained an end-to-end network 6DHPENet to predict the value of head pose $M_{3\times3}^{output}$. The raw form of the head pose is defined as the 3D rotation matrix $M_{3\times3}^{output}$ and can be transformed into other representations, such as Euler angles. First, the features $K(K \in R^{H\times W\times C})$ are extracted from I by the CNN backbone RepVGG-b1g2 or EfficientNet-B0. Second, the features K are reweighted into the fine-grained features $\tilde{K}$ by the SE module. The details are shown in Sect. 3.2. Third, $\tilde{K}$ is sized into the feature space $F\left(F \in R^{1\times1\times C}\right)$ by global average pooling. Fourth, the feature space F is flattened to the feature vector f $(f \in R^{C})$. Fifth, a Multilayer Perceptron (MLP) outputs 6 dims of neurons, called m_6. The tensor m_6 is viewed as a 6D rotation representation matrix $M_{3\times2}$. Finally, the 6D rotation representation matrix $M_{3\times2}$ is mapped into the 3D rotation matrix $M_{3\times3}^{output}$. The 6D mapping process is shown in Sect. 3.4.

Algorithm 2. Complete Structure of 6DHPENet

Input: A single RGB image I with a cropped face.

Output: The 3D rotation matrix $M_{3\times3}^{output}$.

1: Extract features $K(K \in R^{H\times W\times C})$ from I by the CNN backbone.

2: Reweight the features K into the fine-grained feature $\widetilde{K}$ by SE module: $\widetilde{K} = SE(K)$.

3: Size the $\widetilde{K}$ into feature space F $(F \in R^{1\times1\times C})$ by global average pooling.

4: Reshape feature space F into the feature vector f $(f \in R^{C})$.

6: The tensor $m_6 = Z_{linear}(f)$ by the Multilayer Perceptron (MLP) that outputs 6 dims of neurons.

7: m_6 is viewed as a 6D rotation representation matrix $M_{3\times2}^{6D} = [X'_{3\times1}, Y'_{3\times1}]$.

8: $M_{3\times3}^{output} = Ortho(M_{3\times2}^{6D})$ by a 6D mapping module in **Algorithm 1**.

9: **If** on the training process, **then**:

10: use the multiregression loss function L for the gradient descent regression.

11: **return** the 3D rotation matrix $M_{3\times3}^{output}$

4 Experiment and Analysis

4.1 Datasets and Data Preprocessing

There are three popular public benchmark datasets, 300W-LP [2], AFLW2000 [2] and BIWI [28]. According to most of the previous research work, we choose the 300W-LP dataset as the training dataset and the AFLW2000 and BIWI datasets as the testing datasets. What's more, we retain the images with Euler angles θ of three rotations, including the angles where the yaw, pitch, and roll, are between $[-99°, 99°]$ and convert the annotations of the Euler angles into a 3D rotation matrix.

300W-LP. The 300W-LP dataset was derived from the 300 W dataset [29] which unifies several datasets including AFW [30], HELEN [31], IBUG [29] and LFPW [32], for face alignment with 68 landmarks. It generates 61,225 samples through face profiling with 3D image meshing across large poses and further expands to 122,450 samples with a flipping transformation.

AFLW2000. The AFLW2000 dataset contains the first 2,000 images of the AFLW dataset [33] by providing ground-truth 3D faces and the corresponding 68 landmarks. The faces in the dataset have been processed in large pose variations with various illumination conditions and facial appearances.

BIWI. The BIWI dataset is composed of 24 videos of 20 subjects in a controlled indoor environment. There are a total of approximately 15,000 frames in the dataset. The faces in the dataset are detected by MTCNN [34] to obtain face bounding box results.

For the data preprocessing for training the network, we perform three kinds of data augmentation transformations. First, the image is cropped so that it leaves a margin from the head bounding box $(X_{min}, Y_{min}, X_{max}, Y_{max})$ by a random loose factor $\gamma \in [0.1, 0.5]$. The crop function is equal to Eqs. 15, 16, 17 and 18. Second, the image is randomly flipped horizontally by a probability factor $p \in [0, 1]$. If $p > 0.5$, the image is flipped.

Third, the image is randomly blurred by a probability factor $q \in [0, 1]$. If $q > 0.5$, make the image blurred. Finally, before the image is fed into the network, we resize the image into the uniform size 224×224 and normalize the pixel values into [0, 1].

$$X'_{min} = X_{min} - |\gamma * (X_{max} - X_{min})| \tag{15}$$

$$Y'_{min} = Y_{min} - |\gamma * (Y_{max} - Y_{min})| \tag{16}$$

$$X'_{max} = X_{max} + |\gamma * (X_{max} - X_{min})| \tag{17}$$

$$Y'_{max} = Y_{max} + |\gamma * (Y_{max} - Y_{min})| \tag{18}$$

4.2 Experiment Environment

We use PyTorch to implement our proposed network. For data augmentation in the training process, the raw images are randomly cropped into a loose size with a scale factor k, randomly flipped by a probability factor p and randomly blurred by a probability factor q. The details are shown in Sect. 4.1.

We use an Adam optimizer with a learning rate of 1e-4. We set the batch size to 128 and use 80 epochs to train the network. The experiments were performed on a computer with a GTX3090Ti GPU. For experiments practice, at the beginning of the backpropagation, we only use the geodesic loss function before the 30^{th} epoch. After the 30^{th} epoch, we use the multiregression loss combined with the orthogonal loss and set the factor of orthogonal loss to 0.1 in experiment.

4.3 Results and Analysis

Experiment 1: Experiment 1 is implemented here with the factor settings in Sect. 4.2. Our 6DHPENet is composed of the CNN backbone of RepVGG-b1g2 and the squeeze-and-excitation module and outputs the 3D rotation matrix. 6DHPENet is trained on the 300W-LP datasets. Because the 3D rotation matrix is nonintuitive and most works choose Euler angles as output, we convert the predicted rotation matrix to Euler angles for comparison. As shown in Fig. 4, we present some sample results of head pose estimations using the proposed trained method by visualizing the Euler angles. The red

Fig. 4. Sample results of head pose estimation using the proposed method.

line indicates the pitch angle on the x-axis. The green line indicates the yaw angle on the y-axis, and the blue line indicates the roll angle on the z-axis.

On the evaluation of the experiment results, we take the mean absolute error (MAE) to calculate the error between the predicted Euler angles and the ground-truth Euler angles. The MAE is calculated by Eq. (19).

$$MAE = \sum_{\theta=yaw,pitch,roll} \left(\left| \theta_{prdeict} - \theta_{ground-truth} \right| \right) \tag{19}$$

A summary of the results is given in Table 1 below. It achieves the best result of an average mean absolute error (Avg MAE) of 3.63 on the BIWI and AFLW2000 datasets. Specifically, the MAE of the sum of the three angles is improved by 3.2% in the AFLW2000 datasets.

Table 1. Summary of results: Avg[2] MAE denotes the average of the BIWI and AFLW2000 datasets overall results. The Full Range[1] denotes whether the method allows full range predictions.

Method	Full Range[1]	Params ($\times 10^6$)	BIWI MAE	AFLW2000 MAE	Avg[2] MAE
3DDFA [2]	N	–	19.07	7.393	13.231
Hopenet [8]	N	23.9	4.895	6.155	5.525
SSR-Net-MD [35]	N	**0.2**	4.650	6.010	5.330
FSA-Caps-Fusion [10]	N	1.2	4.000	5.070	4.535
WHENet-V [18]	N	4.4	3.475	4.834	4.155
WHENet [18]	Y	4.4	3.814	5.424	4.619
QuatNet [9]	N	–	4.146	4.503	4.325
6DRepNet [19]	Y	4.3	3.470	3.970	3.720
TriNet [20]	Y	–	4.290	4.669	4.480
6DHPENet(ours)	Y	4.4	**3.420**	**3.840**	**3.630**

As shown in Table 2, we make comparisons with the state-of-the-art methods on the BIWI and AFLW2000 datasets in detail by the MAE on each rotation of the head pose and the angles of yaw, pitch and roll. On the BIWI datasets, the MAE on the pitch is 4.01, and the total MAE of the three angles is 3.42. On the AFLW2000 datasets, the MAE on the pitch is 4.68, and the total MAE of the three angles is 3.84. The results show that our 6DHPENet obtains the best result both on the pitch angle and the total MAE of the three angles. This means that our method is adapted to the general scenarios. Although 6DHPENet does not always perform the best, with MAE values of 3.51, 3.65 and 2.74 for the yaw and roll angles, respectively. However, in contrast to other methods, the difference is small, and equals 0.57, 0.02 and 0.06, respectively. Although the MAE on the yaw angle is higher than that of the QuatNet by a difference value of 0.57 (3.51–2.94) in the comparisons, the MAE on the other angles is lower than that of the QuatNet by

an average value of 0.88. The reason is probably the lack of enough full-view samples and the influence of visual instability for the various head poses, especially the yaw and roll angles, which have not yet been eliminated.

Table 2. Comparisons with state-of-the-art methods on the BIWI and AFLW2000 datasets.

	BIWI				AFLW2000			
Method	Yaw	Pitch	Roll	MAE	Yaw	Pitch	Roll	MAE
3DDFA [2]	36.20	12.30	8.78	19.10	5.40	8.53	8.25	7.39
Hopenet ($\alpha = 1$) [8]	4.81	6.61	3.27	4.90	6.92	6.64	5.67	6.41
Hopenet ($\alpha = 2$) [8]	5.12	6.98	3.39	5.12	6.47	6.56	5.44	6.16
FSA-Net [10]	4.27	4.96	2.76	4.00	4.50	6.08	4.64	5.07
HPE [36]	3.12	5.18	4.57	4.29	4.80	6.18	4.87	5.28
WHENet-V [18]	3.60	4.10	2.73	3.48	4.44	5.75	4.31	4.83
WHENet [18]	3.99	4.39	3.06	3.81	5.11	6.24	4.92	5.42
QuatNet [9]	**2.94**	5.49	4.01	4.15	3.97	5.62	3.92	4.50
6DRepNet [19]	3.24	4.48	**2.68**	3.47	**3.63**	4.91	3.37	3.97
TriNet [20]	4.11	4.76	3.05	3.97	4.04	5.77	4.20	4.67
6DHPENet (ours)	3.51	**4.01**	2.74	**3.42**	3.65	**4.68**	**3.18**	**3.84**

Experiment 2: We conduct an error analysis of two landmark-free methods (WHENet and 6DHPENet) on the AFLW2000 datasets. The error analysis visualizes the MAE on the total angle of yaw, pitch and roll, or the difference in each angle by predicting the head pose when a single RGB image is inputted that is generated by WHENet and 6DHPENet. Both methods are trained on 300W-LP. We convert the predicted rotation matrix to Euler angles for comparison. The results are shown in Fig. 5 and Fig. 6.

As shown in Fig. 5, we take some examples on the AFLW2000 datasets from the frontal to the profile views. On the first line, we put some original images and draw the ground-truth annotations of the Euler angles below the image as in $[\theta_{yaw}, \theta_{pitch}, \theta_{roll}]$. On the second line, we show the result by using WHENet and print the total MAE of the three angles below the image. On the third line, we show the result by using the proposed method 6DHPENet and print the total MAE of the three angles below the image. The frontal views of a human face are shown in the first and fourth columns. The predicted results by 6DHPENet make for a total MAE of 2.207° and 2.731°, whereas the predicted results by WHENet make for a total MAE of 4.518° and 8.837°. The profile views of a human face are shown in the second and third columns. The predicted results by 6DHPENet make for a total MAE of 5.981° and 5.631°, whereas the predicted results by WHENet make for a total MAE of 11.945° and 8.494°. Based on observations of the visualization results, it can be concluded that the smaller the total MAE is, the better accuracy the head pose estimation has. The proposed method has good generalization ability from the frontal to the profile views of the head pose. Contrary to our method

based on the 6D rotation representation, the traditional landmark-free methods of directly predicting the Euler angles from images have a large accuracy error.

Fig. 5. Comparison of head pose estimation results on AFLW2000 images.

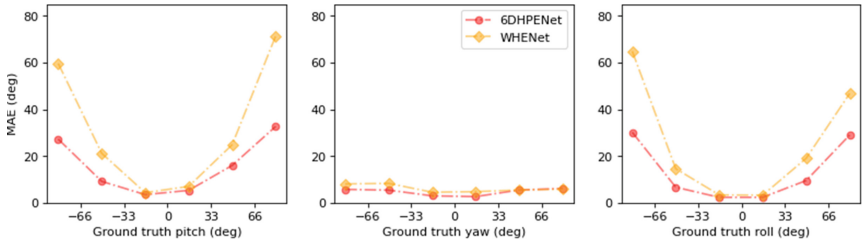

Fig. 6. MAE on AFLW2000 using the landmark-free methods. All were trained on 300W-LP.

As shown in Fig. 6, we draw the lines of the MAE values. The Euler angles' range $\theta \in [-99°, 99°]$ is equally divided into intervals in a span of 33°. The first picture shows the trend in the MAE of a pitch angle between six intervals ($[-99°, -66°)$, $[-66°, -33°)$, $[-33°, 0°)$, $[0°, 33°)$, $[33°, 66°)$, $[66°, 99°]$). The second picture shows the trend in the MAE of the yaw angle between the same six intervals. The third picture shows the trend in the MAE of the roll angle between the same six intervals. The red curve represents the prediction MAE of 6DHPENet, and the yellow curve represents the prediction MAE of WHENet. On the rotation of pitch, yaw and roll, all the MAE curves of 6DHPENet are significantly lower than those of the WHENet. Therefore, it proves that the continuity of 6D rotation representation makes the network learn better and achieve better accuracy.

Experiment 3: We evaluated the effect of resolutions by the trained 6DHPENet on 300W-LP. An image is picked up randomly from the AFLW2000 dataset. The image is

downsampled by a Gaussian kernel. The indicated downsampling factor includes 4X, 16X and 64X. Then, the downsampled images are reshaped to the original size by bilinear interpolation before being supplied to the 6DHPENet. As shown in Fig. 7, the first image is the original, and the predicted value is printed below the corresponding image formed in $[\theta_{yaw}, \theta_{pitch}, \theta_{roll}]$. It can be concluded that the prediction accuracy is not seriously degraded by aggressive downsampling of up to 64X.

The result proves that the 6DHPENet has stability for low-resolution scenes. It can be inferred that 6DHPENet extracts the effective fine-grained facial features in the training process depending on the CNN backbone and SE module.

Fig. 7. Downsampling factor vs. the angles of yaw, pitch & roll. The ground-truth values are $[19.495°, 2.322°, -4.536°]$.

4.4 Ablation Study

To check the impact of different modules for the network, we conduct ablation studies over the different aggregation modules. The results are shown in Table 3.

Table 3. Ablation study over different aggregation modules. (with/without the squeeze-and-excitation module, geodesic loss function or orthogonal loss function).

ID		Modules		AFLW2000			
	SE module	Geodesic loss	Orthogonal loss	Yaw	Pitch	Roll	MAE
1	✓	✓		3.773	4.699	**3.154**	3.874
2		✓	✓	3.666	4.782	3.359	3.936
3		✓		3.692	4.812	3.460	3.988
4	✓	✓	✓	**3.650**	**4.689**	3.180	**3.840**

Test ID1 is composed of an SE module and geodesic loss. Its result is better than the other test without the SE module and it improves by 2.8% over the worst MAE of 3.988. This shows that the SE module generates fine-grained features and is beneficial

to the regression of the network. Test ID2 and test ID3 are without the SE module, one is with orthogonal loss while the other is without orthogonal loss. Observed from the results, the test with orthogonal loss scores a lower MAE of 3.936 as compared to the other one of 3.988. This shows that it is necessary to constrain the calculation error caused by Gram–Schmidt-like process. Test ID4 is composed of all the modules and it obtains the best result of a total MAE of 3.840. It can be inferred that our approach, combined with the SE module and multiregression loss function, is effective and suitable for fine-grained head pose estimation.

Table 4. Comparisons with different CNN backbones.

	CNN backbone	BIWI				AFLW2000			
		Yaw	Pitch	Roll	MAE	Yaw	Pitch	Roll	MAE
1	Efficientnet-b0	3.837	4.230	2.762	3.609	4.020	4.963	3.666	4.216
2	Repvgg-b1g2	**3.510**	**4.012**	**2.738**	**3.420**	**3.650**	**4.689**	**3.180**	**3.840**

As shown in Table 4, we use the Efficientnet-b0 and the Repvgg-b1g2 network separately as the CNN backbone for the 6DHPENet. The experiment results show that Repvgg-b1g2 is better than Efficientnet-b0. There is a phenomenon that the SE module does not obtain a better score even if it is embedded in Efficientnet-b0. We assume that the squeeze-and-excitation module added after the last CNN layer may be more conducive to the spatial integration of the local and global information in the CNN network, as well as the channel relationship for head pose estimation.

5 Conclusions

In this paper, we present a 6D head pose estimation network (6DHPENet) to solve the problem of predicting a head pose without 3D facial landmarks by end-to-end deep learning. The 6DHPENet adopts a 6D rotation representation for 3D rotations as the training objective without 3D facial landmarks. The squeeze-and-excitation module is introduced to construct the local spatial and global channel-wise information by explicitly modeling the interdependencies between the feature channels. A novel multiregression loss function is designed to improve the accuracy for the full-range view of a HPE. The experiment results show that the 6D rotation representation for 3D rotations outperforms the other methods.

In the future, to improve the efficiency of the network, it is promising to extend the larger full-view datasets to train the network. In addition, due to the applications that usually need to be deployed on mobile devices, we expect to study more lightweight networks and improve the corresponding speeds.

References

1. Vincent, L., Pascal, F.: Monocular model-based 3D tracking of rigid objects: a survey, now (2005)

2. Xiangyu, Z., Zhen, L., Xiaoming, L., Hailin, S., Stan, Z.L.: Face alignment across large poses: a 3D solution. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 146–155. IEEE (2016)
3. Murphy-Chutorian, E., Trivedi, M.M.: Head pose estimation and augmented reality tracking: an integrated system and evaluation for monitoring driver awareness. IEEE Trans. Intell. Transp. Syst. **11**(2), 300–311 (2010)
4. Rehder, E., Kloeden H., Stiller, C.: Head detection and orientation estimation for pedestrian safety. In: Proceedings IEEE International Conference Intelligent Transportation Systems 2014, pp. 2292–2297 (2014)
5. Pirsiavash, H., Vondrick, C., Torralba, A.: Assessing the quality of actions. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8694, pp. 556–571. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10599-4_36
6. Adrian, B., Georgios, T.: How far are we from solving the 2D & 3D face alignment problem? (and a dataset of 230,000 3D facial landmarks). In: Proceedings of International Conference on Computer Vision (ICCV) (2017)
7. Blanz, V., Vetter, T., Rockwood, A.: A morphable model for the synthesis of 3D faces. In: SIGGRAPH 1999: Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, pp. 187–194(1999)
8. Ruiz, N., Chong, E., Rehg, J.M.: Fine-grained head pose estimation without keypoints. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 2155–215509 (2018)
9. Hsu H., Wu, S T., Sheng, W., Wing, H.W., Lee C.: QuatNet: quaternion-based head pose estimation with multiregression loss. IEEE Trans. Multimedia **21**(4), 1035–1046 (2019)
10. Tsun-Yi, Y., Yi-Ting, C., Yen-Yu, L., Yung-Yu, C.: FSA-Net: learning fine-grained structure aggregation for head pose estimation from a single image. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1087–1096 (2019)
11. Slabaugh, G.G.: Computing Euler angles from a rotation matrix (1999)
12. Ashutosh, S., Justin, D., Andrew, Y.N.: Learning 3-D object orientation from images. In: 2009 IEEE International Conference on Robotics and Automation, pp. 794–800. IEEE (2009)
13. Zhou, Y., Barnes, C., Jingwan, L., Yang, J., Hao, L.: On the continuity of rotation representations in neural networks. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5738–5746 (2019)
14. Yi, S., Xiaogang W., Xiaoou T.: Deep convolutional network cascade for facial point detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3476–3483 (2013)
15. Mingxing, T., Quoc, V.L.: EfficientNet: rethinking model scaling for convolutional neural networks. arXiv preprint arXiv:1905.11946 (2019)
16. Xiaohan, D., Zhang, X., Ningning, M., Jungong, H., Guiguang, D., Jian, S.: RepVGG: making VGG-style ConvNets great again. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 13728–13737 (2021)
17. Jie, H., Li, S., Gang, S., Albanie, S.: Squeeze-and-Excitation Networks. IEEE Trans. Pattern Anal. Mach. Intell. 99 (2017)
18. Zhou, Y., Gregson , J.: WHENet: real-time fine-grained estimation for wide range head pose. In: 2020 British Machine Vision Conference (BMVC) (2020)
19. Hempel, T., Abdelrahman, A. A., Al-Hamadi A.: 6D rotation representation for unconstrained head pose estimation. arXiv e-prints (2022)
20. Zhi, C., Zong, C., Dong, L., Ying, C.: A vector-based representation to enhance head pose estimation. In: 2021 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1187–1196 (2021)
21. Wu, C.Y., Xu, Q., Neumann, U.: Synergy between 3DMM and 3D Landmarks for Accurate 3D Facial Geometry (2021)

22. Soheil, S., Arya, S., Josep, M.P., Federico, T.: On closed-form formulas for the 3-d nearest rotation matrix problem. IEEE Trans. Robotics **36**(4), 1333–1339 (2020)
23. Kendall, A., Cipolla, R., et al.: Geometric loss functions for camera pose regression with deep learning. In: Proceedings CVPR, vol. 3, p. 8 (2017)
24. Li, F.J., Xu, Z.B.: The essential order of approximation for neural networks. Sci. China Ser. F Inf. Sci. **194**(1), 120–127 (2004)
25. Stiefel manifold. https://en.wikipedia.org/wiki/Stiefel_manifold
26. Bloom, D.M.: Linear algebra and geometry. CUP Archive (1979)
27. Rodrigues, O.: Journal de Math´ematiques 5, 380 (1840)
28. Gabriele, F., Matthias, D., Juergen, G., Andrea, F., Luc, V.G.: Random forests for real time 3D face analysis. Int. J. Comput. Vision **101**(3), 437–458 (2013)
29. Christos, S., Georgios, T., Stefanos, Z., Maja, P.: 300 faces in-the-wild challenge: the first facial landmark localization challenge. In: Proceedings of the IEEE International Conference on Computer Vision Workshops, pp. 397–403 (2013)
30. Xiangxin, Z., Deva, R.: Face detection, pose estimation, and landmark localization in the wild. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 2879–2886. IEEE (2012)
31. Erjin, Z., Haoqiang, F., Zhimin, C., Yuning, J., Qi, Y.: Extensive facial landmark localization with coarse-to-fine convolutional network cascade. In: Proceedings of the IEEE International Conference on Computer Vision Workshops, pp. 386–391 (2013)
32. Peter, N.B., David, W.J., David, J.K., Neeraj, K.: Localizing parts of faces using a consensus of exemplars. IEEE Trans. Pattern Anal. Mach. Intell. **35**(12), 2930–2940 (2013)
33. Peter, M., Roth, M.K., Paul, W., Horst, B.: Annotated facial landmarks in the wild: a large-scale, real-world database for facial landmark localization. In: Proceedings First IEEE International Workshop on Benchmarking Facial Image Analysis Technologies (2011)
34. Kaipeng, Z., Zhanpeng, Z., Zhifeng, L., Yu, Q.: Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Process. Lett. **23**(10), 1499–1503 (2016)
35. Tsun-Yi, Y., Yi-Hsuan, H., Yen-Yu, L., Pi-Cheng, H., Yung-Yu, C.: SSR-Net: a compact soft stagewise regression network for age estimation. In: IJCAI, vol. 5, p. 7 (2018)
36. Bin, H., Renwen, C., Wang, X., Qinbang, Z.: Improving head pose estimation using two-stage ensembles with top-k regression. Image Vis. Comput. **93**, 103827 (2020)
37. Jamie, S., Shaogang, G., Eng-Jon, O.: Understanding pose discrimination in similarity space. In: BMVC, pp. 1–10 (1999)
38. Jeffrey, N., Shaogang, G.: Composite support vector machines for detection of faces across views and pose estimation. Image Vis. Comput. **20**(5–6), 359–368 (2002). https://doi.org/10.1016/S0262-8856(02)00008-2
39. Cai, Q., Gallup, D., Zhang, C., Zhang, Z.: 3D deformable face tracking with a commodity depth camera. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6313, pp. 229–242. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15558-1_17
40. Ruigang, Y., Zhengyou, Z.: Model-based head pose tracking with stereo vision. In Proceedings of Fifth IEEE International Conference on Automatic Face Gesture Recognition, pp. 255–260. IEEE (2002)
41. Srinivasan, S., Boyer, K.L.: Head pose estimation using view based eigenspaces. In: Object Recognition Supported by User Interaction for Service Robots, vol. 4, pp. 302–305. IEEE (2002)