QuatNet: Quaternion-based Head Pose Estimation with Multi-regression Loss

Heng-Wei Hsu, Tung-Yu Wu, Sheng Wan, Wing Hung Wong, and Chen-Yi Lee, *Member*, *IEEE*

***Abstract*—Head pose estimation has attracted immense research interest recently, as its inherent information significantly improves the performance of face-related applications such as face alignment and face recognition. In this paper, we conduct an in-depth study of head pose estimation and present a multi-regression loss function, a $L2$ regression loss combined with an ordinal regression loss, to train a convolutional neural network (CNN) that is dedicated to estimating head poses from RGB images without depth information. The ordinal regression loss is utilized to address the non-stationary property observed as the facial features change with respect to different head pose angles and learn robust features. The $L2$ regression loss leverages these features to provide precise angle predictions for input images. To avoid the ambiguity problem in the commonly used Euler angle representation, we further formulate the head pose estimation problem in quaternions. Our quaternion-based multi-regression loss method achieves state-of-the-art performance on the AFLW2000, AFLW test set and AFW datasets and is closing the gap with methods that utilize depth information on BIWI dataset.**

***Index Terms*—Convolutional neural network, head pose estimation, ordinal regression, quaternion.**

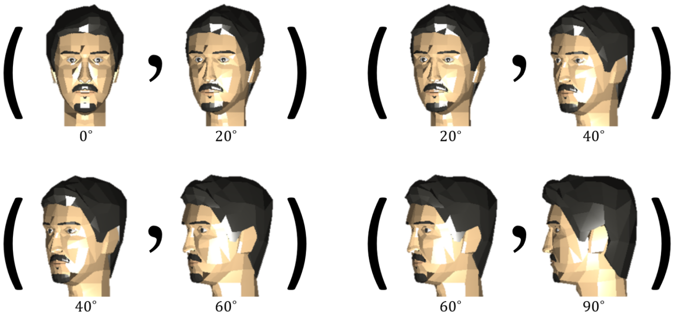


Fig. 1: The non-stationary property observed in yaw angle of the Euler angle representation. The head pose variations within each interval is different. Variations within range 0° to 20° and range 60° to 90° are much smaller than variations within range 20° to 40° and range 40° to 60°.

I. Introduction

Estimating the angles of the head poses is an important topic recently, since the angle information can further improve the performance of face-related tasks. Most recent works that provide accurate angle predictions require not only the RGB image as input but also additional depth information, which is typically obtained by a depth camera [1], [2], [3], [4], [5]. However, depth cameras require active sensing which can be compromised in an environment with external infrared sources (e.g. outdoors), hence limiting the applicability of this method. Furthermore, for applications that require immediate reactions and massive data processing within a short period of time, the use of depth camera and the additional processing time associated with it can be prohibitively expensive. Therefore, head pose estimation that simply uses RGB images serves as a compromise between speed and accuracy and has the potential for more diverse applications.

One common way to estimate head pose is to use the Perspective-n-Point (PnP) [6] method to solve the correspondence between the 2D landmarks of faces and a 3D generic head model. A rotation matrix is thus obtained and can be transformed to angle representations. The accuracy of this two-step process highly relies on the accuracy of the face alignment model. Although some CNN-based face alignment models [7], [8], [9] have already achieved impressive performance for faces in frontal view, their accuracy drop severely for faces in profile view, where appropriate facial landmarks are not visible. Furthermore, head pose estimation has often been regarded as an auxiliary task or by-product in several research areas, including face alignment [10], [11], [12], face recognition [13], [14], [15] and gaze estimation [16], [17]. The performance of these face-related tasks can be significantly improved when the head pose information is considered, but very few methods are dedicated to improving the head pose estimation itself. Motivated by these issues, an end-to-end CNN-based head pose estimation model is considered. We provide a thorough study of the head pose estimation problem in this paper and propose a multi-regression loss function, a $L2$ regression loss combined with an ordinal regression loss, that leads to improved performance of predicting angles from images and achieves state of the art on several public benchmark datasets.

Ordinal regression learns to predict the ordering of the labels instead of the label values themselves, which is important and effective when the relative order of the labels plays a more important role than their exact values. It is recently used in age estimation [18], [19] to address the non-stationary property of facial aging process. A similar non-stationary property can also be observed in head pose estimation as illustrated in Fig. 1. Take the positive yaw angle in Euler angle representation as

Work supported by MOST/MOE of Taiwan under Grant 105-2218-E-009-001/107W210 and NSF of USA under Grant DMS-1407557.

H.-W. Hsu, S. Wan, and C.-Y. Lee are with the Institute of Electronics, National Chiao Tung University, Hsinchu, Taiwan (E-mail: {hengwzx,vjod,cylee}@si2lab.org)

T.-Y. Wu and W. H. Wong are with the Institute for Computational and Mathematical Engineering, Stanford University, Stanford, CA, USA (E-mail: {tungyuwu,whwong}@stanford.edu)

example, we can observe that the facial features do not change smoothly with respect to the angle size. The patterns of the faces that are within 20° (frontal face) change slowly and look alike, but change significantly when the angle is within range 40° to 60°. When the angle is above 60° (profile face), the difference is hardly noticeable since half of the face is not visible. Similar observations can also be made on pitch and roll angles in Euler angle representation, but the patterns vary in different manners. Furthermore, it is always easier to predict whether the angle of a head pose is larger than another than its precise value. Hence, we propose to address the non-stationary property in head pose estimation with ordinal regression loss to learn robust features that can rank different intervals of the angles. We further leverage these features to learn a more precise angle prediction through $L2$ regression loss.

The angle of a head pose can be represented in different forms, each with their pros and cons. Euler angle (yaw, pitch, roll) is the most commonly used representation to describe the rotation of an object since it is intuitive and requires only three elements. It has several drawbacks: i) the rotation order of the axes matters which needs to be specified in advance; ii) the ambiguity problem termed gimbal lock [20] which causes the representation system to lose one degree of freedom when two axis of rotations become parallel. Another representation uses a rotation matrix, which is convenient for computation and transformation between different angle representations, but requires at least 9 elements to represent which is not intuitive and computationally expensive. The axis-angle representation $(\boldsymbol{a}, \theta)$ denotes the rotation with an unit vector $\boldsymbol{a} = (a_x, a_y, a_z)$ that represents the direction of an axis in three-dimensional Euclidean space and an angle θ that indicates the magnitude of the rotation about the axis. The axis-angle representation is compact and straightforward but its downside is that there is no simple interpolation between two axis-angle representations. Furthermore, its composition of rotations is computationally inefficient. To overcome these issues, the quaternion representation which is represented by four elements, (q_x, q_y, q_z, q_w), is commonly used as it can avoid the gimbal lock problem and be interpolated for rendering smooth rotations. Hence, we propose to learn our multi-regression loss CNN model using the quaternion representation, that is, our model predicts quaternion angles of the head pose in an input image.

The main contributions of this paper are summarized as follows.

1) We propose a CNN model focusing on head pose estimation and the model is trained with only RGB images without the need of depth information.
2) To address the non-stationary property in head pose estimation, we design a novel multi-regression loss function that combines $L2$ regression loss with ordinal regression loss.
3) We provide an in-depth study of quaternions and propose a CNN model that directly predicts quaternion for each input head pose image.
4) We achieve state-of-the-art performances on several public benchmark datasets.

II. Related Work

In this section, we will introduce related works in the following three areas: head pose estimation, ordinal regression and quaternion representation.

A. Head Pose Estimation

Standalone head pose estimation: Patacchiola *et al.* [21] investigated the use of dropout and adaptive gradient methods in CNN models that directly build a regression model for the yaw, pitch and roll values from the input head pose images. Ahn *et al.* [22] proposed an efficient CNN model that estimates the head pose through monocular camera and its stability in video applications is enhanced with particle filter based post-processing method. Ruiz *et al.* [23] trained a deep convolutional neural network to predict the head pose directly from image intensities. They applied the cross-entropy loss to binned angles and refined the predictions with regression loss. Chang *et al.* [13] proposed FasePoseNet that uses AlexNet [24] as backbone architecture to directly estimate a regression model on the 6 degrees of freedom for the rotation and translation matrices of the faces. The output of FasePoseNet is then converted to a projection matrix to perform 2D or 3D face alignment. Their results demonstrate that the face recognition accuracy can be improved without the need of sophisticated landmark detectors. Furthermore, several works [25], [26], [14] demonstrate that incorporating the head pose information to face recognition task greatly improves the performance.

Head pose estimation as an auxiliary task: Ranjan *et al.* [27] proposed HyperFace that simultaneously performs face detection, landmark localization, pose estimation, and gender recognition with a single CNN model. They further improved this work with an all-in-one CNN model that additionally includes tasks such as smile detection, age estimation, and face recognition [28]. Head pose estimation is considered as one of the face related tasks in the multi-task learning algorithm to enhance the performance of each individual task. The features are aggregated from multi-level layers of the CNN model to capture both global and local features which lead to a better understanding of faces. Head pose estimation only serves as an auxiliary task in these models which are not specifically designed to optimize its performance. Kumar *et al.* [11] presented a novel Heatmap-CNN that iteratively refines the location of the facial landmarks. The model is jointly trained with the head pose estimation task such that a generic pose information is encoded in the shared shallower layers to improve the performance of the facial landmarks. As a by-product, the head pose performance on the AFLW [29] and AFW [30] datasets demonstrates that the model captures the orientation of the faces. Yang *et al.* [10] proposed a five layer CNN model to directly predict the yaw, pitch and roll angles of the input image. The head pose information provides a better initialization for the cascade face alignment model which is more capable of adapting to head poses that have large variations.

B. Ordinal Regression

Ordinal regression has been applied to many research areas including medical applications [31], [32], age estimation [33],

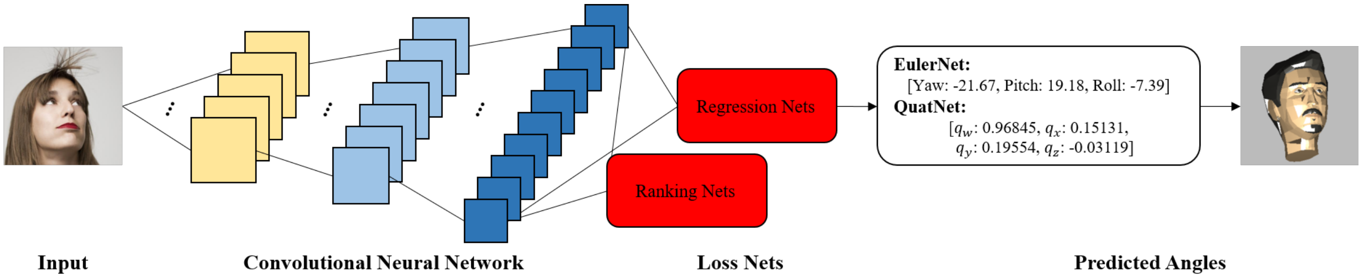

Regression Nets
Ranking Nets
EulerNet:
[Yaw: -21.67, Pitch: 19.18, Roll: -7.39]
QuatNet:
[q_w: 0.96845, q_x: 0.15131,
q_y: 0.19554, q_z: -0.03119]
Input
Convolutional Neural Network
Loss Nets
Predicted Angles

Fig. 2: Illustration of our proposed framework. The input image is passed through a CNN model followed by the regression nets and ranking nets. The ranking nets supervise the network to learn robust features by considering the ordinal relation within labels. The final angle prediction is obtained from the regression nets and the representation is Euler angle if the network is EulerNet and quaternion if it is QuatNet. The simulated 3D head model visualizes the angle representations of the input image which shows a similar orientation.

[18], [19], [34], credit rating [35], and more. A thorough study and taxonomy is proposed in [36] which divides ordinal regression into three categories, naive approaches, ordinal binary decomposition approaches and threshold models, and compares the performance of these methods under various datasets. Several works [37], [33], [19] decompose ordinal regression problems into a series of binary classification subproblems and provide theoretical error bounds to address the consistency issue among the binary classifiers. For problems that have non-stationary property such as age estimation, Chen *et al.* [19] demonstrated that their results show promising performance compared to multi-class cross-entropy loss which completely ignores the ordinal information within labels. Ordinal regression takes advantage of the ranking concept thus showing great improvement when the relative ordering of the labels in the target problem is more reliable than their exact values.

C. *Quaternion Representation*

Quaternion representation is widely used in applications such as computer graphics, computer vision, robotics and navigation. Zhu *et al.* [38] reformulated their original work 3DDFA [39] by using quaternions instead of Euler angles to eliminate the ambiguity which improves the performance of face alignment in large poses. Fathian *et al.* [40] proposed to estimate the camera motion through images by solving a quaternion-based formula. The rotation is represented by quaternion instead of a 3×3 orthogonal matrix which avoids solving a very nonlinear problem. Nevertheless, very little research in head pose estimation has formulated the problem in quaternions, since Euler angles are intuitive and achieve good performance in most cases. In this work, we leverage the advantages of quaternion representations to further improve the performance of predicting angles of head poses.

III. Head Pose Estimation

In this section, we introduce our detailed network structure and the proposed multi-regression loss, a $L2$ regression loss combined with an ordinal regression loss. The non-stationary property observed in the angle changing process of head poses is not addressed in previous related methods, which motivates us to consider this property and propose ranking nets that utilize ordinal regression loss to supervise the training of our network. The proposed regression nets use $L2$ regression loss to predict precise angles by leveraging the learned features from the ranking nets. The common mis-ranking problem in ordinal regression is not an issue since the output angles are predicted solely by the regression nets. Furthermore, predicting the angles of the head poses accurately is important but difficult especially in the case where the head poses are in profile views. Euler angle representation is intuitive and commonly used but it suffers from gimbal lock problem [20] which causes ambiguity during training. This limits its performance when predicting angles for head poses in profile views. To address this problem, we train our network with quaternion representation which inherently avoids the gimbal lock problem and leads to robust performance during testing.

We introduce the notation and assumptions followed by the details of our proposed QuatNet that predicts quaternions for head poses. It consists of three building blocks, a basic CNN structure followed by regression nets and ranking nets. Fig. 2 illustrates the overall framework. This framework can be easily adapted to predict Euler angle representations (EulerNet) by modifying some details in the network to handle the differences in the two angle representations.

A. *Notation and Assumptions*

Let $d^i \in \{d^1, d^2, ..., d^N\}$ denote the input image within a mini-batch of size N and $[\ell^i_{q_x}, \ell^i_{q_y}, \ell^i_{q_z}, \ell^i_{q_w}]$ denote its corresponding unit quaternion label which represents a rotation with four elements $\ell_{q_x}, \ell_{q_y}, \ell_{q_z}, \ell_{q_w} \in \mathbb{R}$, such that $\ell^2_{q_x} + \ell^2_{q_y} + \ell^2_{q_z} + \ell^2_{q_w} = 1$.

B. *QuatNet*

QuatNet is initialized from the pretrained GoogLeNet [41] model, and the layers after the last pooling layer are replaced by our proposed regression nets and ranking nets. The detailed network structure is shown in Fig. 3.

Regression Net: To better capture the variation of the quaternion, we attach four regression nets to the shared pooling layer to learn the four elements, (q_x, q_y, q_z, q_w), of the quaternion representation individually. These four regression nets have the same network structure, namely, a fully connected layer

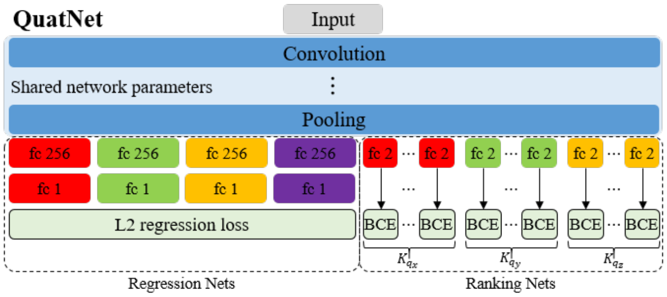


Fig. 3: Detailed structure of QuatNet. The number after ”fc” denotes the dimension of the fully connected layer and BCE represents the binary cross-entropy loss as shown in Eq. 9. The red, green, orange and purple building blocks denote the networks for q_x, q_y, q_z and q_w respectively.

of dimension 256 followed by a fully connected layer of dimension 1. Note that directly regressing these four output neurons on the quaternion label independently leads to inferior performance, since the elements of a quaternion are not totally independent. q_x, q_y, q_z shares the same $\sin\theta/2$ term and q_w has an angle relation with the other elements defined as follows:

$$\begin{cases} q_x = a_x \sin\theta/2 \\ q_y = a_y \sin\theta/2 \\ q_z = a_z \sin\theta/2 \\ q_w = \cos\theta/2 \end{cases} \tag{1}$$

where (a_x, a_y, a_z, θ) is the axis-angle representation. To solve this problem, we apply a nonlinear transform to implicitly encourage the output of these regression nets to learn the independent elements of the axis-angle representation instead of the quaternion representation. The four output neurons of the regression nets are denoted as $f_{a_1}, f_{a_2}, f_{a_3}$ and f_{a_4}, where $(f_{a_1}, f_{a_2}, f_{a_3})$ form the axis and f_{a_4} indicates the angle. These outputs are then transformed to quaternion representation $(f_{q_x}, f_{q_y}, f_{q_z}, f_{q_w})$ which are defined as

$$\begin{cases} f_{q_x} = f_{a_1} \sin f_{a_4} \\ f_{q_y} = f_{a_2} \sin f_{a_4} \\ f_{q_z} = f_{a_3} \sin f_{a_4} \\ f_{q_w} = \cos f_{a_4} \end{cases} \tag{2}$$

The $L2$ regression loss of QuatNet is thus defined as,

$$\begin{aligned} L_{reg}^{Q} = \frac{1}{4N} \sum_{i}^{N} \big[&(f_{q_x}^i - \ell_{q_x}^i)^2 + (f_{q_y}^i - \ell_{q_y}^i)^2 \\ &+ (f_{q_z}^i - \ell_{q_z}^i)^2 + (f_{q_w}^i - \ell_{q_w}^i)^2 \big] \end{aligned} \tag{3}$$

To allow end-to-end training, we derive the back-propagation gradients for the four output neurons $f_{a_1}, f_{a_2}, f_{a_3}$ and f_{a_4} with respect to L_{reg}^Q respectively.

$$\frac{\partial L_{reg}^Q}{\partial f_{a_1}^i} = \frac{1}{2N} (f_{a_1}^i \sin f_{a_4}^i - \ell_{q_x}^i) \sin f_{a_4}^i \tag{4}$$

$$\frac{\partial L_{reg}^Q}{\partial f_{a_2}^i} = \frac{1}{2N} (f_{a_2}^i \sin f_{a_4}^i - \ell_{q_y}^i) \sin f_{a_4}^i \tag{5}$$

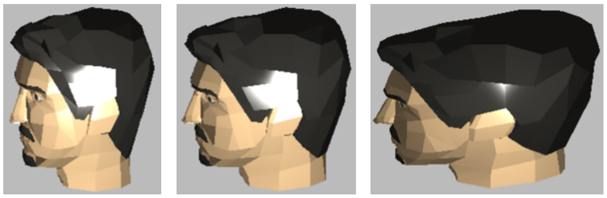


Fig. 4: Simulated 3D head models with quaternions of different $L2$ norm values. The head model would suffer distortion but still capture the orientation of the head pose if the quaternion is not far from unit norm. Empirically, the distortion can be neglected if the $L2$ norm of a quaternion is smaller than 1.1.

$$\frac{\partial L_{reg}^Q}{\partial f_{a_3}^i} = \frac{1}{2N} (f_{a_3}^i \sin f_{a_4}^i - \ell_{q_z}^i) \sin f_{a_4}^i \tag{6}$$

$$\begin{aligned} \frac{\partial L_{reg}^Q}{\partial f_{a_4}^i} = \frac{1}{2N} \big(&(f_{a_1}^i \sin f_{a_4}^i - \ell_{q_x}^i) f_{a_1}^i \cos f_{a_4}^i \\ &+ (f_{a_2}^i \sin f_{a_4}^i - \ell_{q_y}^i) f_{a_2}^i \cos f_{a_4}^i \\ &+ (f_{a_3}^i \sin f_{a_4}^i - \ell_{q_z}^i) f_{a_3}^i \cos f_{a_4}^i \\ &- (\cos f_{a_4}^i - \ell_{q_w}^i) \sin f_{a_4}^i \big) \end{aligned} \tag{7}$$

These gradients are then back propagated to the previous layers through standard CNN learning procedure.

For the regression loss L_{reg}^Q, the four outputs of the regression nets are encouraged to fit the unit norm quaternion labels. Although it seems straightforward to add a $L2$ normalization layer after the outputs before computing L_{reg}^Q, it is observed in our experiments that this makes the training procedure hard to converge. This is due to the elements in the quaternion representation having unbalanced values. In general, the mean value of q_w is much larger than q_x, q_y and q_z (the mean values of q_w, q_x, q_y and q_z are 8.8×10^{-1}, -5.5×10^{-2}, -5.2×10^{-5} and 4.7×10^{-4} respectively in the training dataset), which causes unbalanced gradients during the optimization procedure. Hence we do not explicitly constrain the predicted quaternion to be unit norm, but encourage the network to fit the unit norm quaternion labels. We observed that as long as the training loss L_{reg}^Q decreases and the norm of the predicted quaternion does not diverge too much from the unit norm, we can apply $L2$ normalization during testing, and the testing performance is guaranteed. Fig. 4 illustrates the effect when quaternions are not unit norm but still capture the orientation of the head poses.

Ranking Net: To address the non-stationary property in the angles of the head pose, we propose to formulate head pose estimation as an ordinal regression problem and decompose it into a series of binary classification sub-tasks. Each sub-task is associated with a predefined rank value and the rank values of these sub-tasks follow an ordinal relation. These binary classification sub-tasks learn to rank by predicting whether the angle of the input image is larger than their corresponding rank values.

Let K denote the number of binary classification sub-tasks and r denote the predefined rank value associated with each sub-task. We manually design different binary classification sub-tasks for q_x, q_y, q_z as the variation in each dimension is different. Note that since q_w implies a composite rotation around an axis and its value varies irregularly as the head pose changes, we do not observe ordinal relation in it. Therefore, we only apply ordinal regression loss to q_x, q_y and q_z elements of a quaternion. Specifically, we design K_{q_x} sub-tasks for q_x, with the associated rank value $r^k_{q_x} \in \{r^1_{q_x}, r^2_{q_x}, \ldots, r^{K_{q_x}}_{q_x}\}$. K_{q_y}, $r^k_{q_y}$ and K_{q_z}, $r^k_{q_z}$ are similarly defined. For the same input image d^i, each sub-task has a different binary class label. These binary class labels are one-hot vectors, $[0,1]$ or $[1,0]$, depending on the relation between their labels and the rank values. Take the k^{th} task of the q_x element as an example, its one-hot vector label $\boldsymbol{h}^{i,k}_{q_x}$ is assigned to $[0,1]$ if its quaternion label $\ell^i_{q_x}$ is larger than the k^{th} rank value $r^k_{q_x}$ and $[1,0]$ otherwise.

$$\boldsymbol{h}^{i,k}_{q_x} = \begin{cases} [0,1], & \text{if } \ell^i_{q_x} > r^k_{q_x} \\ [1,0], & \text{otherwise} \end{cases} \tag{8}$$

$\boldsymbol{h}^{i,k}_{q_y}$ and $\boldsymbol{h}^{i,k}_{q_z}$ can be constructed in a similar process.

We apply these binary classification sub-tasks to the shared pooling layer and the cross-entropy losses are applied as loss function to predict the binary labels. Let $\boldsymbol{g}^{i,k}_{q_x}$ denote the binary output vector of the k^{th} q_x sub-task normalized by the softmax function given the input image d^i. The loss function of the ranking net L^Q_{rank} is defined as follows,

$$\begin{aligned} L^Q_{rank} = \frac{-1}{N}\sum_i^N [& \sum_k^{K_{q_x}}\sum_j^m h^{i,k,j}_{q_x}\log(g^{i,k,j}_{q_x}) \\ & + \sum_k^{K_{q_y}}\sum_j^m h^{i,k,j}_{q_y}\log(g^{i,k,j}_{q_y}) \\ & + \sum_k^{K_{q_z}}\sum_j^m h^{i,k,j}_{q_z}\log(g^{i,k,j}_{q_z})] \end{aligned} \tag{9}$$

where $m = 2$ denotes the binary output neurons.

QuatNet is jointly trained with these two losses, hence its overall loss function L^Q is defined as,

$$L^Q = L^Q_{reg} + \lambda^Q L^Q_{rank} \tag{10}$$

where the parameter λ^Q denotes the weighting between these two losses.

The regression nets and ranking nets share the same pooling layer and the angles are solely predicted by the regression nets. The role of the ranking nets is to supervise the network to learn robust features, which further improve the performance of the regression nets. Furthermore, the common mis-ranking problem [18], [19] in ordinal regression would not be an issue since we do not utilize the binary labels of the sub-tasks in the ranking nets during testing.

IV. Experimental Settings

In this section, we evaluate the performance of QuatNet on three public benchmark datasets that provide angle annotations. Although these datasets do not provide quaternion labels, we can transform the provided Euler angle labels to quaternion labels by the following transformation. Let (α, β, γ) represent the value of roll, pitch and yaw angle from Euler angle representation, and the rotation sequence is assumed as yaw to pitch to roll. The corresponding quaternion q is defined as:

$$q = \begin{bmatrix} q_w \\ q_x \\ q_y \\ q_z \end{bmatrix} = \begin{bmatrix} \cos\frac{\alpha}{2}\cos\frac{\beta}{2}\cos\frac{\gamma}{2} + \sin\frac{\alpha}{2}\sin\frac{\beta}{2}\sin\frac{\gamma}{2} \\ \sin\frac{\alpha}{2}\cos\frac{\beta}{2}\cos\frac{\gamma}{2} - \cos\frac{\alpha}{2}\sin\frac{\beta}{2}\sin\frac{\gamma}{2} \\ \cos\frac{\alpha}{2}\sin\frac{\beta}{2}\cos\frac{\gamma}{2} + \sin\frac{\alpha}{2}\cos\frac{\beta}{2}\sin\frac{\gamma}{2} \\ \cos\frac{\alpha}{2}\cos\frac{\beta}{2}\sin\frac{\gamma}{2} - \sin\frac{\alpha}{2}\sin\frac{\beta}{2}\cos\frac{\gamma}{2} \end{bmatrix} \tag{11}$$

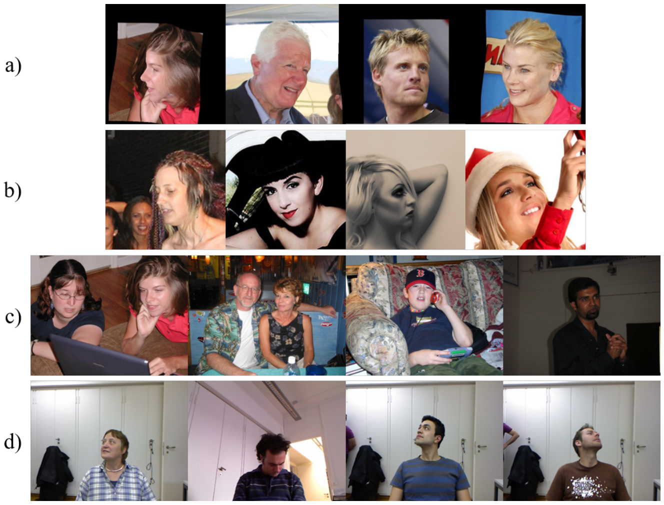


Fig. 5: Sample images of different datasets. a) 300W-LP. b) AFLW2000. c) AFW. d) BIWI.

A. Datasets

300W-LP: 300W-LP [39] is a synthetically generated dataset expanded from 300W [42], a commonly used in-the-wild face alignment dataset, which contains around $4,000$ near frontal face images and is a concatenation of several datasets, including AFW [30], LFPW [43], HELEN [44] and IBUG [42]. 61,225 images with large poses are synthetically generated from 300W through face profiling [39] which is done by predicting the depth of face images and generating the profile views with 3D rotation. 300W-LP only provides Euler angle labels, therefore the quaternions labels are obtained by transforming the Euler angles to quaternions. We utilize this dataset for the training of QuatNet.

AFLW2000: AFLW2000 [39] contains the first 2,000 images from AFLW [29] dataset, and its labels are re-annotated for 3D face alignment. It contains head poses with large variations, different illumination and occlusion conditions. Only Euler angle labels are provided for the images, hence we transform the predictions of QuatNet to Euler angles to evaluate the performance in Euler angle space and also transform the Euler angle labels to quaternion labels to evaluate the performance in quaternion space.

AFW: AFW [30] contains 205 images with 468 faces. These faces have different appearance (occlusion, expression and make-ups) and large pose variations with yaw angle up to 90°. We follow the protocols defined in [30] for performance evaluation and use it only for testing.

TABLE I: Comparison results of different methods on AFLW2000 [39] dataset.

Model	Methods				Euler angle error			Quaternion error				Total	
	I	II	III	IV	Yaw	Pitch	Roll	q_w	q_x	q_y	q_z	EMAE	QMAE
Euler_a					5.496	6.492	6.494	0.0183	0.0396	0.0470	0.0373	6.161	0.0356
Euler_b	✓				5.440	6.432	6.495	0.0179	0.0392	0.0464	0.0373	6.122	0.0352
Euler_c	✓	✓			4.915	6.235	4.859	0.0156	0.0395	0.0415	0.0253	5.336	0.0305
Euler_d	✓		✓		4.901	6.150	4.203	0.0154	0.0384	0.0413	0.0207	5.085	0.0289
EulerNet	✓		✓	✓	**4.859**	**6.022**	**4.180**	**0.0152**	**0.0373**	**0.0409**	**0.0206**	**5.020**	**0.0285**
Quat_a					4.508	5.870	4.362	0.0134	0.0345	0.0383	0.0202	4.913	0.0266
Quat_b	✓				4.322	5.745	4.327	0.0126	0.0332	0.0369	0.0199	4.798	0.0257
Quat_c	✓	✓			4.171	5.740	4.097	0.0124	0.0346	0.0355	0.0194	4.669	0.0255
Quat_d	✓		✓		4.007	5.633	4.025	0.0122	0.0342	0.0339	0.0188	4.556	0.0248
QuatNet	✓		✓	✓	**3.973**	**5.615**	**3.920**	**0.0119**	**0.0339**	**0.0336**	**0.0183**	**4.503**	**0.0244**

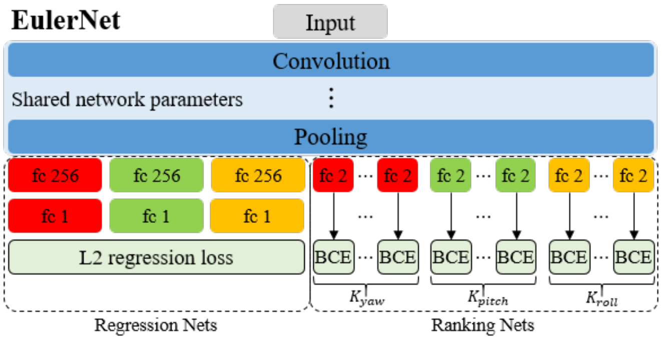


Fig. 6: Detailed structure of EulerNet. The number after "fc" denotes the dimension of the fully connected layer and BCE represents the binary cross-entropy loss. The red, green and orange building blocks denote the networks for yaw, pitch and roll angles respectively.

BIWI: BIWI [45] contains 24 videos with 15,678 frames, which is recorded using Kinect under controlled environment, where the people sit in front of the sensor and freely turn their heads. The range of the head pose angle covers up to 75° for yaw, 60° for pitch and 20° for roll. The labels are in the form of rotation matrix, thus we transform them to Euler angles and quaternions to evaluate the performance of QuatNet. Note that the provided depth information is not used in our experiments.

Sample images of these datasets are shown in Fig. 5.

B. Network Settings

For all our experiments, the proposed methods are implemented on the Caffe framework [46]. QuatNet is initialized by GoogLeNet [41] model, which is pretrained on the ImageNet ILSVRC dataset [47]. All images are first normalized to 256×256 and cropped at 227×227 for the network input. During training, we use random crop and random horizontal mirroring as the basic data augmentation. We further augment the images in the color space and the details and results are discussed in Sec. V. During testing, we crop the center of the image as the network input. We normalize each element of the quaternion labels by its corresponding mean and standard deviation of the training dataset during training to compensate the difference between elements. The same mean and standard deviation parameters are applied at test time. For all experiments, we set the weighting parameter λ^Q to 0.1, and the mini-batch size N is set to 96. We train the network for 30K iterations by using the stochastic gradient descent (SGD) optimizer with the base learning rate set to 5×10^{-3} for all layers and the learning rate is halved every 10K iterations. The whole training process takes around 4.5 hours on a NVIDIA GTX 1080 GPU.

V. Experimental Results

A. Ablation Study

Following similar framework as QuatNet, we also propose EulerNet that predicts Euler angles to demonstrate the effectiveness of training with quaternions. EulerNet is initialized from the pretrained GoogLeNet [41] model, and the layers after the last pooling layer are replaced by the regression nets and ranking nets. We attach three regression nets of the same network structure to the shared pooling layer to learn yaw, pitch and roll angles individually, and the $L2$ regression loss is applied to the regression nets to minimize the distance between the predicted Euler angles and the groundtruth labels. The ranking nets of EulerNet are similar to those of QuatNet, except the associated rank values are redesigned. Let $K_{yaw}, K_{pitch}, K_{roll}$ denote the number of binary classification sub-tasks for yaw, pitch, and roll, and $r^k_{yaw}, r^k_{pitch}, r^k_{roll}$ be the associated rank values. The cross-entropy losses are also applied as loss function to these binary classification sub-tasks. EulerNet is jointly trained with $L2$ regression loss and ordinal regression loss, and the weighting parameter between these two losses, λ^E is set to 0.1. During testing, the Euler angle is obtained solely by the regression nets. The detailed network structure is shown in Fig. 6

We provide an ablation study of EulerNet and QuatNet by conducting controlled experiments to observe how each component affects the performance on AFLW2000 [39] dataset. We evaluate the performance by computing the mean absolute error of the Euler angles (EMAE) and mean absolute error of the quaternions (QMAE). For all the experiments, we use the same basic settings, a GoogLeNet [41] model with the layers after the last pooling layer replaced by a fully connected

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2018.2866770, IEEE Transactions on Multimedia

TABLE II: Details of the ranking nets in EulerNet.

Rank number	Rank value
$K_{yaw} = 6$	$r^i_{yaw} \in \{-60°, -40°, -20°, 20°, 40°, 60°\}$
$K_{pitch} = 6$	$r^i_{pitch} \in \{-60°, -40°, -20°, 20°, 40°, 60°\}$
$K_{roll} = 19$	$r^i_{roll} \in \{-81°, -72°, \ldots, 0°, \ldots, 72°, 81°\}$

layer of dimension 768 followed by a fully connected layer of dimension 3. The $L2$ regression loss is used to regress these 3 neurons on Euler angle labels.We then apply different methods to this basic network to observe the corresponding results. The comparison results of the following methods are shown in Table I.

1) Method I: Use multiple independent fully connected layers? If yes, each element of the angle representation has an independent regression net.
2) Method II: Use cross-entropy nets? If yes, the cross-entropy nets are attached to the shared pooling layer. The details of the cross-entropy nets can be found below.
3) Method III: Use ranking nets? If yes, the ranking nets are attached to the shared pooling layer.
4) Method IV: Use color space data augmentation? If yes, we further augment the data in color space.

For Method III, we set the number of tasks, K_{yaw}, K_{pitch} to 6, and K_{roll} to 19 for EulerNet and their respective rank values are shown in Table II. These rank values partition the label space to several intervals of approximately the same length and the patterns within each interval share similar features. For example, the eyes within range $-20°$ to $20°$ are all visible without occlusion whereas there is only one visible eye for angles larger than $60°$. Similar process can be applied to QuatNet, where we set K_{q_x}, K_{q_y} to 6, and K_{q_z} to 19, and the rank values for QuatNet are obtained by transforming the rank values in Table II to quaternions. The details of rank value selection are discussed in Sec. V-D.

To emphasize the importance of the ordinal relation in head pose estimation, we also compare the performance of EulerNet and QuatNet when the ranking nets are replaced by cross-entropy nets (Method II). The detailed structure of cross-entropy net is shown in Fig. 7. The dimension of the fully connected layer is $K + 1$, since the K binary classification sub-tasks partition the label space to $K+1$ intervals and cross-entropy net classifies the input to one of these intervals.

From the results shown in Table I, we can obtain the following conclusions:

Independent fully connected layers are better. Learning the elements of the angle representation independently slightly gives better performance since the independent fully connected layers learn to disentangle the shared features from the last pooling layer to capture the characteristic of each angle element.

Cross-entropy nets improve learning. Directly mapping the image intensities to the label values through regression is hard for the model to converge smoothly. This is because of the labeling noise within head pose angles. The head pose appears to be almost identical when the angles are within ±5° thus causing ambiguity during learning. Hence, we roughly partition the labels into several intervals for the cross-entropy nets to learn discriminative features which provide better features for predicting precise angles. The results show that the performance is improved by 12.8% for Euler angle and 13.4% for quaternion compared to using $L2$ regression loss only.

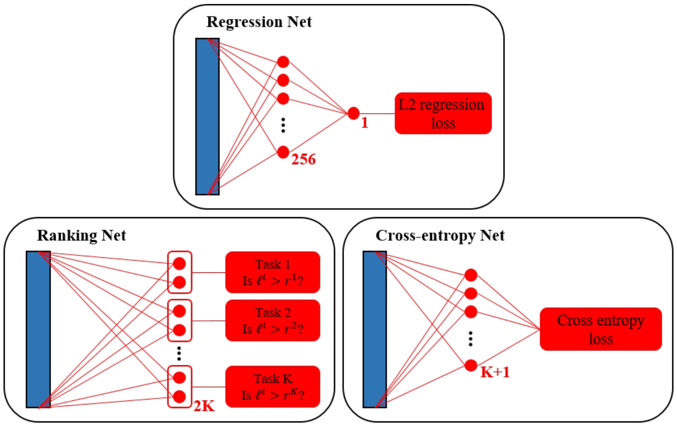


Fig. 7: Detailed structure of regression net, ranking net and cross-entropy net. The blue rectangle indicates the shared pooling layer. The number on the bottom right of each fully connected layer indicates its dimension.

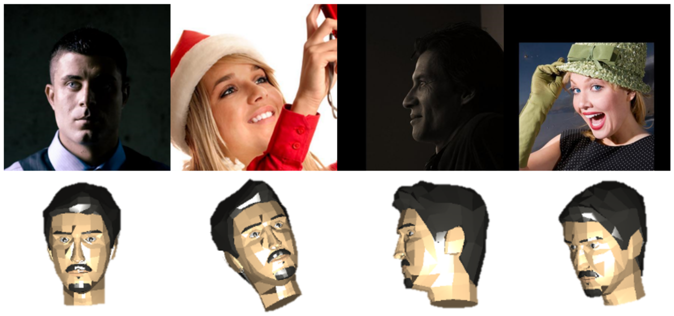
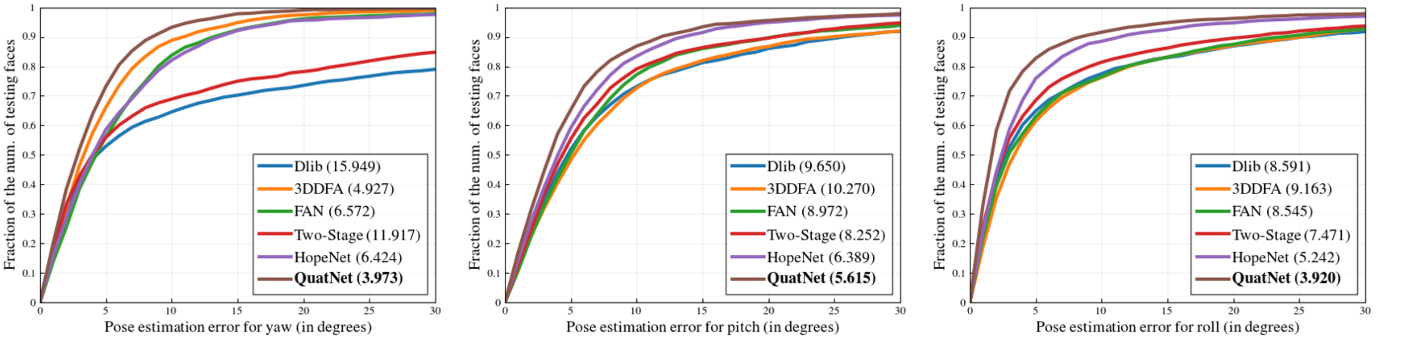


Fig. 8: Qualitative results of QuatNet. The first row is the input images and the second row is the 3D head models simulated using the predictions of QuatNet.

Consider ordinal relation is crucial. Both cross-entropy nets and ranking nets classify the labels into different intervals, but ranking nets additionally consider the ordinal relation between labels which learn more robust features. The performance is improved by 4.7% for Euler angle and 5.2% for quaternion compared to the model trained with cross-entropy nets.

Data augmentation is important. Besides the commonly used basic data augmentation techniques such as random crop and mirroring, we additionally augment the images by adjusting the lightning condition, the saturation and contrast level of the images, which makes the model more robust to unseen images and slightly improves the performance.

Training with quaternions is better. The ambiguity problem in Euler angle representation will confuse the regressor and limit the learning of the model which is a particularly



(a) (b) (c)

Fig. 9: Performance evaluation on AFLW2000 [39] dataset for (a) yaw (b) pitch and (c) roll angles. The EMAE of the angles are shown in the legend respectively.

TABLE III: Mean absolute error of Euler angle using different methods on BIWI [45] dataset. *These methods use additional depth information.

Method	Yaw	Pitch	Roll	EMAE
OpenFace [48]*	7.800	8.000	4.600	6.800
3DMM+FHM [49]*	2.500	1.500	2.200	2.067
Dlib [50]	20.581	15.505	9.324	15.137
3DDFA [39]	8.691	11.180	11.770	10.547
FAN [51]	7.944	13.404	10.233	10.527
KEPLER [11]	8.084	17.277	16.196	13.852
Two-Stage [9]	9.488	11.339	6.002	8.943
HopeNet [23]	5.167	6.975	3.388	5.177
QuatNet	**4.010**	**5.492**	**2.936**	**4.146**

TABLE IV: Mean absolute error of Euler angle using different methods on AFLW test set [29] dataset.

Method	Yaw	Pitch	Roll	EMAE
Dlib [50]	11.171	8.216	7.326	8.904
3DDFA [39]	4.916	7.748	7.045	6.570
FAN [51]	6.301	7.186	6.764	6.750
AdaptGrad [21]	11.040	7.150	4.400	7.530
KEPLER [11]	6.450	5.850	8.750	7.017
Two-Stage [9]	8.264	6.663	5.936	6.954
HyperFace [27]	7.610	6.130	3.920	5.887
HopeNet [23]	6.260	5.890	3.820	5.324
QuatNet	**3.933**	**4.316**	**2.590**	**3.613**

serious problem in images that have large poses. This is the case for the dataset we used for training, 300W-LP [39], which contains many images that are synthesized to large profile views. Quaternion representation overcomes this problem from several aspects as discussed in Sec. I. Therefore, models trained with quaternions consistently lead to better performance.

B. *Qualitative Results*

Fig. 8 shows the performance of QuatNet by generating the simulated 3D head models using the predictions of QuatNet. The simulated 3D head models capture the orientation of the faces in the input images which demonstrates that QuatNet can accurately predict the angles for images under different illumination conditions, various poses and different face expressions.

C. *Quantitative Results*

We follow the conventional evaluation metrics on the benchmark datasets, AFLW2000 [39], BIWI [45], AFLW test set [29] and AFW [30], and compare our QuatNet with several recent works which have shown state-of-the-art performance on these datasets: (i) HopeNet [23], (ii) KEPLER [11], (iii) AdaptGrad [21], (iv) HyperFace [27], (v) OpenFace [48], (vi) 3DMM+FHM [49] (vii) Dlib [50], (viii) FAN [51], (ix) 3DDFA [39], (x) Two-Stage [9], (xi) Multi. HoG [30], (xii) Multi. AAM [30], (xiii) face.com [30], (xiv) FaceDPL [27]. Note that method (vii), (viii), (ix) and (x) are face alignment models that predict landmarks for a given 2D face image. We further assume a generic 3D head model and approximate the intrinsic parameters of the camera by the image shape. The 2D-3D correspondence can then be solved by running an iterative Levenberg-Marquardt optimization method and the rotation matrix is obtained which can be transformed to angle representations for performance evaluation.

For AFLW2000 dataset, we remove 36 images that have Euler angles larger than 90° since there are labeling noises for images that have large poses. We compare the results of our QuatNet with other methods for different pose angles: yaw, pitch and roll. The mean absolute error and the cumulative error distribution curve for each pose angle is reported for performance evaluation. The cumulative error distribution curve reflects the proportion of test images whose errors are below a certain threshold. The comparison results are shown in Fig. 9. The results of predicting yaw, pitch and roll angles from our QuatNet outperform all other methods by a large margin. The 2-step process (detecting landmark, landmark to pose) required by the face alignment methods introduce large error when the landmarks are failed to be detected, whereas our method predicts angles directly from image intensities and consistently leads to better performance. Although HopeNet uses ResNet50 [52] model, a very deep network structure as the CNN backbone architecture, our QuatNet still outperforms their method by using only the GoogLeNet model as backbone

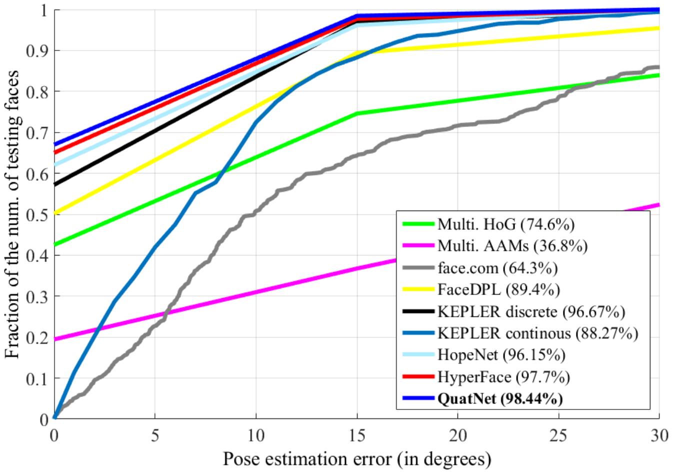


Fig. 10: Mean absolute error of yaw angle using different methods on AFW [30] dataset.

TABLE V: Different settings of the ranking nets.

Rank number	Rank values
$K = 19$	$r^i \in \{-81°, -72°, \ldots, 0°, \ldots, 72°, 81°\}$
$K = 10$	$r^i \in \{-81°, -63°, \ldots, -9°, 9°, \ldots, 63°, 81°\}$
$K = 6$	$r^i \in \{-60°, -40°, -20°, 20°, 40°, 60°\}$

architecture. This demonstrates the effectiveness of our proposed loss function.

For BIWI dataset, we evaluate the performance of the networks by computing the EMAE. The comparison results of our proposed QuatNet and other methods in BIWI dataset are shown in Table III. Since BIWI dataset contains depth information, we also include two methods using depth information, OpenFace [48] and 3DMM+FHM [49]. Our result outperforms OpenFace and is closing the gap between 3DMM+FHM, which is one of the state-of-the-art methods that uses depth information.

For AFLW test set, we follow the protocol defined in [11] by randomly sampled 1,000 images from AFLW [29] dataset and report the EMAE for performance evaluation. The comparison results are shown in Table IV. Our result outperforms all other methods by a large margin on this in-the-wild dataset, which demonstrates the practicability of our method in real world applications.

For AFW dataset, we follow the protocol defined in [30] by rounding-off the predicted yaw angles to the nearest 15°, since the groundtruth labels are provided in multiples of 15°, and compute the absolute error with the groundtruth yaw angles. The cumulative error distribution curves of our proposed QuatNet and other methods are shown in Fig. 10. Our result consistently outperforms other methods for different error margins and can predict the yaw angle within ±15° for 98.44% of faces.

D. Discussions

In this subsection, we discuss the normalization issue in QuatNet, the effect of the parameters used in our experiments, including the rank values and the weighting parameter λ^Q and highlight the performance of QuatNet.

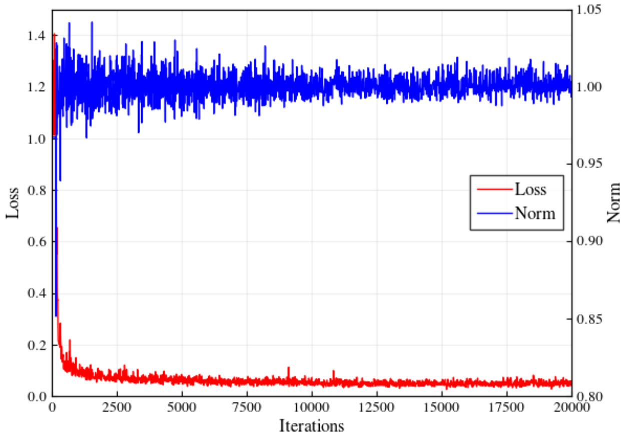


Fig. 11: The average norm value and loss of QuatNet during training.

TABLE VI: Experiment results of different rank values using QuatNet on AFLW2000 [39] dataset.

Model	$[K_{yaw}, K_{pitch}, K_{roll}]$	EMAE	QMAE
model_a	[19,19,19]	4.705	0.0250
model_b	[10,10,19]	4.625	0.0249
model_c	[6,6,19]	**4.556**	**0.0248**
model_d	[6,6,10]	4.569	0.0248

Normalization during testing. Fig. 11 shows the average norm value and training loss of QuatNet in the first 20K iterations, where the norm values are calculated by averaging the $L2$ norm of the quaternion outputs from QuatNet in each training batch. As mentioned in Sec. III, although we do not explicitly add a $L2$ normalization layer, QuatNet learns to fit the unit quaternion labels and its output converges to norm of 1 as the training loss decreases. Therefore, the performance is guaranteed during testing when we normalize the output predictions of QuatNet.

Rank value selection. We conduct several experiments to discuss the effect of using different rank values in the ranking nets. We design the rank values of yaw and pitch angles to be the same and the roll angle can be either same or different, since we observe that the yaw and pitch angles are out-of-plane rotations (facial features suffer from occlusion as the angle becomes larger) and the roll angle is in-plane rotation (facial features are rotated but can still be observed). Table V shows the settings of different rank numbers and the corresponding rank values. Note that here we list the rank values in Euler angle representation form for better understanding and the rank values for QuatNet are obtained by transforming them to quaternions. The results of QuatNet using four different settings of rank values are shown in Table VI. We divide the angles of yaw, pitch and roll uniformly into 20 intervals, without considering the appearance of the faces, which results

TABLE VII: Results of QuatNet in different datasets.

Dataset	Euler angle error			Quaternion error				Total	
	Yaw	Pitch	Roll	q_w	q_x	q_y	q_z	EMAE	QMAE
AFLW2000	3.973	5.615	3.920	0.0119	0.0339	0.0336	0.0183	4.503	0.0244
BIWI	4.010	5.492	2.936	0.0132	0.0431	0.0351	0.0196	4.146	0.0278
AFLW test set	3.933	4.316	2.590	0.0109	0.0329	0.0337	0.0167	3.613	0.0236

TABLE VIII: Experiment results of different λ^Q values using QuatNet on AFLW2000 [39] dataset.

λ^Q	1	0.1	0.01	0.001
EMAE	5.103	**4.556**	4.659	4.710
QMAE	0.0278	**0.0248**	0.0257	0.0255

in a set of 19 rank values (model_a). We compare it with a coarser setting of rank values by reducing the number of rank values to 10 (model_b) for yaw and pitch angles as we do not observe significant difference in the appearance of the faces within angle intervals of length 9. From the results of model_a and model_b, we notice that it is important to divide the angles such that the faces appear similar within each interval and finer intervals do not always lead to better results. Hence by inspecting changes of the face appearance, we partition the angle space into 6 rank values (model_c), such that the patterns within each interval share similar features. For example, faces are considered frontal views within range $-20°$ to $20°$ and profile views for angles larger than $60°$. For faces within range $20°$ to $40°$, both eyes are still visible but one of the eye gets occluded within range $40°$ to $60°$. The performance is further improved with these carefully designed rank values. We also compare the result when the roll angle has a coarser interval (model_d). The result of model_d is comparable to model_c but slightly worse, hence we use the setting of model_c for our proposed QuatNet.

λ^Q selection. We test the model sensitivity to the hyper-parameter λ^Q by varying its value. It is important to combine these two losses since it is difficult for $L2$ regression loss to predict precise angles from the pixel values of the input images directly, therefore it benefits from the features learned by the ordinal regression loss which captures the non-stationary property of head poses. The results are listed in Table VIII. λ^Q controls the weighting of the ordinal regression loss, which learns to roughly partition the angle space to coarse intervals. We observe that the ordinal regression loss can supervise the learning process to learn robust features if the value of λ^Q is near 0.1. If λ^Q is too large, the effect of ordinal regression loss prevents the $L2$ regression loss from learning precise angle predictions which leads to inferior performance. On the other hand, the multi-regression loss is reduced to $L2$ regression loss if λ^Q is too small.

Detail analysis of QuatNet. To highlight the performance of QuatNet for faces in profile views, we separate the AFLW2000 [39] dataset into 3 subsets according to the absolute yaw label of each image. There are $1,310$ images within range $0°$ to $30°$, 378 images within range $30°$ to $60°$ and 276 images within range $60°$ to $90°$. We report the EMAE of different methods for each interval. Two-Stage [9] achieves good performance for faces in frontal views but its performance degrades quickly as the yaw angle increases. QuatNet outperforms all other methods in all three intervals, and the result outperforms the most for the case when faces are in profile views.

TABLE IX: EMAE of 3 yaw angle intervals using different methods on AFLW2000 [39] dataset.

Method	(0° to 30°)	(30° to 60°)	(60° to 90°)
Dlib [50]	4.186	17.202	37.671
3DDFA [39]	4.376	10.092	23.192
FAN [51]	5.038	9.177	20.466
Two-Stage [9]	3.267	13.134	32.067
HopeNet [23]	3.744	7.331	15.014
QuatNet	**2.911**	**5.133**	**11.194**

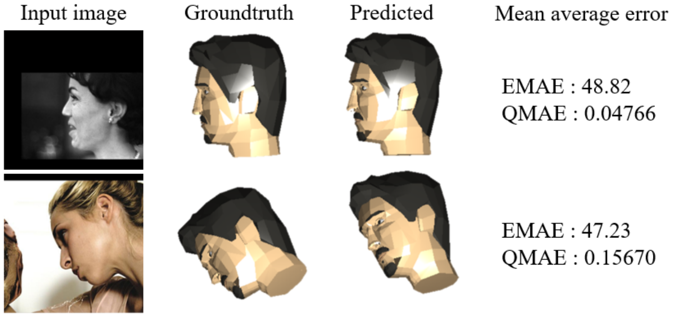


Fig. 12: The problem when using Euler angle representation for performance evaluation.

Advantage of using quaternions. The performance of QuatNet has shown the effectiveness of quaternion representations when training head pose estimation models. We argue that quaternion representation not only improves training but is also more suitable for performance evaluation than Euler angle representation. Fig. 12 illustrates the problem when using Euler angle representation for performance evaluation. The images in the second and third column are the simulated 3D head models of the groundtruth labels and our QuatNet predictions. The MAE of Euler angle and quaternion representations are shown in the last column. If we only consider the EMAE for performance evaluation, it seems that our predictions of these two images have similar error. From the simulated 3D head model, it is clear that our prediction differs more from the groundtruth label on the second image. Hence, we argue that the quaternion representation better captures this property which is more suitable for performance evaluation. For fair comparison with latter works, we additionally report the quaternion performance of AFLW2000, BIWI and AFLW test set datasets using QuatNet in Table VII.

VI. Conclusions

In this paper, we propose a novel multi-regression loss function, a $L2$ regression loss combined with an ordinal regression loss, to train a CNN model that is dedicated to head pose estimation without using depth information. We address the non-stationary property observed in the head pose with the ordinal regression loss which considers the ordinal relation within the angle labels and further improves the performance compared to model trained with cross-entropy loss. We also provide an in-depth study of angle representations and propose to train our model with quaternions to avoid the ambiguity problem in Euler angles. Extensive experiments verify the advantages of each component of our proposed QuatNet and demonstrate state-of-the-art performance on several public benchmark datasets.

References

[1] S. S. Mukherjee and N. M. Robertson, "Deep head pose: Gaze-direction estimation in multimodal video," *IEEE Transactions on Multimedia*, vol. 17, no. 11, pp. 2094–2107, 2015.

[2] G. P. Meyer, S. Gupta, I. Frosio, D. Reddy, and J. Kautz, "Robust model-based 3d head pose estimation," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3649–3657, 2015.

[3] C. Papazov, T. K. Marks, and M. Jones, "Real-time 3d head pose and facial landmark estimation from depth images using triangular surface patch features," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4722–4730, 2015.

[4] S. G. Kong and R. O. Mbouna, "Head pose estimation from a 2d face image using 3d face morphing with depth parameters," *IEEE Transactions on Image Processing*, vol. 24, no. 6, pp. 1801–1808, 2015.

[5] S. G. Kong and R. O. Mbouna, "Head pose estimation from a 2d face image using 3d face morphing with depth parameters," *IEEE Transactions on Image Processing*, vol. 24, pp. 1801–1808, June 2015.

[6] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[7] Y. Sun, X. Wang, and X. Tang, "Deep convolutional network cascade for facial point detection," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pp. 3476–3483, IEEE, 2013.

[8] A. Jourabloo and X. Liu, "Large-pose face alignment via cnn-based dense 3d model fitting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4188–4196, 2016.

[9] J. Lv, X. Shao, J. Xing, C. Cheng, and X. Zhou, "A deep regression architecture with two-stage re-initialization for high performance facial landmark detection," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[10] H. Yang, W. Mou, Y. Zhang, I. Patras, H. Gunes, and P. Robinson, "Face alignment assisted by head pose estimation," in *Proceedings of the British Machine Vision Conference (BMVC)* (M. W. J. Xianghua Xie and G. K. L. Tam, eds.), pp. 130.1–130.13, BMVA Press, September 2015.

[11] A. Kumar, A. Alavi, and R. Chellappa, "Kepler: keypoint and pose estimation of unconstrained faces by learning efficient h-cnn regressors," in *Automatic Face & Gesture Recognition (FG 2017), 2017 12th IEEE International Conference on*, pp. 258–265, IEEE, 2017.

[12] Y. Wu, C. Gou, and Q. Ji, "Simultaneous facial landmark detection, pose and deformation estimation under facial occlusion," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[13] F.-J. Chang, A. T. Tran, T. Hassner, I. Masi, R. Nevatia, and G. Medioni, "Faceposenet: Making a case for landmark-free face alignment," in *2017 IEEE International Conference on Computer Vision Workshop (ICCVW)*, pp. 1599–1608, IEEE, 2017.

[14] I. Masi, S. Rawls, G. Medioni, and P. Natarajan, "Pose-aware face recognition in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4838–4846, 2016.

[15] I. Masi, F. J. Chang, J. Choi, S. Harel, J. Kim, K. Kim, J. Leksut, S. Rawls, Y. Wu, T. Hassner, W. AbdAlmageed, G. Medioni, L. P. Morency, P. Natarajan, and R. Nevatia, "Learning pose-aware models for pose-invariant face recognition in the wild," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, no. 99, pp. 1–1, 2018.

[16] R. Valenti, N. Sebe, and T. Gevers, "Combining head pose and eye location information for gaze estimation," *IEEE Transactions on Image Processing*, vol. 21, pp. 802–815, Feb 2012.

[17] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, "Appearance-based gaze estimation in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4511–4520, 2015.

[18] Z. Niu, M. Zhou, L. Wang, X. Gao, and G. Hua, "Ordinal regression with multiple output cnn for age estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4920–4928, 2016.

[19] S. Chen, C. Zhang, M. Dong, J. Le, and M. Rao, "Using ranking-cnn for age estimation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[20] V. Lepetit, P. Fua, *et al.*, "Monocular model-based 3d tracking of rigid objects: A survey," *Foundations and Trends® in Computer Graphics and Vision*, vol. 1, no. 1, pp. 1–89, 2005.

[21] M. Patacchiola and A. Cangelosi, "Head pose estimation in the wild using convolutional neural networks and adaptive gradient methods," *Pattern Recognition*, 2017.

[22] B. Ahn, J. Park, and I. S. Kweon, "Real-time head orientation from a monocular camera using deep neural network," in *Asian Conference on Computer Vision*, pp. 82–96, Springer, 2014.

[23] N. Ruiz, E. Chong, and J. M. Rehg, "Fine-grained head pose estimation without keypoints," *CoRR*, vol. abs/1710.00925, 2017.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[25] C. Ding and D. Tao, "Robust face recognition via multimodal deep face representation," *IEEE Transactions on Multimedia*, vol. 17, pp. 2049–2058, Nov 2015.

[26] I. Masi, A. T. Trn, T. Hassner, J. T. Leksut, and G. Medioni, "Do we really need to collect millions of faces for effective face recognition?," in *European Conference on Computer Vision*, pp. 579–596, Springer, 2016.

[27] R. Ranjan, V. M. Patel, and R. Chellappa, "Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[28] R. Ranjan, S. Sankaranarayanan, C. D. Castillo, and R. Chellappa, "An all-in-one convolutional neural network for face analysis," in *Automatic Face & Gesture Recognition (FG 2017), 2017 12th IEEE International Conference on*, pp. 17–24, IEEE, 2017.

[29] M. Koestinger, P. Wohlhart, P. M. Roth, and H. Bischof, "Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization," in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pp. 2144–2151, IEEE, 2011.

[30] X. Zhu and D. Ramanan, "Face detection, pose estimation, and landmark localization in the wild," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 2879–2886, IEEE, 2012.

[31] O. M. Doyle, E. Westman, A. F. Marquand, P. Mecocci, B. Vellas, M. Tsolaki, I. Kłoszewska, H. Soininen, S. Lovestone, S. C. Williams, *et al.*, "Predicting progression of alzheimers disease using ordinal regression," *PloS one*, vol. 9, no. 8, p. e105542, 2014.

[32] M. Pérez-Ortiz, M. Cruz-Ramírez, M. D. Ayllón-Terán, N. Heaton, R. Ciria, and C. Hervás-Martínez, "An organ allocation system for liver transplantation based on ordinal regression," *Applied Soft Computing*, vol. 14, pp. 88–98, 2014.

[33] K.-Y. Chang and C.-S. Chen, "A learning framework for age rank estimation based on face images with scattering transform," *IEEE Transactions on Image Processing*, vol. 24, no. 3, pp. 785–798, 2015.

[34] S. Feng, C. Lang, J. Feng, T. Wang, and J. Luo, "Human facial age estimation by cost-sensitive label ranking and trace norm regularization," *IEEE Transactions on Multimedia*, vol. 19, pp. 136–148, Jan 2017.

[35] F. Fernandez-Navarro, P. Campoy-Munoz, C. Hervas-Martinez, X. Yao, *et al.*, "Addressing the eu sovereign ratings using an ordinal regression approach," *IEEE transactions on cybernetics*, vol. 43, no. 6, pp. 2228–2240, 2013.

[36] P. A. Gutiérrez, M. Pérez-Ortiz, J. Sanchez-Monedero, F. Fernández-Navarro, and C. Hervas-Martinez, "Ordinal regression methods: survey and experimental study," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 1, pp. 127–146, 2016.

[37] L. Li and H.-T. Lin, "Ordinal regression by extended binary classification," in *Advances in neural information processing systems*, pp. 865–872, 2007.

[38] X. Zhu, x. Liu, Z. Lei, and S. Z. Li, "Face alignment in full pose range: A 3d total solution," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, no. 99, pp. 1–1, 2017.

[39] X. Zhu, Z. Lei, X. Liu, H. Shi, and S. Z. Li, "Face alignment across large poses: A 3d solution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 146–155, 2016.

[40] K. Fathian, J. P. R. Paredes, E. Doucette, J. W. Curtis, and N. N. Gans, "Quest: A quaternion-based approach for camera motion estimation from minimal feature points," *IEEE Robotics and Automation Letters*, 2018.

[41] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

[42] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "300 faces in-the-wild challenge: The first facial landmark localization challenge," in *Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on*, pp. 397–403, IEEE, 2013.

[43] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, and N. Kumar, "Localizing parts of faces using a consensus of exemplars," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 12, pp. 2930–2940, 2013.

[44] E. Zhou, H. Fan, Z. Cao, Y. Jiang, and Q. Yin, "Extensive facial landmark localization with coarse-to-fine convolutional network cascade," in *Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on*, pp. 386–391, IEEE, 2013.

[45] G. Fanelli, M. Dantone, J. Gall, A. Fossati, and L. Van Gool, "Random forests for real time 3d face analysis," *Int. J. Comput. Vision*, vol. 101, pp. 437–458, February 2013.

[46] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia*, pp. 675–678, ACM, 2014.

[47] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[48] T. Baltrušaitis, P. Robinson, and L.-P. Morency, "Openface: an open source facial behavior analysis toolkit," in *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*, pp. 1–10, IEEE, 2016.

[49] Y. Yu, K. A. F. Mora, and J. Odobez, "Robust and accurate 3d head pose estimation through 3dmm and online head model reconstruction," in *12th IEEE International Conference on Automatic Face & Gesture Recognition*, pp. 711–718, 2017.

[50] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1867–1874, 2014.

[51] A. Bulat and G. Tzimiropoulos, "How far are we from solving the 2d & 3d face alignment problem? (and a dataset of 230,000 3d facial landmarks)," in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[52] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.