Accurate Head Pose Estimation Using Image Rectification and a Lightweight Convolutional Neural Network

Xiao Li[1], Dong Zhang[1, #], Ming Li[2], *Senior Member*, *IEEE*, and Dah-Jye Lee[3], *Senior Member*, *IEEE*

***Abstract*—Head pose estimation is an important step for many human-computer interaction applications such as face detection, facial recognition, and facial expression classification. Accurate head pose estimation benefits these applications that require face images as the input. Most head pose estimation methods suffer from perspective distortion because the users do not always align their face perfectly with the camera. This paper presents a new approach that uses image rectification to reduce the negative effect of perspective distortion and a lightweight convolutional neural network to obtain highly accurate head pose estimations. The proposed method calculates the angle between the optical axis of the camera and the projection vector of the center of the face. The face image is rectified using this estimated angle through perspective transformation. A lightweight network that is only 0.88 MB in size is designed to take the rectified face image as the input to perform head pose estimation. The output of the network, the head pose estimation of the rectified face image, is transformed back to the camera coordinate system as the final head pose estimation. Experiments on public benchmark datasets show that the proposed image rectification method and the newly designed lightweight network improve the accuracy of head pose estimation remarkably. Compared with state-of-the-art methods, our approach achieves both higher accuracy and faster processing speed.**

***Index Terms*—Head pose estimation, Image rectification, Perspective transformation, Convolutional neural networks**

I. Introduction

The goal of head pose estimation in the context of computer vision is to estimate the pose of the head with respect to the camera coordinate system. The estimated head pose is usually expressed by Euler angles (pitch, yaw, roll) [1], as shown in Fig. 1. Head pose estimation plays an important role in many human-computer interaction tasks [2-5]. Accurate head pose estimation improves the performance of landmark detection of human faces [6,7], expression recognition [8], and identity verification [9].

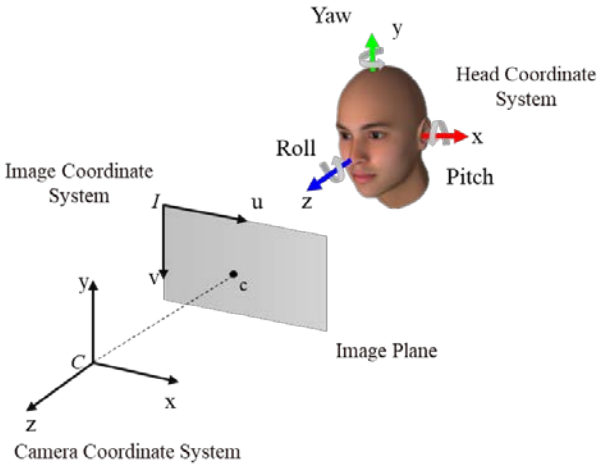


Fig. 1. The three degrees of freedom of a head pose can be described as the egocentric rotation angles (pitch, yaw, and roll) [1].

Early works employed optical motion capturing systems [10], magnetic sensors [11], or laser pointers [12] to estimate the head pose. Optical motion capturing systems and magnetic sensors use multiple detectors to measure the head pose directly and accurately. These two techniques require expensive and complex equipment, which is not always available for practical applications. The technique using a laser pointer is a comparatively simple method for head pose estimation, in which a laser pointer is affixed to the subject's head. However, this technique only provides a rough estimation of the head pose and is not convenient for practical use.

Recently, researchers have developed simpler and more universal techniques to estimate head poses using consumer image-based equipment such as RGB or RGBD cameras. With the prevalence of consumer depth cameras such as Microsoft Kinect and Intel RealSense, depth information has been used to estimate head poses [13-15]. Although the use of depth information significantly improves the head pose estimation accuracy, it suffers from several limitations. Depth information

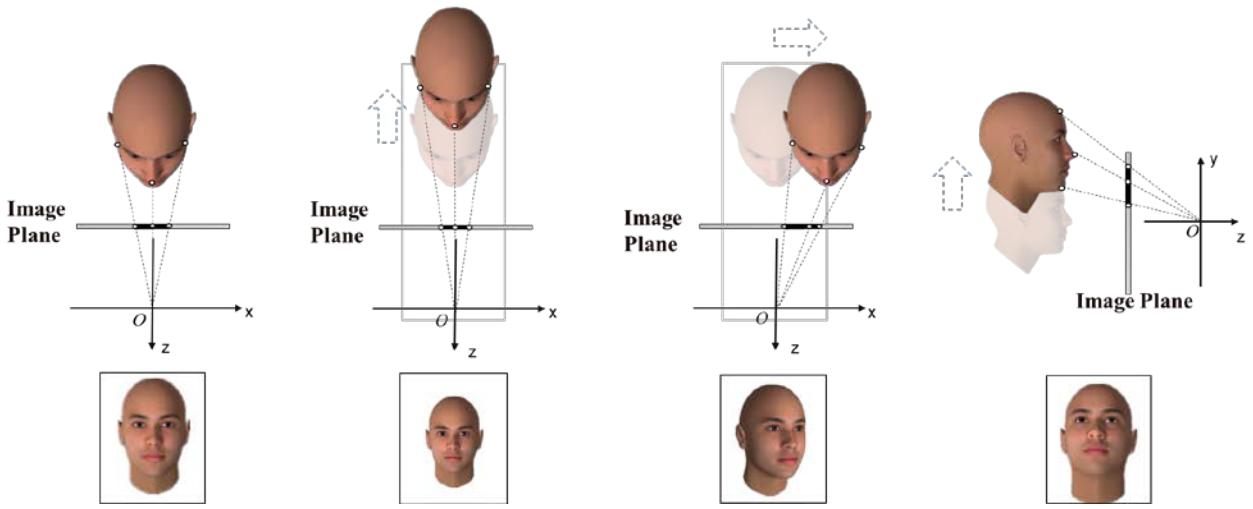
Xiao Li is with the School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou, China, 510006. (E-mail: lixiao37@mail2.sysu.edu.cn).

Dong Zhang is with the School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou, China, 510006. (E-mail: zhangd@mail.sysu.edu.cn).

Ming Li is with Duke Kunshan University, Kunshan, China, 215316. (E-mail: ming.li369@dukekunshan.edu.cn).

Dah-Jye Lee is with the Department of Electrical and Computer Engineering, Brigham Young University, Provo, Utah, USA, 84602 (E-mail: djlee@byu.edu).

Corresponding author: Dong Zhang (Email: zhangd@mail.sysu.edu.cn).



(a) (b) (c) (d)

Fig. 2. Projection of face using the full-perspective projection model: (a) the head is aligned perfectly with the camera coordinate system with zero pitch, yaw, and roll; (b) the head moves away from the camera along the z-axis; (c) the head moves along the x-axis; (d) the head moves along the y-axis. The projected images appear very different even though the head pose remains the same.

is not always available unless special equipment is used. Capturing depth information usually requires two or more cameras or more complex time-of-flight sensors. They usually draw more power than RGB cameras and in some cases are computationally expensive. As depth cameras are only reliable for objects within a limited distance range, head pose estimation algorithms based on depth information may fail if the target head is outside of the working distance.

In addition to the higher cost of obtaining reliable depth information, it is inconvenient or impractical to use 3D sensors for certain real-world applications. Researchers have attempted to estimate head pose directly from RGB images [16-19] to address these challenges. This approach can be divided into two categories: landmark-based methods and landmark-free methods. Landmark-based methods use the locations or geometric information of facial landmarks to estimate the head pose [20, 21]. Although landmark-based methods obtain very accurate results for small head pose angles, they are sensitive to occlusions or large head pose angles. For profile views, landmark-based methods may fail to detect the required facial landmarks accurately, leading to low performance. Rather than relying solely on specific facial landmarks, landmark-free methods estimate the head pose based on features extracted from the face image. Many machine learning algorithms have been used to improve the estimation accuracy. Recently, as convolutional neural networks (CNNs) have exhibited excellent performance in many regression and classification tasks, landmark-free methods using CNNs have become particularly popular due to their robustness against environmental variations and face occlusion. They have obtained very good accuracy for large pose angles and are robust against face occlusion [16, 19].

To the best of our knowledge, all current landmark-free methods ignore the negative impact of perspective distortion on head pose estimation accuracy. In other words, they ignore the fact that the location of the face, in relation to the camera coordinate system, affects how the face projects onto the camera image plane. As shown in Fig. 2, faces with the same head pose presented at different locations in relation to the camera coordinate system result in different face image projections. Fig. 2(a) shows a face that is aligned perfectly with the camera coordinate system with zero pitch, yaw, and roll. As the face moves away from the camera (Fig. 2(b)), to the left or right (Fig. 2(c)), or up or down (Fig. 2(d)), the projections of the face are different even though the head pose remains the same. Using these perspective-distorted images directly for head pose estimation, as all current landmark-free methods do, affects the estimation accuracy.

The hypothesis of this research is that head pose estimation accuracy can be improved if the perspective distortion of the face image can be corrected. We present an image rectification algorithm to correct the perspective distortion of the face image that is caused by the misalignment of the face with the camera coordinate system. The closer the object is to the optical axis, the less the image is affected by perspective distortion [22]. We use rotation transformation to obtain a rectified image that looks as if it were taken by a perfectly aligned camera. Specifically, the face region of the input image is transformed onto a virtual image plane using perspective transformation, and the head pose is estimated using the rectified image. Finally, the estimated head pose in the virtual camera coordinate system is transformed back to the real camera coordinate system to obtain the actual head pose in the camera coordinate system.

A lightweight convolutional network based on depth separable convolution is designed to obtain high estimation accuracy. We aimed at meeting the criteria of a small model size and fast processing speed, which are critical for applications on resource-limited platforms. The model size of our network is merely 0.88MB. Experimental results on public benchmark datasets show that our approach of using image rectification and the special design of our lightweight network achieves both higher accuracy and faster speed than other state-of-the-art methods.

II. Related Work

Landmark-based methods and landmark-free methods are two main approaches for RGB image-based head posed estimation. Landmark-based methods estimate head pose by investigating the geometric relations implied among facial landmarks, while landmark-free methods explore the whole face image and estimate the head pose directly from image intensities.

Huang et al. proved that the 3D pose of a 3-point configuration could be uniquely determined by its 2D projection using the weak-perspective projection model [23]. Huang's discovery became the foundation of many landmark-based head pose estimation methods. The underlying idea of these methods is to model a human face by using several key points (landmarks), and estimate the head pose using the constructed model. To locate these landmarks accurately, some works used regression-based methods to align a face shape template to the real face shape [24-26], and found the locations of landmarks on the real face. A recent design integrated several point-regression tasks on the image plane under one common target, and simultaneously achieved accurate landmarks, face detection, 2D face alignment and 3D face reconstruction with efficient single-shot inference [27].

Recently, several robust facial landmark detectors [7, 28, 29], including 2D and 3D facial landmark detectors, have remarkably improved the accuracy of landmark-based head pose estimation. An efficient facial landmark prediction model has become a prerequisite for a landmark-based head pose estimation method to be successful [30]. 3D facial landmark detectors provide extra depth information for facial landmarks. The depth information provides a clue for extracting the human head contours, and helps improve the accuracy of landmark-based head pose estimation [31]. However, similar to the problem that occurs when using 2D-landmark-based head pose estimation methods, a face model must be defined if one wants to utilize the depth information of facial landmarks. Due to the uniqueness of each individual's biometrics, the predefined face model may not completely fit the real face of each person, which leads to errors in the estimated head pose. To address this problem, a deformable model was proposed to characterize the biometric differences of the face [17, 32]. The deformable face model has been proven to be effective. Nevertheless, the accuracy of these methods is still sensitive to the landmark detection accuracy. In some cases, insufficient landmark points may lead to degraded performance.

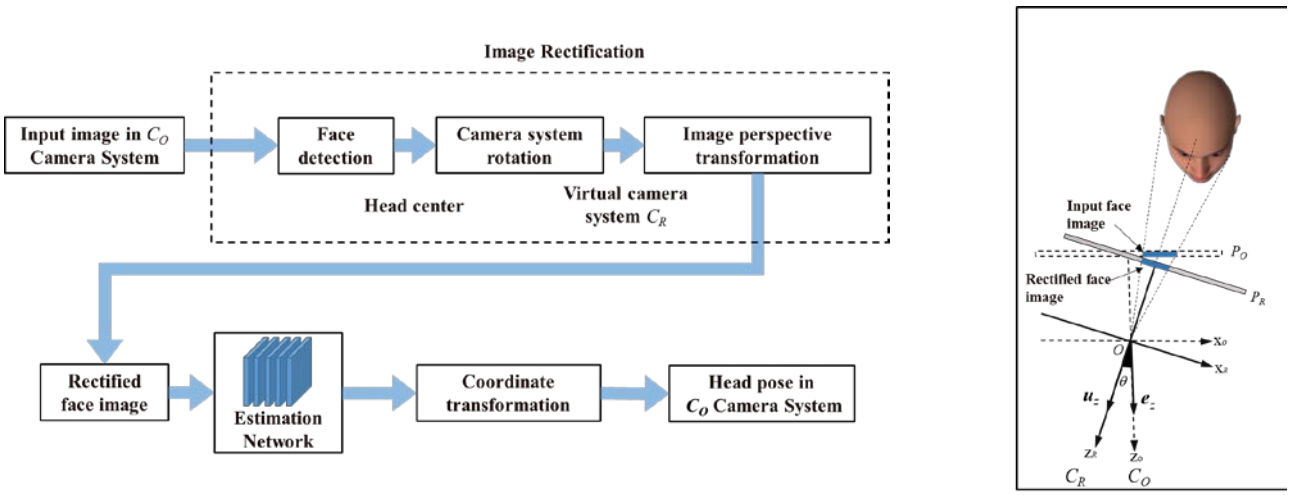
Rather than exploring the geometric relations among facial landmarks, landmark-free methods estimate the head pose by learning the characteristics of the whole image [6, 16, 33]. Benefitting from the advancement of machine learning techniques, especially the success of deep learning, landmark-free methods have achieved promising results on many widely used datasets and have become a popular approach for head pose estimation. Chang et al. [6] used a simple convolutional neural network to regress head poses and improved the face alignment accuracy with the help of the predicted head pose. Patacchiola and Cangelosi [34] explored the role of dropout and adaptive gradient methods in CNN-based models for head pose estimation in the wild. Ruiz et al. [16] employed a multiloss ResNet50, which is also cited as Hopenet, for head pose prediction and achieved accurate results through joint pose classification and regression. Gu et al. [18] used a VGG network to predict head poses in video frames. They utilized a recurrent neural network to leverage the temporal structure for head pose estimation to obtain accurate head pose estimation.

Current landmark-free methods are based on an unrealistic assumption that the face image is formed using weak-perspective projection, which is that the face image remains the same regardless of the position of the face in relation to the camera coordinate system. In reality, for the calibrated pinhole camera, the object is directly projected onto the image plane using full-perspective projection. The approximation error introduced by the weak-perspective projection is closely related to the position of the object. The research on gaze estimation has indicated that when the face center moves away from the optical axis of the camera, the projected images using full-perspective and weak perspective projections are remarkably different [2]. As proven in [22], the approximation error increases as the face center moves away from the optical axis.

Using weak-perspective projection, the translation of the head does not affect the projected face image. In reality, using full-perspective projection as shown in Fig. 2, the position of the head changes the projected image due to perspective distortion and, hence, affects the head pose estimation accuracy. After a translation of the head along the x-axis of the camera coordinate system, as shown in Fig. 2(c), the appearance of the head is perceived as a yaw rotation of the head. Similarly, after a translation of the head along the y-axis of the camera coordinate, as shown in Fig.2(d), the appearance of the head is perceived as a pitch rotation of the head. When the head is close to the optical axis or the center of the image, the accuracy of the current head pose estimation methods is quite good because the perspective distortion is negligible. When the head is positioned away from the camera optical axis and the perspective distortion is more evident, their performance suffers.

III. Image Rectification

In this paper, we propose the use of image rectification to reduce the impact of perspective distortion and design a lightweight CNN for head pose estimation. We mathematically rectify the input image (captured when the head center is not aligned with the camera coordinate system) to align the center of the head with the optical axis of a virtual camera coordinate system. As shown in Fig. 3(b), C_O denotes the real camera system, and C_R denotes the virtual camera system in which the center of the head is aligned with the virtual optical axis. Our method first detects the face in the input image and uses the center of the face bounding box as the head center. The detected face region of the input image in the real camera system C_O is transformed to the virtual camera system C_R by perspective warping. After warping, the head center in the rectified image is aligned with the virtual optical center. The face region of the rectified image is cropped and subsequently used as the input of the estimation network to estimate head pose in the virtual camera system C_R. Finally, the output of the estimation network



(a) (b)

Fig. 3. (a) Flowchart of the proposed head pose estimation method and (b) coordinate transformation for image rectification. P_O and P_R represent the image plane of the camera system and the virtual image plane of the virtual camera system, respectively.

is transformed back to the camera coordinate system C_O as the final head pose estimation.

A. *Camera system transformation*

In our method, C_O denotes the camera system, in which the human face is projected onto the image plane P_O through the pinhole camera model. As shown in Fig. 3 (b), the input image projected on P_O is distorted when the face is positioned away from the optical axis of the camera. This distortion leads to head pose estimation errors. Image rectification is achieved by mathematically rotating the camera so that the head center is aligned with the optical axis of the virtual camera system. The virtual camera system is denoted as C_R. The face region of the input image is transformed onto the virtual image plane P_R of C_R to obtain the rectified image. Head pose estimation is performed on the rectified image.

A rotation matrix defining the transformation between the camera system C_O and the virtual camera system C_R must be calculated to transform the center of the face in C_O onto the optical axis in C_R. The rotation axis vector $\mathbf{r}$ and rotation angle θ determine how the face image must be rotated in 3D for rectification. As shown in Fig. 3 (b), the unit vector of the projection of the head center to the camera optical center O is denoted by $\mathbf{u_z}$ and the unit vector on the z-axis of C_O is denoted by $\mathbf{e_z}$. Suppose $\mathbf{K}$ is the camera projection matrix, *i.e.* $\mathbf{K} = [f_x, 0, c_x; 0, f_y, c_y; 0, 0, 1]$, where f_x and f_y are the focal lengths of the camera in the x and y directions, respectively, and (c_x, c_y) is the location of the image optical center. The homogeneous coordinate of the head center is denoted by $\mathbf{p}$; $\mathbf{p} = [p_x, p_y, 1]^T$, where (p_x, p_y) represents the head center in the input image, and $\mathbf{c}$ is the 3D position of the head center in C_O, *i.e.* $\mathbf{c} = [x_c, y_c, z_c]^T$. The 3D projection can be defined as $\mathbf{c} = z_c\mathbf{K}^{-1}\mathbf{p}$. The unit vector of the projection of the head center to the camera optical center is $\mathbf{u}_z = \dfrac{\mathbf{c}}{\|\mathbf{c}\|_2} = \dfrac{\mathbf{K}^{-1}\mathbf{p}}{\|\mathbf{K}^{-1}\mathbf{p}\|_2}$. $\mathbf{r}$ and θ can be calculated as shown in Eq. 1 and Eq. 2.

$$\mathbf{r} = \mathbf{u}_z \times \mathbf{e}_z , \tag{1}$$

$$\theta = \arccos(\mathbf{u}_z^T \mathbf{e}_z) . \tag{2}$$

The center of the face bounding box is used as the head center. Assuming $\boldsymbol{r}=[r_x, r_y, r_z]^T$ where $r_x^2 + r_y^2 + r_z^2 = 1$, the rotation matrix $\mathbf{M}_{O\to R}$ which defines the transformation between C_O and C_R is expressed as Eq. 3 [35].

$$\mathbf{M}_{O\to R} = \begin{bmatrix} \cos\theta + r_x^2(1-\cos\theta) & r_x r_y(1-\cos\theta) - r_z\sin\theta & r_x r_z(1-\cos\theta) + r_y\sin\theta \\ r_x r_y(1-\cos\theta) + r_z\sin\theta & \cos\theta + r_y^2(1-\cos\theta) & r_y r_z(1-\cos\theta) - r_x\sin\theta \\ r_x r_z(1-\cos\theta) - r_y\sin\theta & r_y r_z(1-\cos\theta) + r_x\sin\theta & \cos\theta + r_z^2(1-\cos\theta) \end{bmatrix} . \tag{3}$$

B. *Image reprojection*

With the rotation matrix $\mathbf{M}_{O\to R}$ obtained in Section III-A, the face region of the input image can be transformed onto the virtual image plane (P_R) in the virtual camera system C_R to calculate the rectified face image in the camera system C_R. Fig. 4 illustrates this process. Through reverse projection, all pixels in the face region of the input image (on P_O) are reprojected back to 3D ($\mathbf{w}$) space in C_O up to an unknown scale factor because the actual face distance is not known. The 3D points ($\mathbf{w}$) in the camera coordinate system C_O are transformed to the virtual camera coordinate system C_R through the rotation matrix $\mathbf{M}_{O\to R}$. The transformed 3D points ($\mathbf{w_r}$) in C_R can then be projected onto the virtual image plane P_R to obtain the rectified image. Details of this rectification process are described below.

We first assume that the depths of all pixels in the face image are known (which we do not) to perform the reverse projection. For a pixel in the input image at (q_{xo}, q_{yo}) with depth z_o (unknown) with respect to camera system C_O, the coordinates of the 3D point ($\mathbf{w}$) corresponding to this pixel can be calculated by a reverse perspective projection, as shown in Eq. 4.

$$\mathbf{w} = z_o\mathbf{K}^{-1}\mathbf{q_o} , \tag{4}$$

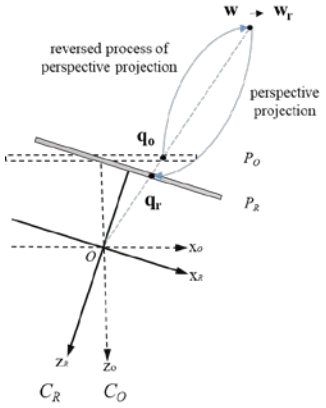


Fig. 4. The rectification transformation from P_O to P_R.

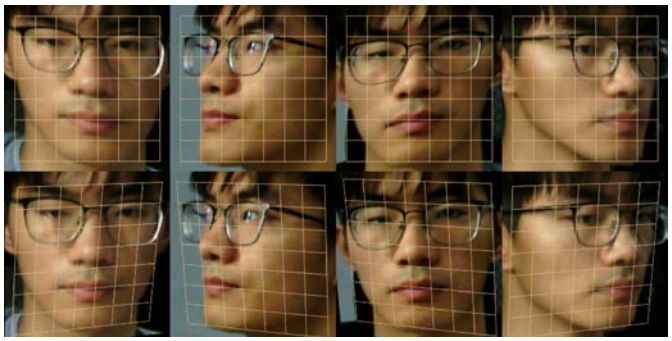


Fig. 5. Examples of images before and after image rectification. The first row shows the images before rectification. The second row shows their corresponding rectified images. The gridlines show the effect of rectification.

where **K** is the camera projection matrix and $\mathbf{q_o}$ is the homogeneous coordinate of the pixel.

The rotation matrix $\mathbf{M}_{O\to R}$, as defined in Section III-A, transforms the 3D points in C_O to C_R. The coordinates of the 3D points ($\mathbf{w_r}$) in C_R can be obtained by multiplying the coordinates of the 3D points ($\mathbf{w}$) in C_O with the rotation matrix $\mathbf{M}_{O\to R}$. The coordinates of the 3D points in C_R ($\mathbf{w_r}$) can be calculated using Eq. 5.

$$\mathbf{w_r} = \mathbf{M}_{O\to R}\mathbf{w}\,. \tag{5}$$

The transformed 3D point ($\mathbf{w_r}$) can then be projected onto the virtual image plane P_R of the virtual camera system C_R using full-perspective projection. Considering that C_O and C_R share the same camera projection matrix **K,** the homogeneous coordinate $\mathbf{q_r}$ of the pixel projected from $\mathbf{w_r}$ can be calculated using Eq. 6.

$$\mathbf{q_r} = \frac{1}{z_r}\mathbf{K}\mathbf{w_r}\,, \tag{6}$$

where z_r is the depth of the 3D point $\mathbf{w_r}$ in the virtual camera coordinate system or the 3rd element of $\mathbf{w_r}$.

All pixels in the face region of the input image can be transformed or rectified using Eqs. (4)-(6). By combining these three equations, the whole rectification process can be formulated with Eq. 7.

$$\frac{z_r}{z_o}\mathbf{q_r} = \mathbf{T}\mathbf{q_o}\,, \tag{7}$$

where the transformation matrix $\mathbf{T} = \mathbf{K}\mathbf{M}_{O\to R}\mathbf{K}^{-1}$. Suppose the pixel at $\mathbf{q_o} = [q_{x_o}, q_{y_o}, 1]^{\mathrm{T}}$ in C_O is transformed to $\mathbf{q_r}=[q_{x_r}, q_{y_r}, 1]^{\mathrm{T}}$ in C_R. The transformation can be represented as follows: $\frac{z_r}{z_o}\begin{bmatrix} q_{x_r} \\ q_{y_r} \\ 1 \end{bmatrix} = \mathbf{T}\begin{bmatrix} q_{x_o} \\ q_{y_o} \\ 1 \end{bmatrix}$. The scale factor $\frac{z_r}{z_0}$ can be calculated as follows: $\frac{z_r}{z_0} = T_{31}\, q_{x_o} + T_{32}\, q_{y_o} + T_{33}$, where T_{mn} represents the parameters of the m-th row and the n-th column of matrix **T**. This proves that the actual depth z_o in C_O or z_r in C_R is not needed for rectification. Only their ratio is needed, which is embedded in the transformation matrix **T**.

Based on the method above, every pixel of the input face image can be projected onto the virtual image plane of the virtual camera system C_R to obtain the rectified face image. Some sample images before and after rectification are shown in Fig. 5.

C. Head pose transformation

The ground truth head pose included in all benchmarking datasets for evaluating head pose estimation algorithms is measured or estimated with respect to the real camera system C_O. Our head pose estimation network operates on the rectified image and estimates the head pose in the virtual camera coordinate system. The ground truth head pose provided by the datasets must be transformed for use in the virtual camera system C_R to train our network and estimate the head pose in C_R. To evaluate the accuracy of our head pose estimation algorithm, the estimated head pose from our network is then transformed back to the camera system C_O for accuracy evaluation. The head pose estimation represented as Euler angles can be converted into the form of a rotation matrix as follows:

$$\mathbf{M}(\alpha,\beta,\gamma) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{bmatrix}\begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{bmatrix}\begin{bmatrix} \cos(\gamma) & -\sin(\gamma) & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{8}$$

where α, β, and γ are the angles of pitch, yaw, and roll, respectively [35]. The ground truth head pose provided by the datasets in camera system C_O can be converted to the same matrix form and is denoted by $\mathbf{M}_O$. Similarly, $\mathbf{M}_R$ represents the rotation matrix of the head pose in the virtual camera system C_R. $\mathbf{M}_R$ can be calculated using the rotation matrix $\mathbf{M}_{O\to R}$ defined in Eq. 3 as follows:

$$\mathbf{M}_R = \mathbf{M}_{O\to R}\mathbf{M}_O\,. \tag{9}$$

The Euler angles in C_R can be calculated by Eq. 10.

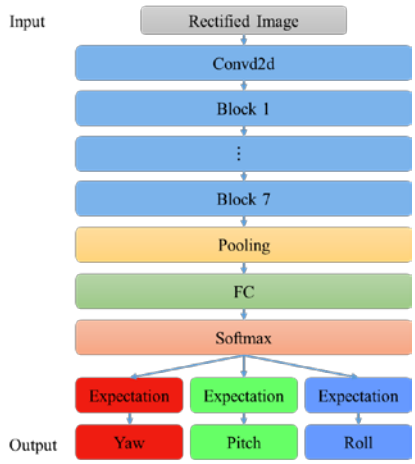


Fig. 6. The architecture of the proposed network.

$$\alpha_R = \arctan(\frac{-M_{R23}}{M_{R33}}),$$
$$\beta_R = \arcsin(M_{R13}), \quad (10)$$
$$\gamma_R = \arctan(\frac{-M_{R12}}{M_{R11}}),$$

where α_R, β_R and γ_R represent pitch, yaw, and roll in C_R, respectively. M_{Rmn} represents the elements of matrix $\mathbf{M}_R$ in the m^{-th} row and the n^{-th} column.

All training images in the datasets are transformed from C_O to C_R using the image rectification method described in Section III-B. The ground truth head pose is also transformed from C_O to C_R using Eq. 10. Our network is trained using the transformed images and ground truth head pose in C_R. For testing, all test images in the datasets must also be rectified or transformed from C_O to C_R. These rectified test images are used for the estimation network to obtain better head pose estimation. Since our network is trained in C_R, the head pose estimation (in terms of Euler angles) output from the network is also in C_R. Our head pose estimation in C_R is transformed back to C_O to be compared with the original ground truth provided in C_O to avoid any possible negative impact on the ground truth by our transformations.

The head pose estimation represented by Euler angles is converted to a matrix $\mathbf{M}_{R\text{-}P}$ using Eq. 8. As a reverse operation of Eq. 9, the head pose estimation in matrix form is transformed back to the camera coordinate system C_O using Eq. 11.

$$\mathbf{M}_{O-P} = \mathbf{M}^{-1}_{O\to R}\mathbf{M}_{R-P}, \quad (11)$$

where $\mathbf{M}_{O\text{-}P}$ is the head pose estimation in matrix form in camera coordinate system C_O. The final Euler angles in C_O for performance evaluation can then be calculated using Eq. 10.

IV. Estimation Network

A lightweight convolutional neural network is designed to estimate the head pose from the rectified image. The proposed network is based on group convolution and depthwise separable convolution techniques. The architecture of the proposed network is shown in Fig. 6. The model size of our network is quite small. A new loss function that is dedicated for this application is designed to achieve high accuracy in head pose estimation.

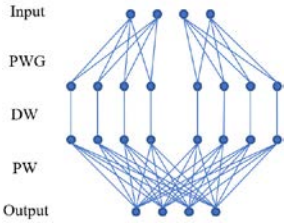


Fig. 7. The structure of our building block.

A. Network architecture

Depth separable convolution is a form of factorized convolution, including depthwise convolution (DW) and pointwise convolution (PW). Using group convolution and depthwise separable convolution techniques in repeated building blocks is an efficient way to construct lightweight networks in many CNN-based recognition algorithms. For example, MobileNetV1[36] and Xception [37] utilized building blocks based on depthwise separable convolution to achieve a good tradeoff between accuracy and computational cost. MobileNetV2 [38] introduced an expansion layer (pointwise convolution) before depthwise separable convolution within a building block to achieve much improved performance. In the building blocks of MobileNetV2, input features are projected into high dimensions using the expansion layer and then filtered with depthwise convolution. The features are subsequently projected to low dimensions by pointwise convolution. The experimental results in [38] showed that using the expansion layer was crucial because it prevents nonlinearities from losing too much information.

TABLE I
DETAILS OF THE PROPOSED NETWORK. LET k DENOTE THE KERNEL SIZE AND s DENOTE THE STRIDE. NL DENOTES THE TYPE OF NONLINEARITY. CONV2D DENOTES STANDARD CONVOLUTION. BLOCK DENOTES THE PROPOSED EFFICIENT BUILDING BLOCK. FC DENOTES FULLY CONNECTED LAYER. HS DENOTES H-SWISH AND RE DENOTES RELU.

Input	Operator	k	s	NL
$224^2\times3$	conv2d	3×3	2	RE
$112^2\times16$	block	3×3	2	RE
$56^2\times32$	block	3×3	2	RE
$28^2\times64$	block	3×3	1	RE
$28^2\times96$	block	5×5	2	HS
$14^2\times128$	block	5×5	1	HS
$14^2\times128$	block	5×5	2	HS
$7^2\times128$	block	3×3	1	HS
$7^2\times128$	avgpool 7x7	-	-	-
1×128	FC	-	-	-
3×66	softmax	-	-	-

Inspired by the success of MobileNetV1 and MobileNetV2, we propose a lightweight network for head pose estimation. The architecture of the proposed network is shown in Fig. 6. It is composed of a stack of building blocks. A building block in our network is constructed with pointwise group convolution (PWG)

Fig. 8. Examples of unreliable training data in the 300W-LP dataset.

and depthwise separable convolution techniques, as shown in Fig. 7. Pointwise group convolution can be regarded as a special implementation of group convolution using kernels with a 1×1 spatial size. Compared with pointwise convolution, pointwise group convolution uses fewer parameters. We use pointwise group convolution to project input features into high dimensions. The high-dimensional features are further processed by depthwise separable convolution. The expansion factor is set to 2 and the number of groups is set to 4 to balance the performance and the size of the network. In the building blocks of our network, each convolution layer is followed by a batch normalization operation and an activation layer.

In the proposed network, we use the following two types of activation functions: ReLU and h-swish [39]. Previous work in MobileNetV3 [39] shows that, compared with ReLU, the introduction of h-swish is able to improve the performance of networks at the cost of more computations. Both activation functions are used in our network to achieve a better balance between the model accuracy and computational cost. Details of the proposed network are listed in Table I.

The loss function employed in the proposed network combines the output of head pose regression and head pose classification. Assuming the range of each Euler angle is [-*M*, *M*] and is split into n intervals, the length of each interval is $2M/n$. In our network, n is set to 66. This design turns head pose estimation into a classification problem. A softmax layer is employed to predict the classification probability for each interval. The sum of the midpoint value of each interval, which is weighted by classification probability, is used to represent the final estimation.

B. *Discriminative loss function*

Previous works on head pose estimation mostly treated all training images equally and assumed that the ground truth head pose from each image provides the same amount of information for training the network. Unreliable training data may deteriorate the performance of the network [40, 41]. Investigations found that in many head pose datasets, e.g., synthetic 300W-LP, there are samples with distinct quality and reliability issues, especially cases with large head poses. As shown in Fig. 8, some samples in the 300W-LP dataset are remarkably distorted and cannot provide reliable supervision to train the network.

Considering the imperfections in some head pose datasets, a discriminative head pose-based weighted loss is used to train our network. First, the sum of cross entropy loss and MSE loss is used to represent the loss of the network output for each image. Then, the weight for the loss of each image is set in each batch. Let δ denote the angle in degrees between the z-axis of the camera coordinate system and the z-axis of the head coordinate system. Using the ground truth head pose provided by the datasets, we can obtain δ with Eq. 12, where $\mathbf{M}_O$ is the head rotation matrix and $\mathbf{u}$ is $[0, 0, 1]^T$.

$$\delta = \arccos(\mathbf{u}^T \mathbf{M}_o \mathbf{u}). \tag{12}$$

$$weight = \begin{cases} 1 & if \ \ \delta < 60 \\ 0.5^{\frac{\delta-60}{5}} & if \ \ \delta > 60 \end{cases}. \tag{13}$$

To obtain the weight of loss for each image, we normalize δ to the range of [0, 1] using Eq. 13. If δ is less than 60 degrees, the weight is set to 1. When δ is greater than 60 degrees, the weight decreases exponentially by a factor of 0.5. The losses for pitch, yaw, and roll are calculated separately. The loss function for each Euler angle can be formulated as in Eq. 14.

$$L = \frac{1}{N}\sum_{i=0}^{N-1} \omega_i \left[CE\left(y_i, \hat{y}_i\right) + MSE\left(y_i, \hat{y}_i\right) \right] \tag{14}$$

where *CE* and *MSE* represent cross entropy and mean squared error loss functions, respectively. N denotes the number of images in one batch of data. ω_i denotes the weight of loss for the i-th image. y_i is the ground truth of the i-th image, while $\hat{y}_i$ is the estimation result. The sum of the loss for each Euler angle is used to represent the overall loss as follows:

$$L_{Total} = L_{Pitch} + L_{Yaw} + L_{Roll}. \tag{15}$$

V. Experiments and Discussion

A major challenge for developing head pose estimation algorithms is the collection of a large number of face images with their accompanying ground truth head poses. Measuring the true head pose at Euler angles is not a trivial task when no special equipment, such as magnetic sensors, is used. Many widely used datasets for head pose estimation were created by cropping out the face region directly from photos or video frames downloaded from the Internet or other similar sources. Because these photos and videos were taken or recorded without their accompanying head pose measurements, the head pose ground truth represented in Euler angles from these datasets had to be estimated from 2D images without any 3D information. Estimating head poses without 3D information inevitably introduces errors in the calculation. As the face could appear anywhere in a photo or video frame, these images could be distorted because of perspective distortion, especially when the face is away from the optical axis of the camera.

The greatest advantage of using the head poses estimated from 2D images as the ground truth in many head pose datasets is that a large dataset could be collected with a relatively easy process without special setups and with low costs. The biggest drawback is that the landmark-free head pose estimation algorithms would be trained and evaluated with imperfect ground truth head pose estimated from 2D images. We selected three popular head pose datasets for our experiments. Of these three datasets, the BIWI dataset is the only one that includes the ground truth head pose calculated from 3D information [42]. The other two datasets, 300W-LP [43] and AFLW2000 [43], use head pose estimated from 2D images as the ground truth.

Even though their ground truth head pose could be affected by perspective distortion, performance evaluation using these two datasets still has value. Many published works used these two datasets for performance evaluation, which allows us to easily compare our approach with other state-of-the-art methods.

We conducted three experiments using the three aforementioned datasets. Our experimental results show that our method outperformed the state-of-the-art methods in terms of accuracy and processing speed. Further experiment shows that the accuracy of head pose estimation was remarkably improved with the help of the proposed rectification and the discriminative weighted loss function.

A. Datasets and their ground truth

The BIWI dataset was collected in a lab environment. It contains 24 videos of 20 people (6 females and 14 males), with approximately 15,000 images in total [42]. RGB images and depth information were included for each frame. The head poses of these images are in the range of ±75° for yaw, ±60° for pitch, and ±50° for roll. The BIWI ground truth head pose was estimated using a 3D morphable head model to generate a personalized head template for each subject. The generated 3D head template was rotated so that it aligned with the 3D head model generated with the 3D measurements from Microsoft Kinect. The required amount of rotation in all three axes to align the 3D head template with the Kinect-measured 3D head model represents a fairly accurate head pose. Because the 3D head template was generated in the camera coordinate system, the estimated head pose was very close to the actual head pose when the image was captured.

300W-LP utilized face profiling with a 3D image model to expand the 300W dataset which combines several face alignment datasets including LFPW [44], AFW [45], HELEN [46], IBUG [47], and XMSVTS [48], with 68 landmarks for each face image. As a result, 61,225 samples (16556 from LFPW [44], 5,207 from AFW [45], 37676 from HELEN [46] and 1,786 from IBUG [47]) were generated. 300WP-LP further flipped all these samples to double the sample size to 122,450. AFLW2000 is a very challenging dataset with large pose variations with various facial expressions and illumination conditions. AFLW2000 provides face images, corresponding ground truth head pose, and 3D face for the first 2000 images in AFLW [49].

Both the 300WP-LP and AFLW2000 datasets used the same technique to estimate the ground truth head pose from 2D images. Similar to BIWI, a 3D morphable model was used to generate a 3D face template for each subject in the 300W-LP and AFLW2000 databases. Unlike BIWI, which fit the 3D head template to the 3D head model measured with Microsoft Kinect, the aim of the fitting process for these two datasets was to minimize the difference between the images projected mathematically from the 3D head template and the collected face images. A 3D head template with the required amount of rotation in all three axes to project a 2D face image similar to the collected face image is used as the head pose ground truth.

Because of the lack of intrinsic camera parameters and 3D information and for the ease of computation, the projection of the 3D head template to the 2D face image was calculated using weak-perspective projection. Unlike the images from full-perspective projection shown in Fig. 2, the linear approximation of weak-perspective projection resulted in the same projection regardless of the position of the face. The estimated head pose is affected by this linear approximation because the collected face image is matched to an unrealistic projection, especially when the face is presented away from the optical axis of the camera. This ground truth generation procedure would have been more realistic if the projection of the 3D head template to a 2D image could use full-perspective projection.

In summary, as mentioned in Section III, the camera projection matrix plays a critical role in projecting realistic images. BIWI is the only dataset that provides the projection matrix of the camera used to capture images. It is also the only dataset that estimates the ground truth head pose using 3D information. It is the ideal dataset for our work that focuses on minimizing the negative impact of perspective distortion to obtain high head pose estimation accuracy. We included 300W-LP and AFLW2000 in our experiments even though their ground truth head pose could be affected by perspective distortion. We believe performance evaluation using these two datasets still has value.

B. Configuration and evaluation metrics

As shown in Fig. 3, the first step of our head pose estimation is face detection. Face detection was performed using DSFD [50] and the face bounding box center was used to approximate the head center. Our network was trained for 70 epochs using Adam optimization [51] with an initial learning rate of 10^{-3}, $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The learning rate was reduced at the 30^{-th} epoch and the 50^{-th} epoch by a factor of 0.1 each. For data augmentation, random cropping, random downsampling and random scaling were applied to the training images. The batch size was set to 8 for Experiment I. The batch size was set to 16 for Experiment II and Experiment III. Each color channel was normalized using the ImageNet mean and standard deviation before training and testing.

Two evaluation metrics were used in all three experiments. One was the average error in the yaw, pitch, and roll angles in degrees. The other metric was the mean absolute error (MAE), which was calculated as follows:

$$\mathbf{MAE} = \frac{1}{3N}\sum_{i=0}^{N-1}\left(\left|\alpha_i - \hat{\alpha}_i\right| + \left|\beta_i - \hat{\beta}_i\right| + \left|\gamma_i - \hat{\gamma}_i\right|\right) \tag{16}$$

where N is the number of samples in the testing dataset. $(\alpha_i, \beta_i, \gamma_i)$ are the ground truth Euler angles and $(\hat{\alpha}_i, \hat{\beta}_i, \hat{\gamma}_i)$ are the estimated Euler angles.

C. Experiment I

As discussed previously, BIWI is the only dataset that includes the ground truth head pose estimated from 3D information. Using BIWI for performance evaluation can better demonstrate the importance of correcting perspective distortion for estimating accurate head poses. All images in the dataset were rectified, and their ground truth head pose was transformed from camera system C_O to virtual camera system C_R. Our network was trained and tested in C_R. To avoid

introducing errors to the ground truth, the head pose estimation in C_R from our network was then transformed back to C_O to compare against the original ground truth head pose from the dataset and compute the average error.

Four state-of-the-art methods were included for comparison because they all used the same BIWI for testing and training. The performance was evaluated in terms of accuracy and model size. FSA-Net [18] and Hopenet [16] are two state-of-the-art head pose estimators using a single RGB image, similar to our method. Gu et al. combined CNN and RNN and used both the RGB images and temporal information included in the dataset to estimate the head pose [18]. Martin et al. utilized both RGB images and depth images to estimate the head pose [13].

We followed the experimental protocol proposed in [52] to randomly split the BIWI dataset into 70% (16 videos) for training and 30% (8 videos) for testing. We repeated this process three times and reported the average measurement error. The average errors in degrees and the network sizes in MB of four other networks are listed in Table II.

Our method had much lower average errors in yaw, pitch, roll angles, and MAE with only one exception. Our method had a slightly higher average error in pitch angle than the method presented in [13]. It is worth emphasizing that the method presented in [13] used both RGB image and depth information, which might not be a fair comparison for methods using only RGB images. We included it for comparison to demonstrate the importance of correcting perspective distortion to obtain accurate head pose estimations. Compared to the methods using CNN, our method had the smallest model size.

TABLE II
SIZE AND AVERAGE ERROR COMPARISONS USING THE BIWI DATASET FOR TRAINING AND TESTING.

	Model Size (MB)	Yaw	Pitch	Roll	MAE
FSA-Net [53]	5.1	2.89	4.29	3.60	3.60
Hopenet [16]	95.9	3.29	3.39	3.00	3.23
Gu et al. [18]	>500	3.14	3.48	2.60	3.07
Martin et al. [13]	-	3.62	**2.54**	2.57	2.91
Ours	**0.88**	**2.24**	2.81	**2.37**	**2.47**

D. *Experiment II*

Of the three datasets included in our study, 300W-LP is the largest dataset, which includes 122,450 images. It has become a very popular benchmarking dataset due to its sample size. As discussed in Section V-A, its ground truth head pose was estimated from a 2D image from weak-perspective projection. For this reason, many researchers have used 300W-LP or AFLW for training and BIWI for testing [16, 28, 53, 54, 55]. We followed this same data arrangement to demonstrate the performance of our method. The camera intrinsic parameters are not available for the 300W-LP dataset because its images were collected from the Internet. Like other methods, the original images in the dataset were used for training without rectification because of the lack of camera intrinsic parameters.

For testing, similar to Experiment I, the images in BIWI were rectified and sent to our network for head pose estimation. The estimated head pose was then transformed back to C_O to calculate the average errors in the Euler angles. We compared our approach with six state-of-the-art head pose estimation methods using the BIWI dataset. The results are shown in Table III.

TABLE III
SIZE AND AVERAGE ERROR COMPARISONS ON THE BIWI DATASET USING 300W-LP AS THE TRAINING DATASET.

	Model Size (MB)	Yaw	Pitch	Roll	MAE
Dlib [16, 28]	-	16.76	13.80	6.19	12.25
FAN [7, 16]	183	8.53	7.48	7.63	7.88
Hopenet [16]	95.9	4.81	6.61	3.27	4.90
QuatNet [54]	-	4.01	5.49	2.94	4.15
FSA-Net [53]	5.1	4.27	4.96	2.76	4.00
Hopenet [16]+IR	95.9	4.05	5.47	2.91	4.14
Img2pose [55]	165.9	4.57	**3.55**	3.24	3.79
Ours	**0.88**	**3.59**	3.94	**2.68**	**3.40**

Several experiments were reported in [16]. Two of them used landmark detectors FAN [7] and Dlib [28] on the BIWI dataset to estimate the head pose based on the geometric information of the landmarks. These two experiments are included in our comparison and are cited as Dlib [28] and FAN [7]. Hopenet [16], FSA-Net [53] and QuatNet [54] are three state-of-the-art methods that estimate the head pose directly from image intensities. Img2pose [55] estimates 6DoF 3D face poses directly from raw images without face detection or landmark localization. To demonstrate the effect of our image rectification method, we ran the Hopenet in [16] using rectified images as the input. We included the result as Hopenet [16]+IR in Table III.

As shown in Table III, our method obtained the best performance among all methods. Compared with the lightweight FSA-Net, our method obtained higher accuracy with a smaller model size. Our network is less than one fifth of the size of FSA-Net and has a 15% lower MAE. Our image rectification method clearly improved Hopenet's performance (Hopenet [16] vs. Hopenet [16]+IR). The improvement is across the board and with 15% lower MAE.

E. *Experiment III*

As opposed to the limited variations in terms of the range of head poses and illumination conditions in BIWI, AFLW2000 was created to evaluate the performance of head pose estimation methods in dealing with large head poses and varying illumination conditions [16, 28, 43, 53, 54, 55]. In this experiment, similar to other methods, our network was trained on the 300W-LP dataset and tested on the AFLW2000 dataset. In this scenario, the proposed image rectification was not applied to the test images because AFLW2000 does not provide the intrinsic parameters of the camera corresponding to each face image. This experiment demonstrates the improvement by

using our network and the proposed discriminative loss function.

We followed the same settings as Hopenet [16] and only considered samples whose Euler angles were within the range of [-99°, 99°]. We compared our approach with seven state-of-the-art methods using the same protocol. Since ground truth facial landmarks are available in the AFLW2000 dataset, [16] also used the ground truth facial landmarks to estimate the head pose. The estimation results using ground truth facial landmarks obtained by [16] are cited as 'Landmarks [16]' in Table IV.

TABLE IV
COMPARISONS OF PERFORMANCE WHEN TRAINING WITH THE 300W-LP DATASET AND TESTING ON THE AFLW2000 DATASET.

	Model Size (MB)	Yaw	Pitch	Roll	MAE
Dlib [16, 28]	-	23.2	13.6	10.5	15.8
FAN [7, 16]	183	6.36	12.3	8.71	9.12
Landmarks [16]	-	5.92	11.76	8.27	8.65
3DDFA [43]	-	5.40	8.53	8.25	7.39
Hopenet [16]	95.9	6.47	6.56	5.44	6.16
FSA-Net [53]	5.1	4.50	6.08	4.64	5.07
QuatNet [54]	-	3.97	5.62	3.92	4.50
Img2pose [55]	165.9	3.43	**5.03**	**3.27**	**3.91**
Ours	**0.88**	**3.36**	5.05	3.56	3.99

As shown in Table IV, our approach outperformed all other state-of-the-art methods except img2pose. Although our method obtained a slightly higher MAE than img2pose, it requires a much smaller model size than img2pose. The experiment demonstrated that our network obtained highly accurate head pose estimation under conditions with large head poses and varying illumination conditions.

F. Processing Speed

To fully understand the performance of the proposed lightweight network, we conducted an experiment to test the processing speed of our network. We compared our network with three state-of-the-art networks with reasonably small model sizes on GPU and CPU platforms. The GPU platform employed a single NVIDIA GeForce GTX 1080Ti. The version of CUDA library was 10.1. The CPU platform used an Intel Core i7-7700 @ 3.60 GHz processor.

TABLE V
THE SPEED OF POPULAR HEAD POSE ESTIMATION NETWORKS

	Image Size	Model Size (MB)	Number of Parameters (M)	Images/sec.	
				GPU	CPU
FAN [7]	256×256	183	47.64	78	4
Hopenet [16]	224×224	95.9	23.92	385	18
FSA-Net [53]	64×64	5.1	1.17	1029	124
Ours	224×224	0.88	0.21	2910	161

For fair comparison, all networks were tested on the same target platform. Each network was evaluated 100 times. The average runtime is shown in Table V. The small model size that our method offers will become even more critical for embedded applications using a small GPU such as the NVIDIA Jetson Nano or requiring FPGA implementations.

G. Significance of Contributions

Experiments were conducted to show the importance of our image rectification and the discriminative weighted loss function. We compared the estimation results using different experimental settings. In Table VI, image rectification means that the testing dataset was rectified by the proposed image rectification process before estimation. Loss weighting means that our discriminative weighted loss function was used to train our network. For experiments without loss weighting, the weight for the loss for each image was set to 1. The experimental results show that both image rectification and discriminative loss weighting were effective in improving the accuracy of the network. For the experiments performed on the BIWI dataset, Table VI shows that our image rectification method played a very important role in improving the performance of our network. The test results on BIWI and AFLW2000 show that the proposed discriminative weighted loss function led to better accuracy.

One head pose estimation method found in the literature [55] is somewhat close to our approach. Both methods take into consideration the face location in the image coordinates when estimating the head pose but with slight differences. Img2pose [55] crops the face region from the whole image and uses the cropped face region to estimate the head pose. It estimates the 6DoF pose values in the local coordinate frame first and converts the result to a global coordinate frame. Cropping the face region from the whole image could introduce error in the head pose estimation because of the loss of perspective distortion information. However, in our method, we alleviate the effect of perspective distortion in the image before head pose estimation is performed.

Considering that the camera calibration matrix is not always available in some cases, especially for the dataset directly taken from the Internet, the authors of [55] proposed a method to approximate the camera calibration matrix. In this experiment, we followed the same approximation and replaced the true camera intrinsic matrix with the approximated camera calibration matrix proposed by [55] to rectify the images in BIWI, and tested our network using the rectified BIWI dataset. The experimental results are listed in TABLE VI. The experiment results show that better results were obtained using an approximated camera calibration matrix than when using methods without image rectification, while the experiment using the true camera calibration matrix performed the best.

A major challenge in comparing results from so many different sources is that none of them used the exact same criteria, parameters, experimental settings, or even the same datasets. In some cases, a comparison that includes a large number of methods reported in the literature may not be truly fair. To address this challenge, we conducted two more

experiments to isolate the effects from image resolution and data augmentation techniques and included our results in Tables VII and VIII.

TABLE VI
EXPERIMENT RESULTS FOR THE SIGNIFICANCE OF IMAGE RECTIFICATION AND LOSS WEIGHTING. ALL ARE TRAINED ON THE 300W-LP DATASET. * DENOTES THE EXPERIMENT USING AN APPROXIMATED CAMERA CALIBRATION MATRIX.

Testing Set	Image Rectification	Loss Weighting	Yaw	Pitch	Roll	MAE
BIWI	-	-	5.50	5.59	3.18	4.75
	-	√	5.21	5.23	3.03	4.49
	√	-	4.03	4.38	2.84	3.75
	*	√	3.96	4.29	2.85	3.70
	√	√	**3.59**	**3.94**	**2.68**	**3.40**
AFLW2000	-	-	3.67	5.43	3.84	4.31
	-	√	**3.36**	**5.05**	**3.56**	**3.99**

Although we obtained our result included in Table IV with the same experimental settings as other STOA methods [16, 53, 55], we recognize that the reported performances of other methods in Table IV were obtained using input images with different resolutions. For example, Hopenet [16] used 224×224, FSA-Net [53] used 64×64, and QuatNet [54] employed 227 × 227, while other methods included in the table did not even report their image resolutions.

To further understand the influence of image resolution and data augmentation methods, we conducted more experiments on the proposed algorithm. Our results shown in Table IV used 224×224 face images as the input. Lower resolution images could reduce the amount of calculation but affect the performance. We used downsampled face images as the input for our network. Both the training dataset and testing dataset were downsampled. The results are listed in TABLE VII. The experimental results on both BIWI and AFLW2000 show that high resolution input led to better accuracy. The results shown in this table and Table IV demonstrate that our small network using the discriminative loss weighting technique achieves better or comparable performance compared to the state-of-the-art approaches included in Table IV when using the same image resolution.

TABLE VII
THE INFLUENCE OF IMAGE RESOLUTION ON THE BIWI AND AFLW2000 DATASETS. ALL ARE TRAINED ON THE 300W-LP DATASET.

Testing Set	Image resolution	Yaw	Pitch	Roll	MAE
BIWI	28×28	6.49	9.12	4.71	6.77
	56×56	4.85	5.92	3.29	4.69
	112×112	4.30	4.27	2.86	3.81
	224×224	**3.59**	**3.94**	**2.68**	**3.40**
AFLW2000	28×28	6.52	8.56	7.19	7.42
	56×56	4.48	6.55	4.80	5.28
	112×112	3.73	5.30	3.91	4.31
	224×224	**3.36**	**5.05**	**3.56**	**3.99**

Data augmentation also plays a very important role in deep learning research. In our training scheme, random cropping, random scaling and random down sampling were employed. To show the influence of each data augmentation method, we enabled each of these data augmentation methods and explored their impact on accuracy. The experiment was conducted on AFLW2000. The experimental results with and without the loss weighting function are listed in TABLE VIII. The experimental results show that data augmentation remarkably improved the performance of our network.

TABLE VIII
THE INFLUENCE OF DATA AUGMENTATION METHODS ON THE AFLW2000 DATASETS. ALL ARE TRAINED ON THE 300W-LP DATASET. A DENOTES RANDOM CROP. B DENOTES RANDOM DOWNSAMPLING. C DENOTES RANDOM SCALE.

Method			Loss Weighting	Yaw	Pitch	Roll	MAE
A	B	C					
-	-	-	-	3.98	6.47	4.72	5.06
√	-	-	-	4.09	6.11	4.32	4.84
-	√	-	-	3.78	6.35	4.31	4.82
-	-	√	-	3.61	5.85	4.25	4.57
√	√	√	-	3.67	5.43	3.84	4.31
-	-	-	√	3.76	6.00	4.07	4.61
√	-	-	√	3.96	5.53	3.87	4.45
-	√	-	√	3.70	5.67	3.96	4.44
-	-	√	√	3.54	5.43	3.86	4.28
√	√	√	√	**3.36**	**5.05**	**3.56**	**3.99**

VI. CONCLUSION

Face images with large perspective distortion do not accurately reflect the true nature of a head pose. Using face images with large perspective distortion undermines the landmark-free head pose estimation accuracy. We propose an approach involving image rectification to reduce the influence of perspective distortion for head pose estimation. After rectification, the head center is aligned with the camera coordinate system, which improves the accuracy of head pose estimation. We develop a lightweight network (only 0.88 MB) to estimate the head pose using the rectified face image. A discriminative weighted loss function that assigns each training image a weight depending on its head pose is designed to train our network. Five experiments were designed to confirm our hypotheses and test the influence of our approach. Experimental results show that both image rectification and the discriminative weighted loss function contribute to improved accuracy. Our network outperforms state-of-the-art methods on three popular benchmark datasets. We further tested our network in terms of speed. Our network can process 2910 images per second on the NVIDIA GeForce GTX 1080Ti GPU platform and 161 images per second on a CPU platform equipped with an Intel Core i7-7700 @ 3.60 GHz processor. The experimental results verify that our network estimates head poses with high accuracy and at faster speed.

ACKNOWLEDGMENT

This work was supported by National Natural Science Foundation of China (62173353, 62171207), Guangzhou Municipal People's Livelihood Science and Technology Plan

(201903010040), Science and Technology Program of Guangzhou, China (202007030011).

REFERENCES

[1] E. Murphy-Chutorian and M. M. Trivedi, "Head Pose Estimation in Computer Vision: A Survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 4, pp. 607-626, April 2009.

[2] Y. Sugano, Y. Matsushita and Y. Sato, "Learning-by-Synthesis for Appearance-Based 3D Gaze Estimation," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2014, pp. 1821-1828.

[3] K. Cao, Y. Rong, C. Li, X. Tang and C. C. Loy, "Pose-Robust Face Recognition via Deep Residual Equivariant Mapping," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2018, pp. 5187-5196.

[4] B. Lee and W. Chung, "Driver Alertness Monitoring Using Fusion of Facial Features and Bio-Signals," *IEEE Sens. J.*, vol. 12, no. 7, pp. 2416-2422, July 2012.

[5] N. Alioua, A. Amine, A. Rogozan, A. Bensrhair, and M. Rziza, "Driver head pose estimation using efficient descriptor fusion," *EURASIP J. Image Video Process.*, vol. 2016, pp. 1–14, Jane 2016.

[6] F. Chang, A. T. Tran, T. Hassner, I. Masi, R. Nevatia and G. Medioni, "FacePoseNet: Making a Case for Landmark-Free Face Alignment," in *Proc. IEEE Int. Conf. Comput. Vision Workshop*, 2017, pp. 1599-1608.

[7] 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 *Proc. IEEE Int. Conf. Comput. Vision*, 2017, pp. 1021-1030.

[8] F. Zhang, T. Zhang, Q. Mao and C. Xu, "Joint Pose and Expression Modeling for Facial Expression Recognition," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2018, pp. 3359-3368.

[9] L. Tran, X. Yin and X. Liu, "Disentangled Representation Learning GAN for Pose-Invariant Face Recognition," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 1283-1292.

[10] E. Murphy-Chutorian, A. Doshi and M. M. Trivedi, "Head Pose Estimation for Driver Assistance Systems: A Robust Algorithm and Experimental Evaluation," *in Proc. IEEE Intell. Transp. Syst. Conf.*, 2007, pp. 709-714.

[11] J. Ng and S. Gong, "Multi-view face detection and pose estimation using a composite support vector machine across the view sphere," in *Proc. IEEE Int. Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, 1999, pp. 14-21.

[12] R. Rae and H. J. Ritter, "Recognition of human head orientation based on artificial neural networks," *IEEE Trans. Neural Netw.*, vol. 9, no. 2, pp. 257-265, March 1998.

[13] M. Martin, F. v. d. Camp and R. Stiefelhagen, "Real Time Head Model Creation and Head Pose Estimation on Consumer Depth Cameras," in *Proc. Int. Conf. 3D Vision*, 2014, pp. 641-648.

[14] G. Fanelli, T. Weise, J. Gall, and L. V. Gool. "Real time head pose estimation from consumer depth cameras," in *Joint Pattern Recognition Symposium*, R. Mester, M. Felsberg, Ed., Berlin, Germany: Springer, 2011, pp. 101-110. [Online]. Available: https://link.springer.com.

[15] G. P. Meyer, S. Gupta, I. Frosio, D. Reddy and J. Kautz, "Robust Model-Based 3D Head Pose Estimation," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2015, pp. 3649-3657.

[16] N. Ruiz, E. Chong and J. M. Rehg, "Fine-Grained Head Pose Estimation Without Keypoints," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit. Workshops*, 2018, pp. 2074–2083.

[17] M. Krinidis, N. Nikolaidis and I. Pitas, "3-D Head Pose Estimation in Monocular Video Sequences Using Deformable Surfaces and Radial Basis Functions," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 2, pp. 261-272, February 2009.

[18] J. Gu, X. Yang, S. D. Mello and J. Kautz, "Dynamic Facial Analysis: From Bayesian Filtering to Recurrent Neural Network," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 1531-1540.

[19] T. Y. Yang, Y. T. Chen, Y. Y. Lin, and Y. Y. Chuang. "Fsa-net: Learning fine-grained structure aggregation for head pose estimation from a single image," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2019, pp. 1087-1096.

[20] Y. Hu, L. Chen, Y. Zhou and H. Zhang, "Estimating face pose by facial asymmetry and geometry," in *Proc. 6th IEEE Int. Conf. Autom. Face Gesture Recognit.*, 2004, pp. 651-656.

[21] P. Martins and J. Batista, "Accurate single view model-based head pose estimation," in *Proc. 8th IEEE Int. Conf. Autom. Face Gesture Recognit.*, 2008, pp. 1-6.

[22] R. Horaud, F. Dornaika and B. Lamiroy, "Object Pose: The Link between Weak Perspective, Paraperspective, and Full Perspective," *Int. J. Comput. Vision*, vol. 22, no. 2, pp. 173–189, March 1997.

[23] T. S. Huang, A. M. Bruckstein, R. J. Holt and A. N. Netravali, "Uniqueness of 3D pose under weak perspective: a geometrical proof," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 12, pp. 1220-1221, December 1995.

[24] P. Dollár, P. Welinder and P. Perona, "Cascaded pose regression," in *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognit.*, 2010, pp. 1078-1085.

[25] D. Lee, H. Park and C. D. Yoo, "Face alignment using cascade Gaussian process regression trees," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2015, pp. 4204-4212.

[26] X. Xiong and F. De la Torre, "Supervised Descent Method and Its Applications to Face Alignment," in *Proc. IEEE Conf. Comput. Vision Pattern* Recognit., 2013, pp. 532-539.

[27] J. Deng, J. Guo, E. Ververas, I. Kotsia and S. Zafeiriou, "RetinaFace: Single-Shot Multi-Level Face Localisation in the Wild," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2020, pp. 5202-5211.

[28] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2014, pp. 1867-1874.

[29] H. Zhang, Q. Li, Z. Sun et al., "Joint voxel and coordinate regression for accurate 3d facial landmark localization," in *Proc. IEEE Int. Conf. Pattern Recognit.*, 2018, pp. 2202-2208.

[30] P. Barra, S. Barra, C. Bisogni, M. De Marsico and M. Nappi, "Web-shaped model for head pose estimation: An approach for best exemplar selection," *IEEE Trans. Image Processing*, 2020, pp.5457-5468.

[31] G. Zhang, J. Liu, H. Li, Y.Q. Chen and L.S. Davis, "Joint human detection and head pose estimation via multistream networks for RGB-D videos," *IEEE Signal Processing Letters*, vol. 24, no. 11, pp. 1666-1670, July 2017.

[32] X. Yu, J. Huang, S. Zhang and D. N. Metaxas, "Face Landmark Fitting via Optimized Part Mixtures and Cascaded Deformable Model," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 11, pp. 2212-2226, November 2016.

[33] R. Ranjan, S. Sankaranarayanan, C. D. Castillo and R. Chellappa, "An All-In-One Convolutional Neural Network for Face Analysis," in *Proc. 12th IEEE Int. Conf. Autom. Face Gesture Recognit.*, 2017, pp. 17-24.

[34] M. Patacchiola and A. Cangelosi, "Head pose estimation in the wild using convolutional neural networks and adaptive gradient methods," *Pattern Recognit.*, 2017, pp.132-143.

[35] A. Zingoni, M. Diani and G. Corsini, "Tutorial: Dealing with rotation matrices and translation vectors in image-based applications: A tutorial," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 34, no. 2, pp. 38-53. February 2019.

[36] A. G. Howard, M. Zhu, B. Chen, et al. "Mobilenets: Efficient convolutional neural networks for mobile vision applications," arXiv preprint arXiv:1704.04861, 2017.

[37] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 1800-1807.

[38] M. Sandler, A. Howard, M. Zhu, et al., "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2018, pp. 4510-4520.

[39] A. Howard, M. Sandler, G, Chu, et al., "Searching for MobileNetV3," in *Proc. IEEE Int. Conf. Comput. Vision*, 2019, pp. 1314-1324.

[40] P. Chen, B. Liao, G. Chen, et al., "Understanding and utilizing deep neural networks trained with noisy labels," arXiv preprint arXiv:1905.05040, 2019.

[41] C. G. Northcutt, L. Jiang and I. L. Chuang, "Confident learning: Estimating uncertainty in dataset labels," arXiv preprint arXiv:1911.00068, 2019.

[42] G. Fanelli, M. Dantone, J. Gall, A. Fossati, and L. V. Gool, "Random forests for real time 3d face analysis," *Int. J. Comput. Vision*, vol. 101, no. 3, pp. 437-458, February 2013.

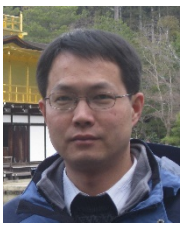
[43] X. Zhu, Z. Lei, X. Liu, H. Shi and S. Z. Li, "Face Alignment Across Large Poses: A 3D Solution," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2016, pp. 146-155.

[44] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman and N. Kumar, "Localizing Parts of Faces Using a Consensus of Exemplars," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 12, pp. 2930-2940, December 2013.

[45] X. Zhu and D. Ramanan, "Face detection, pose estimation, and landmark localization in the wild," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2012, pp. 2879-2886.

[46] E. Zhou, H. Fan, Z. Cao, Y. Jiang and Q. Yin, "Extensive Facial Landmark Localization with Coarse-to-Fine Convolutional Network Cascade," in *Proc. IEEE Int. Conf. Comput. Vision Workshop*, 2013, pp. 386-391.
[47] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou and M. Pantic, "300 Faces in-the-Wild Challenge: The First Facial Landmark Localization Challenge," in *Proc. IEEE Int. Conf. Comput. Vision Workshop*, 2013, pp. 397-403.
[48] K. Messer, J. Matas, J. Kittler, J. Luettin, and G. Maitre, "XM2VTSDB: The extended M2VTS database," in *Proc. 2nd Int. Conf. Audio Video-Based Biometric Person Authentication*, 1999, vol. 964, pp. 965-966.
[49] M. Köstinger, P. Wohlhart, P. M. Roth, et al., "Annotated Facial Landmarks in the Wild: A large-scale, real-world database for facial landmark localization," in *Proc. IEEE Int. Conf. Comput. Vision Workshops*, 2011, pp. 2144-2151.
[50] J. Li, Y. Wang, and C. Wang, "DSFD: dual shot face detector," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2019, pp. 5060-5069.
[51] D. P. Kingma, and J. Ba. "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.
[52] S. S. Mukherjee and N. M. Robertson, "Deep Head Pose: Gaze-Direction Estimation in Multimodal Video," *IEEE Trans. Multimedia*, vol. 17, no. 11, pp. 2094-2107, November 2015.
[53] T. Yang, Y. Chen, Y. Lin, et al, "FSA-Net: Learning Fine-Grained Structure Aggregation for Head Pose Estimation From a Single Image," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2019, pp. 1087-1096.
[54] H. Hsu, T. Wu, S. Wan, et al., "QuatNet: Quaternion-Based Head Pose Estimation With Multiregression Loss," *IEEE Trans. Multimedia*, vol. 21, no. 4, pp. 1035-1046, April 2019.
[55] V. Albiero, X. Chen, X. Yin, et al., "img2pose: Face Alignment and Detection via 6DoF, Face Pose Estimation," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2021, pp. 7617-7627.

Xiao Li received his B.S. degree and M.S. degree from Sun Yat-sen University, China, in 2018 and 2020 respectively. His research interests include head pose estimation, eye-tracking and computer vision.



Dong Zhang received his B.S.E.E. and M. S. degrees from Nanjing University, China, in 1999 and 2003, respectively, and Ph.D. degree from Sun Yat-sen University, China, in 2009. He is currently an associate professor in the school of Electronics and Information Technology, Sun Yat-sen University. His research interests include image processing, pattern recognition, and information hiding.

Ming Li received his Ph.D. in Electrical Engineering from University of Southern California in May 2013. He is currently an associate professor of Electrical and Computer Engineering at Duke Kunshan University, a research scholar at the ECE department of Duke University, and the adjunct professor at Wuhan University. His research interests are in the areas of speech processing and multimodal behavior signal analysis with applications to human centered behavioral informatics notably in health, education and security.

Dah-Jye Lee received his B.S. degree from National Taiwan University of Science and Technology in 1984, M.S. and Ph.D. degrees in electrical engineering from Texas Tech University in 1987 and 1990, respectively. He also received his MBA degree from Shenandoah University, Winchester, Virginia in 1999. He worked in the machine vision industry for eleven years prior to joining BYU in 2001. He is currently a Professor in the Department of Electrical and Computer Engineering at Brigham Young University. His research work focuses on artificial intelligence, robotic vision, high-performance visual computing, and visual inspection automation.