IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

MR-CapsNet：A Deep Learning Algorithm for Image-Based Head Pose Estimation on CapsNet

HAO FANG [1, 2, 3,+], JUN-QING LIU [4+], KAI XIE [1, 2, 3,*], PENG WU [1, 2], XIN-YU ZHANG[1,2,3], CHANG WEN [3, 4], and JIAN-BIAO HE [5]

[1] School of Electronic Information, Yangtze University, Jingzhou, 434023 China
[2] National Demonstration Center for Experimental Electrical & Electronic Education, Yangtze University, Jingzhou, 434023 China
[3] West Institute, Yangtze University, Kelamayi, 834000 China
[4] School of Computer Science, Yangtze University, Jingzhou, 434023 China
[5] School of Computer Science and engineering, Central South University, Changsha 410083, China;

*Corresponding author: KAI XIE (e-mail: pami2009@163.com)

This work was supported in part by the Natural Science Foundation of Xinjiang Uygur Autonomous Region (2020D01A131), the Fund of Hubei Ministry of Education (B2019039), the Graduate Teaching and Research Fund of Yangtze University (YJY201909), the Teaching and Research Fund of Yangtze University (JY2019011), the Undergraduate Training Programs for Innovation and Entrepreneurship of Yangtze University under Grant 2019099, the Undergraduate Training Programs for Innovation and Entrepreneurship of Yangtze University under Grant Yz2020057, and the National College Student Innovation and Entrepreneurship Training Program (202110489003).

ABSTRACT Head pose estimation based on a single image is a challenging endeavor because of the complex background conditions and characteristics of the human face. In this report, we propose a Multi stage Regression-Capsule Network (MR-CapsNet) to predict head posture based on a single image input. In the study, we used the residual attention block and squeeze-and-excitation block to extract features in three levels. CapsNet overcomes the shortcomings of the traditional convolutional neural network and implements module aggregation to describe the spatial relationship of features after aggregation, in addition to realizing a compact and robust model using a multi-stage regression scheme. We tested our method on the AFLW2000 and BIWI datasets obtaining mean absolute errors of 4.26% and 3.95%, respectively. In addition, we discuss the accuracy of our method in the case of eye or mouth occlusion. The results of comprehensive experiments reveal that our method can accurately predict head posture.

INDEX TERMS Head pose estimation, Multi-stage Regression, Squeeze-and-excitation Block, Capsule Network

I. INTRODUCTION

The development of a variety of perceptual devices has served as the basis for recent advancements in personalized entertainment. Head pose estimation is an essential part of human-computer interaction, which can provide information on the direction of human attention. The prediction of head pose based on a single image is still a challenging task. The head pose can be represented by a three-dimensional vector that includes the top view, roll, and yaw angles [1]. To extract head pose information from images, it is necessary to determine the feature mapping between two- and three-dimensional space. The head pose estimation task involves inferring the head pose direction based on images acquired using a camera. In a driving system, it is possible to ascertain the driver's attention and consciousness based on position information [2]. Head pose information is also important for human-computer interaction [3]. The system can interact with the user's head monitoring software [4][5] to estimate the level of interest.

The process of head pose estimation and related tasks is often associated with many challenges, such as imaging problems due to the camera system, complex backgrounds, blurred targets caused by different light sources, and target occlusion problems [6]. In real space, the use of human vision to obtain information often results in significant challenges. Human vision is often too fuzzy to perceive distant objects, and in the case of dim or poor lighting, head pose estimation may result in failure. Therefore, in the field of computer vision, the face alignment method is used in many face detection algorithms [7][8], which places the target information in the same semantic domain as that of the simplified object being detected. In previous studies, the influence of the background on the target was effectively eliminated using facial localization and image clipping, and the impact of noise caused by an image-independent target was reduced [9].

The traditional workflow is based on deep learning, especially convolutional neural networks (CNNs) [10]. These traditional networks have a wide range of learning capabilities, but they also have some key shortcomings. For example, the lack of local equivariant features leads to weak generalization ability necessitating additional parameters for the construction of a deep network whereby the location relationship between local and global features is no longer well-maintained [11], and the robustness is not high. To overcome the shortcomings of CNNs, CapsNet was recently proposed by Sabor [12]. Each capsule in CapsNet is a group of neurons that can represent different instantiation parameters related to different targets and their probability of existence. There has been significant interest in the use of CapsNet in different application fields and the development of different variants. CapsNet has a particularly important feature,a unique "routing" process can effectively handle the transformation model. Only when the son-capsule is consistent with the predicted value, it can be transformed into the parent capsule. Recently, a technique was incorporated into CapsNet to enhance its robustness to transformation. CapsNet is highly sensitive to the image background, which contributes the accuracy of head pose estimation and classification as the detailed information on the position and pose of the object has to be retained, which in turn is useful in learning relations, determining the exact position of the extracted features, and establishing the representation of the object in terms of partial hierarchical structures [13].

The classic head pose estimation methods include machine learning [14][15] as well as appearance template [16][17], geometric model-based [18][19], depth image-based [20], and landmark-based methods. To estimate the head pose from an image, it is necessary to perform a mapping from two- to three-dimensional space. Compared to traditional RGB images, depth images can retrieve missing 3D information from 2D images and provide additional information to estimate head posture. At present, the depth camera has not been popularized and can only be used in certain fixed places. Moreover, the required computational burden and memory is too large for small servers. In the landmark-based method, Adrian [21] proposed converts 2D landmark annotations into 3D, to reasonably enhance and summarize the existing data set. In the course of studying the various aspects of face alignment with respect to different factors, training of the neural network model achieved excellent accuracy. Other methods include the component-based discrimination method proposed by Lin [22], which uses a discriminative search algorithm to identify the shape of the face in the component. The classifier can detect the facial components in the configuration of the face component to effectively improve the accuracy and efficiency of face detection. These methods first recognize the road signs of the face, and then use them to predict the head pose. In the model-based method, Martins [23] proposed a framework to automatically estimate the pose of the human head in a single-view image. This method uses a 3D rigid model of the human body as an approximation of the human head, combined with an active appearance model. With respect to facial feature extraction and tracking, Krinidis [24] proposed a method to estimate head pose in a single-view video sequence. First, a face detector is used to detect the face; then a deformable surface model approximates the tracking technology of facial image strength; and finally, a feature vector is used to realize the head pose. Estimation methods use key points of the face to construct three-dimensional head models, and then obtain the result by training the appearance model. FSA-Net [25] uses a hierarchical coarse-to-fine classification strategy, then a soft phase stepwise regression scheme to extract intermediate features followed by aggregation and regression to predict the final head pose. Based on the deep learning method [26], a convolutional neural network (CNN)-based model is constructed using CNN to estimate the pose of the human head in low-resolution multi-modal RGB-D data. Kumar et al. [27] proposed a method to correlate the trajectory of key points with the trajectory of the head posture, which changes the prediction results in accordance with the transformation of landmarks. Yang [28] proposed an advanced capsule network of RS-CapsNet, which improves the capsule network on the basis of the original network architecture and addresses the shortcomings of the capsule network pertaining to weak feature extraction ability and multiple training parameters, to achieve good performance in image classification. Xia et al. [29] proposed a face marker-assisted pose estimation method. In their work, they combined landmark-based face images with channel-level grayscale images for head pose prediction [30]. Ranjan et al. [31] regularized the shared parameters of the CNN, and a synergy effect was established between different fields and tasks such as smile detection, age estimation, and face recognition. Gu et al. [32] proposed a face feature tracking algorithm based on an RNN. Hyperface [33] uses a CNN to learn common features in the middle layer, which are then inputted into the multitask learning

network for face detection, head pose estimation, and facial gender information. FacePoseNet [34] uses a CNN to perform 3D head pose regression, based on camera positioning, as auxiliary information for target recognition to improve precision. HopeNet [8] calculates the yaw, pitch, and roll angles by combining Resnet50 [35] and multiple loss functions. Zhao used multi-feature fusion to obtain head pose estimation. Wu used hog and pyramid settings to describe local gradient features and global shape features of the image of the face to facilitate head pose estimation in the local occlusion state [36]. Abate [37] proposed the Web-shaped Model algorithm to encode the posture of the face, and then regression for further face posture prediction. This method improves the sensitivity of head posture estimation and prediction accuracy. Recent studies have shown that multitask learning [38] can achieve better results compared to a single task.

Hence, the main contributions of this paper are:

1. We proposed a head pose prediction model based on multi-stage Regression-CapsNet (MR-CapsNet). We built a detection model based on feature extraction, feature aggregation, and multi-stage regression. The model can obtain multi-stage feature information. The probability vector of different stage features are then dynamically combined to predict and improve the accuracy of head pose estimation.

2. We created an accurate feature extraction network, which uses an efficient attentional mechanism model to combine the residual attentional block [38] and squeeze-excitation (SE) block [39]. The network does not only enhance the feature information extraction ability of the network, but also highlights useful features while suppressing useless ones. This structure can better explain the spatial relationship of target features and more accurately estimates the head posture.

3. We first applied the capsule neural network to the head pose estimation task. We applied the capsule structure of the network during the feature aggregation stage of head pose estimation, then constructed intermediate capsules using the "vertical and horizontal sliding method Windows" to select feature information, and finally used the linear combination method between capsules to enhance the representative ability of capsules. Compared with traditional CNN, our method can better discern the spatial relationship of features and improve the prediction accuracy of partially occluded faces.

The structure of this paper is as follows: the second section introduces the theoretical basis of the model in our algorithm, the third section provides the training details and experimental results, and the fourth section presents the conclusion.

II. METHOD

The flowchart in Figure 1 illustrates our head pose estimation algorithm based on the capsule neural network and multi-stage regression. The algorithm can be divided into three parts: (1) The feature extraction network performs the main feature extraction; (2) Capsule network performs feature aggregation on feature information; (3) Multi-stage regression obtains the probability vector of each stage.

First, we preprocess the input image to detect the head region. Then we output the detected image as input into the feature extraction network. In this network, we divide the feature extraction into three stages. Each stage is processed by a residual attention block and an SE block to improve feature processing, to strengthen feature weights of key information, and to enhance facial feature extraction capabilities. There is continuity between the stages to ensure that the effect of feature extraction is enhanced layer by layer. Then, the feature maps obtained in these three stages are inputted into the feature aggregation network. We constructed the intermediate capsule through feature selection so that our capsule neural network would be more sensitive to spatial information. The capsule neural network linearly combines the information graphs, and passes them through a dynamic routing algorithm to obtain richer feature information, which enhances the network's ability to understand the extracted facial features and reduces the impact of missing facial feature information on the prediction results. Finally, we combine the feature maps of the three stages to perform multi-stage regression to obtain the required probability vectors to improve our prediction accuracy.

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/ACCESS.2021.3119615, IEEE Access

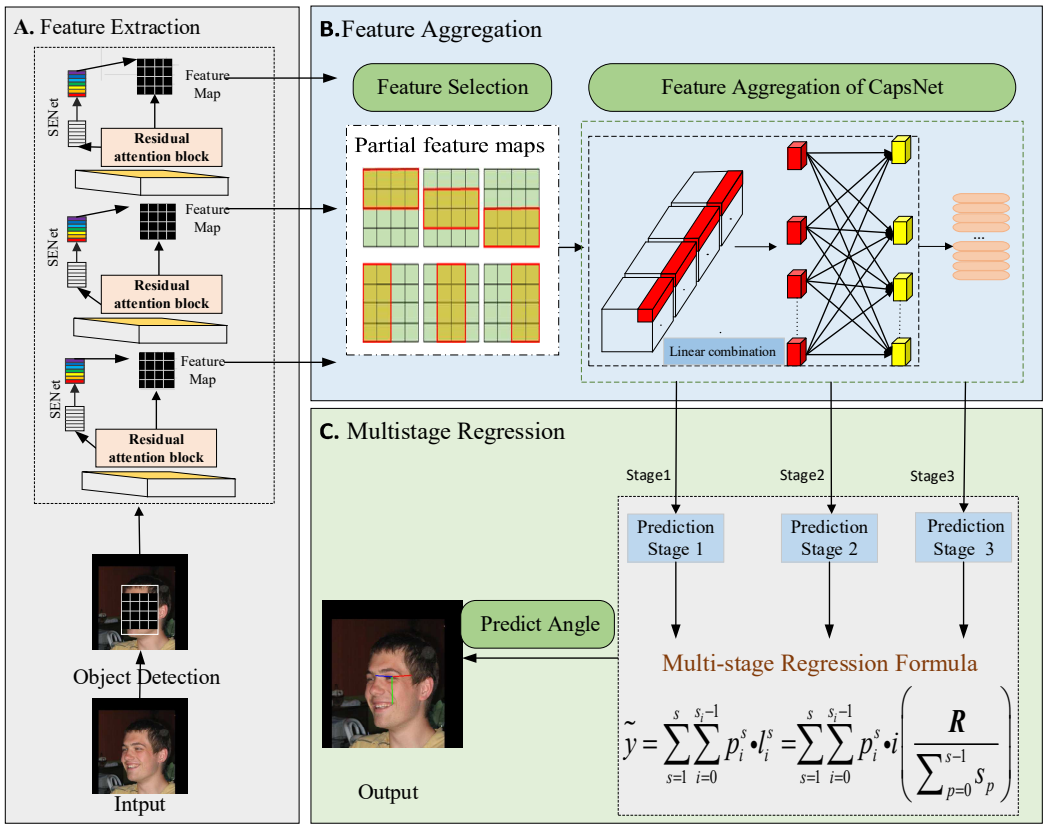


FIGURE 1. **Flow of proposed algorithm**

A. FEATURE EXTRACTION

Our network is based on the network proposed by Song et al. [38], which is a compact model for age estimation from a single image. Our feature extraction network has three branches. Each branch consists of convolution, weight normalization, activation, three basic residual blocks, a pooling layer, and SE blocks. In addition, residual attention blocks are embedded into each stage. The structure of the residual attention block is also composed of convolution, weight normalization, channel, spatial attention, and a fusion layer, similar to the structure depicted in Fig. 1. Different filter cores and down-sampling methods are used for the residual unit. The feature maps with different kernel sizes are combined by multiplying the elements of the two feature maps generated by channel attention. Then, the features maps are inputted into the aggregation space, which focuses on the process of constructing the head rotation. This is illustrated in Fig. 2.

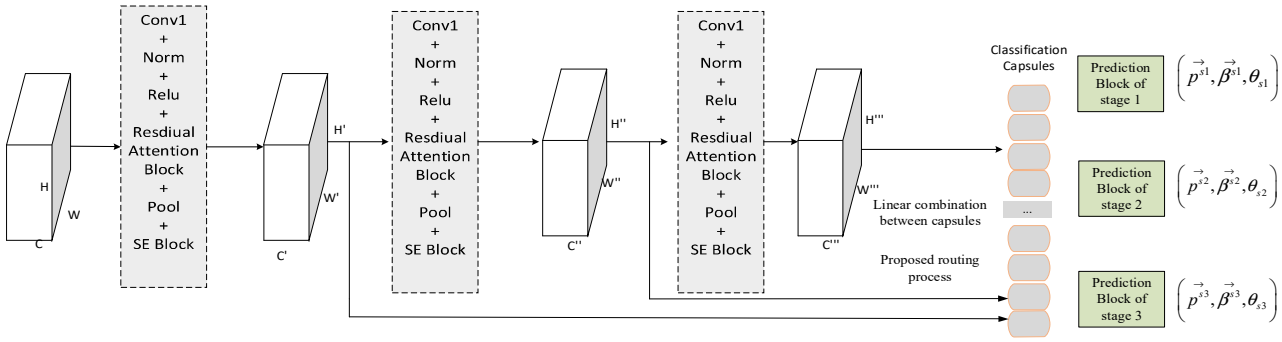


FIGURE 2. **Flow of the feature extraction algorithm**

1) PROBLEM FORMULATION

Recently, head pose estimation based on a single facial image was studied with respect to convolutional networks. Usually, a set of trained facial images $\{(x_1,y_1),(x_2,y_2),\ldots,(x_n,y_n)\}$ is used, where x_i indicates the i-th facial image and y_i indicates the x_i head pose 3D

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/ACCESS.2021.3119615, IEEE Access

vector. The head pose vector can be subdivided into three angles: yaw, pitch angle, and roll. The goal is to learn the mapping function F, and obtain $\tilde{y} = F(x)$ to predict the head pose angle of the input image,

$$\tilde{y} = \vec{p} \bullet \vec{l} = \sum_{i=0}^{R} p_i \bullet l_i \qquad (1)$$

where R indicates the range of the head pose angle, and p_i indicates the probability of the 3D vector l_i. In addition, to ensure the accuracy of the algorithm, we use the Mean Absolute Error (MAE) as the evaluation standard to reduce the error between the predicted head pose angle and the ground truth label,

$$J(x) = \frac{1}{N} \sum_{n=1}^{N} \left\| \tilde{y}_n - y_n \right\| \qquad (2)$$

where $\tilde{y}_n = F(x_n)$ is the predicted pose for the training image x_n.

2) FACE DETECTION

In the unconstrained case, the human head may have a large angle conversion and low resolution in a remote image; therefore, a relatively stable head detector is required. We chose MTCNN [40] as our detector, which can achieve real-time head detection at different scales and angles for a complex background. MTCNN combines face region detection with face key point detection, and its framework is similar to a cascade. It can be divided into three layers: P-net, R-Net, and O-net, which yields a robust detection.

3) RESIDUAL ATTENTION BLOCK

The residual attention block is a type of attention unit that promotes facial feature extraction via transform-ations as follows:

$$F : X \to \widetilde{X}, X \in R^{H \times W \times C}, \widetilde{X} \in R^{H' \times W' \times C'} \qquad (3)$$

$F(\bullet)$ can be regarded as a standard convolution operation along the channel and spatial dimensions. For the channel, we used the multi-scale kernel and pooling operations to map features to obtain distinguishable vectors, and then the results were fused by channel multiplication. The calculation of the space dimension is the same as that of channel size. Figure 3 depicts a more detailed description of the architecture of the residual attention block.

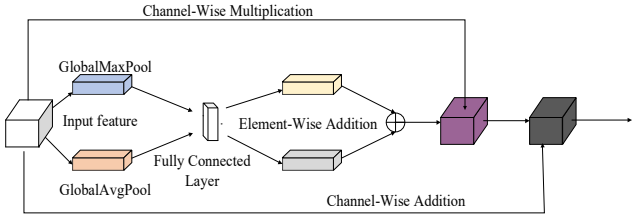


FIGURE 3. Architecture of the residual attention block

4) SQUEEZE-AND-EXCITATION(SE) BLOCK

The SE block is a type of attention block based on a feature graph channel. The core idea of an SE block is to learn the feature weights according to the loss, increase in the weights of the effective feature map, and to be able to reduce the weights of invalid or small feature maps to achieve better results. It has been demonstrated that SE blocks can improve the performance of a network with minimal computational cost. The architecture of the SE block is shown in Fig. 4.

SE blocks map any given input graph into the network module.

$$F_{tr} : X \to \mathrm{U}, \mathrm{X} \in R^{H' \times W' \times C'}, \mathrm{U} \in R^{H \times W \times C} \qquad (4)$$

Here, X is the input graph and U is the extracted feature. To establish the dependence between channels, we need to squeeze the feature u, and aggregate the feature graph to obtain a graph with dimensions $W \times H$, which is used as the feature descriptor. Then, $F_{sq}(\bullet)$, the spatial information of the global receptive field of each feature graph, is placed into the feature graph, which is referred to as the a descriptor. The network layer can then obtain information of the global receptive field, based on the descriptor feature map. To address the problem of exploiting the correlation between channels, SE blocks use the $F_{sq}(\bullet)$ squeeze operation to build correlations between the channels.

$$z_c = F_{sq}(\mathrm{u}_c) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} u_c(\mathrm{i}, \mathrm{j}) \qquad (5)$$

where the subscript c represents the channel, u_c represents the two-dimensional matrix with channel C in U.

Next, the aggregate information obtained from the compression operation is used to fully capture the dependency on channel dimensions. To achieve this goal, we use the following:

$$F_{ex}(\bullet, \mathrm{w}) = \sigma(g(z, W)) = \sigma(W_2 \bullet \sigma \bullet (W_1 \bullet z)) \qquad (6)$$

where σ is the Relu function, W_1, W_2 are the two fully connected layers. The second fully connected layer is followed by the sigmoid function. After these operations are completed, the weights of the feature map are obtained, and these weights are fused with the original view features:

$$F_{scale}(u_c, s_c) = u_c s_c \qquad (7)$$

where $F_{scale}(u_c, s_c)$ denotes s_c and the feature map $u_c \in R^{H \times W}$ scaling index is multiplied by. The function of the two full connection layers is to fuse the feature map information of each channel. After the exception operation, a set of channel weights S' is generated, which represents the weight of the feature maps between the channels. The

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/ACCESS.2021.3119615, IEEE Access

enhanced feature map can then be obtained by multiplying S' and the input feature map.

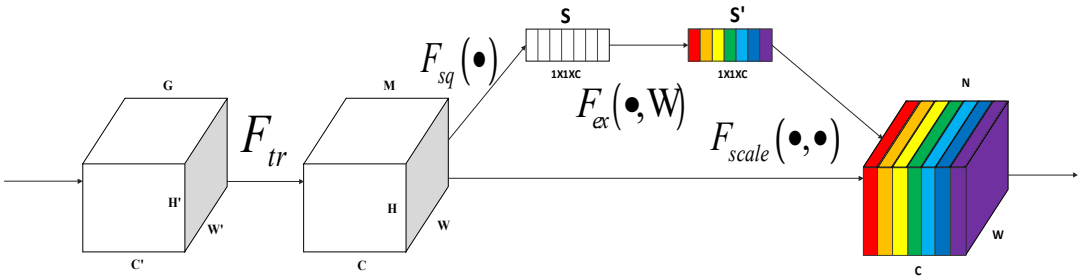


FIGURE 4. Architecture of the SE block

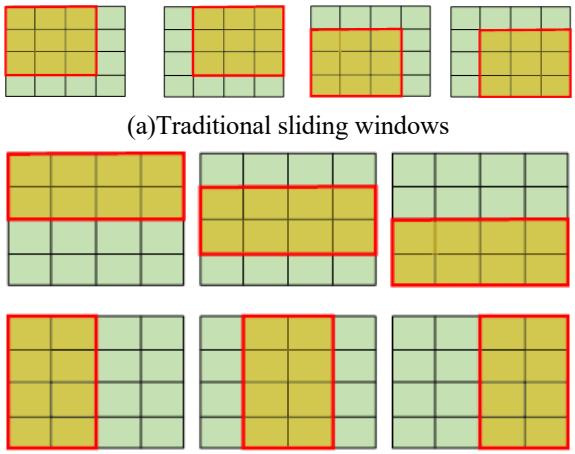
B. FEATURE AGGREGATION

The role of the aggregation module is to aggregate a small number of representative features of the calculated feature maps into local maps. For the aggregation module, we consider CNN and CapsNet. We determined that CNN is ideal for capturing the existence of features because its convolution structure is designed for this purpose. However, when exploring the relationship between feature attributes, CNN is not optimal, causing the input image to lose the exact target information of the feature detector. As such, CNN does not successfully identify the object in case of rotation or other similar situations. In head pose estimation, the human head often has a large rotation angle, and a method based on CapsNet is proposed to overcome the limitations of the CNN method.

The CapsNet in this work was inspired by the RS-CapsNet architecture, which is designed for feature fusion. Therefore, we use CapsNet as our aggregation module for the features. In addition, to reduce the amount of calculation and capsules, we use a 1×1 convolution layer to reduce the number of channels. We remodel all the realized feature maps into capsules, using the linear relationship between the capsules to fuse features and halve the capsule to enhance its ability to express features. We obtain different types of capsules for different local feature maps, implement a dynamic routing algorithm for them, and construct capsules that can represent most of the objects. Each local feature map can construct N_3 capsule networks, where each capsule is D_2 Finally, the intermediate capsule constructed using the local feature map and the original capsule obtained by feature mapping are used to obtain the classified capsule.

1) FEATURE SELECTION

We first divide the feature map generated by the last convolution operation of the input image into small local feature maps, which are then used to construct the "intermediate capsule." This capsule can represent most of the detected objects. The intermediate capsule and the original capsule obtained by feature mapping are used to obtain the classified capsule. Regarding the problem of "how to slice the feature map," we recommend using vertical and horizontal sliding windows, as illustrated in Fig. 5. There are two reasons for selecting the vertical and horizontal sliding window methods. First, for objects with horizontal, vertical, or other symmetrical structures, the "vertical and horizontal sliding window" method is more conducive to maintaining their integrity; second, we expect to use the maximum number of small local feature maps. Compared with the traditional sliding window method shown in Fig. 5, the " Improved sliding window" method allows for more local feature maps.



(a)Traditional sliding windows

(b) Improved sliding windows

FIGURE 5. (a) : traditional sliding windows method, (b) : the improved sliding windows is a vertical and horizontal sliding method

2) LINEAR COMBINATION BETWEEN CAPSULES

To address the problem of the presence of redundant information in the background of the input image, we use a linear relationship in the capsule, and remodel the feature map into capsules such that each capsule represents the detection object in the input image. We then construct a connection between the capsules in the same position, and finally use the linear relationship of the capsule in the input image. The aforementioned linear combination method is utilized to flatten the capsules, maintain their length in [0,1], cause their direction to be constant, and provide a more nonlinear relationship for the entire network. Fig. 6 shows the linear combination method between capsules with the same pixel location.

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/ACCESS.2021.3119615, IEEE Access

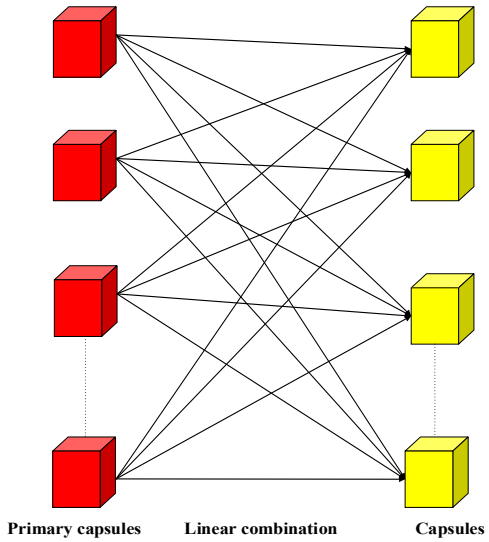


FIGURE 6. **Linear combination between capsules.**

3) DYNAMIC ROUTING ALGORITHM

In the capsule network, the length of the capsule represents the probability that the target is correctly detected.

Dynamic routing based on EM uses the maximum likelihood estimator and clustering technology to group capsules into a part–whole relationship.

The coupling system of higher capsules is calculated by estimating their activation degree and their probability values. In network regularization training, routing is not combined with image reconstruction. By converting convolution and routing to a specific computing domain, the number of parameters can be significantly reduced to achieve better results.

Based on the results of the comparison, dynamic routing based EM is more suitable when the image size changes.

The dynamic routing algorithm works as follows:

Algorithm 1 Dynamic Routing

Routing $\left(\hat{u}_{j|i,r,l}\right)$

for all capsule i in layer l and j in layer $l+1$: $b_{ij} \leftarrow 0$

for r iterations do

for all capsule i in layer l: $c_i \leftarrow softmax(b_i)$

for all capsule j in layer $l+1$: $s_j \leftarrow \sum_i c_{ij}\hat{u}_{j|i}$

for all capsule j in layer $l+1$: $v_j = squash(s_j)$

for all capsule i in layer l and j in layer $l+1$: $b_{ij} \leftarrow b_{ij} + \hat{u}_{j|i} \cdot v_j$

return v_j

C. MULTISTAGE REGRESSION

In traditional age estimation, to improve the accuracy and simplicity of age classification, usually one year is used as the interval. However, given the large number of network parameters and the need for a large amount of computing resources, the training network becomes both complex and time-consuming. To address this shortcoming and maintain the accuracy of age prediction, the scale of the deep neural network is reduced to produce a more compact and effective network, which can transform a regression into a multi-stage process.

As shown in Fig.7, the structure of the multi-stage regression module is as follows: the main branch is composed of 1×1 convolution; ReLU activation function, pooling layer, and three function quantities are output through three branches, respectively. The first branch outputs θ directly through the full connection layer and the tanh activation function. The second branch outputs $\vec{p}$ through the dropout layer, the full connection layer, the tanh activation function, the full connection layer and the softmax function. The third branch outputs $\vec{\beta}$ via the dropout layer, the full connection layer, tanh activation function, the full connection layer and tanh activation function.

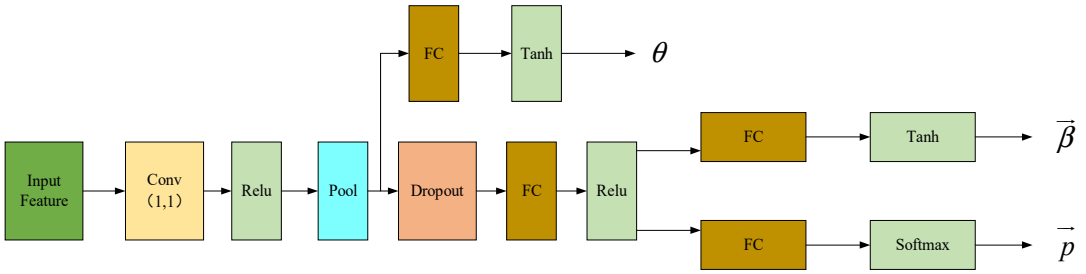


FIGURE 7. **Stage prediction module**

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/ACCESS.2021.3119615, IEEE Access

The model predicts Head Pose based on multistage regression:

$$\tilde{y}=\sum_{s=1}^{s}\sum_{i=0}^{s_i-1}p_i^s\bullet l_i^s=\sum_{s=1}^{s}\sum_{i=0}^{s_i-1}p_i^s\bullet i\left(\frac{\boldsymbol{R}}{\sum_{p=0}^{s-1}s_p}\right) \quad (8)$$

where $\vec{p}=(p_0,p_1,\cdots,p_{\boldsymbol{R}})$ is the index distribution of each probability, which is taken from the top level of the model, $\vec{l}$ is the representative index of each probability, and s_i represents the width of stage I before adjustment. To determine the error of the result; θ_i is the factor that determines the degree of change of the stage width; $\vec{\beta}^s$ is the offset vector; and l_i^s is the size. Given an input image, the parameters of each stage, $\left(\vec{p}^{s_i},\vec{\beta}^{s_i},\theta_{s_i}\right)$ are outputted. Finally, we print our predicted value $\tilde{y}$.

The multistage regression formula can be applied to any regression problem. In this study, we apply multistage regression to head pose estimation. Unlike the age estimation problem, the pose estimation problem obtains vectors instead of a scalar.

III. EXPERIMENTAL RESULTS AND DISCUSSION

In this section, we describe our experimental process in detail as shown in Fig.8. The experiment was divided into four parts. In the first part, we introduced the evaluation criteria for the experiment. In the second part, we described some basic experimental settings. In the third part, we provide the details of the experimental training. Finally, we present the results of head posture prediction using different assessment schemes.

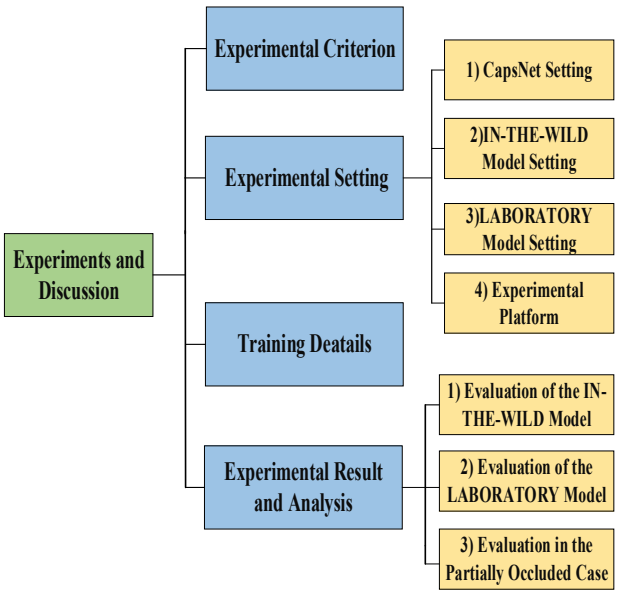


FIGURE 8. Experimental framework

A. EXPERIMENTAL CRITERION

In the following experiments, we evaluate the experimental results using MAE:

$$MAE=\frac{1}{N}\sum_{I=1}^{N}\left\|\hat{l}_i-l_i\right\| \quad (9)$$

where $\hat{l}_i$ and l_i are the ground truth label and the final predicted value of the yaw, pitch, roll angles of the i-th image, respectively, and N is the total number of images of the test set clock.

B. EXPERIMENTEL SETTING

1) CAPSNET SETTING

We consider that n is the number of intermediate capsules generated by each feature map N_3. This value cannot be too large because the capsule is different from that of a CNN as it is generated by the routing process, represents the characteristics of the target object rather well, and does not contain too much superfluous information. Moreover, the value cannot be too small because we obtain the classification capsule based on the weight of the sub capsule. If it is too small, it cannot achieve a good effect. Therefore, we set N_3 to 16.

Given that the original capsule D_2 is constructed directly from the feature map, there is significant surplus information. Thus, the routing intermediate capsule better represents the target. Therefore, we changed the size of the original capsule D_2 to 8.

2) IN-THE-WILD MODEL SETTING

There are three head pose estimation datasets used in the experiment, consisting of 300W-LP [30], AFLW2000 [30] and BIWI [41].

We trained the IN-THE-WILD model using the 300W-LP dataset. 300W-LP is a simulation dataset based on the 300W dataset and the 3DMM dataset. A 3D model is constructed using a 2D image to simulate head pose estimation. The model is then gradually flipped to further enhance the effect of the dataset. The dataset contains large-angle images, and 122450 flipped images are expanded on this basis. It is a good dataset for training head pose estimation models.

AFLW2000 is a challenging dataset, which consists of a large-scale face database with multiple poses and multiple angles It provides real 3D facial pose angle landmarks for the first 2000 images of the AFLW dataset, including pose changes of different characters under different scenes and luminosity. We use the AFLW2000 dataset to test the model, which can verify the generalization ability of the model.

3) LABORATORY MODEL SETTING

We trained the laboratory model using the BIWI dataset. The BIWI dataset was created using Kinect sensors. It consists of 24 sequences with a total number of 15.6 K

frames, and includes 1000 high-quality 3D face pose data samples captured using RGBD cameras, including 24 RGBD cameras capturing 20 different people and 24 videos of 20 different characters, head pose range including approximately ±75° yaw and ±60° pitch. The dataset consists of about 15,000 images, including not only RGB images, but also depth images and annotations. Unlike the other two datasets, which were collected from the field, all BIWI images were taken indoors, it can verify the detection ability of the model in the indoor environment.

4) EXPERIMENTAL PLATFORM

In this work, all experiments were conducted on a platform with a Windows 10 operating system, an NIVIDIA GeForce RTX 2060 with 8 GB graphics memory, and an Intel Core i7-4790K with 16 GB memory. The software platform is Python 3.7.3, based on the Keras and Tensorflow1 framework.

C. TRAINING DETAILS

We used Adam [42] as the optimizer for training, and the initial learning rate was set to 0.001. The learning rate was decreased by 0.1 times every 30 periods. To enhance the ability to process blurred and zoomed images, random clipping and random scaling were applied to the training images to augment the training data. The 3D rotation of the Z-axis in the X-Y-Z axis was consistent with the 2D rotation, thus the rotation of the head along the X-Y-Z axis was fixed. Therefore, to establish a better relationship between image and head posture, we converted the Euler angle of Z-Y-X to X-Y-Z to reduce the average prediction error.

For the IN-THE-WILD model, training was performed on the 300W-LP dataset, whereas the AFLW2000 and BIWI datasets were used for testing. When using the BIWI dataset for evaluation, we only considered images with rotation angles in [–99° and 99°]. The batch size for the training and testing sets was 16.

For the laboratory model, 70% of the training was performed on the BIWI dataset, and the rest was used for testing. The training and test batch sizes were set to 8.

D. EXPERIMENTAL RESULT AND ANALYSIS

1) EVALUATION OF THE IN-THE-WILD MODEL

The IN-THE-WILD model was trained using the 300W-LP dataset. Tables 1 and 2 summarize our methods, for which the AFLW2000 and BIWI datasets were used for comparison with the latest method, using MAE as the evaluation standard. Our method achieved excellent results compared to other advanced approaches. HopeNet [8] uses Resnet50 to separate yaw, roll, and pitch, and uses MAE and cross-entropy to estimate the fine-grained head posture. FSA-Net [25] uses the SSR net collective attention module for soft phase aggregation. 3DDFA [30] matches CNN and RGB images, evaluates shape-related parameters, and transforms the head into a dense 3D model to facilitate detection even in a closed environment. FAN [21] is a landmark detection method that solves 2D-3D problems by merging features of landmarks across multiple layers. Figure 9 compares our model with FSA-Net and HopeNet on a few examples, further demonstrating the robustness of our model.

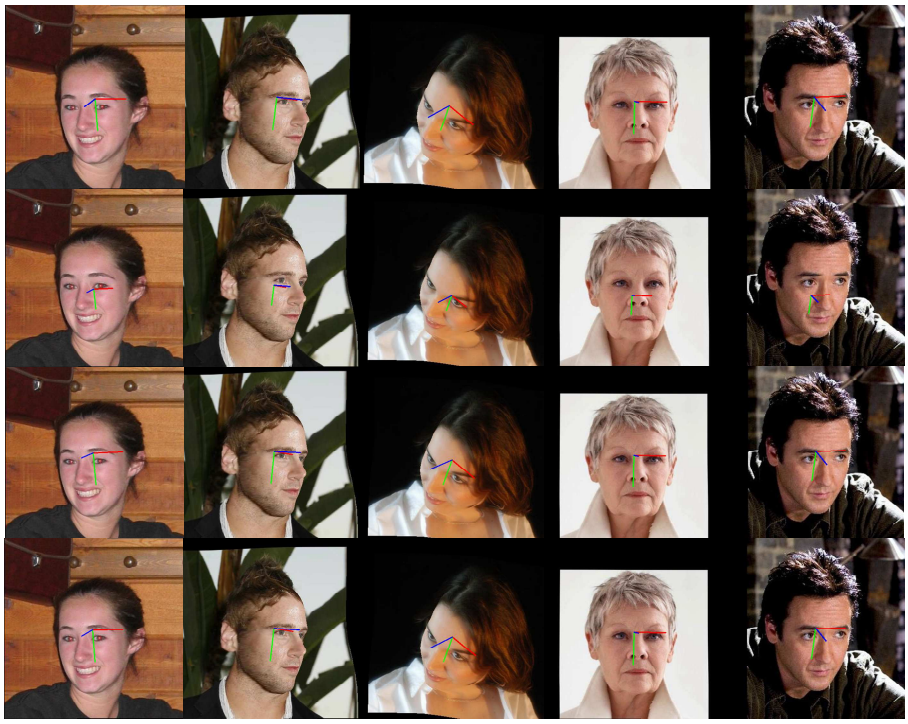


FIGURE 9. **Pose estimation on the AFLW2000 dataset. From top to bottom, the GroundTruth, results of Hopenet, results of FSA-Net and our results. The blue line indicates the direction the subject is facing. The green line represents the downward direction and the red line is pointing to the side.**

Figure 9 displays the ground truth, the results of HopeNet [8], the results of FSA-Net [25], and our results. The blue line indicates the direction the subject is facing; the green line indicates the downward direction; and the red line represents the side. The performance of the method is based on landmarks and depends on the underlying face alignment algorithm, whereas our method does not rely on other auxiliary aspects.

For further analysis, we applied our algorithm to two additional cases (no CapsNet block and SENet block); CapsNet block is the part for feature fusion, and SENet is an attention mechanism network added to feature extraction. All calculations were performed according to the MAE standard to better demonstrate the process on the IN-THE-WILD Model.

As shown in Table 1, this method performs best when tested on the AFLW dataset, reaching the minimum on yaw, pitch, and roll with an average deviation angle value of 4.26. Compared with other methods, the detection result value of this method changed significantly. Therefore, the method in this paper is the best in detection performance. As shown in Table 2, this method also displayed the best performance when tested on the BIWI dataset. It reached the minimum on yaw, pitch, and roll with an average deviation angle value of 3.95. Therefore, it was confirmed that the method in this paper is the best in detection performance.

TABLE 1. **Comparisons with the state-of-the-art methods on the AFLW2000 dataset. All are trained on the 300W-LP dataset.**

Method	Yaw(deg)	Pitch(deg)	Roll(deg)	Avg(deg)
3DDFA [30]	5.40	8.53	8.25	7.39
FAN [21]	6.36	12.2	8.71	9.11
HopeNet [8]	6.47	6.57	5.44	6.16
FSA-Net [25]	4.50	6.08	4.64	5.07
Ours (no SENet block)	4.13	5.34	4.24	4.57
Ours (no CapsNet block)	4.61	5.86	4.58	5.01
Ours	**4.25**	**4.96**	**3.57**	**4.26**

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/ACCESS.2021.3119615, IEEE Access

TABLE 2. Comparisons with the state-of-the-art methods on the BIWI dataset. All are trained on the 300W-LP dataset.

Method	Yaw(deg)	Pitch(deg)	Roll(deg)	Avg(deg)
3DDFA [30]	36.2	12.3	8.78	19.1
FAN [21]	8.53	7.48	7.63	7.89
HopeNet [8]	5.17	6.98	3.39	5.18
FSA-Net [25]	4.27	4.96	2.76	4.00
Ours (no SENet block)	4.40	4.88	3.25	4.17
Ours (no CapsNet block)	4.61	5.76	3.47	4.61
Ours	**4.14**	**4.66**	**3.06**	**3.95**

2) EVALUATION OF THE LABORATORY MODEL

The laboratory model was trained using 70% of the BIWI dataset. This dataset contains a variety of model assessment information. In addition to RGB color information, depth image information and time information can also be used.

As shown in Table 3, Martin et al. [19] estimated head posture using a depth camera to obtain a depth image. Drouard et al. [14] used the hybrid method of linear regression to acquire a high-dimensional feature vector to determine the head posture. The table also records the time spent by each method to test the image.

As shown in Table 3, the performance of the proposed method on the BIWI training dataset was relatively good. The deviation angle on yaw is 2.64, which is the lowest value among all methods, and the average deviation angle is slightly higher than that of Hopenet [8] with 3.31 degrees. However, for the method proposed in this manuscript, the shortest run time is 0.53ms, and the test efficiency is the highest. Therefore, the experimental results demonstrate that this method has certain advantages with respect to detection error and test time in the indoor environment.

TABLE 3. Comparisons with the state-of-the-art methods on the BIWI dataset. 70% of videos are used for training and 30% for testing

Method	Input	Yaw(deg)	Pitch (deg)	Roll (deg)	Avg(deg)	Runtime(ms)
Martin [19]	Depth	3.6	2.5	2.6	2.9	0.76
Drouard [14]	RGB	4.24	5.43	4.13	4.60	0.68
FSA-Net [25]	RGB	2.89	4.29	3.60	3.60	0.60
HopeNet [8]	RGB	**3.29**	**3.39**	**3.00**	**3.23**	**0.56**
Ours (no SENet block)	RGB	3.31	3.42	3.39	3.37	0.62
Ours (no CapsNet block)	RGB	3.46	4.16	3.41	3.68	0.49
Ours	RGB	**2.64**	**3.98**	**3.33**	**3.31**	**0.53**

3) EVALUATION IN THE PARTIALLY OCCLUDED CASE

In order to verify the performance capability of the method in covering the head, we tested the accuracy of our algorithm under different occlusion conditions. As shown in Fig. 10, we divided the facial region into two areas: eyes and mouth. The two regions were then occluded separately to calculate the accuracy for the non-occluded face area. The occlusion rate of the entire face from top to bottom as well as in the opposite direction was 0%, 12.5%, 25%, 37.5%, 50%, corresponding to two important feature intervals of the eye area and the mouth area, respectively.

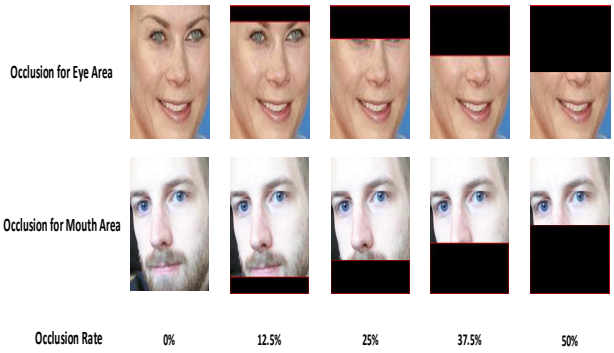


FIGURE 10. Example of facial image with different occlusion rates for eye and Mouth Areas.

Table 4 displays the relative accuracy rate for the head postures with occlusion of the eye area. When the occlusion rate reaches 50%, the eye area is blocked and the mouth area is active; the relative accuracy rate is 82.97%.

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/ACCESS.2021.3119615, IEEE Access

Table 5 displays the relative accuracy rate for the head postures with occlusion of the mouth area. When the occlusion rate reaches 50%, the mouth area is blocked and the eye area is active; the relative accuracy rate is 89.81%.

Table 6 displays the relative accuracy rate of the head postures of each algorithm with occlusion of the eye area. The results show that the proposed algorithm is the best in the case of occlusion of each eye.

Table 7 displays the relative accuracy rate of the head postures of each algorithm with occlusion of the mouth area. The results show that the proposed algorithm is the best in the case of occlusion of the mouth.

Compared to the eye region, the mouth region contributes less to the head pose estimation. This shows that our method can address the problems associated with wearing masks or head-covering. Compared with other algorithms for occlusion experiments, the method proposed in this paper has the highest accuracy rate, which proves the superiority of our model with respect to occlusion.

TABLE 4. Accuracy of Head Posture Recognition with Occlusion of Eye Area

Occlusion Rate	Accurate (%)			
	Yaw	Pitch	Roll	Avg
0%	99.23	86.71	90.86	92.27
12.5%	98.75	87.55	89.64	91.98
25%	96.17	85.23	86.78	89.39
37.5%	93.41	82.28	82.25	85.98
50%	89.22	80.85	78.85	**82.97**

TABLE 5. Accuracy of Head Posture Recognition with Occlusion of Mouth Area

Occlusion Rate	Accurate (%)			
	Yaw	Pitch	Roll	Avg
0%	99.23	86.71	90.86	92.27
12.5%	98.98	87.40	91.51	92.63
25%	98.54	83.78	89.53	90.62
37.5%	98.76	82.07	90.52	90.45
50%	98.08	81.56	89.78	**89.81**

TABLE 6. Comparisons with the state-of-the-art methods with Occlusion of Eye Area

Occlusion Rate	Accurate (%)				
	3DDFA [30]	FAN [21]	HopeNet [8]	FSA-Net [25]	Ours
0%	90.33	92.01	92.13	92.15	**92.27**
12.5%	88.45	89.73	90.34	91.79	**91.98**
25%	83.12	88.21	88.92	89.12	**89.39**
37.5%	78.94	84.17	84.93	85.23	**85.98**
50%	75.52	80.52	81.34	81.98	**82.97**

TABLE 7. Comparisons with the state-of-the-art methods with Occlusion of Mouth Area

Occlusion Rate	Accurate (%)				
	3DDFA [30]	FAN [21]	HopeNet [8]	FSA-Net [25]	Ours
0%	90.33	92.01	92.13	92.15	**92.27**
12.5%	89.47	92.21	92.42	92.56	**92.63**
25%	86.26	88.94	90.03	90.35	**90.62**
37.5%	83.62	87.73	89.54	90.16	**90.45**
50%	82.38	85.22	88.36	89.39	**89.81**

IV. CONCLUSION AND FUTURE WORK

In this study, we developed a deep neural network model MR-CapsNet to predict head posture. Our method can infer the head posture from only an image without additional factors such as a depth map or facial markers. Initially, MTCNN [37] was used to detect the target, which was then divided into three levels. A residual attention block and SE block were used for feature extraction. CapsNet is an emerging network that is more sensitive to posture information, as reflected in facial expressions, than a traditional CNN. Therefore, we combined the extracted feature map with CapsNet to obtain more accurate attitude information. Finally, a multi-stage regression function was used to predict head posture. The MAE of our model is superior to that of other advanced methods.

In the future, we will continue to improve our model. At present, the detection ability in outdoor environments is not ideal. To further improve the pertinence and accuracy of prediction, additional low-resolution datasets need to be integrated. Currently, capsules are emerging; however, there are no relevant application examples of CapsNet in the field of head posture estimation, which requires further attention. In this study, although only the estimation of head posture was considered, the overall framework is widely applicable.

V. AUTHOR CONTRIBUTIONS

Hao Fang conceived the algorithms, Jun-Qing Liu designed the experiments; Kai Xie reviewed the paper; Chang Wen conducted the comparative experiment; Peng Wu and Xin-Yu Zhang is responsible for data collection; Jian-Biao He checked the spelling and made suggestions.

REFERENCES

[1] R. Stiefelhagen, C. Fugen, R. Gieselmann, H. Holzapfel, K. Nickel and A. Waibel, "Natural human-robot interaction using speech, head pose and gestures," 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566), Sendai, Japan, 2004.

[2] E. Murphy-Chutorian and M. M. Trivedi, "Head Pose Estimation and Augmented Reality Tracking: An Integrated System and Evaluation for Monitoring Driver Awareness," in IEEE Transactions on Intelligent Transportation Systems, vol. 11, no. 2, pp. 300-311, June 2010, doi: 10.1109/TITS.2010.2044241.

[3] C. Chen, R. C. Ugarte, C. Wu and H. Aghajan, "Discovering social interactions in real work environments," 2011 IEEE International Conference on Automatic Face & Gesture Recognition (FG), Santa

Barbara, CA, USA, 2011, pp. 933-938, doi: 10.1109/FG.2011.5771376.

[4] Leroy J, Rocca, François, Mancas M, et al. Second screen interaction: an approach to infer tv watcher's interest using 3d head pose estimation. [J]. 2013.

[5] 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, Apr. 2009.

[6] Y. Taigman, M. Yang, M. Ranzato and L. Wolf, "DeepFace: Closing the Gap to Human-Level Performance in Face Verification," 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 2014, pp. 1701-1708.

[7] Schroff F, Kalenichenko D, Philbin J. FaceNet: A Unified Embedding for Face Recognition and Clustering[J]. 2015.

[8] N. Ruiz, E. Chong and J. M. Rehg, "Fine-Grained Head Pose Estimation Without Keypoints," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Salt Lake City, UT, USA, 2018, pp. 2155-215509, doi: 10.1109/CVPRW.2018.00281.

[9] X. Zhu, Z. Lei, X. Liu, H. Shi and S. Li, "Face Alignment Across Large Poses: A 3D Solution," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016 pp. 146-155.

[10] Simonyan, Karen & Zisserman, Andrew. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv 1409.1556.

[11] G. Hinton, S. Sabour, and N. Frosst, "Matrix capsules with EM routing." In Proc. 6th International Conference Learn. Represent., May 2018, pp. 1–15.

[12] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules." In Proc. International Conference Neural Inf. Process. Syst., Nov. 2017, pp. 3856–3866.

[13] H.Liang, J.Hou, J Yuan, D.Thalmann.(2017) "Random Forest with Suppressed Leaves for Hough Voting" Computer Vision – ACCV 2016. ACCV 2016. Lecture Notes in Computer Science, vol 10113. Springer, Cham.

[14] V. Drouard, R. Horaud, A. Deleforge, S. Ba, and G. Evangelidis, "Robust head-pose estimation based on partially-latent mixture of linear regressions,"IEEE Trans. Image Process., vol. 26, no. 3, pp. 1428–1440, Mar. 2017.

[15] I. Chamveha et al., "Appearance-based head pose estimation with scene-specific adaptation," 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), Barcelona, Spain, 2011, pp. 1713-1720.

[16] K. Diaz-Chito, J. Martinez Del Rincon, A. Hernandez-Sabate, and D. Gil, "Continuous head pose estimation using manifold subspace embedding and multivariate regression," IEEE Access, vol. 6, pp. 183[25]–18334, 2018, doi: 10.1109/ACCESS.2018.2817[25]2.

[17] L.Liang, F. Wen, and J. Sun. "Face alignment via component-based discriminative search." Springer Berlin Heidelberg, doi:10.1007/978-3-540-88688-4_6. 2012.

[18] F. Timothy, J.Christopher, H.David and J.Graham. "Active shape models-their training and application," Computer Vision and Image Understanding, 61, pp.38–59, 1995.

[19] M. Martin and R. Stiefelhagen, "Real Time Head Model Creation and Head Pose Estimation on Consumer Depth Cameras," 2014 2nd International Conference on 3D Vision, Tokyo, Japan, 2014, pp. 641-648.

[20] G.Fanelli, T.Weise, J.Gall, and L.Van. "Real time head pose estimation from consumer depth cameras," In Joint Pattern Recognition Symposium, pages 101–110. Springer, 2011.

[21] Adrian Bulat and Georgios Tzimiropoulos. "How far are we from solving the 2D & 3D face alignment problem? (and a dataset of 230,000 3D facial landmarks)." In Proceedings of International Conference on Computer Vision (ICCV), 2017.

[22] L. Lin, X. Rong, W. Fang, and S. Jian. "Face alignment via component-based discriminative search," In Proceedings of European Conference on Computer Vision (ECCV), pages 72–85. Springer, 2008.

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

[24] 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, Feb. 2009.

[25] T. Yang, Y. Chen, Y. Lin and Y. Chuang, "FSA-Net: Learning Fine-Grained Structure Aggregation for Head Pose Estimation From a Single Image," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 2019, pp. 1087-1096.

[26] S. S. Mukherjee and N. M. Robertson, "Deep Head Pose: Gaze-Direction Estimation in Multimodal Video," in IEEE Transactions on Multimedia, vol. 17, no. 11, pp. 2094-2107.

[27] A. Kumar, A. Alavi and R. Chellappa, "KEPLER: Keypoint and Pose Estimation of Unconstrained Faces by Learning Efficient H-CNN Regressors," 2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017), Washington, DC, USA, 2017, pp. [25]8-265.

[28] S. Yang et al., "RS-CapsNet: An Advanced Capsule Network," in IEEE Access, vol. 8, pp. 85007-85018, 2020, doi: 10.1109/ACCESS.2020.2992655.

[29] J. Xia, L. Cao, G. Zhang, and J. Liao, "Head pose estimation in the wild assisted by facial landmarks based on convolutional neural networks," IEEE Access, 7, pp. 48470–48483, 2019.

[30] 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. Vis. Pattern Recognit. (CVPR), Las Vegas, NV, USA, Jun. 2016, pp. 146–155.

[31] R. Ranjan, S. Sankaranarayanan, C. D. Castillo and R. Chellappa, "An All-In-One Convolutional Neural Network for Face Analysis," 2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017), Washington, DC, USA, 2017, pp. 17-24.

[32] J. Gu, X. Yang, S. De Mello and J. Kautz, "Dynamic Facial Analysis: From Bayesian Filtering to Recurrent Neural Network," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 2017, pp. 1531-1540.

[33] 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," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 41, no. 1, pp. 121-135.

[34] F. Chang, A. T. Tran, T. Hassner, I. Masi, R. Nevatia and G. Medioni, "FacePoseNet: Making a Case for Landmark-Free Face Alignment," 2017 IEEE International Conference on Computer Vision Workshops (ICCVW), Venice, 2017, pp. 1599-1608.

[35] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016 pp. 770-778.

[36] D.Chen, S.Ren, Y.Wei, X.Cao, J. Sun "Joint Cascade Face Detection and Alignment," In Computer Vision – ECCV 2014. ECCV 2014. Lecture Notes in Computer Science, vol 8694. Springer, Cham.

[37] A.F.Abate, P.Barra, C.Pero, et al, "Head pose estimation by regression algorithm". Pattern Recognition Letters, pp.179-185, 2020.

[38] C. Song, L. He, W. Q. Yan and P. Nand, "An Improved Selective Facial Extraction Model for Age Estimation," 2019 International Conference on Image and Vision Computing New Zealand (IVCNZ), Dunedin, New Zealand, 2019, pp. 1-6, doi: 10.1109/IVCNZ48456.2019.8960965.

[39] J. Hu, L. Shen, S. Albanie, G. Sun and E. Wu, "Squeeze-and-Excitation Networks," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, no. 8, pp. 2011-2023.

[40] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multi-task cascaded convolutional networks," CoRR, vol. abs/1604.02878, pp. 1–5, Apr. 2016.

[41] G. Fanelli, M. Dantone, J. Gall, A. Fossati, and L. Van Gool, "Random forests for real time 3D face analysis," Int. J. Comput. Vis., vol. 101, no. 3, pp. 437–458, Aug. 2012.

[42] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in Proc. 3rd Int. Conf. Learn. Represent. (ICLR), San Diego, CA, USA, Y. Bengio and Y. LeCun, Eds., May 2015, pp. 1–15.

[43] X. Zhu and D. Ramanan, "Face detection, pose estimation, and landmark localization in the wild," 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 2012, pp. 2879-2886, doi: 10.1109/CVPR.2012.6248014.
[44] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, and N. Kumar, "Localizing parts of faces using a consensus of exemplars," in Proc. 24th IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Colorado Springs, CO, USA, Jun. 2011, pp. 545–552.
[45] M. Kostinger, 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 Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCV), Barcelona, Spain, Nov. 2011, pp. 2144–2151.
[46] A. Fathi, J. K. Hodgins, and J. M. Rehg, "Social interactions: A firstperson perspective," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Providence, RI, USA, Jun. 2012, pp. 1226–1233.