Head posture detection with embedded attention model

Jin Han, Yansheng Liu*
Shandong University of Science and Technology Computer Science and Engineering Qingdao, Shandong, China

*Corresponding author e-mail: 1017620952@qq.com

Abstract. Based on Convolutional Neural Network, the paper presents a compact detection algorithm that can estimate the head pose from a single picture. Our method is based on soft stage wise regression. In order to reduce model complexity, three-dimensional detection of the "pitch, yaw, and roll" of the head posture adopts multi-level classification. Each level of classification requires only a small number of classification tasks and fewer neurons. In order to enhance the feature expression of the algorithm, the attention model is embedded. Attention model includes channel attention structure and spatial attention structure, enhancing the feature expression of the feature map in the two dimensions of the intermediate feature map channel and space. The attention model can be seamlessly integrated into the CNN architecture with low overhead. The experiment proves that the improved algorithm compares the method model proposed by Yang with a smaller complexity of 4.36M and an average absolute error of 0.7%~0.9%.

1. Introduction
Head posture analysis has been a hot research topic in the field of computer vision [1-6, 13-17]. The head attitude is a 3D vector including "Pitch", "Yaw", and "Roll". For head pose detection from a single picture, you need to learn mapping information from 2D space to 3D space. In the existing methods, one type of method uses the depth information of the picture to capture the missing 3D information of the image [7, 8, 11]. This type of method requires special image acquisition equipment for image acquisition, and it needs to learn depth information and time information at the same time. Implementation of high computational cost cyclic structure. Another method based on facial feature coordinates [9, 13-16]. This method needs to learn a large number of facial features. This will generate a lot of calculations, a complex algorithm model is required. The paper gives the CAR-Net algorithm, which is based on the SSR-Net [10] model. It is a lightweight model for pose estimation based on a single image. In order to reduce the complexity of the model, a multi-level classification method is used based on soft-stage regression. Each level of classification is subdivided on the basis of the upper-level classification, requiring only small network parameters. In order to collect more image information, this model uses a five-stream heterogeneous structure, and each stream collects multi-scale heterogeneous information through different activation functions. Each stream extracts intermediate feature maps at various stages, and the extracted intermediate feature maps are sent to the fusion module for information aggregation to obtain multi-feature intermediate feature maps.

In order to improve the performance of convolutional neural networks and the important feature expression of feature maps. The paper is based on the Convolutional Block Attention Module (CBAM)
[18] algorithm which embed attention models. The model can be seamlessly embedded in the CNN architecture, and the overhead can be used to extract the key position information of the feature map under negligible premise. The attention model is divided into a channel attention model and a spatial attention model, and the key information is highlighted in two dimensions of the channel axis and the space axis, respectively. This attention mechanism effectively improves the key information extraction ability of the convolutional neural network and improves the network performance.

2. Head pose detection algorithm

2.1. Algorithm Description

The paper gives a set of head training pictures \( X = \{x_1, x_2, ..., x_n\} \). The real head pose vector \( y_n \) corresponding to each picture \( x_n \). Each vector contains Pitch, Yaw, and Roll. The goal of the paper is to find a function \( F \) that predicts the pose vector \( x \) of each picture. The average absolute error MAE of the true attitude value and the predicted attitude value is reduced. Use this to find the function \( F \).

\[
J(X) = \frac{1}{n} \sum_{i=1}^{n} |\tilde{y}_i - y_i|
\]  

(1)

This algorithm is based on the SSR-Net algorithm design. The SSR-Net algorithm provides a compact model for age estimation from a single image. SSR-Net divides the age domain into classes of several age groups. The network performs classification tasks and outputs the probability distribution of each age group. The CAR-Net algorithm uses a coarse-to-fine classification strategy in order to reduce the model size with high accuracy, and only performs a small number of intermediate classification tasks at each stage. For example, in the current stage classification, the pitch angle is divided into "-30°-0°", "0°-30°", "30°-60°", and "60°-90°". The next stage classification is based on the previous stage classification. Within the classification decision. Therefore, the CAR-Net algorithm uses a hierarchical classification method and uses the following soft-stage regression method to estimate the angle:

\[
\tilde{y} = \sum_{k=1}^{K} \tilde{p}(k) \tilde{\mu}(k)
\]  

(2)

\( K \) is the number of segments, \( \tilde{p}(k) \) is the probability of the \( k \)-th stage, \( \tilde{\mu}(k) \) and is the vector value of this angle group at the \( k \)-th stage.

The network is divided into \( K \) stages, there are \( s_k \) classes in the \( k \)-th stage, and the Pitch stage predicted for each network is:

\[
\tilde{y} = \sum_{k=1}^{K} \tilde{p}(k) \tilde{\mu}(k) = \sum_{k=1}^{K} \sum_{i=0}^{s_k-1} p^k_{i} \cdot i \left( \frac{V}{\prod_{j=1}^{s_j}} \right)
\]  

(3)

The width of each class is \( h_k \), \( h_k = \frac{v}{\prod_{j=1}^{s_j}} \), and \( i \) is the index of each class.

In order to enhance the flexibility of classification and reduce quantization errors, dynamic range classification is introduced. Each class is allowed to shift and scale according to the input image. The shift can adjust the index \( i \) of each class, and the zoom can adjust the class width \( h_k \). In order to achieve the shift, a shift vector \( \tilde{\eta}(k) \) is added to each class index, and the index is changed to \( i + \tilde{\eta}(k) \). In order to achieve scaling, a scale factor \( \Delta k \) is introduced to scale the width of the \( k \)-th class, \( h_k = s_k (1 + \Delta k) \), where the adjusted class width is \( \tilde{h}_k = \frac{v}{\prod_{j=1}^{s_j}} \). Then input a picture into the CAR-Net algorithm, train it on the network, extract different features through the five-flow structure, and output \( K \) group of stage parameters \( \{\tilde{p}(k), \tilde{\eta}(k), \Delta k\} \). The angle will be predicted based on the output parameters.
In the CAR-Net, different from the SSR-Net, CAR-Net uses multi-level classification to detect angles in the Pitch, Yaw, and Roll three-dimensional spaces. SSR-Net estimates age in the one-dimensional space of age.

2.2. Algorithm design

Fig 1 provides the architecture of the algorithm. The input image contains five stream structures with a total of K stages (Fig 1 has 3 stages). Among these five streams, the basic building blocks include 3 * 3 convolution kernels, batch normalization, non-linear activation functions, and 2 * 2 pools. Each stream uses different types of activation functions and pools. ReLU, ELU, PReLU, Tanh, and Sigmoid functions. The pooling layer includes average pooling and maximum pooling. Through these five different types of stream structures, different types of features can be extracted.

![Figure 1. CAR-Net](image)

Each stream extracts feature maps at each stage. For the k-th stage, 5 feature maps extracted from the five streams are input into the convolution kernel with a size of 1 * 1 for convolution operation. The feature map is converted to the c channel, and the average pooling is performed after convolution to reduce the size of the feature map to w × h. Therefore, a feature map \( U_k \) can be obtained which size is w × h × c. After extracting the feature map \( U_k \), in order to aggregate the feature maps into more representative features, this paper introduces the channel attention structure (the channel attention structure will be introduced in detail in Section 2). The channel attention map \( A_k \) can highlight the key features in the feature map \( U_k \) pair channel and space dimensions, and improve the network feature extraction ability without significantly increasing the amount of calculation and parameters. A new feature map \( U'_k \) is obtained by fusing the feature map \( U_k \) and the channel attention map \( A_k \).
With reference to the SSR-Net network, aggregate the new feature maps obtained at each stage, generating new feature map collection. After passing the feature map set through the fully connected layer, \( P \) in the SSR-Net model can be obtained \( \{ P^{(k)}, \eta^{(k)}, \Delta k \} \). The obtained parameters are brought into the SSR-Net model and the head pose estimation will be performed.

2.3. Network structure
As shown in Fig 1, this network model contains five stream structures. Each stream contains Conv (3 * 3) (3 * 3 convolution kernel), BN (batch normalization), activation function and 2 * 2 pooling layer. Each stream uses a different activation function to make the image heterogeneous and to discover different features. Each stream contains 3 stages, and the feature maps generated in each stage will pass the feature map fusion module to generate new feature maps. The fusion module contains Conv (1 * 1), Tanh activation function and pooling layer to get more representative features.

3. Embedded attention module
In order to improve the performance of convolutional neural networks, the dissertation embeds the attention module on the basis of CBAM. In the convolution operation, cross-channel and spatial information are mixed together to extract information features. Given a feature map extracted by a convolution operation, the attention module will derive the attention map from the channel and space dimensions of the feature map, respectively. Therefore, in order to learn more representative features in the two dimensions of channel and space, the attention module is divided into channel attention module and spatial attention module. The channel attention module tells us that the feature map "what" is important, and the spatial attention module tells us that the feature map "where" has important information. With the attention module, it is easier to extract representative features.
A feature map $U_k$ is given as input. $U_k \in \mathbb{R}^{w\times h\times c}$. The feature map passes through the channel attention module and the spatial attention module in turn.

The channel attention module calculates the channel attention map $M_c \in \mathbb{R}^{c\times 1\times 1}$, the spatial attention module calculates the spatial attention map $M_s \in \mathbb{R}^{w\times h\times 1}$. The entire feature map attention module calculation process is as follows:

$$F' = M_c(U_k) \otimes U_k$$

$$F'' = M_s(F') \otimes F'$$

$\otimes$ Represents matrix element-wise multiplication. The flow of the attention module is as follows. The input feature map $F$ and the channel attention map $M_c$ are multiplied element by element to obtain the feature map $F'$. The obtained feature map $F'$ and the spatial attention map $M_s$ are multiplied element by element to output the final attention map $F''$. Attention module process is shown in Fig3.

### 3.1. Channel Attention Module

Each channel of the feature map will be regarded as a feature detector, so it is meaningful that the channel attention module pays attention to which channel of the input image needs to be focused. As the champion of the ImageNet2017 classification task, Hu et al. Proposed the SENet [18] model, which uses global mean-pooling to calculate channel attention. Mean-pooling averages feature points in a neighborhood and weakens strong activation values. In some cases, mean-pooling can achieve better results, because it averages out strong activation values, it leads to loss of important information. In order to retain more useful information, this paper uses max-pooling and mean-pooling operations to calculate channel attention information. Maximum pooling preserves more texture information in the image, and average pooling preserves more background information in the image.

Fig 4 is an example diagram of the channel characteristic module. Input the feature map $U_k$, and then perform maximum pooling and average pooling operations to compress the feature maps to generate two 1x1xC channel maps; Send them into a two-layer shared neural network MLP, the number of neurons in the first layer is $C/r$, the number of neurons in the second layer is $C$, the activation function is ReLU, and two adjusted vectors are output; Two features obtained are added and spliced into a one-dimensional vector, which is output as $M_c$ through a function (Sigmoid function). Multiply $M_c$ with the original feature map $U_k$ to obtain a new channel feature map $F'$. The calculation formula of $M_c$ is:

$$M_c = \sigma \left( \text{MLP} \left( \text{AvgPool}(U_k) \right) + \text{MLP} \left( \text{MaxPool}(U_k) \right) \right)$$

\(\sigma\) is the Sigmoid function.
The spatial attention module is generated by using the spatial relationship of features. Unlike the channel attention module, the spatial attention module mainly focuses on the position information of the feature map. The input feature map $F'$ is still subjected to maximum pooling and average pooling operations to aggregate the channel information of the feature map to obtain two two-dimensional feature maps $F_{avg}^s$ and $F_{max}^s (R^{w \times h \times 1})$, which are stitched according to the channel dimensions to form a dimension 2. Then use a 16 * 16 convolution kernel to perform the convolution operation (after multiple rounds of testing in this article, the 16 * 16 convolution kernel has the best convolution effect). After $\sigma$ function, a new spatial feature map $M_s$ with a channel of 1 is obtained. The obtained feature map is multiplied with the original feature map $F'$ to obtain a new feature map $F''$. The calculation formula of $M_s$ is as follows:

$$M_s = \sigma(f^{16 \times 16}([AvgPool(F'); MaxPool(F')]))$$  \hspace{1cm} (7)$$

$$M_s = \sigma(f^{16 \times 16}([F_{avg}^s, F_{max}^s]))$$  \hspace{1cm} (8)$$

4. Experiments

4.1. Experimental design

In order to evaluate the effectiveness and robustness of the algorithm, the experiment uses 300W-LP-3D as the training set, AFLW2000 and BIWI as the test set, and cross-ablation experiments are performed. 300W-LP-3D [14] is based on the 300W dataset to render faces into more pose forms. This data set has 61225 samples, and the data set is enlarged to 122450 samples by flipping. The AFLW2000 data set is a data set obtained by re-marking the first 2000 2D images of the AFLW data set with 68 coordinate points in 3D.

The experimental environment is as follows: hardware Inter i7 CPU, GTX1050TiGPU, 16G memory. Software: Ubuntu operating system, Python3.7. The initial learning rate is set to 0.01, and the learning rate is reduced by 10 times every 30 cycles, for a total of 90 cycles. The time to train the model is about 1.5 hours, and the time to test each picture is about 70ms.

The experimental process is divided into three phases, namely the data preprocessing phase, the training phase and the testing phase. In the data pre-processing stage, the picture format is processed into a unified raw data format. In the training phase, the unified data set is sent to CAR-Net for training. When the output of CAR-Net does not match the expected value, a back-propagation process is performed. The network weight is adjusted through the training samples and the expected value, and it is continuously reduced. Error. In the test phase, after the test pictures are given, they are sent to CAR-Net to get the test results. Fig5 is a system diagram.
4.2. Experimental results and analysis

4.2.1. Experimental comparison method. FAN [13] is an excellent detection method based on the key coordinates of the face. This algorithm achieves high performance in 2D and 3D face alignment networks. Dlib [14] is a standard face library. It uses cascaded regressors to do face feature alignment with good results. 3DDFA [15] uses CNN to fit a 3D model into an RGB image, and can also detect the alignment of the face when the face is occluded. Hopenet [16] proposed a concise and robust method to determine the pose, which proposed a convolutional neural network with 3 separate losses. FSA-Net [17] proposed a fine-grained network structure to improve the performance of convolutional neural networks through the fusion of three scoring functions. Hopenet, FSA-Net and CAR-Net are all non-face-based methods.

4.2.2. AFLW2000 analysis of test set results. In this experiment, using the average absolute error (MAE) and size (MB) as the evaluation criteria, through testing on the AFLW2000 test set, three non-face-based methods Hopenet [16], FSA-Net and CAR-Net performance is better than the three methods based on face coordinates [14-16]. From the experimental results, it can be seen that this algorithm has a smaller model than the other two non-face based methods, only 4.36M. Compared with the fine-grained structure model proposed by FSA-Net [17], the attention model embedded in this algorithm will overfit in a certain parameter value, but the error is improved and the model is smaller. MAE is only 5.02, with a 0.9% improvement.

| Method          | Yaw  | Pitch | Roll | MAE  | MB  |
|-----------------|------|-------|------|------|-----|
| FAN(12 points)  | 6.35 | 12.27 | 8.71 | 9.11 | 183 |
| Dlib(68 points) | 23.15| 13.63 | 10.54| 15.77| -   |
| 3DDFA           | 5.40 | 8.53  | 8.25 | 7.39 | -   |
| Hopenet(α=2)    | 6.47 | 6.56  | 5.44 | 6.17 | 95.9|
| FSA-Net         | 4.50 | 6.08  | 4.64 | 5.07 | 5.1 |
| CAR-Net         | 4.98 | 5.53  | 4.55 | 5.02 | 4.36|

Table 1. 300W-LP-3D is the training set and AFLW2000 is the test set.
4.2.3. **BIWI analysis of test set results.** In the case of the same training set and test set, the average absolute error (MAE) and size (MB) are still used as comparison standards. By comparing several typical algorithms, it can be concluded that the average absolute error of the methods based on non-face coordinates is better than the methods based on face coordinates [14-16], and the model is smaller. Compared with several non-coordinate-based algorithms, the CAR algorithm is smaller than the other two non-coordinate-based algorithms (Hopenet, FSA-Net), with only 4.36M. According to the experimental results, the CAR algorithm cannot always be better than FSA-Net in Yaw. Subsequent experiments will continue to adjust the parameters to achieve the best results. 0.7% improvement in MAE compared to Yang [17].

Table 2. 300W-LP is the training set, and BIWI is the test set.

|                | Yaw  | Pitch | Roll | MAE  | MB   |
|----------------|------|-------|------|------|------|
| FAN(12 points) | 8.53 | 7.48  | 7.63 | 7.88 | 183  |
| Dlib(68 points)| 16.75| 13.82 | 6.19 | 12.24| -    |
| 3DDFA          | 36.2 | 12.3  | 8.78 | 19.1 | -    |
| Hopenet(α=2)   | 5.17 | 6.98  | 3.39 | 5.18 | 95.9 |
| FSA-Net        | 4.27 | 4.96  | 2.76 | 4.00 | 5.1  |
| CAR-Net        | 4.35 | 4.83  | 2.73 | 3.97 | 4.36 |

5. **Summary**
This paper first presents a lightweight head pose detection model based on SSR-Net, which detects the head pose from a single picture. The five-stream heterogeneous structure is given to obtain more heterogeneous information, and the new feature map is obtained by fusing the intermediate feature maps extracted from different streams. Then, the attention algorithm model is embedded, and the feature expressiveness of the feature map is improved from the channel and space dimensions of the feature map. The experiments show that the average absolute error of this algorithm is better than other algorithms. And the algorithm model is small, the size is only 4.36M.

**References**

[1] LU Yifan, ZHANG Songhai. Object detection in optical remote sensing images with convolutional neural networks [J]. Sciencepaper Online, 2017, 12 (14): 1583 - 1589+1633. (in Chinese).

[2] WANG Feng, LIU Qingshan. Fine pedestrians recognition based on double convolutional neural network [J], Sciencepaper Online, 2017, 12 (14): 1578 - 1582. (in Chinese).

[3] Sankha S Mukherjee and Neil Martin Robertson. Deep head pose: Gaze-direction estimation in multimodal video.IEEE Transactions on Multimedia (TMM), 17 (11): 2094 – 2107, 2015.

[4] LU Hongtao, Zhang Qinchuan. Applications of Deep Convolutional Neural Network in Computer Vision [J]. Journal of Data Acquisition and Processing, 2016, 31 (01): 1-17. (in Chinese).

[5] WU Sinan. Study on face recognition technology based on neural network [J]. Wireless Internet Technology, 2019, 16 (03): 136 - 137. (in Chinese).

[6] Andre Litvin, Kamal Nasrollahi, Sergio Escalera, Cagri Ozcinar, Thomas B.Moeslund, Gholamreza Anbarjafari, A novel deep network architecture for reconstructing RGB facial images from thermal for face recognition [J]. Multimedia Tools and Applications, 2019, Vol.78 (18), pp. 25259 - 25271.

[7] Pingyu Wang, Fei Su, Zhicheng Zhao,Yandong Guo,Yanyun Zhao,Bojin Zhuang. Deep class-skewed learning for face recognition [J]. Neurocomputing, 2019.

[8] Gabriele Fanelli, Thibaut Weise, Juergen Gall, and Luc Van Gool. Real time head pose estimation from consumer depth cameras. In Joint Pattern Recognition Symposium, pages 101 – 110. Springer, 2011.

[9] Adrian Bulat and Georgios Tzimiropoulos. How far are wefrom 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.

[10] Yang, Tsun-Yi and Huang, Yi-Hsuan and Lin, Yen-Yu and Hsiu, Pi-Cheng and Chuang, Yung-Yu, SSR-Net: A Compact Soft Stagewise Regression Network for Age Estimation. IJCAI, 2018.

[11] Gregory P. Meyer, Shalini Gupta, Iuri Frosio, Dikpal Reddy, and Jan Kautz. Robust model-based 3D head pose estimation. In Proceedings of International Conference on Computer Vision (ICCV), 2015.

[12] Kumar A, Alavi A, Chellappa R. KEPLER: Keypoint and Pose Estimation of Unconstrained Faces by Learning Efficient H-CNN Regressors [J]. 2017.

[13] Bulat A, Tzimiropoulos G. How Far are We from Solving the 2D & 3D Face Alignment Problem (and a Dataset of 230,000 3D Facial Landmarks) [J]. 2017.

[14] Zhu, Xiangyu, et al. "Face Alignment Across Large Poses: A 3D Solution." 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) IEEE, 2016.

[15] Kazemi V, Sullivan J. One Millisecond Face Alignment with an Ensemble of Regression Trees [C]/ 2014 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2014.

[16] Nataniel Ruiz, Eunji Chong, and James M. Rehg. Fine-grained head pose estimation without keypoints. In Proceedings of International Conference on Computer Vision Workshops, 2017.

[17] Yang, Tsun-Yi and Chen, Yi-Ting and Lin, Yen-Yu and Chuang, Yung-Yu. FSA-Net: Learning Fine-Grained Structure Aggregation for Head Pose Estimation From a Single Image, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2019.

[18] Woo S, Park J, Lee J Y, et al. CBAM: Convolutional Block Attention Module [J]. 2018.