TokenHPE: Learning Orientation Tokens for Efficient Head Pose Estimation via Transformers

Cheng Zhang\textsuperscript{1}  Hai Liu\textsuperscript{1,*}  Yongjian Deng\textsuperscript{2,3}  Bochen Xie\textsuperscript{4}  Youfu Li\textsuperscript{4,*}

\textsuperscript{1}National Engineering Research Center for E-Learning, Central China Normal University, Wuhan, China
\textsuperscript{2}College of Computer Science, Beijing University of Technology, Beijing, China
\textsuperscript{3}Engineering Research Center of Intelligence Perception and Autonomous Control, Ministry of Education, Beijing, China
\textsuperscript{4}Department of Mechanical Engineering, City University of Hong Kong, Kowloon, Hong Kong, China

\texttt{zc2021@mails.ccnu.edu.cn, hailiu0204@ccnu.edu.cn, yjdeng@bjut.edu.cn, boxie4-c@my.cityu.edu.hk, meyfli@cityu.edu.hk}

Abstract

Head pose estimation (HPE) has been widely used in the fields of human machine interaction, self-driving, and attention estimation. However, existing methods cannot deal with extreme head pose randomness and serious occlusions. To address these challenges, we identify three cues from head images, namely, neighborhood similarities, significant facial changes, and critical minority relationships. To leverage the observed findings, we propose a novel critical minority relationship-aware method based on the Transformer architecture in which the facial part relationships can be learned. Specifically, we design several orientation tokens to explicitly encode the basic orientation regions. Meanwhile, a novel token guide multi-loss function is designed to guide the orientation tokens as they learn the desired regional similarities and relationships. We evaluate the proposed method on three challenging benchmark HPE datasets. Experiments show that our method achieves better performance compared with state-of-the-art methods. Our code is publicly available at \url{https://github.com/zc2023/TokenHPE}.

1. Introduction

Head pose estimation (HPE) is a popular research area in computer vision and has been widely applied to driver assistance [29], human–computer interaction [36], virtual reality [22], and attention detection [5]. In recent years, HPE has been actively studied and the accuracy has been considerably improved in terms of utilizing extra facial landmark information [2,20], extra RGB-depth information [13,26–28], extra temporal information [16], stage-wise regression strategy [42], multi-task learning [1,38], and alternative parameterization of orientation [3,15,18,19,24]. Currently, many methods focus on the representation of the head pose ori-
eral significant facial part changes are observed in specific orientations. For example, in Fig. 1, the two circled facial regions can be distinguished by a significant facial part change, which is the appearance of the right eye. Third, critical minority relationships of facial parts exist, and they can determine the orientation of a head pose despite possible occlusions. As Fig. 1 shows, if a person’s mouth is occluded, the head pose can be determined by the geometric spatial relationships of the eyes, nose, and the outline of the face. In these scenarios, the remaining minor facial parts and their relationships are crucial for high-accuracy HPE.

Given the aforementioned facial part relationships, the question is how to design a model that can utilize this heuristic knowledge. The traditional CNN architecture cannot easily learn these relationships. In contrast, the Transformer architecture can effectively address this drawback of CNN. Recently, Vision Transformer (ViT) [11] emerged as a new choice for various computer vision tasks. The Transformer architecture is known for its extraordinary ability to learn long-distance, high-level relationships between image patches. Therefore, using Transformer to learn the relationships among critical minority facial parts is reasonable. Moreover, the basic orientation regions can be well represented by learnable tokens in Transformer.

Inspired by the three findings and Transformer’s properties, we propose TokenHPE, a method that can discover and leverage facial part relationships and regional similarities via the Transformer architecture. The proposed method can discover facial part geometric relationships via self-attention among visual tokens, and the orientation tokens can encode the characteristics of the basic orientation regions. The latent relationships between visual and orientation tokens can be learned from large HPE datasets. Then, the learned information is encoded into the orientation tokens, which can be visualized by vector similarities. In addition, a special token guide multi-loss function is constructed to help the orientation token learn the general information. Our main contributions can be summarized as follows:

1. Three findings are derived on head images, including facial part relationships and neighborhood orientation similarities. Furthermore, to leverage our findings and cope with challenging scenarios, a novel token-learning model based on Transformer for HPE is presented for the first time.

2. We find that the head pose panoramic overview can be partitioned into several basic regions according to the orientation characteristics. The same number of learnable orientation tokens are utilized to encode this general information. Moreover, a novel token guide multi-loss function is designed to train the model.

3. We conduct experiments on three widely used HPE datasets. TokenHPE achieves state-of-the-art performance with a novel token-learning concept compared with its existing CNN-based counterparts. Abundant visualizations are also provided to illustrate the effectiveness of the proposed orientation tokens.

2. Related Work

2.1. Head Pose Estimation

Existing HPE methods can be roughly divided into three categories: (1) Euler angle regression approaches [9, 10, 31, 42] that regress the three Euler angles progressively, (2) extra information-utilized approaches [1, 7, 30, 38, 39] that exploit extra facial information to facilitate HPE, and (3) Alternative orientation parametrization approaches [3, 15, 18, 19] that substitute Euler angle representation with other representations.

**Euler angle regression approaches.** The paradigm in early studies was to consider HPE as a regression problem. CNNs have been adopted for HPE [31, 42] and remained dominant for many years because convolution can efficiently reveal the visual patterns on human faces. Ruiz et al. [31] applied CNN to HPE in an end-to-end manner to independently predict three Euler angles by using a multi-loss network. In [42], Yang et al. proposed FSA-Net, which reveals aggregated features with fine-grained spatial structures and progressive stage fusions. However, because of the incapacity of CNN to learn the relationships among visual patterns, further facial part relationships are not explored in this category.

**Extra information-utilized approaches.** With graph convolutional network (GCN) being leveraged in many NLP and computer vision tasks [8, 14, 21, 41, 45], Xin et al. [39] proposed a novel method that learns through the facial landmark graph. However, the precision of the model depends largely on the precision of the additional landmark detector. Wu et al. [38] proposed a multi-task model called SynergyNet that predicts complete 3D facial geometry. Improved performance is achieved by synergistic learning of 3D landmarks and 3D morphable model parameters. In these methods, facial part relationships can be learned from landmarks or other extra information. However, many manual annotations are required for training, which is laborious and inefficient.

**Alternative orientation parametrization approaches.** Most contributions to HPE in recent years have focused on alternative parametrization of head pose labels because traditional Euler angle labels inevitably have some problems at specific orientations. Geng et al. [15] proposed a multivariate label distribution as a substitute of Euler angles. In this manner, inaccurate manual annotation can be alleviated and the original label is softened, making the training easy. In [3], Cao et al. proposed a vector-based head pose representation that handles the issue of discontinuity of Euler angle annotation. Recently, Hempel et al. [18] proposed
a rotation matrix-based representation for HPE. The rotation matrix enables full pose regression without suffering from ambiguity problems. Although these methods have achieved impressive results, the intrinsic facial part relationships are still not fully exploited.

### 2.2. Vision Transformer

ViT [11] is a variant model of Transformer [34], which is originally used in NLP. In ViT, an input image is divided into patches that can be viewed as words. The success of ViT has led to its wide application in various vision tasks, including fine-grained classification [17, 25, 35], object detection [43], facial expression recognition [40], human pose estimation [23], and image segmentation [37]. Li et al. [23] proposed the use of learnable tokens to represent each human keypoint entity on the basis of prior knowledge. In this way, visual cue and constraint cue learning are explicitly incorporated through the Transformer architecture. In [6], Cordonnier et al. provided a theoretical explanation of the long-distance information learned in Transformer. Therefore, Transformer is capable to learn the facial part relationships, and neighborhood orientation similarities can be encoded into learnable orientation tokens.

### 3. Our Method

In this section, we first provide an overview of the proposed TokenHPE. Then, the details of the four parts of the model are elaborated. Lastly, we report the implementation details.

#### 3.1. Overview

An overview of our method is shown in Fig. 2. The TokenHPE model consists of four parts. The first part is visual token construction, where the input image is transformed into visual tokens through multiple approaches. The second part is orientation token construction. We provide two strategies to construct orientation tokens based on our finding on head image panoramic overview. The third part is the Transformer module, wherein the relationships of facial parts and orientation characteristics in the basic regions are learned by the self-attention mechanism. The fourth part is token learning-based prediction. A novel token guide multi-loss function is introduced to help the orientation tokens encode general information.

#### 3.2. Visual Token Construction

In this part, an original input RBG image is transformed into visual tokens. We provide three options to obtain the visual tokens: by patch division of the original image (Option 1), by extracting feature maps from a CNN (Option 2), or by directly selecting the tokens from a Transformer extractor, such as ViT [11] (Option 3). For Option 1, suppose we have an input image \( I \) with size \( H \times W \times C \). The image is divided into patches with patch size \( P_h \times P_w \). Then, each patch is resized into a 1D vector of size \( P_h \times P_w \times C \) and a linear projection is applied to obtain a visual token. This operation can be expressed as:

\[
 f : p \rightarrow v \in \mathbb{R}^d, \quad (1)
\]

where \( p \) refers to a 1D patch vector and \( v \) is a visual token.
Figure 3. Construction of orientation tokens. We discover that the head pose panoramic overview can be roughly divided into several basic orientation regions according to the neighbor image similarities. As the division granularity varies, the number of basic orientation regions also varies.

with a dimension of \( d \). For Option 2, the output of the CNN extractor is considered as a set of feature maps with a size of \( H \times W \times C' \). The remaining operations are similar to those in Option 1. For Option 3, the visual tokens are simply gained from the output of a Transformer.

Given that spatial relationships are essential for accurate HPE, positional embedding, \( \text{pos} \), is added to the visual tokens to reserve spatial relationships, which can be expressed as:

\[
[\text{visual}] = \{v_1 + \text{pos}, v_2 + \text{pos}, ..., v_n + \text{pos}\},
\]

where \( n \) is the number of patches. Then, we obtain \( n \) 1D vectors symbolically presented by \([\text{visual}]\) tokens.

### 3.3. Orientation Token Construction

**Basic orientation region partitioning.** We introduce two heuristic partitioning strategies, as shown in Fig. 3. In Strategy I, the panoramic overview is divided into nine basic orientation regions according to the appearance of the eyes and the overlapping of the nose and mouth. In Strategy II, the panoramic overview is divided into 11 regions, with a fine-grained partition in the yaw direction. A detailed description of the partition strategies is included in the supplementary material.

**Orientation token.** We prepend \( k \) learnable \( d \) dimensional vectors to represent \( k \) basic orientation regions. These vectors are symbolized as \([\text{dir}]\) tokens. The \([\text{dir}]\) tokens, together with the \([\text{visual}]\) tokens, are accepted as the input of Transformer. In the end, the processed \([\text{dir}]\) tokens are chosen as the output of Transformer.

### 3.4. Transformer Blocks and MLP Head

With the \([\text{visual}]\) and \([\text{dir}]\) tokens as the input, the Transformer blocks can learn the relationships between tokens. For each Transformer block, we adopt the classical structure (cf. [11, 23]), which can be briefly expressed as:

\[
\begin{align*}
\hat{X}^{l-1} &= \text{MSA}[\text{LN}(X^{l-1})] + X^{l-1}, \\
X^l &= \text{MLP}[\text{LN}(\hat{X}^{l-1})] + \hat{X}^{l-1},
\end{align*}
\]

where MSA denotes multi-head self-attention, MLP means multi-layer perception and LN is layernorm operation. We modify the MLP module by setting the \( \text{Tanh}() \) as the activation function. After the last Transformer layer, the \([\text{dir}]\) tokens are selected as the output of Transformer, whereas the \([\text{visual}]\) tokens are not used in the following steps. Therefore, the output of \( M \) Transformer blocks is denoted as \( \{X^M_1, X^M_2, ..., X^M_k\} \).

The orientation tokens need to be transformed to rotation matrices for training and prediction. We adopt similar transformation strategy as used in [18], which is formulated as:

\[
\hat{R}_i = F_{G\text{S}}(W_2X^M_i),
\]

where \( W \) is a projection matrix to obtain a 6D representation of head pose, and \( \hat{R}_i \) is the predicted rotation matrix of the \( i \)-th basic orientation region. \( F_{G\text{S}}() \) denotes the Gram–Schmidt process. For more details, please refer to the supplementary material.

A set of intermediate rotation matrices \( C = \{\hat{R}_1, \hat{R}_2, ..., \hat{R}_k\} \) can be generated by the transformation above. In order to obtain the final prediction rotation matrix, \( C \) is concatenated and flattened into a vector \( \hat{R} \in \mathbb{R}^{9 \times 6} \) as the input of the MLP head, which can be formulated as:

\[
\hat{R} = F_{G\text{S}}(W_2(tanh(W_1 \cdot \hat{R} + b_1)) + b_2),
\]

where \( W_i \) and \( b_i \) are the parameters of the MLP head. In the training stage, the intermediate rotation matrices and the final prediction rotation matrix are used for calculating the loss for back propagation while in the prediction stage, only the prediction rotation matrix \( \hat{R} \) is used for the model prediction.

### 3.5. Token Guide Multi-loss Function

The prediction of the proposed model is a rotation matrix representation denoted as \( \hat{R} \). Suppose that the groundtruth rotation matrix is \( R \). The geodesic distance [18] is used as the loss between two 3D rotations. The geodesic distance is formulated as:

\[
L_g(R, \hat{R}) = \cos^{-1}\left( \frac{\text{tr}(RR^T) - 1}{2} \right).
\]

**Orientation token loss.** Information can be encoded into the orientation tokens through the orientation token loss, which is defined as the geodesic distance with respect to their corresponding orientation regions. Therefore, the orientation token loss is written as:
where $k$ is the number of basic orientation regions, $R$ is the groundtruth rotation matrix, $\hat{R}_i$ is the predicted rotation matrix from the $i$-th region, and $\mathbb{I}(R, i)$ is an indicator function that determines if a ground truth head pose lies in the $i$-th basic region. $\mathbb{I}(R, i)$ can be expressed as:

$$\mathbb{I}(R, i) = \begin{cases} 1, & \text{if } R \text{ in region } i, \\ 0, & \text{if } R \text{ not in region } i. \end{cases}$$

**Prediction loss.** The predictions from the orientation tokens are aggregated to form the final prediction of our model. This is optimized by the prediction loss, which is formulated as:

$$Loss_{\text{pred}} = L_g(R, \hat{R}),$$

where $\hat{R}$ is the model prediction.

**Overall loss.** The overall loss consists of the orientation token loss and the prediction loss. It can be formulated as:

$$Loss_{\text{overall}} = \alpha Loss_{\text{pred}} + (1 - \alpha) Loss_{\text{ori}},$$

where $\alpha$ is a hyper-parameter that balances prediction loss and orientation token loss.

### 3.6. Architecture Details

The three options mentioned previously can be used to obtain the visual tokens. In Option 1, the raw image patches are directly transformed into visual tokens. In the version added with a CNN feature extractor is added to efficiently extract low-level features, we adopt the widely used stem-net, which quickly downsamples the feature map into 1/4 input resolution in a very shallow convolutional structure [4, 35]. Option 3 is applied in our TokenHPE model by default, in which the ViT-B/16 is set as the feature extractor for a tradeoff between model size and performance. The outputs of ViT are the visual tokens that can be directly used in the second part of the proposed model.

### 3.7. Implementation Details

**Pre-processing.** In our experiments, the image is resized into 240×240 pixels. A random crop is then applied to make the input image size 224×224 pixels. Our method is implemented with the Pytorch toolbox with one TITAN V GPU. All the parameters in our model are trained with random initialization.

**Training.** We train our TokenHPE in an end-to-end manner. The batch size is set to 64, and $\alpha$ is set to 0.6 by default. We train our model for 60 epochs. The learning rate is initialized as 0.00001, which is further decayed by a factor of 10 at the 20th and 40th epochs.

### 4. Experiments

This section describes the three datasets used for training and testing, the evaluation metrics, the experiment results and comparison with several methods, the ablation study, and the model visualization.

#### 4.1. Datasets and Evaluation Metrics

**Datasets.** *BIWI dataset* [12] includes 15,678 images of 20 individuals (6 females and 14 males, 4 individuals are recorded twice). The head pose range covers about ±75° yaw and ±60° pitch. *AFLW2000 dataset* [47] contains 2000 images and is typically used for the evaluation of 3D facial landmark detection models. The head poses are diverse and often difficult to be detected by a CNN-based face detector. *300W-LP dataset* [47] adopts the proposed face profiling to generate about 61k samples across large poses. The dataset is usually employed as the training set for HPE.

**Evaluation metric 1: Mean absolute errors of Euler angles (MAE).** MAE is a standard metric for HPE. Assume a given set of groundtruth Euler angles $\{\alpha, \beta, \gamma\}$ of an image, in which $\alpha$, $\beta$, and $\gamma$ represent pitch, yaw, and roll angle, respectively. The predicted set of Euler angles from a model is denoted as $\{\hat{\alpha}, \hat{\beta}, \hat{\gamma}\}$. Then, MAE is defined as:

$$MAE = \frac{1}{3}(|\alpha - \hat{\alpha}| + |\beta - \hat{\beta}| + |\gamma - \hat{\gamma}|).$$

We adopt MAE as an evaluation metric. However, because this metric is unreliable at extreme degrees, the MAE results are given at the same time for a more accurate measurement of the models.

**Evaluation metric 2: Mean absolute errors of vectors (MAEV).** MAEV is based on rotation matrix representation. For an image, suppose that the groundtruth rotation matrix is $R = [r_1, r_2, r_3]$, where $r_i$ is a 3D vector that indicates a spatial direction. The predicted rotation matrix from a model is denoted as $\hat{R} = [\hat{r}_1, \hat{r}_2, \hat{r}_3]$. MAEV can be formulated as:

$$MAEV = \frac{1}{3} \sum_{i=1}^{3} |r_i - \hat{r}_i|_1.$$
Table 1. Mean absolute errors of Euler angles and vectors on the AFLW2000 dataset. All methods are trained on the 300W-LP dataset. These methods take an RGB image as the input and can be trained free from extra annotations, such as landmarks.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Extra annotation free</th>
<th>Euler angle errors (°)</th>
<th>Vector errors</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Pitch</td>
<td>Yaw</td>
</tr>
<tr>
<td>3DDFA [47]</td>
<td>✗</td>
<td>27.05</td>
<td>4.71</td>
</tr>
<tr>
<td>FAN [2]</td>
<td>✗</td>
<td>12.3</td>
<td>6.36</td>
</tr>
<tr>
<td>EVA-GCN [39]</td>
<td>✗</td>
<td>5.34</td>
<td>4.46</td>
</tr>
<tr>
<td>SynergyNet [38]</td>
<td>✗</td>
<td>4.09</td>
<td>3.42</td>
</tr>
<tr>
<td>img2pose [1]</td>
<td>✗</td>
<td>5.03</td>
<td>3.43</td>
</tr>
<tr>
<td>HopeNet [31]</td>
<td>✔</td>
<td>7.12</td>
<td>5.31</td>
</tr>
<tr>
<td>FSA-Net [42]</td>
<td>✔</td>
<td>6.34</td>
<td>4.96</td>
</tr>
<tr>
<td>LwPosr [10]</td>
<td>✔</td>
<td>6.38</td>
<td>4.80</td>
</tr>
<tr>
<td>Quatnet [19]</td>
<td>✔</td>
<td>5.62</td>
<td>3.97</td>
</tr>
<tr>
<td>TokenHPE-v1 (ours)</td>
<td>✔</td>
<td>5.73</td>
<td>4.53</td>
</tr>
<tr>
<td>TokenHPE-v2 (ours)</td>
<td>✔</td>
<td>5.54</td>
<td>4.36</td>
</tr>
</tbody>
</table>

Table 2. Mean absolute errors of Euler angles and vectors on the BIWI dataset. All methods are trained on the 300W-LP dataset.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Extra annotation free</th>
<th>Euler angle errors (°)</th>
<th>Vector errors</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Pitch</td>
<td>Yaw</td>
</tr>
<tr>
<td>EVA-GCN [39]</td>
<td>✗</td>
<td>4.78</td>
<td>4.01</td>
</tr>
<tr>
<td>HopeNet [31]</td>
<td>✔</td>
<td>5.89</td>
<td>6.01</td>
</tr>
<tr>
<td>FSA-Net [42]</td>
<td>✔</td>
<td>5.21</td>
<td>4.56</td>
</tr>
<tr>
<td>Quatnet [19]</td>
<td>✔</td>
<td>5.49</td>
<td>4.01</td>
</tr>
<tr>
<td>TriNet [3]</td>
<td>✔</td>
<td>4.76</td>
<td>3.05</td>
</tr>
<tr>
<td>WHENet [46]</td>
<td>✔</td>
<td>4.39</td>
<td>3.99</td>
</tr>
<tr>
<td>6DRepNet [18]</td>
<td>✔</td>
<td>4.48</td>
<td>3.24</td>
</tr>
<tr>
<td>TokenHPE-v2 (ours)</td>
<td>✔</td>
<td>4.51</td>
<td>3.95</td>
</tr>
</tbody>
</table>

and test it on AFLW2000 and BIWI datasets. Tables 1 and 2 show the results of the first experiment. An extra column is added to indicate which methods are free from extra annotation for fair comparison. Results show that our method is on par with state-of-the-art methods on AFLW2000 dataset and achieves state-of-the-art results in MAEV on BIWI dataset. Among the compared methods, HopeNet [31] is normally considered the baseline of HPE. Compared with it (Table 1), our model achieves a 24.8% decrease in MAE and a 12.7% decrease in MAEV. TriNet [3] is a vector-based model, in which the head pose is represented by vectors. Its MAE is 0.69 lower than the baseline. A new MAEV metric is also introduced. We adopt this metric for our comparison. Compared with TriNet, our method obtains a slightly lower MAEV value, which indicates that our method is competitive to state-of-the-art methods. Some extra information-utilized methods (i.e., 3DDFA, Dlib, EVA-GCN, SynergyNet, img2pose) are also compared in Table 1. EVA-GCN [39] is a facial landmark graph-based method. A landmark detector is applied to the original image, and EVA-GCN takes the detected landmark graph as the input. The graph convolutional network learns the landmark relationships for HPE. Thus, the model result has an impressive improvement compared with the baseline. SynergyNet is a multitask model, and HPE is a subtask. The model is trained by synergistic learning. Therefore, abundant information, including 3DMM parameters and 3D landmarks, is utilized to enhance the performance. Compared to other methods that mainly based on CNN and its variants, our model is the only Transformer-based token learning method, thus has a stronger ability to learn the facial relationships and the orientation characteristics in the basic regions. Therefore, even on the challenging AFLW2000 dataset that has many difficult-to-predict images, our method still outperforms the majority of the other methods by a large margin. The excellent performance verifies the orientation learning capacity of our proposed TokenHPE.

**Experiment 2.** In our second experiment, we follow the protocol 2 in [42] for fair comparison. 70% of the videos in the BIWI dataset are used for training and the others for testing. Table 3 shows the results of our method compared with those of other state-of-the-art methods that follow the same training–testing protocol. Our method outperform all other methods by a large margin both on MAE and three Euler angles. Compared to 6DRepNet [18] that uses the rotation matrix representation with a CNN backbone, our TokenHPE can learn the general regional information and facial relationships through Transformer architecture, resulting in a 6.4% improve on MAE. The similar results on two experiments show that our method is robust and stable, and its impressive results do not depend on the training dataset but on the method itself.
Table 3. Mean absolute errors of Euler angles on the BIWI dataset. The dataset is split at a ratio of 7:3 for training and testing.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Pitch</th>
<th>Yaw</th>
<th>Roll</th>
<th>MAE</th>
<th>MAEV</th>
</tr>
</thead>
<tbody>
<tr>
<td>FSA-Net [42]</td>
<td>4.29</td>
<td>2.89</td>
<td>3.60</td>
<td>3.60</td>
<td></td>
</tr>
<tr>
<td>FDN [44]</td>
<td>3.98</td>
<td>3.00</td>
<td>2.88</td>
<td>3.29</td>
<td></td>
</tr>
<tr>
<td>Hopenet [31]</td>
<td>3.39</td>
<td>3.29</td>
<td>3.00</td>
<td>3.23</td>
<td></td>
</tr>
<tr>
<td>TriNet [3]</td>
<td>3.04</td>
<td>2.93</td>
<td>2.44</td>
<td>2.80</td>
<td></td>
</tr>
<tr>
<td>6DRepNet [18]</td>
<td>2.92</td>
<td>2.69</td>
<td>2.36</td>
<td>2.66</td>
<td></td>
</tr>
<tr>
<td>TokenHPE-v2 (ours)</td>
<td>3.01</td>
<td>2.28</td>
<td>2.01</td>
<td>2.49</td>
<td></td>
</tr>
</tbody>
</table>

Table 4. Effect of the feature extractor. The models are trained on the 300W-LP dataset and tested on the AFLW2000 dataset.

<table>
<thead>
<tr>
<th>Feature extractor</th>
<th>Pitch</th>
<th>Yaw</th>
<th>Roll</th>
<th>MAE</th>
<th>MAEV</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>6.07</td>
<td>4.96</td>
<td>5.11</td>
<td>5.38</td>
<td>6.65</td>
</tr>
<tr>
<td>CNN</td>
<td>4.68</td>
<td>5.71</td>
<td>4.48</td>
<td>4.96</td>
<td>6.04</td>
</tr>
<tr>
<td>ViT</td>
<td>5.54</td>
<td>4.36</td>
<td>4.08</td>
<td>4.66</td>
<td>5.98</td>
</tr>
</tbody>
</table>

4.3. Ablation Study

**Feature extractor.** The visual tokens are generated from the feature extractor. Therefore, the performance of the model partially depends on the feature extractor. We conduct experiments on different feature extractors to reveal the extent to which performance is affected by the feature extractor. As shown in Table 4, we test versions with two different feature extractors and a version without a feature extractor. The results show that the feature extractor improves performance to a specific extent compared with the version with no feature extractor. The ViT feature extractor has the best performance.

**Positional embedding.** Different from classification tasks, spatial relationships play an important role in HPE. Given that the self-attention operation is positionally invariant, normally, 2D sine positional embedding is added to reserve the spatial relationships for computer vision tasks. To illustrate the effect of positional embedding, we conduct experiments on our TokenHPE model with different positional embedding types (i.e., no positional embedding, learnable positional embedding, and 2D sine positional embedding). As shown in Table 5, the model with 2D sine positional embedding demonstrates the best performance. The learnable positional embedding version has a lower prediction accuracy and model without positional embedding performs the worst. Therefore, fixed positional embedding is important for a model to learn the facial part relationships. Meanwhile, the absence of positional embedding results in the loss of spatial geometric relationships between visual tokens.

**Effect of the token guide multi-loss function.** The proposed model is trained by a token guide multi-loss function. The hyper-parameter $\alpha$ in the multi-loss function controls the importance of the direction loss. When $\alpha$ is set to 1, the model learns the basic orientation regions by itself. As the value of $\alpha$ decreases, orientation token loss plays an increasingly important role in helping the model learn the directional information. The experimental results are shown in Fig. 4. When $\alpha$ decreases, MAE initially decreases then increases. The best result is obtained when $\alpha$ is set to 0.6. This situation indicates that the token guide loss indeed helps the model encode the basic orientation regions. As $\alpha$ decreases, the flexibility of the model is constrained, resulting in poor performance.

4.4. Visualization

In order to illustrate how the proposed TokenHPE explicitly utilizes orientation tokens to find the facial part relationships and orientation characteristics in the basic regions, we visualize the details during inference. We observe that our model exhibits similar behaviors for most common examples. Therefore, we randomly choose some samples from the AFLW 2000 dataset and visualize the details in Figs. 5 to 7.

**Heatmap visualization.** To confirm that our model can learn critical minority facial part relationships, we use Grad-CAM [32] to visualize the attention of a head pose prediction. Two representative methods (HopeNet and 6DRepNet) are adopted for a comparison with our proposed model. As Fig. 5 shows, our method can learn the crucial minority relationships of facial parts, such as the eyes, nose, and ears when the mouth is being occluded or the nose, mouth, and ears when the eyes are occluded by sunglasses. In these scenarios, the compared methods performed poorly when abundant facial information is missed. On Row 1 in Fig. 5, the attention heatmaps show that our method can find the critical minority relationships (nose, eyes, and ears). Row 2 indicates that our method can deduce the spatial location of the eyes to achieve accurate prediction compared with
Figure 5. Heatmap visualization of three models, namely, HopeNet (left), 6DRepNet (middle), and our proposed model (right). The red-color areas mean that the model provides high attention to these facial parts. We select three challenging scenarios in which the mouth (Row 1), the eyes (Row 2), and the right half of the face (Row 3) are missing.

Figure 6. Cosine similarity matrix between the learned orientation tokens. (a) Strategy I: nine basic orientation regions. (b) Strategy II: eleven basic orientation regions.

In summary, the heatmap visualization proves that our method can learn facial part relationships and can deduce the spatial relationships of facial parts.

Similarity matrix of orientation tokens. We visualize the cosine similarities of the orientation tokens. As shown in Fig. 6, the neighbor orientation tokens are highly similar. The orientation tokens that represent symmetric facial regions have higher similarity scores than the tokens that represent the other unrelated regions. Therefore, the results of the similarity matrix verify that the general information is learned by the orientation tokens.

Region information learnt by orientation tokens. The attention maps of orientation tokens are visualized in Fig. 7. In the first few layers, each orientation token pays attention to almost all the other ones to construct the global context. As the network deepens, each orientation token tends to rely on its neighbor region tokens and spatial symmetric tokens to yield the final prediction. As indicated in Fig. 7, at the deeper Transformer blocks, the attention score is higher between neighbor regions (the diagonal) and symmetric regions, such as regions 0 and 2, regions 3 and 5, and regions 6 and 8. In Fig 7, the attention score is higher in regions 3, 4, 6, and 7, indicating that the predicted head pose has more probability in the left–bottom direction, similar to the ground truth. Therefore, from the visualization shown in Fig. 7, we can conclude that our model has the ability to encode the general information of the basic regional orientation characteristics, including neighborhood similarities and symmetric properties.

5. Conclusion

In this work, we proposed a novel token-driven learning method for HPE called TokenHPE. We introduced three findings on head images, namely, neighborhood similarities, significant facial changes, and critical minority relationships. To leverage these properties of head images, we utilized the Transformer architecture to learn the facial part relationships and designed several orientation tokens according to panoramic overview partitions. Experimental results showed that TokenHPE can address the problem of ambiguity and occlusion in HPE and achieves a state-of-the-art performance compared with that of the existing method. In addition, the success of TokenHPE demonstrates the importance of orientation cues in the head pose estimation task, which was ignored by previous research. We hope this initial work can inspire further research on token learning methods for HPE.

Acknowledgement This work was supported in part by the National Key R&D Program of China under Grant 2021YFC3340802, in part by the National Natural Science Foundation of China under Grant (62277041, 62211530433, 62177018, 62173286, 62077020, 62005092, 62203024, 92167102).
References


