Research Article

CNN with Embedding Transformers for Person Reidentification

Mo Jianwen, 1 Mo Lunlin, 1 Yuan Hua, 1 Lin Leping, 1 and Chen Lingping 2

1School of Information and Communication, Guilin University of Electronic Technology, Guilin 541004, China
2Educational Technology Statistics, Guilin Institute of Information Technology, Guilin 541004, China

Correspondence should be addressed to Yuan Hua; yuanhua@guet.edu.cn

Received 7 August 2022; Revised 4 May 2023; Accepted 19 June 2023; Published 14 July 2023

Academic Editor: Michal Kawulok

Copyright © 2023 Mo Jianwen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. Introduction

Person reidentification (ReID) is a difficult problem in the field of computer vision and is often considered a retrieval problem as well. Given the pedestrian probe to be retrieved in the query set, we need to find the same pedestrian under different cameras in the gallery. Most previous works mainly tend to utilize convolutional neural network (CNN) combined with classification loss and metric loss [1] to complete the corresponding classification and recognition tasks. This global feature-based method is difficult to solve some problems such as pose, viewing angle, light intensity, occlusion, etc. (as shown in Figure 1). In order to make the model focus on more local features, some works [2–5] use the pose estimation network to obtain the key point information of the human body so as to use the key points of the human body to divide the parts and let the network focus on the learning of the part features. However, this method requires more graphic processing unit (GPU) resources. Other methods [6, 7] use the horizontal block slice method to extract the component features of pedestrians, but these methods all need to align the component features in order to improve the performance of the model. Although part-based convolutional baseline (PCB) [8] is not aligned, it directly divides the human body parts in the feature map equally and performs global average pooling (GAP) on each equally divided part, then uses fully connected (FC) and softmax to directly classify each pedestrian ID. But this approach will eventually lead to pixel-level feature cross-aliasing (as shown in Figure 2) [9]. Utilize dynamically matching local information (DMLI) to solve the alignment problem. Besides, Kalayeh et al. [10] utilize segmentation and channel attention mechanism to solve the occlusion problem.

Different from the abovementioned methods: horizontal slicing, component alignment, alignment algorithm, attention mechanism, etc., the convolutional network can pay more attention to and learn more robust features. But the CNN itself has a certain sliding window, which is also limited for shallow feature extraction, and leads to aliasing between subsequent features (such as in PCB). Because of transformer [11] structure based on self-attention mechanism is good at capturing correlation between shallow and deep features,
we introduced it into the field of person ReID. A CNN-based embedding transformer (CET) architecture is proposed. In this architecture, the embedded residual transformer (RT) structure is used to model the shallow features extracted by CNN so that each pixel-level feature (token) has global information in the shallow layer. That is conducive to the subsequent feature redirection. For the redirected feature map, the horizontal slice and GAP are used to obtain the features of each component. An external transformer structure with class token is further embedded to model the correlation between each component. By using transformer structure, more useful information is fused into the class token vector. In addition, this paper also adds a feature fuse with learnable vector (FFLV) structure to further strengthen the learning of features.
2. Related Works

2.1. Deep Person ReID. In recent years, due to the development of deep networks, deep learning-based person ReID methods achieved surprising performance [14–16]. The methods can be roughly divided into two categories: representation learning and metric learning. Representation learning positions the problem of person ReID as a classification problem and uses softmax and cross-entropy loss (ID loss) to train in the network. For example, the ID embedding network [17, 18] takes the pedestrian ID as a classification problem, and how many pedestrians in the dataset correspond to how many classification categories [19, 20]. Combined validation loss and ID loss, which achieved better performance on CNN deep networks.

Metric learning methods [15, 21–27] first extract features from raw images by using a deep network and then calculate the distance between features to measure their similarity. Triplet losses [15, 21] are used to shorten the distance of positive sample pairs while pushing the distance of negative sample pairs. In this way, different pedestrians can be distinguished [26, 27]. Combined softmax loss for representation learning and metric learning loss to accelerate model convergence.

2.2. Local Feature Alignment. In recent years, more and more works [2–5, 28–31] have begun to focus on the learning of local features. Besides, the pedestrian’s body pose is also an important clue for person ReID tasks. For example, Zheng et al. [2] align the pedestrian’s standard posture by pose invariant embedding, thereby mitigating the influence of the pose bias. Global–local alignment descriptor [4] utilizes human pose key points to extract features of specific regions of the human body. SpindleNet [2] uses the region part generation network to generate the regions of various parts of the human body and combine the corresponding regions in the feature map to extract features. However, these methods all require additional posture annotation information to assist, which not only increases labor costs but also may introduce wrong posture annotation information.

On the other hand, some works [6–8, 30, 32, 33] enhance the performance of the model by horizontally partitioning and extracting the features of each block separately. Li et al. [6] and Cheng et al. [7] directly align the parts under the premise of pose alignment, which turns out to be an effective method as well.

2.3. Transformer and Person ReID. The attention mechanism focuses on important features while ignoring irrelevant features. The introduction of the attention mechanism can well solve problems such as occlusion and misalignment in person ReID, because the attention mechanism will assign weights to each pedestrian component automatically. Occlusion parts are assigned low weights, while similar weights are learned for identical human parts that are not aligned. Methods based on CNN have dominated the ReID field for many years. However, some pure Transformer methods, as well as hybrid CNN-transformation models, have also become popular in ReID recently.

In 2017, transformer [11] began to emerge in the field of natural language processing (NLP) and continued to develop in the follow-up. NLP models such as generative pre-trained transformer [34] and bidirectional encoder representation from transformers [35] came out one after another, which proved that the transformer’s powerful representation ability for the correlation between sequence tokens. Based on this, Google’s ViT [36] also successfully introduced the transformer structure into the image field to model the correlation between image patches. The results show that the transformer’s performance in the visual field is still comparable to that of CNN. TransReID [37], as the first method to apply the ViT architecture to the field of person ReID, has achieved good performance in both vehicle ReID and person ReID. AAformer [38] also applies the ViT backbone network and additional part tokens (PT) to characterize and aggregate pedestrian part information. LA-transformer [39] replaces the PCB backbone ResNet-50 with transformer and fuses the global token into each human body part. In addition, some other works [40–42] combine transformer and CNN so that some hierarchical features and local features are aligned.

3. Proposed Method


3.1.1. Self-Attention. Might as well, we set $T_{IN} = \{t_1, t_2, \ldots, t_n\}$. Each vector $t_i$ in $T_{IN}$ is multiplied by a set of three learnable matrices $W_q, W_k, W_v$ to obtain three output matrices $Q, K, V$. Then, for each $q_i \ (i = 1, \ldots, n)$ in $Q$ and $k_1 \sim k_n$ in $K$, we calculate the self-attention dot product. In order to prevent the dot product from too large, which would cause problems with subsequent gradients, a scaling factor $\sqrt{d}$ is added here, in which $d$ is the dimension of $q_i$ or $k_i$. In Summary, the encoded output vector of this group can be represented by the following formula:
$o_i = \frac{\text{softmax}(q_i \cdot k_1) \times v_1 + \text{softmax}(q_i \cdot k_2) \times v_2 + \ldots + \text{softmax}(q_i \cdot k_n) \times v_n}{\sqrt{d}} \quad (1)$

Similarly, all other outputs $o_k[k+i]$ can be obtained by performing the same weighted sum operation on other self-attention. The previous process can be further abstracted and written as the following matrix representation:

$$O_{OUT} = \text{Attention}(Q, K, V) = \text{softmax} \left( \frac{QK^T}{\sqrt{d_k}} \right) V \quad (2)$$

### 3.1.2. Multihead Self-Attention

If only a set of $W_q, W_k, W_v$ are used, the attention output is inevitably too single to adapt to the practical and complex application environment. In different environments and different tasks, the model needs to be able to notice the correlation between different parts. Therefore, it is necessary to set up multiple sets of $W_q, W_k, W_v$ matrices to represent complex actual situations, and the multihead attention mechanism can be represented by the following formula:

$$Z_{OUT} = \text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^z \quad (3)$$

Among them, $Z_{OUT}$ is the output matrix of multihead attention, which is the same dimension as the input matrix.

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \quad (4)$$

In a word, the previous $h$ groups of $W_q, W_k, W_v$ matrices will obtain $h$ group of outputs respectively. We perform concatenation on the output matrixes of the $h$ groups and then multiply it by a matrix of $W^z$, so as to obtain $Z_{OUT}$, the encoded output of the same dimension as the input $T_{IN}$.

### 3.1.3. Multiple Layer Perceptron (MLP), Residual Connection, and LayerNorm

Just like residual connection was introduced to solve the gradient disappearance problem caused by too deep network in CNN, residual connection was introduced in every transformer block. In addition, transformer also consists of simple MLP and LayerNorm, as shown in Figure 3(b).

### 3.1.4. Position Embedding

One-dimensional position encoding can be obtained in two ways: (1) autonomous learning; (2) obtained by the following expression:

$$e_{\text{pos}, (2i)} = \sin \left( \frac{\text{pos}}{10000^{2i}} \right). \quad (5)$$

---

**FIGURE 3:** Architecture of CET. First of all, the input image $X$ passes through TIC to get the feature map $F$, and $F$ goes through PGAP to get the component feature $P_t$, which then passes through a linear projection. Then, the input of transformer encoder $Z$ is obtained by embedding global vector GTE and position encoding PE. The output of transformer encoder $O$ is obtained by $Z$. Finally, $O$ is divided into two branches structure called FFLV and TBL, respectively, which are used to fuse the two losses. (a) CNN with embedding transformers (CET architecture); (b) transformer encoder.
Among them, $d$ is the dimension of input vectors; $\text{pos}(1, 2, \ldots, n)$ indicates the position of the corresponding vector in the component block, while $i (i = 0, 1, 2, \ldots, n)$ represents the specific dimension.

### 3.2. Model Overview

The diagram of CET is shown in Figure 3, and it can use arbitrary CNN as the backbone network. In CET, we embed a new feature extractor transformer in CNN (TIC) by combining ResNet-50 and Transformer. First, the input image is passed through TIC to obtain a feature map with global information. Second, the feature map is subjected to horizontal slice; then, we perform part-based global average pooling (PGAP) to obtain the block component features $\mathbf{P}_t$. After $\mathbf{P}_t$, a linear projection block is added. Then, by adding the global classification feature vector called global token embedding (GTE) vector and position encoding (PE), we obtain $\mathbf{Z}$. Then, $\mathbf{Z}$ will be put into transformer encoder, so as to get the output $\mathbf{O}$. Finally, we design a FFLV block, which uses matrix multiplication to weight the output encoding vector to obtain the final one-way classification token vector $\mathbf{f}$. Another way of classification token vector is taken from the global token vector $\mathbf{O}_0$ encoded by the transformer encoder. The two-way vectors pass through TBL block to classify the pedestrian.

### 3.3. Structure of TIC

In TIC (as shown in Figure 4), in order to allow the CNN backbone network to extract features with global information in the shallow layer, we embed RT module, which uses the embedded transformer to model the correlation between global information, so as to alleviate the subsequent feature aliasing phenomenon.

By letting the input pedestrian image be $X (X \in \mathbb{R}^{H \times W \times 3})$, and considering the size of the feature map, we only embed an RT between conv2_x and conv3_x, conv3_x and conv4_x in ResNet-50, respectively. Before embedding the first RT structure, we split the CNN backbone (ResNet-50) into two parts: CNNprelayer (conv1 and conv2_x) and CNNnextlayer (conv3_x, conv4_x and conv5_x). First, we use the following formula to get the input feature map of RT $\mathbf{G}_\text{in} (\mathbf{G}_\text{in} \in \mathbb{R}^{H_t \times W_t \times C_t})$.

$$\mathbf{G}_\text{in} = \text{CNNprelayer}(X).$$

Then, $\mathbf{G}_\text{in}$ pass through Tokenizer in RT to get the input of Transformer1 $\mathbf{T}_\text{in} (\mathbf{T}_\text{in} \in \mathbb{R}^{L \times C_t})$. Tokenizer can be given by the following formula:

$$\mathbf{T}_\text{in} = \text{SOFTMAX}_{W_\text{t}} (\mathbf{G}_\text{in}W_A)^T \mathbf{G}_\text{in}.$$  

$$\mathbf{T}_\text{in} = \mathbf{T}_\text{t} + \mathbf{T}_\text{pre},$$

where $\mathbf{T}_\text{pre} \cdot \mathbf{T}_\text{t} \in \mathbb{R}^{L \times C_t}$, $W_A \in \mathbb{R}^{C_t \times L}$. Random initialization is used in the first $\mathbf{T}_\text{pre}$ in RT. Then, $\mathbf{T}_\text{in}$ pass through Transformer1:

$$\mathbf{T}_\text{out} = \text{Transformer1}(\mathbf{T}_\text{in}).$$

In previous formula, we get $\mathbf{T}_\text{out} (\mathbf{T}_\text{out} \in \mathbb{R}^{L \times C_t})$. The reconstructed features $\mathbf{G}_\text{out} (\mathbf{G}_\text{out} \in \mathbb{R}^{H_t \times W_t \times C_t})$ are obtained through Projector, which is given by the following:

$$e_{(\text{pos}, 2i+1)} = \cos\left(\frac{\text{pos}}{10000^{2i}}\right).$$
\[ G_{\text{out}} = G_{\text{in}} + \text{SOFTMAX}_l \left( \left( G_{\text{in}} W_Q \right) (T_{\text{out}} W_K) ^T \right) T_{\text{out}}. \]  

(11)

where \( W_Q, W_K \in \mathbb{R}^{C \times H \times W}, G_{\text{out}} \in \mathbb{R}^{H \times W \times C} \), \( G_{\text{out}} \) is the output reconstructed feature map.

In order to capture the global information on the shallow feature map better, we add double skip connections in the RT, which are previous Equations (9) and (11), respectively. Similarly, a second RT can be embedded between conv4_x and conv5_x of ResNet-50. Finally, after passing through CNNnextlayer (conv4_x and conv5_x) and PGAP, the output of TIC \( P_T \) is obtained.

\[ P_T = \text{PGAP}(\text{CNNnextlayer}(G_{\text{out}})). \]  

(12)

Among them, \( P_T \in \mathbb{R}^{n \times C} \).

3.4. FFLV and TBL. Formula (12) uses PGAP to horizontally slice the feature map, and after linear projection, we get \( P_T (P_T \in \mathbb{R}^{n \times C}) \). Let \( P_T = \{ p_1, p_2, p_3, p_4, p_5, \ldots, p_n \} \), then we add the global classtoken(\( \text{cls} \)) classification vector for feature fusion in subsequent transformer encoder module.

\[ P = P_T \oplus \text{cls} = \{ p_1, p_2, p_3, p_4, p_5, \ldots, p_n, \text{cls} \}. \]  

(13)

In order not to lose the spatial position relationship of local features, one-dimensional position encoding vector \( PE \) is added to the previous \( P \). The specific formula is as follows:

\[ Z = P + PE = \{ p_1 + e_1, p_2 + e_2, \ldots, p_n + e_n, \text{cls} + e_{n+1} \}. \]  

(14)

Then, the previous \( Z \) is input to the transformer encoder module to get the encoded output:

\[ O = \text{Transformer}(Z). \]  

Finally, the encoder learns an output class token vector \( O_0 (O_0 \in \mathbb{R}^{C \times 1}) \) that combines the key features of \( p_1 \sim p_n \). \( O_0 \) pass through the FC layer to obtain a vector whose dimension \( \text{clsNum} \) is the number of pedestrian categories in the training set, and then we use softmax and cross-entropy to obtain a loss:

\[ L_o = \sum_{i=1}^{\text{clsNum}} -q_i \log(\text{Softmax}(\text{FC}(O_0)))_. \]  

(15)

where \( \text{Softmax}(\text{FC}(O_0))_. \) is our predicted value for pedestrian \( i \), \( q_i \) is the true value. Therefore, the total loss function can be formulated as follows:

\[ L = \lambda L_o + (1 - \lambda) L_f. \]  

(18)

Among them, \( \lambda \) is the balance parameter, which is used to balance the two loss ratios.

4. Experiment

4.1. Dataset. In this section, we briefly introduce the datasets used to evaluate the model (as shown in Table 1).

Since Market1501 and DukeMTMC-reID are two well-known datasets in person ReID, these two datasets are mainly selected for ablation experiments. In addition, we also do some simple performance experiments on CUHK03.

Market1501 contains 32,217 images and 1,501 pedestrian labels, as well as six camera views. There are 751 pedestrian identities for the training set and 750 pedestrian identities for the test set. In total, DukeMTMC-ReID collected 36,441 pedestrian images, including 1,812 pedestrian identities captured by eight cameras. Among them, 702 pedestrian identities were used as the training set, and 1,110 pedestrian identities were used as the test set. CUHK03 contains 7,264 images of 1,467 pedestrian labels. Among them, 767 pedestrian IDs were used as training sets, and 700 IDs were used as test sets. For the previous data sets, mAP and Rank1 were taken as the main evaluation indexes in this experiment.

4.2. Implementation Details. In terms of implementation, for the shallow layer of the ResNet-50 network, a transformer-based TIC feature extractor is embedded according to different feature map dimensions, which contains a residual structure of double skip connections (as shown in Figure 4). Therefore, the network in the shallow layer has a global information, which is convenient for subsequent feature redirection. The rest of the residual structure parameter settings are consistent with the PCB. More details are shown in Table 2.

In the feature map with an output size of \( 24 \times 12 \times C \), we use PGAP (six blocks with a size of \( (24/6) \times 12 \times C \)) in the comparison experiment with PCB [8]; eight blocks with a

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Identities</th>
<th>Cameras</th>
<th>Images</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Market-1501</td>
<td>1,501</td>
<td>6</td>
<td>32,217</td>
<td>128×64</td>
</tr>
<tr>
<td>CUHK03</td>
<td>1,467</td>
<td>10</td>
<td>13,164</td>
<td>Vary</td>
</tr>
<tr>
<td>DukeMTMC-reID</td>
<td>1,812</td>
<td>8</td>
<td>36,441</td>
<td>Vary</td>
</tr>
</tbody>
</table>

Table 1: Overview of datasets.
5. Experimental Results

5.1. Convergence Speed. As shown in Figure 5, we compare the convergence speed of PCB, refined part pooling (RPP), and CET model under the same learning rate and other super-parameters in Market-1501 dataset. From the results, we can see that CET model converges faster than that of PCB and RPP. The main reason that the convergence speed of CET is slightly faster than that of PCB and RPP is that the TIC feature extractor is embedded in the network. The internal Transformer structure in TIC is forwarded every time, and all pixel tokens can be covered in the calculation process. So, TIC can help our network to extract global information at a shallow level and transfer it to the next layer through double residual connections (as shown in Figure 4). Therefore, the feature learning is more efficient, and the convergence speed is naturally faster.

Self-attention in transformer needs to calculate the dot product between the tokens in the feature map. When calculating the self-attention of the dot product, it will be slower than CNN in terms of calculation time. However, since we will map the tokenizer before sending the feature map to the transformer, the calculation speed is improved compared to directly sending the pixel-level token to the transformer.

5.2. Feature Redirection Results. In PCB, we know the phenomenon of cross-aliasing between block features, which is a problem left over by using the idea of slice and block method. For this phenomenon, an RPP method which is used to effectively alleviate the aliasing phenomenon is proposed in the original paper of PCB [8]. In theory, this aliasing phenomenon cannot be completely eliminated but can only be alleviated. In response to this problem, we use TIC’s global grasp of shallow features to indirectly redirect features. After the network converges, we calculate the similarity between

<table>
<thead>
<tr>
<th>Layer name</th>
<th>Output size</th>
<th>50-layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>conv1</td>
<td>192 × 96</td>
<td>7 × 7, 64, stride 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 × 3, max pool, stride 2</td>
</tr>
<tr>
<td>conv2_x</td>
<td>96 × 48</td>
<td>[1 × 1, 64] × 3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[3 × 3, 64] × 3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[1 × 1, 256] × 3</td>
</tr>
<tr>
<td>RT</td>
<td>96 × 48</td>
<td>Block x1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[1 × 1, 128] × 4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[3 × 3, 128] × 4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[1 × 1, 512] × 4</td>
</tr>
<tr>
<td>conv3_x</td>
<td>96 × 48</td>
<td>Block x2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[1 × 1, 128] × 6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[3 × 3, 128] × 6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[1 × 1, 512] × 6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[1 × 1, 1024] × 6</td>
</tr>
<tr>
<td>conv4_x</td>
<td>48 × 24</td>
<td>[1 × 1, 256] × 3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[3 × 3, 256] × 3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[1 × 1, 1024] × 3</td>
</tr>
<tr>
<td>conv5_x</td>
<td>24 × 12</td>
<td>[1 × 1, 512] × 3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[3 × 3, 512] × 3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[1 × 1, 2048] × 3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PGAP, 751-d fc, softmax</td>
</tr>
</tbody>
</table>

Figure 5: Comparison of convergence speed under the same learning rate in Market-1501.
the redirected features and the final pooled features of each component. The similarity results are used to rearrange the features at each pixel level to obtain their own aliasing graphs. The aliasing graphs of PCB and RPP are calculated using the same method and compared with our method. We found that the aliasing phenomenon is effectively alleviated, and our experimental results on Market-1501 are shown in Figure 2.

Compared with PCB and RPP, the main reason why CET can further alleviate the aliasing phenomenon is that the transformer structure inside TIC extracts global information in the shallow layer of the network. These global information are helpful for subsequent feature redirection.

5.3. Number of Blocks. In the slicing method of person ReID, the optimal number of slices in PCB and PCB + RPP is six, while in AlignedReID [9] DMLI algorithm (a shortest path algorithm) increased the optimal number of slices to eight. Since Transformer is more suitable for fine-grained information extraction and fusion, CET directly divides the tensor of $24 \times 12 \times C$ into 4, 6, 8, 12, and 24 blocks along the height direction. Then we draw its mAP and Rank1 change curve with Market-1501 dataset, respectively, and compare it with PCB and RPP (as shown in Figure 6). We found that the mAP and Rank1 of CET showed an overall upward trend with the increase in the number of slice blocks. When the number of blocks reached 12, the performance of the model could be the same as that of PCB, and when the number of blocks reached 24, it was higher than that of PCB + RPP 0.2-0.3 percentage points. So, for a feature map of $24 \times 12 \times C$, our optimal number of blocks is the maximum number of horizontal slices, 24.

From the results, transformer is indeed better at capturing the correlation between fine-grained features. In CNN, the model performance decreases when the number of slices increases to a certain number. It may be because CNN is not good at learning the correlation between finer-grained features in the deep layers of the network. But training a transformer model does not require very large datasets. Although transformer is good at learning the correlation between more fine-grained features, it also requires a large dataset. This paper combines CNN with transformer to make an effective balance. Therefore, we combine CNN with transformer to horizontally slice and block the $24 \times 12 \times C$ tensor to model the correlation between more fine-grained block features and achieve competitive results.

5.4. Correlation of Each Part. Generally speaking, if the parts with relatively high correlation are the same body part, it means that the parts are already aligned. For example, we conduct an experiment on Market-1501. As shown in Figure 7, part 1 and part 3 are relatively similar and both are heads. When the model converges, we demonstrate the correlation of tokens between two images that have been sliced to eight parts of the same pedestrian (as shown in Figure 7) and find that CET can indeed capture the correlation between the same human body part. Compared to directly applying some alignment algorithms (DMLI in Alignedreid [9]), CET can automatically capture this similarity and thus also achieve the alignment effect indirectly.

5.5. Ablation Experiment. In the ablation experiment, we compared the grad-cam heatmap corresponding to the PCB with transformer encoder and without transformer encoder in the dataset of Market-1501. The main method is to calculate the corresponding grad-cam heatmap of the global classification token vector output by the transformer encoder (the $O_0$ or f vector of the TBL module in Figure 3), and at the same time do the backward calculation and calculate the corresponding grad-cam heatmap in a certain block of PCB. The comparison results of these two heatmaps are shown in Figure 8.

It can be seen from the figure that a certain classification token in the PCB only pays attention to a certain part, while our model has a stronger ability to control the overall...
situation. So the global classification token vector output by our transformer encoder contains more global features.

Besides, we conduct another ablation study of CET with different kinds of parts RT, FFLV, and TBL in Market-1501 and DukeMTMC datasets. In this ablation study, it can be found that each part of CET has a certain contribution to the improvement of the network. Finally, CET achieves its best performance when all components are added to the network just, as shown in Table 3.

![Figure 7: Correlation between part tokens in Market-1501.](image)

![Figure 8: Comparison grad-cam with point-specific between PCB (left) and CET (right).](image)

<table>
<thead>
<tr>
<th>Ablation parts</th>
<th>Market-1501</th>
<th>DukeMTMC</th>
</tr>
</thead>
<tbody>
<tr>
<td>CET (ResNet50)</td>
<td>90.3</td>
<td>80.3</td>
</tr>
<tr>
<td>CET (RT)</td>
<td>91.5</td>
<td>81.0</td>
</tr>
<tr>
<td>CET (RT + FFLV)</td>
<td>93.1</td>
<td>82.4</td>
</tr>
<tr>
<td>CET (RT + FFLV + TBL)</td>
<td>94.0</td>
<td>83.2</td>
</tr>
</tbody>
</table>
5.6. Compare with SOTA. Before comparing related indicators, it is necessary to make a brief introduction to them. Commonly used evaluation index—Rank-k (R-k): In the sorted list returned by the algorithm, if there is a search target in the top k positions, it is called a rank-k hit. mAP: It can reflect the degree to which all the correct pictures of the retrieved pedestrians in the database are ranked in front of the sorted list and can more comprehensively measure the performance of the ReID algorithm.

Compared with several previous state-of-the-art methods, our method achieves competitive results on several datasets. Compared with PCB and RPP on Market-1501, we achieved 94.0% on Rank1, an increase of 0.2%, and on mAP, an increase of 0.3% from 81.6% to 81.9%. Table 4 shows the proposed CET net achieves the best rank-1 and mAP performance under single-query settings.

To verify the statistical significance of the performance of our model, we also executed a Wilcoxon sign rank test. We randomly select two sets of pictures of 50 or more different pedestrians from more than 1,500 pedestrians in the market-1501 dataset and then input them into the trained network model to retrieve the real two sets of rank-k y1 and y2 as stats. The two sets of input y1 and y2 of stats.wilcoxon in Python lib are calculated to obtain a corresponding p-value of about 0.021, which shows that our result is the same probability distribution, not accidental, and is in line with statistical significance. We mainly perform it on the Market-1501 dataset, which shows that the accuracy and mAP rate were statistically significant at a significant level of 2%. In other words, our fault tolerance rate is as high as about 98%, which is not an accidental result.

On DukeMTMC-reID dataset, we achieved 83.2% and 70.2% for Rank1 and mAP, respectively. Although it is slightly inferior to Rank 1, our mAP is also little better than PCB + RPP. Table 5 shows details of this comparison.

On CUHK03 dataset, our Rank1 and mAP are both improved to 64.0% and 60.8%, higher than PCB and AlignedReID. More detail can be seen in Table 6. All the obtained results show that our model is effective for the horizontal block slicing method in person ReID.

6. Conclusions and Outlook

In this work, we propose a CET architecture for person ReID, which can capture the global information of the feature by using the transformer structure embedded in the TIC in the shallow layer of the network. It can improve the cross-aliasing phenomenon existing in the slice and block methods like PCB and AlignedReID. In addition, our CET architecture can use the self-attention mechanism in transformer to indirectly and automatically align the sliced component features without using some alignment or matching algorithms. Moreover, our TIC structure can be easily embedded into the CNN network, and it is not complicated to implement. In future work, we will try to slice vertically or combine horizontal slices with Transformer to model more fine-grained features.

Data Availability

Data are available at: http://robustsystems.coe.neu.edu/sites/robustsystems.coe.neu.edu/files/systems/projectpages/reiddataset.html.

Conflicts of Interest

The authors declare that they have no conflict of interest.

Acknowledgments

This research was funded by the National Natural Science Foundation of China (grant nos. 62001133, 62177012, and 61967005), Innovation Project of GUET Graduate Education (grant no. 2021YCX027).

References


