TransReID: Transformer-based Object Re-Identification

Shuting He\textsuperscript{1,2}\textsuperscript{*} Hao Luo\textsuperscript{2} Pichao Wang\textsuperscript{2} Fan Wang\textsuperscript{2} Hao Li\textsuperscript{2} Wei Jiang\textsuperscript{1†}
\textsuperscript{1}Zhejiang University \textsuperscript{2}Alibaba Group

\{shuting.he,jiangwei_zju\}@zju.edu.cn \{michuan.lh,pichao.wang,fan.w,lihao.lh\}@alibaba-inc.com

Abstract

Extracting robust feature representation is one of the key challenges in object re-identification (ReID). Although convolution neural network (CNN)-based methods have achieved great success, they only process one local neighborhood at a time and suffer from information loss on details caused by convolution and downsampling operators (e.g. pooling and strided convolution). To overcome these limitations, we propose a pure transformer-based object ReID framework named TransReID. Specifically, we first encode an image as a sequence of patches and build a transformer-based strong baseline with a few critical improvements, which achieves competitive results on several ReID benchmarks with CNN-based methods. To further enhance the robust feature learning in the context of transformers, two novel modules are carefully designed. (i) The jigsaw patch module (JPM) is proposed to rearrange the patch embeddings via shift and patch shuffle operations which generates robust features with improved discrimination ability and more diversified coverage. (ii) The side information embeddings (SIE) is introduced to mitigate feature bias towards camera/view variations by plugging in learnable embeddings to incorporate these non-visual clues. To the best of our knowledge, this is the first work to adopt a pure transformer for ReID research. Experimental results of TransReID are superior promising, which achieve state-of-the-art performance on both person and vehicle ReID benchmarks. Code is available at https://github.com/heshuting555/TransReID.

1. Introduction

Object re-identification (ReID) aims to associate a particular object across different scenes and camera views, such as in the applications of person ReID and vehicle ReID. Extracting robust and discriminative features is a crucial component of ReID, and has been dominated by CNN-based methods for a long time \cite{19, 37, 36, 44, 42, 5, 12, 13, 53, 15}.

By reviewing CNN-based methods, we find two important issues which are not well addressed in the field of object ReID. (1) Exploiting the rich structural patterns in a global scope is crucial for object ReID \cite{54}. However, CNN-based methods mainly focus on small discriminative regions due to a Gaussian distribution of effective receptive fields \cite{29}. Recently, attention modules \cite{54, 6, 3, 21, 1} have been introduced to explore long-range dependencies \cite{45}, but most of them are embedded in the deep layers and do not solve the principle problem of CNN. Thus, attention-based methods still prefer large continuous areas and are hard to extract multiple diversified discriminative parts (see Figure 1). (2) Fine-grained features with detail information are also important. However, the downsampling operators (e.g. pooling and strided convolution) of CNN reduce spatial resolution of output feature maps, which greatly affect the discrimination ability to distinguish objects with similar appearances \cite{37, 27}. As shown in Figure 2, the details of the backpack are lost in CNN-based feature maps, making it difficult to differentiate the two people.
Recently, Vision Transformer (ViT) [8] and Data-efficient image Transformers (DeiT) [40] have shown that pure transformers can be as effective as CNN-based methods on feature extraction for image recognition. With the introduction of multi-head attention modules and the removal of convolution and downsampling operators, transformer-based models are suitable to solve the aforementioned problems in CNN-based ReID for the following reasons. (1) The multi-head self-attention captures long range dependencies and drives the model to attend diverse human-body parts than CNN models (e.g. thighs, shoulders, waist in Figure 1). (2) Without downsampling operators, transformer can keep more detailed information. For example, one can observe that the difference on feature maps around backpacks (marked by red boxes in Figure 2) can help the model easily differentiate the two people. These advantages motivate us to introduce pure transformers in the object ReID.

Despite its great advantages as discussed above, transformers still need to be designed specifically for object ReID to tackle the unique challenges, such as the large variations (e.g. occlusions, diversity of poses, camera perspective) in images. Substantial efforts have been devoted to alleviating this challenge in CNN-based methods. Among them, local part features [37, 44, 20, 48, 28] and side information (such as cameras and viewpoints) [7, 61, 35, 30], have been proven to be essential and effective to enhance the feature robustness. Learning part/stripe aggregated features makes it robust against occlusions and misalignments [49]. However, extending the rigid stripe part methods from CNN-based methods to pure transformer-based methods may damage long-range dependencies due to global sequences splitting into several isolated subsequences. In addition, taking side information into consideration, such as camera and viewpoint-specific information, an invariant feature space can be constructed to diminish bias brought by side information variations. However, the complex designs for side information built on CNN, if directly applied to transformers, cannot make full use of the inherent encoding capabilities of transformers. As a result, specific designed modules are inevitable and essential for a pure transformer to successfully handle these challenges.

Therefore, we propose a new object ReID framework dubbed TransReID to learn robust feature representations. Firstly, by making several critical adaptations, we construct a strong baseline framework based on a pure transformer. Secondly, in order to expand long-range dependencies and enhance feature robustness, we propose a jigsaw patches module (JPM) by rearranging the patch embeddings via shift and shuffle operations and re-grouping them for further feature learning. The JPM is employed on the last layer of the model to extract robust features in parallel with the global branch which does not include this special operation. Hence, the network tends to extract perturbation-invariant and robust features with global context. Thirdly, to further enhance the learning of robust features, a side information embedding (SIE) is introduced. Instead of the special and complex designs in CNN-based methods for utilizing these non-visual clues, we propose a unified framework that effectively incorporates non-visual clues through learnable embeddings to alleviate the data bias brought by cameras or viewpoints. Taking cameras for example, the proposed SIE helps address the vast pairwise similarity discrepancy between inter-camera and intra-camera matching (see Figure 6). SIE can also be easily extended to include any non-visual clues other than the ones we have demonstrated.

To our best knowledge, we are the first to investigate the application of pure transformers in the field of object ReID. The contributions of the paper are summarised:

- We propose a strong baseline that exploits the pure transformer for ReID tasks for the first time and achieve comparable performance with CNN-based frameworks.
- We design a jigsaw patches module (JPM), consisting of shift and patch shuffle operation, which facilitates perturbation-invariant and robust feature representation of objects.
- We introduce a side information embeddings (SIE) that encodes side information by learnable embeddings, and is shown to effectively mitigate the feature bias.
- The final framework TransReID achieves state-of-the-art performance on both person and vehicle ReID benchmarks including MSMT17[46], Market-1501[55], DukeMTMC-reID[33], Occluded-Duke[31], VeRi-776[25] and VehicleID[24].

2. Related Work

2.1. Object ReID

The studies of object ReID have been mainly focused on person ReID and vehicle ReID, with most state-of-the-art methods based on the CNN structure. A popular pipeline for object ReID is to design suitable loss functions to train a CNN backbone (e.g. ResNet [11]), which is used to extract features of images. The cross-entropy loss (ID loss) [56] and triplet loss [23] are most widely used in the deep ReID. Luo et al. [27] proposed the BNNeck to better combine ID loss and triplet loss. Sun et al. [36] proposed a unified perspective for ID loss and triplet loss.

Fine-grained Features. Fine-grained features have been learned to aggregate information from different part/region. The fine-grained parts are either automatically generated by roughly horizontal stripes or by semantic parsing. Methods like PCB [37], MGN [44], AlignedReID++ [28], SAN
Our method has two main stages, i.e., feature extraction and supervision learning. As shown in Figure 3, given an image \( x \in \mathbb{R}^{H \times W \times C} \), where \( H, W, C \) denote its height, width, and number of channels, respectively, we split it into \( N \) fixed-sized patches \( \{ x_{p}^{i} | i = 1, 2, \ldots, N \} \). An extra learnable [cls] embedding token denoted as \( x_{\text{cls}} \) is prepended to the input sequences. The output [cls] token serves as a global feature representation \( f \). Spatial information is incorporated by adding learnable position embeddings. Then, the input sequences fed into transformer layers can be expressed as:

\[
\mathcal{Z}_{0} = [x_{\text{cls}}; \mathcal{F}(x_{p}^{1}); \mathcal{F}(x_{p}^{2}); \cdots; \mathcal{F}(x_{p}^{N})] + \mathcal{P},
\]

where \( \mathcal{Z}_{0} \) represents input sequence embeddings and \( \mathcal{P} \in \mathbb{R}^{(N+1) \times D} \) is position embeddings. \( \mathcal{F} \) is a linear projection mapping the patches to \( D \) dimensions. Moreover, \( l \) transformer layers are employed to learn feature representations. The limited receptive field problem of CNN-based methods is addressed, because all transformer layers have a global receptive field. There are also no downsampling operations, so the detailed information is preserved.

**Overlapping Patches.** Pure transformer-based models (e.g., ViT, DeiT) split the images into non-overlapping patches, losing local neighboring structures around the patches. Instead, we use a sliding window to generate patches with overlapping pixels. Denoting the step size as \( S \), size of the patch as \( P \) (e.g., 16), then the shape of the area where two adjacent patches overlap is \( (P - S) \times P \). An input image with a resolution \( H \times W \) will be split into \( N \) patches.

\[
N = N_{H} \times N_{W} = \left\lfloor \frac{H + S - P}{S} \right\rfloor \times \left\lfloor \frac{W + S - P}{S} \right\rfloor
\]

**Side Information.** For images captured in a cross-camera system, large variations exist in terms of pose, orientation, illumination, resolution, etc. caused by different camera setup and object viewpoints. Some works \([61, 7]\) use side information such as camera ID or viewpoint information to learn invariant features. For example, Camera-based Batch Normalization (CBN) \([61]\) forces the image data from different cameras to be projected onto the same subspace, so that the distribution gap between inter- and intra-camera pairs is largely diminished. Viewpoint/Orientation-invariant feature learning \([7, 60]\) is also important for both person and vehicle ReID.

**Figure 3:** Transformer-based strong baseline framework (a non-overlapping partition is shown). Output [cls] token marked with * is served as the global feature \( f \). Inspired by \([27]\), we introduce the BNNNeck after the \( f \).
where $\lfloor \cdot \rfloor$ is the floor function and $S$ is set smaller than $P$. $N_H$ and $N_M$ represent the numbers of splitting patches in height and width, respectively. The smaller $S$ is, the more patches the image will be split into. Intuitively, more patches usually bring better performance with the cost of more computations.

**Position Embeddings.** As the image resolution for ReID tasks may be different from the original one in image classification, the position embedding pretrained on ImageNet cannot be directly loaded here. Therefore, a bilinear 2D interpolation is introduced to help handle any given input resolution. Similar to ViT, the position embedding is also learnable.

**Supervision Learning.** We optimize the network by constructing ID loss and triplet loss for global features. The ID loss $\mathcal{L}_{ID}$ is the cross-entropy loss without label smoothing. For a triplet set $\{a, p, n\}$, the triplet loss $\mathcal{L}_T$ with soft-margin is shown as follows:

$$ \mathcal{L}_T = \log \left[ 1 + \exp \left( \frac{||f_a - f_p||_2^2 - ||f_a - f_n||_2^2}{\tau} \right) \right] \tag{3} $$

**3.2. Jigsaw Patch Module**

Although transformer-based strong baseline can achieve impressive performance in object ReID, it utilizes information from the entire image for object ReID. However, due to challenges like occlusions and misalignments, we may only have partial observation of an object. Learning fine-grained local features such as striped features has been widely used for CNN-based methods to tackle these challenges.

Suppose the hidden features input to the last layer are denoted as $Z_{l-1} = [z_{l-1}^0; z_{l-1}^1; z_{l-1}^2; \ldots; z_{l-1}^N]$. To learn fine-grained local features, a straightforward solution is splitting $[z_{l-1}^1; z_{l-1}^2; \ldots; z_{l-1}^N]$ into $k$ groups in order which concatenate the shared token $z_{l-1}^0$ and then feed $k$ feature groups into a shared transformer layer to learn $k$ local features denoted as $\{f_j^k | j = 1, 2, \ldots, k\}$ and $f_j^k$ is the output token of $j$-th group. But it may not take full advantage of global dependencies for the transformer because each local segment only considers a part of the continuous patch embeddings.

To address the aforementioned issues, we propose a jigsaw patch module (JPM) to shuffle the patch embeddings and then re-group them into different parts, each of which contains several random patch embeddings of an entire image. In addition, extra perturbation introduced in training also helps improve the robustness of object ReID model. Inspired by ShuffleNet [52], the patch embeddings are shuffled via a shift operation and a patch shuffle operation. The sequences embeddings $Z_{l-1}$ are shuffled as follow:

- **Step1:** The shift operation. The first $m$ patches (except for [cls] token) are moved to the end, i.e. $[z_{l-1}^1; z_{l-1}^2; \ldots; z_{l-1}^N]$ is shifted in $m$ steps to become $[z_{l-1}^m; z_{l-1}^{m+1}; \ldots; z_{l-1}^{m-1}; z_{l-1}^1; z_{l-1}^2; \ldots; z_{l-1}^N]$.

- **Step2:** The patch shuffle operation. The shifted
patches are further shuffled by the patch shuffle operation with \( k \) groups.

With the shift and shuffle operation, the local feature \( f^j_i \) can cover patches from different body or vehicle parts which means that the local features hold global discriminative capability.

As shown in Figure 4, paralleling with the jigsaw patch, another global branch which is a standard transformer encodes \( Z_{l-1} \) into \( Z_l = \{ f^j_0, z^1_1, z^2_1, \ldots, z^N_1 \} \), where \( f^j_0 \) is served as the global feature of CNN-based methods. Finally, the global feature \( f^j_0 \) and \( k \) local features are trained with \( \mathcal{L}_{ID} \) and \( \mathcal{L}_T \). The overall loss is computed as follow:

\[
\mathcal{L} = \mathcal{L}_{ID}(f^j_0) + \mathcal{L}_T(f^j_0) + \frac{1}{k} \sum_{j=1}^{k} (\mathcal{L}_{ID}(f^j_i) + \mathcal{L}_T(f^j_i)) \quad (4)
\]

During inference, we concatenate the global feature and local features \( \{ f^j_0, f^j_1, f^j_2, \ldots, f^j_k \} \) as the final feature representation. Using \( f^j_0 \) only is a variation with lower computational cost and slight performance degradation.

### 3.3. Side Information Embeddings

After obtaining fine-grained feature representations, features are still susceptible to camera or viewpoint variations. In other words, the trained model may easily fail to distinguish the same object from different perspectives due to scene-bias. Therefore, we propose a Side Information Embedding (SIE) to incorporate the non-visual information, such as cameras or viewpoints, into embedding representations to learn invariant features.

Inspired by position embeddings which encode positional information adopting learnable embeddings, we plug learnable 1-D embeddings to retain side information. Particularly, as illustrated in Figure 4, SIE is inserted into the transformer encoder together with patch embeddings and position embeddings. In specific, suppose there are \( N_C \) camera IDs in total, we initialize learnable side information embeddings as \( \mathcal{S}_C \in \mathbb{R}^{N_C \times D} \). If camera ID of an image is \( r \), then its camera embeddings can be denoted as \( \mathcal{S}_C[r] \). Different from the position embeddings which vary between patches, camera embeddings \( \mathcal{S}_C[r] \) are the same for all patches of an image. In addition, if viewpoint of the object is available, either by a viewpoint estimation algorithm or human annotations, we can also encode the viewpoint label \( q \) as \( \mathcal{S}_V[q] \) for all patches of an image where \( \mathcal{S}_V \in \mathbb{R}^{N_V \times D} \) and \( N_V \) represents the number of viewpoint IDs.

Now comes the problem about how to integrate two different types of information. A trivial solution might be directly adding the two embeddings together like \( \mathcal{S}_C[r] + \mathcal{S}_V[q] \). However, it might make the two embeddings counteract each other due to redundant or adversarial information. We propose to encode the camera and viewpoint jointly as \( \mathcal{S}_{C,V} \in \mathbb{R}^{N_C \times N_V \times D} \).

Finally, the input sequences with camera ID \( r \) and viewpoint ID \( q \) are fed into transformer layers as follows:

\[
\mathcal{Z}_0' = \mathcal{Z}_0 + \lambda \mathcal{S}_{C,V}[r \ast N_V + q], \quad (5)
\]

where \( \mathcal{Z}_0 \) is the raw input sequences in Eq. 2 and \( \lambda \) is a hyperparameter to balance the weight of SIE. As the position embeddings are different for each patch but the same across different images, and \( \mathcal{S}_{C,V} \) are the same for each patch but may have different values for different images. Transformer layers are able to encode embeddings with different distribution properties which can then be added directly.

Here we have only demonstrated the usage of SIE with camera and viewpoint information which are both categorical variables. In practice, SIE can be further extended to encode more kinds of information, including both categorical and numerical variables. In our experiments on different benchmarks, camera and viewpoint information is included wherever available.

### 4. Experiments

#### 4.1. Datasets

We evaluate our proposed method on four person ReID datasets, Market-1501 [55], DukeMTMC-reID [33], MSMT17 [46], Occluded-Duke [31], and two vehicle ReID datasets, VeRi-776 [25] and VehicleID [24]. It is noted that, unlike other datasets, images in Occluded-Duke are selected from DukeMTMC-reID and the training/query/gallery set contains 9%/100%/10% occluded images respectively. All datasets except VehicleID provide camera ID for each image, while only VeRi-776 and VehicleID dataset provide viewpoint labels for each image. The details of these datasets are summarized in Table 1.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Object</th>
<th>ID</th>
<th>Image</th>
<th>Cam</th>
<th>View</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSMT17</td>
<td>Person</td>
<td>4,101</td>
<td>126,441</td>
<td>15</td>
<td>-</td>
</tr>
<tr>
<td>Market-1501</td>
<td>Person</td>
<td>1,501</td>
<td>32,668</td>
<td>6</td>
<td>-</td>
</tr>
<tr>
<td>DukeMTMC-reID</td>
<td>Person</td>
<td>1,404</td>
<td>36,441</td>
<td>8</td>
<td>-</td>
</tr>
<tr>
<td>Occluded-Duke</td>
<td>Person</td>
<td>1,404</td>
<td>36,441</td>
<td>8</td>
<td>-</td>
</tr>
<tr>
<td>VeRi-776</td>
<td>Vehicle</td>
<td>776</td>
<td>49,357</td>
<td>20</td>
<td>8</td>
</tr>
<tr>
<td>VehicleID</td>
<td>Vehicle</td>
<td>26,328</td>
<td>221,567</td>
<td>2</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 1: Statistics of datasets used in the paper.

#### 4.2. Implementation

Unless otherwise specified, all person images are resized to 256 x 128 and all vehicle images are resized to 256 x 256. The training images are augmented with random horizontal flipping, padding, random cropping and random erasing [57]. The batch size is set to 64 with 4 images per ID. SGD optimizer is employed with a momentum of 0.9 and the weight decay of 1e-4. The learning rate is initialized
Table 2: Comparison of different backbones. Inference time is represented by comparing each model to ResNet50 as only relative comparison is necessary. All the experiments were carried out on the same machine for a fair comparison. ViT-B/16 is regarded as the baseline model and abbreviated as Baseline in the rest of this paper.

As 0.008 with cosine learning rate decay. Unless otherwise specified, we set $m = 5, k = 4$ and $m = 8, k = 4$ for person and vehicle ReID datasets, respectively.

All the experiments are performed with one Nvidia Tesla V100 GPU using the PyTorch toolbox \(^1\) with FP16 training. The initial weights of ViT are pre-trained on ImageNet-21K and then finetuned on ImageNet-1K, while the initial weights of DeiT are trained only on ImageNet-1K.

Evaluation Protocols. Following conventions in the ReID community, we evaluate all methods with Cumulative Matching Characteristic (CMC) curves and the mean Average Precision (mAP).

4.3. Results of Transform-based Baseline

In this section, we compare CNN-based and transformer-based backbones in Table 2. To show the trade-off between computation and performance, several different backbones are chosen. DeiT-small, DeiT-Base, ViT-Base denoted as DeiT-S, DeiT-B, ViT-B, respectively. ViT-B/16\(_{s=14}\) means ViT-Base with patch size 16 and step size $S = 14$ in overlapping patches setting. For a comprehensive comparison, inference time consumption of each backbone is included as well.

We can observe a large gap in model capacity between the ResNet series and DeiT/ViT. DeiT-S/16 is a little bit better in performance and speed compared to ResNet50. DeiT-B/16 and ViT-B/16 achieve similar performance with ResNeSt50 [50] backbone, with less inference time than ResNeSt50 (1.79x vs 1.86x). When we reduce the step size of the sliding window $S$, the performance of the Baseline can be improved while the inference time is also increasing. ViT-B/16\(_{s=12}\) is faster than ResNeSt200 (2.81x vs 3.12x) and performs slightly better than ResNeSt200 on ReID benchmarks. Therefore, ViT-B/16\(_{s=12}\) achieves better

\(^1\)http://pytorch.org

Table 3: The ablation study of jigsaw patch module. ‘w/o rearrange’ means the patch features are split into parts without rearranging including shift and shuffle operation. ‘w/o local’ means we evaluate the global feature without concatenating local features.

4.4. Ablation Study of JPM

The effectiveness of the proposed JPM module is validated in Table 3. JPM provides +2.6% mAP and +1.0% mAP improvements compared to baseline on MSMT17 and VeRi-776, respectively. Increasing the number of groups $k$ can improve the performance while slightly increasing inference time. In our experiment, $k = 4$ is a choice to trade off speed and performance. Comparing JPM and JPM w/o rearrange, we can observe that the shift and shuffle operation helps the model learn more discriminative features with +0.5% mAP and +0.2% mAP improvements on MSMT17 and VeRi-776, respectively. It is also observed that, if only the global feature $f_g$ is used in inference stage (still trained with full JPM), the performance (denoted as “w/o local”) is nearly comparable with the version of full set of features, which suggests us to only use the global feature as an efficient variation with lower storage cost and computational cost in the inference stage. The attention maps visualized in Figure 5 show that JPM with the rearrange operation can help the model learn more global context information and more discriminative parts, which makes the model more robust to perturbations.

4.5. Ablation Study of SIE

Performance Analysis. In Table 4, we evaluate the effectiveness of the SIE on MSMT17 and VeRi-776. MSMT17 does not provide viewpoint annotations, so the
Table 4: Ablation study of SIE. Since the person ReID datasets do not provide viewpoint annotations, viewpoint information can only be encoded in VeRi-776.

Table 5: The ablation study of TransReID.

Visualisation of Distance Distribution. As shown in Figure 6, the distribution gaps with cameras and viewpoints variations are obvious in Figure 6a and Figure 6b, respectively. When we introduce the SIE module into Baseline, the distribution gaps between inter-camera/viewpoint and intra-camera/viewpoint are reduced, which shows that the SIE module weakens the negative effect of the scene-bias caused by various cameras and viewpoints.

Ablation Study of $\lambda$. We analyze the influence of weight $\lambda$ of the SIE module on the performance in Figure 7. When $\lambda = 0$, Baseline achieves 61.0% mAP and 78.2% mAP on MSMT17 and VeRi-776, respectively. With $\lambda$ increasing, the mAP is improved to 63.0% mAP ($\lambda = 2.0$ for MSMT17) and 79.9% mAP ($\lambda = 2.5$ for VeRi-776), which means the SIE module now is beneficial for learning invariant features. Continuing to increase $\lambda$, the performance is degraded because the weights for feature embedding and the position embedding are weakened.

4.6. Ablation Study of TransReID

Finally, we evaluate the benefits of introducing JPM and SIE in Table 5. For the Baseline, JPM and SIE improve the performance by +2.6%/+1.0% mAP and +1.4%/+1.4% mAP on MSMT17/VeRi-776, respectively. With these two modules used together, TransReID achieves 64.9% (+3.9%) mAP and 80.6% (+2.4%) mAP on MSMT17 and VeRi-776, respectively. The experimental results show the effectiveness of our proposed JPM, SIE, and the overall framework.

4.7. Comparison with State-of-the-Art Methods

In Table 6, our TransReID is compared with state-of-the-art methods on six benchmarks including person ReID, occluded ReID and vehicle ReID.

Person ReID. On MSMT17 and DukeMTMC-reID, TransReID* (DeiT-B/16) outperforms the previous state-of-the-art methods by a large margin (+5.5%/+2.1% mAP). On Market-1501, TransReID* (256 × 128) achieves comparable performance with state-of-the-art methods especially on mAP. Our method also shows superiority when compared with methods that also integrate camera information like CBN [61]. We also surpass a typical ‘CNN backbone + Transformer’ method named PAT in all person datasets.
Table 6: Comparison with state-of-the-art methods. DukeMTMC denotes the DukeMTMC-reID benchmark. The star * in the superscript means the backbone is with a sliding-window setting. Results are shown for person ReID datasets (left) and vehicle ReID datasets (right). Only the small subset of VehicleID is used in this paper. \( \circ \) and \( \heartsuit \) indicate the methods are using camera IDs and viewpoint labels, respectively. \( \spadesuit \) means both are used. Viewpoint and camera information are used wherever available. Best results for previous methods are labeled in blue and best of our methods are labeled in bold.

Occluded ReID. ISP implicitly uses human body semantic information through iterative clustering and \( \text{HOReID} \) introduces external pose models to align body parts. TransReID (DeiT-B/16) achieves 55.6% mAP with a large margin improvement (at least +3.3% mAP) compared to aforementioned methods, without requiring any semantic and pose information to align body parts, which shows the ability of TransReID to generate robust feature representations. Furthermore, TransReID* improves the performance to 58.1% mAP with overlapping patches.

Vehicle ReID. On Veri-776, TransReID* (DeiT-B/16) reaches 82.3% mAP surpassing GLAMOR by 2.0% mAP. When only using viewpoint annotations, TransReID* still outperforms VANet and \( \text{SAVER} \) on both Veri-776 and VehicleID. Our method achieves state-of-the-art performance about 85.2% Rank-1 accuracy on VehicleID.

DeiT vs ViT vs CNN. TransReID* (DeiT-B/16) reaches competitive performance with existing methods under a fair comparison (ImageNet-1K pre-training). Extra results of our methods with ViT-B/16 are also reported in Table 6 for further comparison. DeiT-B/16 achieves similar performance with ViT-B/16 for shorter image patch sequences. When the number of input patches is increasing, ViT-B/16 reaches better performance than DeiT-B/16, which shows ImageNet-21K pre-training provides ViT-B/16 better generalization capability. Although CNN-based methods mainly report performance with the ResNet50 backbone, they may include multiple branches, attention modules, semantic models, or other modules that increase computational consumption. We have conducted a fair comparison on inference speed between TransReID* and \( \text{MGN} \) [44] on the same computing hardware. Compared with \( \text{MGN} \), TransReID* is 4.8% faster in speed.

5. Conclusion

In this paper, we investigate a pure transformer framework for the object ReID task, and propose two novel modules, \( \text{i.e.} \), jigsaw patch module (JPM) and side information embedding (SIE). The final framework TransReID outperforms all other state-of-the-art methods by a large margin on several popular person/vehicle ReID datasets. Based on the promising results achieved by TransReID, we believe the transformer has great potential to be further explored for ReID tasks. Based on the rich experience gained from CNN-based methods, it is in prospect that more efficient transformer-based networks can be designed with better representation power and less computational cost.

Acknowledgement This work was supported by Alibaba Group through Alibaba Research Intern Program, the National Natural Science Foundation of China (No.62173302) and the Fundamental Research Funds for the Central Universities (2021XZZX022).


[38] Ziruo Sun, Xiushan Nie, Xiaoming Xi, and Yilong Yin. Clvnet: A multi-branch network for vehicle re-identification based on common field of view. In ACMMM, pages 3523–3531, 2020. 8


