Distributed Memory Network For Person Re-identification

Abstract

In this paper, we propose a novel unsupervised adaptive person re-identification method for distributed edge computing environments.The core of the method is to solve the problem of non-independent and identical training data distribution of edge devices in a distributed environment and mining cross-camera pedestrian relationships in a distributed environment. In the field of traditional person re-identification, whenever a new data distribution is encountered, it is necessary to aggregate the data of the labeled source domain and the unlabeled target domain for incremental training. Under the circumstance that the privacy protection of personal data is being paid more attention, it is obviously inappropriate to collect and store pedestrian data on a large scale.Combined with the popular use of edge computing devices such as jetson tx2 in recent years, we set up the following distributed scenarios: edge devices have the ability to train models, edge devices can only get pedestrian data collected by local cameras, and between edge devices No exchange of pedestrian raw image data is allowed. In the above scenario, our method is mainly divided into 4 stages: (1) edge device data style transfer (2) random walk training (3) cross-camera relationship mining and (4) random walk incremental training. Our experiments on the main pedestrian re-identification dataset have proved the effectiveness of the method. In addition, we also analyzed the model training time, inference time, and data interaction bandwidth on the jetson tx2 development board. Prove the practicality of the method

1 Introduction

.....

1.1 subsection

.....

2 Related work

....

2.1 subsection

....

..

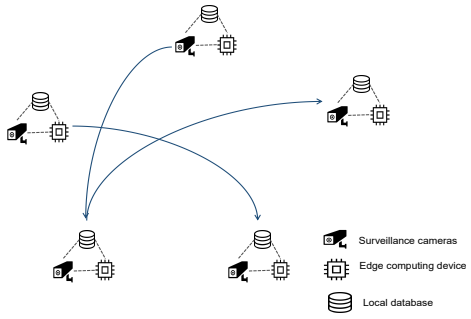

Figure 1

3 Method

Definition In a distributed environment,we adopt the sparse space-time tracklets sampling(SSTT) [Li *et al.*, 2018] to sample the training tracklets from each camera. Each tracklet is represented as$\{t_n^m, y_n^m, g_n^m\}$, where t_n^m means the n-th tracklet in m-th camera.Denoting $t_n^m = \left\{I_1^{t_n^m}, I_2^{t_n^m}, ..., I_k^{t_n^m}\right\}$, where $I_k^{t_n^m}$ is the k-th image of the n-th tracklet in m-th camera. Each tracklet has two labels, one is a local label and the other is a global label.The local label is represented by y_n^m, whenever the edge device collects new tracklet data, it will automatically be assigned a local label, local labels are distinguishable only in local data and will be used for random walk training. The global label is represented by g_n^m, when distributed clustering is completed, each tracklet will obtain a global label, global labels are distinguishable in global data and will be used for incremental training.f_{model} is the person re-identically model to extract high-dimensional features of input images.

3.1 Overview

Our framework is mainly divided into the following functional modules: style transfer module, random walk training

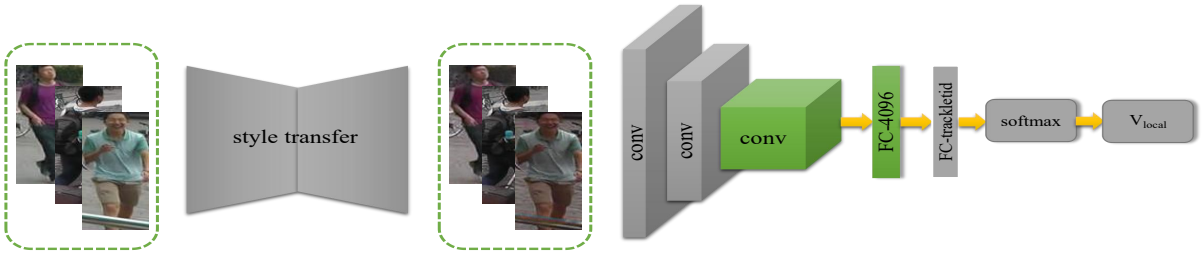

Figure 2

module, clustering module, incremental training module, and global unified module. The style transfer module is mainly responsible for style transfer of the collected pedestrian tracklet data, the edge device will save the data after the style transfer to the local and assign a local label. The random walk training module is mainly responsible for model training using locally saved pedestrian data and local labels. The trained model will replace the local model and randomly select the next hop to continue training. The process of style transfer and local training is shown in Fig.2 . The next is a memory-based clustering module, as shown in Fig.3 , memory is a high-dimensional feature fusion of pedestrian tracklet, and memoryBank is a collection of local memories.The clustering module will interchanges memoryBank between edge devices and builds neighbor list for each memory according to "mutual top1" and "Triangle Relation" mentioned in section 3.3. We can obtain the global label of each memory through the neighbor relationship between the memories. The incremental random walk training module will use the local data of the edge device and its global labels to incrementally train the model. Finally, the global unification module will unify the models on all edge devices.

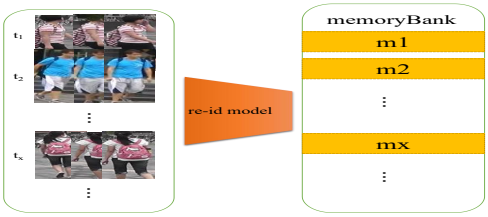

Figure 3

3.2 Single Camera style Transfer

In order to minimize the negative impact of non-Independent and identically distributed(non-IID) in the distributed environment on model performance, each edge device will transform the locally collected data to the target domain through the style transfer. When each edge device is initialized, it will obtain the same pedestrian data as the target domain data for style transfer. we adopt Cycle-GAN [Zhu *et al.*, 2017] as the style transfer model. Given two different domains A and B, the goal of Cycle-GAN is to train two mapping networks G and F so that the distribution of $G(A)$ is as close as possible to the distribution of B and the distribution of $F(B)$ is as close as possible The distribution of A, The overall CycleGAN loss function is expressed as:

$$\begin{aligned} \upsilon(G, F, D_A, D_B) = & \upsilon_{GAN}(G, D_B, A, B) \\ & + \upsilon_{GAN}(F, D_A, B, A) \\ & + \lambda \upsilon_{cyc}(G, F) \end{aligned} \tag{1}$$

where $\upsilon_{GAN}(G, D_B, A, B)$ and $\upsilon_{GAN}(F, D_A, B, A)$ are the adversarial loss. $\upsilon_{cyc}(G, F)$ is the cycle consistency loss. In order to ensure that the color of the image remains stable before and after the style transfer, we adopt the identity mapping loss [Zhu *et al.*, 2017], which is expressed as:

$$\begin{aligned} \upsilon_{identity}(G, F) = & E_{x \sim p_x}\left[\|F(x) - x\|_1\right] \\ & + E_{y \sim p_y}\left[\|G(y) - y\|_1\right] \end{aligned} \tag{2}$$

where $x \in A$ and $y \in B$

3.3 Random Walk Training

In the edge computing environment, image data is not allowed to be transmitted between edge devices, and only model data or high-dimensional feature vectors are allowed to be transmitted. For this scenario, we proposed random walk training method. The model walks randomly between edge devices, every time it reaches a edge devices, it will train with local data.

We consider the model training on a single edge device as a classification problem [Zheng *et al.*, 2016b], but each edge device only contains data with local labels, so whenever the model walks to a certain edge device, the model classification layer needs to be re-initialized according to the local data category to optimize the cross-entropy loss. the cross-entropy loss is formulated as:

$$\nu_{local} = -\frac{1}{\sum_{n=1}^{N} K_n} \sum_{n=1}^{N} \sum_{k=1}^{K_n} \log p(y_n^m \mid I_k^{t_n^m}) \tag{3}$$

where N is the number of local tracklets and K_n is the number of images the n-th tracklet contains. $p(y_n^m \mid I_k^{t_n^m})$ is the predicted probability that the image $I_k^{t_n^m}$ belongs to y_n^m, which is obtained by the classification module.

The model initialized on each edge device is ResNet-50 pre-trained on ImageNet, we fix the first two residual layers of ResNet-50 to save computing costs. At the same time, the classification layer of the network will re-initialize the

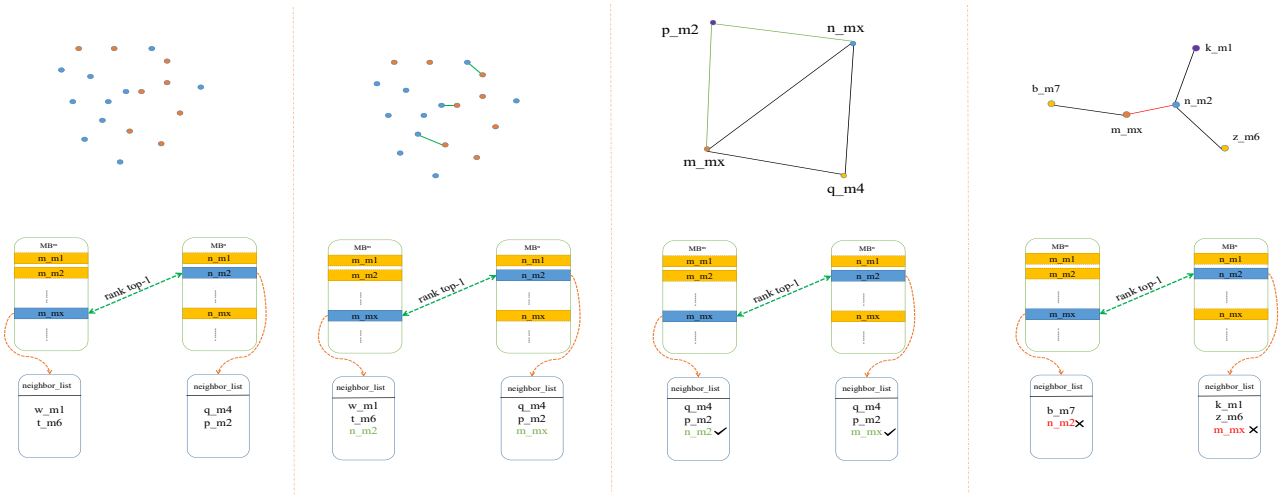

Figure 4

weights for training on each edge device based on local data, so as shown in the fig.n, the model of random walk between edge devices is only the middle part of the entire network structure.

3.4 Distributed Clustering

Memory Bank To mine relationships between cross camera tracklets, we need to extract high-dimensional feature vectors for each trackle. We call the high-dimensional feature vector of each tracklet $memory$, the $memory$ is obtained as follows:

$$memory_n^m = \sum_{k=1}^{K_n} f_{model}(I_k^{t_n^m}) \tag{4}$$

where $memory_n^m$ means the memory of the n-th tracklet in m-th camera. Each edge device will set a container to store the memory of all local tracklets, which we call $MemoryBank$. The $MemoryBank$ in the m-th camera is represented by MB^m, $MB^m = \{memory_n^m\}_{n=1}^N$.

Relationship Graph As shown in Fig.3a, each memory can be regarded as a point in the feature space. Our goal is to mine the relationship between memories and find the memory belonging to the same person across cameras. Then we can use this cross-camera relationship to incremental training model. So we need to build a relationship graph between memories, if two memories belong to the same class, we will establish connecting edges for them in the relationship graph.

We designed the method named $mutual\ top1$ to initialize the relationship graph. As shown in fig.3b, Suppose we have two memory banks from camera a and camera b, which named MB^a and MB^b. $memory_t^a$ is the memory of the t-th tracklet in camera a and $memory_k^b$ is the memory of the k-th tracklet in camera b. We use the $L2$ distance to measure the pairwise similarities between $memory_t^a$ and all memories in MB^b and also measure the pairwise similarities between $memory_k^b$ and all memories in MB^a.if $memory_k^b$ has the minimal pairwise distance with $memory_t^a$ and $memory_t^a$ has the minimal pairwise distance with $memory_k^b$, we say both of them satify $mutual\ top1$ and we will establish connecting edges for them in the relationship graph.

In a distributed environment, each $memory$ will maintain a neighbor list.Establishing connected edges in the relationship graph means update the neighbor lists of two $memories$ that satisfying the $mutual\ top1$.

Triangle Relation As shown in Fig.3c, in the relationship graph constructed from $mutual\ top1$, a wrong connected edge may bring "butterfly effect", which will greatly affect the clustering effect. In order to ensure the accuracy of the connected edges in the relationship graph as much as possible, we propose a method named $triangle\ relation$. As shown in fig.3d, two memories satisfying $triangle\ relation$ means that at least α memories exists in the relationship graph can establish a connected edge with both of them.

In a distributed environment, each $memory$ will compare its neighbor list with the neighbor list of its neighbors to find if there have at least α public neighbors, and then delete the neighbors that do not meet the conditions from its neighbor list.

Global Label For further incremental training, we need to assign a globally identifiable label g_n^m to each memory based on the clustering results obtained from the relationship graph.

In a distributed environment, each memory sends its neighbor list to its neighbors. Each memory will expand its neighbor list according to the received list. If there is an update, it will continue to flood the latest neighbor list to its neighbors. When the above process ends, the same type of memory will have the same neighbor list, which is equivalent to having globally recognizable labels.

3.5 Incremental Random Walk Training

The incremental random walk training settings are basically the same as the random walk training settings. The difference is that the classification layer of the model is initialized according to the number of global categories after clustering, and the training data uses global labels. The incremental

training cross-entropy loss is formulated as:

$$\nu_{incremental} = -\frac{1}{\sum_{n=1}^{N} K_n} \sum_{n=1}^{N} \sum_{k=1}^{K_n} \log p(g_n^m \mid I_k^{t_n^m}) \quad (5)$$

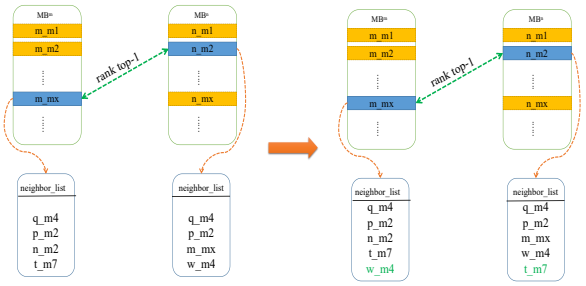

Figure 5

3.6 Global Model Unification

Each edge device stores a model locally, in order to better improve the accuracy of system retrieval, we propose two methods to unify models on all edge devices. The first one is that all edge devices use the model on the target device which is randomly selected. The second one is fusion models of all edge devices and calculate the average model, all edge devices will adopt the average model. In the experiment we used the first method.

4 Experiment

4.1 datasets

All experiments are evaluated on three large-scale image person re-identification datasets(Market-1501 [Zheng *et al.*, 2015], DukeMTMC-reID [Ristani *et al.*, 2016; Zheng *et al.*, 2017] and MSMT17 [Wei *et al.*, 2018])and one video person re-identification dataset(Mars [Zheng *et al.*, 2016a]). Performance is evaluated by the cumulative matching characteristic (CMC) and mean Average Precision (mAP).We adopt the experiments settings and tracklet sampling methods of TAUDL [Li *et al.*, 2018] and UTAL [Li *et al.*, 2019]. For image datasets, we assume all images of a person in one camera are belong to a single tracklet. For video datasets, the same person has multiple tracklets under a single camera, we randomly select one for training.

4.2 Experimental Setup

Style Transfer Model

As introduced in section 3.2, we employ Cycle-GAN [Zhu *et al.*, 2017] as the style transfer model. In each dataset, we choose camera 1 as the target domain for other camera performing a style transfer. we follow the parameter settings of, we resize all images to 256×256 and set the batch size = 1. Beside, the learning rate is 0.0001 for the Discriminator and 0.0002 for the Generator.

Person Re-identification Model

We use ResNet-50 [He *et al.*, 2016] pre-trained on ImageNet [Deng *et al.*, 2009] as our backbone network. Considering the limited computing resources on the edge device, we fix the first two residual layers of ResNet-50, the input image resizes to 256×128. When training, we use random flip, random cut, random erase [Zhong *et al.*, 2017] for data augmentation. we set the learning rate = 0.1 and use SGD optimizer to train the model.The batch size is set to 0.1.Each edge device needs to be trained for 6 epochs, 8 epochs, 3 epochs, 6 epochs when trained on Market-1501, DukeMTMC-reID, MSMT17 and MARS. In testing, we extract the L2-normalized output of Pooling-5 layer as the image feature and adopt the Euclidean distance to measure the similarities between query and gallery images, which reference [Zhong *et al.*, 2019]

Baseline

We use ResNet50 pre-trained on ImageNet as the experimental baseline.

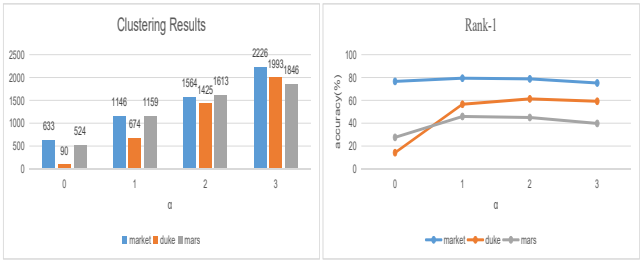

Figure 6

4.3 Parameter Analysis

We first perform a sensitivity analysis on the hyper-parameter in our experiments.As mentioned in section3.4, the indicator of whether the two memories are of the same type in the clustering process is that they form at least α triangles in the relationship graph. We will experiment with different α values in different datasets to detect clustering effects.

Fig 3.1 shows the clustering results when the value of α is 0, 1, 2, 3 on the three datasets of market, duke, and mars. The results show that when the value of α is larger, it is difficult for most memories to establish connected edges with other memories in the relationship graph and that means we will get more categories.When α is set to 0, the number of categories after clustering is less than the actual number of categories on all three datasets. Figure 3.2 shows the performance of the model after incremental training using the corresponding clustering results. The performance is optimal when α = 1 on the market dataset, the best performance is α = 2 on the duke dataset, and on the mars dataset, the performance is optimal when α = 1. On three datasets, when α = 0, the effect is worse than before the incremental training, which proves that the clustering has no effect.

4.4 Evaluation

Baseline Performance

Table 1 shows the results of the baseline .We tested the model pre-trained on ImageNet directly(called Direct Transfer) on

method	Market			DukeMTMC			MARS			MSMT17		
	R-1	R-5	mAP	R-1	R-5	mAP	R-1	R-5	mAP	R-1	R-5	mAP
baseline	8.4	17.5	2.3	7.3	13.8	2.1	6.5	13.9	2.8	3.8	7.6	0.7
rw	60.0	73.0	35.6	52.3	68.1	33.7	40.2	56.1	23.9	9.8	12.7	3.1
rw+t	76.8	88.7	51.8	59.2	73.8	36.6	44.0	60.9	26.3	25.2	38.8	8.9
rw+t+i	79.4	90.6	55.5	61.3	76.0	36.8	45.9	61.7	26.6	28.6	42.1	10.0

Table 1: Latex default table

module	time(ms)
style transfer model inference(per image)	0.1s
person re-id model inference(per images)	0.2s
style transfer model training(batchsize=128)	0.2s
person re-id model training(batchsize=1)	0.2s

Table 2: Run time of each module

data packets	size(Kb)
style transfer model inference	0.1s
person re-id model inference	0.2s
style transfer model training	0.1s
person re-id model training	0.1s

Table 3: Latex default table

the target testing dataset and found that the effect was very poor. For example, the rank-1 on the market dataset was only 8.4%, and the mAP was only 2.3%.

Ablation experiment

To prove the effectiveness of our proposed method, we conduct ablation studies in Table 1.We first show the effectiveness of random walk training.....Secondly we will show the effectiveness of style transfer.....Finally We will show the effectiveness of clustering.....

Distributed Performance

Run-time analysis We performed model run-time analysis on NVIDIA JETSON TX2. As shown in Table 2, we counted the run-time of the style transfer model inference module, person re-identification model inference module, person re-identification model training module and style transfer model training module on jetson tx2. The style transfer model inference module and the person re-identification model inference module count the time it takes to process an image, and the style transfer model training module and the person re-identification training module count the time it takes to train each mini-batch of data.

Communication packet analysis We have statistics on the data packets that the system needs to interact with in a distributed environment. The main types and sizes of data packets are statistically summarized. The statistical results are shown in Table 3.

5 Conclusion

....

6 Acknowledgements

....

References

[Deng *et al.*, 2009] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[Li *et al.*, 2018] Minxian Li, Xiatian Zhu, and Shaogang Gong. Unsupervised person re-identification by deep learning tracklet association. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 737–753, 2018.

[Li *et al.*, 2019] Minxian Li, Xiatian Zhu, and Shaogang Gong. Unsupervised tracklet person re-identification. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[Ristani *et al.*, 2016] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *European Conference on Computer Vision*, pages 17–35. Springer, 2016.

[Wei *et al.*, 2018] Longhui Wei, Shiliang Zhang, Wen Gao, and Qi Tian. Person transfer gan to bridge domain gap for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 79–88, 2018.

[Zheng *et al.*, 2015] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *Proceedings*

method	Market			DukeMTMC		
	R-1	R-5	mAP	R-1	R-5	mAP
LOMO	8.4	17.5	2.3	7.3	13.8	2.1
Bow	60.0	73.0	35.6	52.3	68.1	33.7
UMDL	76.8	88.7	51.8	59.2	73.8	36.6
PTGAN	79.4	90.6	55.5	61.3	76.0	36.8
PUL	79.4	90.6	55.5	61.3	76.0	36.8
SPGAN	79.4	90.6	55.5	61.3	76.0	36.8
CAMEL	79.4	90.6	55.5	61.3	76.0	36.8
MMFA	79.4	90.6	55.5	61.3	76.0	36.8
SPGAN+LMP	79.4	90.6	55.5	61.3	76.0	36.8
Tj-AIDL	79.4	90.6	55.5	61.3	76.0	36.8
CamStyle	79.4	90.6	55.5	61.3	76.0	36.8
HHL	79.4	90.6	55.5	61.3	76.0	36.8
Ours	79.4	90.6	55.5	61.3	76.0	36.8

Table 4: Latex default table

of the IEEE international conference on computer vision, pages 1116–1124, 2015.

[Zheng *et al.*, 2016a] Liang Zheng, Zhi Bie, Yifan Sun, Jingdong Wang, Chi Su, Shengjin Wang, and Qi Tian. Mars: A video benchmark for large-scale person re-identification. In *European Conference on Computer Vision*, pages 868–884. Springer, 2016.

[Zheng *et al.*, 2016b] Liang Zheng, Yi Yang, and Alexander G Hauptmann. Person re-identification: Past, present and future. *arXiv preprint arXiv:1610.02984*, 2016.

[Zheng *et al.*, 2017] Zhedong Zheng, Liang Zheng, and Yi Yang. Unlabeled samples generated by gan improve the person re-identification baseline in vitro. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3754–3762, 2017.

[Zhong *et al.*, 2017] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. *arXiv preprint arXiv:1708.04896*, 2017.

[Zhong *et al.*, 2019] Zhun Zhong, Liang Zheng, Zhiming Luo, Shaozi Li, and Yi Yang. Invariance matters: Exemplar memory for domain adaptive person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 598–607, 2019.

[Zhu *et al.*, 2017] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.