Hierarchical Clustering with Hard-batch Triplet Loss for Person Re-identification

Kaiwei Zeng[1], Munan Ning[2], Yaohua Wang[3]*, Yang Guo[4]*
National University of Defense Technology
Changsha,China
[1]zengkaiwei1997@gmail.com, [2]munanning@gmail.com, [3]nudtyh@gmail.com, [4]guoyang@nudt.edu.cn

Abstract

*For clustering-guided fully unsupervised person re-identification (re-ID) methods, the quality of pseudo labels generated by clustering directly decides the model performance. In order to improve the quality of pseudo labels in existing methods, we propose the HCT method which combines **H**ierarchical **C**lustering with hard-batch **T**riplet loss. The key idea of HCT is to make full use of the similarity among samples in the target dataset through hierarchical clustering, reduce the influence of hard examples through hard-batch triplet loss, so as to generate high quality pseudo labels and improve model performance. Specifically, (1) we use hierarchical clustering to generate pseudo labels, (2) we use PK sampling in each iteration to generate a new dataset for training, (3) we conduct training with hard-batch triplet loss and evaluate model performance in each iteration. We evaluate our model on Market-1501 and DukeMTMC-reID. Results show that HCT achieves 56.4% mAP on Market-1501 and 50.7% mAP on DukeMTMC-reID which surpasses state-of-the-arts a lot in fully unsupervised re-ID and even better than most unsupervised domain adaptation (UDA) methods which use the labeled source dataset. Code will be released soon on https://github.com/zengkaiwei/HCT*

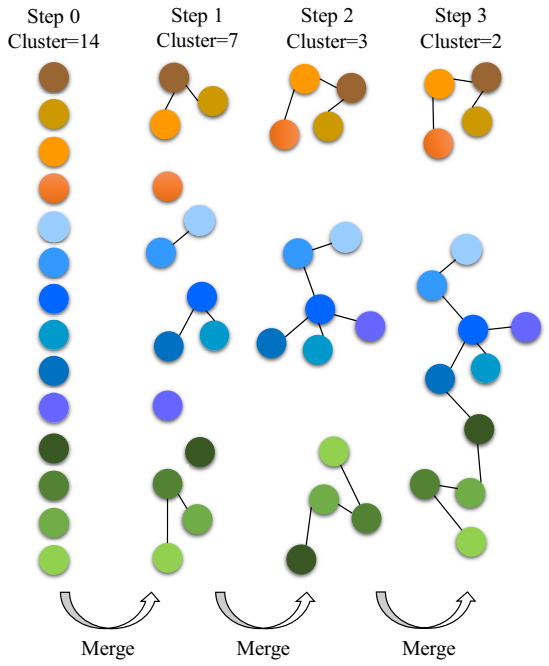

Figure 1. Hierarchical clustering. Each circle represents a sample, and the step represents the current merging stage. We use a bottom-up method to merge clusters step by step according to the distance between clusters in the current step.

1. Introduction

Person re-identification (re-ID) is mainly used to match pictures of the same person that appears in different cameras, which is usually used as an auxiliary method of face recognition to identify pedestrian information. Currently, re-ID has been widely used in the field of security and has been the focus of academic research. With the development of convolutional neural networks (CNN), supervised re-ID [9, 11, 21, 25, 30, 2, 31, 23] has achieved excellent performance. However, due to the data deviation in different datasets, the performance of the model trained on the source domain will significantly decline when it is directly transferred to the target domain. Besides, supervised learning requires a large amount of manually annotated data, which is costly in real life. Therefore, supervised re-ID is difficult to meet the requirement of practical application and people tend to focus on unsupervised re-ID.

Recently, people pay more attention on unsupervised re-ID and achieve good progress. Some works focus on unsupervised domain adaptation (UDA). UDA usually needs manually annotated source data and unlabeled target data.

*Corresponding author

In UDA, some people use GAN to transform the style of images in the source domain to the style of the target domain [4, 27, 38, 15]. They keep labels unchanged, then they conduct training on generated labeled images. Others focus on the change of images between different cameras and datasets. They identify images by learning differences between the source domain and the target domain [36]. Although the expansion of the dataset will generate many reliable data, it is highly dependent on the quality of generated images. Besides, it will also generate some awful images, which will mislead the training and affect model performance. More importantly, these UDA methods only try to reduce differences between the target domain and source domain. However, similarities of images within the target domain are ignored. Besides, UDA methods need a labeled source dataset which still cost a lot.

In recent studies, a fully unsupervised method BUC is proposed [14] and it does not use any manually labeled dataset. BUC only compares the similarity of images in the target dataset and directly use the bottom-up hierarchical clustering to merge samples. BUC merges a fixed number of clusters, updates pseudo labels, and fine-tunes the model step by step until convergence. Finally, it achieves good performance and even surpasses some methods of UDA [4, 6, 26]. However, the performance of BUC has a significant drop in later merging steps. Because BUC just relies on similarities among samples in merging, it makes BUC difficult to distinguish hard examples, especially in early merging steps when the model is poor. Hard examples mean those similar samples but have different identities. Their features are close to each other in high dimensional space so it is difficult to distinguish them by clustering and they will lead to wrong merging. In the later, these wrong merging will generate lots of false pseudo labels which mislead training and result in a decline in performance.

In order to solve these problems and make full use of the similarity of images in the target dataset, we propose HCT, which also a fully unsupervised method just uses the target dataset without any manually annotated labels. The process of hierarchical clustering is shown in Figure 1. In the beginning, we regard each sample as a cluster which has different identities, and then we select a fixed number of clusters for merging in each step according to the distance between clusters. Finally, all clusters will be merged gradually and we set pseudo labels according to clustering results. After clustering, we use hard-batch triplet loss [9] to optimize the model. Hard-batch triplet loss can reduce the distance between similar samples and increase the distance between different samples. It can effectively reduce the influence of hard examples. Specifically, (1) we use hierarchical clustering to merge samples and generate pseudo-labels according to clustering results, (2) we randomly select K instances from P identities (PK sampling) to generate a new dataset for training to meet the need of hard-batch triplet loss, (3) we fine-tune the model and evaluate model performance. We repeat the process of clustering, PK sampling, fine-tuning training, evaluation until the model reaches convergence.

To summarize, our contributions are:

- We propose a fully unsupervised re-ID method HCT. Based on pre-trained ResNet-50[8] on ImageNet, we directly use pseudo labels generated by hierarchical clustering as supervision to conduct model training on the target dataset without any manually annotated labels.
- We use PK sampling to generate a new dataset for training after hierarchical clustering in each iteration. Compared to using the whole datasets, PK sampling meets the need of hard-batch triplet loss[9] which can reduce the influence of hard examples and improve model performance.
- To correct false pseudo labels, we initialize all pseudo labels at the beginning of each iteration until the quality of pseudo labels stabilizes and the model performance no longer improves.
- We evaluate our method on Market-1501 and DukeMTMC-reID. Extensive experiments show that our method surpass state-of-the-arts a lot in fully unsupervised re-ID, even better than most UDA methods.

2. Related Work

2.1. Unsupervised Domain Adaptation Re-ID

In the past, people tend to use traditional manual features [1, 13] to conduct unsupervised domain adaptation, but the performance on large datasets is usually poor. With the popular of CNN, people begin to apply deep learning to unsupervised domain adaptation.

Deng et al. put forward SPGAN [4]. They believe that the main reason for the poor performance of direct transfer is the different camera styles of different datasets. They use CycleGAN [38] to translate images styles from the source domain to the target domain while keeping image labels unchanged. Finally, they perform supervised learning on generated images. Zhong et al. propose ECN [37], ECN focuses on exemplar-invariance [28, 29], camera-invariance [36], and neighborhood-invariance [3]. Based on these, ECN separately sets triplet loss to increase the distance between different samples and reduce the distance between similar samples. ECN stores samples in the exemplar memory model [18, 24] and sets pseudo labels according to it.

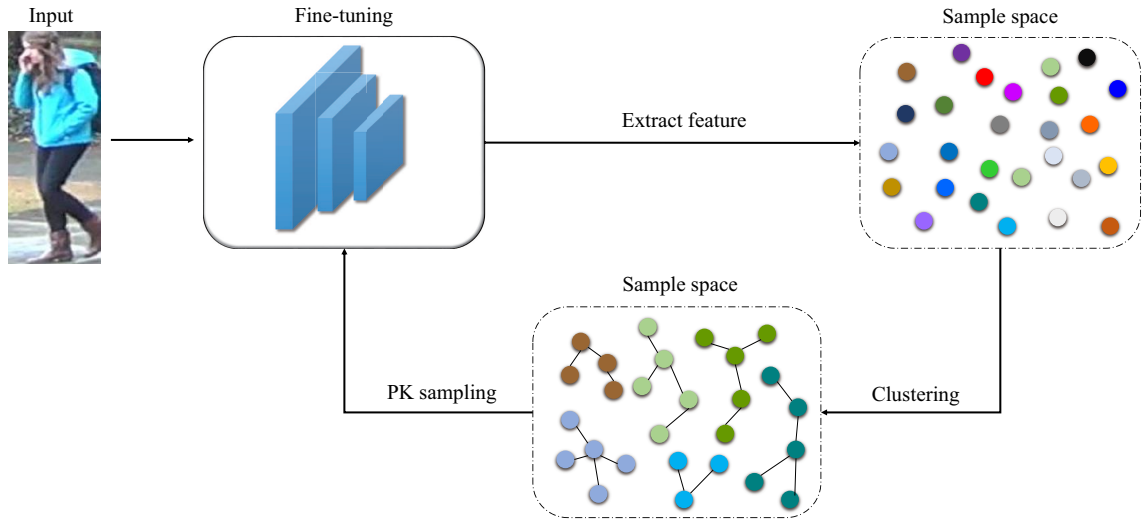

Figure 2. The structure of our HCT. Different colors represents different pseudo labels. We use pre-trained ResNet-50 [8] on ImageNet as our backbone. The input of HCT are unlabeled target images.

Finally, ECN also conducts training according to pseudo labels and get good performance.

In addition to set pseudo labels as supervision, people also try to use models to learn some auxiliary information to improve generalization ability. Zhong et al. propose HHL [36]. HHL improves model performance through camera-invariance and domain connectivity. Xiao et al. propose EANet [10]. EANet proposes Part Aligned Pooling (PAP) and Part Segmentation Constraint (PSC). PAP cuts and aligns images according to the key points of the body posture. PSC enables the model to predict labels of different part about feature map and locate the corresponding position of each part accurately. EANet combines PAP with PSC to make full use of pedestrians pose segmentation information to improve performance.

Although these methods have achieved some improvements, most of them only focus on the difference between the source domain and the target domain. However, they do not fully explore the similarity of images in the target domain.

2.2. Clustering-guided Re-ID

Clustering-guided re-ID is usually trained with pseudo labels generated by clustering, which can be divided into clustering-guided domain adaptation and clustering-guided fully unsupervised re-ID.

For clustering-guided domain adaptation, Hehe et al. [6] propose PUL. PUL gets the pre-trained model through training on the labeled source dataset, then uses CNN to fine-tune the model and uses K-means to cluster samples. At the beginning of training, PUL only selects a part of reliable samples which close to the clustering centroid for training to avoid falling into local optimum. As the model becomes better, more samples will be selected. This strategy effectively promotes the convergence of the model and improves performance. However, K-means is very sensitive to the k value. Besides, as a a partition based clustering method, clustering centroids are easily dragged by outliers, it will generate lots of false pseudo labels which seriously affect the optimization of the model and ultimately limit model performance.

For clustering-guided fully unsupervised re-ID, Lin et al. propose BUC [14]. BUC does not use any labeled source data, only use unlabeled target data and pre-trained model on ImageNet instead of other re-ID dataset. BUC extracts image features with CNN, then merges a fixed number of clusters according to the distance between clusters in each step. After merging, BUC fine-tunes the model with generated pseudo labels, repeats the progress of merging and fine-tuning until the model performance no longer improves. However, the performance of BUC has a significant drop in later merging steps. That is due to the poor pre-trained model in the beginning and some hard examples in the target dataset. BUC cannot solve the problem of false pseudo labels, which will affect the optimization of the model. These false pseudo labels have a superposition effect in later merging steps and result in a significant performance drop in the end. In this paper, we aim to further improve the quality of pseudo labels and get better performance than these methods.

3. Our Method

3.1. Hierarchical Clustering with Hard-batch Triplet Loss

Our network structure is shown in Figure 2. The model is mainly divided into three stages: hierarchical clustering, PK sampling, and fine-tuning training. We extract image features to form a sample space and cluster samples step by step according to the bottom-up hierarchical clustering in Figure 1. After hierarchical clustering, we label samples in the same cluster with the same pseudo label. Finally, we use PK sampling to generate a new dataset for training according to clustering results. Our goal is to explore similarities among images in the target dataset through hierarchical clustering, distinguish hard examples through hard-batch triplet loss, and generate pseudo labels to guide model training in the end. Compared to other methods, our HCT can further improve the quality of pseudo labels and finally get better model performance.

For a dataset $X = \{x_1, x_2, \cdots, x_N\}$, we will have manually annotated labels $Y = \{y_1, y_2, \cdots, y_n\}$ in supervised learning, so we can directly use cross-entropy loss to optimize the model. However, we do not have any manually annotated labels in fully unsupervised re-ID, so we need to generate pseudo labels as supervision instead of using manually annotated labels. Although hierarchical clustering can fully explore the similarity of samples, build the underlying structure through a bottom-up clustering and generate some good pseudo labels. But due to the deficiency of hierarchical clustering, this strategy cannot effectively distinguish hard examples and will generate lots of false pseudo labels in merging. These false pseudo labels will mislead the optimization of model and limit model performance.

In order to solve this problem, HCT uses hard-batch triplet loss with PK sampling to reduce the distance between similar samples and increase the distance between different samples, which can better distinguish hard examples. Besides, we will initialize all pseudo labels at the beginning of each iteration so that we are able to correct all false pseudo labels generated in the previous iteration. Theoretically, as pseudo labels of hierarchical clustering are approaching manually annotated labels step by step, the model performance is approaching the baseline. Baseline represents the supervised learning method of hard-batch triplet loss.

3.2. Distance Measurement

For all clustering-guided re-ID [14, 6, 20], the quality of pseudo labels generated by clusters directly determines the performance of the model. For hierarchical clustering, the distance measurement method used in the merging stage decide how we choose clusters to merge and finally affects the clustering result and pseudo labels.

BUC [14] uses the minimum distance as the distance measurement in the merging stage. Minimum distance only calculates one pair of the nearest pairwise distance in two clusters. That is not a good method because it ignores other samples in clusters. Especially when there are lots of samples in a cluster, minimum distance is easily influenced by outliers and finally results in wrong merging and false pseudo labels. To improve the distance measurement and finally get a better result, we should consider the pairwise distance of all the samples in two clusters.

In HCT, we use euclidean distance to measure the distance between each sample. Then, according to the unweighted average linkage clustering (unweighted pair-group method with arithmetic means, UPGMA) [19], we define the distance between clusters as:

$$D_{ab} = \frac{1}{n_a n_b} \sum_{i \in C_a, j \in C_b} D\left(C_{a_i}, C_{b_j}\right) \tag{1}$$

where C_{a_i},C_{b_j} are two samples in the cluster C_a,C_b respectively. n_a,n_b represent the number of samples in C_a,C_b, $D(\cdot)$ means the euclidean distance. UPGMA takes into account all the pairwise distance between two clusters and each pairwise distance has the same weight. It effectively reduces the influence of outliers in sample space, promote more rational merging and finally get better results compared to other distance measurement according to discussion in [5].

3.3. Loss Function

Hard-batch triplet loss [9] is proposed to mine the relationship between $anchor$ with $positive\ sample$ and $negative\ sample$, which can reduce the distance between similar samples and increase the distance between different samples. In order to use hard-batch triplet loss in HCT, we use PK sampling to generate a new dataset for training. Specifically, We randomly select K instances from P identities for each mini-batch (batchsize = P×K). So our loss is defined as:

$$L_{triplet} = \sum_{i=1}^{P} \sum_{a=1}^{K} \Big[m + \overbrace{\max_{p=1\ldots K} D\left(x_a^i, x_p^j\right)}^{\text{hardest positive}} - \underbrace{\min_{\substack{j=1\ldots P \\ n=1\ldots N}} D\left(x_a^i, x_n^j\right)}_{\text{hardest negative}} \Big] \tag{2}$$

where x_a^i is the $anchor$, x_p^j is the $positive\ sample$ which has the same identity as x_a^i, x_n^j is the $negative\ sample$ which identity is different from x_a^i. $D(\cdot)$ means the euclidean distance and m is the hyperparameter $margin$ in hard-batch triplet loss. Hard-batch triplet loss makes sure that give an anchor x_a^i, x_p^j is closer to x_a^i than x_n^j. As a result, samples which have the same identity will be closer to

each other than other samples which have different identities. In other words, these samples will form a cluster gradually in high dimensional space. So we can use hard-batch triplet loss to distinguish hard examples, promote better clustering, and improve model performance.

3.4. Model Update

As shown in the algorithm, we use pre-trained ResNet-50 [8] on ImageNet for training. For each iteration, at the beginning of hierarchical clustering, we regard N samples as N different identities and initialize all pseudo labels. We set a hyperparameter mp to control the speed of the merging and a hyperparameter s represents total merging steps of hierarchical clustering, $m = n \times mp$ represents the number of clusters merged in each step. We calculate all pairwise distance between samples in the target dataset and generate a $n \times n$ distance matrix $dist$. According to $dist$ and UMPGA distance measurement in Eq.(1), we generate a $c \times c$ distance matrix $\boldsymbol{D}$, $\boldsymbol{D}$ represents the distance between clusters in each step, c represents the current number of clusters. We will merge m pairs of nearest clusters in each step until the $s\text{-}th$ step and generate pseudo labels according to the clustering result. Specifically, we regard samples in the same cluster have the same pseudo labels. Then we use PK sampling to generate a new dataset as the input of CNN, we conduct fine-tuning training with the new dataset and evaluate model performance in the end. We regard hierarchical clustering, PK sampling, fine-tuning training and evaluation as one iteration. We iterate the model until the performance no longer improves.

Algorithm 1 HCT Algorithm

Require:
Input $X = \{x_1, x_2, \cdots, x_N\}$;
Merging percent $mp \in (0, 1)$;
Merging step s;
Iteration t.

Ensure:
Best model $f(\boldsymbol{w}, x_i)$.

Initialize:
  sample number $n = N$,
  cluster number $c = n$,
  merging number $m = n \times mp$,
  iteration $iter = 0$,
  merging step $step = 0$.
while $iter < t$ **do**
  Initialize pseudo labels: $Y = \{y_i = i\}_{i=1}^{N}$;
  Extract features, calculate the pairwise distance between each sample, and generate a $n \times n$ distance matrix $dist$;
  while $step < s$ **do**
    Calculate distance between each cluster according to Eq.(1), generate a $c \times c$ distance matrix $\boldsymbol{D}$;
    Select clusters to merge according to $\boldsymbol{D}$ and start to merge clusters:
      $c = c - m$;
    Update Y with new pseudo labels:
      $Y = \{y_i = j, \quad \forall x_i \in C_j\}_{i=1}^{N}$;
      $step = step + 1$;
  end while
  Generate a new dataset with PK sampling according to Y;
  Fine-tuning model with the new dataset according to hard-batch triplet loss;
  Evaluate model performance;
  if $mAP_i > mAP_{best}$ **then**
    $mAP_{best} = mAP_i$;
    Best model $f(\boldsymbol{w}, x_i)$;
  end if
  $iter = iter + 1$;
end while

4. Experiment

4.1. Datasets

Market-1501 Market1501 [33] includes 32,668 images of 1501 pedestrians captured by 6 cameras. Each pedestrian is captured by at least two cameras. Market1501 can be divided into a training set which contains 12,936 images of 751 people and a test set which contains 19,732 images of 750 people.

DukeMTMC-reID DukeMTMC-reID [34] is a subset of pedestrian re-identification dataset DukeMTMC [17]. DukeMTMC contains a 85 minutes high-resolution video, which is collected from eight different cameras. DukeMTMC-reID consists of 36411 labelled images belonging to 1404 identities which contains 16,522 images for training, 2,228 images for query, and 17,661 images for gallery.

4.2. Implementation Details

HCT Training Setting We directly use pre-trained ResNet [8] on ImageNet as our backbone. After clustering, we randomly selected $P = 16$ identities and $K = 4$ images to generate a new train dataset, so $batchsize = P \times K = 64$. During the training, we adjust the size of the input image to 256×128, we also use random cropping, flipping and random erasing for data augmentation [35]. We use SGD to optimize the model and set a momentum [22] of 0.9 without dampening. The learning rate is 6×10^{-5}, the weight decay is 0.0005, iteration is 20, and $margin$ is 0.5 in hard-batch triplet loss. In Market-1501, merging percent mp is 0.07, merging step s is 13, epoch is 60. Note that the model is easily to overfit and will have a significant drop in the later iteration, we adopt an early stop strategy to get best perfor-

Methods	Labels	Market-1501				DukeMTMC-reID			
		rank-1	rank-5	rank-10	mAP	rank-1	rank-5	rank-10	mAP
Baseline[20]	Supervised	91.6	-	-	78.2	80.8	-	-	65.4
Direct transfer	None	11.1	22.1	28.6	3.5	8.6	16.4	21.0	3.0
HCT	None	80.0	91.6	95.2	56.4	69.6	83.4	87.4	50.7

Table 1. Comparison with baseline and direct transfer on Market-1501 and DukeMTMC-reID . "Baseline" means supervised learning method about hard-batch triplet loss. "Direct transfer" means directly use pre-trained ResNet-50 on ImageNet to evaluate without any fine-tuning. The label column lists the type of supervision used by the method. "Supervised" means supervised learning, "None" denotes no any manually annotated labels are used, which is fully unsupervised learning.

Methods	Labels	Market-1501				DukeMTMC-reID			
		rank-1	rank-5	rank-10	mAP	rank-1	rank-5	rank-10	mAP
UMDL[16]	Transfer	34.5	52.6	59.6	12.4	18.5	31.4	37.4	7.3
OIM[29]*	None	38.0	58.0	66.4	14.0	24.5	38.8	46.0	11.3
PUL[6]	Transfer	45.5	60.7	66.7	20.5	30.0	43.4	48.5	16.4
SPGAN[4]	Transfer	51.5	70.0	76.8	22.8	41.1	56.6	63.0	22.3
TJ-AIDL[26]	Transfer	58.2	74.8	81.1	26.5	44.3	59.6	65.0	23.0
HHL[36]	Transfer	62.2	78.8	84.0	31.4	46.9	61.0	66.7	27.2
BUC[14]	None	66.2	79.6	84.5	38.3	47.4	62.6	68.4	27.5
ARN[12]	Transfer	70.3	80.4	86.3	39.4	60.2	73.9	79.5	33.4
MAR[32]	Transfer	67.7	81.9	-	40.0	67.1	79.8	-	48.0
ECN[37]	Transfer	75.1	87.6	91.6	43.0	63.3	75.8	80.4	40.4
EANet[10]	Transfer	78.0	-	-	51.6	67.7	-	-	48.0
Theory[20]	Transfer	75.8	85.9	93.2	53.7	68.4	80.1	83.5	49.0
HCT	None	**80.0**	**91.6**	**95.2**	**56.4**	**69.6**	**83.4**	**87.4**	**50.7**

Table 2. Comparison with other unsupervised methods. The label column lists the type of supervision used by the method. "Transfer" means uses an manually annotated source dataset, which is UDA method. "*" means results are reported by [14]. Results that surpass all competing methods are **bold**.

mance.

Evaluating Setting We use the single-shot setting [21] in all experiments. In evaluation, for an image in query, we calculate cosine distance with all gallery images and then sort it as the result. We use the mean average precision (mAP) [33] and the rank-k accuracy to evaluate the performance of the model. Rank-k emphasizes the accuracy, it means the query picture has the match in the top-k list. Beside, mAP is computed from the Cumulated Matching Characteristics (CMC) [7]. CMC curve shows the probability that a query has the match in different size of lists. Given a single query, the Average Precision (AP) is computed according to its precision-recall curve, the mAP is the mean of AP.

4.3. Ablation Study

Comparison with Baseline and Direct Transfer In order to reflect the effect of our HCT, we compare HCT with a supervised learning method about hard-batch triplet loss and direct transfer from pre-trained ResNet-50 on ImageNet. Our results are reported in Table 1. The results for direct transfer and baseline represent the floor and upper limit of model performance. Theoretically, when the quality of our pseudo labels approach to manually annotated labels, HCT will gradually approach the baseline.

We can see that the performance of direct transfer is very poor, only get 3.5% mAP on Market-1501 and 3.0% mAP on DukeMTMC-reID. That is because the model is pre-trained on ImageNet for a classification task which is completely different from re-ID task. HCT outperforms the direct transfer method by 52.9% mAP on Market-1501 and 47.7% mAP on DukeMTMC-reID. That is only less than supervised method baseline 21.8% mAP and 14.7% mAP respectively, which indicates that the quality of pseudo labels generated by HCT is very high, so our model performance is good.

Effectiveness of HCT As shown in Table 2, we compare our HCT with other unsupervised methods. On Market-1501, we obtain **rank-1 =80.9%, mAP =56.4%**. On DukeMTMC-reID, we obtain **rank-1 =69.6%, mAP =50.7%**. HCT not only surpasses other fully unsupervised methods a lot, but also better than many UDA methods. Note that we do not use any manually labeled data for training, we just use unlabeled target data. Results indicates the

Merge step	Market-1501			
	IDs	epoch	rank-1	mAP
s = 12	2069	15	72.2	46.2
s = 13	1171	60	**80.0**	**56.4**
s = 14	258	300	×	×

Table 3. Performance comparison with different merging steps on Market-1501. "IDs" means identities number, it also represent the number of clusters after hierarchical clustering. "Epoch" means the training epoch in each iteration. "×" means the model is difficult to converge.

Merge percent	Market-1501			
	rank-1	rank-5	rank-10	mAP
mp = 0.04	79.6	90.9	94.6	55.3
mp = 0.05	78.7	91.1	94.6	55.0
mp = 0.06	78.1	91.1	94.2	54.3
mp = 0.07	**80.0**	**91.6**	**95.2**	**56.4**
mp = 0.08	77.0	90.4	94.1	53.0
mp = 0.09	77.9	90.8	94.2	54.6
mp = 0.1	77.4	90.9	94.6	53.7

Table 4. Performance comparison with different merging percents on Market-1501.

importance of fully exploring the similarity of the samples in the target domain. Besides, it also proves that hard-batch triplet loss can effectively reduce the influence of hard examples, further improve the quality of pseudo labels, and get better performance.

Comparison with Different Merging Steps In hierarchical clustering, merging step s controls the termination of merging, determines clusters number, and finally affects the quality of pseudo labels. In order to get the best performance, we set $mp = 0.07$ and evaluate the impact of different s on Market-1501. Our results are reported in Table 3. When we set $s = 14$, we find the model is difficult to converge even if we set the training epoch very high. Market-1501 have 751 IDs in the training set, but now HCT only have 258 IDs. We believe that in the final merge step, hierarchical clustering will generate lots of awful clusters and false pseudo labels that cannot be optimized. So we should adopt an early stop strategy in hierarchical clustering. However, too small s means too many IDs number of pseudo labels which will also cause problems. When we set $s = 12$, we can see a significant decline on performance. So too early stop will reduces the performance of the model. Besides, we have to decrease the training epoch to 15, because we find a large epoch easily leads to overfitting when s is too small. Finally, we get the best performance when we set $s = 13$.

Comparison with Different Merging Percents In hierarchical clustering, merging percent mp controls the speed of merging. It decides the number of clusters merged in each step and finally affects generated pseudo labels. In order to get the best performance and evaluate the influence of mp, we evaluate different mp values on Market-1501. Based on discussion above, we adopt an early stop strategy in our experiments for the setting of s in all experiments. In Table 4, we can see that when we set $mp = 0.07$, we get the best performance. We believe that both too many and too few merging in each step will result in a decline of cluster quality. Besides, compared to change merging step s, changing merging percent mp only causes a slight change in performance.

Qualitative Analysis of T-SNE Visualization As shown in Figure 3, we can see that BUC cannot effectively distinguish hard examples, so there are lots of $False\ Positive$ samples in clusters. These $False\ Positive$ samples are close to each other in high dimensional space and easily result in wrong merging in hierarchical clustering. Besides, the distribution of clustering results is dispersing and will generate many $False\ Negative$ samples. Different from hard examples, these $False\ Negative$ samples belong to the same identity. But they are not very close to each other in high dimensional space, so we cannot effectively use hierarchical clustering to merge them into one cluster. Our method HCT solves these problems and get better performance. We can see that HCT can promote more compact clustering, so the number of $False\ Negative$ samples is greatly reduced. Besides, HCT can effectively distinguish hard examples, so the number of $False\ Positive$ samples is also greatly reduced. These results illustrate the effectiveness of hard-batch triplet loss and the high quality of pseudo labels generated by HCT. In general, due to the significant improvement of clustering result, HCT surpasses a lot than other unsupervised methods.

5. Conclusion

In this paper, we propose a fully unsupervised re-ID method HCT. HCT directly use unlabeled dataset without using any manually annotated labels for training. We make full use of similarities between images in the target dataset through hierarchical clustering. We also effectively reduced the influence of hard examples in training by PK sampling and hard-batch triplet loss. Besides, we further improve the quality of generated pseudo labels through initializing pseudo labels and training alternately. Finally, as the quality of pseudo labels gradually improving, our model performance are improving step by step. Extensive experiments prove that HCT surpasses state-of-the-arts in fully unsupervised methods by a large margin, even better than most UDA methods.

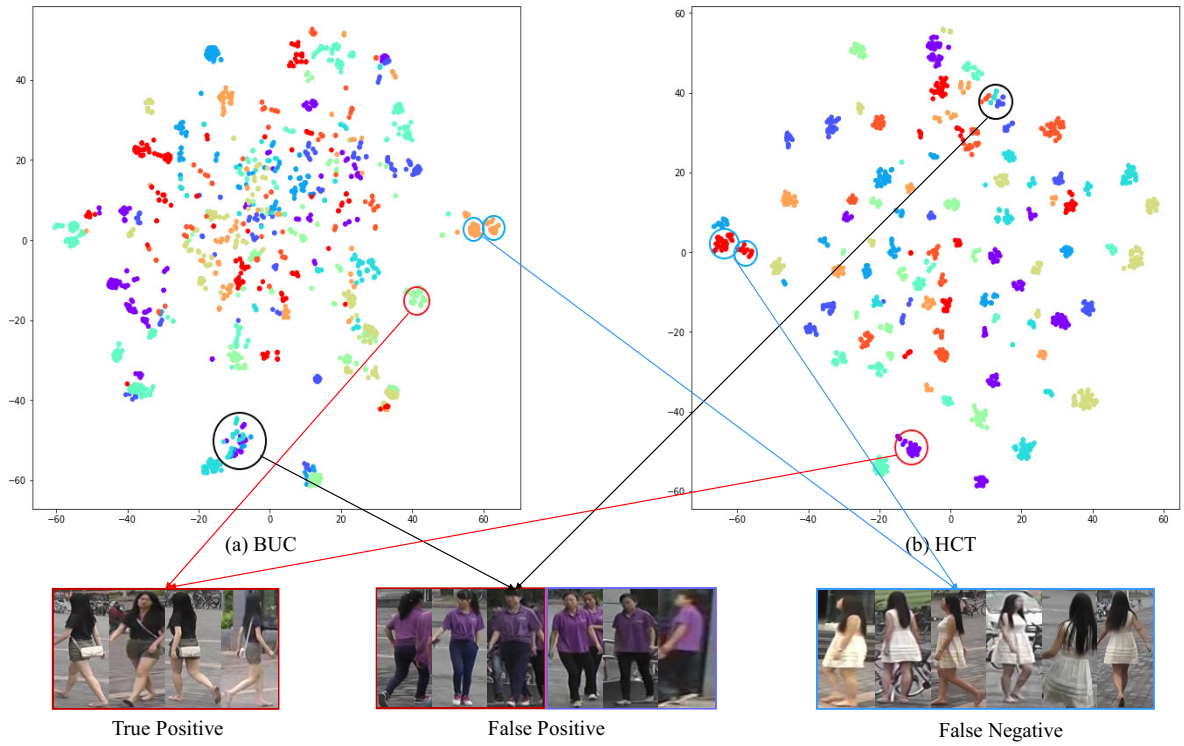

Figure 3. T-SNE visualization of the feature representation on a subset of Market-1501 about BUC (100 identities and 1747images) and HCT (100 identities and 1656 images). Samples with the same color represent them have same real labels. $True\ Positive$ means correct pseudo labels generated by model. $False\ Positive$ means model generate the same pseudo label for images that belong to different identities in fact. $False\ Negative$ means model generate different pseudo labels for images that belong to the same identity in fact. Both $False\ Positive$ and $False\ Negative$ will generate false pseudo labels which reduce the model performance.

6. Acknowledgements

We thank the reviewers for their feedback. We thank our group members for feedback and the stimulating intellectual environment they provide. This research was supported by The Science and Technology Planning Project of Hunan Province (No.2019RS2027) and National Key Research and Development Program of China (No.2018YFB0204301).

References

[1] Loris Bazzani, Marco Cristani, and Vittorio Murino. Symmetry-driven accumulation of local features for human characterization and re-identification. *Computer Vision and Image Understanding*, 2013.

[2] Xiaobin Chang, Timothy M Hospedales, and Tao Xiang. Multi-level factorisation net for person re-identification. In *CVPR*, 2018.

[3] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *CVPR*, 2018.

[4] Weijian Deng, Liang Zheng, Qixiang Ye, Guoliang Kang, Yi Yang, and Jianbin Jiao. Image-image domain adaptation with preserved self-similarity and domain-dissimilarity for person re-identification. In *CVPR*, 2018.

[5] Guodong Ding, Salman Khan, Zhenmin Tang, Jian Zhang, and Fatih Porikli. Towards better validity: Dispersion based clustering for unsupervised person re-identification. *arXiv preprint arXiv:1906.01308*, 2019.

[6] Hehe Fan, Liang Zheng, Chenggang Yan, and Yi Yang. Unsupervised person re-identification: Clustering and fine-tuning. *TOMM*, 2018.

[7] Douglas Gray, Shane Brennan, and Hai Tao. Evaluating appearance models for recognition, reacquisition, and tracking. In *Proc. IEEE International Workshop on Performance Evaluation for Tracking and Surveillance (PETS)*. Citeseer, 2007.

[8] 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.

[9] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017.

[10] Houjing Huang, Wenjie Yang, Xiaotang Chen, Xin Zhao, Kaiqi Huang, Jinbin Lin, Guan Huang, and Dalong Du. Eanet: Enhancing alignment for cross-domain person re-identification. *arXiv preprint arXiv:1812.11369*, 2018.

[11] Wei Li, Rui Zhao, Tong Xiao, and Xiaogang Wang. Deepreid: Deep filter pairing neural network for person re-identification. In *CVPR*, 2014.

[12] Yu-Jhe Li, Fu-En Yang, Yen-Cheng Liu, Yu-Ying Yeh, Xiaofei Du, and Yu-Chiang Frank Wang. Adaptation and re-identification network: An unsupervised deep transfer learning approach to person re-identification. In *CVPR*, 2018.

[13] Shengcai Liao, Yang Hu, Xiangyu Zhu, and Stan Z Li. Person re-identification by local maximal occurrence representation and metric learning. In *CVPR*, 2015.

[14] Yutian Lin, Xuanyi Dong, Liang Zheng, Yan Yan, and Yi Yang. A bottom-up clustering approach to unsupervised person re-identification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.

[15] Jianming Lv, Weihang Chen, Qing Li, and Can Yang. Unsupervised cross-dataset person re-identification by transfer learning of spatial-temporal patterns. In *CVPR*, 2018.

[16] Peixi Peng, Tao Xiang, Yaowei Wang, Massimiliano Pontil, Shaogang Gong, Tiejun Huang, and Yonghong Tian. Unsupervised cross-dataset transfer learning for person re-identification. In *CVPR*, 2016.

[17] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *ECCV*. Springer, 2016.

[18] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *ICML*, 2016.

[19] R.R. Sokal, C.D. Michener, and University of Kansas. *A Statistical Method for Evaluating Systematic Relationships*. University of Kansas science bulletin. University of Kansas, 1958.

[20] Liangchen Song, Cheng Wang, Lefei Zhang, Bo Du, Qian Zhang, Chang Huang, and Xinggang Wang. Unsupervised domain adaptive re-identification: Theory and practice. *arXiv preprint arXiv:1807.11334*, 2018.

[21] Yifan Sun, Liang Zheng, Yi Yang, Qi Tian, and Shengjin Wang. Beyond part models: Person retrieval with refined part pooling (and a strong convolutional baseline). In *ECCV*, 2018.

[22] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *ICML*, 2013.

[23] Hui Tian, Xiang Zhang, Long Lan, and Zhigang Luo. Person re-identification via adaptive verification loss. *Neurocomputing*, 359:93–101, 2019.

[24] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *NIPS*, 2016.

[25] Guanshuo Wang, Yufeng Yuan, Xiong Chen, Jiwei Li, and Xi Zhou. Learning discriminative features with multiple granularities for person re-identification. In *2018 ACM Multimedia Conference on Multimedia Conference*, 2018.

[26] Jingya Wang, Xiatian Zhu, Shaogang Gong, and Wei Li. Transferable joint attribute-identity deep learning for unsupervised person re-identification. In *CVPR*, 2018.

[27] Longhui Wei, Shiliang Zhang, Wen Gao, and Qi Tian. Person transfer gan to bridge domain gap for person re-identification. In *CVPR*, 2018.

[28] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, 2018.

[29] Tong Xiao, Shuang Li, Bochao Wang, Liang Lin, and Xiaogang Wang. Joint detection and identification feature learning for person search. In *CVPR*, 2017.

[30] Jing Xu, Rui Zhao, Feng Zhu, Huaming Wang, and Wanli Ouyang. Attention-aware compositional network for person re-identification. In *CVPR*, 2018.

[31] Jiaming Xu and En Zhu. Learning bias-free representation for large-scale person re-identification. *IEEE Access*, 7:143331–143346, 2019.

[32] Hong-Xing Yu, Wei-Shi Zheng, Ancong Wu, Xiaowei Guo, Shaogang Gong, and Jian-Huang Lai. Unsupervised person re-identification by soft multilabel learning. In *CVPR*, 2019.

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

[34] 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.

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

[36] Zhun Zhong, Liang Zheng, Shaozi Li, and Yi Yang. Generalizing a person retrieval model hetero-and homogeneously. In *ECCV*, 2018.

[37] Zhun Zhong, Liang Zheng, Zhiming Luo, Shaozi Li, and Yi Yang. Invariance matters: Exemplar memory for domain adaptive person re-identification. In *CVPR*, 2019.

[38] Zhun Zhong, Liang Zheng, Zhedong Zheng, Shaozi Li, and Yi Yang. Camera style adaptation for person re-identification. In *CVPR*, 2018.