Journal of Real-Time Image Processing (2023) 20:107
https://doi.org/10.1007/s11554-023-01362-z

RESEARCH

Real-time online unsupervised domain adaptation for real-world person re-identification

Christopher Neff[1] · **Armin Danesh Pazho**[1] · **Hamed Tabkhi**[1]

Received: 15 February 2023 / Accepted: 30 August 2023 / Published online: 24 September 2023

Abstract
Following the popularity of Unsupervised Domain Adaptation (UDA) in person re-identification, the recently proposed setting of Online Unsupervised Domain Adaptation (OUDA) attempts to bridge the gap toward practical applications by introducing a consideration of streaming data. However, this still falls short of truly representing real-world applications. This paper defines the setting of Real-world Real-time Online Unsupervised Domain Adaptation (R^2OUDA) for Person Re-identification. The R^2OUDA setting sets the stage for true real-world real-time OUDA, bringing to light four major limitations found in real-world applications that are often neglected in current research: system generated person images, subset distribution selection, time-based data stream segmentation, and a segment-based time constraint. To address all aspects of this new R^2OUDA setting, this paper further proposes Real-World Real-Time Online Streaming Mutual Mean Teaching (R^2MMT), a novel multi-camera system for real-world person re-identification. Taking a popular person re-identification dataset, R^2MMT was used to construct over 100 data subsets and train more than 3000 models, exploring the breadth of the R^2OUDA setting to understand the training time and accuracy trade-offs and limitations for real-world applications. R^2MMT, a real-world system able to respect the strict constraints of the proposed R^2OUDA setting, achieves accuracies within 0.1% of comparable OUDA methods that cannot be applied directly to real-world applications.

Keywords Person re-identification · Online learning · Unsupervised learning · Domain adaptation · Real-world · Real-time · Computer vision · Domain shift · Mutual mean teaching

1 Introduction

Person re-identification (ReID) is the task of matching a person in an image with other instances of that person in other images, either from the same camera or a different one. More specifically, it is associating a person's query with its match in a gallery of persons [45]. Person ReID is a common task in many real-world applications. Such applications include video surveillance (e.g., determining when unauthorized people are present in an area), public safety (e.g., understanding pedestrian motion to avoid accidents), and smart health (e.g., mobility assessment and fall detection for seniors needing assistance). Thus, achieving accurate and robust person ReID for any environment is an important research goal for the community.

Many methods have been developed for person ReID [18, 40, 51, 53], and many high quality datasets have been created for the task [25, 35, 42, 49, 52]. Deep learning approaches have been able to achieve incredible accuracies, nearly reaching saturation in some cases [32, 41, 43, 54]. However, person ReID is a highly context-specific task, and models trained on one dataset often fail to perform well on others [45]. Unsupervised Domain Adaptation (UDA) has been studied to combat this domain shift [2, 8, 28, 36, 42, 45]. In UDA, initial training is performed on the labeled data of the source domain, and then inference is done in a different target domain. UDA methods

This research is supported by the National Science Foundation (NSF) under Award No. 1831795 and NSF Graduate Research Fellowship Award No. 1848727.

✉ Armin Danesh Pazho
adaneshp@uncc.edu

Christopher Neff
cneff1@uncc.edu

Hamed Tabkhi
htabkhiv@uncc.edu

[1] University of North Carolina at Charlotte, Charlotte, NC, USA

generally achieve lower accuracies than State-of-the-Art (SotA) deep learning approaches that train directly on the target domain. However, recent approaches have begun to close that gap [11, 12, 48].

One common thread among these approaches is the reliance on having the entirety of the target domain available at training time. While this is convenient for research, many practical applications do not have unrestricted access to the entire target domain. Recently, [33] introduced the setting of Online Unsupervised Domain Adaptation (OUDA). OUDA specifies that data from the target domain can only be accessed through a data stream, bringing research more in line with real-world applications. OUDA adopts a batch-based relaxation [9] where different identities are separated among batches to simulate streaming data. OUDA also argues that confidentiality regulations make it such that many real-world applications can only store data for a limited amount of time, applying a restriction that image data cannot be stored beyond the batch in which it was collected.

Table 1 shows the challenges of real-world applications, and how UDA and OUDA fail to fully address them. Like UDA before it, OUDA uses hand-crafted person ReID datasets for the target domain. Not only is the data stream only simulated, but the provided person images were hand selected by the creators of the dataset. In a real-world system, person images need to be generated by the system itself, creating a layer of noise not present in hand-crafted datasets. Further, using hand-crafted datasets, the distribution of person images is guaranteed to be suitable for training. Specifically, most person ReID dataset tend to have a fairly uniform distribution, having around the same number of person images for each identity [26]. However, in real-world applications, there is no guarantee that person images generated from streaming data will form a uniform distribution in identities. There is also no guarantee that every identity in the dataset will be available for training. Additionally, in real-world applications, we often see multi-camera systems that rely on processing all this information in real-time. The UDA and OUDA settings do not address this.

To bring the field closer to the real-world, this paper proposes Real-World Real-Time Online Unsupervised Domain Adaptation (R^2OUDA), a setting designed to address the challenges found in real-world applications, as seen in Table 1. R^2OUDA defines four major considerations beyond the OUDA setting needed to develop systems for the real world. First, R^2OUDA considers that person images must be generated algorithmically from streaming data. Second, the distribution of data to be used in training must also be determined algorithmically. Third, R^2OUDA expands the batched-based relaxation [9] of online learning to use time segments, relating the conceptual mini-batch to the real-world notion of time inherent in streaming data. Fourth, R^2OUDA defines a time constraint such that the time spent training a single time segment cannot interfere with the training for subsequent time segments. The first two considerations address the noisy data inherent in real-world systems, while the last two considerations address the time-based streaming nature of data seen in real-time systems.

To address all aspects of the new R^2OUDA setting, this paper further proposes Real-World Real-Time Online Streaming Mutual Mean Teaching (R^2MMT). R^2MMT is an end-to-end multi-camera system designed for real-world person ReID. Using object detection, pedestrian tracking, human pose estimation, and a novel approach for Subset Distribution Selection (SDS), R^2MMT is able to generate person crops directly from a data stream, filter them based on representation quality, and create a subset with a suitable distribution for real-time training. To show the viability of R^2MMT to meet the challenges of real-world applications, and to explore the breadth of the R^2OUDA setting, an exhaustive set of experiments were conducted on the popular and challenging DukeMTMC dataset [35]. Using R^2MMT, over 100 data subsets were created and more than 3000 models were trained, capturing the trade-offs and limitations of real-world applications and the R^2OUDA setting. R^2MMT is a real-world system that can meet the demanding requirements of the proposed R^2OUDA setting, and is able to achieve over 73% Top-1 accuracy on DukeMTMC-reid, within 0.1% of comparable OUDA methods that cannot be directly applied for real-world applications.

To summarize, this paper's contributions are as follows:

Table 1 Challenges of real-world applications and if they are addressed in the UDA, OUDA, and R^2OUDA settings

Real-world	UDA	OUDA	R^2 OUDA (Ours)
Data from target domain is only available through a data stream	✗	✓ †	✓
Person crops are not provided and must be generated online	✗	✗	✓
There is no guarantee that every identity will be available during training	✗	✗	✓
The distribution of person crops must be determined online	✗	✗	✓
Training time must be accounted for	✗	✗	✓

† Streaming data is simulated

- We define the setting of Real-World Real-Time Online Unsupervised Domain Adaptation, accounting for the challenges of real-world applications and bridging the gap between research and application.
- We propose Real-World Real-Time Online Streaming Mutal Mean Teaching, a novel end-to-end multi-camera person ReID system designed to meet the challenges of R^{2}OUDA and real-world applications.
- We perform exhaustive experimentation, creating over 100 data subsets and training over 3000 models, to explore the breadth of the R^{2}OUDA setting and understand the trade-offs and limitations of real-world applications.

2 Related work

The UDA setting for person ReID has been extensively explored by the research community [24, 36, 45, 50]. In general, there are two main categories of algorithms used to perform UDA for person ReID: style transfer methods and target domain clustering methods.

2.1 Style transfer

Style transfer-based methods generally use Generative Adversarial Networks (GANs) [15] to perform image-to-image translation [20], modifying images from the source domain to look like the target domain without affecting the context of the original images. [4] uses self-similarity and domain-dissimilarity to ensure transferred images maintain cues to the original identity without matching to other identities in the target domain, while [14] introduces an online relation-consistency regularization term to ensure relations of the source domain are kept after transfer to the target domain. [28] separates transfers into factor-wise sub-transfers, across illumination, resolution, and camera view, to better fit the source images into the target domain. [2] uses a dual conditional GAN to transfer source domain images to multiple styles in the target domain, creating a multitude of training instances for each source identity. [42] uses a cycle consistent loss [55] with an emphasis on the foreground to better maintain identities between styles. [19] looks at domain shift as background shift and uses a GAN to remove backgrounds without damaging foregrounds, while a densely associated 2-stream network integrates identity-related cues present in backgrounds.

2.2 Target domain clustering

Target domain clustering approaches focus on using clustering algorithms to group features of the target domain for use as labels to fine tune a neural network pre-trained on the source domain [7]. This is usually done in an iterative fashion, where clustering is performed between training epochs to update the group labels as the model learns. [46] proposes using a dynamic graph matching framework to better handle large cross-camera variations. [10] introduces a self-similarity group to leverage part-based similarity to build clusters from different camera views. [26] utilizes a diversity regularization term to enforce a uniform distribution among the sizes of clusters. [13] introduces hybrid memory to dynamically generate instance-level supervisory signal for feature representation learning. [11] builds on [38], using two teacher models and their temporally averaged weights to produce soft pseudo labels for target domain clustering. [3] utilizes both target domain clustering and adversarial learning to create camera invariant features and improve target domain feature learning.

2.3 Online Unsupervised Domain Adaptation

While Online Unsupervised Domain Adaptation has been explored for other AI tasks [6, 16, 23, 29, 30, 39, 47], it was first defined for the field of person ReID in [33]. OUDA for Person ReID aims to create a practical online setting similar to that found in practical applications. OUDA builds upon the UDA setting by adding two considerations. First, data from the target domain is accessed via a data stream and not available all at once. Second, due to confidentiality concerns common in many countries, data from the target domain can only be stored for a limited time and only model parameters trained on that data may be persistent.

3 Proposed R^{2}OUDA setting

The proposed setting of Real-World Real-Time Online Unsupervised Domain Adaptation, building off OUDA [33], considers that we have access to a completely annotated source dataset D_S as well as partial access to an unlabeled target dataset D_T in the domain of our target application. In contrast to standard UDA, in both OUDA and R^{2}OUDA, the data from D_T is only accessible as an online stream of data. Whereas both UDA and OUDA use person crops from hand-crafted datasets, R^{2}OUDA specifies that person crops from D_T must be generated algorithmically from the data stream. This reflects how data is gathered in the real world. Where hand selected crops from datasets are generally highly representative, crops generated from a data stream will have varying levels of quality. This introduces noise in D_T, both in

quality and in the inevitable missed detections, which needs to be accounted for.

Additionally, hand-crafted datasets choose person images to fit a distribution suitable for training. However, since crops in R^{2}OUDA are generated from streaming data, such a distribution can not be assumed. This leads to the second consideration of R^{2}OUDA, that the distribution of data to be used in training must be determined algorithmically. Instead of relying on a predefined set of person images, systems must generate their own data subset, determining its size and distribution appropriately. This also reflects the real-world, as it is rarely known beforehand the amount and distribution of person crops that will be collected by an application.

Continuing with the batched-based relaxation [9] of the online learning scenario proposed in [33], we further introduce a time constraint for R^{2}OUDA. First, instead of separating our "mini-batches" ("tasks" as defined in [33]) across identities, since R^{2}OUDA requires actual streaming data, the data stream is separated into discrete time segments. We consider that for a chosen time segment of length τ, the streaming data will be divided into equal, non-overlapping time segments of length τ whose combined contents are equivalent to the original data stream.

For R^{2}OUDA, we must account both for applications that run continuously (i.e., the total length of the data stream is infinite) and the fact that, in the real world, computation resources are not unlimited. This leads to the necessity of a time constraint, but one that is not simple to define. Training time is inherently linked to hardware, and there are many techniques to hide latency or increase throughput in system design. As such, we simply define the time constraint such that, for any time segment τ_i, the length of time spent training on data collected during τ_i must be such to not interfere with the training for the data collected during τ_{i+1}. This is to prevent the training time deficit from increasing infinitely as i increases.

In summary, R^{2}OUDA introduces four new considerations to better match real-world applications:

- Person crops from the target domain must be generated algorithmically from a data stream.
- The selection and distribution of data to be used in training must be determined algorithmically.
- An expansion of the batch-based relaxation to use time segments, relating the conceptual mini-batch to the real-world notion of time inherent in streaming data.
- An additional time constraint such that the time spent training a single time segment cannot interfere with the training for any subsequent time segments.

4 Real-World Real-Time Online Streaming MMT

To address the challenges of R^{2}OUDA, we present Real-World Real-Time Online Streaming Mutual Mean Teaching, a novel multi-camera system for real-world person ReID. Similar to [31], R^{2}MMT is comprised of multiple Local Nodes and a single Global Node. Local nodes have access to the data stream directly from the cameras and are responsible for generating quality person images. The Global Node has access to all data generated by Local Nodes and is responsible for global ReID, subset distribution selection, and target domain training. An overview of R^{2}MMT can be seen in Fig. 1.

On the Local Node, YOLOv5 [22] is used as an object detector to find people in the video stream. Image crops are created for each person and sent to both a pose estimator

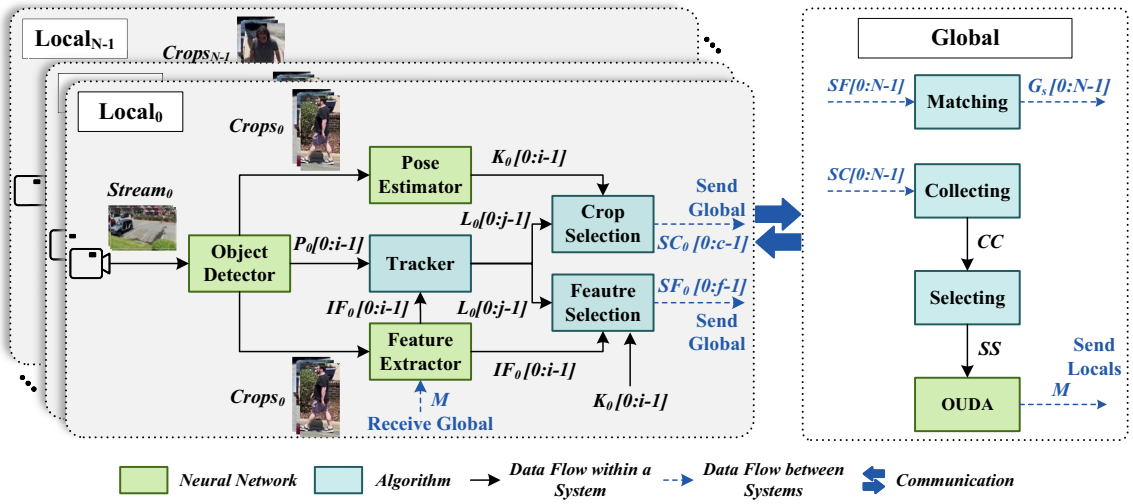

Fig. 1 System view of Real-World Real-Time Online Streaming Mutual Mean Teaching

(HRNet [37]) and a ReID feature extractor (ResNet-50 [17]). Coordinates for each person and features generated by the feature extractor are sent to a tracker [44] for local ReID. Afterward, feature and crop selection are performed to ensure that features and person crops sent to the Global Node for global ReID and crop collection are highly representative. This process utilizes person bounding box coordinates from the tracker to filter out any persons that have significant overlap (IoU ≥ 0.3) with other persons. This limits the number of crops used for training and features used for ReID containing multiple persons. The pose estimator is used to determine the quality of the features themselves. We reason that if a highly representative feature is present, then poses generated from the person crop should be of high confidence, while the number of keypoints present can help determine if there is significant occlusion or cutoff. Only crops and features with poses containing 15 or more keypoints (out of 17 total [27]) with at least 50% confidence are sent to the Global Node. This helps ensure that the quality of the crops used for training is similar to the quality of crops found in hand-crafted datasets.

On the Global Node, local identities and features are received from the Local Nodes and sent to a matching algorithm. This matching algorithm, as described in [31], performs global (i.e., multi-camera) ReID. Concurrently, person crops from all cameras are collected for a single time segment. Generally, far more features will be collected than can reasonably be used during training. For instance, when DukeMTMC-Video [35] is sampled every frame, the system produces over 4 million crops that pass feature selection. To reduce redundancy and computation, R^{2}MMT samples crops for selection once every 60 frames.

After all person crops from a single time segment are collected, the Subset Distribution Selection algorithm is used to create a subset that maintains a uniform distribution and number of crops suitable for training. R^{2}MMT uses an SDS algorithm based on the metric facility location problem [34]. We define that given a number of features in a metric space, we wish to find a subset of k features such that the minimum distance between any two features within the subset is maximized. However, this problem is known to be NP-hard [21], making it unsuitable for our real-world applications. R^{2}MMT instead uses a greedy implementation of the algorithm proven to be $\Omega(log\, k)$-competitive with the optimal solution while proving to be significantly faster, especially for larger sets of data [1]. For ease of readability, we adopt the nomenclature of K to mean the number of instances per identity. Therefore the total number of person crops in a subset k is equal to the number of identities in the dataset times K. To further reduce complexity, SDS is performed on the data from each camera individually, and their results are combined to form the complete subset. The SDS process helps ensure we have a uniform distribution of identities in our training data, similar to what is found in hand-crafted ReID datasets.

Once the training subset is created, domain adaptation is performed using Mutual Mean Teaching (MMT) [11]. R^{2}MMT follows the training methodology described in [11], except that epochs and iterations are variable. Clustering is done using DBSCAN [5], as GPU acceleration allows it to perform much faster than CPU-based approaches. Exact training parameters, both for pre-training on the source domain and domain transfer on the target domain, are as detailed in [11] unless otherwise noted.

Both SDS and training are time consuming, particularly when dealing with large amounts of data. To meet the time constraint of the R^{2}OUDA setting, R^{2}MMT utilizes a pipelined processing model, taking advantage of parallel computing resources while hiding the latency of the aforementioned tasks. An illustration of this pipelined approach can be seen in Fig. 2. Crop collection, SDS, and training are separated into their own pipeline stages. This means that while a model collects data for the current time segment, SDS on that data will occur the following time segment, and the training for that subset will occur the time segment after that. More formally, during a single time segment T_N, a model trained on data from T_{N-3} is used to collect data from time segment T_N, while subset distribution selection is performed on data collected during T_{N-1} and another network is being trained on a subset created from data from T_{N-2}. All of

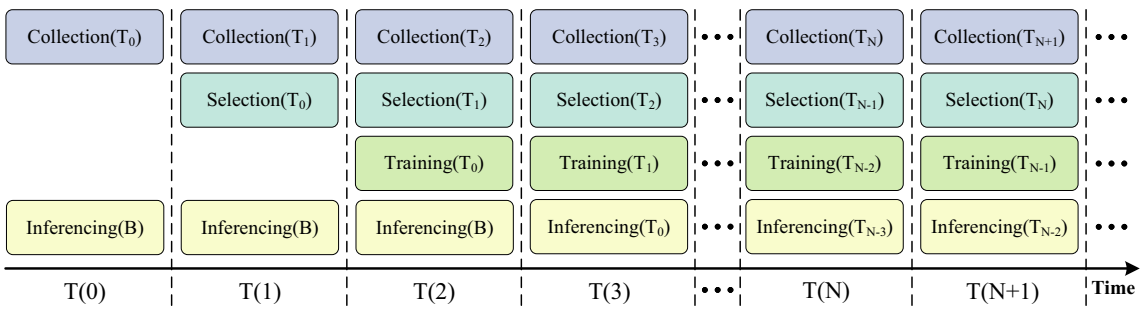

Fig. 2 Illustration of computation overlap through time

these processes will finish before T_{N+1}. This means there will always be a latency of two time segments between collection and inference for a single time segment. However, due to the pipeline structure, training throughput remains at a rate of one time segment per time segment. This satisfies the time constraint of R^{2}OUDA.

5 Experimental results

To explore the setting of R^{2}OUDA, we select the Market 1501 dataset [49] as the source domain and the DukeMTMC dataset [35] as the target domain. The DukeMTMC dataset is desirable as a target domain because it has both a video dataset (DukeMTMC-video) and a hand-crafted person ReID dataset (DukeMTMC-reid), both in the same domain. The video dataset is required in order to satisfy the streaming data constraint of the R^{2}OUDA setting. The hand-crafted ReID dataset brings two benefits. First, it allows us to directly observe the effect of noisy system generated crops compared hand selected person images when used for training. Second, testing on the ReID dataset allows direct comparison with works done in the UDA and OUDA space. As such, all our Top-1 accuracies are reported on the DukeMTMC-reid dataset. Similarly, we determining subset size, we treat the number of identities for both DukeMTMC-reid and DukeMTMC-video to be 702, as described in [35]. The number of person crops in a subset k is always equal to $k \times 702$.

For all experiments, R^{2}MMT is used to perform domain adaptation. Parameters in all experiments are the same as in [11], except where noted otherwise. All Local Nodes are run on a single server with two AMD EPYC 7513 CPUs, 256 GB of RAM, and three Nvidia V100 GPUs. The Global Node is run on a workstation with an AMD Threadripper Pro 3975WX CPU, 256 GB RAM, and three Nvidia RTX A6000 GPUs. All timing results presented in this section are using this Global Node.

5.1 Subset Distribution Selection

We first explore the effect of using our baseline Subset Distribution Selection algorithm for training on the DukeMTMC-reid dataset. Using hand-selected person crops from the dataset, we remove the effect of noise generated by our system and single out the impact of our SDS algorithm and the reduction in amount of data on domain adaptation. We vary the number of person images per identity K, iterations per epoch I, and total epochs E as shown below. Note that using the entire DukeMTMC-reid dataset would be equivalent to $K = 25$.

$$\begin{aligned} K &\in [2, 4, 6, 8, 10, 12, 14, 16, 18, 20] \\ I &\in [100, 250, 500, 750, 1000, 1500] \\ E &\in [1, 2, 3, 5]. \end{aligned} \tag{1}$$

These variable ranges lead to 240 training permutations, which is difficult to list in a single table. Instead, the results are plotted in a three-dimensional space and can be seen in Fig. 3. *Training Time* and *Top-1* make up the x and y axes, *Epochs* are the z axis, *Iterations* are noted by color, and k is indicated by size, with bigger circles representing higher values of k. As the purpose of these experiments is to focus on the effects of our SDS algorithm, the system pipeline described in Sect. 4 is ignored and timing results count SDS and training sequentially. More detailed information on these experiments can be found in the supplementary materials.

From these graphs, we can understand the general trend of the data. Intuitively, we see a fairly linear trend where more data generally results in higher Top-1 accuracy. Likewise,

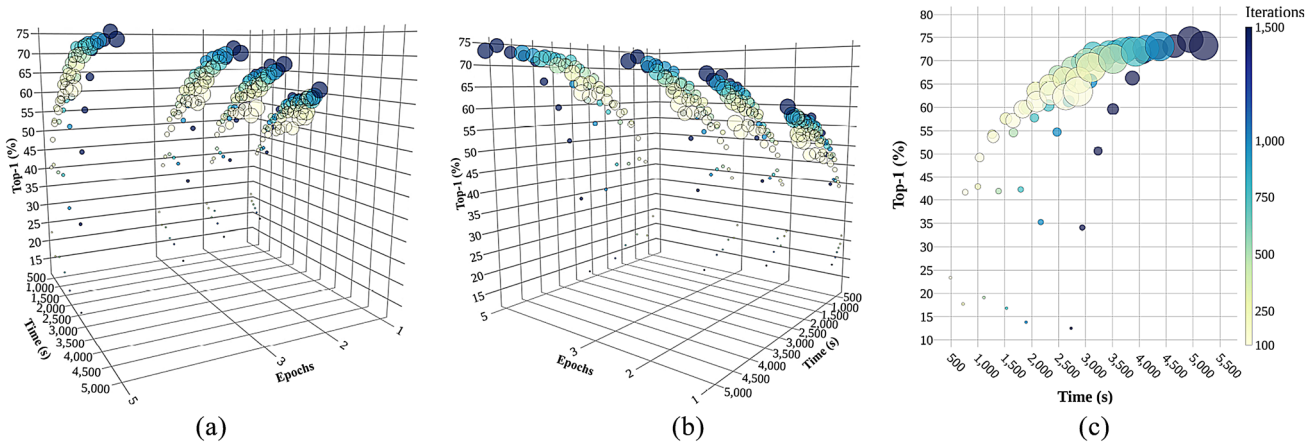

Fig. 3 Results exploring SDS on the hand-crafted DukeMTMC-reid dataset [35]. **a** and **b** Show two views of the results plotted in three-dimensional space, while (**c**) shows a two-dimensional view when $E = 5$. Larger circles represent larger values of k

more iterations per epoch and more epochs also tend to result in higher accuracy. Interestingly, with lower values of k we see the reverse effect; more time spent training results in decreased accuracy, sometimes even below the pre-trained accuracy of 42.0%. In general, at least 6 person images per identity are needed to consistently learn, while we start to see diminishing returns at around 16 person images per identity. The top result occurs when $K = 20$, $I = 1500$, and $E = 5$, achieving a Top-1 accuracy of 74.55% with a training time of 82 minutes. This is only 3.5% less than what comparable algorithms are able to achieve in the UDA setting [11] and over 2% greater than the same algorithm in the OUDA setting [33]. When using the same hardware, R^{2}MMT is 2.6× faster than its UDA counterpart.

5.2 System Generated Data

As explained in Sect. 3, one of the requirements of the R^2 OUDA setting is that person crops must be generated algorithmically from a data stream. As such, it is necessary to explore the effects of the noise this introduces. The structure of these experiments are exactly the same as in Sect. 5.1, except that instead of using DukeMTMC-reid, R^{2}MMT generates data from the DukeMTMC-video dataset. Similar to Sect. 5.1, we ignore the system pipeline and focus on the effects of the generated data. Based on the larger amount of data available in DukeMTMC-video, the ranges for our experimental variables are adjusted as shown below. Using all generated data would be equivalent to $K = 99$.

$$\begin{aligned} K &\in [16, 18, 20, 25, 30, 40] \\ I &\in [100, 250, 500, 1000, 1500] \\ E &\in [1, 2, 3, 5]. \end{aligned} \tag{2}$$

The results of this exploration can be seen in Fig. 4, with more details available in the supplementary materials. Axes are identical to Fig. 3, with color and size representing iterations and k respectively. These graphs show a somewhat similar trend as in Sect. 5.1 with some interesting deviations. While the trend starts off with accuracy increasing as k gets larger, there is a sharp decrease in accuracy when k increases beyond a certain point. The scale of the decrease, as well as how early it occurs, lessens with both iterations and epochs. This is likely a byproduct of how many identities are present in DukeMTMC-video. While DukeMTMC only labels a total of 1404 identities, our system is able to detect far more. Increasing iterations has such a drastic effect here because it determines how many of and how often these identities are seen during an epoch. Further increasing iterations and epochs could help mitigate this, but would also increase overall training time. This, combined with the fact that more epochs and more iterations always result in higher accuracy, suggests that accuracy saturation has not been reached here, and the main limiting factor is training time. The highest accuracy achieved on this noisy data was a Top-1 of 69.34%, with $K = 20$, $I = 1500$, $E = 5$, and a total training time of just under 57 minutes. This is notably worse than both the 74.55% achieved in Sect. 5.1 and the 72.3% MMT achieves in the OUDA setting [33]. This demonstrates the extreme impact noisy data can have on unsupervised domain adaptation, and why the extra considerations of the R^{2}OUDA setting are a necessity when designing algorithms for real-world applications.

5.3 R^{2}MMT

Finally, we make the first attempt at addressing the R^{2}OUDA setting. An exhaustive set of experiments are conducted with R^{2}MMT, producing a fully functional, end-to-end system that

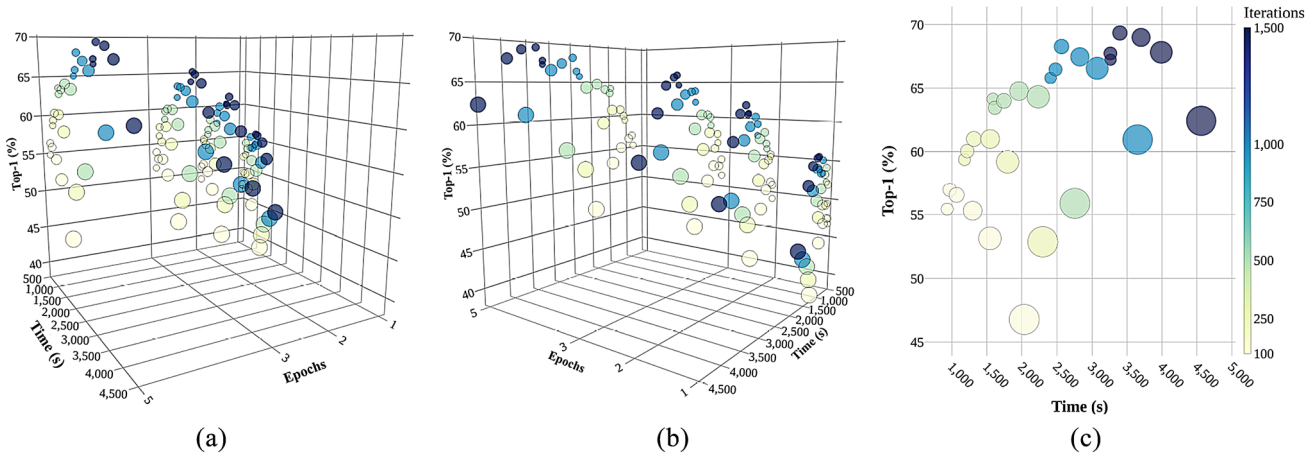

Fig. 4 Results exploring the use of system generated data using DukeMTMC-video [35]. **a** and **b** Show two views of the results plotted in three-dimensional space, while (**c**) shows a two-dimensional view when $E = 5$. Larger circles represent larger values of k

meets all the requirements of the R^{2}OUDA setting. R^{2}MMT generates person crops from a stream of data, uses SDS to construct training subsets, operates on the notion of time segments, and must adhere to the strict time constraint outlined in Sect. 3. A successful implementation will conform to all of those standards while achieving the highest accuracy possible, ideally within range of what was seen in Sect. 5.1.

One hour of DukeMTMC-video is used as the data stream, split into equal sized continuous segments of size τ. SDS is performed at each time segment on each camera individually, and k refers to the total number of person crops across all training subsets for the full hour. Two methods are used to determine the number of crops needed at each time segment. In the standard method, only data collected in a time segment may be used for training related to that time segment. The second method uses a form of memory, allowing the use of data from the current time segment and previous time segments still in memory. For these experiments, we assume a memory length of up to 60 min. Equations 3 and 4 are used to calculate the number of person crops needed from each camera at each time segment, for the standard and memory-based methods respectively.

$$k = \sum_{t=0}^{\frac{60}{\tau}-1} \sum_{i=1}^{8} P(C_i)P(C_i \cap \tau_t), \tag{3}$$

$$k = \sum_{t=0}^{\frac{60}{\tau}-1} \sum_{i=1}^{8} P(C_i) \sum_{\eta=0}^{t} P(C_i \cap \tau_\eta), \tag{4}$$

where k is the total number of person crops desired for the training subset over an hour of video stream, τ_t is a time segment of length τ minutes that begins at $\tau \times t$ minutes, C_i is the i^{th} camera, $P(C_i)$ is the percentage of total person crops received from C_I when compared to all cameras over an hour of video, and $P(C_i)P(C_i \cap \tau_t)$ is the percentage of person crops received during τ_t for C_i compared to all person crops received from C_i over an hour of video.

This ensures the number of person crops selected for a subset from each camera at each time segment is proportional to the number of person crops received. The variable ranges used in these experiments are shown below.

$$\begin{aligned} K &\in [18, 20, 25, 30, 40, 50] \\ I &\in [100, 250, 500, 750, 1000, 1500] \\ E &\in [1, 2, 3, 5] \\ \tau &\in [15, 20, 30] \\ t &\in \mathbb{Z} : \{0 \le t \le (\frac{60}{\tau} - 1)\}. \end{aligned} \tag{5}$$

This creates over 2500 data points across the two methods, becoming difficult to visualize even in three-dimensional space. Figure 5 displays the distribution of training accuracies for each τ at each time segment. Out of the 864 configurations tested, more than half of them failed to consistently meet the time requirement of R^{2}OUDA and are not included in the statistics. Most notably, all configurations that used memory failed to consistently meet the time requirement when given a τ of 15. When memory is utilized, the time required for SDS greatly increases for successive time segments as more images accumulate. This limits how large k can be, restricting K to 20 or below when $\tau = 20$ and 30 or below when $\tau = 30$. Even without memory, the time constraint proves very limiting. Only when $\tau = 20$ is the entire range of K able to be utilized. For a more fine grain look at all 2500+ data points in this experiment, please see the supplementary materials.

The data in general follows similar trends as seen in Sects. 5.1 and 5.2, but to more of an extreme. In addition to disqualifying several configurations off the bat, the segmented data stream and time constraint generally mean R^{2}MMT has less data to work with during any given training. Unlike in the previous experiments, the time constraint prevents the system from just throwing more data and more training at the problem. Instead, a balance must be found. We see an overall increase in top accuracies when τ increases, both in standard and memory configurations. Top accuracies also increase over time, with one notable exception. When $\tau = 15$, accuracy actually drops in the final time segment. This is due to the extremely low amount of data available in that particular time segment.

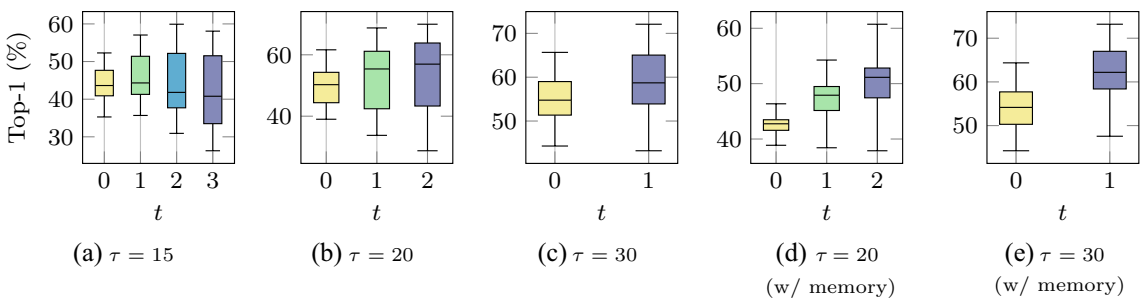

Fig. 5 Distribution of accuracies achieved on DukeMTMC [35] with R^{2}MMT

Table 2 Distribution of accuracies achieved on DukeMTMC [35] with R^{2}MMT

τ	t	Min	Q_1	Q_2	Q_3	Max
R^{2}MMT						
15	0	35.28	40.93	43.76	47.80	**52.29**
	1	35.68	41.43	44.48	51.66	**57.05**
	2	30.92	37.75	41.97	52.74	**59.92**
	3	26.30	33.62	40.89	51.71	**58.08**
20	0	39.00	44.39	50.27	54.29	**61.63**
	1	33.75	42.42	55.39	61.15	**68.76**
	2	28.73	43.31	56.96	63.85	**69.97**
30	0	44.30	51.35	54.76	59.04	**65.66**
	1	43.22	53.91	58.71	65.04	**72.08**
R^{2}MMT with memory						
20	0	38.87	41.67	42.77	43.65	**46.36**
	1	38.42	45.20	48.03	49.87	**54.26**
	2	37.88	47.44	51.35	53.90	**60.73**
30	0	44.26	50.30	54.17	57.72	**64.36**
	1	47.58	58.39	62.17	67.00	**73.21**

Another interesting observation can be made by looking at $\tau = 20$ both with and without memory. While the standard R^2 MMT achieves higher overall top accuracies, the distribution is a lot more varied when compared to R^2 MMT with memory. Many configurations actually lose accuracy, far more than when memory is present. This suggests that while memory is limiting, it may add stability to training over time. This is further demonstrated when $\tau = 30$. When memory is used the maximum accuracy is lower in the first time segment, being restricted to a lower value of K, but is higher in the second time segment due to the increased range of available data.

Table 2 and Figure 6 show the best configurations of R^{2}MMT, both with and without memory, for each τ. The overall highest accuracy is achieved with memory when $\tau = 30$, $K = 30$, $E = 5$, and $I = 500$, reaching an impressive 73.2% Top-1. Despite the much harsher requirements of the R^{2}OUDA setting, this is within 0.1% of the best possible accuracy using MMT in the OUDA setting [33]. However, with a τ of 30 it also has a latency of 60 min between collecting data and inferencing with a model trained on that data. This can be reduced to 30 min by changing τ to 15, but then accuracy drops to a disappointing 58.08%. Interestingly, with a τ of 15, accuracy actually drops in the final time segment. This is due to a limited number of persons present in the dataset during that time, leading to less data available for training and making the model less generalizable. A τ of 20 splits the difference, achieving a final Top-1 of 69.97% while reducing the inference latency to 40 min. This is within 4% of our best overall result, and reduces the delay by over 30%.

The strict time constraint disqualified many of the configurations in Sect. 5.3. However, if we ignore the time constraint for a moment, we see accuracies reaching up to 76.53% when $\tau = 15$, $K = 40$, $E = 5$, and $I = 1500$ in a system with memory, putting it within 1.5% of MMT in the UDA setting [11]. With further optimization or more powerful hardware, R^{2}MMT might be able to achieve higher accuracies with decreased latency between collection and inference. This shows that there is a lot of room for improvement and growth in the R^{2}OUDA setting. Overall, it is clear that larger values of K and more training time lead to better results, but the time constraint limits both of these factors. For any practical implementation, a balance must be found for that specific use case. The explorations in this paper can serve as a guideline for future works. The specific optimal ranges for each variable will shift with different target domains, but the overall trends and optimization techniques will be consistent.

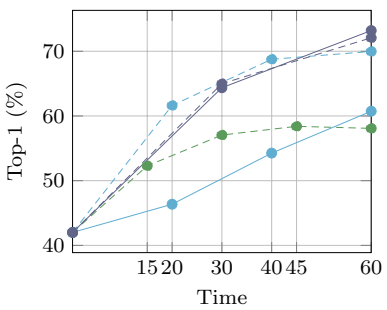

Fig. 6 Best results for each system configuration. Dashed lines (- -) represent standard configurations. Solid lines (–) represent configurations with memory. Green, blue, and purple denote τ values of 15, 20, and 30 respectively

6 Conclusion

This paper proposed the setting of R^2OUDA, to better represent the unique challenges of real-world applications. R^2MMT was introduced as the first attempt at a real-world, end-to-end system that can address all the demands of the R^2OUDA setting. An exhaustive set of experiments were conducted, using R^2MMT to create over 100 data subsets and train more than 3000 models, exploring the breadth of the R^2OUDA setting. While meeting the harsh requirements of R^2OUDA, including noisy data and time constraints, R^2MMT was able to achieve over 73% Top-1 accuracy, reaching within 0.1% of comparable SotA OUDA approaches without noisy data or a time constraint, that cannot be directly applied to real-world applications.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s11554-023-01362-z.

Acknowledgements This research is supported by the National Science Foundation (NSF) under Award No. 1831795 and NSF Graduate Research Fellowship Award No. 1848727.

Author Contributions All authors have been involved in the research and writing the manuscript. The amount of contribution is based on the order of the authors. All authors reviewed the manuscript.

Declarations

Conflict of interest The authors declare no competing interests.

References

1. Arthur, D., Vassilvitskii, S.: K-means++: The advantages of careful seeding. In: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '07, pp. 1027–1035. Society for Industrial and Applied Mathematics, USA (2007)
2. Chen, Y., Zhu, X., Gong, S.: Instance-guided context rendering for cross-domain person re-identification. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 232–242 (2019). https://doi.org/10.1109/ICCV.2019.00032
3. Delorme, G., Xu, Y., Lathuiliere, S., Horaud, R., Alameda-Pineda, X.: Canu-reid: a conditional adversarial network for unsupervised person re-identification. In: 2020 25th International Conference on Pattern Recognition (ICPR), pp. 4428–4435. IEEE Computer Society, Los Alamitos, CA, USA (2021). https://doi.org/10.1109/ICPR48806.2021.9412431. https://doi.ieeecomputersociety.org/10.1109/ICPR48806.2021.9412431
4. Deng, W., Zheng, L., Ye, Q., Kang, G., Yang, Y., Jiao, J.: Image-image domain adaptation with preserved self-similarity and domain-dissimilarity for person re-identification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
5. Ester, M., Kriegel, H.P., Sander, J., Xu, X., et al.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: kdd, vol. 96, pp. 226–231 (1996)
6. Ewen, N., Khan, N.: Online unsupervised learning for domain shift in Covid-19 CT scan datasets. In: 2021 IEEE International Conference on Autonomous Systems (ICAS), pp. 1–5 (2021). https://doi.org/10.1109/ICAS49788.2021.9551146
7. Fan, H., Zheng, L., Yan, C., Yang, Y.: Unsupervised person re-identification: clustering and fine-tuning. ACM Trans. Multimedia Comput. Commun. Appl. (2018). https://doi.org/10.1145/3243316
8. Feng, H., Chen, M., Hu, J., Shen, D., Liu, H., Cai, D.: Complementary pseudo labels for unsupervised domain adaptation on person re-identification. IEEE Trans. Image Process. **30**, 2898–2907 (2021). https://doi.org/10.1109/TIP.2021.3056212
9. Fini, E., Lathuilière, S., Sangineto, E., Nabi, M., Ricci, E.: Online continual learning under extreme memory constraints. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M. (eds.) Computer Vision - ECCV 2020, pp. 720–735. Springer International Publishing, Cham (2020)
10. Fu, Y., Wei, Y., Wang, G., Zhou, Y., Shi, H., Huang, T.S.: Self-similarity grouping: a simple unsupervised cross domain adaptation approach for person re-identification. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2019)
11. Ge, Y., Chen, D., Li, H.: Mutual mean-teaching: pseudo label refinery for unsupervised domain adaptation on person re-identification. In: International Conference on Learning Representations (2020). https://openreview.net/forum?id=rJlnOhVYPS
12. Ge, Y., Zhu, F., Chen, D., Zhao, R., Li, H.: Self-paced contrastive learning with hybrid memory for domain adaptive object re-id. In: Advances in Neural Information Processing Systems (2020)
13. Ge, Y., Zhu, F., Chen, D., Zhao, R., Li, H.: Self-paced contrastive learning with hybrid memory for domain adaptive object re-id. In: Advances in Neural Information Processing Systems (2020)
14. Ge, Y., Zhu, F., Chen, D., Zhao, R., Wang, X., Li, H.: Structured domain adaptation with online relation regularization for unsupervised person re-id (2020). https://doi.org/10.48550/ARXIV.2003.06650. arXiv:2003.06650
15. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, K. Weinberger (eds.) Advances in Neural Information Processing Systems, vol. **27**. Curran Associates, Inc. (2014). https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf
16. He, W., Ye, Y., Li, Y., Pan, T., Lu, L.: Online cross-subject emotion recognition from ECG via unsupervised domain adaptation. In: 2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), pp. 1001–1005 (2021). https://doi.org/10.1109/EMBC46164.2021.9630433
17. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 770–778 (2016)
18. Hermans*, A., Beyer*, L., Leibe, B.: In Defense of the Triplet Loss for Person Re-Identification. arXiv preprint arXiv:1703.07737 (2017)
19. Huang, Y., Wu, Q., Xu, J., Zhong, Y.: Sbsgan: suppression of inter-domain background shift for person re-identification. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 9526–9535. IEEE Computer Society, Los Alamitos, CA, USA (2019). https://doi.org/10.1109/ICCV.2019.00962. https://doi.ieeecomputersociety.org/10.1109/ICCV.2019.00962
20. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.: Image-to-image translation with conditional adversarial networks. In: 2017 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5967–5976 (2017). https://doi.org/10.1109/CVPR.2017.632
21. Jali, N., Karamchandani, N., Moharir, S.: Greedy kk-center from noisy distance samples. IEEE Trans. Signal Inf. Process. Netw. **8**, 330–343 (2022). https://doi.org/10.1109/TSIPN.2022.3164352

22. Jocher, G., Chaurasia, A., Stoken, A., Borovec, J., NanoCode012, Kwon, Y., TaoXie, Michael, K., Fang, J., imyhxy, Lorna, Wong, C., Yifu, Z., V, A., Montes, D., Wang, Z., Fati, C., Nadar, J., Laughing, UnglvKitDe, tkianai, yxNONG, Skalski, P., Hogan, A., Strobel, M., Jain, M., Mammana, L., xylieong: ultralytics/yolov5: v6.2 - YOLOv5 Classification Models, Apple M1, Reproducibility, ClearML and Deci.ai integrations (2022). https://doi.org/10.5281/zenodo.7002879
23. Kuznietsov, Y., Proesmans, M., Gool, L.V.: Towards unsupervised online domain adaptation for semantic segmentation. In: 2022 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW), pp. 261–271 (2022). https://doi.org/10.1109/WACVW54805.2022.00032
24. Leng, Q., Ye, M., Tian, Q.: A survey of open-world person re-identification. IEEE Trans. Circ. Syst. Video Technol. **30**(4), 1092–1108 (2020). https://doi.org/10.1109/TCSVT.2019.2898940
25. Li, W., Zhao, R., Xiao, T., Wang, X.: Deepreid: deep filter pairing neural network for person re-identification. In: CVPR (2014)
26. Lin, Y., Dong, X., Zheng, L., Yan, Y., Yang, Y.: A bottom-up clustering approach to unsupervised person re-identification. Proceedings of the AAAI Conference on Artificial Intelligence **33**(01), 8738–8745 (2019). https://doi.org/10.1609/aaai.v33i01.33018738. https://ojs.aaai.org/index.php/AAAI/article/view/4898
27. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) Computer Vision - ECCV 2014, pp. 740–755. Springer International Publishing, Cham (2014)
28. Liu, J., Zha, Z.J., Chen, D., Hong, R., Wang, M.: Adaptive transfer network for cross-domain person re-identification. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7195–7204 (2019). https://doi.org/10.1109/CVPR.2019.00737
29. Moon, J.H., Das, D., Lee, C.G.: Multi-step online unsupervised domain adaptation. In: ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 41172–41576 (2020). https://doi.org/10.1109/ICASSP40776.2020.9052976
30. Moon, J., Das, D., George Lee, C.S.: A multistage framework with mean subspace computation and recursive feedback for online unsupervised domain adaptation. IEEE Trans. Image Process. **31**, 4622–4636 (2022). https://doi.org/10.1109/TIP.2022.3186537
31. Neff, C., Mendieta, M., Mohan, S., Baharani, M., Rogers, S., Tabkhi, H.: Revamp2t: real-time edge video analytics for multicamera privacy-aware pedestrian tracking. IEEE Internet Things J. **7**(4), 2591–2602 (2020). https://doi.org/10.1109/JIOT.2019.2954804
32. Ni, X., Rahtu, E.: Flipreid: closing the gap between training and inference in person re-identification. In: 2021 9th European Workshop on Visual Information Processing (EUVIP), pp. 1–6 (2021). https://doi.org/10.1109/EUVIP50544.2021.9484010
33. Rami, H., Ospici, M., Lathuilière, S.: Online unsupervised domain adaptation for person re-identification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, pp. 3830–3839 (2022)
34. Rana, R., Garg, D.: Heuristic approaches for k-center problem. In: 2009 IEEE International Advance Computing Conference, pp. 332–335 (2009). https://doi.org/10.1109/IADCC.2009.4809031
35. Ristani, E., Solera, F., Zou, R., Cucchiara, R., Tomasi, C.: Performance measures and a data set for multi-target, multi-camera tracking. In: European Conference on Computer Vision workshop on Benchmarking Multi-Target Tracking (2016)
36. Song, L., Wang, C., Zhang, L., Du, B., Zhang, Q., Huang, C., Wang, X.: Unsupervised domain adaptive re-identification: theory and practice. Pattern Recogn. (2020). https://doi.org/10.1016/j.patcog.2019.107173
37. Sun, K., Xiao, B., Liu, D., Wang, J.: Deep high-resolution representation learning for human pose estimation. In: CVPR (2019)
38. Tarvainen, A., Valpola, H.: Mean teachers are better role models: weight-averaged consistency targets improve semi-supervised deep learning results. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17, pp. 1195–1204. Curran Associates Inc., Red Hook, NY, USA (2017)
39. Termöhlen, J.A., Klingner, M., Brettin, L.J., Schmidt, N.M., Fingscheidt, T.: Continual unsupervised domain adaptation for semantic segmentation by online frequency domain style transfer. In: 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), pp. 2881–2888 (2021). https://doi.org/10.1109/ITSC48978.2021.9564566
40. Wang, Y., Chen, Z., Wu, F., Wang, G.: Person re-identification with cascaded pairwise convolutions. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1470–1478 (2018). https://doi.org/10.1109/CVPR.2018.00159
41. Wang, G., Lai, J., Huang, P., Xie, X.: Spatial-temporal person re-identification. Proceedings of the AAAI Conference on Artificial Intelligence **33**(01), 8933–8940 (2019). https://doi.org/10.1609/aaai.v33i01.33018933. https://ojs.aaai.org/index.php/AAAI/article/view/4921
42. Wei, L., Zhang, S., Gao, W., Tian, Q.: Person transfer gan to bridge domain gap for person re-identification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
43. Wieczorek, M., Rychalska, B., Dabrowski, J.: On the unreasonable effectiveness of centroids in image retrieval. ArXiv arXiv:abs/2104.13643 (2021)
44. Wojke, N., Bewley, A., Paulus, D.: Simple online and realtime tracking with a deep association metric. In: 2017 IEEE International Conference on Image Processing (ICIP), pp. 3645–3649 (2017). https://doi.org/10.1109/ICIP.2017.8296962
45. Ye, M., Shen, J., Lin, G., Xiang, T., Shao, L., Hoi, S.C.H.: Deep learning for person re-identification: a survey and outlook (2020). https://doi.org/10.48550/ARXIV.2001.04193. arxiv:2001.04193
46. Ye, M., Li, J., Ma, A.J., Zheng, L., Yuen, P.C.: Dynamic graph co-matching for unsupervised video-based person re-identification. IEEE Trans. Image Process. **28**(6), 2976–2990 (2019). https://doi.org/10.1109/TIP.2019.2893066
47. Ye, Y., Pan, T., Meng, Q., Li, J., Shen, H.T.: Online unsupervised domain adaptation via reducing inter- and intra-domain discrepancies. IEEE Trans. Neural Netw. Learn. Syst. (2022). https://doi.org/10.1109/TNNLS.2022.3177769
48. Zeng, K.: Hierarchical clustering with hard-batch triplet loss for person re-identification (2019). https://doi.org/10.48550/ARXIV.1910.12278. arxiv:1910.12278
49. Zheng, L., Shen, L., Tian, L., Wang, S., Wang, J., Tian, Q.: Scalable person re-identification: a benchmark. In: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 1116–1124 (2015). https://doi.org/10.1109/ICCV.2015.133
50. Zheng, L., Yang, Y., Hauptmann, A.: Person re-identification: past, present and future. ArXiv arXiv:abs/1610.02984 (2016)
51. Zheng, L., Zhang, H., Sun, S., Chandraker, M., Yang, Y., Tian, Q.: Person re-identification in the wild. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3346–3355 (2017). https://doi.org/10.1109/CVPR.2017.357
52. Zheng, L., Bie, Z., Sun, Y., Wang, J., Su, C., Wang, S., Tian, Q.: Mars: A video benchmark for large-scale person re-identification. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) Computer Vision - ECCV 2016, pp. 868–884. Springer International Publishing, Cham (2016)
53. Zhong, Z., Zheng, L., Cao, D., Li, S.: Re-ranking person re-identification with k-reciprocal encoding. In: Proceedings of the IEEE

Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
54. Zhu, Z., Jiang, X., Zheng, F., Guo, X., Huang, F., Zheng, W., Sun, X.: Viewpoint-aware loss with angular regularization for person re-identification (2019). https://doi.org/10.48550/ARXIV.1912.01300. arxiv:1912.01300
55. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 2242–2251 (2017). https://doi.org/10.1109/ICCV.2017.244