KNOWLEDGE DISTILLATION FOR MULTI-TARGET DOMAIN ADAPTATION IN REAL-TIME PERSON RE-IDENTIFICATION: SUPPLEMENTARY MATERIAL

Félix Remigereau, Djebril Mekhazni, Sajjad Abdoli, Le Thanh Nguyen-Meidine, Rafael M. O. Cruz and Eric Granger

Laboratoire d'Imagerie, de Vision et d'Intelligence Artificielle (LIVIA)
Dept. of Systems Engineering, École de technologie supérieure, Montreal, Canada

1. SUPPLEMENTARY MATERIAL

1.1. Resnet Experiments

Table 1 shows the accuracy of our proposed approach when compared to the baseline and state-of-the-art techniques. Multiple STDA Models represents the case where we train a model for each target domain, it is equivalent to the performance of the teacher models for our KD-REID approach. STDA on blended datasets and KD-REID correspond to the approaches a) and b) in figure **??** respectively. The first thing to notice is that while multiple STDA models offer a good accuracy, they requires significantly more computational resources. Not only is each of the models a Resnet50 instead of a Resnet18 (less than half the parameters) but the number of models scales with the number of target domains. This solution is inappropriate for a real-world surveillance scenario where the number of targets is high and the resources are limited. In this regard, the blended datasets and KD-REID approaches both only require to keep a single Resnet18 in memory.

When the base STDA technique used is D-MMD, our method outperforms the blending approach on every dataset and yields the highest average performance for the complexity. The results differ when using SPCL a base STDA technique as blending becomes the best approach. The average performance however is quite low as SPCL does not perform well on the CUHK03 dataset. This might be explained by the fact CUHK03 is a much smaller dataset with few cameras. SPCL teachers, however, performs better on Market1501 and DukeMTMCReID than their D-MMD counterparts. Table **??** shows how we can use our proposed approach to take advantage of both types of teachers at the same time. We can see that by using the best available teacher for each target domain we obtain the highest average accuracy for the complexity. This shows the superior versatility of our approach compared to blended datasets approach which cannot combine STDA techniques.

1.2. Impact of Student's Capacity:

Evaluating the robustness to reduced model complexity is important given the limited resource available in surveillance applications. Figure 1 shows the difference in performance of the blending and KD-REID methods for different common backbone architectures. We notice that performance increases for higher model complexities up to a certain complexity. We see that accuracy increases when going from a Resnet18 to Resnet34 and also from a Resnet34 to Resnet50. Going from a Resnet50 to Resnet101 does not have a big impact on accuracy. It seems the Resnet50 architecture is sufficiently complex to store information on all targets and additional complexity beyond a certain point is inefficient. Going from Resnet50 to Resnet18 represents a reduction of over 50% of the number of parameters in the model and translates to only 3% drop in accuracy. This shows our method can use smaller models efficiently.

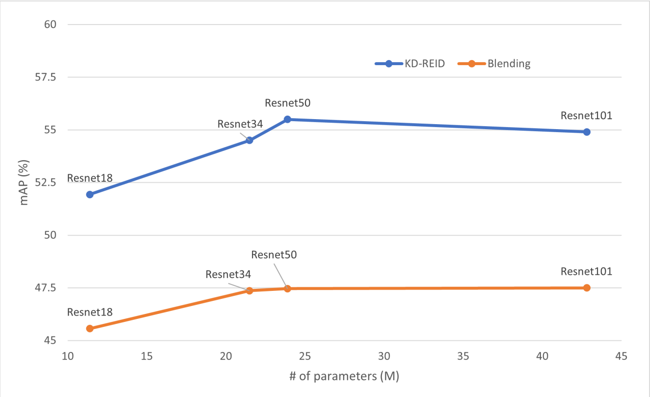

Fig. 1. Average accuracy of the common backbone model for the blending and KD-REID methods for different common backbone complexities

1.2.1. Impact of Number of Target Domains:

A good MTDA approach must scale well with an increasing number of target domains. Table 3 shows results when various targets are omitted. We see that there is a small reduction of accuracy as the number of targets increases. We expect this reduction as the problem becomes more complex, but the small loss of performance (2% in the worst case) indicates that our

Base STDA Technique	MTDA Approach	M		D		C		Average		Complexity
		mAP (%)	rank-1 (%)	mAP (%)	rank-1 (%)	mAP (%)	rank-1 (%)	mAP (%)	rank-1 (%)	# of parameters
-	Only Pre-Training on source (Lower bound)	26.0	51.2	31.4	50.5	13.0	12.7	23.5	41.7	11.4 M
D-MMD	Multiple STDA Models (Teachers)	51.4	74.9	51.4	69.3	61.8	65.9	54.9	70.0	23.9 M x T
	STDA on blended datasets	40.3	64.5	42.2	61.8	54.2	58.0	45.6	61.4	11.4 M
	KD-REID (Ours)	48.9	71.9	48.9	66.9	**58.0**	**61.7**	51.9	66.5	11.4 M
SPCL	Multiple STDA Models (Teachers)	54.2	75.3	50.0	69.6	33.4	34.8	45.9	59.9	23.9 M x T
	STDA on blended datasets	**54.8**	**77.2**	**51.6**	**70.2**	38.2	42.0	48.2	63.1	11.4 M
	KD-REID (Ours)	54.1	75.2	46.5	65.7	38.1	41.1	46.2	60.7	11.4 M
Mixed Teachers M: SPCL D: SPCL C: D-MMD	KD-REID (Ours)	55.2	76.3	50.5	68.8	53.5	57.8	**53.1**	**67.6**	11.4 M

Table 1. Performance of the different approaches on three target datasets: **Market1501 (M)**, **DukeMTMCReID (D)** and **CUHK03 (C)**. with two base STDA techniques: D-MMD and SPCL. The MSMT17 Dataset is used as source dataset, Resnet50 as the architecture of target specific backbones and Resnet18 as the architecture of common backbones. The teacher's accuracy is presented as an upper bound is not considered when determining the highest accuracy as it requires a significantly higher model complexity.

Student Model architecture (# of parameters)	M		D		C	
	mAP (%)	rank-1 (%)	mAP (%)	rank-1 (%)	mAP (%)	rank-1 (%)
Teachers (Resnet50)	51.4	74.9	51.4	69.3	61.8	65.9
Resnet101 (42.8 M)	52.8	74.6	53.2	70.3	58.7	62.4
Resnet50 (23.9 M)	52.8	75.5	53.3	71.0	60.4	64.5
Resnet34 (21.5 M)	52.3	75.5	52.1	69.7	59.1	63.9
Resnet18 (11.4 M)	48.9	71.9	48.9	66.9	58.0	61.7

Table 2. Impact of the complexity of the student architecture on performance

method is fairly robust to the addition of more target domains. The same experiment was conducted using the Market1501 dataset as source domain, the results are shown in table 4. This is a more difficult problem because MSMT17 is the most complex dataset and using it as source is very advantageous. Comparing the last line of the table with the last line of table 3, we notice that we obtain similar results on the CUHK03 and DukeMTMC-ReID datasets (<2% difference) even though we have a much weaker source training and we have added a very complex dataset to the pool of target datasets. This showcases the robustness of our method to complex targets and weaker sources. We notice in table 4 the performance for the blending approach deteriorates as we add more targets to the blended dataset. This is expected as more data from various domains makes it more difficult for the model to find a common representation.

Targets (Source:MSMT17)	M		D		C	
	mAP (%)	rank-1 (%)	mAP (%)	rank-1 (%)	mAP (%)	rank-1 (%)
Teachers	51.4	74.9	51.4	69.3	61.8	65.9
M	49.2	71.7	-	-	-	-
D	-	-	50.0	67.8	-	-
C	-	-	-	-	60.1	64.2
M + D	49.8	73.6	49.2	67.0	-	-
M + C	47.9	70.9	-	-	59.7	64.0
D + C	-	-	49.5	68.2	58.1	62.0
M+D+C	48.9	71.9	48.9	66.9	58.0	61.7

Table 3. The impact of varying the number of target dataset. These experiments are conducted using a resnet18 student and a resnet50 teacher trained with D-MMD.

Targets (Source: Market1501)	Ms		D		C	
	mAP (%)	rank-1 (%)	mAP (%)	rank-1 (%)	mAP (%)	rank-1 (%)
Teachers	15.8	34.1	51.3	69.3	68.5	72.7
Ms	12.5	28.6	-	-	-	-
D	-	-	48.3	65.3	-	-
C	-	-	-	-	61.0	65.1
Ms + D	12.1	27.0	47.3	64.4	-	-
Ms + C	11.6	24.8	-	-	60.4	64.8
D + C	-	-	41.6	60.0	58.8	63.6
Ms+D+C	11.2	25.2	47.6	65.5	59.8	64.1

Table 4. The impact of varying the number of target dataset. These experiments are conducted using a resnet18 student and a resnet50 teacher trained with D-MMD.

Targets	M		D		C	
(Source:MSMT17)	mAP (%)	rank-1 (%)	mAP (%)	rank-1 (%)	mAP (%)	rank-1 (%)
M	51.4	74.9	-	-	-	-
D	-	-	51.4	69.3	-	-
C	-	-	-	-	61.8	65.9
M + D	46.7	71.9	48.6	67.9	-	-
M + C	48.1	72.3	-	-	52.5	56.6
D + C	-	-	48.7	67.4	50.2	53.9
M + D + C	43.0	67.1	45.8	64.9	50.6	55.7

Table 5. Impact of adding datasets to the Blended Dataset. These experiments are conducted by training a resnet50 using the D-MMD technique on a single blended dataset.

A. APPENDIX

A.1. Detailed Mathematical Formulation for D-MMD Approach

To produce robust teachers adapted to various specific domains, we employ the UDA technique proposed by mekhazni2020unsupervised. This technique aims to align the pair-wise dissimilarity between domains. This approach involves computing the within-class and between-class distances in feature space. The within-class distances based on euclidean distance between each different pair of images $\mathbf{x}^v$ and $\mathbf{x}^u$ belonging to the same identity, i, are computed as follows:

$$d_i^{wc}(\Pi(\mathbf{x}_i^u), \Pi(\mathbf{x}_i^v)) = \Pi(\mathbf{x}_i^u) - \Pi(\mathbf{x}_i^v)_2, u \neq v \tag{1}$$

where $\Pi(.)$ is the model being adapted and $\mathbf{x}_i^u$ is the u-th image of class i. Note that we use tracklet information to determine the images with the same identity on the unlabeled target dataset. Similarly, the between-class distances are computed on each different pair of images $\mathbf{x}_i^u$ and $\mathbf{x}_j^z$ from different identities:

$$d_{i,j}^{bc}(\Pi(\mathbf{x}_i^u), \Pi(\mathbf{x}_j^z)) = \Pi(\mathbf{x}_i^u) - \Pi(\mathbf{x}_j^z)_2, i \neq j, u \neq v \tag{2}$$

Accordingly, $\mathbf{d^{wc}}$ and $\mathbf{d^{bc}}$ as the distributions of distance values d_i^{wc} and $d_{i,j}^{bc}$ respectively. These distributions characterize the features in the dissimilarity space. MMD gretton2012kernel, which is a metric used to evaluate the distance between two distributions, is defined as:

$$\begin{aligned} MMD(P(A), Q(B)) = & \frac{1}{I^2} \sum_{i=1}^{I} \sum_{j=1}^{J} k(a_i, a_j) \\ & + \frac{1}{J^2} \sum_{i=1}^{I} \sum_{j=1}^{J} k(b_i, b_j) \\ & - \frac{2}{IJ} \sum_{i=1}^{I} \sum_{j=1}^{J} k(a_i, b_j) \end{aligned} \tag{3}$$

where $P(A)$ is the distribution of the source domain A and $Q(B)$ is the distribution of target domain B. $k(\cdot, \cdot)$ is a kernel. a_i is the i-th sample for distribution $P(A)$ and b_i is the i-th sample from distribution $Q(B)$. I and J are the total number of samples in distributions $P(A)$ and $Q(B)$ respectively. The goal is to minimize this metric between source and target domain. Therefore, we optimize by minimizing the MMD metric in the dissimilarity space as well as the feature space. Considering the distribution distance evaluation defined in Eq. 3, the D-MMD loss is defined as follows:

$$\begin{aligned} \mathcal{L}_{D-MMD} = & MMD(\mathbf{d_s^{wc}}, \mathbf{d_t^{wc}}) \\ & + MMD(\mathbf{d_s^{bc}}, \mathbf{d_t^{bc}}) + MMD(S, T), \end{aligned} \tag{4}$$

where S and T are defined as the distribution of the source and target images represented in the feature space, respectively. The subscript s and t indicate whether the distance distributions were computed on source or target samples mekhazni2020unsupervised.

Based on theses assumptions the overall domain adaptation loss, $\mathcal{L}_{DA}$, can be expressed as:

$$\mathcal{L}_{DA}(\mathbf{x}^s, \mathbf{x}^t) = \mathcal{L}_{D-MMD}(\mathbf{x}^s, \mathbf{x}^t) + \mathcal{L}_{sup}(\mathbf{x}^s), \tag{5}$$

$$\mathcal{L}_{sup}(\mathbf{x}^s) = \mathcal{L}_{ces}(\mathbf{x}^s) + \mathcal{L}_{tri}(\mathbf{x}^s) \tag{6}$$

where $\mathcal{L}_{ces}$ and $\mathcal{L}_{tri}$ are the cross-entropy and triplet loss functions, respectively. The $\mathcal{L}_{DA}$ loss aims to align the distances of features produced by the model for the source and target domains which produces a model to perform well on the target domain as well. The details on computing the within-class and between-class distances and MMD metric is presented in Appendix A.1.

A.2. Unified Contrastive Loss

SPCL ge2020selfpaced is another effective technique to produce strong teacher models is which is based on clustering. Since it is assumed that the data samples from the target domains are not annotated, the initial step is to generate pseudo-labels for them. Clustering could be performed using a standard clustering algorithm such as DBSCAN ester1996density.

The method relies on the criteria of compactness and independence to determine which clusters are reliable. Samples in clusters deemed reliable are assigned a pseudo-label matching their cluster and samples that are not in a reliable cluster are considered as un-labeled samples. The feature vectors for every sample are then kept in a hybrid memory module under three categories: *labeled* source samples, *pseudo-labeled* target samples and *unlabeled* target samples.

For each label in the source samples category, a class centroid corresponding to the mean feature vector of the label is generated. Similarly, class centroids are generated using the pseudo-labels of the clustered target samples. Once all the data is properly categorized and class centroids are determined, a unified contrastive loss function is used to adapt the teacher models to corresponding target datasets. The domain adaptation loss, $\mathcal{L}_{DA}$ defined as:

$$\mathcal{L}_{DA} = -log\frac{exp(\langle \mathbf{f}, (z^{+})\rangle/\tau)}{\sum_{k=1}^{n^s} exp(\langle \mathbf{f}, (w_k)/\tau) + \sum_{k=1}^{n_c^t} exp(\langle \mathbf{f}, (c_k)/\tau) + \sum_{k=1}^{n_o^t} exp(\langle \mathbf{f}, (v_k)/\tau)} \tag{7}$$

Where $\mathbf{f}$ is the feature vector being used to compute the loss, τ is a temperature hyperparameter, $\mathbf{w_k}$ is the class centroid for the source label k, $\mathbf{c_k}$ is the class centroid for target pseudo-label k and $\mathbf{v_k}$ is the feature vector of un-labeled target sample k. $\mathbf{z}^{+}$ corresponds to the positive category for sample $\mathbf{f}$. n^s, n_c^t and n_o^t are the number of labeled source samples, number of pseudo-labeled target samples and number of un-labeled samples respectively within the mini-batch. In essence, this loss brings every sample closer to it's positive centroid while pushing it away from negative centroids. The clusters are re-evaluated every epoch and centroids are updated accordingly each epoch until convergence.Roughly speaking, SPCL loss brings every sample closer to it's positive centroid while pushing it away from negative centroids. The clusters are re-evaluated every epoch and centroids are updated accordingly each epoch until convergence.

A.3. Pre-training the Models on the Source Domain Dataset:

The pre-training process is performed using two supervised loss functions: Softmax cross-entropy ($\mathcal{L}_{ces}$) and triplet loss ($\mathcal{L}_{tri}$). $\mathcal{L}_{ces}$ [1] is defined as:

$$\mathcal{L}_{ces} = (1-\epsilon)\cdot\mathcal{L}_{ce} + \epsilon/C \tag{8}$$

where C is the number of classes and $\epsilon \in [0, 1]$ is a hyper-parameter. $\mathcal{L}_{ce}$ is the standard cross-entropy loss. We also use a hard samples mining triplet loss, $\mathcal{L}_{tri}$, as proposed by hermans2017defense by randomly sampling P classes (person identities) and then randomly sampling K images of each class (person). For each sample α in the batch the triplet loss is defined as:

$$\begin{aligned}\mathcal{L}_{tri} = \sum_{i=1}^{P}\sum_{\alpha=1}^{K}[m + \max_{p=1\ldots K} d(\Pi(\mathbf{x}_i^{\alpha}, \mathbf{x}_i^{p})) \\ - \min_{\substack{j=1\ldots P\\ n=1\ldots K\\ j\neq i}} d(\Pi(\mathbf{x}_i^{\alpha}, \mathbf{x}_j^{n}))]_{+}\end{aligned} \tag{9}$$

where, $\Pi(.)$ to be the features extracted right before the classifier layer of model Π, m is a hyper-parameter margin and $d(\cdot,\cdot)$ is the euclidean distance. α is the anchor sample while p and n indicate a positive, same-identity, sample or a negative, different-identity, sample, respectively. The overall loss for pre-training on source is expressed as:

$$\mathcal{L}_{pre-train}(\mathbf{x}^s) = \mathcal{L}_{ces}(\mathbf{x}^s) + \mathcal{L}_{tri}(\mathbf{x}^s) \tag{10}$$

The set of teacher models and the student model are trained by back-propagation algorithm using mentioned losses.

B. REFERENCES

[1] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *ICCV 2016*.