

# Attention Network Robustification for Person ReID

Hussam Lawen\*      Avi Ben-Cohen\*

Matan Protter      Itamar Friedman      Lihi Zelnik-Manor  
Machine Intelligence Technology, DAMO Academy, Alibaba Group

{hussam.lawen, avi.bencohen, matan.protter, itamar.friedman, lihi.zelnik }  
@alibaba-inc.com

## Abstract

The task of person re-identification (ReID) has attracted growing attention in recent years with improving performance but lack of focus on real-world applications. Most state of the art methods use large pre-trained models, e.g., ResNet50 (~25M parameters), as their backbone, which makes it tedious to explore different architecture modifications. In this study, we focus on small-sized randomly initialized models which enable us to easily introduce network and training modifications suitable for person ReID public datasets and real-world setups. We show the robustness of our network and training improvements by outperforming state of the art results in terms of rank-1 accuracy and mAP on Market1501 (96.2, 89.7) and DukeMTMC (89.8, 80.3) with only 6.4M parameters and without using re-ranking. Finally, we show the applicability of the proposed ReID network for multi-object tracking.

## 1. Introduction

The objective in person re-identification (ReID) is to assign a stable ID to a person in multiple camera views. In this study we are interested in the development of small sized models for ReID with high accuracy for two main reasons. First, it is beneficial for practical deployment and productization of ReID solutions. Second, the research for models that provide high accuracy requires exploration of many architecture variations and training schemes. When the backbone is heavy, re-training consumes both a lot of time and computing resources, hence, we wish to avoid this. Our approach differs from many SotA methods, that rely on large pre-trained backbone models, such as ResNet50, e.g. [31, 26, 28, 13].

We argue that a cost-effective ReID model should be computationally efficient, capable of running on low-res video input, and robust to multiple camera setting. Hence,

\*Equal contribution

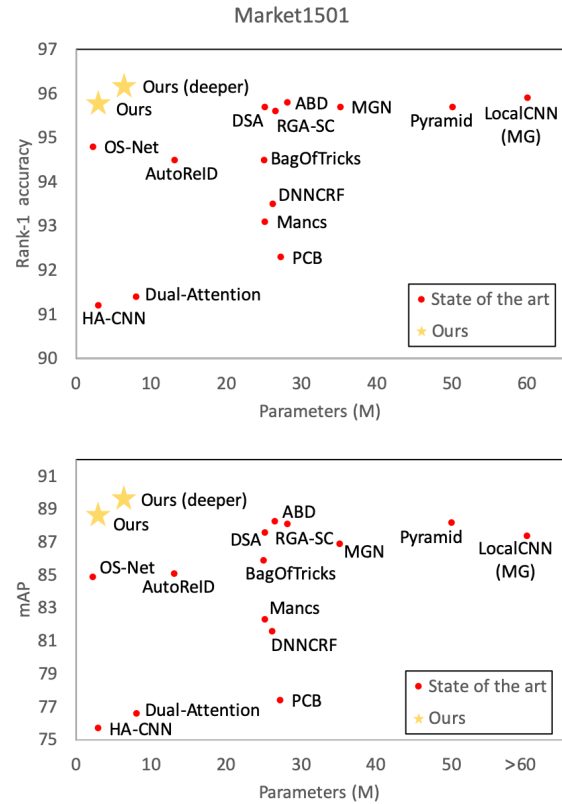


Figure 1. Performance comparison of our approach and SotA ReID methods on Market1501 dataset. Top: rank-1 accuracy vs. number of parameters. Bottom: mAP vs. number of parameters.

we propose an efficient ReID model and training schemes that demonstrate state of the art performance under these requirements. To reduce the computational burden, we aim to decrease the number of parameters and use a relatively small ReID model. Figure 1 shows the current state of the art results [1, 11, 13, 19, 24, 26, 28, 32, 36, 34, 35, 29, 2, 42] and the number of parameters compared to our proposed method on the popular Market1501 [37] dataset in terms of

rank-1 accuracy and mAP. For some methods, the number of parameters was not known so we used an estimated lower bound. Using our proposed training framework we achieve state of the art results with an order of magnitude smaller model compared to the best existing ReID CNN.

The importance of training “tricks” for deep person ReID was been discussed before in [13]. In this paper, we suggest training techniques and architecture modifications that rubstify an harmonious attention network [11] to achieve similar or better results than much larger and complicated models. The contribution of this paper is thus three-fold:

- We propose a robust deep person ReID model. Our model achieves state of the art results on two popular person ReID datasets (Market1501 and DukeMTMC ReID [21]) despite having a small number of parameters, number of FLOPS, and low resolution input image in comparison to current leading methods.
- We explore a variety of training schemes and network choices that we studied. While we have not explored their affect for other models we believe they could be of interest to others to examine.
- We demonstrate the applicability of the proposed person ReID model by improving multi-target multi-camera tracking.

In the following section we describe the baseline ReID network we started with. The training techniques and architecture modifications that were explored in this study are presented in section 3. Next, the experimental results including an ablation study, additional analysis, and comparison to state of the art are presented (section 4). Finally, multi camera multi target tracking results are presented in section 5.

## 2. Baseline ReID network - HA-CNN

We wanted to obtain a robust model with a small number of parameters and capability to deal with low-res input images to reduce computational complexity.

We chose the Harmonious Attention CNN [11] as our primary baseline due to it being a light-weight yet deep model, that can be trained from scratch thus obviating the need to pre-train on additional data, while providing nice results taking into consideration the small number of parameters (2.7M). In addition, the input image size for this network is relatively small compared to other person ReID networks.

The HA-CNN is an attention network with several attention modules including a soft spatial and channel-wise attention and a hard attention to extract local regions. The network architecture holds two branches: a global one and

a local one that uses the regions extracted based on the hard attention. Finally, the output vectors of both branches are concatenated for the final person image descriptor. Holding two branches and multiple attention modules improves the network perception and despite these features the HA-CNN keeps a small number of parameters making it accurate and efficient. However, parts of the architecture can still be optimized as well as the training scheme. Optimizing it can further improve the HA-CNN and obtain a more accurate and robust model.

## 3. Methods

Deep learning model performance is highly dependent on the training schemes being used. Recent works have shown that adding different training procedure refinements can improve the model results significantly [13, 6]. Architecture modifications and refinements can also have an important impact on the final result. In this section we elaborate on the training schemes and architecture modifications we used in this study starting with HA-CNN as our baseline. The training techniques will be presented in section 3.1 and the architecture modifications will be presented in section 3.2. In section 3.3 we mention several modifications that didn’t improve the model performance.

### 3.1. Training techniques

The following training techniques were used in this study:

**Random erasing augmentation (REA)** [40] - randomly erasing a rectangle in an image has shown to improve the model generalization ability. We used REA with the following parameters: probability for random erasing an image of 0.5, area ratio of erasing a region in the range of  $0.02 < S_e < 0.4$ , and with aspect ratio in the range of  $0.3 < r < 3.3$ .

**Warmup** [4] - used to bootstrap the network for better performance. Starting with a smaller learning rate has shown to improve the training process stability, especially when using a randomly initialized model. Using warmup we start the training with a small learning rate and then gradually increase it. We used the following learning rate scheme:

$$lr(t) = \begin{cases} 3 \times 10^{-2} \times \frac{t}{10} & \text{if } t \leq 10 \\ 3 \times 10^{-2} & \text{if } 10 < t \leq 150 \\ 3 \times 10^{-3} & \text{if } 150 < t \leq 225 \\ 3 \times 10^{-4} & \text{if } 225 < t \leq 350 \end{cases} \quad (1)$$

**Label smoothing** [27] - widely used for classification problems by encouraging the model to be less confident during training and prevent over-fitting. We used label smoothing in a similar way as proposed in [13].

**Weighted triplet loss** - Triplet loss is widely used in Person Re-Identification methods and other computer vi-

sion tasks like Face Recognition and Few-Shot Learning. The original triplet loss was proposed by Schroff *et al.* [23]. We denote an anchor sample by  $x_a$ , positive samples as  $x_p \in P(a)$  and negative samples as  $x_n \in N(a)$ , then the triplet loss can be written as:

$$L_1 = [m + d(x_a, x_p) - d(x_a, x_n)]_+ \quad (2)$$

where  $m$  is the given inter-class separation margin,  $d$  denotes distance of appearance, and  $[\cdot]_+ = \max(0, \cdot)$ .

Hermans *et al.* [7] and Mischuk *et al.* [54] have proposed the batch-hard triplet loss that selects only the most difficult positive and negative samples:

$$L_2 = \left[ m + \max_{x_p \in P(a)} d(x_a, x_p) - \min_{x_n \in N(a)} d(x_a, x_n) \right]_+ \quad (3)$$

In contrast to the original triplet loss, the batch-hard triplet loss emphasizes hard examples. However, it is sensitive to outlier samples and may discard useful information due to its hard selective approach. To deal with these problems, Ristani *et al.* proposed the batch-soft triplet loss:

$$L_3 = \left[ m + \sum_{x_p \in P(a)} w_p d(x_a, x_p) - \sum_{x_n \in N(a)} w_n d(x_a, x_n) \right]_+ \quad (4)$$

$$w_p = \frac{e^{d(x_a, x_p)}}{\sum_{x \in P(a)} e^{d(x_a, x)}} , \quad w_n = \frac{e^{-d(x_a, x_n)}}{\sum_{x \in N(a)} e^{-d(x_a, x)}}$$

One hyper-parameter that exists in all of the triplet loss variations shown above is the margin denoted as  $m$ . In this paper we used a modified version of the batch-soft triplet loss presented in Equation (4) that eliminates the need to manually determine the margin value as will be explained next.

**Soft margin** [7] - We used the softplus function  $\ln(1 + \exp(\cdot))$  instead of  $[\cdot]_+ = \max(0, \cdot)$  in the triplet loss function:

$$L_4 = \text{Softplus} \left( \sum_{x_p} w_p d(x_a, x_p) - \sum_{x_n} w_n d(x_a, x_n) \right) \quad (5)$$

This modification is called soft margin since it replaces the hard cut off of the max function (as in the *ReLU* activation function) with a soft exponential decay. By using the soft margin there's no need to choose a margin parameter. Using a hard margin value, once the negative samples' distance is larger than the positive samples' distance by more than the hard margin value there is no incentive to push the positive samples closer or push the negative samples further away. The soft margin encourages a continuous reduction of the positive distance to the anchor while increasing the negative distance. Figure 3 illustrates the difference between

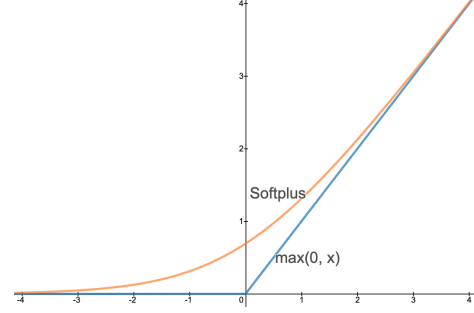


Figure 2. The *Softplus* function ( $\ln(1 + \exp(\cdot))$ ) compared to  $\max(0, \cdot)$ .

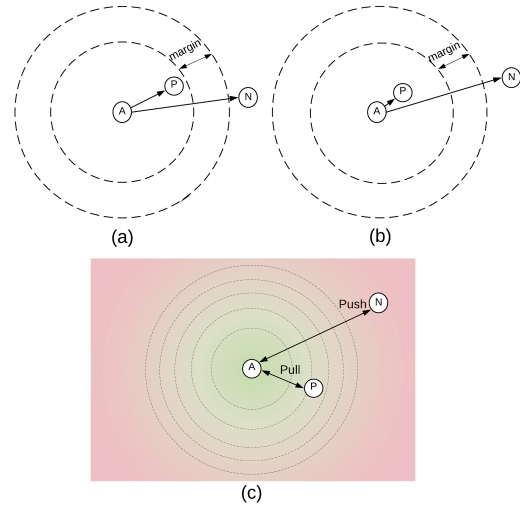


Figure 3. Example of hard margin and soft margin. Although (b) is more desirable than (a), using hard margin in the triplet loss we will get a loss value of zero for both while in (c), using the soft margin the loss will continue to pull the positive sample closer to the anchor while pushing the negative sample away and will encourage going from (a) to (b).

the soft and the hard margin. The examples in (a) and (b) will obtain a similar loss value of zero since it answers the requirement based on the hard margin. While in (c), using the soft margin, the computed loss will continue to push the positive sample closer to the anchor while pushing the negative sample away.

**L2 normalization** - The normalization of the feature vectors can be important when using two different loss functions such as cross-entropy and triplet loss which are optimized using different distance measures. Lou *et al.* [13] have tackled the normalization problem by adding a batch normalization layer after the feature vectors (right before the fully connected layer). In our empirical studies we found that simply using  $L_2$  normalization for each feature vector (global and local) during training achieved a better performance. Figure 4 shows the additional  $L_2$  normalization used during training and inference.

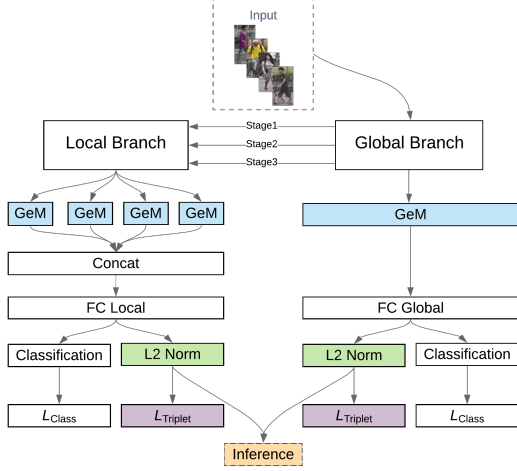


Figure 4. Robust-ReID architecture diagram showing the proposed modifications over the original HA-CNN:  $L_2$  normalization during training, GeM instead of average pooling, and soft triplet loss.

**SWAG [15]**- A common technique to further boost the performance of a model is using ensemble approaches. The standard ensemble method uses several models in test time for the final prediction utilizing much more computing resources. Stochastic weight averaging (SWA) [9] suggests to form an ensemble during training and output a single model for inference. SWA essentially conducts a uniform average over several model weights traversed by SGD during training to achieve a wider region of the loss minima. In order to use SWA a modified learning rate scheduler is required. In this study we used cosine annealing learning rate scheduler with  $cycles = 15$  of 35 epochs and cycle decay factor of 0.7 after each cycle. At the end of each cycle we average the weights of the current model with the previous models taken from the end of each cycle. This is different than the original learning rate scheduler used in [9] as it was shown empirically to be better in our experiments. Another variation of the SWA is the SWA-Gaussian (SWAG) [15]. SWAG fits a Gaussian distribution using the SWA solution and diagonal covariance forming an approximate posterior distribution over neural network weights. Next, SWAG performs a Bayesian model averaging based on the Gaussian distribution. We used SWAG with the modified learning rate scheduler in this study.

### 3.2. Architecture modifications

In addition to the training techniques listed above, we further explored architecture modifications:

**Shuffle blocks [14]**- Our goal was to improve our network accuracy while maintaining a small number of parameters. To do this, we examined replacing the inception blocks with the shuffle blocks presented in Figure 5.

Shuffle-A is more efficient than the original inception

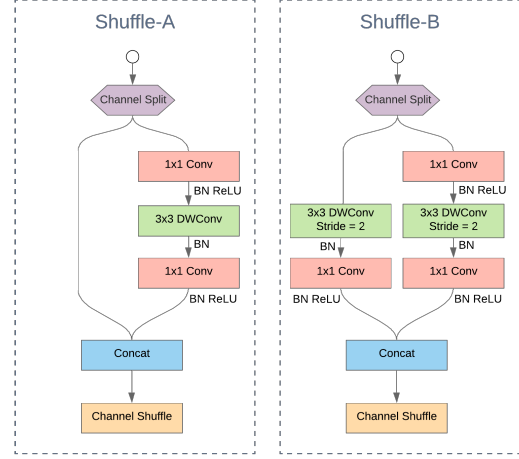


Figure 5. The shuffle blocks used in this study to replace the original HA-CNN inception blocks.

block since it splits the input features into two equal branches, the first branch remains as is while three convolution operators are applied to the second branch. In addition, one of the convolution operators is depth-wise convolution. The Shuffle-A blocks can be used in a repeated sequence and still maintain the same number of parameters as the original inception block. Hence, we were able to build a deeper network with similar number of parameters. The Shuffle-B block is similar to Shuffle-A but it can be used for spatial down-sampling or channel expansion. These characteristics require convolution operators to be applied to the first branch as well. Table 1 summarizes the repeated sequences of Shuffle blocks used in our proposed architecture.

**Generalized Mean (GeM) [20]** - In the original HA-CNN version the pooling method being used (just before the fully connected layer) is global average pooling. However, we found that using global Max-pool instead can achieve different results and it wasn't clear which one performs better and why. Hence, we used the trainable GeM pooling which generalizes both max and average pooling methods. The GeM operator for a single feature map  $f_k$  can be described as:

$$GeM(f_k = [x_0, x_1, \dots, x_n]) = \left[ \frac{1}{n} \sum_{i=1}^n x_i^{p_k} \right]^{\frac{1}{p_k}} \quad (6)$$

We initialized the GeM parameter with  $p_k = 3$  in our experiments. Figure 4 shows where it is used during training and inference.

**Deeper and wider** - We further propose a deeper and wider version of our architecture by modifying the number of shuffle blocks as well as the number of output channels in each stage. Table 1 presents these modifications in bold.

Local Branch	Global Branch	Layer	Input	Stride	1×		2×		
					Repeat	Output Ch.	Repeat	Output Ch.	
	Conv1	Conv 3x3	160×64	2	1	32	1	<b>36</b>	
	Stage1	Shuffle-B	80×32	1	1	128	1	<b>240</b>	
		Shuffle-A		1	7		<b>8</b>		
		Shuffle-B		2	1		1		
	Soft-Attn1	HA-Block	40×16	1	1		1		
Hard-Attn1		Shuffle-B	4×(24×28)	1	1		1		
	Stage2	Shuffle-B	40×16	1	1	256	1	<b>320</b>	
		Shuffle-A		1	10		<b>11</b>		
		Shuffle-B		2	1		1		
	Soft-Attn2	HA-Block	20×8	1	1		1		
Hard-Attn2		Shuffle-B	4×(12×14)	1	1		1		
	Stage3	Shuffle-B	20×8	1	1	384	1	<b>480</b>	
		Shuffle-A		1	7		<b>8</b>		
		Shuffle-B		2	1		1		
	Soft-Attn3	HA-Block	10×4	1	1		1		
Hard-Attn3		Shuffle-B	4×(6×7)	1	1		1		
	Pooling	GeM	10×4	1	1		1		
Pooling		GeM	4×(3×4)	1	1		1		
	FC Global	Linear	1×1	1	1	512	1	<b>960</b>	
FC Local		Linear	1×1	1	1		1		
FLOPs						0.72B		<b>1.68B</b>	
# of Params.						2.9M		<b>6.4M</b>	

Table 1. Overall architecture of Robust-ReID, for 2 different levels of complexities. Since our architecture uses a low-resolution input of 160x64, we down-scale the feature maps by applying strided convolution only in the last layer of each stage and not in the beginning. This way the network can leverage a higher spatial resolution in most of the network.

	HA-CNN [11]	a	b	c	d	e	f	g	h	i	Robust-ReID
BagOfTricks [13]		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Soft triplet				✓	✓	✓	✓	✓	✓	✓	✓
L2 normalization			✓		✓	✓	✓	✓	✓	✓	✓
Shuffle blocks			✓	✓		✓	✓	✓	✓	✓	✓
Soft margin				✓	✓		✓	✓	✓	✓	✓
GeM			✓	✓	✓	✓		✓	✓	✓	✓
Deeper & wider			✓	✓	✓	✓	✓			✓	✓
SWAG									✓		✓
Market1501 Rank1	91.2	93.2	94.1	94.5	94.9	95.1	95.3	95.4	95.8	95.7	96.2
mAP	75.7	82.0	85.3	85.7	86.6	87.1	87.9	87.3	88.7	88.1	89.7

Table 2. Ablation study on Market1501. The first column indicates the different training techniques and architecture modifications we tried including some of the tricks mentioned in BagOfTricks [13]: warmup, random erase, label smoothing, and no bias in the classification layers. The baseline we started with, i.e. the original HA-CNN implementation, is presented in the second column for comparison. The last column shows the results of our proposed Robust-ReID network including all of the training techniques and architecture modifications proposed in this study. Columns a-i demonstrates the impact of each modification by turning it off.

### 3.3. Additional tricks we tried

Some tricks that were introduced in prior works failed to improve the performance when used with our baseline. The following tricks didn’t work:

- As mentioned, max and average pooling provide different results so one way to benefit from both pooling

methods is by concatenation of their output. Basically we tried to replace the global average pooling used in the original HA-CNN architecture with these two pooling methods and concatenations. It resulted in a similar accuracy with more parameters in the final model.

- The batch norm neck used in [13] provided inferior

results when compared to the simple  $L_2$  normalization.

- Hard triplet loss instead of the soft version was too sensitive to outliers.
- Using Shuffle blocks without  $L_2$  normalization or soft margin in the triplet loss didn't improve the performance.
- Training for more epochs didn't improve the performance. The only way it did lead to an improvement was using the Cyclic LR scheme.
- Cyclic LR scheme didn't improve the results when used from scratch from the beginning of the training. It only worked when used in additional training epochs after the the model converged.

## 4. Experimental results

In the following we evaluate the performance boost using each one of the methods discussed in section 3. Our models are evaluated on Market1501 and DukeMTMC ReID datasets based on rank-1 accuracy and mAP.

**Implementation details** - All person images are resized to  $160 \times 64$ . We used SGD for optimization with a learning rate schedule as in Equation (1) for 350 epochs. When using SWAG we train for 15 cycles of 35 epochs which sums up to 525 additional epochs. We randomly sample 8 identities and 4 images per person in each training batch.

### 4.1. Ablation study

To evaluate the different training techniques explored in this study we set several experiments in an ablation study. Table 2 shows the different modifications starting from the original HA-CNN architecture. The first row indicates using some of the tricks from [13] that showed an improvement when tested on Market1501 using the HA-CNN architecture. These include warm-up, random erasing, and no-bias in the fully connected layers. These tricks alone (experiment a) provided an improvement of 2% in rank-1 accuracy and 6.3% in mean average precision compared to the original HA-CNN paper result (i.e. our baseline).

Next, to test the influence of some of our modifications we report the performance after disabling them. The most significant decrease in results compared to column i was caused by disabling the weighted triplet loss and soft margin (using the original triplet loss as in equation (2) instead) with a drop of 1.6% in rank-1 accuracy and 2.8% in mAP (column b). Cancelling the  $L_2$  normalization caused a decrease of 1.2% in rank-1 accuracy and 2.4% in mAP (column c). Reduction of other modifications such as shuffle blocks, soft margin, GeM, and deeper and wider network caused a decrease in the performance as well indicating the benefit of using it.

Setup	Update	Market1501	
		r = 1	mAP
1	-	93.8	83.6
	+LR Scheme	94.3	84.8
	+SWAG	<b>94.5</b>	<b>85.3</b>
2	-	95.4	87.3
	+LR Scheme	<b>95.8</b>	88.2
	+SWAG	<b>95.8</b>	<b>88.7</b>
3	-	95.7	88.1
	+LR Scheme	95.7	88.9
	+SWAG	<b>96.2</b>	<b>89.7</b>

Table 3. Performance evaluation on Market1501 for SWAG with cosine annealing with decay factor learning scheme.



Figure 6. Performance evaluation of SWAG compared to SGD using the cosine annealing learning scheme on Market1501 dataset showing the average of 5 runs. Top: rank-1 accuracy vs. epoch. Bottom: mAP vs. epoch.

Finally, we used the SWAG in two experiments: experiment h and the final Robust-ReID. Continuing the training with SWAG provided an improvement in both rank-1 and mAP in both experiments. The SWAG is used in this study as a post process for models that already achieve high accuracy to show its contribution on top of that.

### 4.2. Exploring SWAG

Our empirical experiments showed that the SWAG method consistently improved our model performance. However, it requires additional training time and uses a custom made cosine annealing learning scheme with a decay factor. Therefore, we wanted to further explore the SWAG contribution by analyzing some of our experimental

results. Table 3 shows the results when testing the learning rate scheme with and without SWAG for three different setups. In the first setup we used our proposed architecture minus three main modifications: GeM, Shuffle blocks, and deeper and wider. The second and third setups are experiments g and i in Table 2 respectively. Evidently, adding the LR scheme provided a nice improvement, and adding the SWAG performed even better. The most significant improvements were in terms of mAP.

Figure 6 presents the average over five experiments comparing SWAG and standard SGD in terms of Rank1 accuracy and mAP on Market1501 dataset. Using SWAG the accuracy trend seems more consistent and robust compared to standard SGD. In addition, it is consistently and significantly better in terms of mAP.

### 4.3. Comparison to state of the arts

We compare our models performance to different state of the art methods (Table 4). Our best model achieves state of the art results in terms of rank-1 accuracy and mAP on Market1501 (96.2, 89.7) and DukeMTMC (89.8, 80.3) with only 6.4M parameters. To our best knowledge, our model achieves the best performance on these public datasets. It should be noted that the smaller version of our model (2.9M parameters) also achieves state of the art results on both datasets. In terms of FLOPS our final network has 1.7B FLOPS while the ResNet 50 used in Luo *et al.* [13] implementation has 4.1B FLOPS. We did not apply re-ranking for clear comparison and since it is currently not relevant for real world practice.

## 5. Application to multi object tracking

Although the public datasets used in this study for person ReID are valuable for comparison between different architectures and models, we wanted to evaluate the model’s applicability by using it to improve multi target multi camera tracking in two different scenes.

### 5.1. Indoor multi target multi camera tracking

We first explore whether the proposed model can be used for tracking purposes in a room with people coming in and out. Testing the model in a real world setting such as tracking is much more challenging. A wrong ReID assignment can affect the assignment of other persons since we only compare each query image to tracks that are not active (not present in the room at the time of the query). In addition, for each query we need to decide if we open a new track or assign it to an existing track (ReID), meaning that in some cases the gallery does not include images of the person found in the query.

We used the LAB sequence which is a part of the Task-Decomposition database [8] of multi-view sequences for people tracking. The LAB sequence is about 12.5 minutes

Type	Method	Market1501		DukeMTMC	
		r = 1	mAP	r = 1	mAP
Mask-guided	SPReID [10]	92.5	81.3	84.4	71.0
	MaskReID [17]	90.0	75.3	78.8	61.9
Stripe-based	AlignedReID [33]	90.6	77.7	81.2	67.4
	SCPNet [5]	91.2	75.2	80.3	62.6
	LocalCNN [32]	91.5	77.7	82.2	66.0
	Pyramid[36]	92.8	82.1	-	-
	PCB [26]	93.8	81.6	83.3	69.2
	BFE[3]	94.5	85.0	88.7	75.8
	MGN [29]	95.7	86.9	88.7	78.4
	Pyramid[36]	95.7	88.2	89.0	79.0
	LocalCNN (MG) [32]	95.9	87.4	-	-
Dense-semantics	DSA [34]	95.7	87.6	86.2	74.3
GAN-based	Camstyle [41]	88.1	68.7	75.3	53.5
	PN-GAN [18]	89.4	72.6	73.6	53.2
	DG-Net [38]	94.8	86.0	86.6	74.8
Global feature	IDE [39]	79.5	59.9	-	-
	SVDNet [25]	82.3	62.1	76.7	56.8
	TriNet[7]	84.9	69.1	-	-
	AWTL[22]	89.5	75.7	79.8	63.4
	OS-Net [42]	94.8	84.9	88.6	73.5
	BagOfTricks [13]	94.5	85.9	86.4	76.4
NAS	Auto-ReID [19]	94.5	85.1	88.5	75.1
Attention-based	HA-CNN [11]	91.2	75.7	80.5	63.8
	DuATM [24]	91.4	76.6	81.2	62.3
	Manes [28]	93.1	82.3	84.9	71.8
	ABD [2]	95.6	88.3	89.0	78.6
	RGA-SC [35]	95.8	88.1	86.1	74.9
	<b>Ours (2.9M)</b>	95.8	88.7	88.8	78.9
	<b>Ours (6.4M)</b>	<b>96.2</b>	<b>89.7</b>	<b>89.8</b>	<b>80.3</b>

Table 4. Comparison of state-of-the-arts methods.

long<sup>1</sup>, the tracking domain is about 5\*6 meters in dimension, and the images were captured at 15 Hz with a resolution of 640\*480 pixels, where four cameras are installed at the corners of the room. Through the sequence, people enter, walk around, sit down and exit the room randomly, causing frequent occlusions. The maximum number of people in the scene at the same time is 7. We first used an internal software for global people tracking which uses the calibration provided for each camera and report the results we got with and without using the model for ReID in terms of MOTA and IDF1. We used ReID each time a person enters the room by comparing it to several images per person that is currently not tracked inside the room.

Table 5 shows the results obtained using different models including: the original HA-CNN and our proposed Robust-ReID model. The Robust-ReID performed better than the original HA-CNN in terms of IDF1 using Market1501 or DukeMTMC for training. Due to the original resolution of the videos, the size of the bounding box of each query and gallery image can get very small in size. Our model showed robustness to the low-res images since it was trained on small sized input.

<sup>1</sup>Information in the database website mentions 3.5 minutes but the downloaded videos are actually 12.5 minutes long.



Model	Trained Dataset	MOTA	IDF1
Robust-ReID	DukeMTMC	96.1	<b>89.1</b>
Robust-ReID	Market1501	96.1	79.6
HA-CNN	DukeMTMC	96.1	78.9
HA-CNN	Market1501	96.1	65.7
No ReID	-	96.1	57.1

Table 5. Multi camera multi target tracking results on LAB dataset using our proposed Robust-ReID model compared to the original HA-CNN.

Model	MOTA	IDF1
Robust-ReID	<b>36.7</b>	<b>46.2</b>
Long <i>et al.</i> [12] (code)	36.5	46.1
Long <i>et al.</i> [12] (paper)	35.7	45.3

Table 6. Multi target tracking results on MOT16 validation set using our proposed Robust-ReID model compared to Long *et al.* [12].

## 5.2. Outdoor multi target tracking

To evaluate the performance of the proposed ReID model for outdoor object tracking we conducted an additional experiment on the MOT16 dataset [16]. We followed Long *et al.* [12], using the same exact tracker. We only replaced the GoogLeNet based ReID model with our ReID model. As in [12] we did not train on MOT16 and we used the same validation set for comparison including 5 video sequences. Each sequence varies in the camera angle and the distance from the subjects. One sequence can have large person bounding boxes captured from front view angle and other can have small bounding boxes with a much different camera angle. Therefore, having a single ReID model that can excel in all of these domains is challenging. For the purpose of multi target training, our model was trained on multiple ReID datasets (DukeMTMC, Market1501 and MSMT17 [30]) in order to have a more generic representation that can cope with the variability that exists in the MOT16 sequences. Note that ReID domain adaptation is still an active area of research and its out of the scope of this work. Table 6 shows the results in terms of IDF1 and MOTA. The proposed ReID model showed robustness to these difficulties with a slight improvement in both metrics.

## 6. Conclusions

This paper explores several training techniques and architecture modifications and suggests how to integrate them into a harmonious attention based network for person ReID. Each training technique is tested as well as some of the tricks presented in other prior works. Using the proposed training scheme and network modifications we were able to outperform SotA works achieving 96.2% rank1 accuracy and 89.7% mAP on Market1501 and 89.8% rank1 accuracy

and 80.3% mAP on DukeMTMC with only 6.4M parameters. In addition, we show that even for a smaller version (2.9M parameters) we achieve state of the art results. Finally, we show the applicability of our proposed model by utilizing it to improve existing methods for multi object tracking on a public dataset. Future work entails more experiments using other deep ReID networks as our baseline, as well as tackling the cross-domain challenges in person ReID.

## Acknowledgments

We would like to thank Sagi Rorlich and Genadiy Vasserman for their help in some of the experiments.

## References

- [1] D. Chen, D. Xu, H. Li, N. Sebe, and X. Wang. Group consistent similarity learning via deep crf for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8649–8658, 2018.
- [2] T. Chen, S. Ding, J. Xie, Y. Yuan, W. Chen, Y. Yang, Z. Ren, and Z. Wang. Abd-net: Attentive but diverse person re-identification. *arXiv preprint arXiv:1908.01114*, 2019.
- [3] Z. Dai, M. Chen, S. Zhu, and P. Tan. Batch feature erasing for person re-identification and beyond. *arXiv preprint arXiv:1811.07130*, 2018.
- [4] X. Fan, W. Jiang, H. Luo, and M. Fei. Spheredrid: Deep hypersphere manifold embedding for person re-identification. *Journal of Visual Communication and Image Representation*, 60:51–58, 2019.
- [5] X. Fan, H. Luo, X. Zhang, L. He, C. Zhang, and W. Jiang. Scpnet: Spatial-channel parallelism network for joint holistic and partial person re-identification. In *Asian Conference on Computer Vision*, pages 19–34. Springer, 2018.
- [6] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li. Bag of tricks for image classification with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 558–567, 2019.
- [7] A. Hermans, L. Beyer, and B. Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017.
- [8] T. Hu, S. Messelodi, and O. Lanz. Dynamic task decomposition for probabilistic tracking in complex scenes. In *2014 22nd International Conference on Pattern Recognition*, pages 4134–4139. IEEE, 2014.
- [9] P. Izmailov, D. Podoprikin, T. Garipov, D. Vetrov, and A. G. Wilson. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018.
- [10] M. M. Kalayeh, E. Basaran, M. Gökmen, M. E. Kamasak, and M. Shah. Human semantic parsing for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1062–1071, 2018.
- [11] W. Li, X. Zhu, and S. Gong. Harmonious attention network for person re-identification. In *Proceedings of the IEEE Con-*



- ference on Computer Vision and Pattern Recognition, pages 2285–2294, 2018.
- [12] C. Long, A. Haizhou, Z. Zijie, and S. Chong. Real-time multiple people tracking with deeply learned candidate selection and person re-identification. In *ICME*, volume 5, page 8, 2018.
  - [13] H. Luo, Y. Gu, X. Liao, S. Lai, and W. Jiang. Bag of tricks and a strong baseline for deep person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.
  - [14] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 116–131, 2018.
  - [15] W. Maddox, T. Garipov, P. Izmailov, D. Vetrov, and A. G. Wilson. A simple baseline for bayesian uncertainty in deep learning. *arXiv preprint arXiv:1902.02476*, 2019.
  - [16] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler. Mot16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*, 2016.
  - [17] L. Qi, J. Huo, L. Wang, Y. Shi, and Y. Gao. Maskreid: A mask based deep ranking neural network for person re-identification. *arXiv preprint arXiv:1804.03864*, 2018.
  - [18] X. Qian, Y. Fu, T. Xiang, W. Wang, J. Qiu, Y. Wu, Y.-G. Jiang, and X. Xue. Pose-normalized image generation for person re-identification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 650–667, 2018.
  - [19] R. Quan, X. Dong, Y. Wu, L. Zhu, and Y. Yang. Auto-reid: Searching for a part-aware convnet for person re-identification. *arXiv preprint arXiv:1903.09776*, 2019.
  - [20] F. Radenović, G. Toliás, and O. Chum. Fine-tuning cnn image retrieval with no human annotation. *IEEE transactions on pattern analysis and machine intelligence*, 41(7):1655–1668, 2018.
  - [21] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *European Conference on Computer Vision*, pages 17–35. Springer, 2016.
  - [22] E. Ristani and C. Tomasi. Features for multi-target multi-camera tracking and re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6036–6046, 2018.
  - [23] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.
  - [24] J. Si, H. Zhang, C.-G. Li, J. Kuen, X. Kong, A. C. Kot, and G. Wang. Dual attention matching network for context-aware feature sequence based person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5363–5372, 2018.
  - [25] Y. Sun, L. Zheng, W. Deng, and S. Wang. Svdnet for pedestrian retrieval. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3800–3808, 2017.
  - [26] Y. Sun, L. Zheng, Y. Yang, Q. Tian, and S. Wang. Beyond part models: Person retrieval with refined part pooling (and a strong convolutional baseline). In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 480–496, 2018.
  - [27] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.
  - [28] C. Wang, Q. Zhang, C. Huang, W. Liu, and X. Wang. Mancs: A multi-task attentional network with curriculum sampling for person re-identification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 365–381, 2018.
  - [29] G. Wang, Y. Yuan, X. Chen, J. Li, and X. Zhou. Learning discriminative features with multiple granularities for person re-identification. In *2018 ACM Multimedia Conference on Multimedia Conference*, pages 274–282. ACM, 2018.
  - [30] L. Wei, S. Zhang, W. Gao, and Q. Tian. Person transfer gan to bridge domain gap for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 79–88, 2018.
  - [31] D. Wu, S.-J. Zheng, X.-P. Zhang, C.-A. Yuan, F. Cheng, Y. Zhao, Y.-J. Lin, Z.-Q. Zhao, Y.-L. Jiang, and D.-S. Huang. Deep learning-based methods for person re-identification: A comprehensive review. *Neurocomputing*, 2019.
  - [32] J. Yang, X. Shen, X. Tian, H. Li, J. Huang, and X.-S. Hua. Local convolutional neural networks for person re-identification. In *2018 ACM Multimedia Conference on Multimedia Conference*, pages 1074–1082. ACM, 2018.
  - [33] X. Zhang, H. Luo, X. Fan, W. Xiang, Y. Sun, Q. Xiao, W. Jiang, C. Zhang, and J. Sun. Alignedreid: Surpassing human-level performance in person re-identification. *arXiv preprint arXiv:1711.08184*, 2017.
  - [34] Z. Zhang, C. Lan, W. Zeng, and Z. Chen. Densely semantically aligned person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 667–676, 2019.
  - [35] Z. Zhang, C. Lan, W. Zeng, X. Jin, and Z. Chen. Relation-aware global attention. *arXiv preprint arXiv:1904.02998*, 2019.
  - [36] F. Zheng, X. Sun, X. Jiang, X. Guo, Z. Yu, and F. Huang. A coarse-to-fine pyramidal model for person re-identification via multi-loss dynamic training. *arXiv preprint arXiv:1810.12193*, 2018.
  - [37] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian. Scalable person re-identification: A benchmark. In *Proceedings of the IEEE international conference on computer vision*, pages 1116–1124, 2015.
  - [38] Z. Zheng, X. Yang, Z. Yu, L. Zheng, Y. Yang, and J. Kautz. Joint discriminative and generative learning for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2138–2147, 2019.
  - [39] Z. Zheng, L. Zheng, and Y. Yang. A discriminatively learned cnn embedding for person reidentification. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 14(1):13, 2018.

- [40] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang. Random erasing data augmentation. *arXiv preprint arXiv:1708.04896*, 2017.
- [41] Z. Zhong, L. Zheng, Z. Zheng, S. Li, and Y. Yang. Cam-style: A novel data augmentation method for person re-identification. *IEEE Transactions on Image Processing*, 28(3):1176–1190, 2018.
- [42] K. Zhou, Y. Yang, A. Cavallaro, and T. Xiang. Omni-scale feature learning for person re-identification. *arXiv preprint arXiv:1905.00953*, 2019.