

# Augmented Hard Example Mining for Generalizable Person Re-Identification

Masato Tamura  
Hitachi, Ltd.

Tomokazu Murakami  
Hitachi, Ltd.

{masato.tamura.sf, tomokazu.murakami.xr}@hitachi.com

## Abstract

Although the performance of person re-identification (Re-ID) has been much improved by using sophisticated training methods and large-scale labelled datasets, many existing methods make the impractical assumption that information of a target domain can be utilized during training. In practice, a Re-ID system often starts running as soon as it is deployed, hence training with data from a target domain is unrealistic. To make Re-ID systems more practical, methods have been proposed that achieve high performance without information of a target domain. However, they need cumbersome tuning for training and unusual operations for testing. In this paper, we propose **augmented hard example mining**, which can be easily integrated to a common Re-ID training process and can utilize sophisticated models without any network modification. The method discovers hard examples on the basis of classification probabilities, and to make the examples harder, various types of augmentation are applied to the examples. Among those examples, excessively augmented ones are eliminated by a classification based selection process. Extensive analysis shows that our method successfully selects effective examples and achieves state-of-the-art performance on publicly available benchmark datasets.

## 1. Introduction

Re-ID has received much attention thanks to its diverse applications such as surveillance and marketing. In Re-ID, pedestrian images across non-overlapping cameras are matched by features extracted from the images. Since the appearances of images drastically change due to variations in illumination, viewpoints, poses, and occlusions, it is difficult to acquire an identical feature from various images of the same pedestrian. To overcome this problem, many sophisticated methods have been proposed in the past few years [2–6, 9, 11, 15, 17–22, 24, 27, 28, 30, 36–41, 43, 44], and new approaches are being developed.

Many existing approaches assume that data from a tar-

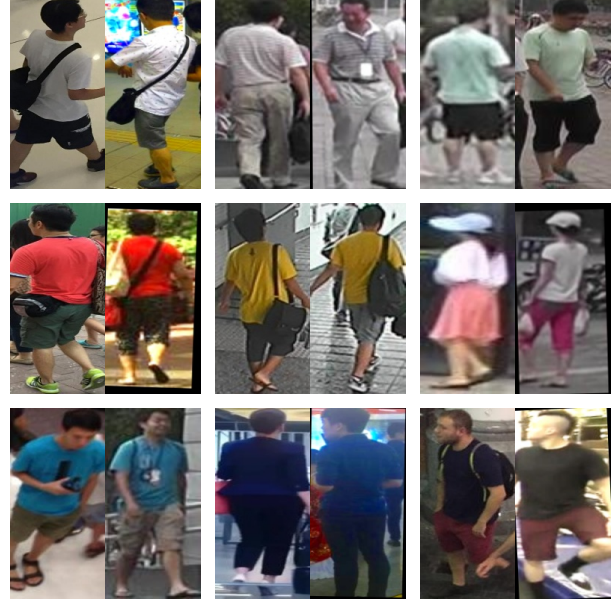


Figure 1: Examples of augmented hard examples. In each pair, the left image is the key of selecting hard examples, and the right image is a selected hard example to which various types of augmentation are applied.

get domain are available during training. Some of these approaches undergo supervised training with data from a target domain [2, 3, 5, 6, 17, 20, 24, 27, 39, 40]. They exhibit great performance when large-scale labelled datasets are prepared [18, 19, 38, 41, 43], but these approaches suggest that a large number of annotations is needed on each deployment. Since the annotation process is time consuming, these supervised approaches are infeasible for practical use. To obviate the need for annotation on each deployment, unsupervised domain adaptation (UDA) approaches have recently been proposed [4, 9, 21, 28, 36, 44]. These approaches adapt source domains to target domains by image translation, feature alignment, or multi-task learning. By this adaptation, domain-specific knowledge acquired from large-scale labelled datasets can be utilized for unlabelled datasets. The UDA approaches are more practical than the

supervised approaches, but they still need data from a target domain during training.

In practice, data from a target domain are often unavailable until deployment, hence Re-ID models have to be trained only with data in existing domains and to match identities in an unseen domain. This setting is categorized as domain generalization (DG). If Re-ID models are simply trained in a supervised manner, the domain shift between training and testing is reported to substantially degrade performance [9, 21, 36, 37, 44], which suggests that the trained Re-ID models are over-fitted and have poor generalization performance. To solve this problem, a few methods have been proposed [15, 30]. They successfully improve Re-ID accuracy by adding some operations in MobileNetV2 [29]. However, modifying a sophisticated model requires cumbersome tuning for training until satisfactory performance can be achieved. Furthermore, additional operations slow inference speed, which is a significant drawback in practical applications. Considering this, sophisticated models should not be modified.

In this paper, we propose a data augmentation based method that can enhance the generalization performance without any network modification. The problem of data augmentation lies in determining augmentation policies. If the discrepancy between the statistics of augmented images and those of real images is huge, the performance will degrade. To solve this problem, automatic augmentation methods have been proposed [7, 16, 34]. These methods are learning based approaches. Effective augmentation policies can be learned by the methods. However, complex training procedures are needed in addition to original task procedures. Different from these methods, we adopt a simple selection strategy that does not need to learn about augmentation. In our method, first, hard examples are sampled on the basis of classification probabilities of an input mini-batch. Then, various types of augmentation are applied to make the examples harder. Finally, the hardest example is selected from them. Input mini-batches are augmented with only random horizontal flipping, hence our model is basically trained with realistic images. Thanks to this, excessively augmented examples are eliminated in the final selection process. Figure 1 shows that our method successfully selects appropriate augmented examples.

Finally, we summarize our contributions as follows:

- We propose a simple selection strategy for data augmentation, which eliminates excessively augmented images. Since our method needs only one ordinary network for training and testing, existing highly optimized models can be utilized without any network modification.
- We investigate not only model accuracy but also computation cost for practical use.
- We demonstrate state-of-the-art performance on Re-ID

benchmarks and the robustness of our method against changes of augmentation parameters.

## 2. Related work

### 2.1. Domain generalized person re-identification

Although Re-ID has been researched for years, only a few methods focus on generalization performance [15, 30]. In [30], Song et al. proposed a meta-learning [33] based model called domain-invariant mapping network (DIMN). Different from a common way that uses feature distances for matching scores, DIMN generates classifier weights from gallery images and then takes the dot product of the classifier weights and probe image features to calculate matching scores. This meta-learning pipeline makes the model domain-invariant, but the complicated meta-learning procedures make optimization difficult. In addition, classifier weight generation during testing slows the inference speed. Considering these drawbacks, a simpler approach that utilizes normalization was proposed by Jia et al. [15]. They regard style and content variations as the cause of domain bias and suppress them by inserting instance normalization (IN) [35] to bottlenecks in shallow layers and a batch normalization (BN) [14] to a feature extraction layer. The evaluation results show that normalization successfully eliminates domain bias and improves the accuracy, but they also show that the location and amount of IN are important. As stated by Nam and Kim [25], insertion of instance normalization should be carefully investigated because excessive normalization suppresses styles that are the key factors to discriminate objects. This investigation process is cumbersome. Furthermore, both IN and BN add computation cost, hence inference speed slows down. Different from these methods, our method adopts a data augmentation based method and does not need any network modification, which makes our method more practical.

### 2.2. Automatic data augmentation

Although data augmentation effectively enhances generalization performance, the types and their parameters are difficult to determine. In common cases, they are determined by intuition or trial-and-error operation with validation images, but the results of this approach are unstable and troublesome. To solve this problem, automatic data augmentation methods have been proposed [7, 16, 34]. Lemley et al. [16] proposed a network that merges two or more samples to generate an augmented sample. Tran et al. [34] also proposed a generation based method, but they used a Bayesian approach and generative adversarial networks [10] for generation. Different from these two methods, Cubuk et al. [7] employed a searching strategy. In this method, appropriate data augmentation policies are investigated by reinforcement learning [32] with a recurrent neural

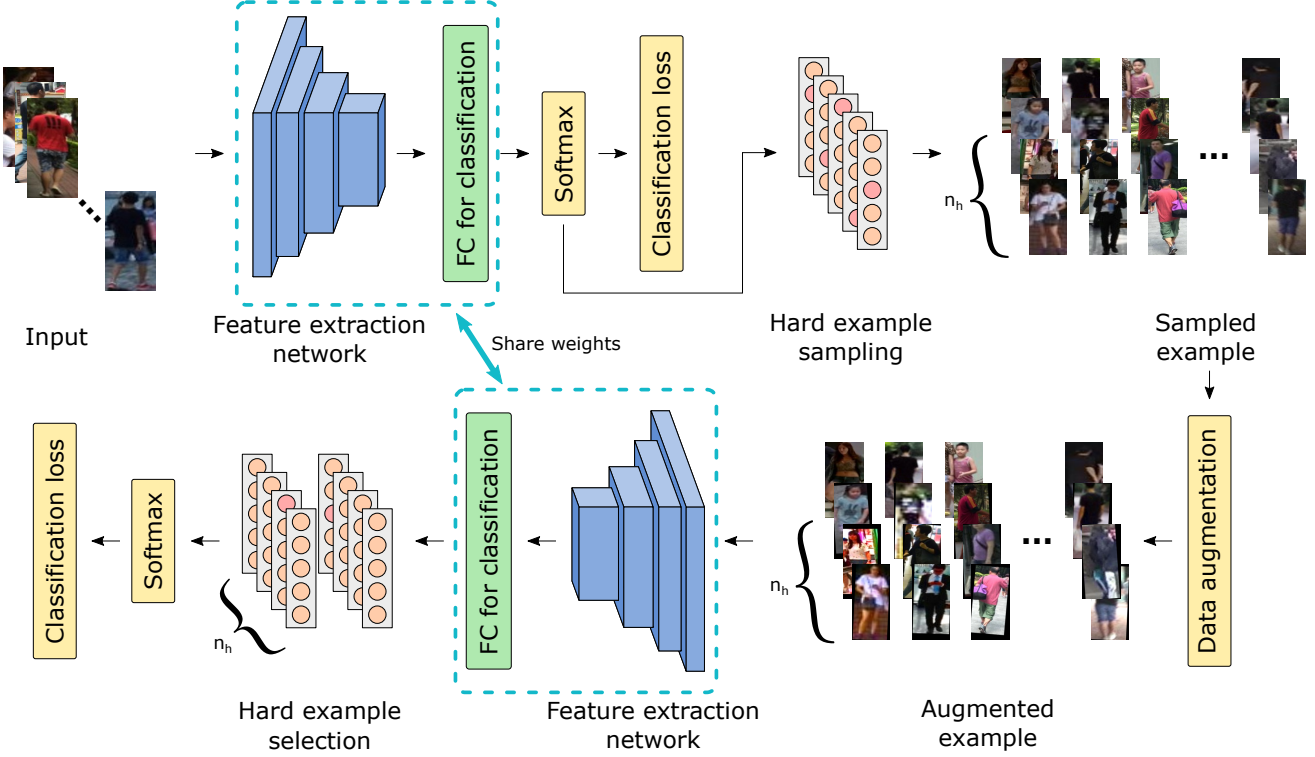


Figure 2: Structure of the proposed method.

network controller. All the methods improve performance, but they need additional networks that have to be trained for data augmentation. This makes original task training complicated. Unlike these methods, our method does not need any additional networks for data augmentation and can be easily integrated into a common training process.

### 3. Proposed method

#### 3.1. Overview

For setting domain generalized Re-ID, we assume that we have  $K$  source domains (datasets)  $\mathcal{D} = \{D_i \mid i = 1, 2, \dots, K\}$ . Each domain  $D_i = \{X^{(i)}, Y^{(i)}\}$  contains image-label pairs and has its own label space  $y^{(i)} \in \{l_j^{(i)} \mid j = 1, 2, \dots, M^{(i)}\}$ , where  $M^{(i)}$  is the number of identities in  $D_i$ . Since each label space is disjointed from others, we take the union of the label spaces for a training label space. As a result, the size of the label space becomes as follows:

$$N = \sum_{i=1}^K M^{(i)}. \quad (1)$$

For a simple yet strong baseline, we take a naive deep learning approach called aggregation (AGG). In AGG, a model is trained to minimize cross-entropy (CE) loss of all

identities from all domains:

$$L_{CE} = \frac{1}{n_{bs}} \sum_{i=1}^{n_{bs}} l^{(CE)}(g_{\phi}(f_{\theta}(x_i)), y_i). \quad (2)$$

Here,  $n_{bs}$  is the number of images in a mini-batch,  $f_{\theta}$  is a feature extractor parameterized by  $\theta$ , and  $g_{\phi}$  is a classifier parameterized by  $\phi$ . After training, the feature extractor  $f_{\theta}$  is used to extract features from images. Then the features are used to calculate matching scores as follows:

$$s = 1 - \frac{\|\hat{z}_p - \hat{z}_g\|}{2}. \quad (3)$$

Here,  $\hat{z}_p$  is an L2 normalized feature of a probe image, and  $\hat{z}_g$  is that of a gallery image. Gallery images having high scores are considered to be the images of the same identity in the probe image.

On the basis of this baseline, we propose two methods: hard example mining with CE loss (Sec.3.2) and augmented hard example selection (Sec.3.3). We explain them in the following sections.

#### 3.2. Hard example mining with cross entropy loss

Hard example mining is a way to improve the performance in borderline cases and enhance generalization performance. Although the importance of hard example mining with triplet loss is mentioned by Hermans et al. [12], CE

loss based hard example mining has never been explored. In this section, we explain how to select hard examples during training with CE loss.

As shown in Fig. 2, first, a mini-batch is input to a feature extraction network, and then the extracted features are input to a fully-connected (FC) layer. Classification loss of the mini-batch  $L_{batch}$  is calculated in accordance with Eq. 2. The outputs of Softmax are recalculated for hard example mining. The recalculation is as follows:

$$\Pr(i | x_{id=j}) = \frac{e^{p_i}}{\sum_{k \neq j} e^{p_k}} \quad (i \neq j). \quad (4)$$

Here,  $i$  is the class index of an identity,  $j$  is the class index of an identity in the input mini-batch, and  $p$  is the output of the FC layer. On the basis of this probability, hard examples against identities in the input mini-batch are selected. Concretely,  $n_h$  identities are sampled with replacement in accordance with the probability calculated by Eq. 4, and then an image of each sampled identity is randomly selected from the images of the identity. A new mini-batch is created by collecting images of hard examples sampled against all the identities in the input mini-batch, and CE loss of the new mini-batch is calculated in the same iteration.

In the initial stage of training, there are no clues for Re-ID, hence the described sampling method works as random sampling. As the training proceeds, the probability calculated by Eq. 4 indicates the similarity of identities, hence the method works as hard example mining. These characteristics well fit deep learning training.

### 3.3. Augmented hard example selection

Our method has two augmentation policies: one for input mini-batches, and the other for hard examples. Since random horizontal flipping exactly has a positive impact, it is included in both policies. As for other augmentation methods, the impacts are unknown, hence they are applied to only hard examples, and excessively augmented examples are eliminated in a selection process.

As shown in Fig. 2, images of a mini-batch created by the hard example mining are augmented before being input to a feature extraction network. Then, the augmented images are input to the network and classified by the FC layer. Note that the weights of the network and the FC layer are shared with those of the network and the FC layer used for an input mini-batch. For selecting appropriate hard examples, the outputs of the FC layer are used.

Suppose that there are  $n_h$  images of identities sampled as hard examples against one identity whose class index is  $i$ , and outputs of the images from the FC layer are denoted by  $\mathcal{Q} = \{q^{(i,j)} | j = 1, 2, \dots, n_h\}$ . From the outputs, one output is selected as follows:

$$k_i = \arg \max_j q_i^{(i,j)}. \quad (5)$$

Table 1: Dataset statistics.

(a) Training datasets.

Dataset	#IDs	#Images
CUHK02	1,816	7,264
CUHK03	1,467	14,097
Duke MTMC	1,812	36,411
Market1501	1,501	29,419
PersonSearch	11,934	34,574
	18,530	121,765

(b) Test datasets. (“Pr.”: Probe, “Ga.”: Gallery)

Dataset	#Pr. IDs	#Ga. IDs	#Pr. images	#Ga. images
VIPeR	316	316	316	316
PRID	100	649	100	649
GRID	125	900	125	900
i-LIDS	60	60	60	60

Here,  $q_i^{(i,j)}$  is the  $i$ -th entry of  $q^{(i,j)}$ . By using the selected outputs, loss of the augmented hard examples is calculated as follows:

$$L_{aug} = \frac{1}{n_{bs}} \sum_{i=1}^{n_{bs}} l^{(CE)}(q^{(i,k_i)}, y^{(i,k_i)}). \quad (6)$$

Here,  $n_{bs}$  is the number of images in a mini-batch, and  $y^{(i,k_i)}$  is the label of the selected hard example.

Since an example is selected from augmented hard examples on the basis of the similarity of identities, the example can be harder, and at the same time, excessively augmented examples are eliminated. By combining hard example mining and data augmentation, the two methods work complementarily. As a result, a trained model can robustly discriminate similar identities.

Finally, total loss is calculated as follows:

$$L_{total} = \frac{(L_{batch} + L_{aug})}{2}. \quad (7)$$

## 4. Experiments

### 4.1. Datasets and evaluation settings

To evaluate our method, we follow the settings described by Jia et al. [30] and Song et al. [15]. In the settings, large-scale datasets are combined to train Re-ID models, and small-scale datasets are individually used to evaluate model performance. The statistics of training and evaluation datasets are shown in Tables 1a and 1b, respectively. For training, CUHK02 [18], CUHK03 [19], Duke MTMC [43], Market1501 [41], and PersonSearch [38] are used. All the datasets have more than a thousand identities and thousands of images. By combining the datasets, Re-



Table 2: The types of data augmentation applied to images sampled in hard example mining.

Type	Parameter
Random crop	Edge offset: $-10-10$
Random horizontal flip	Probability: $0.5$
Random rotation	Degree: $-5^{\circ}-5^{\circ}$
	Hue value: $-0.1-0.1$
Random color jitter	Saturation scale: $0.5-2.0$
	Value scale: $0.5-2.0$

ID models are trained with 121,765 images of 18,530 identities. For evaluation, VIPeR [11], PRID [13], GRID [22], and i-LIDS [42] are used. They are relatively small-scale datasets and have at most a thousand identities. From the identities in each dataset, probe identities and gallery identities are randomly sampled in accordance with the number shown in Table 1b. With the sampled identities, Re-ID models are evaluated in a single-shot Re-ID manner. We do the sampling and the evaluation 10 times for each dataset and average the results.

## 4.2. Evaluation metrics

To show Re-ID model performance, we use cumulative matching characteristics (CMC). CMC shows Re-ID accuracy for each rank  $k$ .  $k$  is set to 1, 5, and 10.

## 4.3. Implementation details

We use MobileNetV2 [29] as a feature extraction network. Two width multipliers, which are 0.75 and 1.0, are used to analyze computation cost. For training, the FC layer of the original MobileNetV2 is replaced with a FC layer that has units equal in number to the identities in the training datasets. The network is fine-tuned from weights pretrained on ImageNet [8] using the combined dataset described in Sec. 4.1 for 30 epochs. The initial learning rate is set to 0.01 and decayed by 0.1 after 20 epochs. To optimize the network, we use stochastic gradient descent with momentum, which is set to 0.9. Input images are resized to  $256 \times 128$ . Batch size is set to 16. To prevent over-fitting, weight decay, label smoothing [23, 43], dropout [31], and data augmentation are used. The weight decay rate is set to 0.0005, smoothing value is set to 0.1, and dropout rate is set to 0.5. As for data augmentation, images in input mini-batches are horizontally flipped with a probability of 0.5. On the other hand, images selected in hard example mining are augmented by various types of augmentation, which are detailed in Table 2. Random cropping, random flipping, random rotation, and random color jitter are used. If a parameter is denoted with a range, an applied value is uniformly sampled within the range at every augmentation process. For random cropping, an offset from each edge is determined by a sampled value. If offset positions

are the outside of an image, the image is padded with zero values. The number of augmented hard examples for each identity, which is denoted by  $n_h$ , is set to 4. To prevent all the augmented hard examples from being excessively augmented, one is augmented by only random horizontal flipping. For evaluation, extracted features are L2 normalized before matching scores are calculated. Note that we do not use any test-time data augmentation.

## 4.4. Comparison against state-of-the-art

To demonstrate the superiority of our method, we compare it with previously proposed baselines. For Re-ID, three types of approaches have been proposed. The types are as follows:

**Supervised training with a target dataset** This is the most basic type and has been researched for years. Although high performance is realized with a large-scale dataset, the performance is still low with a small-scale dataset. To solve this problem, many methods have been proposed [2, 3, 5, 6, 20, 24, 27, 39, 40]. The upper part of Table 3 shows their benchmark results. Among them, SpindleNet [40], SSM [2], and JLML [20] perform well. Since they have different settings from our method, fair comparison is difficult. However, we show their results as references. Except for PRID, our method shows competitive or even better results. This means that domain specific characteristics can be covered by combining multiple large-scale datasets and appropriate data augmentation.

**Unsupervised domain adaptation** The purpose of UDA is to transfer knowledge from large-scale labelled datasets to unlabelled datasets. In accordance with this purpose, some UDA approaches have been proposed for Re-ID [4, 21, 28, 36]. The middle part of Table 3 shows their benchmark results. Synthesis [4] performs the best among them by utilizing a synthetic dataset. The same as the supervised training with a target dataset, the UDA methods have different settings from ours. However, we show their results as references. For all the benchmark datasets, our method outperforms the UDA methods. This means that our method can competitively utilize large-scale datasets.

**Domain generalization** DG setting has the most practical assumption that a target dataset cannot be seen during training. Because of this setting, DG methods have to learn general feature representation from existing datasets. For this purpose, a few methods have been proposed [15, 30], and our method is also evaluated under this setting. The lower part of Table 3 shows the benchmark results of the methods. In this comparison, we set the width multiplier of MobileNetV2 to 1.0. We put the AGG result of each method to show that the baselines are almost the same in all the methods. DualNorm [15] outperforms the others for VIPeR and PRID, while our method outperforms the others for GRID and i-LIDS. These results demonstrate the effectiveness of

Table 3: Comparison results against baselines. (“R”: Rank, “S”: Supervised training with a target dataset, “U”: UDA, “DG”: Domain generalization, “-”: No report)

Method	Type	VIPeR			PRID			GRID			i-LIDS		
		R-1	R-5	R-10	R-1	R-5	R-10	R-1	R-5	R-10	R-1	R-5	R-10
Ensembles [27]	S	45.9	77.5	88.9	17.9	40.0	50.0	-	-	-	50.3	72.0	82.5
DNS [39]	S	42.3	71.5	82.9	29.8	52.9	66.0	-	-	-	-	-	-
ImpTrpLoss [6]	S	47.8	74.4	84.8	22.0	-	47.0	-	-	-	60.4	82.7	90.7
GOG [24]	S	49.7	<b>79.7</b>	88.7	-	-	-	24.7	47.0	58.4	-	-	-
MTDnet [5]	S	47.5	73.1	82.6	32.0	51.0	62.0	-	-	-	58.4	80.4	87.3
OneShot [3]	S	34.3	-	-	41.4	-	-	-	-	-	51.2	-	-
SpindleNet [40]	S	53.8	74.1	83.2	<b>67.0</b>	<b>89.0</b>	<b>89.0</b>	-	-	-	66.3	86.6	91.8
SSM [2]	S	53.7	-	<b>91.5</b>	-	-	-	27.2	-	61.2	-	-	-
JLML [20]	S	50.2	74.2	84.3	-	-	-	37.5	61.4	69.4	-	-	-
UCTL [28]	U	31.5	-	-	24.2	-	-	-	-	-	49.3	-	-
TJAIDL [36]	U	38.5	-	-	34.8	-	-	-	-	-	-	-	-
MMFAN [21]	U	39.1	-	-	35.1	-	-	-	-	-	-	-	-
Synthesis [4]	U	43.0	-	-	43.0	-	-	-	-	-	56.5	-	-
AGG (DIMN) [30]	DG	42.9	61.3	68.9	38.9	63.5	75.0	29.7	51.1	60.2	69.2	84.2	88.8
AGG (DualNorm) [15]	DG	42.1	-	-	27.2	-	-	28.6	-	-	66.3	-	-
AGG (Ours)	DG	42.4	61.1	69.2	22.3	45.2	54.3	31.4	49.8	58.7	69.8	88.3	93.5
DIMN [30]	DG	51.2	70.2	76.0	39.2	67.0	76.7	29.3	53.3	65.8	70.2	89.7	94.5
DualNorm [15]	DG	<b>53.9</b>	-	-	60.4	-	-	41.4	-	-	74.8	-	-
Ours	DG	49.8	70.8	77.0	34.3	56.2	65.7	<b>46.6</b>	<b>67.5</b>	<b>76.1</b>	<b>76.3</b>	<b>93.0</b>	<b>95.3</b>

Table 4: Computation cost comparison against DG baselines. (“W”: Width multiplier, “MAdd”: Multiply-adds, “R”: Rank)

Method	W	MAdd	Time	VIPeR			PRID			GRID			i-LIDS		
				R-1	R-5	R-10	R-1	R-5	R-10	R-1	R-5	R-10	R-1	R-5	R-10
DIMN [30]	1.4	1523M	2.23 ms	51.2	70.2	76.0	39.2	<b>67.0</b>	<b>76.7</b>	29.3	53.3	65.8	70.2	89.7	94.5
DualNorm [15]	1.0	791M	2.68 ms	<b>53.9</b>	-	-	<b>60.4</b>	-	-	41.4	-	-	74.8	-	-
Ours	0.75	<b>543M</b>	<b>2.06 ms</b>	49.6	69.6	75.2	33.5	51.7	63.0	41.1	61.3	69.0	<b>77.2</b>	91.3	95.0
Ours	1.0	783M	2.10 ms	49.8	<b>70.8</b>	<b>77.0</b>	34.3	56.2	65.7	<b>46.6</b>	<b>67.5</b>	<b>76.1</b>	76.3	<b>93.0</b>	<b>95.3</b>

our method.

#### 4.5. Computation cost analysis

For practical use, inference time is an important factor for Re-ID performance. We compare the computation cost and the inference time of the models in the DG setting. For fair comparison, we set the input image size to  $256 \times 128$  for all the methods. Table 4 shows the comparison results. Multiply-adds (MAdd) is estimated by Tensorflow profiler [1]. For calculating inference time, we use RTX 2080Ti with CUDA ver. 10.0 [26]. The inference time is the time it takes to calculate a matching score for one pair of a probe image and a gallery image. To analyze computation cost, the results of our method with 0.75 and 1.0 width multipliers are shown in the table.

Our method has a shorter inference time than the other

two, because we do not add any operations to the original MobileNetV2. The magnitude of the difference is only 0.1 ms, but it accumulates while matching scores are calculated for all the pairs of probe images and gallery images. Considering this, our method is more practical than the other two.

The difference in inference time between DualNorm and ours with the 1.0 width multiplier (2.68 vs. 2.10 ms) is larger than that between ours with the 0.75 and 1.0 width multipliers (2.06 vs. 2.10 ms) even though MAdd of DualNorm is almost the same as that of ours with the 1.0 width multiplier (791M vs. 783M). This means that the instance normalization causes slow inference speed. In general, unusual operations are not optimized for high speed computation in usual deep learning libraries, hence they take a long time regardless of computation cost. Since the optimization

Table 5: Ablation study on the impact of different components. In the table, only rank-1 accuracy is shown. (“Aug.”: Augmented)

Component	VIPeR	PRID	GRID	i-LIDS
Baseline	42.4	22.3	31.4	69.8
Augment	43.0	29.0	36.4	71.2
Mining	47.3	27.4	38.2	73.5
Augment + mining	47.3	28.7	41.4	74.5
Aug. mining select	49.8	34.3	46.6	76.3

process is cumbersome, practical models should be composed of usual operations (e.g., convolution and batch normalization). From this point of view, our method has an advantage.

#### 4.6. Ablation study

To analyze the effect of each component in our method, we evaluate rank-1 accuracy with each component. In this evaluation, the width multiplier is set to 1.0. Table 5 shows the evaluation results. Each component is as follows:

**Baseline** AGG.

**Augment** The proposed hard example mining is not carried out. Instead, all the images in input mini-batches are augmented by the methods shown in Table 2.

**Mining** The proposed hard example mining is carried out, but sampled hard examples are augmented by only random horizontal flipping. In this case,  $n_h$  is set to 1, and the selection process is skipped.

**Augment + mining** Combination of **Augment** and **Mining**. This means that all the images input to a network are augmented by the methods shown in Table 2.

**Aug. mining select** The proposed method.

We can see from the results of **Augment** and **Mining** that both the data augmentation and the proposed hard example mining improve the generalization performance. However, the results of **Augment + mining** show that just combining the two methods does not improve the performance much from each method. Compared to **Augment + mining**, **Aug. mining select** has better effect on the performance. This means that our method successfully selects hard examples that have positive impact on the performance, and the proposed selection is important for the improvement.

In total, our method improves the rank-1 accuracy of **Baseline** by 7.4%, 12.0%, 15.2%, and 6.5% for VIPeR, PRID, GRID, and i-LIDS, respectively.

#### 4.7. Robustness of selection strategy

To show the robustness of our selection strategy, we train the model with three patterns of data augmentation and evaluate its performance. The patterns are shown in Table 6. The moderate data augmentation is the same as the augmentation shown in Table 2. The parameter ranges of weak

Table 6: Data augmentation patterns. (“H”: Hue value, “S”: Saturation scale, “V”: Value scale)

		Weak	Moderate	Strong
Crop		-5-5	-10-10	-15-15
Rotation		0°	-5°-5°	-10°-10°
Color	H	-0.05-0.05	-0.1-0.1	-0.15-0.15
	S	0.67-1.5	0.5-2.0	0.4-2.5
	V	0.67-1.5	0.5-2.0	0.4-2.5

Table 7: Comparison of three augmentation patterns with and without the proposed method. In the table, only rank-1 accuracy is shown. (“min.”: mining)

Pattern	VIPeR	PRID	GRID	i-LIDS
Weak w/o min. select	45.2	30.2	37.2	72.0
Moderate w/o min. select	43.0	29.0	36.4	71.2
Strong w/o min. select	40.8	28.1	37.3	69.7
Weak w/ min. select	49.2	36.2	43.0	76.2
Moderate w/ min. select	49.8	34.3	46.6	76.3
Strong w/ min. select	48.8	36.2	45.4	75.7

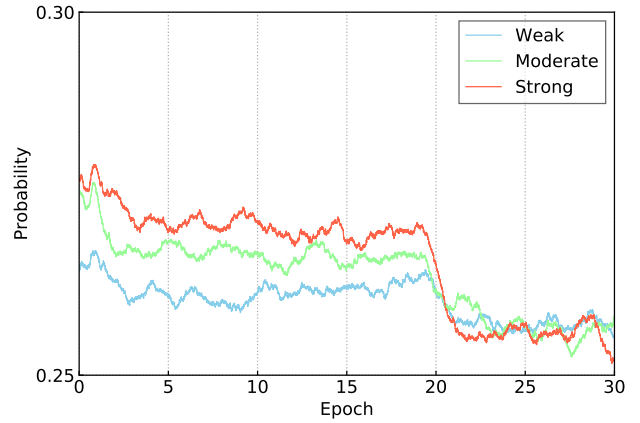


Figure 3: Probabilities of selecting an example augmented by only random horizontal flipping for three augmentation patterns.

and strong data augmentation are narrower and broader than that of moderate data augmentation, respectively. In all the patterns, random horizontal flipping is used with probability of 0.5.

Table 7 shows the evaluation results. The upper three rows show the results of usual input data augmentation with the three patterns, and the lower three rows show those of the proposed method with the three patterns. With the input data augmentation, stronger data augmentation degrades the performance except for GRID, whereas with the hard example data augmentation, stronger data augmentation does not change the performance much or even improves it. This

shows that our method broadened the acceptable range of data augmentation and that our selection strategy is robust.

To further clarify this consideration, we examine the probability of selecting the example augmented by only random horizontal flipping. Figure 3 shows the probability. Since  $n_h$  is set to 4, one example is supposed to be selected with the probability of 0.25, but the probability is higher. In addition, the stronger augmentation makes the probability higher until the learning rate is decayed. This means that as the probability of containing excessively augmented images becomes higher, the probability of selecting realistic images becomes higher. We can see from this result that our method works as intended. After the learning rate is decayed, the probabilities of all the augmentation patterns become the same. We will investigate the cause and effect of this in our future work.

## 5. Conclusion

In this paper, we have proposed a simple selection strategy for data augmentation to improve Re-ID performance. In our method, various augmentation methods are applied to only hard examples sampled on the basis of classification probabilities, and excessively augmented examples are eliminated as easy examples. Since our method uses classification probability for selection, it can be easily integrated into a common training process. In addition, our method does not need any unusual operations in networks, so highly optimized models can be utilized without any modification. Experiments on four public benchmark datasets show that our method can achieve state-of-the-art performance for practical use in Re-ID.

## References

- [1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Man, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Vyas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. Tensorflow: Large-scale machine learning on heterogeneous distributed systems, 2015. 6
- [2] S. Bai, X. Bai, and Q. Tian. Scalable person re-identification on supervised smoothed manifold. In *CVPR*, July 2017. 1, 5, 6
- [3] S. Bak and P. Carr. One-shot metric learning for person re-identification. In *CVPR*, July 2017. 1, 5, 6
- [4] S. Bak, P. Carr, and J.-F. Lalonde. Domain adaptation through synthesis for unsupervised person re-identification. In *ECCV*, September 2018. 1, 5, 6
- [5] W. Chen, X. Chen, J. Zhang, and K. Huang. A multi-task deep network for person re-identification. In *AAAI*, February 2017. 1, 5, 6
- [6] D. Cheng, Y. Gong, S. Zhou, J. Wang, and N. Zheng. Person re-identification by multi-channel parts-based cnn with improved triplet loss function. In *CVPR*, June 2016. 1, 5, 6
- [7] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le. Autoaugment: Learning augmentation strategies from data. In *CVPR*, June 2019. 2
- [8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, June 2009. 5
- [9] W. Deng, L. Zheng, Q. Ye, G. Kang, Y. Yang, and J. Jiao. Image-image domain adaptation with preserved self-similarity and domain-dissimilarity for person re-identification. In *CVPR*, June 2018. 1, 2
- [10] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, December 2014. 2
- [11] D. Gray and H. Tao. Viewpoint invariant pedestrian recognition with an ensemble of localized features. In *ECCV*, October 2008. 1, 5
- [12] A. Hermans, L. Beyer, and B. Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017. 3
- [13] M. Hirzer, C. Belezni, P. M. Roth, and H. Bischof. Person re-identification by descriptive and discriminative classification. In *SCIA*, May 2011. 5
- [14] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, July 2015. 2
- [15] J. Jia, Q. Ruan, and T. M. Hospedales. Frustratingly easy person re-identification: Generalizing person re-id in practice. In *BMVC*, September 2019. 1, 2, 4, 5, 6
- [16] J. Lemley, S. Bazrafkan, and P. M. Corcoran. Smart augmentation learning an optimal data augmentation strategy. *IEEE Access*, 5:5858–5869, 2017. 2
- [17] K. Li, Z. Ding, S. Li, and Y. Fu. Discriminative semi-coupled projective dictionary learning for low-resolution person re-identification. In *AAAI*, February 2018. 1
- [18] W. Li and X. Wang. Locally aligned feature transforms across views. In *CVPR*, June 2013. 1, 4
- [19] W. Li, R. Zhao, T. Xiao, and X. Wang. DeepReID: Deep filter pairing neural network for person re-identification. In *CVPR*, June 2014. 1, 4
- [20] W. Li, X. Zhu, and S. Gong. Person re-identification by deep joint learning of multi-loss classification. In *IJCAI*, August 2017. 1, 5, 6
- [21] S. Lin, H. Li, C.-T. Li, and A. C. Kot. Multi-task mid-level feature alignment network for unsupervised cross-dataset person re-identification. In *BMVC*, September 2018. 1, 2, 5, 6
- [22] C. C. Loy, T. Xiang, and S. Gong. Multi-camera activity correlation analysis. In *CVPR*, June 2009. 1, 5
- [23] H. Luo, Y. Gu, X. Liao, S. Lai, and W. Jiang. Bag of tricks and a strong baseline for deep person re-identification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019. 5
- [24] T. Matsukawa, T. Okabe, E. Suzuki, and Y. Sato. Hierarchical gaussian descriptor for person re-identification. In *CVPR*, June 2016. 1, 5, 6



- [25] H. Nam and H.-E. Kim. Batch-instance normalization for adaptively style-invariant neural networks. In *NeurIPS*, 2018. 2
- [26] J. Nickolls, I. Buck, M. Garland, and K. Skadron. Scalable parallel programming with cuda. *Queue*, 6:40–53, 03 2008. 6
- [27] S. Paisitkriangkrai, C. Shen, and A. van den Hengel. Learning to rank in person re-identification with metric ensembles. In *CVPR*, June 2015. 1, 5, 6
- [28] P. Peng, T. Xiang, Y. Wang, M. Pontil, S. Gong, T. Huang, and Y. Tian. Unsupervised cross-dataset transfer learning for person re-identification. In *CVPR*, June 2016. 1, 5, 6
- [29] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. MobileNetV2: Inverted residuals and linear bottlenecks. In *CVPR*, June 2018. 2, 5
- [30] J. Song, Y. Yang, Y.-Z. Song, T. Xiang, and T. M. Hospedales. Generalizable person re-identification by domain-invariant mapping network. In *CVPR*, June 2019. 1, 2, 4, 5, 6
- [31] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *JMLR*, 15(1):1929–1958, January 2014. 5
- [32] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, 2018. 2
- [33] S. Thrun and L. Pratt, editors. *Learning to Learn*. Kluwer Academic Publishers, 1998. 2
- [34] T. Tran, T. Pham, G. Carneiro, L. Palmer, and I. Reid. A bayesian data augmentation approach for learning deep models. In *NIPS*, December 2017. 2
- [35] D. Ulyanov, A. Vedaldi, and V. S. Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016. 2
- [36] J. Wang, X. Zhu, S. Gong, and W. Li. Transferable joint attribute-identity deep learning for unsupervised person re-identification. In *CVPR*, June 2018. 1, 2, 5, 6
- [37] L. Wei, S. Zhang, W. Gao, and Q. Tian. Person transfer gan to bridge domain gap for person re-identification. In *CVPR*, June 2018. 1, 2
- [38] T. Xiao, S. Li, B. Wang, L. Lin, and X. Wang. Joint detection and identification feature learning for person search. In *CVPR*, July 2017. 1, 4
- [39] L. Zhang, T. Xiang, and S. Gong. Learning a discriminative null space for person re-identification. In *CVPR*, June 2016. 1, 5, 6
- [40] H. Zhao, M. Tian, S. Sun, J. Shao, J. Yan, S. Yi, X. Wang, and X. Tang. Spindle Net: Person re-identification with human body region guided feature decomposition and fusion. In *CVPR*, July 2017. 1, 5, 6
- [41] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian. Scalable person re-identification: A benchmark. In *ICCV*, December 2015. 1, 4
- [42] W.-S. Zheng, S. Gong, and T. Xiang. Associating groups of people. In *BMVC*, September 2009. 5
- [43] Z. Zheng, L. Zheng, and Y. Yang. Unlabeled samples generated by gan improve the person re-identification baseline in vitro. In *ICCV*, October 2017. 1, 4, 5
- [44] Z. Zhong, L. Zheng, S. Li, and Y. Yang. Generalizing a person retrieval model hetero- and homogeneously. In *ECCV*, September 2018. 1, 2