# Object Re-identification Using Teacher-Like and Light Students

Yi Xie[1]
1016231744@qq.com
Hanxiao Wu[2]
hxwu@stu.hqu.edu.cn
Fei Shen[3]
feishen@njust.edu.cn
Jianqing Zhu[1]
jqzhu@hqu.edu.cn
Huanqiang Zeng[1]
zeng0043@hqu.edu.cn

[1] College of Engineering, Huaqiao University, Quanzhou, China
[2] College of Information Science and Engineering, Huaqiao University, Xiamen, China
[3] School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China

Corresponding authors: Jianqing Zhu and Huanqiang Zeng

**Abstract**

Recent object re-identification (Re-ID) approaches tend to use heavy models (e.g., ResNet-50 or ResNet-101) to guarantee performance, which requires massive computations. Although knowledge distillation (KD) methods can be applied to learn light student models from heavy teacher models, numerous existing KD research has shown that significant architectural differences between students and teachers prevent students from achieving good accuracy. For that, we propose a joint distillation and pruning (JDP) method to learn teacher-like and light (TLL) students for object Re-ID. Given a heavy teacher, JDP applies a student that holds the same overall architecture but a tiny local adjustment. Specifically, we design a pruner-convolution-pruner (PCP) block to replace a $K \times K$ convolutional layer of the student network. The pruner is a $1 \times 1$ convolutional layer and initialized identity matrices to maintain the original output. During the student training phase, the student is jointly supervised by the KD loss and group LASSO loss functions. The KD loss function promotes the student to learn knowledge from the teacher. The group LASSO loss function enforces pruners to realize the channel sparsity for filtering unimportant channels. A PCP block can be simplified into a light convolutional layer during the testing phase since multiple linearly convolutional layers in series can be equivalently merged into one convolutional layer. As a result, a TLL student is acquired. Extensive experiments show that our JDP method has superiority in terms of accuracy and computations, e.g., on the VeRi-776 dataset, given the ResNet-101 as a teacher, our TLL student saves 80.00% parameters and 78.52% FLOPs, while the mAP only drops by 0.17%.

## 1 Introduction

Given an image as a query, object re-identification (Re-ID) aims to retrieve object images of the same identity from a large-scale image gallery. Due to the high demand in intelligent video surveillance systems, persons and vehicles are two dominant targets in object Re-ID. However, due to person or vehicles captured by different places, times, and camera views, both person Re-ID and vehicle Re-ID are challenging.

Although object Re-ID methods [11, 20, 31, 34] achieve significant progress, they require large testing computations because of applying a heavy network. In order to reduce testing computations, there are three critical methods: (1) knowledge distillation (KD) methods [1, 7, 9, 10, 17, 18, 19, 23, 28, 29], (2) structured pruning [5, 14, 27, 35], and (3) structural re-parameterization [2, 3, 4]. KD methods assign a light student network to learn dark knowledge from a well-trained heavy teacher network, i.e., minimizing Kullback-Leibler divergences or Euclidean distances between data resulting from the teacher and the student. However, due to significant architectural differences (e.g., great depth differences and different basic blocks) between teachers and students hinder students learning knowledge from teachers, limiting students' performance.

Unlike KD methods, structured pruning methods abandon unimportant channels to directly prune a light network from a heavy network. For example, the structured sparsity learning method [27] applies sparse functions (e.g., group LASSO [30]) on parameters of convolutional layers to find unimportant channels. Similarly, the slimming method [14] further thins models via allocating group LASSO functions on parameters of batch normalization layers. Although structured pruning methods could learn light models, they would lose accuracy performance since parameter sparse functions conflict with traditional parameter decay (i.e., $L_2$ regularization), who tend to learn evenly distributed parameters.

Recently, for boosting accuracy performance, structural re-parameterization [3, 4] methods firstly complicate a network by replacing a convolutional layer as diverse branches of different scales and complexities, e.g., sequences of convolutions, multi-scale convolutions, and average pooling. Then, structural re-parameterization [3, 4] equivalently converted those diverse branches into a single convolutional layer for accelerating inference. Although structural re-parameterization methods could improve accuracy performance and inference speed, there is still room for accelerating inference because they only convert complicated networks to the original ones. In summary, we still lack an approach to effectively combine those three critical methods to compress models and maintain good accuracy performance.

In this paper, we propose a joint distillation and pruning (JDP) method to learn teacher-like and light (TLL) students for object re-identification, which essentially combines KD, pruning and re-parameterization technologies. The student initially holds the same architecture to the teacher. We design a pruner-convolution-pruner (PCP) block to replace a $K \times K$ convolutional layer of the student network. In each PCP, group LASSO functions are applied to supervise the two new pruners rather than the convolution layer, which is to avoid accuracy performance loss caused by direct pruning the convolution layer. Besides, for further ensuring the student's performance, we apply the KD loss functions to directly guide the student to output similar feature maps to that of the teacher, without introducing any dimension uniform units. During the inference phase, we convert PCP blocks to light convolutional layers by using structural re-parameterization. Consequently, a student holding the same depth to the teacher but lighter computations is acquired, which is the so-called TTL student.Therefore, the novelty of this paper lies in the PCP block, which allows an essential combination of KD, pruning and re-parameterization technologies to obtain a good trade-off of accuracy performance and computation cost.

Our main contributions are summarized as follows. (1) We design PCP blocks to slim the teacher, and PCP blocks can be convenient simplified to light convolutional layers by structural re-parameterization. (2) We propose to jointly apply KD to supervise the PCP based student for avoiding an unconstrained pruning. (3) Extensive experiments show that our JDP method outperforms state-of-the-art methods in terms of accuracy and computations.

# 2 Related Works

## 2.1 Knowledge Distillation for Object Re-ID

Knowledge distillation (KD) methods [1, 9, 17, 23, 25] have attracted much attention, which transfers knowledge from heavy networks to light students, accelerating the network without losing too much accuracy performance. Hinton et al. [9] introduced the idea of temperature in the network's outputs to better represent information and adopt a heavy teacher's output logit values as soft labels to supervise a light student. In addition to using the teacher's output logit values, Fitnets [23] firstly uses the output of the teacher's hidden layers to supervise the output of student's hidden layers. After that, many KD methods also use the output of the hidden layer to transfer knowledge. However, the hidden layer's feature dimensions are different because of the architectural differences between students and teachers. Thus, many KD methods adopt the dimensions align operation (e.g., $1 \times 1$ convolution) to address this problem, causing that the student can not comprehensively absorb the teacher's knowledge. Many KD methods have been applied in object Re-ID tasks because object Re-ID requires a high computational cost. For example, CCKD [18] proposes transfers of the instance-level information and the correlation between instances for person Re-ID. Similarly, MBDL [28] design a matching behavior difference matrix to ensure the student simulates the teacher network's matching behaviors relationship for vehicle Re-ID. VKD [19] pins this visual variety relationship as a supervision signal within a teacher-student framework, the teacher educates students who observe fewer views. UMTS [10] designs an uncertainty-aware knowledge distillation loss (UA-KDL), which can efficiently regularize the feature learning at different semantics levels for object Re-ID. However, these methods still minimize the gap between students and teachers by designing the knowledge learning methods rather than minimize the architectural differences to narrow the gap between students and teachers.

## 2.2 Pruning and Structural Re-parameterization

Pruning techniques can be mainly categorized into unstructured pruning [5, 35] and structured pruning methods [14, 27]. Unstructured pruning [5, 35] methods remove the individual weights scattered of convolutional layers, while structured pruning methods [14, 27] filter unimportant channels via group sparse. Therefore, structured pruning are beneficial to slim a heavy network than unstructured pruning. In practice, pruning are prone to loss accuracy performance, although fine-tuning for pruned model can recover some performance.

Structural re-parameterization methods [2, 3, 4] decouple the training time and inference-time by complicating a network during the training phase and converting the complicated network back into a network of the original scale during the inference phase. The structural re-parameterization's basic is that a series of linear operations can be equivalently converted to one linear operation. For example, Ding et al. [3] proposed the diverse branch block (DBB) of multiple convolutions in the training phase and equivalently converted DBB to a single convolutional layer. Structural re-parameterization methods still have room for accelerating inference because they only complicated networks to a network of the original scale.

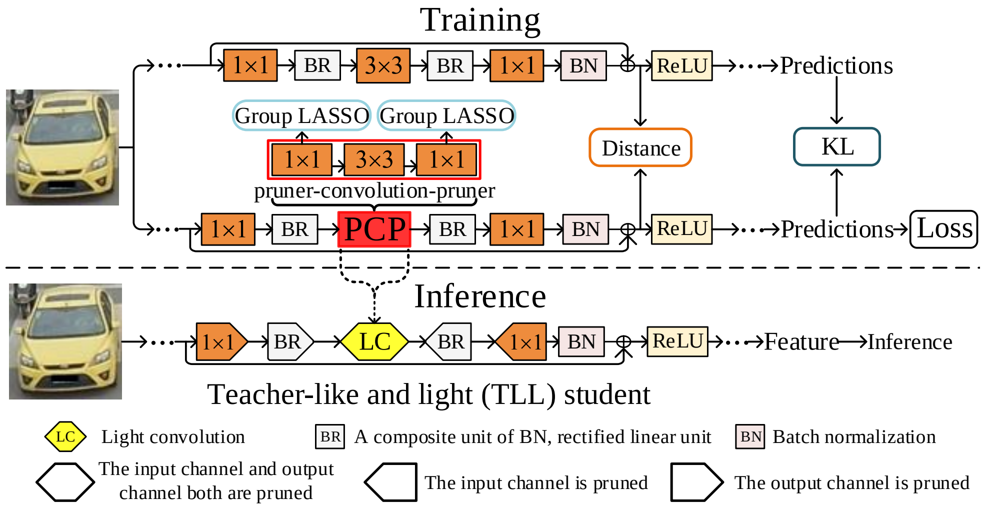
Training
1×1
BR
3×3
BR
1×1
BN
ReLU
Predictions
Group LASSO
Group LASSO
1×1
3×3
1×1
pruner-convolution-pruner
Distance
KL
1×1
BR
PCP
BR
1×1
BN
ReLU
Predictions
Loss
Inference
1×1
BR
LC
BR
1×1
BN
ReLU
Feature
Inference
Teacher-like and light (TLL) student
LC Light convolution
BR A composite unit of BN, rectified linear unit
BN Batch normalization
The input channel and output channel both are pruned
The input channel is pruned
The output channel is pruned

Figure 1: The framework of the proposed JDP method.

# 3 Approach

## 3.1 Overview

The framework of JDP is shown in Figure 1, where contains three key components including the designed pruner-convolution-pruner (PCP) block, joint knowledge distillation and pruning in training phase, and light inference. The PCP block including $1 \times 1$, $K \times K$, $1 \times 1$ convolutional layers, which will replace a $K \times K$ convolutional layer of the student network who holds the same teacher network. The number of channels of pruner of the PCP block can be pruned under the supervision of group LASSO, which indirectly controls the number of channels of the $K \times K$ convolutional layer. Furthermore, to avoid an unconstrained pruning, KD loss is added to supervise the student to promoting the student to learn knowledge from the teacher. Finally, during the student inference phase, a student holding the same depth as the teacher but lighter basic blocks can be acquired via simplifying PCP blocks' re-parameterized [3, 4] to a light $K \times K$ convolutional layer, the so-called TTL student.

## 3.2 The Pruner-Convolution-Pruner Block Using Group LASSO

Inspired by the study of re-parameterized model [3, 4], we construct a pruner-convolution-pruner (PCP) block, which including $1 \times 1$ convolutional layer, $K \times K$ convolutional layer with a bias term, and $1 \times 1$ convolutional layer to replace all $K \times K$ convolutional layer of the student that holds the same teacher network. The $K \times K$ convolutional layer's parameters of the PCP block are the same as the original $K \times K$ convolutional layer. The pruner of the PCP block is initialized as identity matrices to maintain the original output of the $K \times K$ convolutional layer.

The PCP block can equivalently convert into one convolutional layer as follows:

$$\bar{I} = Trans(Trans(I_2 \circledast Trans(I_1)) \circledast I_3), \quad \bar{b} = b \circledast I_3, \tag{1}$$

where $\bar{I} \in \mathbb{R}^{F \times C \times K \times K} \in$ and $\bar{b} \in \mathbb{R}^{F}$ represent one new convolutional layer and it's bias.

$Trans(\cdot)$ is the transpose function, such as $Trans(I_1)$ is to convert $I_1 \in \mathbb{R}^{D\times C\times 1\times 1}$ to $I_1' \in \mathbb{R}^{C\times D\times 1\times 1}$ and $\circledast$ represent the convolution operator. $I_1 \in \mathbb{R}^{D\times C\times 1\times 1}$ and $I_3 \in \mathbb{R}^{F\times E\times 1\times 1}$ represent the parameters of first pruner and last pruner. The $I_2 \in \mathbb{R}^{E\times D\times K\times K}$ and $b \in \mathbb{R}^E$ represent the $K \times K$ convolution kernel's parameters and it's bias.

From Eq. 1, we can find that the channel number of the $K \times K$ convolutional layer of the PCP block is controlled by the pruner of the PCP block. Therefore, based on the properties of the PCP block, in the student training phase, the group LASSO is applied to sparse the pruner of the PCP block to indirectly pruning the $K \times K$ convolutional layer. The pruning loss function is designed as follows:

$$L_{pruning}(P_1, P_2) = L_I(P_1) + L_O(P_2), \tag{2}$$

where $P_1 \in \mathbb{R}^{D\times C\times 1\times 1}$ and $P_2 \in \mathbb{R}^{F\times E\times 1\times 1}$ represent the first pruner and last pruner of the PCP block. $L_I$ and $L_O$ both is the group LASSO loss function, but $L_I$ enforce the input channel of pruner to realize the channel sparsity for filtering unimportant channels, while $L_O$ is enforce the output channel of pruner to realize the channel sparsity. $L_I$ and $L_O$ respectively are formulated as follows:

$$L_I(P_1) = \sum_{i=1}^{D} \sqrt{\sum_{j=1}^{C} P_{1_{i,j,1,1}}^2}; \quad L_O(P_2) = \sum_{j=1}^{E} \sqrt{\sum_{i=1}^{F} P_{2_{i,j,1,1}}^2}. \tag{3}$$

## 3.3 Joint Knowledge Distillation and Pruning

To avoid unconstrained pruning, KD is applied to supervise the student to promoting the student learn knowledge from the teacher. First, based on the PCP block, in the student training phase, the student has the same depth as the teacher but slightly modified the basic block and trained under the teacher's supervision. Therefore, the student can comprehensively absorb knowledge from the teacher because the architectural differences between the student and the teacher are slight. Such as, an obvious advantage of the slight architectural difference between students and teachers is that teachers can directly supervise the hidden layer features of students without requiring any dimension uniform unit (e.g., 1 × 1 convolutional layer). Thus, KD loss function consists of two parts as follows:

$$L_{KD} = \lambda L_{kl}(z^s, z^t) + \beta L_{dl}(F_i^s, F_i^t) \tag{4}$$

where $L_{kl}$ is the Kullback-Leibler divergence (KL) loss [9], which enables the teacher's output logit value to supervise the student's output logit value; $L_{dl}$ is the straightforward distance loss function to minimize the distance between the student's hidden layer features and the teacher's hidden layer because it is only to verify the advantages of students with similar architecture. $\lambda \geq 0$ and $\beta \geq 0$ are both a hype-parameter, respectively, used to control the contribution of $L_{kl}$ and $L_{dl}$, and their default value are set to 1 and 0.5, respectively. The distance loss $L_{dl}$ is formulated as follows:

$$L_{dl}(F_i^s, F_i^t) = \sum_{i=1}^{N} \|F_i^t - F_i^s\|_2, \quad 1 \leq i \leq N. \tag{5}$$

where $\|\cdot\|$ is Euclidean norm function; $F_i^s$ and $F_i^t$ represent the output feature by the global average pool layer of the $i$-th block of the student and the teacher; $N$ is block number of the network.

Therefore, during the student training phase, the student's total loss function as follows:

$$L_{student} = L_{lsrce} + L_{triplet} + \lambda L_{kl} + \beta L_{dl} + \alpha L_{pruning}, \tag{6}$$

where $L_{lsrce}$ is cross-entropy loss function with using the label smooth regularization and the label smooth constant is set to 0.1, as done in [24]; $L_{triplet}$ is the triplet loss function using hard sample exploring strategy [8]. The default value of $\alpha$ is $2 \times 10^{-3}$.

### 3.4 Teacher-Like and Light Student during Inference

According to the re-parameterization formulated in Eq. 1, all PCP blocks can be simplified as light $K \times K$ convolutional layers. Besides, if the channel of the pruner of the PCP block with a weight value is less than the threshold $\tau$, the corresponding channel is disregard. In this process, those adjacent layers before and after each PCP are pruned accordingly, as shown in Figure 1. For example, for the $1 \times 1$ convolutional layer before PCP, its output can be pruned because the first pruner of PCP knows the channel weights. As a result, a teacher-like and light student is constructed during the inference phase.

## 4 Experiments

To validate our JDP method's superiority, we conduct amounts of experiments on three large scale Re-ID datasets, including DukeMTMC-reID [22], MSMT17 [26] and VeRi-776 [13]. In this section, we firstly introduce these datasets, and the experiment implementation details. Then, we compare the proposed JDP with the state-of-the-arts methods. Furthermore, we conduct ablation experiments to analyze the effectiveness different components of JDP. Finally, we evaluate the effect of several important hyper-parameters.

### 4.1 Datasets and Evaluation Protocol

**DukeMTMC-reID** [22] is a large-scale labeled multi-target multi-camera pedestrian Re-ID dataset derived from DukeMTMC [22], which includes comprises of 36,411 pedestrian images of 1,404 identities. It is split into training subset and test subset. The training subset contains 16,522 training images of 702 identities, while the test subset contains 2,228 query images of 702 identities and 17,661 gallery images of 1,100 identities.

**MSMT17** [26] contains contains 126441 images of 4101 pedestrian identities captured by 3 indoor cameras and 12 outdoor cameras. It is split into training subset and test subset. The training subset includes 32,621 training images of 1,041 identities, while the test subset include 11,659 query images and 82,161 gallery images of 3,060 identities.

**VeRi-776** [13] is constructed by 20 cameras in the unconstrained traffic scenarios. VeRi-776 is divided into a training subset and a testing subset. The training subset contains 37,746 images of 576 subjects. The test subset includes a probe subset of 1,678 images of 200 subjects and a gallery subset of 11,579 images of the same 200 subjects.

The cosine distance is applied as the similarity metric. The rank-1 identification rate (R1) [13] and mean average precision (mAP) are used to assess the accuracy performance. Model parameters (MP), floating-point of operations (FLOPs), and the feature extraction time (FET) per image [33] are used to measure the model size, the computational complexity and the real inference time, respectively.

### 4.2 Implementation Details

The network training configuration has some differences between pedestrian Re-ID task and vehicle Re-ID task. The common training configuration of the three datasets is as follows.

| Methods | Teacher | Student | DukeMTMC | | | | MSMT17 | | | | VeRi-776 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Models | Models | MP | FLOPs | mAP | R1 | MP | FLOPs | mAP | R1 | MP | FLOPs | mAP | R1 |
| FD [21] | ResNet-101 | ResNet-101 | 43.50 | 6.50 | 76.32 | 87.07 | - | - | - | - | - | - | - | - |
| CCKD [18] | ResNet50 | ResNet-18 | - | - | - | - | 11.44 | 1.99 | 30.7 | 59.4 | - | - | - | - |
| UMTS [10] | ResNet-50 | ResNet-50 | - | - | - | - | - | - | - | - | 25.54 | 8.13 | 75.9 | **95.8** |
| VKD [19] | ResNet-101 | ResNet-101 | - | - | - | - | - | - | - | - | 42.50 | 12.99 | **80.62** | 95.53 |
| Teacher | - | ResNet-101 | 43.50 | 6.49 | **77.44** | **88.06** | 43.50 | 6.49 | **54.08** | **77.19** | 43.50 | 12.99 | 78.68 | 95.65 |
| LKD [9] | ResNet-101 | ResNet-34 | 21.54 | 3.66 | 73.27 | 85.86 | 21.54 | 3.66 | 46.55 | 72.52 | 21.54 | 7.31 | 75.15 | 94.76 |
| Fitnets [23] | ResNet-101 | ResNet-34 | 21.54 | 3.66 | 73.37 | 85.77 | 21.54 | 3.66 | 49.08 | 74.26 | 21.54 | 7.31 | 76.42 | 94.99 |
| MBDL [28] | ResNet-101 | ResNet-34 | 21.54 | 3.66 | 76.10 | 87.57 | 21.54 | 3.66 | 50.58 | 74.32 | 21.54 | 7.31 | 77.08 | 95.35 |
| JDP | ResNet-50 | TLL | **11.05** | **1.94** | 76.36 | 86.94 | **7.58** | **1.45** | 49.81 | 73.57 | **6.08** | **2.27** | 77.66 | 95.35 |
| JDP | ResNet-101 | TLL | 15.94 | 2.47 | 77.20 | 87.43 | 11.22 | 1.80 | 52.50 | 75.43 | 8.70 | 2.79 | 78.51 | 94.99 |

Table 1: Comparison with state-of-the-art methods on three datasets.

(1) ResNet [6] is applied to evaluate the performance of the proposed JDP. (2) ResNet [6] using 'last stride=1' training trick [16] and pre-trained on ImageNet [12]. (3) The z-score normalization, random cropping, random erasing [32], and random horizontal flip operations are implemented for the data augmentation. The probabilities of horizontal flip and random erasing operations are both set to 0.5. (4) The mini-batch stochastic gradient descent method [12] is applied to optimize parameters. The mini-batch size is set to 128, including 32 identities, and each identity contains four images. (5) The weight decays are set to $5 \times 10^{-4}$, and the momentums are set to 0.9. (6) the hyper-parameters $\tau$ is set to $1 \times 10^{-3}$. (7) The cosine annealing strategy [15] is applied to the learning rate of the network. Specifically, the initial learning rate is set to $2 \times 10^{-2}$ and we adopt the warmup learning strategy and spend 10 epochs linearly increasing the learning rate from $2 \times 10^{-3}$ to $2 \times 10^{-2}$.

There are some different configurations on different datasets. For Duke MTMC-reID [22] and MSMT17 [26], the special training configuration as follows. (1) the resolution of input images is set to $256 \times 128$. (2) The decay epoch of cosine annealing strategy [15] is set to 50-th epoch and total training epoch is 180. For the VeRi-776 [13], the special training configuration as follows. (1) the resolution of input images is set to $256 \times 256$. (2) The decay epoch of cosine annealing strategy [15] is set to 40-th epoch and total training epoch is 120.

## 4.3 Comparison with the State-of-the-Art

Table 1 shows the performance comparison of the proposed JDP with the state-of-the-art methods on three datasets. For a fair comparison, we reproduce some knowledge distillation (KD) methods including logit knowledge distillation (LKD) [9], Fitnets [23] and MBDL [28]. Given a ResNet-50 [6] as the teacher, JDP outperforms UMTS [10] by +1.70 % in mAP accuracy on VeRi-776 [13]. Given a ResNet-101 [6] as the teacher, JDP defeats MBDL [28] by a 2.22% larger mAP and a larger 1.43% mAP on MSMT17 [26] and VeRi-776 [13], respectively. Fitnets [23] uses the same distance loss function to JDP and extra dimension uniform units (i.e., $1 \times 1$ convolution) to minimizes the hidden layer features between students and the teacher. However, JDP still outperforms Fitnets [23] on three datasets, e.g., a 1.43 % larger in mAP and a 4.52 G FLOPs fewer computational cost on VeRi-776 [13].

Compared with VKD [19], JDP's mAP is about 2% lower than that of VKD [19] on VeRi-776 [13]. However, we believe this is acceptable, because VKD [19] is a self-distillation method, which focuses on improving the accuracy performance of the original model (i.e., teacher) itself but not educating a lightweight student. In contrast, our JDP aims to learn a lightweight student and to maintain accuracy performance. As a result, given the same

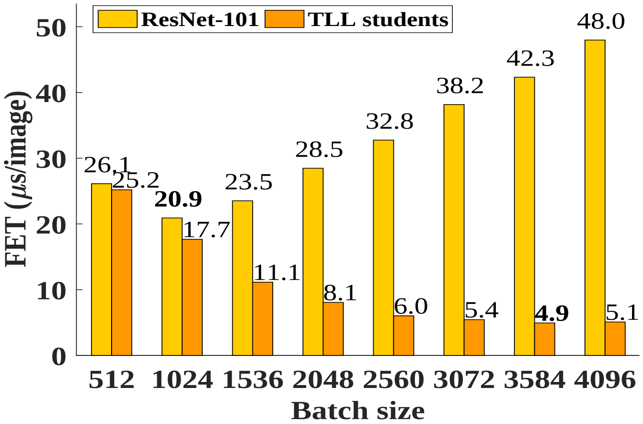

Figure 2: The FET per image results on the test subset of MSMT17 [26].

| Methods | Teacher | Student | PCP | LASSO | KL | DL | MP (M) | FLOPs (G) | mAP | R1 |
|---|---|---|---|---|---|---|---|---|---|---|
| Teacher | - | ResNet-101 | | | | | 43.50 | 12.99 | 78.68 | 95.65 |
| JDP without PCP | ResNet-101 | TLL | | ✓ | ✓ | ✓ | 7.19 | 2.21 | 73.75 | 93.89 |
| JDP without KD | - | TLL | ✓ | ✓ | | | 4.79 | 1.41 | 73.08 | 93.68 |
| JDP without DL | ResNet-101 | TLL | ✓ | ✓ | ✓ | | 5.18 | 1.52 | 76.00 | 94.82 |
| JDP | ResNet-101 | TLL | ✓ | ✓ | ✓ | ✓ | 8.70 | 2.79 | 78.51 | 94.99 |

Table 2: Ablation studies on VeRi-776 [13]. The teacher of methods is the teacher of JDP. KL and DL are KL loss and distance loss.

teacher (i.e., ResNet101), VKD [19] defeats the teacher but does not compress the teacher, and our JDP drops by 0.17% mAP but greatly compresses the teacher, i.e., we save 78.52% of FLPOs and 80.00% parameters on VeRi-776 [13].

The feature extraction time (FET) [33] per image is applied to evaluate the proposed JDP method's inference speed on one V100 32G GPU. The results are shown in Figure 2. We can see that our TTL student is consistently faster than the ResNet-101 teacher under different batch size settings. The ResNet-101 teacher has the best 20.9$\mu s$ FET performance when batch size is set to 1024, while our TTL student holds the best 4.9$\mu s$ FET performance when batch size is set to 3584, which illustrates our TTL student's inference time only 23.4% of the ResNet-101 teacher's inference time.

## 4.4 Ablation Experiments

We conduct ablation experiments to analyze the effectiveness of different components of JDP on VeRi-776 [13]. We gradually increase each component to JDP respectively including the PCP block, the group LASSO loss, the KL loss and the distance loss (denoted as DL). Ablation experimental results are shown in Table 2.

Firstly, from Table 2, compared to other ablation results, we can see that the performance of JDP without the PCP block is not competitive. For example, compared to JDP without DL, the computation cost of JDP without the PCP block wastes 28.67% model parameters and 31.22% FLOPs, but the MAP accuracy performance is reduced from 76.00 % to 73.75 %. It demonstrates that the PCP block is crucial to make a good balance of accuracy performance and computation cost.

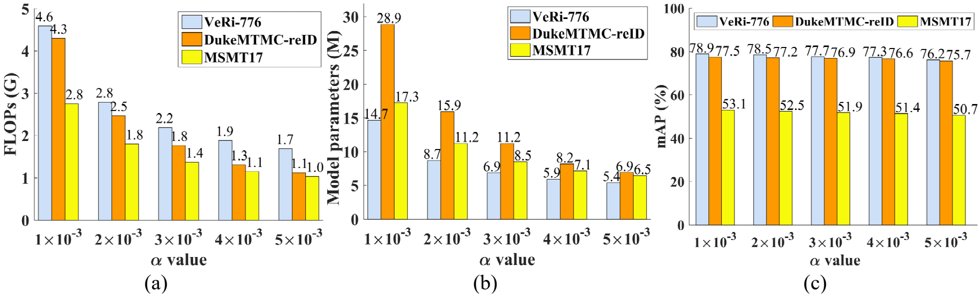

Figure 3: The influence of $\alpha$ value on the performance of JDP. (a) The influence of $\alpha$ value on the FLOPs performance. (b) The influence of $\alpha$ value on the model parameters performance. (c) The influence of $\alpha$ value on the mAP performance.

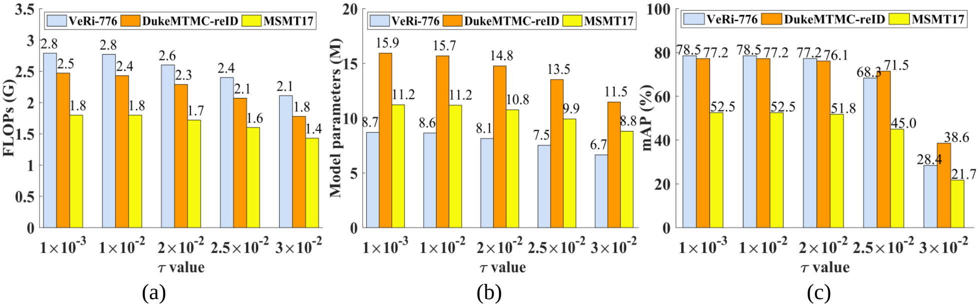

Figure 4: The influence of $\tau$ value on the performance of JDP. (a) The influence of $\tau$ value on the FLOPs performance. (b) The influence of $\tau$ value on the model parameters performance. (c) The influence of $\tau$ value on the mAP performance.

Secondly, comparing to JDP without KD, we can find that the KL loss is applied to JDP, causing the mAP accuracy performance to rise from 73.08 % to 76.00 %, but the computation cost is slightly increased. Furthermore, the DL loss is added to JDP. The mAP accuracy performance rises from 76.00 % to 78.51%. It demonstrates that KD can prevent the PCP block from losing many essential channel parameters and improving student performance.

Thirdly, comparing with the teacher, we acquire a teacher-like and light (TLL) student, which saves 80.00% model parameters and 78.52% FLOPs, while the mAP accuracy performance only drops by 0.17%. This result shows that our method can make a good balance of accuracy performance and computation cost.

## 4.5 Parameter Analysis

**The influence of LASSO loss weight $\alpha$.** There are key hyper-parameters $\alpha$ in Eq .6 in the proposed JDP. We test different values of $\alpha$ on three datasets to explore the impact of the $\alpha$ value on TLL student's performance. The experimental results are shown in Figure 3, where we can see that as the $\alpha$ value increasing, the accuracy performance of JDP decreases slightly, but the computational performance increases significantly. For example, when the $\alpha$ value increases from $1 \times 10^{-3}$ to $5 \times 10^{-3}$ on DukeMTMC-reID [22], the FLOPs performance of the TLL student gradually declines from 4.3 G to 1.1 G. At the same time, the mAP accuracy performance of the TLL student is just reduced by 1.8%. It demonstrates that the proposed JDP has good robustness to the $\alpha$ value and can accelerating the student under

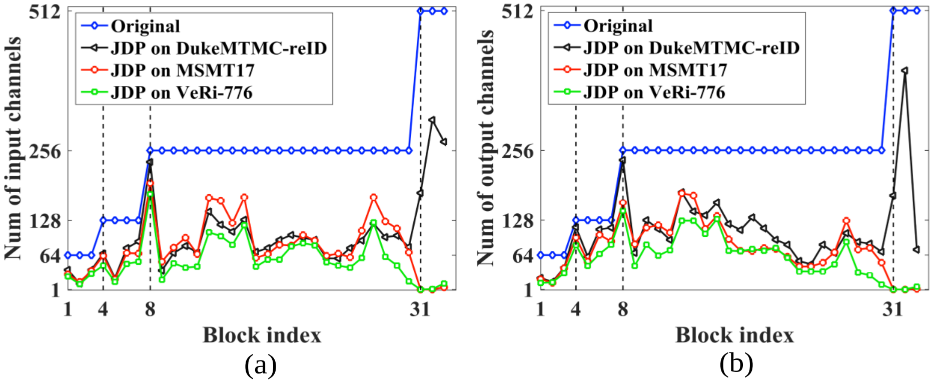

Figure 5: The finally number channel of the $K \times K$ convolution of student using JDP. (a) The number of input channel of the $K \times K$ convolution of student. (b) The number of output channel of $K \times K$ convolution of student.

the premise of preserving the accuracy performance of the student as much as possible.

**The influence of pruning threshold $\tau$.** The larger $\tau$, the more channels the more channels would be pruned, resulting in lower computational cost and more performance loss. From Figure 4, we can observe this situation. Especially, when $\tau$ surpasses $2.5 \times 10^{-2}$, mAP performance drops sharply on three datasets. For example, the JDP's mAP performance drops sharply by 39.9% on VeRi-776 [13] when the $\tau$ value increases from $2.5 \times 10^{-2}$ to $3 \times 10^{-2}$. As a result, $\tau$ is recommended to be less than $2.5 \times 10^{-2}$.

To understand how JDP pruning convolution channel, the final width of each $K \times K$ convolutional layer of the TLL student is shown in Figure 5. We can find that the deeper the convolutional layer, the more number of channels is reduced. For example, comparing with the input channel number of the 31-th block of the original network, the input channel number of the 31-th block of the TLL student saves 511 channel numbers on VeRi-776 [13]. Similarly, comparing with the output channel number of the 32-th block of the original network, the output channel number of the 32-th block of the TLL student saves 511 channel numbers on MSMT17 [26].

# 5 Conclusion

In this paper, we propose JDP method that acquires a teacher-like and light (TLL) student. Specifically, we consider to reduce the architectural differences between students and teachers to minimize the gap between students and teachers. Thus, we design a pruner-convolution-pruner (PCP) block, which is applied to replace all $K \times K$ convolutional layers of the student to acquire a teacher-like student with the same depth as the teacher but slightly modified the basic block in the student training phase. Meanwhile, the teacher-like student is jointly supervised by the KD loss and group LASSO loss functions. The KD loss function promotes the student to learn knowledge from the teacher to avoid unconstrained pruning. The group LASSO loss function enforces the pruners of the PCP block to realize the channel sparsity for filtering unimportant channels. In the student test phase, the pruner of the PCP block is pruned, and all PCP blocks will convert to a light $K \times K$ convolutional layer. At last, the teacher-like will convert into a TLL student. Extensive experiments show that our JDP method outperforms state-of-the-art methods in terms of accuracy and computations.

# Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under the Grants 61976098, 61871434, and 61802136, in part by the National Key R & D Program of China under the Grant 2021YFE0205400, in part by the Natural Science Foundation for Outstanding Young Scholars of Fujian Province under the Grant 2019J06017, in part by the Key Science and Technology Project of Xiamen City under the Grant 3502ZCQ20191005, in part by the Science and Technology Bureau of Quanzhou under the Grant 2018C115R. The authors also thank Huawei Technologies Co., Ltd. for providing the NAIE cloud computing service.

# References

[1] Gustavo Aguilar, Yuan Ling, Yu Zhang, Benjamin Yao, Xing Fan, and Chenlei Guo. Knowledge distillation from internal representations. In *AAAI Conference on Artificial Intelligence*, pages 7350–7357, 2020.

[2] Xiaohan Ding, Yuchen Guo, Guiguang Ding, and Jungong Han. Acnet: Strengthening the kernel skeletons for powerful cnn via asymmetric convolution blocks. In *International Conference on Computer Vision*, pages 1911–1920, 2019.

[3] Xiaohan Ding, Xiangyu Zhang, Jungong Han, and Guiguang Ding. Diverse branch block: Building a convolution as an inception-like unit. In *Conference on Computer Vision and Pattern Recognition*, pages 10886–10895, 2021.

[4] Xiaohan Ding, Xiangyu Zhang, Ningning Ma, Jungong Han, Guiguang Ding, and Jian Sun. Repvgg: Making vgg-style convnets great again. In *Conference on Computer Vision and Pattern Recognition*, pages 13733–13742, 2021.

[5] Song Han, Jeff Pool, John Tran, and William J Dally. Learning both weights and connections for efficient neural networks. In *Conference on Neural Information Processing Systems*, pages 1135–1143, 2015.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[7] Byeongho Heo, Jeesoo Kim, Sangdoo Yun, Hyojin Park, Nojun Kwak, and Jin Young Choi. A comprehensive overhaul of feature distillation. In *International Conference on Computer Vision*, pages 1921–1930, 2019.

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

[9] Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. In *Conference on Neural Information Processing Systems Workshops*, 2015.

[10] Xin Jin, Cuiling Lan, Wenjun Zeng, and Zhibo Chen. Uncertainty-aware multi-shot knowledge distillation for image-based object re-identification. In *AAAI Conference on Artificial Intelligence*, pages 11165–11172, 2020.

[11] Pirazh Khorramshahi, Amit Kumar, Neehar Peri, Sai Saketh Rambhatla, Jun-Cheng Chen, and Rama Chellappa. A dual-path model with adaptive attention for vehicle re-identification. In *International Conference on Computer Vision*, pages 6132–6141, 2019.

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Conference on Neural Information Processing Systems*, pages 1097–1105, 2012.

[13] Xinchen Liu, Wu Liu, Huadong Ma, and Huiyuan Fu. Large-scale vehicle re-identification in urban surveillance videos. In *IEEE International Conference on Multimedia & Expo*, pages 1–6, 2016.

[14] Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. Learning efficient convolutional networks through network slimming. In *International Conference on Computer Vision*, pages 2736–2744, 2017.

[15] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *International Conference on Learning Representations*, 2017.

[16] Hao Luo, Wei Jiang, Youzhi Gu, Fuxu Liu, Xingyu Liao, Shenqi Lai, and Jianyang Gu. A strong baseline and batch normalization neck for deep person re-identification. *IEEE Transactions on Multimedia*, 20(10):2597–2609, 2020.

[17] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant. In *AAAI Conference on Artificial Intelligence*, pages 5191–5198, 2020.

[18] Baoyun Peng, Xiao Jin, Jiaheng Liu, Dongsheng Li, Yichao Wu, Yu Liu, Shunfeng Zhou, and Zhaoning Zhang. Correlation congruence for knowledge distillation. In *International Conference on Computer Vision*, pages 5007–5016, 2019.

[19] Angelo Porrello, Luca Bergamini, and Simone Calderara. Robust re-identification by multiple views knowledge distillation. In *European Conference on Computer Vision*, pages 93–110, 2020.

[20] Nan Pu, Wei Chen, Yu Liu, Erwin M. Bakker, and Michael S. Lew. Lifelong person re-identification via adaptive knowledge accumulation. In *Conference on Computer Vision and Pattern Recognition*, pages 7901–7910, 2021.

[21] Pengyuan Ren and Jianmin Li. Factorized distillation: Training holistic person re-identification model by distilling an ensemble of partial reid models. *arXiv preprint arXiv:1811.08073*, 2018.

[22] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *European Conference on Computer Vision*, pages 17–35, 2016.

[23] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In *International Conference on Learning Representations*, 2015.

[24] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016.

[25] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. In *International Conference on Learning Representations*, 2020.

[26] Longhui Wei, Shiliang Zhang, Wen Gao, and Qi Tian. Person transfer gan to bridge domain gap for person re-identification. In *International Conference on Computer Vision*, pages 79–88, 2018.

[27] Wei Wen, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. Learning structured sparsity in deep neural networks. In *Conference on Neural Information Processing Systems*, pages 2074–2082, 2016.

[28] Yi Xie, Jianqing Zhu, Huanqiang Zeng, Canhui Cai, and Lixin Zheng. Learning matching behavior differences for compressing vehicle re-identification models. In *International Conference on Visual Communications and Image Processing*, pages 523–526, 2020.

[29] Yi Xie, Fei Shen, Jianqing Zhu, and Huanqiang Zeng. Viewpoint robust knowledge distillation for accelerating vehicle re-identification. *EURASIP Journal on Advances in Signal Processing*, 2021(1):1–13, 2021.

[30] Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

[31] Z. Zheng, T. Ruan, Y. Wei, Y. Yang, and T. Mei. Vehiclenet: Learning robust visual representation for vehicle re-identification. *IEEE Transactions on Multimedia*, pages 2683–2693, 2021.

[32] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *AAAI Conference on Artificial Intelligence*, pages 13001–13008, 2020.

[33] Jianqing Zhu, Huanqiang Zeng, Shengcai Liao, Zhen Lei, Canhui Cai, and LiXin Zheng. Deep hybrid similarity learning for person re-identification. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(11):3183–3193, 2018.

[34] Jianqing Zhu, Jingchang Huang, Huanqiang Zeng, Xiaoqing Ye, Baoqing Li, Zhen Lei, and Lixin Zheng. Object reidentification via joint quadruple decorrelation directional deep networks in smart transportation. *IEEE Internet of Things Journal*, pages 2944–2954, 2020.

[35] Michael Zhu and Suyog Gupta. To prune, or not to prune: exploring the efficacy of pruning for model compression. *International Conference on Learning Representations*, 2018.