Towards a Smaller Student: Capacity Dynamic Distillation for Efficient Image Retrieval

Yi Xie[1] Huaidong Zhang[1*] Xuemiao Xu[1,4,5,6*] Jianqing Zhu[2] Shengfeng He[3]
[1]South China University of Technology [2]Huaqiao University [3]Singapore Management University
[4]State Key Laboratory of Subtropical Building Science
[5]Ministry of Education Key Laboratory of Big Data and Intelligent Robot
[6]Guangdong Provincial Key Lab of Computational Intelligence and Cyberspace Information
ftyixie@mail.scut.edu.cn, {huaidongz, xuemx}@scut.edu.cn
jqzhu@hqu.edu.cn, shengfenghe@smu.edu.sg

Abstract

Previous Knowledge Distillation based efficient image retrieval methods employ a lightweight network as the student model for fast inference. However, the lightweight student model lacks adequate representation capacity for effective knowledge imitation during the most critical early training period, causing final performance degeneration. To tackle this issue, we propose a Capacity Dynamic Distillation framework, which constructs a student model with editable representation capacity. Specifically, the employed student model is initially a heavy model to fruitfully learn distilled knowledge in the early training epochs, and the student model is gradually compressed during the training. To dynamically adjust the model capacity, our dynamic framework inserts a learnable convolutional layer within each residual block in the student model as the channel importance indicator. The indicator is optimized simultaneously by the image retrieval loss and the compression loss, and a retrieval-guided gradient resetting mechanism is proposed to release the gradient conflict. Extensive experiments show that our method has superior inference speed and accuracy, e.g., on the VeRi-776 dataset, given the ResNet101 as a teacher, our method saves 67.13% model parameters and 65.67% FLOPs without sacrificing accuracy. Code is available at https://github.com/SCY-X/Capacity_Dynamic_Distillation.

1. Introduction

Image retrieval [2] aims to rank all the instances in a retrieval set based on their relevance to the query image. However, many image retrieval methods [45, 50] use heavy networks to acquire a high accuracy, causing a low inference speed and hindering practical applications. As an efficient network compression technology [21, 52, 58], knowledge distillation (KD) [42, 43, 57, 59] has been widely validated to be useful for boosting the performance of the lightweight student model by transferring knowledge from a heavy teacher model, which is also applied to accelerate image retrieval, as done in [40, 54, 56].

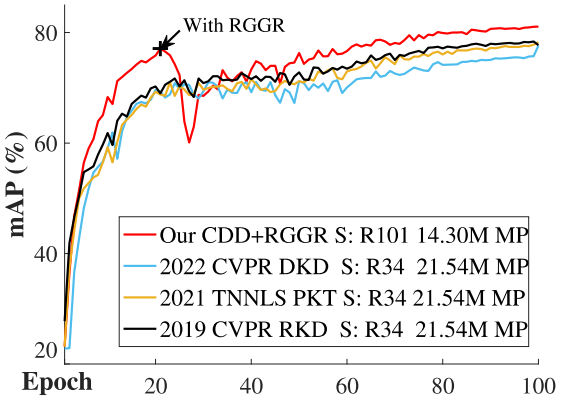

Figure 1. The mAP (%) of KD methods in per epoch on VeRi776 [30]. R101 and R34 denote the ResNet101 and ResNet34 students, respectively. Our CDD+RGGR outperforms other KD methods because the heavy student has more talented in effectively learning distilled knowledge than a light student in the early epochs.

Previous KD-based image retrieval methods [40, 47] assign a lightweight network as the student model to acquire a fast inference speed. However, we have an experimental observation that a lightweight student model is less talented in effectively learning distilled knowledge than a heavy student model in the early epochs, leading to final performance degeneration. As shown in Fig. 1, a heavy ResNet101 (i.e., our) improves its performance faster than the light ones (ResNet34, other methods) in the early epochs (first 20 epochs) of KD optimization. This finding is consistent with previous KD works [6, 29, 37], which have shown that a ca-

*Corresponding authors

pacity mismatch between teachers and students can hinder light students from acquiring knowledge from large teachers [6]. This is similar to a kid only comprehending a small portion of taught knowledge [3]. In addition, many studies [1, 5, 14] have shown that the quality of the initial training period is crucial in determining the final optimization solution. Thus, the light student needs to effectively learn from large teachers during the critical early training stages.

Motivated by the above finding, we propose a new KD framework, **C**apacity **D**ynamic **D**istillation (CDD), to allow dynamic model compression during KD learning. Unlike existing KD-based image retrieval methods that use a lightweight student model, CDD employs a heavy initial network as the student model. Thus the student has a high representation capacity to comprehensively understand knowledge from teachers in the early KD iterations. To acquire a fast inference speed, we design a **D**istillation **G**uided **C**ompactor (DGC) module inserted after the heavy convolutional layer of the student as the channel importance indicator of each convolutional layer. Then, we implement a parametric sparse loss on DGC during KD learning to find the unimportant channel of the heavy convolutional layer, thus gradually reducing the capacity of the student network. The overall training process can be done end-to-end in one KD training period. After training, the sparse DGC will be pruned to a slim DGC, and the slim DGC and previous heavy convolutional layer can convert to a slim convolutional layer. Therefore, the heavy student model will be converted to a lightweight model.

To dynamically edit the student model capacity, DGC is optimized simultaneously by the image retrieval loss and the parametric sparse loss, resulting in a gradient conflict between knowledge accumulation and forgetting. To release the gradient conflict, we propose a retrieval-guided gradient resetting mechanism (RGGR), which introduces a binary mask to zero out the knowledge accumulation gradient. Specifically, we first use the train data to simulate the retrieval result. Then, RGGR selects channels with little influence on simulation retrieval results and zeros the knowledge accumulation gradient, achieving a high prunability. As demonstrated in Fig. 1, when we activate RGGR (at the 21st epoch), the heavy student model focuses more on compression and suffers transient performance degradation. But, thanks to the well-trained KD optimization of students in the early epochs, the student model finally achieves a good balance between accuracy and inference speed.

The main contributions of this paper are listed as follows:

(1) We propose a capacity dynamic distillation framework (CDD) to effectively learn distilled knowledge in the early training epochs.

(2) We propose a retrieval-guided gradient resetting mechanism (RGGR) to release the gradient conflict between the image retrieval loss and the parametric sparse loss.

(3) Extensive experiments demonstrate that our method is superior to state-of-the-art approaches in terms of inference and computations, by a large margin of 24.13%MP and 21.94% FLOPs.

2. Related Works

Network Pruning. Network Pruning (NP) [7, 31, 52] aims to obtain a light network by removing unimportant parts from a well-trained yet heavy network. Recent NP works primarily focus on structured pruning [31, 52], which applies sparsity functions to the convolutional layers of well-trained large models to filter out unimportant channels. However, NP often leads to a reduction in accuracy due to its irreversible sparsification process, which can cause significant damage to the network [7].

Knowledge Distillation. Most KD-based image retrieval works [40,53,56] employ a lightweight model as the student and transfers the knowledge among samples from teachers to guide the student optimization. Although these works have helped light students improve their accuracy, the weak representational capacity hinders further accuracy improvement of students. Therefore, recent KD works [3, 35] have paid attention to this situation. However, they only focused on how to transfer knowledge well to facilitate students' understanding of teachers' knowledge without considering enhancing students' representational capacity. Although self-distillation methods [24, 26, 41] assign a large student the same size as the teacher to better understand the teacher's knowledge, they suffer from large students' high computational cost in the inference phase. To address this issue, a natural solution is to aggregate NP with KD in a two-stage design, i.e., self-KD + NP. Specifically, the two-stage methods first assign a large student to effectively distilled knowledge from teachers, and then the well-trained large student network is pruned to a slim network. However, the two-stage approach may suffer a significant performance loss during the pruning stage since it still faces the challenge of sparsification process deviation. In this paper, we present an end-to-end KD method integrated with NP, which exploits the proposed distillation guided compactor (DGC) module to effectively alleviate the conflict between KD and NP.

Re-parameterization. The re-parameterization (Rep) methods [8, 10, 11] construct a sequence of multiple convolutional layers to replace a single convolutional layer of an original network to enhance the feature learning ability during training. Then, those sequences are simplified to a convolutional layer to avoid extra computation consumption during inference. Recently, Ding et al. [9] proposed a gradient resetting and compactor re-parameterization (ResRep) method, which is the first attempt to apply Rep to NP. Motivated by this, we explore the gradient resetting technique in knowledge distillation and propose zeroing out the selected channel's gradients according to feature retrieval results.

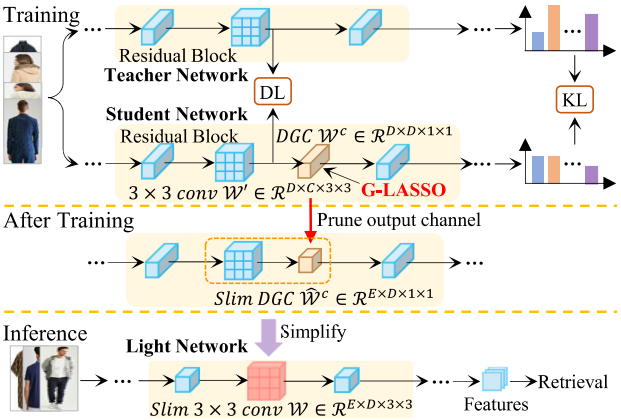

Figure 2. The capacity dynamic distillation framework. The high-capacity student can comprehensively understand teachers' knowledge in the early KD iterations. After training, the student can be converted into a light network to acquire a fast inference speed.

3. Method

This section presents our CDD framework for efficient image retrieval, which uses a new distillation pipeline for a dynamic representational capacity student model. Unlike previous KD methods [3, 59] training a student model with fixed representation capacity, our distillation strategy employs a heavy student network to learn transferred knowledge effectively from teachers while gradually compressing the student's capacity for obtaining a final model with fast inference speed. The DGC module is integrated into our CCD framework to achieve dynamic representation compression during KD learning, as described in Section 3.1.

Furthermore, we propose the RGGR mechanism to compress the student model by releasing the conflict between retrieval-related losses and compression loss. RGGR first evaluates the channel importance of students with respect to image retrieval. Then, the gradients derived from image retrieval-related losses are zeroed to eliminate the conflict in the final objective. This approach ensures that the selected channels' weight values are solely optimized by the compression loss, enabling more efficient convergence to zero. RGGR method will be discussed in detail in Section 3.2.

3.1. Dynamic Representation Distillation

From Fig. 2, our CDD framework is designed to transfer knowledge from a well-trained teacher model $P_t(y|x,\theta_t)$ to an untrained student model $P_s(y|x,\theta_s)$, where x is the model input and θ denotes the model parameters. Different from previous KD methods employing models following the rule $|\theta_s| < |\theta_t|$ for model compression, CDD employs the student and the teacher satisfying $|\theta_s| \geq |\theta_t|$ for effective early learning, as discussed in Fig. 1.

3.1.1 Dynamic Representation Learning

Distillation Learning. From Fig. 2, we employ ResNet [17] with cascaded residual blocks as an example. During training, we consider globally and locally consistent distillation to transfer knowledge from teachers to students.

For global-consistent distillation, we use the Kullback-Leibler divergence (KL) loss L_{kl} [21] to transfer logit knowledge, as done in previous KD-based image retrieval works [40, 41]. For local-consistent distillation, different from layer-wise alignment adopted in previous works [19, 27], thanks to the large capacity student has the same network depth as teachers, we perform a block-wise alignment by focusing on the 3×3 convolutional layer of each residual block to more completely transfers knowledge as follows:

$$L_{dl}(f^t, f^s) = \frac{1}{M}\sum_{i=1}^{M} \mathrm{DL}\big(\mathrm{GAP}(f^t), \mathrm{GAP}(f^s)\big), \quad (1)$$

where f^t is the feature map output by the 3×3 convolutional layer of teachers and f^s is the corresponding feature map from students; M is the number of residual blocks, such as ResNet101 contains 33 residual blocks; $\mathrm{DL}(\cdot,\cdot)$ denotes Euclidean distance loss; The $\mathrm{GAP}(\cdot)$ denotes the global average pool layer to compress feature maps spatially.

Distillation Guided Compactor (DGC). To compress the student model dynamically during the training period, we add a proposed DGC module after each 3×3 convolutional layer of students. Specifically, the DGC module here is a 1×1 convolutional layer without bias being initialized as an identity matrix, which is used to evaluate the importance of each channel of the 3×3 convolutional layer. In this way, we can sparse the 1×1 convolutional layer during the training period. After training, the 3×3 convolutional layer can be pruned by merging these two layers. The merging operation is introduced in Section 3.1.2.

To sparse the matrix, we implement G-LASSO loss L_{np} [9, 23] on DGC to gradually decrease the student network's representational capacity as follows:

$$L_{np}(\mathcal{W}^c) = \sum_{i=1}^{D}\sqrt{\sum_{j=1}^{D}(\mathcal{W}^c_{i,j,1,1})^2}, \quad (2)$$

where $\mathcal{W}^c \in \mathbb{R}^{D\times D\times 1\times 1}$ represents kernel parameters of DGC modules. The first and second dimensions of $\mathcal{W}^c$ represent the output and input channels, respectively.

Finally, the student's overall loss L_{all} includes L_{dl} presented in Eq. 1, KL loss L_{kl}, two widely used image retrieval losses and L_{np} presented in Eq. 2 as follows:

$$L_{all} = \frac{1}{2}L_{dl} + L_{id} + L_{tri} + L_{kl} + \alpha L_{np}, \quad (3)$$

where L_{id} is the label smooth regularization cross-entropy [46], as done in [54]; L_{tri} is the triplet loss using hard sample mining strategy [20]; α is a hyper-parameter to control the sparsity level, with a default value of 0.004.

Discussion. Intuitively, we argue that the DGC design is better than the previous two-stage "self-KD+NP" methods

because the sparse process can't severely deviate under the supervision of the teacher network. Furthermore, the DGC design outperforms the previous end-to-end "KD with NP" methods because DGC can alleviate the conflict between KD and NP by decoupling their action positions.

3.1.2 Efficient Inference with Pruning

After training, the output channel of DGC modules is treated as an unimportant channel and discarded if the output channel weight is less than the pruning threshold λ. The default value of λ is 1×10^{-5}, as done in [9, 12]. Thus, the DGC module is trimmed to a slim one $\hat{\mathcal{W}}^c \in \mathbb{R}^{E \times D \times 1 \times 1}, E \leq D$. Then, the 3×3 convolutional layer and the slim DGC module in one residual block can be simplified to being a slim 3×3 convolutional layer as follows:

$$\mathcal{W} = T\big(T(\mathcal{W}') \circledast \hat{\mathcal{W}}^c\big), \quad B = B' \circledast \hat{\mathcal{W}}^c, \tag{4}$$

where $\mathcal{W} \in \mathbb{R}^{E \times C \times 3 \times 3}$ and $B \in \mathbb{R}^E$ represent the slim 3×3 convolutional layer's kernel parameters and its bias, respectively. E and C represent the output and input channel sizes of the slim 3×3 convolutional layer. $T(\cdot)$ and $\circledast$ denote the transpose function and the convolution operator, respectively. $\mathcal{W}' \in \mathbb{R}^{D \times C \times 3 \times 3}$ and $B' \in \mathbb{R}^D$ represent the 3×3 convolutional layer's kernel parameters and its bias.

Finally, the 3×3 convolutional layer and the $\hat{\mathcal{W}}^c$ in each residual block are merged into a slim 3×3 convolutional layer. Besides, the last bottleneck layer (i.e., 1×1 convolutional layer) in each residual block also is thinned by discarding those unimportant channels according to the channel value in $\hat{\mathcal{W}}^c$. As a result, the heavy student can be transformed into a lightweight network to acquire a fast inference speed for image retrieval.

3.2. Retrieval-Guided Gradient Resetting

Although CCD has achieved early learning with large representation capacity and efficient inference by dynamic compression, the optimization conflict remains unsolved. Specifically, the gradient conflict between the knowledge accumulation gradient from image retrieval loss $L_{acc} = \frac{1}{2}L_{dl} + L_{id} + L_{tri} + L_{kl}$ and the knowledge forgetting gradient from pruning loss L_{np} may cause a low compression rate. More specifically, most of the output channels' weight of DGC may be updated to be approximated to 0 via L_{np}, but never close enough for perfect pruning due to L_{acc} [9].

To this end, we propose the Retrieval-Guided Gradient Resetting mechanism (RGGR) to release this gradient conflict, as shown in Fig. 3. Specifically, RGGR employs a D-dimensional binary mask $M \in \{0, 1\}$ to zero the knowledge accumulation gradient of some DGC's output channels to acquire a resetting gradient $\hat{G}$ as follows:

$$\hat{G}(\mathcal{W}_i^c) = \frac{\partial L_{acc}(X, Y, \theta_s)}{\partial \mathcal{W}_i^c} M_i + \alpha \frac{\partial L_{np}(X, Y, \theta_s)}{\partial \mathcal{W}_i^c}, \tag{5}$$

where $\mathcal{W}_i^c = \mathcal{W}_{i,:,:,:}^c$ denotes the $\mathcal{W}^c$ matrix's i-th output channel; X and Y are data examples and labels respectively.

From Eq. (5), with $M_i = 0$, the i-th output channel's knowledge accumulation gradient of DGC is reset to zero, causing the first term to be ignored. Thus, $M_i = 0$ can make DGC's i-th output channel weight value steadily move towards 0 to achieve high compression. Intuitively, once the knowledge accumulation gradient of important output channels is zeroed, it causes a decrease in the retrieval performance of students. Therefore, to maintain the student's retrieval performance, RGGR will zero out the knowledge learning gradient of the output channel that has the most negligible impact on the retrieval results as follows.

Retrieval Rank Matrix R Formulation. To evaluate the channel importance of each residual block with respect to image retrieval, we build a query set Q and a gallery set G on each block during training to simulate the retrieval results. Different from using only one batch of teacher features as the gallery set, we construct a large gallery set G to reveal the overall distribution of the data X adequately. Specifically, given a batch size of N samples, we extract the teacher feature $\mathcal{F}_t = [f_1^t, f_2^t, ..., f_N^t] \in \mathbb{R}^{N \times D}$ from the 3×3 convolutional layer and GAP layers to update the gallery G. The gallery set is implemented in a queue with a fixed size L, which means the first-in-first-out policy is followed to maintain the queue length, as done in [16, 48].

For building the query set Q, we should extract the student feature $\mathcal{F}_s = [f_1^s, f_2^s, ..., f_N^s] \in \mathbb{R}^{N \times D}$ from DGC and GAP layers as the query set Q. However, $\mathcal{F}_s$ is not invariant and semantic enough for presenting the image information during early training, causing the matching results between $\mathcal{F}_s$ and $\mathcal{F}_t$ to be unreliable. Therefore, we use $\mathcal{F}_t$ instead of $\mathcal{F}_s$ as the query set to retrieve the gallery set G. Formally, we obtain the retrieval rank matrix $R \in \mathbb{R}^{N \times L}$ as follows:

$$R = \{r_j^i \in \mathbb{R}^{1 \times D} \big| 1 \leq i \leq N, 1 \leq j \leq L\}, \tag{6}$$

where r_j^i represents the i-th query feature retrieved from the gallery set G and then returning the j-th gallery feature.

Binary Mask M Formulation based on R. Since the retrieval rank matrix R is obtained by sorting the retrieval distance matrix, we conclude that R can indicate the output channels with the most negligible impact on the retrieval result. In addition, to reduce the negative impact of gallery features with low relevance, we select top-K retrieval gallery feature from R as the retrieval result $R_K \in \mathbb{R}^{N \times K}$ to find unimportant output channel as follows:

$$R_K = \{r_j^i \in \mathbb{R}^{1 \times D} \big| 1 \leq i \leq N, 1 \leq j \leq K\}, \tag{7}$$

where $K = 2$ empirically.

Thus, the unimportant output channel index I between the query set Q and the retrieval result R_K is as follows:

$$I = \bigcap_{i=1}^{N} \bigcap_{j=1}^{K} I_{ij}, \tag{8}$$

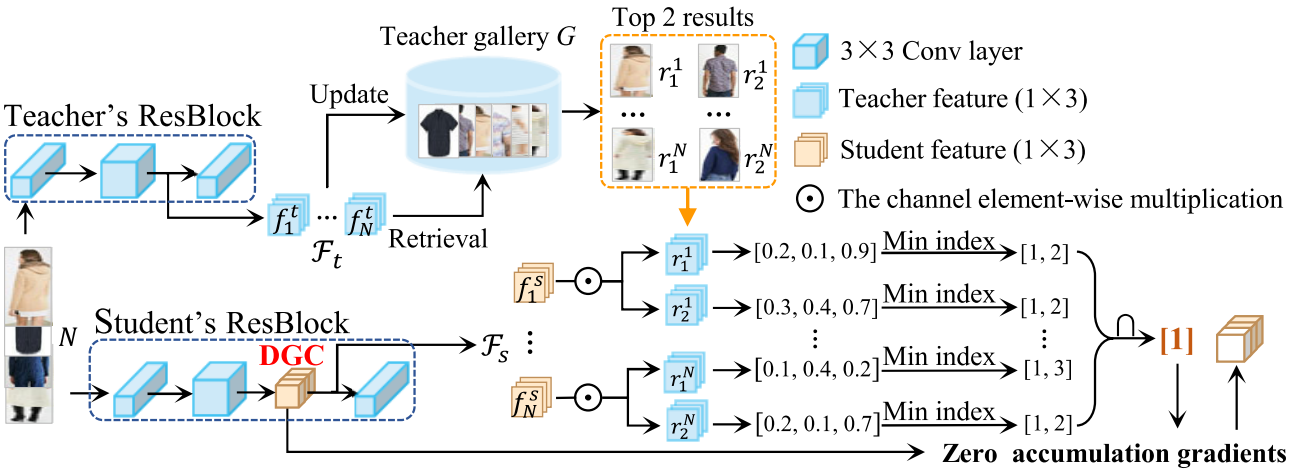

Figure 3. The retrieval-guided gradient resetting mechanism (RGGR). For ease of visualization, we assume that the DGC's output channel is 3. RGGR believes that the DGC's first channel has the most negligible influence on retrieval results by the absolute value of the channel product between f_s^i and r_j^i. Then, RGGR resets the accumulation gradient of the first channel to 0 to push its weight value toward 0.

where I_{ij} denotes the unimportant output channel index between the i-th query feature f_i^s and r_j^i as follows:

$$I_{ij} = \omega\big(A(f_i^s, r_j^i)\big), 1 \le i \le N, 1 \le j \le K, \quad (9)$$

where $\omega(\cdot)$ represents a series of operations, including sorting the importance of the output channels in ascending order, picking the smallest one from the ordered index queue, storing its index, and stopping picking when $p = 50\%$ ratios of the index have been selected. The value of p is chosen empirically. The function $A(\cdot)$ that evaluates the importance of each output channel of the DGC modules, based on the absolute value of the product of f_s^i and r_j^i as follows:

$$A(f_i^s, r_j^i) = [A_d], A_d = \left|f_{i,d}^s \cdot r_{j,d}^i\right|, 1 \le d \le D, \quad (10)$$

where $f_{i,d}^s$ and $r_{j,d}^i$ represent the d-th channel value of f_s^i and the d-th channel value of r_j^i, respectively.

Overall, based on the unimportant output channel index I, we update the binary mask M to zero the knowledge accumulation gradient of the unimportant output channel of DGC as follows:

$$M_i = \begin{cases} 0, & \text{if } i \in I, \\ 1, & \text{otherwise,} \end{cases} \quad 1 \le i \le D. \quad (11)$$

4. Experiments

To demonstrate the superiority of our CDD+RGGR method, we conduct evaluations on three widely used public image retrieval datasets: In-Shop [32], VeRi776 [30], and MSMT17 [51]. In what follows, we briefly introduce datasets and performance metrics. Then, we conduct ablation experiments and compare CDD+RGGR with state-of-the-art methods. we conduct analysis experiments to examine the role of CDD+RGGR comprehensively.

4.1. Datasets and Performance Metric

In-Shop Clothes Retrieval (In-Shop) [32] is a commonly used clothes retrieval database, which contains 72,712 images of clothing items belonging to 7,986 categories. The training set includes 3,997 classes with 25,882 images. The query set with 14,218 images of 3,985 classes. The gallery set has 3,985 classes with 12,612 images.

VeRi776 [30] is a vehicle retrieval dataset. The training set contains 37,746 images of 576 subjects. The query set of 1,678 images of 200 subjects. The gallery set of 11,579 images of the same 200 subjects.

MSMT17 [51] is the largest pedestrian retrieval database, which contains 126,441 images of 4,101 pedestrian identities. The training set includes 32,621 training images of 1,041 identities. The test set includes 11,659 query images and 82,161 gallery images of 3,060 identities.

The **Cosine** distance is used as the retrieval algorithm, where the more similar the gallery image, the higher its ranking. The retrieval accuracy is evaluated using mean average precision (mAP) [44] and rank-1 identification rate (R1) [30]. Model size and computational complexity are measured using the number of model parameters (MP) and floating-point operations (FLOPs), respectivel.

4.2. Implementation Details

The software tools used for our experiments are Pytorch 1.12 [39], CUDA 11.6, and python 3.9. The hardware device is one GeForce RTX 3090Ti GPU. The network training configurations are as follows. (1) We use ImageNet pretrained ResNet101 [17] as backbones and set the last stride of ResNet101 to 1, as done in [54, 56]. (2) The teacher network is frozen during students' training. (3) The data augmentation includes z-score normalization, random cropping, random erasing [60], and random horizontal flip oper-

Table 1. Ablation results on In-Shop [32] and VeRi776 [30].

METHOD	TEACHER	STUDENT	In-Shop				VeRi776			
			MP (M)	FLOPs (G)	mAP(%)	R1(%)	MP (M)	FLOPs (G)	mAP(%)	R1(%)
Teacher	-	ResNet101	43.50	12.99	**81.72**	**95.23**	43.50	12.99	80.50	96.42
CDD w/o DGC	ResNet101	ResNet101	35.89	10.94	71.87	90.45	39.80	11.96	62.14	93.27
CDD	ResNet101	ResNet101	20.14	6.31	81.38	95.01	18.59	5.77	80.36	96.48
CDD+RGGR	ResNet101	ResNet101	**14.97**	**4.68**	81.28	95.14	**14.30**	**4.46**	**80.67**	**96.66**

ations, as done in [54]. (4) The mini-batch stochastic gradient descent method [28] is used as the optimizer. The mini-batch size is set to 96, including 16 identities, and each identity holds 6 images. (5) Setting weight decays as 5×10^{-4} and momentums as 0.9. (6) The cosine annealing strategy [33] and linearly warmed strategy [15] are applied to adjust learning rates. (7) The learning rates are initialized to 1×10^{-3}, then linearly warmed up to 1×10^{-2} in the first 10 epochs, the drop point for learning rates is the 40-th epoch. For MSMT17 [51], the total training epoch is 120. For In-Shop [32] and VeRi776 [30], the total training epoch is 100. (8) Considering that pedestrians datasets and other datasets have different aspect ratios, for MSMT17 [51], the image resolution is set as 320×160. For In-Shop [32] and VeRi776 [30], the image resolution is set as 256×256.

4.3. Ablation Experiments

As shown in Table 1, we conduct ablation experiments on In-Shop [32] and VeRi776 [30]. Teacher means that we directly use the ResNet101 teacher to evaluate performance. CDD w/o DGC means that the student discards DGC module, causing KD of Eq. 1 and NP of Eq. 2 to act on the same 3×3 convolutional layers.

First, compared to Teacher, we can find that CDD w/o DGC slightly wins MP and FLOPs but severely degrades R1 and mAP. Specifically, CDD w/o DGC is inferior to Teacher by 18.36% mAP and 3.15% R1 on VeRi776 [30]. These results show that playing KD and NP on the same convolutional layers can compress networks but is injured for retrieval performance. The fault lies in the potential conflict between KD and NP because KD guides students to inherit more teachers' knowledge, but NP encourages students to forget knowledge. Furthermore, with the usage of DGC, CDD outperforms CDD w/o DGC in terms of all metrics. For example, on VeRi776 [30], CDD outperforms CDD w/o DGC by 18.22% mAP, 3.21% R1, 21.21M MP, and 6.19G FLOPs. These comparisons demonstrate that DGC can alleviate the conflict between KD and NP.

Second, with using RGGR, the student further improved inference performance without loss of accuracy performance. Compared with CDD, CDD+RGGR saves 25.67% MP and 25.80% FLOPs on In-Shop [32], and 23.08% MP and 22.70% FLOPs on VeRi776 [30]. The comparison demonstrates that RGGR can further boost the inference performance of students without sacrificing accuracy.

Table 2. Performance comparison on In-Shop.

METHOD [1]	TEACHER	STUDENT	MP	FLOPs	mAP	R1
Fastretri [18]	-	ResNet50	25.54	8.13	-	91.97
CCKD [40] ∗	ResNet101	ResNet34	21.54	7.31	80.65	94.78
FT [25] ∗	ResNet101	ResNet34	21.54	7.31	80.51	94.55
SP [49] ∗	ResNet101	ResNet34	21.54	7.31	80.81	94.89
AT [27] ∗	ResNet101	ResNet34	21.54	7.31	80.85	94.91
RKD [36] ∗	ResNet101	ResNet34	21.54	7.31	81.03	94.81
PKT [38] ∗	ResNet101	ResNet34	21.54	7.31	80.59	94.68
MBDL [56] ∗	ResNet101	ResNet34	21.54	7.31	80.76	94.71
DKD [59] ∗	ResNet101	ResNet34	21.54	7.31	80.93	94.86
CSKD [53] ∗	ResNet101	ResNet34	21.54	7.31	80.65	94.77
KDPE [19] ∗	ResNet101	ResNet34	21.54	7.31	78.99	94.08
ResRep [9]∗	ResNet101	ResNet101	16.27	5.11	79.86	94.68
CDD+RGGR	ResNet101	ResNet101	**14.97**	**4.68**	**81.28**	**95.14**

[1] The ∗ represents the result is re-implemented.

Table 3. Performance comparison on VeRi776.

METHOD [1]	TEACHER	STUDENT	MP	FLOPs	mAP	R1
PVEN [34]	-	ResNet50	25.54	8.13	79.50	95.60
ViT [13] ∗		ViT [13]	86.7	55.6	78.92	95.84
CCKD [40]∗	ResNet101	ResNet34	21.54	7.31	76.34	95.23
FT [25] ∗	ResNet101	ResNet34	21.54	7.31	77.18	95.77
AT [27] ∗	ResNet101	ResNet34	21.54	7.31	77.98	95.41
SP [49] ∗	ResNet101	ResNet34	21.54	7.31	77.50	95.77
RKD [36] ∗	ResNet101	ResNet34	21.54	7.31	78.56	95.65
PKT [38] ∗	ResNet101	ResNet34	21.54	7.31	78.45	95.47
MBDL [56] ∗	ResNet101	ResNet34	21.54	7.31	77.91	95.53
DKD [59] ∗	ResNet101	ResNet34	21.54	7.31	76.40	95.59
CSKD [53] ∗	ResNet101	ResNet34	21.54	7.31	73.60	94.34
KDPE [19] ∗	ResNet101	ResNet34	21.54	7.31	74.27	94.93
ResRep [9] ∗	ResNet101	ResNet101	15.87	4.87	77.60	95.11
CDD+RGGR	ResNet101	ResNet101	**14.30**	**4.46**	**80.67**	**96.66**
UMTS [24]	ResNet50	ResNet50	25.54	8.13	75.9	95.8
VKD [41]	ResNet50	ResNet50	25.54	8.13	**79.17**	95.23
ResRep [9] ∗	ResNet50	ResNet50	11.24	3.83	75.74	94.66
CDD+RGGR	ResNet50	ResNet50	**9.33**	**3.20**	78.75	**95.95**

[1] The ∗ represents the result is re-implemented.

4.4. Comparison with State-of-the-art Methods

Table 2, 3 and 4 summarize comparisons on In-Shop [32], VeRi-776 [30], and MSMT17 [51] datasets, respectively. Here, both MP (M), FLOPs (G), mAP (%), R1 (%) are obtained by only using the student during inference. For a fair comparison, Hinton's original Kullback-Leibler divergence strategy [21] is applied to all compared re-implemented KD methods, as done in previous works

Table 4. Performance comparison on MSMT17.

METHOD [1]	TEACHER	STUDENT	MP	FLOPs	mAP	R1
IANet [22]	-	ResNet50	25.54	4.07	46.8	75.5
BINet [4]	-	ResNet50	25.54	4.07	52.8	76.1
CCKD [40]∗	ResNet101	ResNet34	21.54	5.71	56.65	80.58
FT [25] ∗	ResNet101	ResNet34	21.54	5.71	56.67	80.38
AT [27] ∗	ResNet101	ResNet34	21.54	5.71	58.70	81.62
SP [49] ∗	ResNet101	ResNet34	21.54	5.71	57.32	80.37
RKD [36] ∗	ResNet101	ResNet34	21.54	5.71	57.97	81.02
PKT [38] ∗	ResNet101	ResNet34	21.54	5.71	57.02	80.31
MBDL [56] ∗	ResNet101	ResNet34	21.54	5.71	57.18	80.50
DKD [59] ∗	ResNet101	ResNet34	21.54	5.71	56.95	80.63
CSKD [53] ∗	ResNet101	ResNet34	21.54	7.31	54.89	79.17
KDPE [19] ∗	ResNet101	ResNet34	21.54	7.31	52.45	78.18
ResRep [9] ∗	ResNet101	ResNet101	16.49	3.94	52.69	75.63
CDD+RGGR	ResNet101	ResNet101	**15.00**	**3.66**	**60.98**	**81.68**

[1] The ∗ represents the result is re-implemented.

[40, 41, 55]. Moreover, an advanced NP method (i.e., ResRep [9]) was also re-implemented to compare with our method. The comparison analyses are discussed as follows.

In-Shop dataset. Table 2 shows that CDD+RGGR has several significant advantages over other compression methods. Firstly, when using the same ResNet101 teacher, CDD+RGGR achieves the highest performance across all metrics compared to other KD methods. Specifically, regarding inference performance, CDD+RGGR outperforms these methods by 5.27M MP and 2.63G FLOPs. Secondly, compared to ResRep [9], CDD+RGGR achieves a higher accuracy performance of 1.42% mAP and 0.46% R1 while maintaining similar inference performance.

VeRi776 dataset. From Table 3, the compressed method CDD+RGGR outperforms non-compressed methods like PVEN [34] and ViT [13], which use complex backbone networks. Compared with other compressed methods that use the same ResNet101 teacher, CDD+RGGR can achieve higher accuracy with a lighter and more efficient student model with 14.30M MP and 4.46G FLOPs, along with 80.67% mAP and 96.66% R1. Specifically, regarding mAP accuracy, CDD+RGGR outperforms RKD [36] by 2.11%, PKT [38] by 2.22%, and all other competitors by a significant margin while using only 61.01% FLOPs of other KD methods. Besides, using ResNet50 as the teacher, CDD+RGGR loses first place in mAP to VKD [41], a self-distillation method that focuses on improving the accuracy of the original model (i.e., teacher models) itself rather than educating a lightweight student. Nevertheless, our approach is still valuable because CDD+RGGR can save 63.47% MP and 59.33% FLOPs with a slight loss in accuracy.

MSMT17 dataset. From Table 4, it is evident that the CDD+RGGR method using the ResNet101 model as the teacher network exhibits significant advantages in terms of accuracy performance metrics when compared to the non-compressed approaches, namely, IANet and BINet.

Table 5. Evaluation of two-stage cascaded designs on MSMT17.

Methods	MP (M)	FLOPs (G)	mAP (%)	R1 (%)
SSL [52]	15.11	3.57	46.23	70.89
Self-KD + SSL [52]	14.84	**3.56**	54.31	76.97
ResRep [9]	14.85	3.59	51.17	74.61
Self-KD + ResRep [9]	**14.82**	3.59	56.17	78.30
CDD + RGGR	15.00	3.66	**60.98**	**81.68**

Table 6. Ablation results on the retrieval matrix R on VeRi776.

Methods	Retrieval Metric	Queue	FLOPs	mAP	R1
RGGR w/o Cosine	Euclidean	✓	5.05	80.62	96.48
RGGR w/o Queue	Cosine	×	4.47	80.32	96.42
RGGR	Cosine	✓	**4.46**	**80.67**	**96.66**

Specifically, CDD+RGGR achieves the highest mAP (i.e., 60.98%) and the highest R1 (i.e., 81.68%) among all the methods. Additionally, it is worth noting that CDD+RGGR also outperforms the KD methods by 6.54M MP and 2.05G FLOPs in terms of inference performance, which further underscores its superiority as a compression method.

4.5. Analysis Experiments

The two-stage method *vs* CDD+RGGR. To evaluate the effectiveness of our integrated end-to-end KD+NP method, we conducted a comparative study as shown in Table. 5. For a fair comparison, we decompose our approach into a cascaded setting. Specifically, we first employ self-KD to distill a heavy student model. Then, the well-trained heavy student model is pruned to a light student by ResRep [9]. Notably, RGGR retrieval-based metric is disabled, and we use the convolutional layer weight metric of ResRep [9] instead for stable training. Finally, the light student is fine-tuned to restore some performance. From Table 5, it can be observed that CDD + RGGR significantly outperforms self-KD + ResRep by 4.18% mAP and 3.38% R1 because ResRep may be difficult for the pruned model to completely retain the performance of the distilled large student without teachers' guidance. Moreover, we have also compared CDD + RGGR with another pruning method SSL [52], and a similar conclusion can be observed. These results demonstrate the superiority of the integrated end-to-end KD+NP method over the two-stage cascaded method.

The ablation results for the retrieval matrix R. As shown in Table 6, we conduct ablation experiments for the retrieval matrix R on Veri776 [30]. Table 6, we can find that RGGR performs better performance with the cosine distance and queue mechanism in the retrieval matrix R.

The influence of top-K retrieval results (i.e. K in Eq. (7)). When the value of K is increased, RGGR has more reference information to zeroing the knowledge accumulation gradient of unimportant channels. As shown in Fig. 5, on Veri776 [30], the mAP performance of CDD+RGGR at $K > 1$ outperforms at $K = 1$.

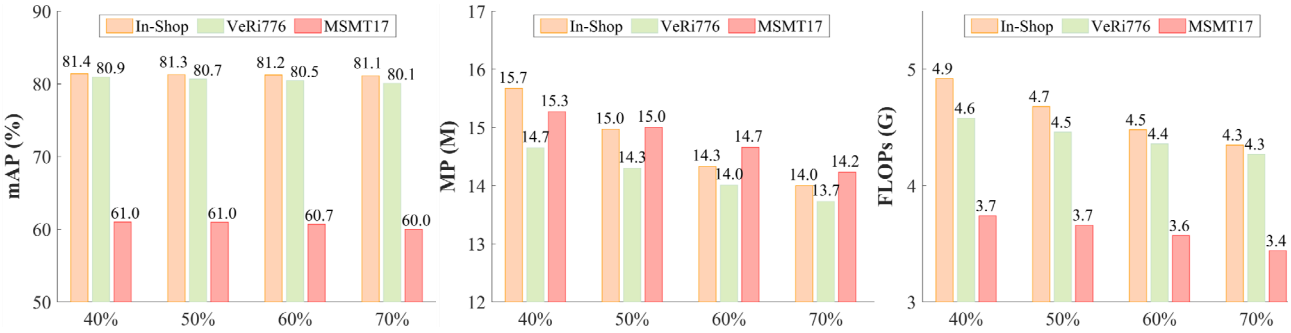

Figure 4. The influences of p values. (a) on mAP. (b) on MP. (c) on FLOPs. As p increases, the knowledge accumulation gradient of more channels is zeroed, resulting in a slight decrease in mAP and a noticeable improvement in inference performance.

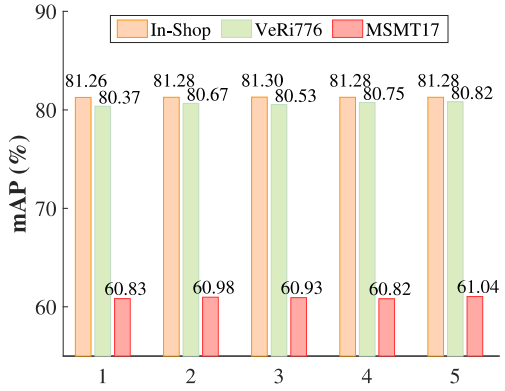

Figure 5. The K influence on mAP.

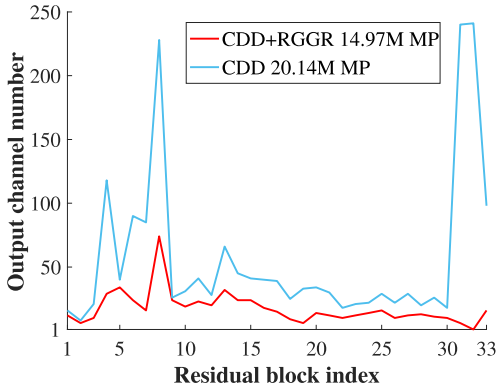

Figure 6. The output channel number of DGC in each residual block on In-shop [32]. With RGGR, the output channel number of DGC is significantly reduced, while mAP only drops 0.1%.

The influence of the channel selection mask ratio (i.e., $p(\%)$ in Eq. (9)). Fig. 4 show that the impact of p values on performance. Specifically, as the value of p increases, the knowledge accumulation gradient of more channels becomes zero, leading to a slight decrease in mAP performance but an increase in inference performance for CDD+RGGR. For instance, increasing the p value from 40% to 70% on In-shop [32], the FLOPs performance improved from 4.9 G to 4.3 G while mAP only dropped 0.27%.

Qualitative results. Fig. 6 shows the output channel number of slim DGC in each residual block on In-shop [32]. We can find that RGGR can effectively sparse DGC and thus significantly reduce the output channel number. For example, in the 32-th residual block, RGGR can reduce the output channel number of DGC from 241 to 1.

5. Conclusion

This paper proposes a Capacity Dynamic Distillation method for efficient image retrieval. Specifically, we use a heavy model as students to fully understand teachers' knowledge in early training. Simultaneously, the student model is gradually compressed during the training by the distillation guided compactor module. Furthermore, we propose the retrieval-guided gradient resetting mechanism to release the conflict between the learning gradient and the forgetting gradient. Extensive experiments demonstrate the heavy student model can be converted into a lightweight model without critical performance degeneration.

Limitation. Although our method achieves promising results on efficient CNNs, the performance on the transformer network is yet to be validated. In the future, we will extend our method to transformer-based knowledge distillation.

Broader Impact. Our method demonstrates that the end-to-end aggregation of KD and NP helps construct large-capacity student models, which can inspire the community to continue to explore compression methods for large-capacity student models. In addition, our approach can be applied to learn the efficient model on other matching-dependent tasks (e.g., Object re-identification).

Acknowledgement. The work is supported by Guangdong International Technology Cooperation Project (No.2022A0505050009); Key-Area Research and Development Program of Guangdong Province, China (2020B010166003, 2020B010165004); National Natural Science Foundation of China (No. 61972162); Guangdong Natural Science Foundation (No. 2021A1515012625); and Guangdong Natural Science Funds for Distinguished Young Scholar (No. 2023B1515020097).

References

[1] Alessandro Achille, Matteo Rovere, and Stefano Soatto. Critical learning periods in deep networks. In *ICLR*, 2018. 2

[2] Andrew Brown, Weidi Xie, Vicky Kalogeiton, and Andrew Zisserman. Smooth-ap: Smoothing the path towards large-scale image retrieval. In *ECCV*, pages 677–694, 2020. 1

[3] Pengguang Chen, Shu Liu, Hengshuang Zhao, and Jiaya Jia. Distilling knowledge via knowledge review. In *CVPR*, pages 5008–5017, 2021. 2, 3

[4] Xiumei Chen, Xiangtao Zheng, and Xiaoqiang Lu. Bidirectional interaction network for person re-identification. *IEEE Transactions on Image Processing*, 30:1935–1948, 2021. 7

[5] Xu Cheng, Zhefan Rao, Yilan Chen, and Quanshi Zhang. Explaining knowledge distillation by quantifying the knowledge. In *CVPR*, pages 12922–12932, 2020. 2

[6] Jang Hyun Cho and Bharath Hariharan. On the efficacy of knowledge distillation. In *ICCV*, pages 4794–4802, 2019. 1, 2

[7] Xiaohan Ding, Guiguang Ding, Jungong Han, and Sheng Tang. Auto-balanced filter pruning for efficient convolutional neural networks. In *AAAI*, volume 32, 2018. 2

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

[9] Xiaohan Ding, Tianxiang Hao, Jianchao Tan, Ji Liu, Jungong Han, Yuchen Guo, and Guiguang Ding. Resrep: Lossless cnn pruning via decoupling remembering and forgetting. In *ICCV*, pages 4510–4520, 2021. 2, 3, 4, 6, 7

[10] Xiaohan Ding, Xiangyu Zhang, Jungong Han, and Guiguang Ding. Diverse branch block: Building a convolution as an inception-like unit. In *CVPR*, pages 10886–10895, 2021. 2

[11] Xiaohan Ding, Xiangyu Zhang, Ningning Ma, Jungong Han, Guiguang Ding, and Jian Sun. Repvgg: Making vgg-style convnets great again. In *CVPR*, pages 13733–13742, 2021. 2

[12] Xiaohan Ding, Xiangxin Zhou, Yuchen Guo, Jungong Han, Ji Liu, et al. Global sparse momentum sgd for pruning very deep neural networks. In *NIPS*, pages 6379–6391, 2019. 4

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 6, 7

[14] Jonathan Frankle, David J Schwab, and Ari S Morcos. The early phase of neural network training. In *ICLR*, 2019. 2

[15] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017. 6

[16] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pages 9729–9738, 2020. 4

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 3, 5

[18] Lingxiao He, Xingyu Liao, Wu Liu, Xinchen Liu, Peng Cheng, and Tao Mei. Fastreid: A pytorch toolbox for general instance re-identification. *arXiv preprint arXiv:2006.02631*, 2020. 6

[19] Ruifei He, Shuyang Sun, Jihan Yang, Song Bai, and Xiaojuan Qi. Knowledge distillation as efficient pre-training: Faster convergence, higher data-efficiency, and better transferability. In *CVPR*, pages 9161–9171, 2022. 3, 6, 7

[20] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *ArXiv*, 1703.07737, 2017. 3

[21] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015. 1, 3, 6

[22] Ruibing Hou, Bingpeng Ma, Hong Chang, Xinqian Gu, Shiguang Shan, and Xilin Chen. Interaction-and-aggregation network for person re-identification. In *CVPR*, pages 9317–9326, 2019. 7

[23] Zehao Huang and Naiyan Wang. Data-driven sparse structure selection for deep neural networks. In *ECCV*, pages 317–334, 2018. 3

[24] Xin Jin, Cuiling Lan, Wenjun Zeng, and Zhibo Chen. Uncertainty-aware multi-shot knowledge distillation for image-based object re-identification. In *AAAI*, pages 11165–11172, 2020. 2, 6

[25] Jangho Kim, Seonguk Park, and Nojun Kwak. Paraphrasing complex network: Network compression via factor transfer. In *NIPS*, pages 2765–2774, 2018. 6, 7

[26] Kyungyul Kim, ByeongMoon Ji, Doyoung Yoon, and Sangheum Hwang. Self-knowledge distillation with progressive refinement of targets. In *ICCV*, pages 6567–6576, 2021. 2

[27] Nikos Komodakis and Sergey Zagoruyko. Paying more attention to attention: improving the performance of convolutional neural networks via attention transfer. In *ICLR*, 2017. 3, 6, 7

[28] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1106–1114, 2012. 6

[29] Li Liu, Qingle Huang, Sihao Lin, Hongwei Xie, Bing Wang, Xiaojun Chang, and Xiaodan Liang. Exploring inter-channel correlation for diversity-preserved knowledge distillation. In *ICCV*, pages 8271–8280, 2021. 1

[30] Xinchen Liu, Wu Liu, Huadong Ma, and Huiyuan Fu. Large-scale vehicle re-identification in urban surveillance videos. In *ICME*, pages 1–6, 2016. 1, 5, 6, 7

[31] Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. Learning efficient convolutional networks through network slimming. In *ICCV*, pages 2755–2763, 2017. 2

[32] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *CVPR*, pages 1096–1104, 2016. 5, 6, 8

[33] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *ICLR*, 2017. 6

[34] Dechao Meng, Liang Li, Xuejing Liu, Yadong Li, Shijie Yang, Zheng-Jun Zha, Xingyu Gao, Shuhui Wang, and Qingming Huang. Parsing-based view-aware embedding network for vehicle re-identification. In *CVPR*, pages 7101–7110, 2020. 6, 7

[35] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant. In *AAAI*, pages 5191–5198, 2020. 2

[36] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *CVPR*, pages 3967–3976, 2019. 6, 7

[37] Nikolaos Passalis, Maria Tzelepi, and Anastasios Tefas. Heterogeneous knowledge distillation using information flow modeling. In *CVPR*, pages 2339–2348, 2020. 1

[38] Nikolaos Passalis, Maria Tzelepi, and Anastasios Tefas. Probabilistic knowledge transfer for lightweight deep representation learning. *IEEE Transactions on Neural Networks and Learning Systems*, 32(5):2030–2039, 2020. 6, 7

[39] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *NIPS*, 32, 2019. 5

[40] Baoyun Peng, Xiao Jin, Jiaheng Liu, Dongsheng Li, Yichao Wu, Yu Liu, Shunfeng Zhou, and Zhaoning Zhang. Correlation congruence for knowledge distillation. In *ICCV*, pages 5007–5016, 2019. 1, 2, 3, 6, 7

[41] Angelo Porrello, Luca Bergamini, and Simone Calderara. Robust re-identification by multiple views knowledge distillation. In *ECCV*, pages 93–110, 2020. 2, 3, 6, 7

[42] Sucheng Ren, Yong Du, Jianming Lv, Guoqiang Han, and Shengfeng He. Learning from the master: Distilling cross-modal advanced knowledge for lip reading. In *CVPR*, pages 13325–13333, 2021. 1

[43] Sucheng Ren, Zhengqi Gao, Tianyu Hua, Zihui Xue, Yonglong Tian, Shengfeng He, and Hang Zhao. Co-advise: Cross inductive bias distillation. In *CVPR*, pages 16773–16782, 2022. 1

[44] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *ECCV*, pages 17–35, 2016. 5

[45] Fei Shen, Jianqing Zhu, Xiaobin Zhu, Yi Xie, and Jingchang Huang. Exploring spatial significance via hybrid pyramidal graph network for vehicle re-identification. *IEEE Transactions on Intelligent Transportation Systems*, 23(7):8793–8804, 2022. 1

[46] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, pages 2818–2826, 2016. 3

[47] Jialin Tian, Xing Xu, Zheng Wang, Fumin Shen, and Xin Liu. Relationship-preserving knowledge distillation for zero-shot sketch based image retrieval. In *ACMMM*, pages 5473–5481, 2021. 1

[48] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. In *ICLR*, 2019. 4

[49] Frederick Tung and Greg Mori. Similarity-preserving knowledge distillation. In *ICCV*, pages 1365–1374, 2019. 6, 7

[50] Kan Wang, Pengfei Wang, Changxing Ding, and Dacheng Tao. Batch coherence-driven network for part-aware person re-identification. *IEEE Transactions on Image Processing*, 30:3405–3418, 2021. 1

[51] Longhui Wei, Shiliang Zhang, Wen Gao, and Qi Tian. Person transfer gan to bridge domain gap for person re-identification. In *CVPR*, pages 79–88, 2018. 5, 6

[52] Wei Wen, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. Learning structured sparsity in deep neural networks. In *NIPS*, pages 2074–2082, 2016. 1, 2, 7

[53] Hui Wu, Min Wang, Wengang Zhou, Houqiang Li, and Qi Tian. Contextual similarity distillation for asymmetric image retrieval. In *CVPR*, pages 9489–9498, 2022. 2, 6, 7

[54] 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. 1, 3, 5, 6

[55] Yi Xie, Hanxiao Wu, Fei Shen, Jianqing Zhu, and Huanqiang Zeng. Object re-identification using teacher-like and light students. In *BMVC*, 2021. 7

[56] Yi Xie, Jianqing Zhu, Huanqiang Zeng, Canhui Cai, and Lixin Zheng. Learning matching behavior differences for compressing vehicle re-identification models. In *VCIP*, pages 523–526, 2020. 1, 2, 5, 6, 7

[57] Chao Ye, Huaidong Zhang, Xuemiao Xu, Weiwei Cai, Jing Qin, and Kup-Sze Choi. Object detection in densely packed scenes via semi-supervised learning with dual consistency. In *IJCAI*, pages 1245–1251, 2021. 1

[58] Jiehua Zhang, Zhuo Su, Yanghe Feng, Xin Lu, Matti Pietikäinen, and Li Liu. Dynamic binary neural network by learning channel-wise thresholds. In *ICASSP*, pages 1885–1889, 2022. 1

[59] Borui Zhao, Quan Cui, Renjie Song, Yiyu Qiu, and Jiajun Liang. Decoupled knowledge distillation. In *CVPR*, pages 11953–11962, 2022. 1, 3, 6, 7

[60] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *AAAI*, pages 13001–13008, 2020. 5