ADAPTIVE REGION POOLING FOR FINE-GRAINED REPRESENTATION LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Fine-grained recognition aims to discriminate the sub-categories of the images within one general category. It is fundamentally difficult due to the requirement to extract fine-grained features from subtle regions. Nonetheless, a Convolutional Neural Network typically applies strided operations to downsample the representation, which would excessively spoil the feature resolution and lead to a significant loss of fine-grained information. In this paper, we propose Adaptive Region Pooling (ARP): a novel downsampling algorithm that makes the network only focus on a smaller but more critical region, and simultaneously increase the resolution of sub-sampled feature. ARP owns a trade-off mechanism that allows users to actively balance the scale of receptive field and the granularity of feature. Also, without any learning-based parameters, ARP provides the network a stabler training process and an earlier convergence. Extensive experiments qualitatively and quantitatively validate the effectiveness and efficiency of the proposed pooling operation and show superior performance against the state-of-the-arts in both the tasks of image classification and image retrieval.

1 INTRODUCTION

The main goal of fine-grained recognition is to discriminate the sub-categories of the images that belong to a same general class. Its practical applications include fine-grained image classification (Welinder et al., 2010; Krause et al., 2013; Maji et al., 2013), re-identification (Zheng et al., 2017b; Liu et al., 2016b;a), etc. These tasks are fundamentally challenging due to considerable inter-class similarity. Consequently, an ideal feature extractor should be capable of extracting discriminative fine-grained features from subtle image regions, such as the beak to classify the bird species or the stickers on the windshield to identify the vehicle.

While the general design of Convolutional Neural Network (CNN) (Simonyan & Zisserman, 2014; He et al., 2016) applies either strided convolution or strided pooling as downsampling operations to reduce the computational requirement of the network and also increase the receptive field of subsequent convolutions, these operations, however, have been substantiated to be harmful to fine-grained recognition (Sun et al., 2018; Luo et al., 2019a). Its disadvantage is two-fold: 1) false assumption on spatial equality which would overemphasize the background region while neglecting the smaller but more discriminative regions and 2) excessive reduction of resolution which would spoil the granularity of feature. To cope with the problem, existing approaches (Zhang et al., 2014; Fu et al., 2017; Chen & Deng, 2019; Zhang et al., 2019) introduce cropping operation that guides the network to only focus on the most critical region while preserving more details in the sampled part. However, to maintain computational complexity, full conservation of feature resolution would instead restrict a limited scale, or area, of the cropped region. Comprehensively, as shown in Figure 1(a)(c), the problems of two commonly used downsamplings can be summarized into a trade-off problem between the scale of receptive field and the granularity of representation.

To tackle the problem, in this work, we introduce a novel pooling algorithm, named *Adaptive Region Pooling (ARP)*, to bridge strided and cropping operations. It consists of two steps. First, ARP automatically crops the feature from the most critical region with a well-estimated and adaptive cropping scale. Second, after being cropped into different sizes, the feature is further downsampled to a consistent size through bilinear downsampling. As illustrated in Figure 1(b), with the meticulous evaluation of cropping scale, ARP can better handle the trade-off issue and leverage the

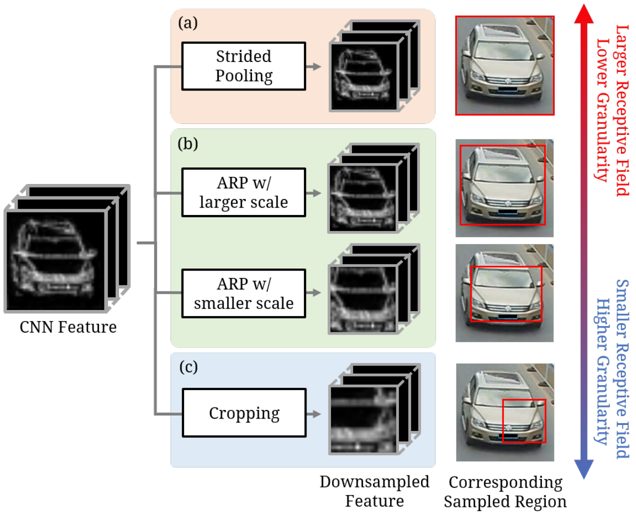

Figure 1: **Concept of Adaptive Region Pooling.** We compare three operations to downsample the feature into half size: (a) pooling with stride = 2, (b) our Adaptive Region Pooling (ARP), and (c) cropping operation. ARP smoothly bridges two widely used operations by automatically sampling the feature from the most critical region (red box) with a well-estimated cropping scale. Furthermore, as in two cases of (b), users can manually balance the scale of receptive field and the granularity of downsampled feature through a controllable trade-off mechanism.

beneficial properties of both strided and cropping operations. Besides, ARP also features other three advantages. First, it is incorporated with a controllable mechanism that supports users to actively reach a more satisfying balance in the trade-off problem, as in two cases in Figure 1(b). Second, compared to the previous work (Lin et al., 2015a; Huang et al., 2016; Kim et al., 2018; Zheng et al., 2019) which implement the cropping operations by an additional neural network, ARP utilizes a more efficient yet effective operation without any learning-based parameters. Hence, it makes the network achieve lower computational overhead, experience a stabler training process, and converge more quickly. Finally, ARP can be simply plugged into any CNN layer and the network can be optimized in a one-stage end-to-end manner, which is completely identical to the training of original CNN backbone.

At last, we demonstrate that ARP could facilitate fine-grained recognition through our *Multiple Scale and Granularity Network (MSGN)*. In MSGN, ARP serves as a downsampling alternative to extract the feature of finer details complementary to the coarse feature provided by the typical downsampling method. The contributions of this paper can be summarized as follows:

- We introduce Adaptive Region Pooling (ARP): a novel downsampling that smoothly bridges strided pooling and cropping operations. The sub-sampled representation focuses on the most discriminative region and simultaneously contains more fine-grained information.
- To our knowledge, ARP is the first pooling algorithm with an adjustable trade-off mechanism that allows users to apply the human domain knowledge of the target task to strike a more desirable balance between the scale of receptive field and the granularity of downsampled feature.
- With the usage of ARP, MSGN outperforms the existing frameworks in terms of both effectiveness and efficiency on multiple benchmarks for both the tasks of image classification and retrieval.

2 Related Work

Fine-grained recognition focuses on learning discriminative representations to classify or identify the images within a same general category. There are two common applications: re-identification and fine-grained image classification.

Re-identification (re-ID) studies the problem of identifying the individuals of persons (Zheng et al., 2015; 2017b; Ristani et al., 2016) or vehicles (Liu et al., 2016b;c;a) in different camera views. To distinguish two similar targets with only slight differences, most of the existing methods (Yao et al., 2019; Kalayeh et al., 2018; Liu et al., 2021; Wang et al., 2017; Zhou & Shao, 2018; Chen et al., 2020b) adopt spatial attentive mechanism to emphasize the features extracted from critical parts, such as the head for person re-ID or the tires for vehicle re-ID. While previous work get significant improvement, they generally suffer from massive loss of fine-grained details when CNN backbone downsamples the feature through strided operations. Sun et al. (2018) and Luo et al. (2019a) both

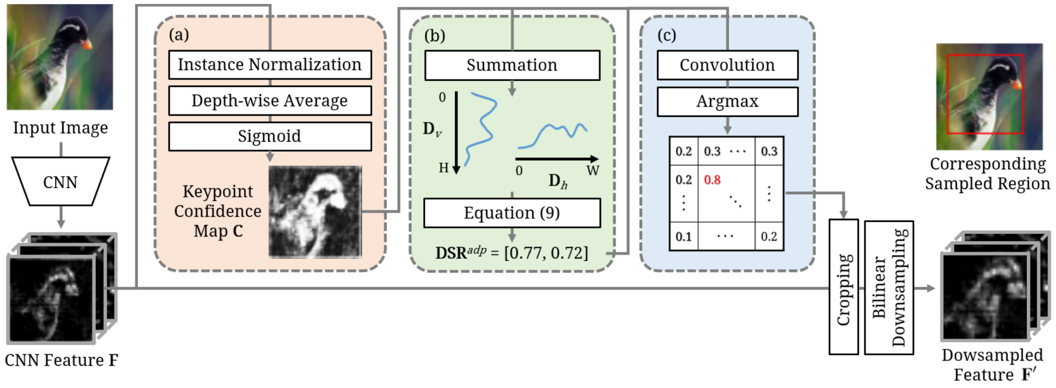

Figure 2: **Overall workflow of Adaptive Region Pooling.**

find a huge performance gain in the task of person re-ID, numerically 1.4% for mAP on Market1501 (Zheng et al., 2015), by simply removing the pooling operation in the last CNN layer.

Fine-grained image classification (FGIC) aims to classify the sub-classes of images, such as the species of birds (Welinder et al., 2010) or the models of vehicles (Krause et al., 2013). Compared to re-ID, the datasets of FGIC have a severe problem of image misalignment which impels lots of work to apply another sampling operation, cropping, to align the image or representation and also guide the network to focus on the sampled region. Several prior work (Welinder et al., 2010; Zhang et al., 2013; 2014; Lin et al., 2015a; Huang et al., 2016; Kim et al., 2018; He et al., 2019) leverage extra object detection datasets or human-labeled bounding box annotations to learn the localization of discriminative regions. However, such auxiliary labels are hard to acquire in the real-world and large-scale scenarios. Recently, there have been emerging work with a more general setting by proposing unsupervised approaches. Jaderberg et al. (2015) introduce a differentiable STN that can learn to well align the images or representations by themselves. While it can perform well on the digit datasets (LeCun et al., 1998; Netzer et al., 2011), lack of explicit guidance leads to the unstable training process and being hard to converge on more sophisticated tasks (see Section 4.3 for more details and discussions). As the following work, Fu et al. (2017) propose RACNN which is a cascade coarse-to-fine CNN model. The similar architecture is also adopted by Chen & Deng (2019) and Zhang et al. (2019). Although they show the great capability of cropping the discriminative region, the cascade architecture requires multi-stage training for better optimization and struggles with the requirement of heavy computation. In this work, we also introduce a sampling operation with an unsupervised approach; however, Adaptive Region Pooling (ARP) contains neither additional neural network nor extra learning-based parameters and makes the network using ARP more computation-efficient and able to be optimized in a simple one-stage end-to-end manner.

3 Proposed Methodology

In this section, we introduce the proposed pooling algorithm in three parts. First, in Section 3.1, *Region Pooling (RP)* considers a simplified task which aims to crop the hidden representation from the most critical region with a fixed cropping scale. Then, in Section 3.2, *Adaptive Region Pooling (ARP)* estimates the adaptive cropping scale for each representation to better capture the interesting object. Finally, we introduce *Multiple Scale and Granularity Network (MSGN)* as an example of network architecture with the usage of ARP in Section 3.3.

3.1 Region Pooling

Given a CNN feature and a fixed downsampling rate, the main goal of Region Pooling (RP) is to crop the feature from the most critical region into a consistent scale. An intuitive approach is to use an extra network to predict the cropping center which nevertheless, often suffers from the requirement of expensive bounding box annotations or complicated multi-stage learning strategy to train the network. In contrast, RP uses a more efficient yet effective operation with two steps.

In the first step, as shown in Figure 2 (a), RP would evaluate the significance score for each location and generate the keypoint confidence map **C** based on the C-dim feature vectors, where C denotes the number of channels. Here, we define "significance" in an engineering sense as follows: a location, whose feature vector includes more highly-activated neurons, is more significant. The physical meaning is that a channel-wise feature map is actually a detector of a certain semantic object (Chen et al., 2017; 2020a), such as a bird's beak or wings. For a well-trained CNN, most of the channels would detect the objects which are more beneficial to the target task; thus, the locations activated by more channels typically cover critical semantic objects and are more significant.

Based on the definition, we then give a formal formulation as follows. For the input feature $\mathbf{F} \in \mathcal{R}^{C\times H\times W}$ with width W, height H and C channels and its element (or activated value) denoted as $x_{(i,j,k)}$ at i^{th} channel, j^{th} row and k^{th} column, in order to balance the influence of each channel in the subsequent operations, we first standardize each channel-wise feature map as:

$$\widehat{x}_{(i,j,k)} = \frac{x_{(i,j,k)} - \mu_{(i)}}{\sigma_{(i)}}, \text{ where } \mu_{(i)} = \frac{\sum\limits_{j}^{H}\sum\limits_{k}^{W} x_{(i,j,k)}}{H\cdot W}, \; \sigma^2_{(i)} = \frac{\sum\limits_{j}^{H}\sum\limits_{k}^{W}(x_{(i,j,k)} - \mu_{(i)})^2}{H\cdot W}, \tag{1}$$

and $\widehat{x}$ represents the element of the standardized feature. Such standardization can be simply implemented by Instance Normalization (Ulyanov et al., 2016). Afterwards, we evaluate the significance score of each location based on the average of their feature vectors with the form as:

$$s_{(j,k)} = \frac{\sum\limits_{i=1}^{C} \widehat{x}_{(i,j,k)}}{C} \tag{2}$$

where $s_{(j,k)}$ represents the significance score of the location (j,k). Complied with our definition, Equation 2 assigns larger s for the locations that are highly activated in more channels. Next, we estimate the confidence of a location to be a "keypoint", or a critical location, by Sigmoid function $\sigma(\cdot)$. Specifically, we generate a keypoint confidence map $\mathbf{C} \in \mathcal{R}^{H\times W}$ by:

$$\mathbf{C} = [\, c_{(1,1)},\; ...,\; c_{(H,W)}\,] = [\,\sigma(s_{(1,1)}),\; ...,\; \sigma(s_{(H,W)})\,] \tag{3}$$

where $c_{(j,k)}$ denotes the keypoint confidence at the location (j,k). We visualize the generated keypoint confidence maps **C** in Figure 5 and 6. If we project them to the original images, the locations with larger keypoint confidences effectively represent the more discriminative regions, such as the beak and colorful feather of a bird or the bumper and lamp of a vehicle.

In the second step, as in Figure 2(c), given a fixed downsampling rate $\mathbf{DSR} = [DSR_H, DSR_W]$, RP would crop the most critical region with height $(H\cdot DSR_H)$ and width $(W\cdot DSR_W)$. To this end, we apply a convolution operation on the keypoint confidence map **C** with a kernel $\mathbf{K} \in \mathcal{R}^{(H\cdot DSR_H)\times(W\cdot DSR_W)}$. The region with the largest weighted summation of keypoint confidences would be selected as the cropped region, which can be written as:

$$(j^*, k^*) = \underset{(j,k)}{\arg\max}(\mathbf{K}\odot \mathbf{C}_{(j,k)}) \tag{4}$$

where $\odot$ represents inner product, $\mathbf{C}_{(j,k)}$ is the cropped keypoint confidence map centered at the location (j,k), and (j^*,k^*) indicates the center location of the selected cropped region.

Discussion of kernel K. Empirically, we find that using Gaussian filter as kernel **K** yields the best performance. A potential explanation is that it tends to select the region with more keypoints locating near the center of cropped region and hence, also solves the problem of feature misalignment.

3.2 Adaptive Region Pooling

Although Region Pooling demonstrates the great ability to sample the feature from the most critical region, nonetheless, there is no single downsampling rate suitable for all features. Specifically, the features with highly separate keypoints should have a larger downsampling rate to cover a wider interesting region while in contrast, the ones with more concentrated keypoints should use a smaller rate. To meet this requirement, we further introduce Adaptive Region Pooling (ARP) that automatically estimates a better adaptive downsampling rate based on the distribution of keypoints, or more precisely: the standard deviation (abbreviated as std) of keypoints distribution.

As depicted in Figure 2(b), to evaluate the adaptive downsampling rate of height DSR_H^{adp} or width DSR_W^{adp}, ARP respectively refers the vertical or horizontal normalized distribution of keypoints:

$$\mathbf{D}_v = \left[\sum_{k=1}^{W} \frac{c_{(1,k)}}{\Sigma\,\mathbf{C}},\ ...,\ \sum_{k=1}^{W} \frac{c_{(H,k)}}{\Sigma\,\mathbf{C}} \right] \in \mathcal{R}^H,\ \mathbf{D}_h = \left[\sum_{j=1}^{H} \frac{c_{(j,1)}}{\Sigma\,\mathbf{C}},\ ...,\ \sum_{j=1}^{H} \frac{c_{(j,W)}}{\Sigma\,\mathbf{C}} \right] \in \mathcal{R}^W \tag{5}$$

where $\Sigma\mathbf{C}$ is the summation of keypoint confidence map.

Since the derivations of DSR_H^{adp} and DSR_W^{adp} take similar forms, we use the former as an example in the following. Here, we first consider a simplified condition that each image only contains one consecutive interesting object and the keypoints uniformly distribute in the target object. We will explore more circumstances in the following discussion and Appendix A. Under this assumption, the keypoints distribution can be regarded as a discrete uniform distribution:

$$\mathbf{D}_v(y) = \begin{cases} \dfrac{1}{b-a}, & a < y \leqslant b \\ 0\,, & \text{otherwise}\,, \end{cases} \qquad \forall\, 0 \leq a < b \leq H, \tag{6}$$

with its std:

$$\sigma(\mathbf{D}_v) = \sqrt{\frac{(b-a)^2 - 1}{12}} \approx \frac{b-a}{\sqrt{12}}. \tag{7}$$

Note that $\mathbf{D}_v$ has the keypoints all lying within the interval $(a, b]$. Therefore, if the cropped region is expected to cover all keypoints while ignoring the background, the ideal length of cropped region should be $(b - a)$ and hence, the expected downsampling rate should be $(b - a)/H$. Combing with Equation 7, we can rewrite the adaptive downsampling rate as:

$$DSR_H^{adp} = \frac{b-a}{H} = \frac{\sqrt{12}}{H} \cdot \sigma(\mathbf{D}_v). \tag{8}$$

Finally, with Equation 8, we can use the std of keypoints distribution to well estimate the adaptive downsampling rate that could exactly include all keypoints in the cropped region.

Discussion of different distributions D. Intuitively, in Equation 8, the adaptive downsampling rate is proportional to the std of keypoints distribution. Therefore, for more sparse or dense keypoints distribution (e.g., M-shaped or bell-shaped distribution), the cropped region could automatically be wider or narrower to better capture the interesting object. In Appendix A, we also empirically test the stability and rationality of Equation 8 on various randomly simulated distributions and show that Equation 8 can consistently evaluate an intelligible adaptive downsampling rate that can exactly cover most of the keypoints.

Subsequently, as in Figure 2, ARP first crops the feature $\mathbf{F}$ from the most critical region based on Equation 4 and adaptive downsampling rate as $\mathbf{F}_{crop} \in \mathcal{R}^{C\times(H\cdot DSR_H^{adp})\times(W\cdot DSR_W^{adp})}$ and then, subsamples it to a consistent size as $\mathbf{F}' \in \mathcal{R}^{C\times(H\cdot DSR_H)\times(W\cdot DSR_W)}$ through bilinear downsampling. Such a "cropping and downsampling" process can be viewed as the trade-off between the scale of receptive field and the resolution of feature. Specifically, the feature with a larger cropped region would encounter a huge reduction of the feature granularity in the following downsampling which makes ARP act as strided operation. In contrast, a smaller cropped region would preserve more fine-grained details in the downsampled feature; in such a case, ARP intrinsically behaves like cropping function. To allow users to strike a better balance in the trade-off problem, we further incorporate ARP with a controllable mechanism by revising Equation 8 as:

$$DSR_H^{adp} = \min\left(\max\left(k \cdot \frac{\sqrt{12}}{H} \cdot \sigma(\mathbf{D}_v), DSR_H\right), 1\right). \tag{9}$$

k is the trade-off coefficient and min-max operation ensures a reasonable downsampling rate.

Discussion of trade-off coefficient *k*. k is a meaningful coefficient. When it is set to 1, the cropped region would nearly cover the whole target object, such as an entire bird or vehicle. And if a smaller k is set, the length of cropped region would be proportionally decreased. Through k, users can actively reach a satisfying sampling scale according to the human domain knowledge of the target task. Take the below case of Figure 1(b) as an example, users can set a smaller k to extract the feature only from the front face of vehicle, which usually contains many identity-relative features, rather than the whole vehicle. Hence, the sub-sampled feature would keep richer fine-grained details.

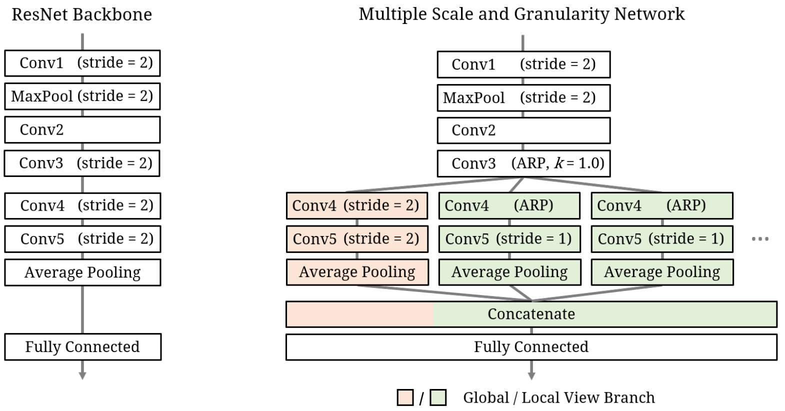

Figure 3: **Architecture of Multiple Scale and Granularity Network.**

3.3 Multiple Scale and Granularity Network

We introduce Multiple Scale and Granularity Network (MSGN) as an example of the network architecture that uses ARP as the downsampling operations. As in Figure 3, MSGN has one low-level feature extractor followed by multiple parallel high-level extractors. Considering that the hidden representations before Conv3 merely contain the information of low-level patterns which are unreliable to determine the critical region, we only adopt ARP with $k = 1$ in Conv3 to remove some background regions in the early stage. As for the following high-level extractors, including one global and several local view branches, we respectively apply strided operation and ARP with smaller k in Conv4 to downsample the feature with different scales and granularities. We also follow the strategy from Sun et al. (2018) and Luo et al. (2019a) to remove pooling operations in Conv5 of local view branches. Finally, the features extracted from the multiple branches would be concatenated and fed into a fully connected layer to comprehensively generate the final representative embedding by referring to the features from both global and local views. We will first validate MSGN with one local view branch in Section 4 and further, discuss the network with more branches in Appendix B.

4 Experiments

In the following, we evaluate ARP through MSGN in both tasks of fine-grained image classification (FGIC) and vehicle re-identification (re-ID) for image retrieval.

4.1 Implementation Details

We implement MSGN with three backbones: ResNet-50, ResNet-101 and ResNeXt-101 32x8d (He et al., 2016; Xie et al., 2017). For a fair comparison, we set **DSR** of ARP as $[0.5, 0.5]$ to share the same height and width of the downsampled feature as from the pooling with stride $= 2$. Following the same protocol by Zhang et al. (2019) and Ji et al. (2020), we augment input images by resizing to 512×512 then randomly cropping to 448×448 with random horizontal flipping and use cross-entropy to supervise the training. For vehicle re-ID, as in standard literature (Luo et al., 2019a; Chen et al., 2020b), we directly resize images to 224×224 and use the combinations of batch-hard triplet loss (Hermans et al., 2017) and cross-entropy loss for the supervision. The training lasts for 100 epochs. More training and testing details can be found in Appendix C.

4.2 Comparison with the State-of-the-Arts

Fine-grained image classification. We first evaluate MSGN on FGIC in terms of both effectiveness and efficiency and report the results in Table 1. MSGN achieves the new state-of-the-art performance: 90.2% on CUB-200-2011, 95.6% on Stanford Cars, and 94.4% on FGVC-Aircraft with ResNeXt-101. With ResNet-50 or ResNet-101, MSGN also shows a consistent improvement over the previous counterparts with the same CNN backbone. In addition to effectiveness, our model also achieves a significant enhancement in efficiency. While previous work apply the additional network

Table 1: **Comparison on fine-grained image classification.** We report Top-1 classification accuracy (%) on CUB-200-2011 (Wah et al., 2011), Stanford Cars (Krause et al., 2013), and FGVC-Aircraft (Maji et al., 2013). We also compare floating-point operations (FLOPs) with the recent work using an input image size of 224 × 224. All models are from the official repositories.

Method	Backbone	GFLOPs ↓	CUB	Cars	Aircraft
			Acc.	Acc.	Acc.
STN (Jaderberg et al., 2015)	GoogleNet	-	84.1	-	-
B-CNN (Lin et al., 2015b)	VGG-16	-	84.1	91.3	84.1
RA-CNN (Fu et al., 2017)	VGG-19	-	85.3	92.5	-
MA-CNN (Zheng et al., 2017a)	VGG-19	-	86.5	92.8	89.9
HBP (Yu et al., 2018)	VGG-16	-	87.1	93.7	90.3
DFL-CNN (Wang et al., 2018)	VGG-16	-	87.4	93.8	92.0
TASN (Zheng et al., 2019)	ResNet-50	37.4	87.9	93.8	-
MGE (Zhang et al., 2019)	ResNet-101	69.2	89.4	93.6	-
S3N (Ding et al., 2019)	ResNet-50	40.8	88.5	94.7	92.8
CrossX (Luo et al., 2019b)	ResNet-50	32.1	87.7	94.6	92.7
ACNet (Ji et al., 2020)	ResNet-50	49.8	88.1	94.6	92.4
MSGN (Ours)	ResNet-50	**17.8** (44.5%↓)	89.1	95.0	93.6
MSGN (Ours)	ResNet-101	**32.7** (52.7%↓)	89.8	95.1	94.1
MSGN (Ours)	ResNeXt-101	32.8	**90.2**	**95.6**	**94.4**

Table 2: **Comparison on vehicle re-identification.** We report Mean Average Precision (Zheng et al., 2015) (%) and rank-1 and rank-5 accuracy in Cumulative Matching Characteristics (%) on VeRi-776 (Liu et al., 2016b;c) and VehicleID (Liu et al., 2016a). We use ResNet-50 as the backbone.

Method	VeRi		VID-800		VID-1600		VID-2400	
	mAP	R-1	R-1	R-5	R-1	R-5	R-1	R-5
OIFE (Wang et al., 2017)	48.0	89.4	-	-	-	-	67.0	82.9
VAMI (Zhou & Shao, 2018)	50.1	-	63.1	83.3	52.9	75.1	47.3	70.3
RAM (Liu et al., 2018)	61.5	88.6	75.2	91.5	72.3	87.0	67.7	84.5
EALN (Lou et al., 2019)	57.4	84.4	75.1	88.1	71.8	83.9	69.3	81.4
AAVER (Khorramshahi et al., 2019)	61.2	89.0	74.7	93.8	68.6	90.0	63.5	85.6
PRN (He et al., 2019)	74.3	94.3	78.4	92.3	75.0	88.3	74.2	86.4
PVEN (Meng et al., 2020)	79.5	95.6	84.7	97.0	80.6	94.5	77.8	92.0
VehicleX (Yao et al., 2020)	73.3	95.0	79.8	93.2	76.7	90.3	73.9	88.2
SAVER (Khorramshahi et al., 2020)	79.6	96.4	79.9	95.2	77.6	91.1	75.3	88.3
MSGN (Ours)	**81.8**	**97.3**	**85.0**	**97.1**	**80.7**	**94.5**	**78.0**	**92.1**

to generate the sub-sampling parameters (Lin et al., 2015a; Zheng et al., 2019; Ding et al., 2019) or implement the cascade coarse-to-fine architecture (Fu et al., 2017; Chen & Deng, 2019; Zhang et al., 2019), ARP only utilizes efficient operations for sampling. Therefore, the computational cost of MSGN is approximately half of the previous models, specifically 44.5% lower than CrossX (Luo et al., 2019b) on ResNet-50 and 52.7% lower than MGE (Zhang et al., 2019) on ResNet-101.

Vehicle re-identification. We also evaluate the representation learning of MSGN on vehicle re-ID and report the results in Table 2. Prior approaches (Wang et al., 2017; Zhou & Shao, 2018; Meng et al., 2020; Chen et al., 2020b) mostly focus on spatial attentive mechanism but ignoring the loss of fine-grained information when the representation are sub-sampled through strided operation. In contrast, MSGN effectively solves the problem by using ARP as the downsampling operation. Hence, MSGN outperforms the previous frameworks on both VeRi-776 and VehicleID, especially on VeRi-776 by a notable margin of 2.2% on mAP compared to SAVER (Khorramshahi et al., 2020).

4.3 Comparison with Spatial Transformer Network

Spatial Transformer (ST) (Jaderberg et al., 2015) module is a differentiable sampling operation that can automatically learn to perform an affine transformation on an image or a hidden representation. In light of the similar functionality between ST module and ARP, we additionally compare these two poolings. In the experiment, we adopt ST module or ARP as the downsampling operation in Conv4 of local view branch in MSGN and plot the learning curves and the learned sampling outcomes in Figure 4. We then discuss the comparison under two aspects: stability and intervenability.

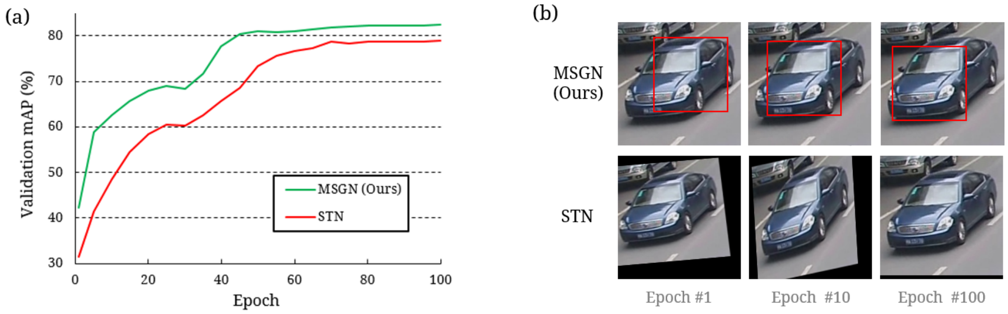

Figure 4: **Learning progress of MSGN or STN on VeRi-776.** We evaluate the network using ST module (Jaderberg et al., 2015) or ARP as the downsampling operation and respectively show the learning curves in (a) and the projections of learned downsamplings to the input images in (b).

Table 3: **Effectiveness analysis of the proposed components.** We validate the components in the proposed model including the usage of local view branch (LVB) and the pooling operation in Conv4.

Method	Components		CUB	VeRi
	LVB	Downsampling	Acc. (%)	mAP (%)
Baseline	✗	stride of 2	85.6	79.5
Baseline w/ ARP	✗	ARP ($k = 0.9$)	86.2	80.0
MSGN w/ stride	✓	stride of 2	87.1	80.4
MSGN w/ RP	✓	RP	88.0	81.1
MSGN w/ ARP (Ours)	✓	ARP ($k = 0.7$)	**89.1**	**81.8**

First, as shown in Figure 4(a), the network using ST module comparably experiences a slower learning progress, reaches later convergence, and eventually performs worse at the end of training. We can find a possible explanation from Figure 4(b); since ST module relies on learning-based parameters for sampling, in the first ten epochs, it performs unstably and generates unaccountable downsampling results which instead disturb the training. In contrast, ARP is able to correctly sample from the foreground object (red box in Figure 4(b)) even in the first epoch. Following the training, with more channels detecting the beneficial vehicle parts, the sampled region progressively covers the entire front face of vehicle which contains more identity-relative features. It is also worth noting that due to the instability of ST module, in the paper (Jaderberg et al., 2015), ST module is restricted to perform the 2 DoF transformation (i.e., only translation) in the harder task, like FGIC.

Next, at the end of the training, ST module fails to learn an effective transformation which is supposed to concentrate on the more critical region. It leads to the subsequent discussion: for learning-based cropping operations, users can only passively accept the training results and lack interpretable parameters to adjust the sampling operation to better fit the target task. Conversely, owing to a meaningful trade-off mechanism, ARP allows users to actively use the prior knowledge to intervene in the representation learning and reach the more desirable sub-sampling characteristic.

4.4 Ablation Studies

We evaluate the effectiveness of each component in the proposed framework and report the results in Table 3. The baseline model simply uses the original ResNet-50 as the extractor. We then replace the downsampling in Conv4 with ARP of $k = 0.9$. Leveraging the attention on more discriminative regions, the model achieves the improvement of 0.6% on CUB-200-2011 and 0.5% on VeRi-776. Next, as listed in the third to fifth rows in Table 3, we apply another high-level extractor with three different downsampling operations in Conv4, including strided convolution, RP with **DSR** $= [0.5, 0.5]$, and ARP with $k = 0.7$. It shows that, with a parallel branch, the performances are all boosted. However, without using RP and ARP, the network comparably performs worse due to the lack of fine-grained information. Furthermore, the network with ARP yields the best performance, which describes that ARP can adaptively estimate the better downsampling rate for each representation and properly cover the critical region. Finally, with all proposed components, MSGN with ARP outperforms the baseline model by 3.5% on CUB-200-2011 and 2.3% on VeRi-776.

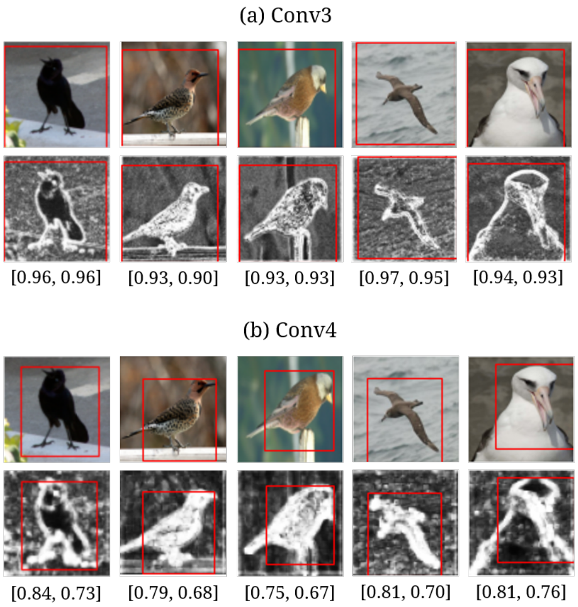

Figure 5: **Visualization on CUB-200-2011.** The first and second rows respectively show the input images and the keypoint confidence maps **C** with the cropped regions (red box). **DSR**^{adp} is also listed at the bottom. (a)(b) represent the results from Conv3 and Conv4.

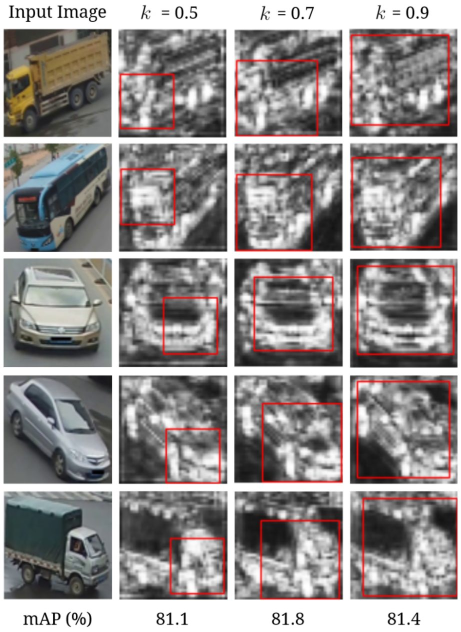

Figure 6: **Study of *k* on VeRi-776.** The second to fourth columns show the cropped regions (red box) determined by ARP with different k. The corresponding validation results are also listed at the bottom.

4.5 Visualization

Qualitative results of Adaptive Region Pooling. To understand the practical effect of ARP, we plot the keypoint confidence maps **C** and the cropped regions in Figure 5 and 6 (especially for the final setting of $k = 0.7$). Note that the demonstrated images are in various manners, such as different actions of birds (e.g., standing and flying) and different viewpoints or types of vehicles (e.g., truck, bus, SUV, and sedan) to validate the robustness of ARP. First, as in Figures 5(b) and 6, the keypoint confidence maps **C** from Conv4 are able to recognize the critical locations in images, such as the bird's beak and colorful feathers or the vehicle's bumper and decoration. Along with the proper adaptive downsampling rate, the sampled regions can cover the most critical and consistent part in different images. Also, as in Figure 5(a), ARP in Conv3 can eliminate some negligible background regions in the early stage while enriching the feature resolution in the foreground object.

Selection of trade-off coefficient *k*. We visualize the cropped regions and list the validation results with different k in Figure 6. As in the second to fourth columns, users can set $k = 0.5$ or 0.7 or 0.9 if the sampled region is respectively expected to cover merely a part of front face, or the whole front face with the windshield, or nearly the entire vehicle. Commonly, the front face of vehicle includes several important but fine identity-relative features which are beneficial to vehicle re-ID. Hence, with the aid of domain knowledge, the final setting of $k = 0.7$ yields the best performance.

5 Conclusion

In this paper, we propose a novel downsampling operation, Adaptive Region Pooling (ARP), which automatically samples the feature from the most critical region and in the same time, increases the feature granularity. Besides, ARP owns an adjustable trade-off mechanism which supports users to actively balance the scale of receptive field and the resolution of sub-sampled feature. Without any learning-based parameters, ARP can be simply integrated into a CNN backbone and the network can be stably end-to-end optimized. The experiments show the effectiveness and efficiency of the proposed model in both the tasks of image classification and image retrieval.

References

Binghui Chen and Weihong Deng. Hybrid-attention based decoupled metric learning for zero-shot image retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 1, 3, 7

Long Chen, Hanwang Zhang, Jun Xiao, Liqiang Nie, Jian Shao, Wei Liu, and Tat-Seng Chua. Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 4

Tsai-Shien Chen, Man-Yu Lee, Chih-Ting Liu, and Shao-Yi Chien. Viewpoint-aware channel-wise attentive network for vehicle re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2020a. 4

Tsai-Shien Chen, Chih-Ting Liu, Chih-Wei Wu, and Shao-Yi Chien. Orientation-aware vehicle re-identification with semantics-guided part attention network. In *European Conference on Computer Vision*, 2020b. 2, 6, 7

Yao Ding, Yanzhao Zhou, Yi Zhu, Qixiang Ye, and Jianbin Jiao. Selective sparse sampling for fine-grained image recognition. In *IEEE International Conference on Computer Vision*, 2019. 7

Jianlong Fu, Heliang Zheng, and Tao Mei. Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 1, 3, 7

Bing He, Jia Li, Yifan Zhao, and Yonghong Tian. Part-regularized near-duplicate vehicle re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 3, 7

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 1, 6

Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017. 6

Shaoli Huang, Zhe Xu, Dacheng Tao, and Ya Zhang. Part-stacked cnn for fine-grained visual categorization. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 2, 3

Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, 2015. 3, 7, 8

Ruyi Ji, Longyin Wen, Libo Zhang, Dawei Du, Yanjun Wu, Chen Zhao, Xianglong Liu, and Feiyue Huang. Attention convolutional binary neural tree for fine-grained visual categorization. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 6, 7

Mahdi M Kalayeh, Emrah Basaran, Muhittin Gökmen, Mustafa E Kamasak, and Mubarak Shah. Human semantic parsing for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2

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 *IEEE International Conference on Computer Vision*, 2019. 7

Pirazh Khorramshahi, Neehar Peri, Jun-cheng Chen, and Rama Chellappa. The devil is in the details: Self-supervised attention for vehicle re-identification. *arXiv preprint arXiv:2004.06271*, 2020. 7

Jin-Hwa Kim, Jaehyun Jun, and Byoung-Tak Zhang. Bilinear attention networks. In *Advances in Neural Information Processing Systems*, 2018. 2, 3

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *IEEE International Conference on Computer Vision Workshops*, 2013. 1, 3, 7

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998. 3

Di Lin, Xiaoyong Shen, Cewu Lu, and Jiaya Jia. Deep lac: Deep localization, alignment and classification for fine-grained recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015a. 2, 3, 7

Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. Bilinear cnn models for fine-grained visual recognition. In *IEEE International Conference on Computer Vision*, 2015b. 7

Chih-Ting Liu, Man-Yu Lee, Tsai-Shien Chen, and Shao-Yi Chien. Hard samples rectification for unsupervised cross-domain person re-identification. *arXiv preprint arXiv:2106.07204*, 2021. 2

Hongye Liu, Yonghong Tian, Yaowei Wang, Lu Pang, and Tiejun Huang. Deep relative distance learning: Tell the difference between similar vehicles. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016a. 1, 2, 7

Xiaobin Liu, Shiliang Zhang, Qingming Huang, and Wen Gao. Ram: a region-aware deep model for vehicle re-identification. In *IEEE International Conference on Multimedia and Expo*, 2018. 7

Xinchen Liu, Wu Liu, Huadong Ma, and Huiyuan Fu. Large-scale vehicle re-identification in urban surveillance videos. In *IEEE International Conference on Multimedia and Expo*, 2016b. 1, 2, 7, 13

Xinchen Liu, Wu Liu, Tao Mei, and Huadong Ma. A deep learning-based approach to progressive vehicle re-identification for urban surveillance. In *European Conference on Computer Vision*, 2016c. 2, 7, 13

Yihang Lou, Yan Bai, Jun Liu, Shiqi Wang, and Ling-Yu Duan. Embedding adversarial learning for vehicle re-identification. *IEEE Transactions on Image Processing*, 2019. 7

Hao Luo, Youzhi Gu, Xingyu Liao, Shenqi Lai, and Wei Jiang. Bag of tricks and a strong baseline for deep person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019a. 1, 2, 6

Wei Luo, Xitong Yang, Xianjie Mo, Yuheng Lu, Larry S Davis, Jun Li, Jian Yang, and Ser-Nam Lim. Cross-x learning for fine-grained visual categorization. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019b. 7

Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 1, 7

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 *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 7

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011. 3

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 Workshop on Benchmarking Multi-Target Tracking*, 2016. 2

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1

Yifan Sun, Liang Zheng, Yi Yang, Qi Tian, and Shengjin Wang. Beyond part models: Person retrieval with refined part pooling (and a strong convolutional baseline). In *European Conference on Computer Vision*, 2018. 1, 2, 6

Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016. 4

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 7, 13

Yaming Wang, Vlad I Morariu, and Larry S Davis. Learning a discriminative filter bank within a cnn for fine-grained recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 7

Zhongdao Wang, Luming Tang, Xihui Liu, Zhuliang Yao, Shuai Yi, Jing Shao, Junjie Yan, Shengjin Wang, Hongsheng Li, and Xiaogang Wang. Orientation invariant feature embedding and spatial temporal regularization for vehicle re-identification. In *IEEE International Conference on Computer Vision*, 2017. 2, 7

Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. Caltech-ucsd birds 200. 2010. 1, 3

Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 6

Hantao Yao, Shiliang Zhang, Richang Hong, Yongdong Zhang, Changsheng Xu, and Qi Tian. Deep representation learning with part loss for person re-identification. *IEEE Transactions on Image Processing*, 2019. 2

Yue Yao, Liang Zheng, Xiaodong Yang, Milind Naphade, and Tom Gedeon. Simulating content consistent vehicle datasets with attribute descent. In *European Conference on Computer Vision*, 2020. 7

Chaojian Yu, Xinyi Zhao, Qi Zheng, Peng Zhang, and Xinge You. Hierarchical bilinear pooling for fine-grained visual recognition. In *European Conference on Computer Vision*, 2018. 7

Lianbo Zhang, Shaoli Huang, Wei Liu, and Dacheng Tao. Learning a mixture of granularity-specific experts for fine-grained categorization. In *IEEE International Conference on Computer Vision*, 2019. 1, 3, 6, 7

Ning Zhang, Ryan Farrell, Forrest Iandola, and Trevor Darrell. Deformable part descriptors for fine-grained recognition and attribute prediction. In *IEEE International Conference on Computer Vision*, 2013. 3

Ning Zhang, Jeff Donahue, Ross Girshick, and Trevor Darrell. Part-based r-cnns for fine-grained category detection. In *European Conference on Computer Vision*, 2014. 1, 3

Heliang Zheng, Jianlong Fu, Tao Mei, and Jiebo Luo. Learning multi-attention convolutional neural network for fine-grained image recognition. In *IEEE International Conference on Computer Vision*, 2017a. 7

Heliang Zheng, Jianlong Fu, Zheng-Jun Zha, and Jiebo Luo. Looking for the devil in the details: Learning trilinear attention sampling network for fine-grained image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 2, 7

Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *IEEE International Conference on Computer Vision*, 2015. 2, 3, 7

Zhedong Zheng, Liang Zheng, and Yi Yang. Unlabeled samples generated by gan improve the person re-identification baseline in vitro. In *IEEE International Conference on Computer Vision*, 2017b. 1, 2

Yi Zhou and Ling Shao. Viewpoint-aware attentive multi-view inference for vehicle re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2, 7

A Robustness Testing of Adaptive Downsampling Rate

In Equation 8, we show that the adaptive downsampling rate of height or width can be respectively derived by the standard deviation (abbreviated as std) of 1-D vertical or horizontal keypoints distribution. In this section, we further evaluate the stability and rationality of Equation 8 on the various simulated keypoints distributions. We comply with the following rules to simulate the distributions: 1) most of the keypoints locate in one consecutive interval; 2) the interval can have any length and can be anywhere in the entire distribution; 3) the distribution within the interval is random. These rules reflect the fact that the images for fine-grained recognition typically contain only one target object (e.g., a bird or a vehicle) with various scales and locations, and the critical features (e.g., a bird's beak or a vehicle's tires) are randomly distributed in the target object. Following these principles, we simulate the distributions by first assigning a random significance score at each location and filtering by a 1-D Gaussian kernel with a random mean and std. We plot the distributions and also show the cropped lengths and regions in Figure 7.

We can observe that for each keypoints distribution with different sparsity or at a different location, Equation 8 can consistently evaluate an appropriate adaptive downsampling rate that can cover most of the keypoints while removing the negligible background region with a relatively few keypoints. Subsequently, through the trade-off mechanism, users can actively adjust the scale of cropped region to better capture the interesting parts in the target object.

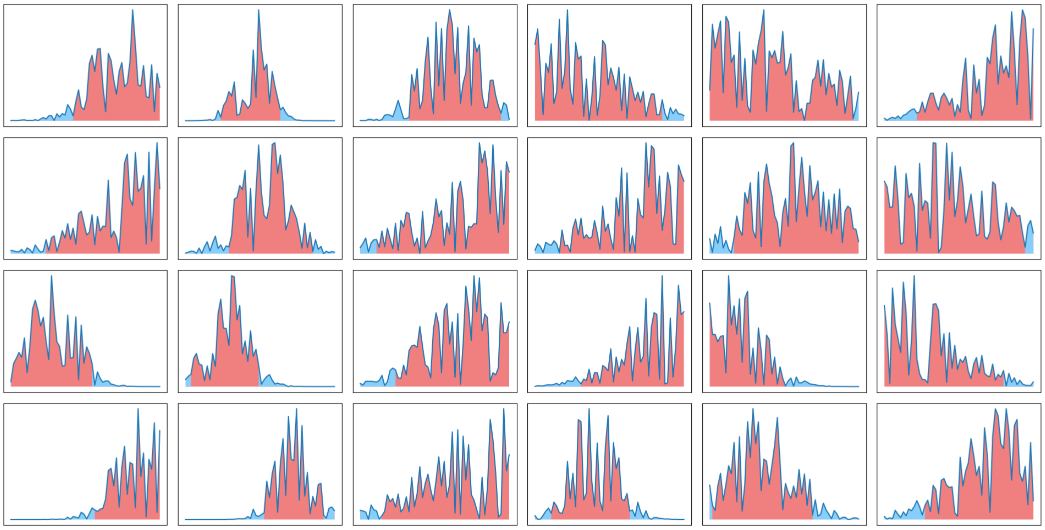

Figure 7: **Visualization of the cropped regions on the simulated keypoints distributions.** For each sub-figure, x-axis indicates different locations and y-axis represents the accumulated keypoint confidences at each location (i.e., Equation 5). The blue line is the simulated 1-D keypoints distribution while the color under the line means whether each location is within the cropped region or not (red means yes while blue means no).

B Multiple Scale and Granularity Network with More Local View Branches

In Section 4, we have shown the effectiveness of MSGN with one local view branch (LVB) and will further discuss the behavior of MSGN with more (i.e. 2 and 4) LVBs in the following. To avoid excessive overlaps among the sampled regions from different LVBs, we set a smaller $k = 0.5$ for Adaptive Region Pooling (ARP) in Conv4 to reduce the scale of cropped region. We evaluate the models on two datasets: CUB-200-2011 (Wah et al., 2011) and VeRi-776 (Liu et al., 2016b;c), and respectively report the visualization results and the validation performance in Figure 8 and Table 4.

We observe that, for MSGN with two LVBs, two branches can learn to focus on the scattered parts in the target object, such as the head and torso of a bird, and can jointly cover larger critical regions in contrast to the one-branch model (by the comparison with the visualization in Figure 5). Also, due to smaller k, the model can leverage the higher resolution of sub-sampled feature and the richer details in the cropped region. Therefore, as listed in Table 4, we can get a performance gain when we increase the number of LVB from one to two. However, there is no more improvement when we further apply four LVBs. A possible explanation is that the sampled regions from the two-branches model have already covered most of the parts in the interesting object; therefore, the extra LVBs would merely focus on the overlapping regions and provide redundant information.

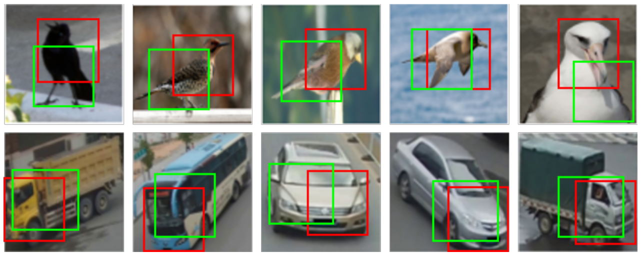

Figure 8: **Visualization of the cropped regions from two-branches MSGN.** The green and red boxes respectively represent the cropped regions by different branches of MSGN.

Table 4: **Effectiveness of MSGN with different number of LVBs.**

Method	Backbone	CUB	VeRi	
		Acc. (%)	mAP (%)	R-1 (%)
MSGN w/ 1 LVB	ResNet-50	89.1	81.8	97.3
MSGN w/ 2 LVB	ResNet-50	**89.5**	**82.1**	**97.8**
MSGN w/ 4 LVB	ResNet-50	**89.5**	82.0	97.5

C Training and Validating Details

The network is initialized with ImageNet pretrained weights. During the training phase of 100 epoch, we set the initial learning rate as 0.001 and adopt a linear warm-up strategy to linearly increase the learning rate to 0.01 in the first ten epochs; then, we use a multi-stage learning rate scheduler by multiplying the learning rate by 0.1 in 40^{th} and 70^{th} epoch. We use SGD optimizer with a weight decay of 0.0005 and momentum of 0.9. Considering that the images for vehicle re-identification are usually well-aligned, hence, we use original strided pooling as downsampling in Conv3. As for fine-grained image classification, during the inference phase, the input images are resized to 512×512 then center-cropped to 448×448. Our network is implemented in PyTorch and all experiments have been conducted on one NVIDIA Titan RTX GPU with 24GB memory.