Intra-class Part Swapping for Fine-Grained Image Classification

Lianbo Zhang*[1], Shaoli Huang*[2], and Wei Liu[1]

[1]Advanced Analytics Institute, School of Computer Science, FEIT, University of Technology Sydney,
Chippendale, NSW, Australia
[2]UBTECH Sydney AI Centre, School of Computer Science, FEIT, University of Sydney, Darlington,
NSW 2008, Australia
{lianbo.zhang@student., wei.liu@}uts.edu.au shaoli.huang@sydney.edu.au

Abstract

Recent works such as Mixup and CutMix have demonstrated the effectiveness of augmenting training data for deep models. These methods generate new data by generally blending random image contents and mixing their labels proportionally. However, this strategy tends to produce unreasonable training samples for fine-grained recognition, leading to limited improvement. This is because mixing random image contents may potentially produce images containing destructed object structures. Further, as the category differences mainly reside in small part regions, mixing labels proportionally to the number of mixed pixels might result in label noisy problem. To augment more reasonable training data, we propose Intra-class Part Swapping (InPS) that produces new data by performing attention-guided content swapping on input pairs from the same class. Compared with previous approaches, InPS avoids introducing noisy labels and ensures a likely holistic structure of objects in generated images. We demonstrate InPS outperforms the most recent augmentation approaches in both fine-grained recognition and weakly object localization. Further, by simply incorporating the mid-level feature learning, our proposed method achieves state-of-the-art performance in the literature while maintaining the simplicity and inference efficiency. Our code is publicly available[†].

1. Introduction

Deep neural networks have made enormous progress in many computer vision tasks such as object recognition [4, 10, 24], object detection [9, 22]. One inherent limitation of these neural networks is that they have tremendous parameters to learn, leading to overfitting and poor generalization. To alleviate this issue, a variety of training strategies such as data augmentation and regularization, have been proposed, among which mixing-based methods [28, 13, 39] have been recently demonstrated as a new direction to improve model generalization. The general strategy of these methods is to extend the training distribution by blending random image contents and mixing their labels proportionally. The augmented data significantly benefits generic object classification as it helps regularize deep neural networks in training.

However, their superiority might be undermined for fine-grained recognition. Unlike general object classification, fine-grained objects often share a common part structure, while mixing random contents tends to generate images with corrupted object structures. Therefore these methods might potentially generate training images that are not consistent with the data characteristics of the task. On the other hand, mixing labels according to the mixing ratio of image content will inevitably produce unfavorable label noise, as the semantic information is usually disproportionate to the number of image pixels. For example (Figure 1), there is label mixing in Mixup [39] and CutMix [38], of which CutMix produce new label based on the category area, which might lead to the noisy label. As illustrated in Figure 1, although *Eared Grebe* dominates ground-truth label, the output is visually more like *California Gull* to a human. It is also noted that, in the mixing process, Cutout and CutMix cause structure corruption, and Mixup combines two images unreasonably.

To remedy these limitations, we propose Intra-class Part Swapping (InPS) that imposes prior restrictions on both image contents and label pairs to be mixed. Specifically, InPS randomly selects input pairs from the same class and then constructs an attention pool to guide content swapping between two potential part regions. Compared with existing

[*] These authors contributed equally to this work.
[†] https://github.com/lbzhang/InPS.git

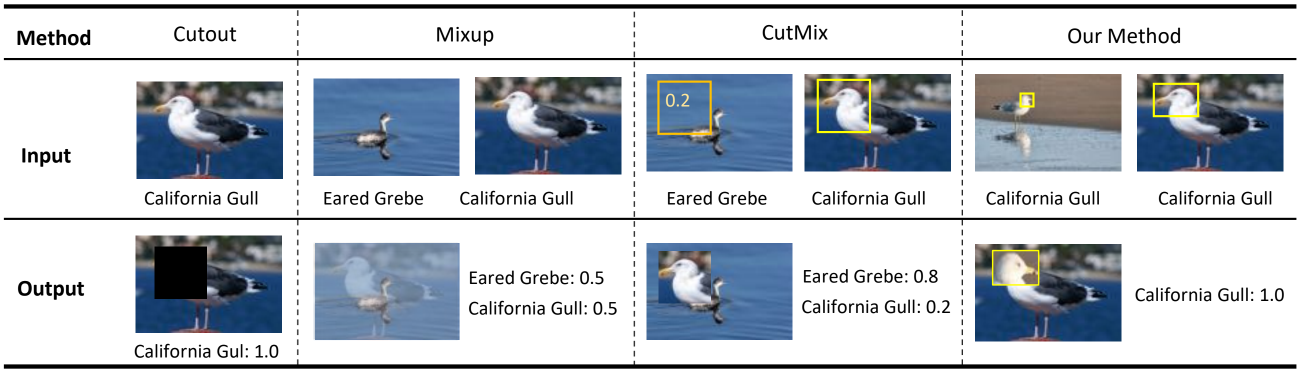

Figure 1. Comparison of Cutout, Mixup, CutMix, and the proposed method. Note that there is label mixing in MixUp and CutMix, and CutMix produces a new label based on category area. This might lead to noisy labels; for example, although Eared Grebe dominates ground-truth label, the output is visually more like California Gull to a human. In terms of object structure, Cutout and CutMix cause structure corruption; Mixup combines two input images unreasonably. Instead, our method generates more reasonable samples and clean supervision information.

mixup-based methods, InPS synthesis images without destructing too much object structure and avoid label noise, which is a promising solution to augment fine-grained training data.

We evaluate our method on fine-grained benchmarks including CUB-200-2011 [29], Stanford-Cars [16], and FGVC-Aircraft [20], and demonstrate superior performance over the most recent mixed-up based strategies. Furthermore, by simply incorporating mid-level features, our proposed method achieves state-of-the-art performance in the literature while maintaining simplicity and inference efficiency. Compared with mixing-based methods such as Mixup, Cutout, and Cutmix, our method also exhibits better performance in weakly supervised object localization, which indicates that InPS trains the neural networks to be more sensitive to the object integrity.

The rest of the paper is organized as follows. We discuss works related to our method in section 2 and present details of our method in section 3. Section 4 will report implementations and experiment results and conclude the paper in the last section.

2. Related Works

2.1. Mixing Regularization

Image mixing [28, 13, 39] is an effective augmentation strategy to regularize the training of neural networks. One simple way [13] to mix image is randomly picking two a pair of images from training data before synthesizing a new sample, where the pixel values of selected images are averaged in the new sample. In this work, the label of the first image is maintained as the supervised information when the synthetic image is fed through the network. Zhang *et al.* [39] extends the training distribution by linear interpolating both the input images and associated targets. This concept if further investigated by Summers and Dineen [25] who explores a more generalized form and considers a broader scope of non-linear mixing up. More recent works are RICAP [27] and CutMix [38]. Among them, RICAP randomly crops four images and concatenates them to construct a new sample for training. In CutMix, one patch of each image is cut and pasted among training data. Compared with Cutout [5], or randomly erasing [45], CutMix claims to make better use of the image information.

These researches tend to focus on random mixing but fail to consider the structural integrity of objects so that the network is not trained to make decisions from the global-level features. The additional problem is that the label mixing over the whole data leads to noisy labels. Because the uneven distribution of regional importance is neglected, pixel number based supervision of new synthetic samples becomes noisy, which will further confuse the neural networks if the attention signal is used to guide the mixing process. In contrast, the proposed method alleviates label noise by mixing images of positive samples. Since intra-class images are more likely to share similar responses, the object integrity is potentially preserved while images parts are swapped between positive samples.

2.2. Fine-Grained Classification

Researches for fine-grained image recognition [32, 12, 33, 36, 26] have focused on extracting diverse features from a single image by locating or sampling significant parts. To find object parts with specific semantic information, early works [12, 41, 34, 18] design extra part-location sub-network trained from bounding box and part annotations. Despite effective results benefiting from strong supervised information, the process of obtaining such anno-

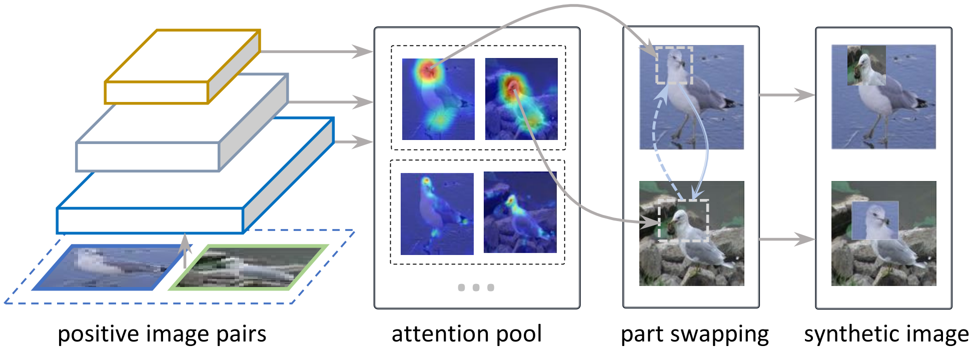

Figure 2. Overview of our network architecture. InPS takes positive image pairs as input and then construct an attention pool using multiple-level features. After that, an attention pair is randomly selected before deploying a threshold to determine attended parts, which are swapped to generate synthetic images.

tation is costly and not practical for many fine-grained data [16, 20]. Recent part-based methods [42, 43, 26, 6, 35] tend to use category labels to supervise model learning and develop a variety of attention techniques to find class-related parts. Inspired by the intuition that a convolutional filter can be treated as a certain visual pattern detector [42, 23], MA-CNN [43] clusters feature channels of convolutional layers to generate multiple parts. A similar idea is used by MAMC [26], which use Squeeze Excitation (SE) [11] mechanism with metric constrain to learn multiple attention regions. S3N [6] and TASN [44] consider regions with a high response in attention map as informative parts. The corresponding region in the image is re-sampled to highlight fine-grained details. This soft manner retains context while amplifying local regions, which alleviates the information loss from the hard part cropping strategy.

Most of these methods rely on a complex pipeline to extract fine-grained details. This leads to less efficient training and evolves as a limit to the study of attention-based methods. Rather than learning fine-grained information by designing complex network structures, we choose to reinforce the existing networks. In particular, feature extraction pipelines are simplified from multiple backbones to only one backbone. The proposed method can also be easily embedded into the existing networks to improve the model performance on fine-grained classification and localization without introducing extra resources.

3. Proposed Method

In this section, we describe the details of the proposed Intra-class Part Swapping (InPS). An overview of InPS can be found in Figure 2. The network recognizes objects mainly using features from the target object in a given image. Inspired by this, we functionally separate the image into two zones in the spatial dimension, the internal attended zone that contains critical information and the external preserving zone. Given the attended region, we aim to drive networks to understand the fine-grained object in more diverse contexts, where the network is supervised by correct ground-truth information. To achieve this goal, we need to solve the following two problems: (1) Given only image-level labels, how to define and obtain the determined zones. (2) How to preserve the supervised information when mixing images into new samples.

3.1. Attention Priors

Since we only have category supervision, it is difficult to obtain a precise object zone. In fact, we seek what is more attractive to obtain significant regions as attention priors. We feed the images to the network to obtain initial attention map M_a using Classification Activation Map (CAM) [46]. Then we threshold M_a with δ for a binary mask B_a. Regions with values larger than δ in M_a will be treated as the interested zone. Thus, we define the binary mask as follows: $B_{a,(i,j)} = 1$ if $M_{a,(i,j)} \geqq \delta$, and $B_{a,(i,j)} = 0$ otherwise. It is reasonable that the attention zone covers parts that contain semantic information of the target object.

Our architecture contains one shared backbone, from which the attention map can be generated from different layers. Take two attentions as an example, the corresponding two sub-networks are denoted by S_a and S_b, respectively. S_a and S_b differ in the number of convolutional layers and the pooling method before the linear classifier. However, both sub-networks start from a convolutional block, the goal of which is to determine initial attention.

By combining initial attention maps, we introduce the attention pool. Given the attention map M_a produced by S_a, separately. We can obtain binary masks B_a according to the above section. By sampling threshold δ from a specified distribution, the attention space is expanded to a potentially larger one. In the training step, we randomly sample attention pairs to guide the swapping operation.

3.2. Intra-class Part Swapping

Let (I_1, l_1) and (I_2, l_2) be an image pair sampled from the training set, where $l_1 = l_2$ and $I_1, I_2 \in \mathbb{R}^{3\times h\times w}$. To perform swapping between positive samples, the output region is computed by applying an affine transformation T (spatial scaling and translation) to the attended patch in the source image. Taking swapping a part from I_2 to I_1 as an example, the synthetic image, in this case, is calculated as,

$$\tilde{I}_1 = S(F(G(I_2),\ T_\theta),\ B_2 * I_2) + (1 - B_1) * I_1 \quad (1)$$

where the transformation function F is parameterized by an augmented matrix T_θ with size of 2×3 in 2D coordinate system. Since the affine transformation works on the coordination of pixels, a sampler S is used to grid sample transformed patch from old coordinates to a new one. Since we are only interested in spatial scaling and translation, the affine matrix can be simplified as

$$T_\theta = \begin{bmatrix} a_x & 0 & c_x \\ 0 & a_y & c_y \end{bmatrix} \quad (2)$$

where a_x, a_y are the scaling factor, and c_x, c_y are the bias in the coordinates of x and y. We further factorized T_θ into a scaling matrix $A = \begin{bmatrix} a_x & 0 \\ 0 & a_y \end{bmatrix}$ and a translation matrix $C = \begin{bmatrix} 1 & 0 & c_x \\ 0 & 1 & c_y \end{bmatrix}$. The affine matrix is converted to

$$T_\theta = \begin{bmatrix} 1 & 0 & c_x \\ 0 & 1 & c_y \end{bmatrix} \cdot \begin{bmatrix} a_x & 0 & 0 \\ 0 & a_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3)$$

Here, we denote the I_2 that provide the internal attended zone as the source image, and I_1 that provide external preserving zone as the target image. After applying a threshold to the selected attention map of I_2, the attended region is formulated as a rectangle box determined by its top-left point and bottom-right point with the location of (x_1^{tl}, y_1^{tl}) and (x_1^{br}, y_1^{br}) for I_1. Similarly, the corresponding points of I_2 are (x_2^{tl}, y_2^{tl}) and (x_2^{br}, y_2^{br}).

Given these two sets of anchors, we can directly obtain the scaling factor and translation factor without introducing extra parameters. We first determine the scaling matrix A_θ from I_2 to I_1, which are calculated as

$$a_x = \frac{x_1^{br} - x_1^{tl}}{x_2^{br} - x_2^{tl}}, \quad a_y = \frac{y_1^{br} - y_1^{tl}}{y_2^{br} - y_2^{tl}} \quad (4)$$

Specifically, assume that the (x, y) is the location of one pixel in selected patch of image I_2. After applying the scaling transformation, the coordinates are converted to

$$\begin{bmatrix} a_x & 0 \\ 0 & a_y \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} = x \begin{bmatrix} a_x \\ 0 \end{bmatrix} + y \begin{bmatrix} 0 \\ a_y \end{bmatrix} = \begin{bmatrix} a_x x \\ a_y y \end{bmatrix} \quad (5)$$

Next, we decide the translation factor c_x, c_y on the basis of scaled coordinates. After introducing an extra dimension 1 to the coordinate vector, the coordinate is denoted as $(a_x x, a_y y, 1)$. We then solve the following equation:

$$\begin{bmatrix} 1 & 0 & c_x \\ 0 & 1 & c_y \end{bmatrix} \cdot \begin{bmatrix} a_x x \\ a_y y \\ 1 \end{bmatrix} = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} \quad (6)$$

We get value of translation parameters c_x and c_y in the following form

$$c_x = a_x x_2 - x_1, \quad c_y = a_y y_2 - y_1 \quad (7)$$

To perform a swapping transformation between the image pair, a sampler mush take the set of sampling point $F(G(\cdot), T_\theta)$, along with the input image I_2 and produce the sampled output image $\tilde{I}_1$. Each (x, y) coordinate in $F(G(\cdot), T_\theta)$ defines the spatial location in the output where a grid sampler is applied to get the value at particular pixel in the input image. Denote the output of grid sampler as V, the sampling is then written as

$$V^{(T_\theta \cdot G(I_2))_{x,y}} = (B_2 * I_2)^{x,y}, \ \ \forall x \in [1 \dots w],\ y \in [1 \dots h] \quad (8)$$

Different from STN [14] that learn parameters to define the transformation matrix, the affine matrix in our method is directly calculated from the selected attention. By multiplying the affine matrix to the masked source image, we align the size and location of the source patch to the target location, which is then linearly combined with the external zone of the target image. The swapping operation is only used in the training stage, and during test time the network behaves the same as the backbone used. In practice, our such transformation is applied by transforming the grid of the target image size by T and interpolating the source image at the resulting coordinates.

InPS takes advantage of both intra-class swapping and attention signal. The plausible combination of internal zone and external zone from positive example creates a large context space, making it harder to overfit the fine-grained dataset. Local parts contribute differently when recognizing the object in a different context, with more contexts to explore, InPS understands the categories with better knowledge. This helps the network recognize fine-grained objects by accurately using information from more object parts, which, therefore, benefits localization capability. We note that Attentive CutMix[30] also introduced attention to mixing strategy, but our method is different. InPS is specifically designed for fine-grained tasks by performing image

Dataset	# Class	# Train	# Test	# Total
CUB-200-2011	200	5,994	5,794	11,788
Stanford-Cars	120	12,000	8,580	2,0580
FGVC-Aircraft	102	6,667	3,333	10,200

Table 1. Dataset Statistics of CUB-200-2011, Stanford-Cars and FGVC-Aircraft.

Method	Accuracy(%)
Random Mix (CutMix)	86.64
Positive Mix (CutMix + positive)	86.80
Positive Mix of Attention (InPS)	**87.56**

Table 2. Effectiveness of positive pair and attention pool on CUB-200-2011

mixing among samples of the same classes, which avoided label confusion in the learning process. Besides, the attentive patch to be swapped in our method covers the connected area, which maintains the integrity of discriminative parts.

4. Experiments

This section evaluates the performance of the proposed InPS method for fine-grained image classification and localization. We first introduce the benchmark datasets and implementation details of InPS. In weakly supervised localization, we compare InPS with mixing-based methods, including Mixup, Cutout, and CutMix. We report the superior performance of InPS compared with state-of-the-art approaches in the classification task.

4.1. Dataset

To verify the effectiveness of our proposed approach, we conduct experiments on three fine-grained datasets, namely CUB-200-2011 [29], Stanford-Cars [16], and FGVC-Aircraft [20]. Details about these three datasets are summarized in Table 1.

CUB-200-2011 dataset contains 11,788 bird images of 200 categories with roughly 30 training images per category. The dataset also contains 5994 instances as the training data and other 5794 as testing data. Each image in the dataset is annotated with a bounding box, part locations as well as attribute labels.

Stanford Car dataset contains 196 car categories for the fine-grained task. There are 8144 examples in the training set, and for the testing set, the data size is 8041, making 16,185 images in total in the dataset. Car images from the dataset are taken from various angles, and the categories are assigned based on production year and car model, e.g., *2012 Tesla Model S* or *2012 BMW M3 couple*.

FGVC-Aircraft dataset contains 102 aircraft categories with 100 images for each class making 10,200 images in the dataset. There are 6667 examples in the training set, and the testing data has a data size of 3333. The main aircraft in each image is annotated with a bounding box and a hierarchical airplane model label.

We compared our method with baselines methods that only use category labels without additional data.

4.2. Implementation Details

For our intra-class part swapping network, we use ResNet-50 with ImageNet pre-trained weights as our base model, open-sourced PyTorch [21] as our code-base and trained all models on $1 \times$V100 GPU. We use stochastic gradient descent (SGD) as the optimizer. The initial learning rate of new layers is set to be 0.01 while the learning rate for the pre-trained layer is reduced by one-tenth. The batch size is set to 10. We train our model for 100 epochs while decaying the learning rate by multiplying 0.1 at 40^{th}, 70^{th}, 90^{th} epoch. We report the results using the model from the last epoch.

We take different augmentation strategies for three datasets. For CUB-200-2011, during training time, we first augment images by randomly resized to 512 along the shorter side while keeping the image from deforming, then we crop the images to size 448×448 with randomly horizontal flipping. We also resize the test image using the same method while only performing center cropping to size 448×448. For Stanford-Car and FGVC-Aircraft, we augment the training images by first resizing them to 512×512, then random crop to size 448×448 as the input. Test image are also resize to 512×512 before central cropping to size 448×448 for recognition. The reason we do this is that part shape is more important for bird recognition when performing local feature extraction, and if we resize the input to a square, the local parts are deformed, which reduces the local information diversity. This augmentation strategy is used for all experiments in this paper, including classification and localization.

We briefly describe the settings for baseline augmentation schemes. Cutout [5] requires to fix mask size, following setting used in [38, 5], we set the mask size to be half of the image size 224×224, and the dropping out location is uniformly sampled. Similarly, for CutMix [38] and Mixup [39] we set the mixing probability to be 0.5.

During inference time, for weakly supervised localization, we resize the input images to a fixed size and then resize the resulting attention map back to the original resolution. We use the last convolutional layer to generate the attention map for weakly supervised localization.

4.3. Intra-Class Attention Analysis

In Table 2, we consider reporting the results by adding positive swapping strategy and attention signal separately

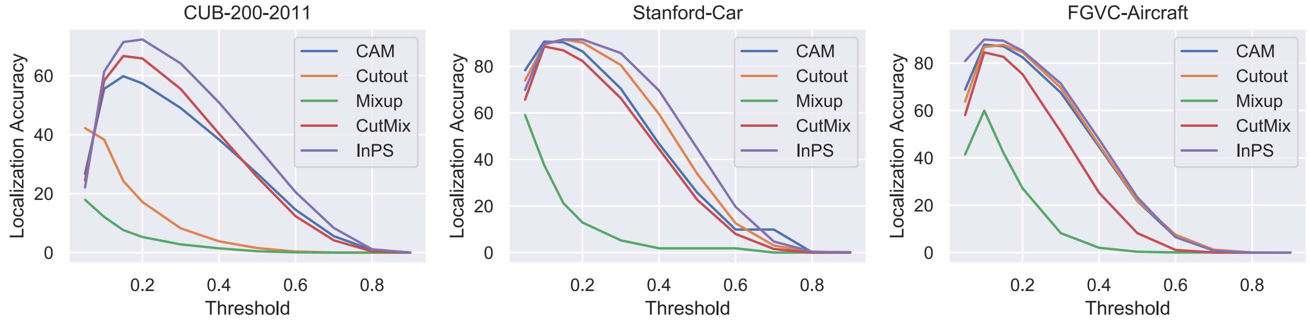

Figure 3. Weakly localization comparison under different threshold σ on CUB-200-2011, Stanford-Cars and FGVC-Aircraft.

Method	Localization Accuracy(%)					
	CUB-200-2011		Stanford-Car		FGVC-Aircraft	
	best	mean	best	mean	best	mean
ResNet50 + CAM	59.82	30.42	90.61	45.54	87.82	42.53
ResNet50 + Mixup	17.92	4.34	59.08	12.60	59.95	16.50
ResNet50 + Cutout	42.20	12.36	91.46	48.60	86.83	42.62
ResNet50 + CutMix	66.67	32.15	90.16	50.62	84.55	35.13
ResNet50 + InPS	**72.28**	**37.10**	**91.52**	**51.54**	**90.01**	**45.05**

Table 3. Weakly supervised object localization comparison of state-of-the-art mixing-image approaches on CUB-200-2011, Stanford-Card, and FGVC-Aircraft.

Method	Classification Accuracy(%)		
	CUB	Cars	Aircraft
ResNet-50	85.92	93.51	91.69
Mixup	86.28	92.90	91.27
Cutout	83.41	93.78	91.51
CutMix	86.64	93.96	92.14
InPS(ours)	**87.56**	**94.59**	**92.65**

Table 4. Classification comparison of baseline(ResNet-50) and state-of-the-art augmentation methods (Mixup, Cutout, CutMix) on CUB-200-2011, Stanford-Cars, and FGVC-Aircraft.

to the CutMix method. Note that the proposed InPS can be treated as adding these two techniques to the CutMix. According to Table 2, one can observe that with the positive swapping added to the CutMix, the positive CutMix achieves better performance with 0.2% improvement. By further adding the attention signal to guide the swapping process, our method achieves the best results. Compared with original CutMix, we have a performance gain of 0.92% in terms of top-1 classification accuracy, which reflects the high quality of the fine-grained representation produced by our approach.

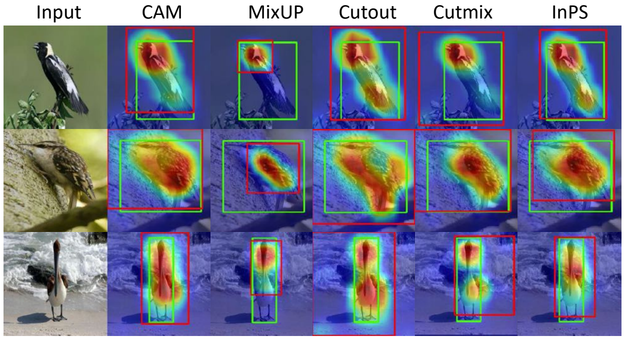

Figure 4. Qualitative comparison of the baseline (ResNet-50), Mixup, Cutout, CutMix and InPS for weakly supervised object localization task on CUB-200-2011 dataset. Ground truth and predicted bounding boxes are denoted as green and red, respectively.

4.4. Weakly Supervised Localization

Weakly supervised localization methods aim to localize objects using category labels. To measure the localization accuracy of models, the Intersection-Over-Union(IOU) between the estimated bounding box and the ground-truth positive is larger than 0.5, and, at the same time, the estimated class label should be correct. Otherwise, the localization accuracy treats the estimation is wrong. A good model in this task tries to find more diverse responses in target objects as

Method	Backbone	Accuracy(%)		
		CUB-200-2011	Stanford-Cars	FGVC-Aircraft
RA-CNN [7]	3 × VGG-19	85.3	92.5	88.2
RAM [17]	3 × Resnet-50	86.0	-	
S3N [6]	3 × Resnet-50	88.5	94.7	92.8
MGN-CNN [40]	3 × Resnet-50	88.5	93.9	-
STN [14]	5 × Inception	84.1	-	-
MA-CNN [43]	5 × VGG-19	86.5	91.5	89.9
NTS-Net [37]	5 × Resnet-50	87.5	93.9	91.4
B-CNN [19]	1 × VGG-16	84.1	91.3	84.1
Compact B-CNN [8]	1 × VGG-16	84.0	-	-
Low-rank B-CNN [15]	1 × VGG-16	84.2	90.9	87.3
Kernel-Activation [1]	1 × VGG-16	85.3	91.7	88.3
Kernel-Pooling [3]	1 × VGG-16	86.2	92.4	86.9
DFL-CNN [31]	1 × VGG-16	86.7	93.8	92.0
MAMC [26]	1 × Resnet-101	86.5	93.0	-
ResNet-50	1 × Resnet-50	86.1	93.2	91.3
DFL-CNN [31]	1 × Resnet-50	87.4	93.1	91.7
DCL [2]	1 × Resnet-50	87.8	94.5	93.0
TASN [44]	1 × Resnet-50	87.9	93.8	-
Mixup [39]	1 × Resnet-50	87.80	94.14	92.35
Cutout [5]	1 × Resnet-50	86.74	94.74	92.23
CutMix [38]	1 × Resnet-50	87.88	94.64	92.77
InPS(ours)	1 × Resnet-50	88.82	94.96	93.76
InPS(ours)	1 × Resnet-101	**89.23**	**95.03**	**94.06**

Table 5. Performance comparison with state-of-the-art methods on CUB200-2011, Stanford-Cars and FGVC-Aircraft.

well as correctly recognize the category. InPS creates more meaningful combinations of object parts so that the network is trained to understand the fine-grained object from detail to the whole. We follow the existing strategy [38] to evaluate the localization capability on the fine-grained benchmarks. We compared the proposed method with various data-augmentation techniques: Mixup, Cutout, CutMix. Meanwhile, all implementation details follow the classification setting and all input sizes to models are 448×448. The classification activation map (CAM) is used to estimate the bounding box by applying a threshold on it.

Table 3 quantitative evaluates the best and average localization results. The threshold is set between 0.05 and 0.9 as we notice that the localization accuracy becomes 0 when the threshold is larger than 0.9. From the table, we consistently observe that InPS outperforms the baseline method CAM and all data-augmentation methods that are based on image-level supervision. It is worthy to note that Mixup poses a negative impact on localization ability, leading to a poor localization performance on all three datasets. This is because Mixup encourages the network to focus on the smaller region as shown in Figure 4. Although CutMix makes better use of pixels than Cutout, it potentially shares a similar issue as Cutout, where irrelevant areas might also be activated. This limits further improvement in localization performance. The problem is alleviated by InPS, which focuses on attended regions swapping, thus reducing interference from the background. In Table 3, InPS improves the localization accuracy from 59.82% to 72.28%, which shows that InPS is helpful to learning correct regions.

The proposed method also exhibits robustness to the threshold. From Figure 3, we can see that our method achieves the best performance under a high threshold, and the metric declines slower than competitors. One reason for this is that InPS finds more diverse part representation, the response of which is strong enough to maintain influence even when the threshold increases to a high value. This can be further verified by Table 3, where InPS achieves comparable mean localization accuracy on all three benchmarks against other data-augmentation methods.

4.5. Fine-Grained Classification

To evaluate the performance of the proposed method in fine-grained classification, we include the middle-level feature in the network. The middle-level information has been proved to be effective in learning complementary rep-

Method	Classification Accuracy(%)		
	CUB	Cars	Aircraft
ResNet50	84.40	91.52	90.04
Mixup	85.76	93.75	91.75
Cutout	85.95	93.35	91.18
CutMix	86.24	**93.86**	91.81
InPS(ours)	**86.69**	93.58	**92.26**

Table 6. Performance of middle-level representation on CUB200-2011, Stanford-Cars and FGVC-Aircraft.

resentation for fine-grained recognition [31, 40] at a low cost. In particular, we add (`Conv1×1 - Max Pooling - Linear Classifier`) after 4^{th} block of ResNet to learn middle-level feature. Since the feature from the 4^{th} block is detached before being fed to the new sub-network, gradients of the new branch will not propagate back to the backbone network, thus not affect the training of the existing network (GAP branch in Table 4). We further illustrate the detailed performance of the new branch in Table 6. Noted that middle-level feature shows strong representative ability on fine-grained datasets. This capability can also be enhanced using mixing strategies by different amplitudes. Compared with baseline, the proposed InPS improve the model accuracy by more than 2% on all three datasets.

The fine-grained classification is evaluated by the top-1 classification accuracy(%). As shown in Table 5, our model significantly outperforms the ResNet-50 baseline (fine-tune from the ImageNet) by 2.8%, 1.8%, and 2.4% on three challenging datasets respectively, which shows the ability of our InPS to learn good representation from fine-grained images. Table 5 also reporting results with state-of-the-art approaches. In particular, compared with DFL-CNN [31] which enhances mid-level representation learning within the CNN framework by learning a bank of convolutional filters to capture class-specific discriminative patches, we get a better result with a relative accuracy improvement of 1.4%. Our method outperforms MAMC [26] which uses metrics to learn multiple attention region features by 2.3%. Although our baseline is already strong, the improvement with a large margin indicates that a better representation can still be learned even with a deeper network. It is noted that mixing strategies also boost the fine-grained classification. Although not necessarily works on all benchmarks, CutMix achieves better performance than all existing fine-grained methods. We also get the best performance on Stanford-Cars (94.96%) and FGVC-Aircraft(93.76%). By using ResNet-101 as a strong feature extractor, the model performance on three benchmarks can be further improved, achieving 89.23%, 95.03%, and 94.06% respectively.

α β	0.5 0.5	0.5 1.0	1.0 1.0	2.0 2.0	2.0 5.0	5.0 2.0
Acc(%)	87.3	87.6	87.6	86.9	87.4	86.7

Table 7. Performance comparison in terms of classification accuracy (Acc) under different α, β on CUB-200-2011 dataset.

4.6. Ablation Study

In this section, we conduct ablation studies to understand the design of the proposed InPS method.

Determination of α and β. We use a beta distribution from which we randomly sample a threshold for each sample. There are two hyper-parameters in beta distribution, α and β, and by setting different values, we can sample from different distributions. We report experimental results of regularizing the baseline network (ResNet-50) guided by high-level attention on the CUB-200-2011 dataset, illustrated in Table 7. Overall, the proposed method fluctuates depending on the two hyper-parameter values, and we use $\alpha = 1.0$, $\beta = 1.0$ throughout the paper.

Model complexity analysis. Since the proposed InPS is designed to augment the existing network, no extra parameters are introduced to the baseline network. The network is efficient to train, and the model also takes the same number of iterations as the baseline to converge. During the testing time, the same backbone network is used. Compared with ResNet-50, our method is 1.6% better and after 2.9% better, introducing middle-level features without extra time cost.

5. Conclusion

In this paper, we presented Intra-class Part Swapping (InPS) for fine-grained recognition. In particular, InPS performs attention-guided swapping on positive samples. In this way, InPS avoids inter-class mixing, thus alleviating label noise in the mixing process. Besides, using the attention signal to guide the swapping between significant regions created reasonable combinations, eliminating the potential structure of new samples. Experiments demonstrated that InPS consistently outperforms the recent augmentation approaches on both fine-grained classification and weakly-supervised localization. Compared with the the-state-of-art fine-grained methods, InPS achieved superior performance in computational efficiency, accuracy, and simplicity. We believe InPS can be further applied to augment the low-level feature, further saving computational resources.

6. Acknowledgements

This work was supported by the Australian Laureate Fellowship project FL170100117.

References

[1] S. Cai, W. Zuo, and L. Zhang. Higher-order integration of hierarchical convolutional activations for fine-grained visual categorization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 511–520, 2017.

[2] Y. Chen, Y. Bai, W. Zhang, and T. Mei. Destruction and construction learning for fine-grained image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5157–5166, 2019.

[3] Y. Cui, F. Zhou, J. Wang, X. Liu, Y. Lin, and S. Belongie. Kernel pooling for convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*.

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[5] T. DeVries and G. W. Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

[6] Y. Ding, Y. Zhou, Y. Zhu, Q. Ye, and J. Jiao. Selective sparse sampling for fine-grained image recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6599–6608, 2019.

[7] J. Fu, H. Zheng, and T. 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*, pages 4438–4446, 2017.

[8] Y. Gao, O. Beijbom, N. Zhang, and T. Darrell. Compact bilinear pooling. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 317–326, 2016.

[9] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[11] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[12] S. Huang, Z. Xu, D. Tao, and Y. Zhang. Part-stacked cnn for fine-grained visual categorization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1173–1182, 2016.

[13] H. Inoue. Data augmentation by pairing samples for images classification. *arXiv preprint arXiv:1801.02929*, 2018.

[14] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025, 2015.

[15] S. Kong and C. Fowlkes. Low-rank bilinear pooling for fine-grained classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7025–7034, 2017.

[16] J. Krause, M. Stark, J. Deng, and L. Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 554–561, 2013.

[17] Z. Li, Y. Yang, X. Liu, F. Zhou, S. Wen, and W. Xu. Dynamic computational time for visual attention. 2017.

[18] D. Lin, X. Shen, C. Lu, and J. Jia. Deep lac: Deep localization, alignment and classification for fine-grained recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1666–1674, 2015.

[19] T.-Y. Lin, A. RoyChowdhury, and S. Maji. Bilinear cnn models for fine-grained visual recognition. In *IEEE International Conference on Computer Vision*, pages 1449–1457, 2015.

[20] S. Maji, E. Rahtu, J. Kannala, M. Blaschko, and A. Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

[21] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. 2017.

[22] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[23] M. Simon and E. Rodner. Neural activation constellations: Unsupervised part model discovery with convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1143–1151, 2015.

[24] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations*, 2015.

[25] C. Summers and M. J. Dinneen. Improved mixed-example data augmentation. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1262–1270. IEEE, 2019.

[26] M. Sun, Y. Yuan, F. Zhou, and E. Ding. Multi-attention multi-class constraint for fine-grained image recognition. *European Conference on Computer Vision*, 2018.

[27] R. Takahashi, T. Matsubara, and K. Uehara. Data augmentation using random image cropping and patching for deep cnns. *IEEE Transactions on Circuits and Systems for Video Technology*, 2019.

[28] Y. Tokozume, Y. Ushiku, and T. Harada. Between-class learning for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5486–5494, 2018.

[29] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[30] D. Walawalkar, Z. Shen, Z. Liu, and M. Savvides. Attentive cutmix: An enhanced data augmentation approach for deep learning based image classification. *arXiv preprint arXiv:2003.13048*, 2020.

[31] Y. Wang, V. I. Morariu, and L. S. Davis. Learning a discriminative filter bank within a cnn for fine-grained recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4148–4157, 2018.

[32] X.-S. Wei, P. Wang, L. Liu, C. Shen, and J. Wu. Piecewise classifier mappings: Learning fine-grained learners for novel categories with few examples. *IEEE Transactions on Image Processing*, 28(12):6116–6125, 2019.

[33] X.-S. Wei, C.-W. Xie, J. Wu, and C. Shen. Mask-cnn: Localizing parts and selecting descriptors for fine-grained bird

species categorization. *Pattern Recognition*, 76:704–714, 2018.

[34] T. Xiao, Y. Xu, K. Yang, J. Zhang, Y. Peng, and Z. Zhang. The application of two-level attention models in deep convolutional neural network for fine-grained image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 842–850, 2015.

[35] G.-S. Xie, X.-Y. Zhang, W. Yang, M. Xu, S. Yan, and C.-L. Liu. Lg-cnn: From local parts to global discrimination for fine-grained recognition. *Pattern Recognition*, 71:118–131, 2017.

[36] Z. Xu, S. Huang, Y. Zhang, and D. Tao. Webly-supervised fine-grained visual categorization via deep domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(5):1100–1113, 2016.

[37] Z. Yang, T. Luo, D. Wang, Z. Hu, J. Gao, and L. Wang. Learning to navigate for fine-grained classification. In *European Conference on Computer Vision*, pages 420–435, 2018.

[38] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019.

[39] H. Zhang and M. Cisse. Ynddl-p. 2018. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*.

[40] L. Zhang, S. Huang, W. Liu, and D. Tao. Learning a mixture of granularity-specific experts for fine-grained categorization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8331–8340, 2019.

[41] N. Zhang, J. Donahue, R. Girshick, and T. Darrell. Part-based r-cnns for fine-grained category detection. In *European conference on computer vision*, pages 834–849. Springer, 2014.

[42] X. Zhang, H. Xiong, W. Zhou, W. Lin, and Q. Tian. Picking deep filter responses for fine-grained image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1134–1142, 2016.

[43] H. Zheng, J. Fu, T. Mei, and J. Luo. Learning multi-attention convolutional neural network for fine-grained image recognition. In *IEEE international conference on computer vision*, pages 5209–5217, 2017.

[44] H. Zheng, J. Fu, Z.-J. Zha, and J. Luo. Looking for the devil in the details: Learning trilinear attention sampling network for fine-grained image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5012–5021, 2019.

[45] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang. Random erasing data augmentation. *arXiv preprint arXiv:1708.04896*, 2017.

[46] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2921–2929, 2016.