Stochastic Partial Swap: Enhanced Model Generalization and Interpretability for Fine-grained Recognition

Shaoli Huang[1], Xinchao Wang[2], Dacheng Tao[3,1]
[1]The University of Sydney, Australia, [2]National University of Singapore,
[3]JD Explore Academy, China
shaoli.huang@sydney.edu.au, xinchao@nus.edu.sg, dacheng.tao@gmail.com

Abstract

Learning mid-level representation for fine-grained recognition is easily dominated by a limited number of highly discriminative patterns, degrading its robustness and generalization capability. To this end, we propose a novel Stochastic Partial Swap (SPS)[1] scheme to address this issue. Our method performs element-wise swapping for partial features between samples to inject noise during training. It equips a regularization effect similar to Dropout, which promotes more neurons to represent the concepts. Furthermore, it also exhibits other advantages: 1) suppressing over-activation to some part patterns to improve feature representativeness, and 2) enriching pattern combination and simulating noisy cases to enhance classifier generalization. We verify the effectiveness of our approach through comprehensive experiments across four network backbones and three fine-grained datasets. Moreover, we demonstrate its ability to complement high-level representations, allowing a simple model to achieve performance comparable to the top-performing technologies in fine-grained recognition, indoor scene recognition, and material recognition while improving model interpretability.

1. Introduction

Fine-grained recognition is more challenging than general object recognition, as the discriminative differences of categories often reside in subtle parts of objects. Conventional methods that succeed in generic object classification, therefore, often fail to deliver gratifying results in fine-grained classification, since they mainly focus on learning high-level features and overlook subtle variations. Existing works [54, 8, 55, 9, 56, 7, 35, 55, 28, 44, 53, 37, 20] attempt to complement this capacity by exploring various techniques. Part-based [54, 8, 55, 9] and sampling-based [56, 7] are the most popular solutions in recent literature. The former primarily localize part regions by strongly-supervised detection pipelines or weakly-supervised learning framework, and then extract the discriminative local features as complementary to high-level features. The sampling-based approaches seek to enrich the representation learning by conducting attention sampling over the input images. Although these two techniques have succeeded in improving performance, they either require complex training procedures or intense computation in inference, limiting their applicability to real-world situations.

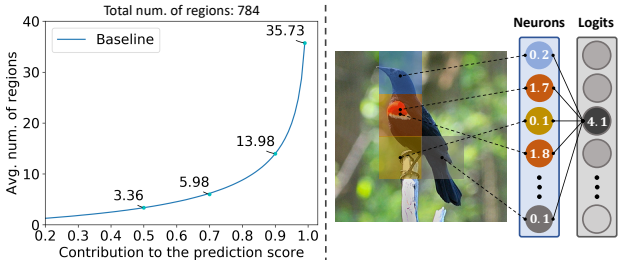

Figure 1. The left column shows the average number of regions per image contributing to a given predicted value, when using the baseline model trained on the CUB dataset. A more detailed description of this experiment is given in the Experiment section. The right column illustrates an issue of adopting deep mid-level models for fine-grained classification. In training data, the "Red-yellow patch" pattern may distinguish the "Red-winged blackbird" from most bird species. During training, the neural network tends to associate this part mostly to this label by biasing its weights to the corresponding region. The resulting mid-level model would predict whether an object is a "Red-winged blackbird" mainly based on this pattern while largely ignoring other roles of other patterns.

Incorporating deep mid-level models into fine-grained recognition has demonstrated its potential in recent endeavours [35, 22, 52, 23], due to its unique merits. First, mid-level models are easy to obtain and flexible to exploit, thanks to the hierarchical structure of deep neural networks. Second, they also exhibit a strong capability to capture lo-

[1]https://github.com/Shaoli-Huang/SPS.

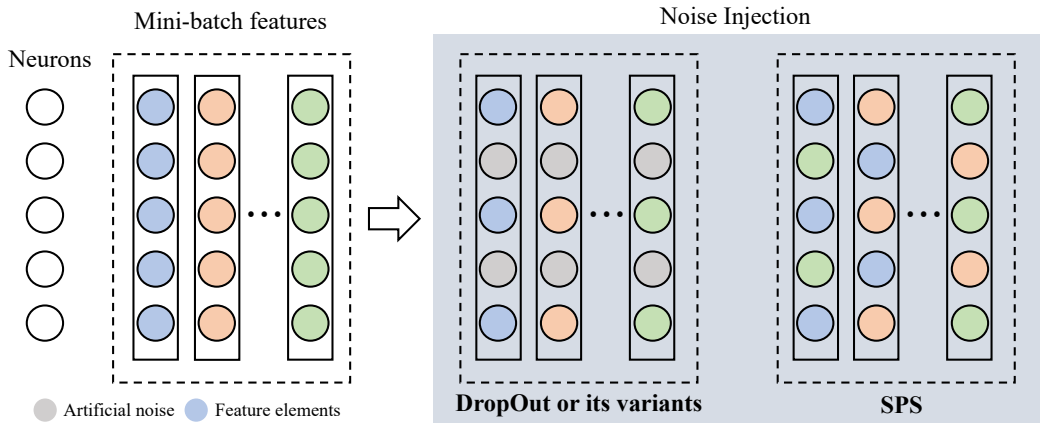

Figure 2. An illustration of the difference between the Dropout techniques and SPS. Dropout or its variants mainly inject manually-designed noise into features, while SPS exploits samples as a source for noise injection.

cal information and serve as a critical complement to the high-level representation approach in fine-grained recognition. The works of [22, 52] showcase that coupling mid-level and high-level classification models indeed leads to enhanced performance.

Despite the promising results, prior approaches have been merely adopting off-the-shelf mid-level model in a plug-and-play fashion rather than enhancing the mid-level model *per se*. In this paper, we make one step forward along this line, and strive to learn better mid-level representations for fine-grained recognition. We observe that a mid-level model determines the label primarily based on a small number of image regions. As shown in the left column of Fig. 1, for a baseline model on the CUB-200-2011 training dataset, on average the top 35.73 of the total 784 regions per image, in fact, contribute to 99% of the final prediction scores. We speculate that, this is because some subtle object parts exhibit extremely powerful discriminability in the training set, and thus the neural network bias its weight more toward these few patterns. For example, as illustrated in the right column of Fig. 1, a "Red-yellow patch" pattern is very distinguishable for Red-winged blackbird. In this case, the neural network tends to learn more neurons highly responsive to this pattern, making it dominantly contribute to the prediction. The resulting model will be therefore dominated by a limited number of part patterns, degrading its robustness and generalization ability.

To this end, we propose a novel Stochastic Partial Swap (SPS) strategy to enhance the generalization of mid-level models. The swapping-noise strategy randomly selects one sample feature as a noise source during training and swaps some of its feature units into the corresponding locations of another sample. Our proposed method differs from existing injection methods in exploiting sample features as a noise source (illustrated in Fig.2). This strategy delivers several advantages in learning mid-level representation. First, if most of the swapped-in elements are inactive neurons, our method has a similar regularization effect of DropOut, which encourages more neurons for feature representation. Second, our approach helps suppress some neurons that dominate the predictions. For instance, if some neurons with dominant roles in predicting one category are swapped into one sample feature of another class, they may cause the sample to be mispredicted. In this case, the cost function will penalize these neurons for their misleading influence. Last but not least, this strategy comes with augmentation abilities to enhance the classifier's robustness. For example, swapping partial features between the intra-class samples will allow the classifier to see more pattern combination of the class. Also, exchanging partial neurons between the inter-class instances produces a sample feature that contains noisy patterns of another category.

We extensively evaluate our method on seven datasets across three different tasks. Experiments show that our approach improves the performance of the baseline by a large margin. Our learned mid-level model obtains an accuracy of 87.29% on the CUB dataset and outperforms other regularization methods and even the high-level model. By incorporating the high-level representation, our approach with simplicity and efficiency further achieves state-of-the-art or comparable performance on CUB-200-2011, Aircraft, Stanford Cars, Food101, MIT indoor, and GTOS datasets.

2. Related Work

Fine-grained recognition plays a crucial role in various image and video tasks [25, 46, 34, 43, 32, 45]. In what follows, we give a brief review of domains related to fine-grained recognition.

Fine-grained Recognition. Despite remarkable suc-

cess in generic classification, high-level representation approaches based on deep neural networks fail to achieve satisfactory fine-grained recognition performance. This primary because it cannot capture the subtle visual difference yet is critical to fine-grained recognition. Therefore, various methods [54, 8, 55, 42, 9, 56, 7, 35, 41, 55, 28, 44, 53, 37, 20] have been proposed to explore a complementary representation to the high-level models. These methods can be divided into two main categories: part-based and mid-level representation. The first mainly exploit a way to localize part regions and consequently learn a part-based representation. Typical approaches to learn a part detector involves using strongly-supervised learning [53, 13] or weakly-supervised learning [54, 8, 55, 28]. Recent works in this group attempt to learn a richer part representation by employing dense part sampling [44, 56] or sparse-part-sampling [7]. Although part-based methods have shown great success in improving the classification performance, they require either strongly-annotated training data, which is costly to obtain, or the need to design more sophisticated frameworks to inference part regions.

Exploiting mid-level representation has attracted increasing attention in this field [35, 22, 52]. [35] introduced a cross-channel pooling layer to improve the discriminative ability of the mid-level model. [52] directly incorporated the mid-level with high-level features to build a strong expert network. [22] exploited the spatial relation between the mid-level and high-level features to learn robust multi-scale features. These works relate to our work mostly. However, they mainly embedded the mid-level model into their methods. In contrast, this paper investigates the mid-level model's learning problem and further explores a noise perturbation strategy to address the issue.

Noise Injection Methods. Due to the massive learning parameters, deep neural networks usually suffer from overfitting, which necessitates regularization methods. Our proposed approach relates to noise regularization techniques [50, 27, 24, 1, 19, 48]. Classic methods, including Dropout and its variants, mainly inject noises during training by adding or multiplying noise. For example, Dropout randomly drops neurons during training, Gaussian Dropout [27] multiplies the feature units by Gaussian random noise. Compared with these methods that inject manually-defined noise to feature vectors, our proposed scheme generates noise features by exploiting partial elements of other sample as noise sources, which can effectively inhibit some neurons from expressing overconfidence for a specific category. It also provides a more reasonable way to mimic real data noise for improving the robustness of the classifier.

Mixing-based Data Augmentation. Recent mixing-based methods [50, 48, 12, 51] show impressive performance by combining images and further fusing the labels accordingly. Our work differs from these works on mixing augmentation from two aspects. First, typical mixing-based methods (such as Mixup and CutMix) augment the training data distribution by providing vicinal samples of a data point. By contrast, our proposed method is motivated to solve the problem that *mid-level* representations concentrate on a few patterns. Second, although our work is inspired by sample mixing, our work differs from these works in method design and working mechanism due to different motivations. The mixing-based methods focus on generating vicinal samples (by blending images and mixing the corresponding labels) to make the input distribution smoother for training the neural networks. By comparison, we are concerned about how to inject noise into features (by swapping feature units of samples) to ensure the neuron representation involves more patterns in making predictions.

3. Approach

In this section, we first describe a common practice to learn a deep mid-level representation and introduce our proposed method SPS in the following subsection.

3.1. Learning Deep Mid-level Representation

Standard practice for fine-grained recognition is to finetune a backbone network on the target training set. The resulting model extracts features by applying global average pooling over the last convolutional layer, therefore capturing high-level information of objects. However, recent works [35, 22, 52] have demonstrated that the mid-level representation exhibits strong complementary ability to the high-level representation in fine-grained recognition. Although their method designs for learning mid-level models are different, they all share some standard practices. Therefore, based on these approaches' implementation detail, we sum up a simple learning framework as a baseline. As illustrated in Fig.3, the baseline plugs a mid-level classification branch to a standard classification framework. This new branch's feature block contains a 1x1 convolutional layer initialized with random weights, a ReLU activation layer, and a Global Max Pooling layer. During training, the two branches are trained jointly. It is also worth mentioning that we block gradients passing from the mid-level component to the backbone layers until the later training stage. This training practice helps stabilize the training procedure and prevents the intermediate layers from learning toward abstracting high-level information.

3.2. Stochastic Partial Swap

Although the mid-level representation can supplement the high-level representation to improve accuracy, the performance remains unsatisfactory when used solely. This might because the learned mid-level model tends to be dominated by a small number of highly discriminative patterns

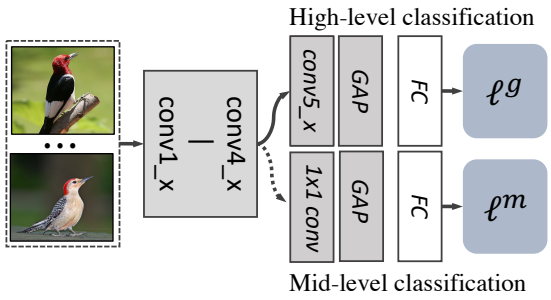

Figure 3. A baseline framework to learn mid-level representation. Here, ℓ^g and ℓ^m are both cross-entropy losses for the mid-level and high-level classification branches, respectively.

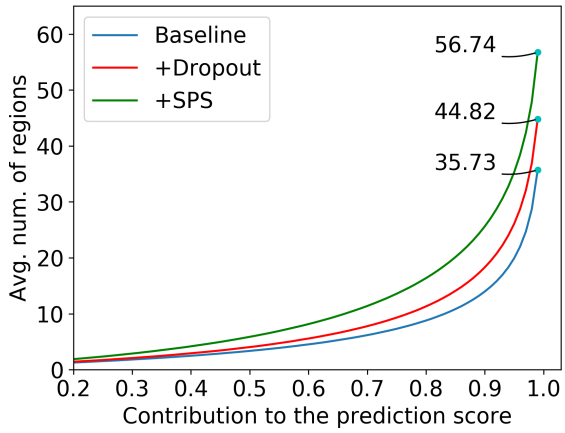

Figure 4. Comparison of SPS with baseline and dropout in how many regions they mainly rely on to determine the label on CUB-200-2011 training set.

degrading its robustness and generalization capability. A naive solution to address this problem is to apply dropout to the feature layer. However, since the dropout manly erases some neurons to zeros, it does not suppress some over-confident neurons (with highly high activation values) in training, as the gradients passing to the erased neurons will be zero. Therefore, Although dropout encourages the neural network to activate more neurons to represent a concept, the resulting representation might still be dominated by some highly activated neurons. To remedy this situation, we propose a Stochastic Partial Swap (SPS) that performs element-wise swapping for partial features between samples to inject noise in training the neural network. This training strategy delivers more advantages over existing noise injection methods. First, it provides a way to allow gradients to suppress the overconfident neurons. For example, when a neuron is highly activated to one sample yet its corresponding feature value is injected into another instance from a different category, the neural network would produce a significant error in classifying the injected sample and penalize the neuron's excessive activation. It also serves as a better way to mimic real noise data for training the classifier, as it injects real activation values of one sample into another sample instead of artificial noise. In the following, we describe in detail our proposed method.

Noise Injection. The main idea of SPS is to inject partial features of one sample into those of another example. For each sample (feature vector) in the mini-batch, we first randomly select another sample from the same mini-batch as a noise source and swap their partial feature elements element-wise. Given a sample x_i and a randomly selected sample x_j from the same mini-batch, the noise injecting operation can be expressed as:

$$\tilde{f}^m_{\rho\sim U(\alpha,\beta)}(x_i) = M \odot f^m(x_i) + (1-M) \odot f^m(x_j), \quad (1)$$

where U stands for uniform distribution with two parameters α and β, $\odot$ denotes element-wise multiplication, $f^m(\cdot)$ refers to feature extractor of mid-level branch, and $M \in \mathbb{R}^{dim(f^m(x))}$ denotes a binary mask. Here, we generated M based on the drawn value ρ that specifies the proportion of the number of feature elements to be exchanged, that is

$$M[k] = \begin{cases} 1 & rand(0,1) \leq \rho \\ 0 & rand(0,1) > \rho \end{cases}, \quad (2)$$

where $k \in [0, dim(f^m(x)) - 1]$ is the dimension index.

Training loss. As mentioned above, we inject noise by exchanging some feature units, which helps prevent the neural network from over-focusing on a small number of discriminative patterns. In addition, this strategy can also provide samples that simulate noise patterns, thereby enhancing the robustness of the classifier against noise. To amplify these properties, we also apply the noise inject operation multiple times to yield more perturbed cases of each sample within a mini-batch. Thus, the training loss for a single training instance (x_i, y_i) is defined as:

$$\sum_{t=1}^{T} \ell(C^m(\tilde{f}^m_{\rho\sim U(\alpha,\beta)}(x_i)), y_i) + \lambda \ell(C^g(f^g(x_i)), y_i), \quad (3)$$

where $\ell(\cdot)$ is the cross-entropy loss, T denotes number of times the noise injection is applied to a sample, $C(\cdot)$ refers to the classifier, and $f^g(\cdot)$ is the feature extractor of the high-level branch. Here, the regularization term is omitted for simplicity. It is also worth mentioning that the noise injection operation will be disabled in testing.

4. Experiments

4.1. Experimental Setup

Datasets. We conduct experiments on seven datasets across three different tasks: Caltech-UCSD Birds-200-2011(CUB-200-2011) [31], Stanford Cars [18] FGVC Aircrafts [23],

Datasets	Method	ResNet-50	ResNet-101	DenseNet-121	InceptionV3
CUB-200-2011	H-baseline	85.15	86.28	85.85	84.71
	M-baseline	82.87	84.83	83.12	84.55
	SPS	**87.29**	**87.65**	**87.11**	**86.93**
Stanford-Car	H-baseline	93.09	93.16	91.77	92.53
	M-baseline	90.42	92.06	89.75	91.13
	SPS	**94.35**	**94.22**	**93.63**	**93.48**
FGVC Aircrafts	H-baseline	91.05	91.74	90.49	90.28
	M-baseline	89.61	90.54	89.68	90.34
	SPS	**92.31**	**92.32**	**92.21**	**92.04**

Table 1. Performance comparisons with baselines using different network backbones. The value in the bracket indicates the performance improvement of our method relative to the M-baseline.

Food-101 [2], NABirds [30], MIT67 [26], and Ground Terrain in Outdoor Scenes(GTOS) [39]. The first four datasets are widely used to evaluate the performance of fine-grained classification methods. NABirds is a large-scale fine-grained dataset containing 555 categories. MIT67 is a benchmark dataset for indoor scene recognition. GTOS is a dataset of ground materials in the outdoor scene introduced recently for material classification. For each dataset, we use the provided train and test splits for all the experiments. If not specified, we resize images to 512x512 and then crop to 448x448 and train all models only using the class labels using standard data augmentation practices, including random cropping and flipping. In testing, we use a center-cropped image.

Backbone Networks and Baselines. In our experiments, we evaluate our method based on four network backbones including ResNet-50 [10], ResNet-101 [10], InceptionV3 [29], and DenseNet-121 [11]. For each backbone, we construct two baselines and term them as *H-baseline* and *M-baseline* throughout the rest of the paper.

H-baseline is a standard fine-tuning method, in which the backbone's last layer replaced with a new layer and then fine-tune the network on the target datasets.

M-baseline has been recently explored in recent works [44, 52] aiming to learn a mid-level representation for fine-grained recognition. We construct the M-baseline by inserting another classification branch into the intermediate layer of the H-baseline network. As suggested in [52], the new branch consists of a *1x1 conv layer*, a *ReLU layer*, a *global Max Pooling (GMP) layer*, and a *fully connected layer*. Specifically, for the ResNet and DenseNet structure, we used the output of the penultimate Conv block (residual or dense block) as the input to the new branch . In terms of the InceptionV3, since it contains a auxiliary classification branch placed on top of the intermediate layer, we simply replace the this branch with the new branch.

Training Details. We used pre-trained weights on Imagenet to initialize the backbone networks. The initial learning rate was set to 0.001 for the pre-trained layers and 0.01 for the rest layers. We then trained the networks for 160 epochs using an SGD optimizer with a learning rate decay factor of 0.1 for every 40 epochs. We used center-cropped images as inputs and obtained the average accuracy of the last five epochs as the performance measure in testing.

Method	Parameter	Accracy(%)
M-baseline	-	82.87
+Dropout	p = 0.5	85.48
+AlphaDropout	p = 0.4	84.58
+Mixup	$\lambda \sim Beta(0.2, 0.2)$	85.53
+Cutmix	$\lambda \sim Beta(1, 1)$	85.71
SPS	$\rho = 0.4$	86.65
SPS	$\rho \sim U(0.3, 0.5)$	87.29

Table 2. Evaluations and comparisons to regularization methods on CUB-200-2011 dataset.

Quantifying the contribution of image regions to the prediction In order to further verify the observed issue about learning the mid-level representation, we counted how many regions of the image that the learned model mainly use to make predictions. Next, we describe how to quantify an image region's contribution to the prediction. Given an image I with the label y, we denote its intermediate output feature maps $F \in \mathbb{R}^{d\times a\times b}$, where d and (a, b) are the channels number and the output size respectively. Each spatial location index $i \in [1, a \times b]$ of F corresponds to an image region R_i,which makes a total number of $a \times b$ regions for the image. Suppose we have the resulting mid-level feature $f \in \mathbb{R}^d$ and the classifier weight vector $w \in \mathbb{R}^d$ corresponding to the label y, then the output logit $o = w^t \cdot max(F)$, where $max(\cdot)$ refers to the global max pooling operation. Now, we calculate the contribution of the region R_i by

$$C(R_i) = \sum_{\substack{k\in[1,d] \\ argmax(F^k)=i}} (max(F^k) \times w^k)/o, \qquad (4)$$

where F^k and w^k corresponds to the k^{th} channel of F and

the k^{th} element of w respectively. Finally, given a percentage value, we sort the regions' contribution values in descending order and then continuously accumulate regions to determine at least how many areas can contribute this percentage to the output.

The hyperparameters of comparing methods. We compared our method with several representative regulation methods including Dropout [27], AlphaDropout [15], Mixup [50], and Cutmix [48]. We applied these methods to the mid-level baseline branch and reported their results with optimal hyperparameter for each method.

Complementary capabilities for classification. To verify the complementary ability of the learned mid-level representation to the high-level representation, we fuse the output logits from the mid-and high-level classification branch and test the performance. Since the high-level branch is used as an auxiliary classifier in learning SPS, obtaining such a fused output does not require an additional training process. For the following reported results, we use **High-level + SPS** to represent this way of using the combined output to make predictions. We also tested the complementary ability when using multiple SPS modules. We found the two SPS modules perform the best, this is because adding more SPS modules will bias the fused prediction towards the mid-level representation. Thus, we employ two SPS modules for experiments in this setting and used the same hyperparameters for all datasets.

4.2. Results and Analysis

SPS involves more regions to make prediction. We compared our SPS with the M-baseline and DropOut in how many regions they mainly rely on to determine the label. Here, we used the backbone network Resnet-50 and training dataset CUB-200-2011. For each image, we first computed the region contribution score described in the section 4.1 and counted the least number of regions that can contribute 99% to the prediction. Then we obtained the average number over the training data and showed the result in Fig. 4. We can find that, for the baseline model, on average, there are 35.73 (out of 784) regions per image can contribute 99% of the prediction score. This result shows that the resulting mid-level model tends to rely on a minimal number of part patterns in inferencing the label. By comparison, our proposed method SPS increases the number to 56.74, which demonstrates it can promote more regions to jointly contribute to the prediction.

SPS improves generalization performance on test data. To verify the effectiveness of learning mid-level representation, we tested our method on four network backbones and three fine-grained datasets. As shown in Table 1, our method considerably outperforms the baselines consistently regardless of different datasets or network structures. In terms of learning mid-level representation, SPS outperforms the M-baseline by a large margin. In addition, we can observe that the accuracy improvement is more significant on CUB dataset. This may be explained by that this data set has a small number of training images and contains some highly distinguishable parts, making the middle-level model more likely to be dominated by a few regions. We also compared our method with different regularization methods and showed the results in Table 2. We can see that all regularization methods improve the mid-level model, while our proposed method performs the best in performance improvement.

SPS exhibits strong complementary ability. To verify the complementary ability of the learned mid-level representation, we fused outputs from the SPS and the high-level classification branch for final prediction. As shown in Table 3, the combined model (considerably outperform the H-baseline which demonstrated the learned mid-level representation can complement the high-level representation in terms of classification capacity. For instance, the performance of using a single high-level representation is 85.15% on the CUB dataset, yet the accuracy can be improved to 88.42% when coupled with an SPS module. Moreover, Table 4 shows that this superiority also holds for large-scale datasets.

Fine-grained classification. Table 3 shows the performance comparison of different techniques of fine-grained recognition. Our proposed method achieved comparable or better performance than other approaches. It is worth mentioning that our method only used a single backbone network both in training and testing, while top-performing methods require multiple backbone feature extractions. Both S3N [7] and MGE-CNN [52] need three network feedforward passes to produce prediction, which requires larger GPU memory and more expensive computation.

Indoor scene and Outdoor ground material recognition. We evaluated our method on the MIT indoor and GTOS dataset. As shown in Table 5. the technique [17] achieves the best result. However, it may benefit from using pretrained weights on the Place205 dataset, which has similar domain information with the MIT indoor dataset. Nonetheless, compared to the latest methods [21, 47] using the same experimental settings, our method performs the best. We also conduct experiments on a new challenging dataset GTOS introduced for material classification. The results also reveal that our approach using a single scale in both training and testing outperforms all the comparing methods.

4.3. Visualization and model interpretation

Filter visualization. To gain an intuitive understanding of the superiority of our method, we visualized and compared the top *5* filters learned by the M-baseline and the SPS. Fig. 5 shows the visualization results. First, we can observe that the baseline model's top 5 filters are mostly activated in

Method	Backbone	Accracy(%)			
		CUB-200-2011	Stanford-Car	Aircraft	Food101
Low-rank B-CNN [16]	1xVGG-16	84.2	90.9	87.3	-
GP-256 [36]	1xVGG-16	85.8	92.8	89.8	85.7
MA-CNN [55]	3xVGG-19	86.5	91.5	89.9	-
Kernel-Pooling [5]	1xVGG-16	86.2	92.4	86.9	85.5
MAMC [28]	1xResnet-50	86.5	93.0	-	-
DFL-CNN [35]	1xResnet-50	87.4	93.1	91.7	-
NTS-Net [44]	3xResnet-50	87.5	93.9	91.4	-
DCL [3]	1xResnet-50	87.8	94.5	**93.0**	-
TASN [56]	1xResnet-50	87.9	93.8	-	-
Cross-X [22]	1xResnet-50	87.7	94.6	92.6	-
S3N [7]	3xResnet-50	88.5	94.7	92.8	-
MGE-CNN [52]	3xResnet-50	88.5	93.9	-	-
H-baseline	1xResnet-50	85.15	93.09	91.05	87.32
H-baseline + M-baseline	1xResnet-50	87.32	93.78	91.48	88.19
SPS	$\frac{4}{5}$xResnet-50	87.29	94.35	92.31	87.65
H-baseline + SPS	1xResnet-50	88.42	94.65	92.55	89.33
H-baseline + SPS*	1xResnet-50	**88.70**	**94.93**	92.73	**89.70**

Table 3. Comparison of different techniques on four fine-grained datasets. Here,$n \times backbone$ means the method requires n forward pass of the backbone network in testing, while $\frac{4}{5}$xResnet-50 indicates the first four of five Conv blocks of Resnet-50 is needed. SPS* denote results obtained by using two mid-level branches.

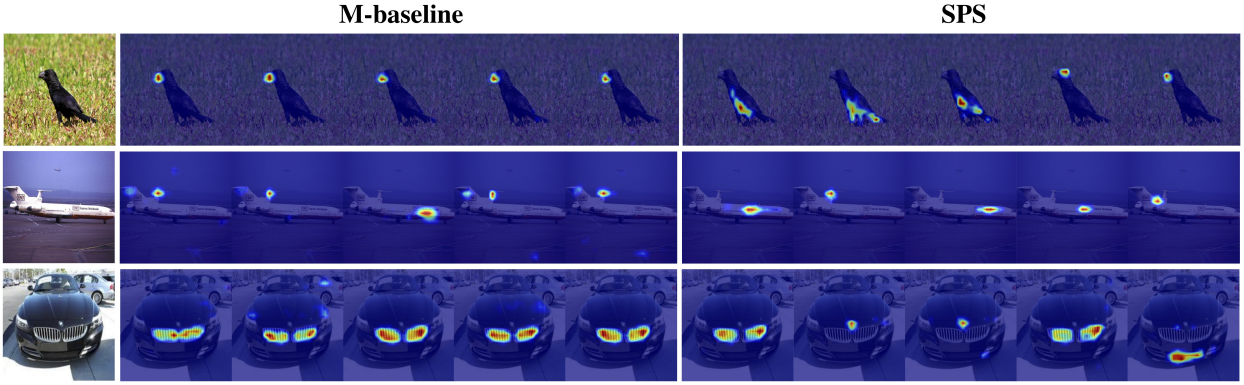

Figure 5. Visualization and comparison of the top 5 filters learned by the baseline method and the proposed method. (Better view zoomed in and with color.)

Method	ResNet-50	ResNet-101
H-baseline	84.43	85.36
M-baseline	82.85	85.21
SPS	86.11	86.92
H-baseline + M-baseline	86.10	86.97
H-baseline + SPS	87.11	87.85

Table 4. Result on the large-scale fine-grained dataset NABirds

the same part regions, while ours are distributed in different regions. This reveals that our approach effectively reduces the predominance of some certain strong patterns and encourages using more diverse patterns to represent the input. Also, the baseline method may be easily influenced by noisy patterns. For instance, the 2^{th} row shows the baseline model detects a tail style, a noisy pattern that is accidentally constructed by combining the tail from another aircraft.

Model interpretation. Unlike Global Average Pooling-based methods that provide model explanation using global visual cues, our approach is based on Global Max Pooling and thereby can provide a more detailed interpretation (subtle-part importance) of the model prediction. Besides, our method also enhances model interpretability by encouraging attention to more part regions and thus a more complete presentation. We introduce the following method for our model interpretation. Given an image and a feature map location index i, we first compute the region contribution score $C(R_i)$ as described in Sec. 4.1. Next, we generate a

	Method	Backbone	Input Size	Accracy(%)
MIT indoor	Places-205 [33] *	VGG16	224	81.2
	Spectral Features [14] *	VGG16	224	84.3
	SOP+SC+SigmE [17] *	Resnet-50	336	86.3
	Deep Filter Banks [4]	VGG19	224	81.0
	FASON [6]	FSON	448	81.7
	SMSO [47]	Resnet-50	448	79.7
	λ democratic [21]	Resnet-101	448	84.3
	H-baseline + SPS(Ours)	Resnet-50	448	83.1
	H-baseline + SPS(Ours)	Resnet-101	448	**84.6**
GTOS	Deep Filter Banks [4]	VGG19	240	77.1
	Multiview DAIN [40]	Resnet-50	240	81.4
	DeepTEN [38]	Resnet-50	ms	84.5
	MAP-net [49]	Resnet-50	224	84.7
	H-baseline + SPS(Ours)	Resnet-50	224	**85.6**

Table 5. Evaluations and comparisons to the state of the art on MIT indoor and GTOS dataset. (Methods marked with * use model pretrained on the Places dataset, others on the Imagenet datast)

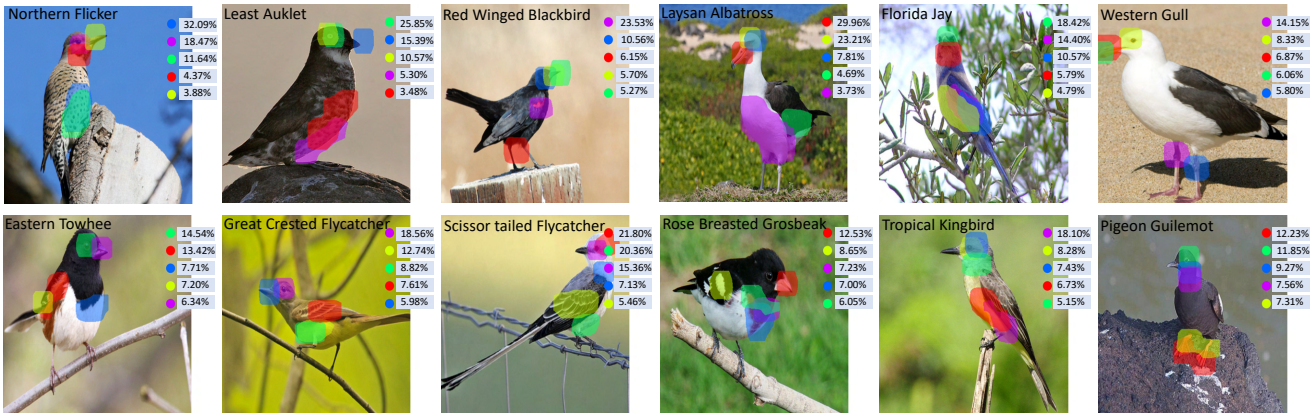

Figure 6. Some model interpretation results. For each image, we show the predicted class name, the top-5 important part regions with different color masking, and the corresponding contribution scores. (Better view zoomed in and with color.)

Local Class Activation Map (LCAM) by averaging CAMs whose max indices are i. Here, the LCAM represents the attention of certain filters that are most activated at the same location, which is useful to locate a part region. Finally, we sort the region contribution scores and show the top-k contribution scores and the corresponding regions. Fig. 6 provides some interpretation results. We can see that our method provides a comprehensible interpretation of what part regions are most important for model prediction. Taking the first picture as an example, we can understand that the model predicts it as *Northern Flicker* mainly according to some distinguishable patterns in areas such as the beak, the upper breast, the lower breast, and belly.

5. Conclusion

By statistics, we observed that the deep mid-level model has an issue that only a tiny percentage of image regions mainly contribute to the prediction. We thereby presented a Stochastic Partial Swap method to address the issue. Our main idea is to utilize real features as noises to disturb another feature during training. We demonstrated that this strategy effectively promotes the neural network to rely on more regions in making a prediction. We also showed its superiority in enhancing model generalization and interpretability. Despite these advantages, SPS is not directly applicable to Global Pooling Based models. In future work, we will explore a more general version of SPS for representation learning, and generalize SPS to wilder domains such as person Re-identification, and few-shot learning.

6. Acknowledgements

Dr. Shaoli Huang is supported by ARC FL-170100117. Xinchao Wang is supported by the Start-up Fund of National University of Singapore.

References

[1] Jimmy Ba and Brendan Frey. Adaptive dropout for training deep neural networks. In *Advances in neural information processing systems*, pages 3084–3092, 2013.

[2] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *ECCV*, pages 446–461. Springer, 2014.

[3] Yue Chen, Yalong Bai, Wei Zhang, and Tao Mei. Destruction and construction learning for fine-grained image recognition. In *CVPR*, pages 5157–5166, 2019.

[4] Mircea Cimpoi, Subhransu Maji, and Andrea Vedaldi. Deep filter banks for texture recognition and segmentation. In *CVPR*, pages 3828–3836, 2015.

[5] Yin Cui, Feng Zhou, Jiang Wang, Xiao Liu, Yuanqing Lin, and Serge Belongie. Kernel pooling for convolutional neural networks. In *CVPR*.

[6] Xiyang Dai, Joe Yue-Hei Ng, and Larry S Davis. Fason: First and second order information fusion network for texture recognition. In *CVPR*, pages 7352–7360, 2017.

[7] Yao Ding, Yanzhao Zhou, Yi Zhu, Qixiang Ye, and Jianbin Jiao. Selective sparse sampling for fine-grained image recognition. In *ICCV*, pages 6599–6608, 2019.

[8] Jianlong Fu, Heliang Zheng, and Tao Mei. Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition. In *CVPR*, pages 4438–4446, 2017.

[9] Weifeng Ge, Xiangru Lin, and Yizhou Yu. Weakly supervised complementary parts models for fine-grained image classification from the bottom up. In *CVPR*, pages 3034–3043, 2019.

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

[11] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[12] Shaoli Huang, Xinchao Wang, and Dacheng Tao. Snapmix: Semantically proportional mixing for augmenting fine-grained data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1628–1636, 2021.

[13] Shaoli Huang, Zhe Xu, Dacheng Tao, and Ya Zhang. Part-stacked cnn for fine-grained visual categorization. In *CVPR*, pages 1173–1182, 2016.

[14] Salman H Khan, Munawar Hayat, and Fatih Porikli. Scene categorization with spectral features. In *ICCV*, pages 5638–5648, 2017.

[15] Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. Self-normalizing neural networks. In *NIPS*, pages 971–980, 2017.

[16] Shu Kong and Charless Fowlkes. Low-rank bilinear pooling for fine-grained classification. In *CVPR*, pages 7025–7034, 2017.

[17] Piotr Koniusz, Hongguang Zhang, and Fatih Porikli. A deeper look at power normalizations. In *CVPR*, pages 5774–5783, 2018.

[18] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013.

[19] Zhe Li, Boqing Gong, and Tianbao Yang. Improved dropout for shallow and deep learning. In *Advances in neural information processing systems*, pages 2523–2531, 2016.

[20] Di Lin, Xiaoyong Shen, Cewu Lu, and Jiaya Jia. Deep lac: Deep localization, alignment and classification for fine-grained recognition. In *CVPR*, pages 1666–1674, 2015.

[21] Tsung-Yu Lin, Subhransu Maji, and Piotr Koniusz. Second-order democratic aggregation. In *ECCV*, pages 620–636, 2018.

[22] 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 *ICCV*, pages 8242–8251, 2019.

[23] S. Maji, J. Kannala, E. Rahtu, M. Blaschko, and A. Vedaldi. Fine-grained visual classification of aircraft. Technical report, 2013.

[24] Dmitry Molchanov, Arsenii Ashukha, and Dmitry Vetrov. Variational dropout sparsifies deep neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2498–2507. JMLR. org, 2017.

[25] Jiayan Qiu, Yiding Yang, Xinchao Wang, and Dacheng Tao. Hallucinating visual instances in total absentia. In *ECCV*, 2020.

[26] Ariadna Quattoni and Antonio Torralba. Recognizing indoor scenes. In *CVPR*, pages 413–420. IEEE, 2009.

[27] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *JMLR*, 15(1):1929–1958, 2014.

[28] Ming Sun, Yuchen Yuan, Feng Zhou, and Errui Ding. Multi-attention multi-class constraint for fine-grained image recognition. *ECCV*, 2018.

[29] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[30] Grant Van Horn, Steve Branson, Ryan Farrell, Scott Haber, Jessie Barry, Panos Ipeirotis, Pietro Perona, and Serge Belongie. Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 595–604, 2015.

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

[32] Jue Wang, Shaoli Huang, Xinchao Wang, and Dacheng Tao. Not all parts are created equal: 3d pose estimation by modelling bi-directional dependencies of body parts. In *ICCV*, 2019.

[33] Limin Wang, Sheng Guo, Weilin Huang, and Yu Qiao. Places205-vggnet models for scene recognition. *arXiv preprint arXiv:1508.01667*, 2015.

[34] Xinchao Wang, Engin Turetken, Francois Fleuret, and Pascal Fua. Tracking interacting objects using intertwined flows. *TPAMI*, 38:2312–2326, 2016.

[35] Yaming Wang, Vlad I Morariu, and Larry S Davis. Learning a discriminative filter bank within a cnn for fine-grained recognition. In *CVPR*, pages 4148–4157, 2018.

[36] Xing Wei, Yue Zhang, Yihong Gong, Jiawei Zhang, and Nanning Zheng. Grassmann pooling as compact homogeneous bilinear pooling for fine-grained visual classification. In *ECCV*, pages 355–370, 2018.

[37] Tianjun Xiao, Yichong Xu, Kuiyuan Yang, Jiaxing Zhang, Yuxin Peng, and Zheng Zhang. The application of two-level attention models in deep convolutional neural network for fine-grained image classification. In *CVPR*, pages 842–850, 2015.

[38] Jia Xue, Hang Zhang, and Kristin Dana. Deep texture manifold for ground terrain recognition. In *CVPR*, pages 558–567, 2018.

[39] Jia Xue, Hang Zhang, Kristin Dana, and Ko Nishino. Differential angular imaging for material recognition. In *CVPR*, pages 764–773, 2017.

[40] Jia Xue, Hang Zhang, Kristin Dana, and Ko Nishino. Differential angular imaging for material recognition. In *CVPR*, pages 764–773, 2017.

[41] Yiding Yang, Zunlei Feng, Mingli Song, and Xinchao Wang. Factorizable graph convolutional networks. In *NeurIPS*, volume 33, 2020.

[42] Yiding Yang, Jiayan Qiu, Mingli Song, Dacheng Tao, and Xinchao Wang. Distilling knowledge from graph convolutional networks. In *CVPR*, pages 7074–7083, 2020.

[43] Yiding Yang, Zhou Ren, Haoxiang Li, Chunluan Zhou, Xinchao Wang, and Gang Hua. Learning Dynamics via Graph Neural Networks for Human Pose Estimation and Tracking. In *CVPR*, 2021.

[44] Ze Yang, Tiange Luo, Dong Wang, Zhiqiang Hu, Jun Gao, and Liwei Wang. Learning to navigate for fine-grained classification. In *ECCV*, pages 420–435, 2018.

[45] Jingwen Ye, Yixin Ji, Xinchao Wang, Kairi Ou, Dapeng Tao, and Mingli Song. Student Becoming the Master: Knowledge Amalgamation for Joint Scene Parsing, Depth Estimation, and More. In *CVPR*, 2019.

[46] Xiaoqing Yin, Xinchao Wang, Jun Yu, Maojun Zhang, Pascal Fua, and Dacheng Tao. FishEyeRecNet: A Multi-Context Collaborative Deep Network for Fisheye Image Rectification. In *ECCV*, 2018.

[47] Kaicheng Yu and Mathieu Salzmann. Statistically-motivated second-order pooling. In *ECCV*, pages 600–616, 2018.

[48] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019.

[49] Wei Zhai, Yang Cao, Jing Zhang, and Zheng-Jun Zha. Deep multiple-attribute-perceived network for real-world texture recognition. In *ICCV*, pages 3613–3622, 2019.

[50] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018.

[51] Lianbo Zhang, Shaoli Huang, and Wei Liu. Intra-class part swapping for fine-grained image classification. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3209–3218, 2021.

[52] Lianbo Zhang, Shaoli Huang, Wei Liu, and Dacheng Tao. Learning a mixture of granularity-specific experts for fine-grained categorization. In *ICCV*, pages 8331–8340, 2019.

[53] Ning Zhang, Jeff Donahue, Ross Girshick, and Trevor Darrell. Part-based r-cnns for fine-grained category detection. In *ECCV*, pages 834–849. Springer, 2014.

[54] Xiaopeng Zhang, Hongkai Xiong, Wengang Zhou, Weiyao Lin, and Qi Tian. Picking deep filter responses for fine-grained image recognition. In *CVPR*, pages 1134–1142, 2016.

[55] Heliang Zheng, Jianlong Fu, Tao Mei, and Jiebo Luo. Learning multi-attention convolutional neural network for fine-grained image recognition. In *ICCV*, pages 5209–5217, 2017.

[56] 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 *CVPR*, pages 5012–5021, 2019.