Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 1245924, 10 pages
https://doi.org/10.1155/2020/1245924

Research Article

PLANET: Improved Convolutional Neural Networks with Image Enhancement for Image Classification

Chaohui Tang,[1,2] **Qingxin Zhu**,[1] **Wenjun Wu**,[2] **Wenlin Huang**,[2] **Chaoqun Hong**,[2] **and Xinzheng Niu**[1,3]

[1]*School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China*
[2]*School of Computer and Information Engineering, Xiamen University of Technology, Xiamen 361024, China*
[3]*School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China*

Correspondence should be addressed to Chaoqun Hong; cqhong@xmut.edu.cn

Received 23 September 2019; Revised 1 January 2020; Accepted 30 January 2020; Published 11 March 2020

Academic Editor: Luis Payá

In the past few years, deep learning has become a research hotspot and has had a profound impact on computer vision. Deep CNN has been proven to be the most important and effective model for image processing, but due to the lack of training samples and huge number of learning parameters, it is easy to tend to overfit. In this work, we propose a new two-stage CNN image classification network, named "Improved Convolutional Neural Networks with Image Enhancement for Image Classification" and PLANET in abbreviation, which uses a new image data enhancement method called InnerMove to enhance images and augment the number of training samples. InnerMove is inspired by the "object movement" scene in computer vision and can improve the generalization ability of deep CNN models for image classification tasks. Sufficient experiment results show that PLANET utilizing InnerMove for image enhancement outperforms the comparative algorithms, and InnerMove has a more significant effect than the comparative data enhancement methods for image classification tasks.

1. Introduction

Deep learning [1] maps data of different classes into separate high dimension space by constructing a complex function with a great number of parameters, which are different from those that obtain instance relationship based on set theory [2, 3]. Deep learning accelerates the rapid development of artificial intelligence in all fields of our daily lives and has imposed far-reaching impacts on scientific fields such as computer vision [4, 5] and natural language processing [6], as well as medical image processing [7]. In the whole picture of deep learning, deep convolutional neural networks have been proved to be the most important and effective models for image processing. However, CNNs learn features of images with so many parameters that they could easily tend to be overfitting and so have poor ability of generalization. In such a case, CNNs could learn features of all the training data well but have poor performance on unseen data, i.e., test data.

To improve the generalization performance of deep models is now a hot research topic for deep convolutional neural networks. The first choice is to reduce the complexity of the deep CNN models. There have been a few approaches proposed such as Dropout [8] and batch normalization [9].

Another choice to improve generalization performance of deep models turns out to be providing sufficient training samples, which are usually hard to satisfy. In most cases, data enhancement skills are adopted to generate more training samples than existing data by adopting various simple operations, such as translation, rotation, flipping, and cropping. Random flipping and cropping are widely used for their good performance in training of deep CNNs, and there have also been some new works for data enhancement, such as mixup [10], RICAP [11], RandomErasing [12], and cutMix [13].

In this work, we propose a novel two-stage CNN image classification network, PLANET, which uses a novel image data enhancement method called InnerMove to enhance

training samples to obtain more effective discriminating features for image classification.

All in all, the main contributions of PLANET are as follows:

(i) Due to hardware barriers, it builds an improved two-stage CNN network for high-resolution image classification. For deep neural networks, a larger input image means a larger model, many more parameters, and more training samples required. The first stage extracts semantic features for a single patch of the image; the second-stage network fuses the semantic features of all patches of the image to classify the image.

(ii) It proposes a novel image enhancement method named InnerMove. InnerMove enhances an image for classification by randomly exchanging the positions of two patches in an image. Position and size of the image patch are randomly generated under certain constraints. It is worth noting that InnerMove can also be used to augment a dataset by applying InnerMove to it multiple times. Augmentation of training samples can inevitably improve the generalization of CNN networks. However, due to hardware barriers, InnerMove is used only once on any input image for data enhancement and augmentation.

2. Related Works

InnerMove is proposed mainly for image data enhancement which can be easily adopted in deep CNNs for image classification tasks, which also acts as a regularization approach to improve generalization performance of deep CNNs. In this section, we demonstrate a few recent works related to deep network regularization and image enhancement.

2.1. Deep Neural Network Regularization. Regularization plays a fundamental role in preventing deep neural networks from overfitting, which makes neural networks perform well on training data but poor on test data. There have been various deep neural network regularization approaches in the last few years [8, 14–17].

Dropout [8] adopts strategy of randomly dropping neurons and corresponding connections during the training process, and experiment results demonstrate that Dropout can significantly improve the performance of deep CNNs on supervised learning tasks in computer vision, speech recognition, document classification, and so on. Then, a generalized version of Dropout named DropConnect [14] was proposed for regularizing large fully connected layers within deep neural networks by randomly setting a subset of activations to zeros. Adaptive Dropout [16] chooses target activations with a probability from a binary belief network, while Stochastic Pooling [17] from a multinomial distribution.

2.2. Image Data Enhancement. Data enhancement can be used to enhance existing data to obtain more discriminating features, and in most cases for deep neural network training, data enhancement is often adopted to generate much larger number of data than the existing data for training parameters in deep neural networks. Therefore, data enhancement can also be considered as a form of regularization which achieves regularization target not by adjusting the architecture of the deep neural network but by generating more effective input data [5, 10–13, 18].

There are many basic operations for data enhancement such as rotation, flipping, translation, and random cropping [19]. Mixup [10] blends two randomly chosen images and their labels to regularize deep convolutional neural networks. RandomErasing [12] randomly chooses a rectangle region of an image and replaces the corresponding pixels with random values or mean value of images from ImageNet, while Cutout [18] masks out square regions of input images during training.

Both Cutmix [13] and RICAP [11] reconstruct new images by cropping and pasting patches within mini-batches with label smoothing; however, there is difference between them. Cutmix crops and switches patches between two images and adjusts their labels with regard to the area of the patches, while RICAP crops four patches from four different images to reconstruct a new image.

3. Improved Convolutional Neural Networks with Image Enhancement

In this section, we present the details of the proposed CNN network PLANET and image enhancement approach InnerMove.

3.1. Details of PLANET. Due to the hardware barriers, we propose an improved two-stage convolutional neural network for high-resolution image classification with a newly proposed image enhancement approach named PLANET. It follows [20] to create many more training samples, and the flowchart of PLANET is illustrated in Figure 1.

PLANET first extracts fixed-size patches from an input image by sliding a window of size $K \times K$ on the image and taking S as the stride. This gives us a total of $[1 + (W - K)/S] \times [1 + (H - K)/S]$ patches, where W and H are the width and height of the image, respectively. In order to illustrate the details of PLANET, without loss of generality, we assume that the resolution (W, H) of training images is (2048, 1536). In patch-wise experiment, we choose patches with $K = 512$ and $S = 256$, so $7 \times 5 = 35$ overlapping image patches are produced, and then each patch will be further enhanced by InnerMove which is detailed in Section 3.2.

PLANET adopts GoogLeNet-based patch-wise CNN for semantic feature extraction from patches. Note that the labels for individual patches are unknown, so we use cross-entropy loss based on the corresponding image label to train the patch-wise CNN. Precondition for doing so is that the distribution of effective classification features of the image is relatively uniform, and most patches contain features related to the image label. Experiment results confirm that our strategy is effective and feasible.

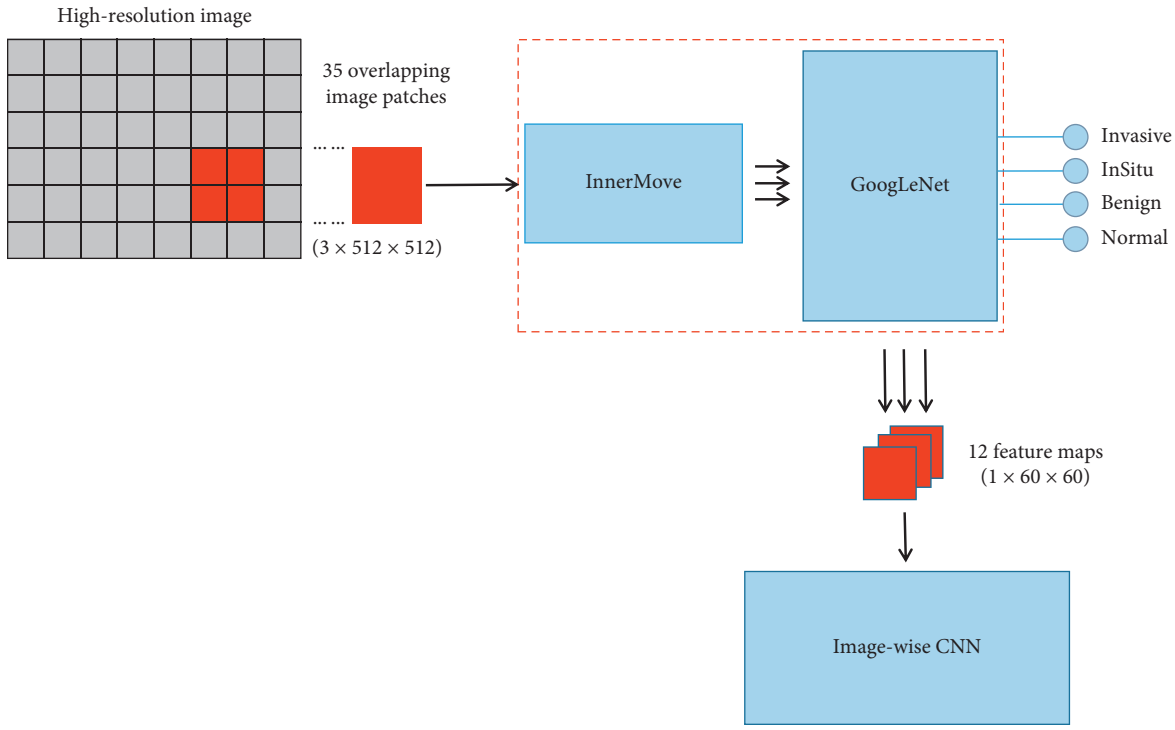


Figure 1: Flowchart of the proposed improved convolutional neural networks with image enhancement for high-resolution image classification (PLANET). Without loss of generality, we assume that the resolution of training images is 2048 ∗ 1536. For any resolution, the value of stride for patch extraction is 256 and 512 for the patch-wise and image-wise network, respectively.

PLANET designs an image-wise CNN for image classification by fusing all patch semantic features, and it follows the architecture of [20] demonstrated in Figure 2 and makes detailed adjustments. It uses a series of 3×3 convolutional layers followed by a 2×2 convolutional layer. The pooling layer uses the stride of 2. Each layer has a batch normalization and ReLu activation function. The last convolutional layer is followed by 3 fully connected layers and finally a softmax classifier. To improve the generalization ability of this network, we use Dropout to regularize the network at a rate of 0.5, and use early stopping to limit overfitting when verification accuracy does not improve.

In order to train the image-wise network, we no longer extract overlapping patches from images by setting $S = 512$, so the total number of patches extracted from an image is $P = 12$. We use GoogLeNet [21] as our backbone of the patch-wise network, and for each patch, the layer we extract feature maps from for the image-wise classification network has 16 channels of 15×15 feature maps. These 16 channels of 15×15 feature maps extracted from the very end level of GoogLeNet are reconstructed into one channel of 60×60 feature maps. The feature maps extracted from the patch-wise CNN for all 12 patches are then stacked together as a smaller 3D output with the channel size of 12. The image-wise network also uses cross-entropy loss to train its parameters and learns to classify images based on local features generated from the image patch and global information shared among patches.

3.2. InnerMove. InnerMove is motivated by the scene of object relative moving in many computer vision tasks such as object tracing, detecting, and recognition. By randomly switching positions of two patches within an input image, InnerMove enhances images for the object moving scene and forces deep neural networks to take the whole scene context into consideration rather than certain local vision features. Image samples enhanced by InnerMove are demonstrated in Figure 3.

In our strategy for data enhancement, InnerMove first randomly crops two patches from an input image and then switches their positions. The enhanced image has the same size and label with the original image.

In this work, when we extract a patch, it may not contain a whole object because the position and size of patches are chosen randomly. But, it can still force networks to learn more context information of input images and improve the generalization of deep CNNs.

3.2.1. Details of InnerMove. In machine learning, a test set is used to validate the performance of a proposed model by measuring some metrics (M), but we cannot optimize learning models on the test set. Therefore, learning models are usually indirectly optimized on the training set by reducing a certain cost function, noted by $J(\boldsymbol{\theta})$ which is defined as (1), during the training process.

$$J(\boldsymbol{\theta}) = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\widehat{p}_{\text{train}}} L(f(\mathbf{x};\boldsymbol{\theta}),\mathbf{y}), \tag{1}$$

where L is the lost function for training samples, $f(\mathbf{x};\boldsymbol{\theta})$ is the model which maps $\mathbf{x}$ to the prediction y, $\widehat{p}_{\text{train}}$ is one kind of empirical distribution of training data, and $\boldsymbol{\theta}$ is the parameter of the learning model f. Ideally, we should get the

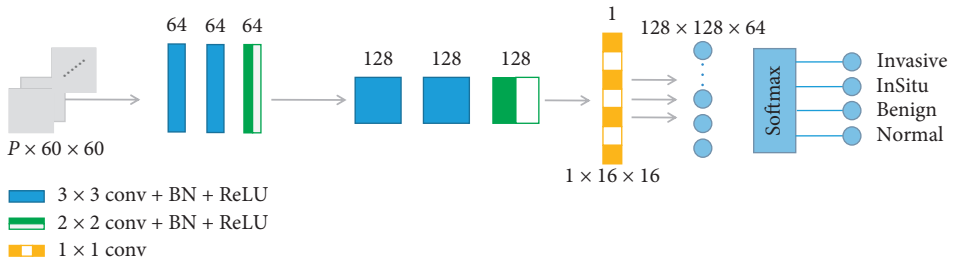


Figure 2: Architecture of image-wise CNN. It follows the image-wise CNN in [20] and makes detailed adjustments.

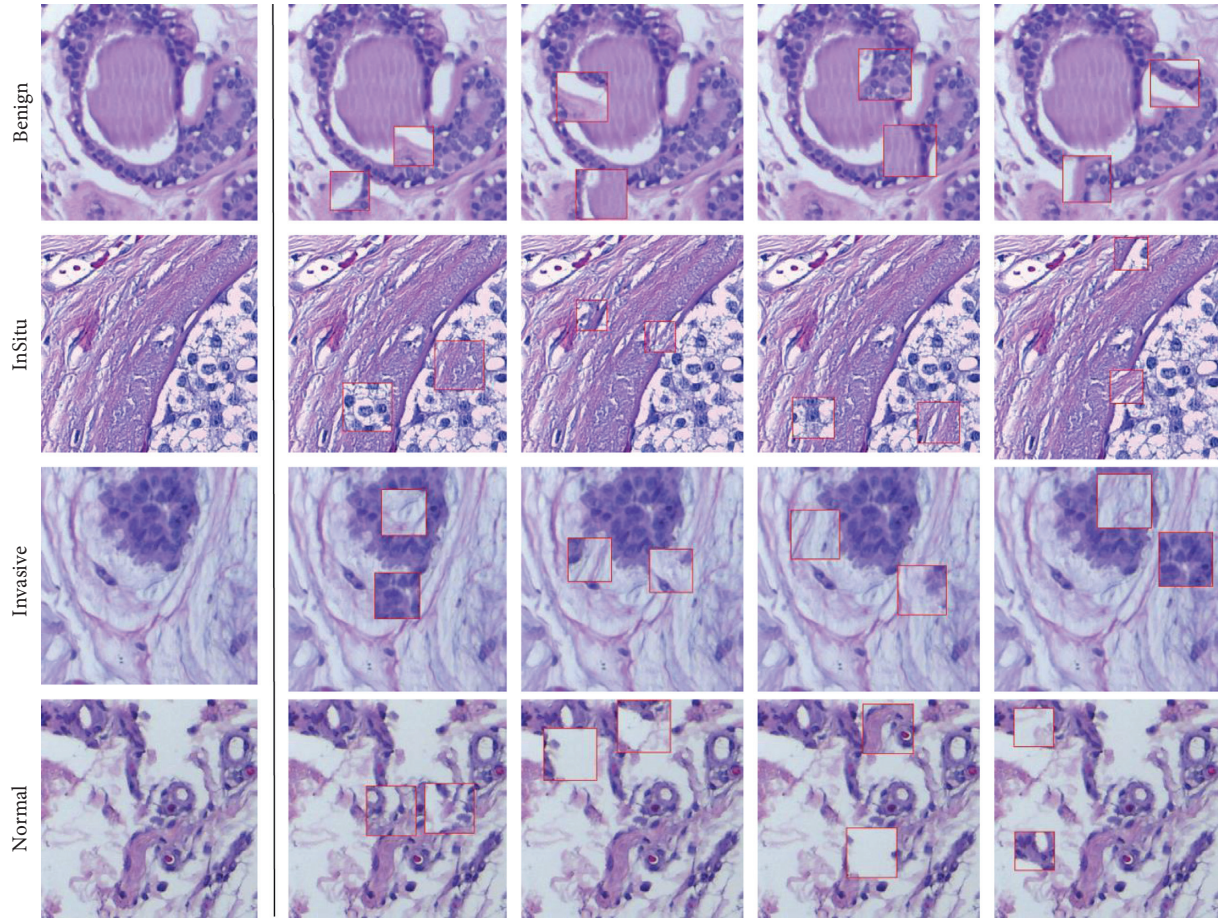


Figure 3: Samples of InnerMove. Four input images of four classes chosen from ICIAR (BACH) 2018 [22] are listed on the left; images on the right side are enhanced samples by InnerMove. The patches chosen to move are noted with small boxes.

actual distribution of the data $\hat{p}_{\text{data}}$ and then calculate the cost function which is defined as (2) for all the data under that distribution.

$$J^*(\boldsymbol{\theta}) = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\hat{p}_{\text{data}}} L(f(\mathbf{x};\boldsymbol{\theta}),\mathbf{y}), \tag{2}$$

where $\hat{p}_{\text{data}}$ is the actual distribution of the data.

The target of machine learning approach is to reduce the value of (2), which is widely known as risk. If $\hat{p}_{\text{data}}$ is given, $f(\mathbf{x};\boldsymbol{\theta})$ will be well trained by optimizing (2). However, in most cases, $\hat{p}_{\text{data}}$ is unknown.

Therefore, sometimes, the optimization of (2) could be simplified into (3) by replacing $\hat{p}_{\text{data}}$ with $\hat{p}_{\text{train}}$:

$$\mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\hat{p}_{\text{train}}} L(f(\mathbf{x};\boldsymbol{\theta}),\mathbf{y}) = \frac{1}{n}\sum_{i=1}^{n} L\left(f\left(\mathbf{x}^{(i)};\boldsymbol{\theta}\right),\mathbf{y}^{(i)}\right), \tag{3}$$

where $\hat{p}_{\text{train}}$ gives every single training sample with the same probability and n is the number of training samples.

The inability to know the true distribution of the data makes our training data limited in number and content, and data enhancement approaches can be adopted to augment training samples for unknown data distribution to some extent.

The partial contribution of this paper is to propose a method (4) to improve the distribution of training data by enhancing training data based on existing data.

$$\widehat{p}_{\text{train}} = \mu\left(\widetilde{x}, \widetilde{y} \mid \mathbf{x}_i, \mathbf{y}_i\right), \tag{4}$$

where $(\widetilde{x}, \widetilde{y}) = \text{InnerMove}(\mathbf{x}_i, \mathbf{y}_i)$, and InnerMove will be detailed soon.

So, (1) can be optimized as

$$\mathbb{E}_{(\widetilde{x},\widetilde{y})\sim\widehat{p}_{\text{train}}} L\left(f\left(\widetilde{x}; \boldsymbol{\theta}\right), \widetilde{y}\right) = \frac{1}{n}\sum_{i=1}^{n} L\left(f\left(\widetilde{x}^{(i)}; \boldsymbol{\theta}\right), \widetilde{y}^{(i)}\right). \tag{5}$$

In order to optimize (5), gradients of (5) to $\boldsymbol{\theta}$ should be calculated as (6) for m training samples in a mini-batch, with which SGD is utilized to update $\boldsymbol{\theta}$.

$$\widehat{g} = \frac{1}{m}\nabla_{\boldsymbol{\theta}} \sum_{i=1}^{m} L\left(f\left(\widetilde{x}^{(i)}; \boldsymbol{\theta}\right), \widetilde{y}^{(i)}\right). \tag{6}$$

InnerMove randomly chooses two patches without overlapping areas and switches their positions as illustrated in Figure 4. Box $= (p, w, h)$ represents a box which is used to crop a patch within an image. When any input image I_o with resolution (w_o, h_o) comes, two boxes, $b_1 = (p_1, w, h)$ and $b_2 = (p_2, w, h)$, with two random position points p_1, p_2, width w, and height h of the patches are generated within the image. In detail, w and h are generated by $w \longleftarrow \text{int}(r * w_o)$ and height $h \longleftarrow \text{int}(r * h_o)$, while $r \in (0, 1)$ is obtained from beta distribution as (7). We choose beta distribution for the reason that it can generate $r \in (0, 1)$, while restrict r to a narrow scope we desire with a high probability.

$$\text{beta}(\alpha, b) = \frac{\boldsymbol{\theta}^{a-1}(1-\boldsymbol{\theta})^{b-1}}{B(\alpha, b)} \propto \boldsymbol{\theta}^{a-1}(1-\boldsymbol{\theta})^{b-1}, \tag{7}$$

where $B(\alpha, b)$ is used to ensure that the integral of $\text{beta}(\alpha, b)$ is equal to 1.

We need to guarantee that the generated boxes, $b_1 = (p_1, w, h)$ and $b_2 = (p_2, w, h)$, are totally contained within the image, or we will try this process again. It is worth noting that, in this work, $b_1 \cap b_2$ is $\{\phi\}$. Then, two patches, P_1, P_2, are cropped from the image I_o by performing operations: $P_1 \longleftarrow \text{patchExtra}(b1, I_o)$; $P_2 \longleftarrow \text{patchExtra}(b2, I_o)$. The last step we need to do is pasting these two patches to the chosen positions as $\text{paste}(P_1, p_2, I_o), \text{paste}(P_2, p_1, I_o)$.

3.2.2. Comparison with Cutout. Cutout [18] is motivated by the scene of object occlusion in image processing, and it adopts the strategy of randomly masking out the square region of the input image to simulate the object occlusion scene. However, InnerMove tries to take the scene of "object moving" into consideration. Both Cutout and InnerMove manage to force deep CNN to take a wider range of features into consideration by cropping chosen patches away from their original positions as shown in Figure 5, and the difference is that Cutout drops chosen patches while InnerMove moves them to other positions within images so that no extra information is lost.

4. Experiment Results

4.1. Experiment Setting. In this section, we evaluate the proposed PLANET mainly on image classification tasks and three popular data sets: CIFAR-10 [23], CIFAR-100 [23], and ICIAR (BACH) 2018 [22]. We conduct our experiments on a single NVIDIA Titan XP GPU.

There are 4 classes in ICIAR (BACH) 2018 as illustrated in Figure 6: normal, benign, inSitu and invasive, and each class has 100 training samples. We adopt 3-fold cross-validation and choose 80 images of each class for training and 20 for validation.

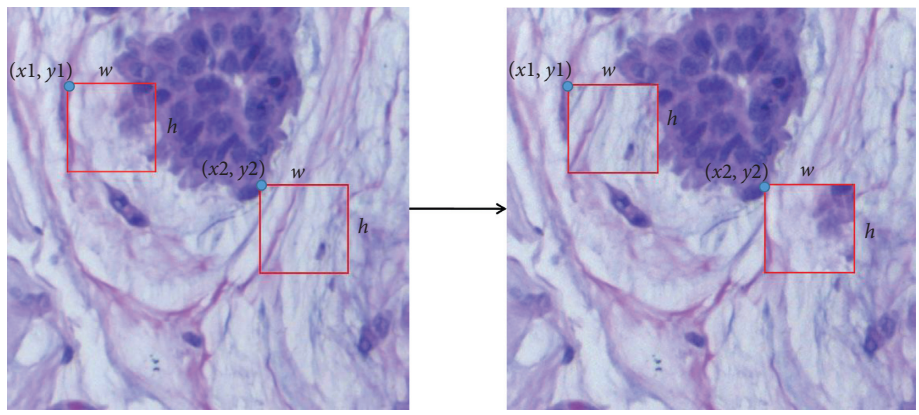
Due to the serious shortage of training data, this article utilizes the methods adopted by [20] to simply transform each image patch to generate more training samples. Transformation methods include rotations, mirroring, and random color perturbations. By default, the experiments carried out in this article all use patches augmented by the above transformation methods as training samples, which have the same labels as the corresponding images.

4.2. Results for InnerMove. It is worth noting that InnerMove can be easily extended by changing the number of patches to switch positions. In this work, InnerMove-2 indicates that the number of patches is 2. Images in CIFAR-10 and CIFAR-100 are small, and the image resolution is 32×32, much smaller than 2048×1536 of ICIAR (BACH) 2018. We conduct TSCNN [20] and Resnet18 [24] experiments using InnerMove-2 and InnerMove-4, respectively, for different databases to demonstrate the generalization of InnerMove.

4.2.1. Parameter Selection for InnerMove. For both InnerMove-2 and InnerMove-4, we conduct experiments to evaluate the impact of side length ratio of patches to the original image. As illustrated in Figure 7(a), TSCNN with InnerMove-2 performs better on ICIAR (BACH) 2018 with the patch side length ratio ranging from 0.28 to 0.35; results of Resnet18 with InnerMove-4 on CIFAR-10 are demonstrated in Figure 7(b). We can see that the best side length ratio differs with regard to different data sets; therefore, we assign different side length ratio ranges to (r_min, r_max). For CIFAR-10 and CIFAR-100, we adopt (0.35, 0.45). It is worth noting that results of InnerMove depend not only on the side length ratio but also the positions of chosen patches, so the results might slightly vary for different trials.

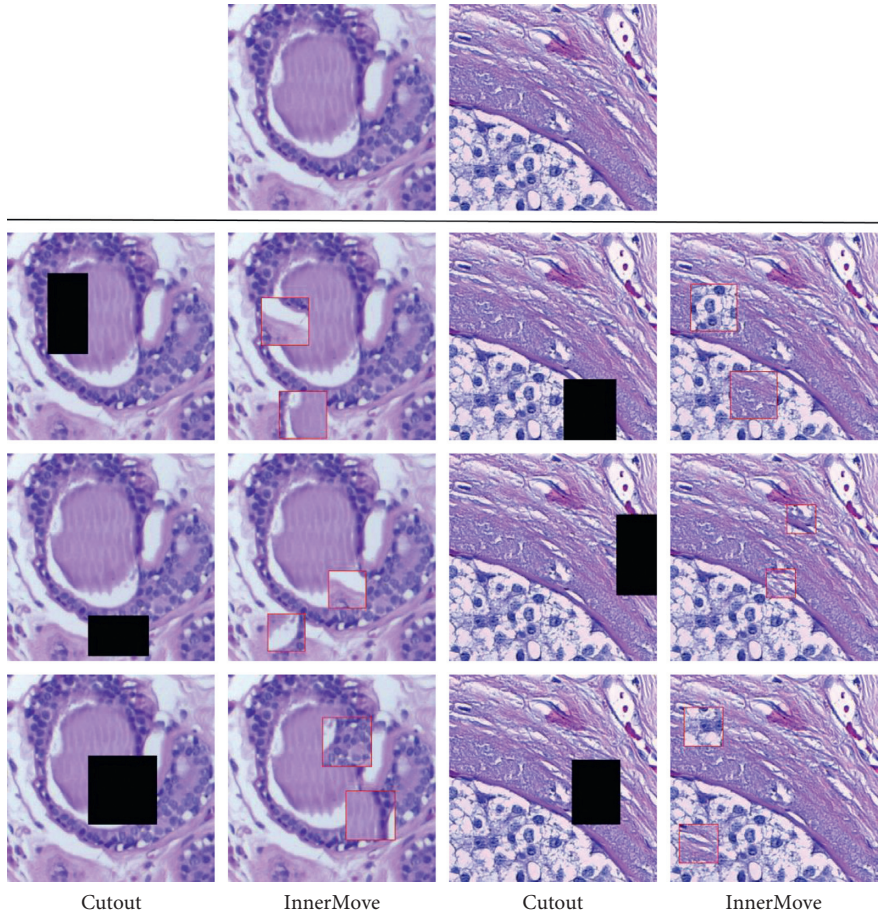
When side length ratio range is given, we verify the effectiveness of InnerMove using TSCNN as the baseline algorithm. Average experiment accuracies of TSCNN with about half pretrained parameters reused on all classes of images in ICIAR (BACH) 2018 within 30 training epochs are illustrated in Figure 8. Without loss of generality, we use InnerMove-2 for image enhancement. As demonstrated in Figure 8, in all epochs, accuracies of TSCNN with InnerMove are higher than those without InnerMove when parameters have been well trained.

4.2.2. Comparison with Cutout and RandomErasing. We verify the effectiveness of InnerMove-4 by conducting experiments on CIFAR-10 and CIFAR-100, using Resnet18 as the baseline deep CNN model, and configure the learning model as suggested in [18]. It is worth noting that InnerMove-4 is



Input image (a) Output image (b)

FIGURE 4: Illustration of InnerMove. It randomly chooses two patches without overlapping areas and switches their positions.



Cutout InnerMove Cutout InnerMove

FIGURE 5: Difference between InnerMove and Cutout. Two images are placed above the line, and the corresponding enhanced images from InnerMove and Cutout are listed under the line. They both manage to force deep CNN to take a wider range of features into consideration by cropping chosen patches away from their original positions.

different from InnerMove-2 in choosing patches from each image: (1) the original image is divided into four blocks of the same size without intersection; (2) InnerMove-4 chooses 4 patches randomly from four blocks with side length ratio from 0.35 to 0.45 which is generated from (7).

The experiment comparison between InnerMove, Cutout, and RandomErasing is illustrated in Table 1. As shown in Table 1, Resnet18 with InnerMove, Cutout, and RandomErasing performs better than Resnet18 itself, but InnerMove works better than Cutout and RandomErasing

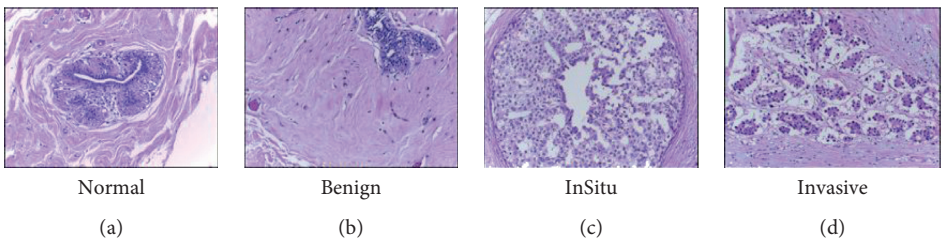


Figure 6: Samples of ICIAR (BACH) 2018. There are 4 classes in ICIAR (BACH) 2018: normal, benign, inSitu, and invasive, and each class has 100 training samples. We use 80 of each class for training and 20 for validation.

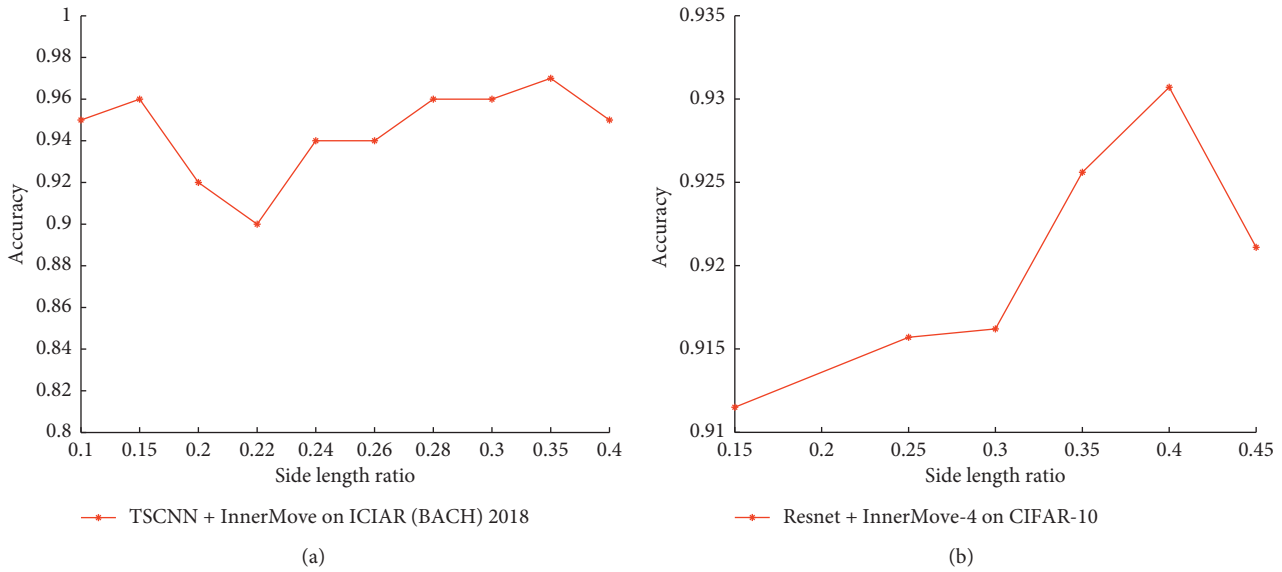


Figure 7: Average accuracy of 2 approaches with InnerMove for the same training and validation data with regard to different side length ratios of the chosen patch to the original image.

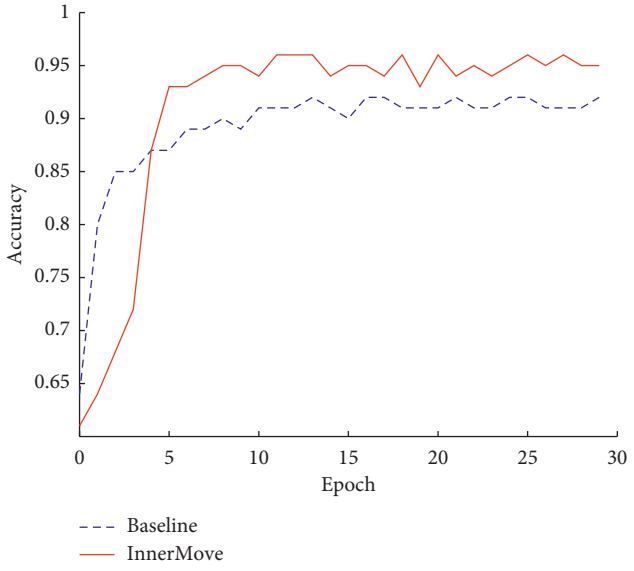


Figure 8: Average accuracy of TSCNN (baseline in the figure) and TSCNN with InnerMove-2 (InnerMove in the figure) on all classes of images in ICIAR (BACH) 2018 in 30 epochs for 3-fold cross-validation.

Table 1: Test error rates (%) of InnerMove, Cutout, and RandomErasing on CIFAR-10 and CIFAR-100 and C10 and C100 for brevity are illustrated in this table. "+" means training data are augmented by color jitter and brightness adjustment. It is worth noting that the best results are marked in bold.

Method	C10	C10+	C100	C100+
Resnet18 [24]	10.04	6.67	35.84	26.52
Resnet18+Cutout	9.14	5.91	34.37	26.77
Resnet18+RandomErasing	9.47	5.65	35.2	27.16
Resnet18+InnerMove-4	**8.6**	**5.6**	**32.42**	**25.56**

Table 2: Experiment results, mean ± std (%), for 3-fold cross-validation of PLANET with and without InnerMove on ICIAR (BACH) 2018 [22] are illustrated in this table. Precision, Recall, and Accuracy are abbreviated as *P*, *R*, and Acc, respectively. It is worth noting that the better results of all the 13 comparative values are marked in bold.

	PLANET without InnerMove				PLANET with InnerMove			
Class	*P*	*R*	*F*1	Acc	*P*	*R*	*F*1	Acc
Normal	**95 ± 3.51**	**95 ± 3.66**	**95 ± 3**	92	92 ± 2.89	93 ± 2.89	93 ± 2.52	**95**
Benign	**94 ± 3.81**	**97 ± 2.89**	**95 ± 3.46**		89 ± 2.31	92 ± 2.89	90 ± 0.0	
InSitu	86 ± 3.51	93 ± 2.89	90 ± 2.89		**100**	**97 ± 2.89**	**98 ± 1.73**	
Invasive	94 ± 0.0	83 ± 2.89	88 ± 1.73		$**100 ± 0.0**$	**98 ± 2.89**	**99 ± 1.73**	

when Resnet18 trains on CIFAR-10 and CIFAR-100 with or without other data enhancement methods such as color jitter and brightness adjustment. In detail, InnerMove has significantly improved the accuracy of Resnet18 on C10, C10+, C100, and C100+ by 1.44%, 1.07%, 3.42%, and 0.96%, respectively.

4.3. Results for PLANET. We conduct comparative experiments on ICIAR (BACH) 2018 [22] between PLANET and the following approaches. To simplify the setup of comparative experiments, PLANET and the compared algorithms use InnerMove-2 to enhance images by default.

(i) TSCNN [20]: TSCNN's network framework is similar to PLANET, and it also builds a two-stage CNN network. The first-stage network is used to learn the semantic features of image patches, and the second-stage network is used to fuse spatial information of the patches to classify images;

(ii) Resnet [24]: Resnet cleverly uses shortcut connections to solve the problem of model degradation in deep networks so that it can be designed deeper and get higher-level semantic information.

(iii) GoogLeNet [21]: GoogLeNet uses multiple inception modules to build the deep network. In the single inception module, multiple small convolution kernels are utilized to replace convolution operations of a large convolution kernel, which greatly reduce the number of parameters. At the same time, inception can make the network extract more kinds of local semantic features.

The best experiment results of PLANET within 30 epochs are demonstrated in Table 2. As shown in Table 2, in total of 13 comparative indicator values of PLANET with InnerMove, 7 of them are better than those of PLANET without InnerMove, and accuracy has been improved by 3%, especially for inSitu and invasive classes which are of much more importance for clinical diagnosis. For InSitu and Invasive classes, the results of PLANET with InnerMove are much better than those of PLANET without InnerMove in Precision, Recall, and *F*1 indicators.

The best experiment results of TSCNN within 30 epochs are demonstrated in Table 3. As shown in Table 3, (1) the overall performance of the classification algorithm is improved after using InnerMove as image enhancement approach, and accuracy has been improved by 4%; (2) in total of 26 comparative indicator values of TSCNN, only 4 of them are better than those of PLANET with InnerMove, especially for InSitu and Invasive classes which are of much more importance for clinical diagnosis. For InSitu and Invasive classes, the results of PLANET with InnerMove are much better than those of TSCNN with or without InnerMove in Precision, Recall, and *F*1 indicators. Therefore, PLANET with InnerMove outperforms TSCNN even when TSCNN also utilizes InnerMove for image enhancement.

The best experiment results of Resnet18 within 30 epochs are demonstrated in Table 4. As shown in Table 4, (1) the overall performance of the classification algorithm is improved after using InnerMove as image enhancement approach, and accuracy has been improved by 3%; (2) in total of 26 comparative indicator values of Resnet18, only 3 of them are better than those of PLANET with InnerMove, especially for InSitu and Invasive classes which are of much more importance for clinical diagnosis. For InSitu and Invasive classes, the results of PLANET with InnerMove are much better than those of Resnet18 with or without InnerMove in Precision, Recall, and *F*1 indicators. Therefore, PLANET with InnerMove outperforms Resnet18 even when Resnet18 also utilizes InnerMove for image enhancement.

The best experiment results of GoogLeNet within 30 epochs are demonstrated in Table 5. As shown in Table 5, (1)

Table 3: Experiment results, mean ± std (%), for 3-fold cross-validation of TSCNN on ICIAR (BACH) 2018 [22] are illustrated in this table. Precision, Recall, and Accuracy are abbreviated as *P*, *R*, and Acc, respectively. Corresponding results better than PLANET with InnerMove are marked in bold.

	TSCNN without InnerMove				TSCNN with InnerMove			
Class	*P*	*R*	*F*1	Acc	*P*	*R*	*F*1	Acc
Normal	72 ± 1.73	**100 ± 0.0**	84 ± 1.15	89	88 ± 1.73	**97 ± 2.89**	92 ± 1.73	93
Benign	**100 ± 0.0**	65 ± 0.0	79 ± 0.0		91 ± 4.62	83 ± 5.77	87 ± 1.15	
InSitu	95 ± 4.51	**100 ± 0.0**	98 ± 2.52		97 ± 2.89	95 ± 0.0	96 ± 1.15	
Invasive	100 ± 0.0	92 ± 2.89	96 ± 1.15		98 ± 2.89	96 ± 2.31	97 ± 1.53	

Table 4: Experiment results, mean ± std (%), for 3-fold cross-validation of Resnet18 on ICIAR (BACH) 2018 [22] are illustrated in this table. Precision, Recall, and Accuracy are abbreviated as *P*, *R*, and Acc, respectively. Corresponding results better than PLANET with InnerMove are marked in bold.

	Resnet18 without InnerMove				Resnet18 with InnerMove			
Class	*P*	*R*	*F*1	Acc	*P*	*R*	*F*1	Acc
Normal	82 ± 2.31	**95 ± 0.0**	87 ± 1.15	87	85 ± 2.08	**97 ± 2.89**	91 ± 2.52	90
Benign	**93 ± 0.58**	72 ± 2.89	81 ± 1.73		90 ± 5.51	77 ± 5.77	83 ± 3.06	
InSitu	85 ± 1.73	93 ± 2.89	89 ± 1.15		88 ± 2.31	93 ± 2.89	91 ± 1.73	
Invasive	93 ± 2.89	90 ± 0.0	91 ± 1.15		98 ± 2.89	95 ± 5	97 ± 1.53	

Table 5: Experiment results, mean ± std (%), for 3-fold cross-validation of GoogLeNet on ICIAR (BACH) 2018 [22] are illustrated in this table. Precision, Recall, and Accuracy are abbreviated as *P*, *R*, and Acc, respectively. Corresponding results better than PLANET with InnerMove are marked in bold.

	GoogLeNet without InnerMove				GoogLeNet with InnerMove			
Class	*P*	*R*	*F*1	Acc	*P*	*R*	*F*1	Acc
Normal	87 ± 2.31	93 ± 2.89	90 ± 0.0	90	89 ± 3.46	93 ± 2.08	91 ± 1.53	92
Benign	90 ± 6.93	83 ± 5.77	86 ± 0.0		**93 ± 2.65**	87 ± 2.65	90 ± 0.0	
InSitu	90 ± 4.51	95 ± 0.0	93 ± 2.52		92 ± 3.79	92 ± 3.79	92 ± 0.58	
Invasive	97 ± 2.89	92 ± 2.89	94 ± 1.73		95 ± 2.52	96 ± 4.16	95 ± 0.58	

the overall performance of the classification algorithm is improved after using InnerMove as image enhancement approach, and accuracy has been improved by 2%; (2) in total of 26 comparative indicator values of GoogLeNet, only one of them is better than those of PLANET with InnerMove, especially for InSitu and Invasive classes which are of much more importance for clinical diagnosis. For InSitu and Invasive classes, the results of PLANET with InnerMove are much better than those of GoogLeNet with or without InnerMove in Precision, Recall, and *F*1 indicators. Therefore, PLANET with InnerMove outperforms GoogLeNet even when GoogLeNet also utilizes InnerMove for image enhancement.

5. Conclusion

In this work, we propose an improved two-stage image classification CNN network called PLANET. The first-stage network is used to extract semantic features of image patches, and the second-stage network is used to fuse the semantic features of all patches to classify images. This paper also proposes an image enhancement method called InnerMove, which randomly selects two or more patches and switches their positions within an image to simulate the "object movement" scene of computer vision tasks. Sufficient experiments have been conducted for classification tasks on CIFAR-10, CIFAR-100, and ICIAR (BACH) 2018. Experiment results show that PLANET has better classification performance than comparative algorithms, and InnerMove is effective and feasible for data enhancement in image classification tasks. We plan to further investigate the usage of InnerMove in other computer vision tasks such as image segmentation and object detection.

Data Availability

The ICIAR (BACH) 2018, CIFAR-10, and CIFAR-100 used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Ministry of Education Pre-Research Foundation, the Fujian Provincial Natural Science Foundation of China (no. 2018J01573), and the Foundation of Fujian Educational Committee (no. JAT160357).

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] W. Zhu, "Relationship among basic concepts in covering-based rough sets," *Information Sciences*, vol. 179, no. 14, pp. 2478–2486, 2009.

[3] W. Zhu, "Relationship between generalized rough sets based on binary relation and covering," *Information Sciences*, vol. 179, no. 3, pp. 210–225, 2009.

[4] T. B. Moeslund and E. Granum, "A survey of computer vision-based human motion capture," *Computer Vision and Image Understanding*, vol. 81, no. 3, pp. 231–268, 2001.

[5] C. Tang, Q. Zhu, and C. Hong, "Exploiting geometrical structures using autoencoders and click data for image re-ranking," *Neurocomputing*, vol. 235, pp. 157–163, 2017.

[6] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *Journal of Machine Learning Research*, vol. 12, no. 1, pp. 2493–2537, 2011.

[7] T. Araujo, G. Aresta, E. Castro et al., "Classification of breast cancer histology images using convolutional neural networks," *PLoS One*, vol. 12, no. 6, Article ID e0177544, 2017.

[8] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[9] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 2015 International Conference on International Conference on Machine Learning*, Gangzhou, China, July 2015.

[10] H. Zhang, M. Cissé, Y. N. Dauphin, and D. Lopez-Paz, "mixup: beyond empirical risk minimization," 2017, https://arxiv.org/abs/1710.09412.

[11] R. Takahashi, T. Matsubara, and K. Uehara, "Data augmentation using random image cropping and patching for deep cnns," 2018, https://arxiv.org/abs/1811.09030.

[12] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," 2017, https://arxiv.org/abs/1708.04896.

[13] S. Yun, D. Han, S. J. Oh et al., "Cutmix: regularization strategy to train strong classifiers with localizable features," 2019, https://arxiv.org/abs/1905.04899.

[14] L. Wan, M. D. Zeiler, S. Zhang, Y. LeCun, and R. Fergus, "Regularization of neural networks using dropconnect," in *Proceedings of the 30th International Conference on Machine Learning, ICML 2013*, pp. 1058–1066, Atlanta, GA, USA, June 2013.

[15] G. Kang, X. Dong, L. Zheng, and Y. Yang, "Patchshuffle regularization," 2017, https://arxiv.org/abs/1707.07103.

[16] L. J. Ba and B. J. Frey, "Adaptive dropout for training deep neural networks," in *Proceedings of the 27th Annual Conference on Neural Information Processing Systems 2013*, pp. 3084–3092, Lake Tahoe, NV, USA, December 2013.

[17] M. D. Zeiler and R. Fergus, "Stochastic pooling for regularization of deep convolutional neural networks," in *Proceedings of the 1st International Conference on Learning Representations, ICLR 2013*, Scottsdale, AZ, USA, May 2013.

[18] T. Devries and G. W. Taylor, "Improved regularization of convolutional neural networks with cutout," 2017, https://arxiv.org/abs/1708.04552.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 26th Annual Conference on Neural Information Processing Systems 2012*, pp. 1106–1114, Lake Tahoe, NV, USA, December 2012.

[20] K. Nazeri, A. Aminpour, and M. Ebrahimi, "Two-stage convolutional neural network for breast cancer histology image classification," in *Proceedings of the 15th International Conference on Image Analysis and Recognition, ICIAR 2018*, pp. 717–726, Póvoa de Varzim, Portugal, June 2018.

[21] S. Christian, W. Liu, Y. Jia et al., "Going deeper with convolutions," 2015, https://arxiv.org/abs/1409.4842.

[22] G. Aresta, T. Araújo, S. Kwok et al., "BACH: grand challenge on breast cancer histology images," *Medical Image Analysis*, vol. 56, pp. 122–139, 2019.

[23] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Master's thesis, University of Toronto, Toronto, Canada, 2009.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proceedings of the 14th European Conference on Computer Vision, ECCV 2016*, pp. 630–645, Amsterdam, Netherlands, October 2016.