SPDA: Superpixel-based Data Augmentation for Biomedical Image Segmentation

Yizhe Zhang[1] YZHANG29@ND.EDU
Lin Yang[1] LYANG5@ND.EDU
Hao Zheng[1] HZHENG3@ND.EDU
Peixian Liang[1] PLIANG@ND.EDU
Colleen Mangold[2] CAV154@PSU.EDU
Raquel G. Loreto[2] RAQUELGLORETO@GMAIL.COM
David P. Hughes[2] DHUGHES@PSU.EDU
Danny Z. Chen[1] DCHEN@ND.EDU
[1]*Department of Computer Science and Engineering, University of Notre Dame, USA*
[2]*Department of Entomology and Department of Biology, Center for Infectious Disease Dynamics, Pennsylvania State University, USA*

Abstract

Supervised training a deep neural network aims to "teach" the network to mimic human visual perception that is represented by image-and-label pairs in the training data. Superpixelized (SP) images are visually perceivable to humans, but a conventionally trained deep learning model often performs poorly when working on SP images. To better mimic human visual perception, we think it is desirable for the deep learning model to be able to perceive not only raw images but also SP images. In this paper, we propose a new superpixel-based data augmentation (SPDA) method for training deep learning models for biomedical image segmentation. Our method applies a superpixel generation scheme to all the original training images to generate superpixelized images. The SP images thus obtained are then jointly used with the original training images to train a deep learning model. Our experiments of SPDA on four biomedical image datasets show that SPDA is effective and can consistently improve the performance of state-of-the-art fully convolutional networks for biomedical image segmentation in 2D and 3D images. Additional studies also demonstrate that SPDA can practically reduce the generalization gap.

1. Introduction

Traditional data augmentation methods use a combination of geometric transformations to artificially inflate training data (Perez and Wang, 2017). For each raw training image and its corresponding annotated image, it generates "duplicate" images that are shifted, zoomed in/out, rotated, flipped, and/or distorted. These basic/traditional data augmentation methods are generally applicable to classification problems where the output is a vector and segmentation problems where the output is a segmentation map.

Recently, generative adversarial networks (GANs) have been used for data augmentation (e.g., (Antoniou et al., 2017)). Encouraging the generator to produce realistic looking images (comparing to the original images) is a main consideration when training the generator. A key issue to this consideration is that it does not define/imply what kind of generated images would be use-

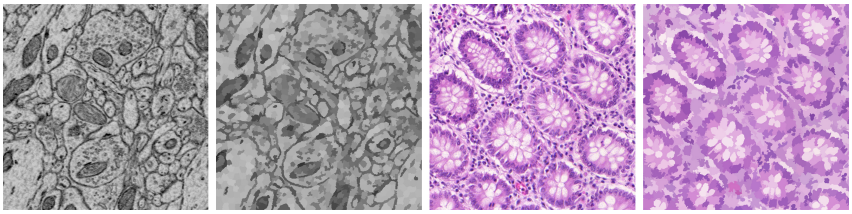

Figure 1: From left to right: An electron micrograph of neuronal structure, its superpixelized image, an H&E stained pathological image of glands, and its superpixelized image. The superpixels preserve the essential objects and their boundaries.

ful/meaningful for data augmentation purpose, and the generator does not necessarily converge to a model version that generates useful new data for training a better segmentation or classification model. (Wang et al., 2018) was proposed to deal with this issue using a task-related classifier for training an image generator. However, the method in (Wang et al., 2018) was designed for classification problems; in segmentation, the distributions of labels are usually much more complicated and it is quite non-trivial to extend the method (Wang et al., 2018) to segmentation tasks.

As an algorithm based (non-learning based) data augmentation technique, mixup (Zhang et al., 2017) was proposed to generate new image samples "between" pairs of training samples for image classification problems. It was motivated based on the principles of Vicinal Risk Minimization (Chapelle et al., 2001) and its experimental results showed promising classification accuracy improvement. In (Eaton-Rosen et al., 2018), it extended the mixup method to medical image segmentation, showing that mixup is also applicable to data augmentation for segmentation problems.

In this paper, we propose a new algorithm-based data augmentation technique that uses superpixels for better training a deep learning model for biomedical image segmentation. Our method is based on a common experience that superpixelized (SP) images are visually perceivable to humans (see Fig. 1), but a conventionally trained deep learning model (trained using only raw images) often performs poorly when working on SP images. This phenomenon implies that a conventionally trained deep learning model may not mimic human visual behaviors well enough. Thus, we think encouraging a deep learning network to be able to perceive not only raw images but also SP images can make it more closely mimic human visual perception. Our method is built on this idea, by adding SP images to the training data for training a deep learning model. Our new superpixel-based data augmentation (SPDA) method can work together with traditional data augmentation methods and be generally applicable to many deep learning based image segmentation models.

A short summary of our SPDA method is as follows. For each raw image, we apply a superpixel generation method (e.g., SLIC (Achanta et al., 2012)) to obtain superpixel cells. Superpixel cells are groups of pixels that are visually similar and spatially connected. For every superpixel cell C, we compute the average pixel value(s) for all the pixels in C and assign the computed average value(s) to all the pixels in C. In this way, we effectively remove very local image details and emphasize more on the overall colors, shapes, and spatial relations of objects in the image (see Fig. 1). After "superpixelizing" all the raw images in the original training set, we put all the superpixelized (SP) images into the training set together with the original training images for training a deep learning model. Our experiments of SPDA on four biomedical image datasets show that SPDA is effec-

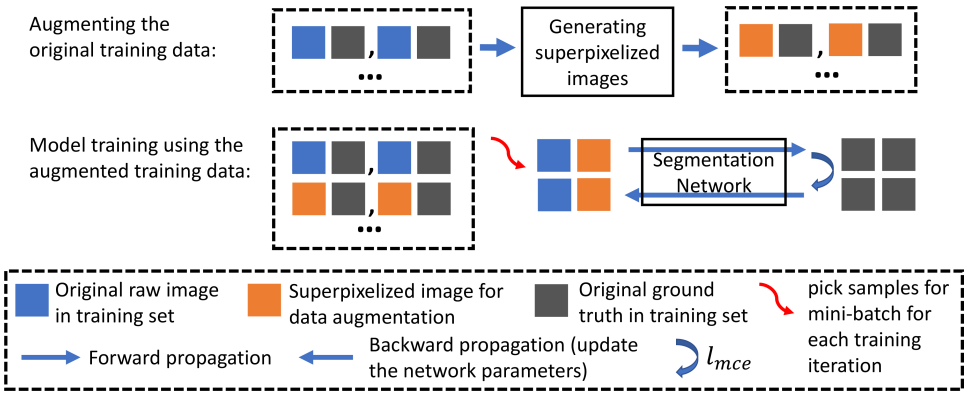

Figure 2: An overview of our SPDA method. During training, we first generate superpixelized images for all the raw images, and then add the SPDA-generated data to the training data for training a segmentation model. Note that the trained network is applied to only raw images during model testing.

tive and can consistently improve the performance of state-of-the-art fully convolutional networks (FCNs) for biomedical image segmentation in 2D and 3D images.

In Section 2, we discuss several technical considerations on generating superpixelized images for data augmentation, and present our exact procedure for generating and using superpixelized images to train deep learning models. In Section 3, we evaluate SPDA using multiple widely used FCNs on four biomedical image segmentation datasets, and show that SPDA consistently yields segmentation performance improvement on these datasets.

2. Superpixels for Data Augmentation

First, we give some notation and background of data augmentation. Then, we discuss several technical considerations on using superpixels for data augmentation. Finally, we present the key technical components: (i) What superpixel generation method we choose to use and the logic behind it; (ii) the exact procedure for generating superpixelized images; (iii) the training objective function and algorithm for using SPDA-generated images in deep learning model training. Fig. 2 gives an overview of our SPDA method for model training.

2.1. Notation and preliminaries

Given a set of image samples $X = \{x_1, \ldots, x_n\}$ and their corresponding ground truth $Y = \{y_1, \ldots, y_n\}$, for training a segmentation model (e.g., an FCN) $f \in F$ that describes the relationship between x_i and y_i, the empirical risk is:

$$\frac{1}{n}\sum_{i=1}^{n}\mathscr{L}(f(x_i), y_i) \tag{1}$$

where $\mathscr{L}$ is a loss function (e.g., the cross-entropy). Learning the function f is by minimizing Eq. (1), which is also known as Empirical Risk Minimization.

One could use some proper functions to generate more data based on the original training data pair (x_i, y_i). In general, we denote the generated data by (x_i^{aug}, y_i^{aug}).

When there are multiple (k) versions of augmented data for one pair (x_i, y_i), the loss with augmented data can be written as:

$$\frac{1}{n}\sum_{i=1}^{n}(\mathscr{L}(f(x_i), y_i) + \lambda \sum_{j=1}^{k} \mathscr{L}(f(x_i^{aug_j}), y_i^{aug_j})) \tag{2}$$

where λ is a hyper-parameter that controls the importance of the data augmentation term. Different ways of data augmentation produce different new data, and thus directly affect the learning procedure of f. As a common practice, flipping, rotation, cropping, etc. are widely used for data augmentation. This type of data augmentation applies geometric transformations (g_k, for k different geometric transformations) to both x_i and y_i, to generate new pairs of training data. For this type of data augmentation, Eq. (2) can be rewritten as:

$$\frac{1}{n}\sum_{i=1}^{n}(\mathscr{L}(f(x_i), y_i) + \lambda \sum_{j=1}^{k} \mathscr{L}(f(g_j(x_i)), g_j(y_i))) \tag{3}$$

Another type of data augmentation makes no change on y_i, and the only modification/augmentation is on x_i (e.g., color jittering (Krizhevsky et al., 2012)). For this type of augmentation, Eq. (2) can simply be:

$$\frac{1}{n}\sum_{i=1}^{n}(\mathscr{L}(f(x_i), y_i) + \lambda \sum_{j=1}^{k} \mathscr{L}(f(G(x_i)), y_i)) \tag{4}$$

where $G(\cdot)$ is a label-preserving transformation. Our new SPDA method belongs to this category. We propose to generate superpixlized images (denoted by $SP(\cdot)$) as a type of label-preserving (perception-preserving) transformation for data augmentation.

Below we discuss several technical considerations on using superpixels for data augmentation, the technical details of $SP(\cdot)$, and how to use SPDA-generated data for model training.

2.2. Technical considerations

In this subsection, we discuss three technical considerations on generating superpixelized images for data augmentation.

(1) Superpixelizing an image x removes or reduces local image details in x that might be less relevant to modeling $P(y|x, \theta_f)$ (θ_f denotes the parameters of the segmentation model f). A superpixelized image $SP(x)$ is a simplified version of the original image x. Letting a deep learning model learn from $SP(x)$ to predict y means asking the model to use little or no local (insignificant) pixel value changes and focus more on higher-level semantic information. Since model parameters are shared between predicting y when given x and predicting y when given $SP(x)$, modeling $P(y|SP(x), \theta_f)$ will influence modeling $P(y|x, \theta_f)$. As a result, because of the joint modeling of $P(y|SP(x), \theta_f)$, the learned function for predicting y given x would become more invariant/insensitive to local image noise and small details, and would learn and utilize more higher level image information and representations. Note that all the original training images with all their local image details are still fully kept in the training dataset. Hence, whenever needed, the learning procedure is still able to use any local image details for modeling $P(y|x, \theta_f)$.

(2) SPDA provides new image samples that are "close" to the original training samples. Under the principle of Vicinal Risk Minimization or VRM (Chapelle et al., 2001), a vicinity or neighborhood around every training sample is defined or suggested based on human knowledge. Additional samples then can be drawn from this vicinity distribution of the training samples to increase or enlarge the support of the training sample distribution (Zhang et al., 2017). Superpixelized images most of the time are conceptually meaningful to human eyes. It is likely that superpixelized images are also close[1] to their corresponding original image samples in the data space. If this "close neighborhood" property is true, then adding SPDA-generated data to the training data should be helpful to improve the generalization capability of the model, according to VRM (Chapelle et al., 2001). In **Appendix A.1**, we show that after using a generic dimensionality reduction method (e.g., PCA, t-SNE (Maaten and Hinton, 2008)), one can observe that each superpixelized image is in a close neighborhood of its corresponding original image.

(3) Adding superpixelized images to the training set makes the data distribution of the training set thus resulted closer to the test data distribution or the true data distribution. Superpixelized images form a more general and broader base for the visual conception related to the learning task. Adding superpixelized images to the original training data makes the training data distribution have a more generic base that can potentially better support unseen test images. In **Appendix A.2**, using variational auto-encoders (VAEs) (Kingma and Welling, 2013) and the Kullback-Leibler divergence (Kullback and Leibler, 1951), we show that the training set with SPDA-generated data is closer to the test set in terms of the overall data distribution.

2.3. Choosing a superpixel generation method

Boundary recall and compactness are two key criteria for generation of superpixels. Boundary recall evaluates how well the generated superpixels represent or cover the important object boundaries/contours in an image. Compactness describes how regular and well-organized the superpixels are. Compactness of superpixels tends to constrain superpixels to fit some irregular and subtle object boundaries. In general, one aims to generate superpixels with high boundary recall and high compactness.

For deep learning model training, we aim to generate superpixels with the following properties: (i) good boundary recall, (ii) being compact and pixel-like, and (iii) only pixel values and local image features are used to generate superpixels. Note that many fully convolutional networks work in a bottom-up fashion; superpixels that are generated using global-level information may confuse the training of an FCN model. Hence, we prefer to use superpixel generation method that only utilizes local image information for the pixel grouping process.

SLIC (Achanta et al., 2012) is one of the most widely used methods for generating superpixels. SLIC is fast to compute and can produce good quality superpixels with an option to let the user control the compactness of the generated superpixels. Also, SLIC utilizes only local image information for grouping pixels into superpixels, which is a desired feature by SPDA for training deep learning models. Thus, in our experiments, we use SLIC (Achanta et al., 2012) to generate superpixels for our superpixel-based data augmentation method. The added computational cost for applying SLIC to every training sample is very small comparing to the model training time cost.

1. Being close means the distance (e.g., Euclidean distance) between an SPDA-generated image and its corresponding raw image is smaller than the distance between this raw image and any other raw image.

2.4. Generating superpixelized images

Suppose the given training set contains n training samples $(x_i, y_i), i = 1, 2, \ldots, n$, where x_i is a raw image and y_i is its corresponding annotation map. We apply a superpixel generation method (e.g., SLIC (Achanta et al., 2012)) $F(x_i, s)$ to each image x_i to obtain superpixel cells $c_j^i, j = 1, 2, \ldots, s$. Each superpixel cell contains a connected set of pixels. Here, s is part of the input to F that specifies the desired number of superpixels that F should produce. We will discuss how to choose the values of s below. Any two different superpixel cells have zero common elements (pixels). The union of the pixels of all the superpixel cells for x_i is all the pixels in the image x_i.

To generate a superpixelized image for x_i, for each superpixel cell c_j^i, we compute the mean values of all the pixels in c_j^i and update the values of all the pixels in c_j^i using such computed mean values. This step aims to erase low-level pixel variance so that the mid-level and high-level information can be better emphasized by the superpixelized images. We repeat this process for all the superpixel cells of x_i, and then form a superpixelized image $SP(x_i, s)$, where s indicates that this superpixelized image is generated using s superpixels. To avoid artificially changing the distribution of annotation (label) maps, the annotation map for $SP(x_i, s)$ is kept as the original y_i. Thus, we put $(SP(x_i, s), y_i)$ into our new training data set generated using (x_i, y_i).

The value s specifies the desired number of superpixels to generate. A small number of superpixels would make a superpixelized image too coarse to represent the essential object structures in the original image. A large number of superpixels would make a superpixelized image too similar to the original image. We aim to model a relatively continuous change from each original image sample to its superpixelized images, from fine to coarse, so that the VRM distribution (or neighborhood distribution) around the original image sample can be better captured. As a result, we choose a range $[s_l, s_u]$ of values for s, and form a set of superpixelized images $SP(x_i, s)$, $s = s_l, \ldots, s_u$, for each original image x_i.

For biomedical image datasets, the imaging settings are usually known in practice. In particular, the scales and size of the images, and the range of sizes of objects in the images are often known. Thus, one can set the values of s_l and s_u based on prior knowledge of these image aspects. For different image sets and applications, one can set s_l and s_u differently. In our experiments, for simplicity and for demonstrating the robustness of SPDA, we choose a common setting of s_l and s_u for all the 2D segmentation datasets ($s_l = 800$ and $s_u = 2000$). For the 3D image dataset, due to the increase of image dimensionality, we set $s_l = 2000$ and $s_u = 4000$.

2.5. Model training using SPDA

The loss function for training a deep learning based segmentation network using both the original training data and the augmented data is:

$$\frac{1}{n}\sum_{i=1}^{n}(\mathscr{L}(f(x_i), y_i) + \lambda \sum_{s=s_l}^{s_u} \mathscr{L}(f(SP(x_i, s)), y_i)) \tag{5}$$

where $\mathscr{L}$ is a spatial cross-entropy loss, f is the segmentation model under training, SP is for generating a superpixelized image, and s is a parameter for SP that specifies how many superpixels are desired to be generated. We set λ as simple as a normalization term $\frac{1}{s_u - s_l + 1}$. We aim to minimize the above function with respect to the parameters of f.

A common way of optimizing the objective function above is to use a mini-batch based stochastic gradient descent method. Following the loss function in Eq. (5), half of the total samples in the

mini-batch is drawn from the original image samples and the other half is from the SPDA-generated samples. We provide the pseudo-code (**Algorithm 1**) for the model training procedure below.

Algorithm 1: Model training using SPDA-augmented training data

Data: (x_i, y_i) and $(SP(x_i, s), y_i)$, $i = 1, 2, \ldots, n$ and $s = s_l, \ldots, s_u$.
Result: A trained FCN model.
Initialize an FCN model with random weights, mini-batch = $\emptyset$;
while *stopping condition not met* **do**
  for m = *1 to batch-size*$/2$ **do**
    $p = random.randint(1, n)$;
    add (x_p, y_p) to the mini-batch;
    $k = random.randint(s_l, s_u)$;
    add $(SP(x_p, k), y_p)$ to the mini-batch;
  end
  Update FCN using data in the mini-batch using the Adam optimizer;
  mini-batch = $\emptyset$;
end

3. Experiments

Four biomedical image segmentation datasets are used to evaluate our SPDA method. These datasets are: (1) 3D magnetic resonance (MR) images of myocardium and great vessels (blood pool) in cardiovascular (Pace et al., 2015), (2) electron micrographs (EM) of neuronal structures (Lee et al., 2015), (3) an in-house 2D electron micrographs (EM) of fungal cells that invade animal (ant) tissues, and (4) 2D H&E stained histology images of glands (Sirinukunwattana et al., 2017). Note that SPDA can be extended to segmentation of 3D images using a straightforward extension of SLIC (Achanta et al., 2012) that generates supervoxels instead of superpixels.

On the 2D segmentation datasets, our experiments of SPDA use two common FCN models: U-Net (Ronneberger et al., 2015) and DCN (Chen et al., 2016a). In addition to showing the effectiveness of SPDA, on the neuronal structure and fungus datasets, we also compare SPDA with the elastic deformation for data augmentation (EDDA) used in (Ronneberger et al., 2015). On the 3D segmentation dataset, a state-of-the-art DenseVoxNet (Yu et al., 2017) is utilized for experiments with our SPDA. Experiments on this 3D dataset aim to show the capability of SPDA for 3D image data.

We made a simple extension of the original DCN model (Chen et al., 2016a), which now contains 5 max-pooling layers (deeper than the original DCN) (see Fig. 3). The extension allows DCN to have a larger receptive field for making use of higher-level image information. Random cropping, flipping, and rotation are applied as standard/basic data augmentation operations to all the instances in the experiments. We denote this set of basic data augmentation operations as DA_{basic}. For fair comparison, we keep all the training settings (e.g., random seed, learning rate, mini-batch size, etc) the same for all the model training. Adam (Kingma and Ba, 2014) optimizer is used for model optimization. As in a common practice, the learning rate for model training is set as 0.0005 for the first 30000 iterations, and then decays to 0.00005 for the rest of the training. The mini-batch size is set as 8. The compactness parameter for SLIC (Achanta et al., 2012) is set as its default value 20. The

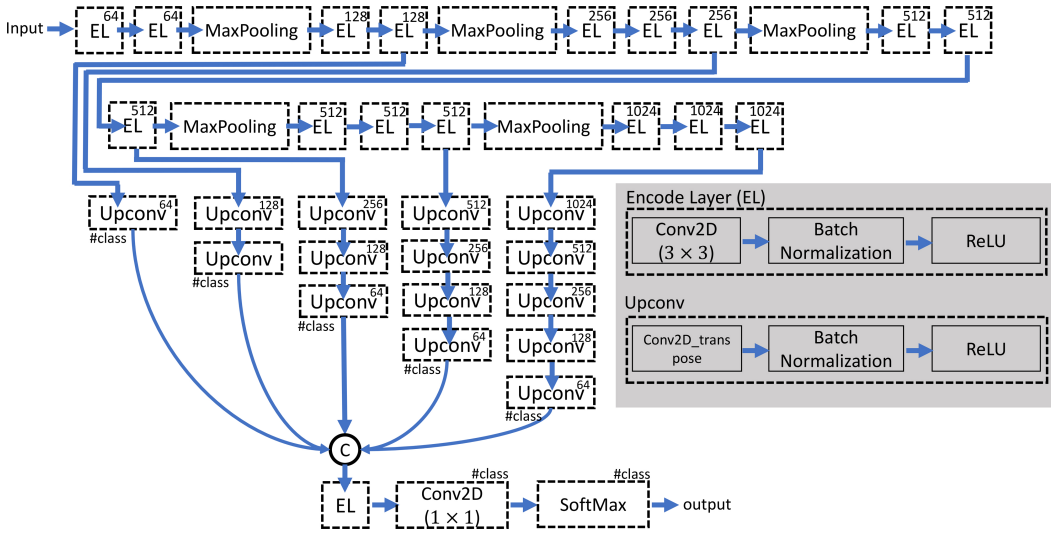

Figure 3: The architecture of the deeper DCN model (which we modify from the original DCN (Chen et al., 2016a)).

size of the input and output of an FCN model is set as 192×192 for 2D images and $64 \times 64 \times 64$ for 3D images. The training procedure stops its execution when there is no significant change in the training errors.

3D cardiovascular segmentation. The HVSMR dataset (Pace et al., 2015) was used for segmenting myocardium and great vessels (blood pool) in 3D cardiovascular magnetic resonance (MR) images. The original training dataset contains 10 3D MR images, and the test data consist of another 10 3D MR images. The ground truth of the test data is not available to the public; the evaluations are done by submitting segmentation results to the organizers' server.

The Dice coefficient, average distance of boundaries (ADB), and symmetric Hausdroff distance are the criteria for evaluating the quality of the segmentation results. A combined score S, computed as $S = \sum_{class}(\frac{1}{2}Dice - \frac{1}{4}ADB - \frac{1}{30}Hausdorff)$, is used by the organizers, and this score aims to measure the overall quality of the segmentation results.

When applying SPDA to 3D image data, supervoxels (instead of superpixels) are generated. We use a 3D version of SLIC for generating supervoxels. SPDA is tested using the DenseVoxNet (Yu et al., 2017), which is a state-of-the-art FCN for 3D voxel segmentation. In Table 1, we show the results from DenseVoxNet + DA_{basic}, DenseVoxNet + DA_{basic} + SPDA, and other known models on this dataset. One can see that SPDA improves the segmentation results significantly, especially on the average distance of boundaries and Hausdroff distance metrics.

Fungal segmentation. We further evaluate SPDA using an in-house EM fungus dataset that contains 6 large 2D EM images (4000×4000 each) for segmentation. Since the input window size of a fully convolutional network is set as 192×192, there are virtually hundreds and thousands unique image samples for model training and testing. This dataset contains three classes of objects of interest: fungal cells, muscles, and nervous tissue. We use 1 large microscopy image for training, and 5 large microscopy images for testing. This experiment aims to evaluate the effectiveness of SPDA in a difficult situation in which the training set is smaller than the test set (not uncommon

Table 1: Comparison of segmentation results on the HVSMR dataset.

Method	Myocardium			Blood pool			Overall score
	Dice	ADB	Hausdorff	Dice	ADB	Hausdorff	
3D U-Net (Çiçek et al., 2016)	0.694	1.461	10.221	0.926	0.940	8.628	−0.419
VoxResNet (Chen et al., 2018)	0.774	1.026	6.572	0.929	0.981	9.966	−0.202
DenseVoxNet (Yu et al., 2017) + DA_{basic}	**0.821**	0.964	7.294	0.931	0.938	9.533	−0.161
DenseVoxNet (Yu et al., 2017) + DA_{basic} + SPDA	0.817	**0.723**	**3.639**	**0.938**	**0.778**	**5.548**	**0.196**

in biomedical image segmentation). In Table 2, Student's t-test suggests that all our improvements are significant. The p-values for MeanIU of U-Net vs U-Net + SPDA, U-Net + EDDA vs U-Net + SPDA, DCN vs DCN + SPDA, and DCN + EDDA vs DCN + SPDA are all < 0.0001.

Table 2: Comparison results on the fungus segmentation dataset: The Intersection-over-Union (IoU) scores for each object class and the MeanIU scores across all the classes of objects. U-Net (Ronneberger et al., 2015), DCN (Chen et al., 2016a), and EDDA: elastic deformation data augmentation (used in (Ronneberger et al., 2015)) are considered.

Method	Fungus	Muscle	Nervous tissue	MeanIU
U-Net + DA_{basic}	0.849 ± 0.008	0.976 ± 0.003	0.506 ± 0.029	0.777 ± 0.008
U-Net + DA_{basic} + EDDA	0.881 ± 0.007	0.975 ± 0.004	0.549 ± 0.035	0.8019 ± 0.014
U-Net + DA_{basic} + SPDA	0.927 ± 0.001	0.973 ± 0.002	0.667 ± 0.020	$\mathbf{0.856 \pm 0.007}$
DCN + DA_{basic}	0.783 ± 0.064	0.970 ± 0.009	0.349 ± 0.092	0.701 ± 0.055
DCN + DA_{basic} + EDDA	0.863 ± 0.042	0.970 ± 0.008	0.453 ± 0.183	0.762 ± 0.078
DCN + DA_{basic} + SPDA	0.907 ± 0.011	0.973 ± 0.005	0.630 ± 0.026	$\mathbf{0.837 \pm 0.012}$

Neuronal structure segmentation. We experiment with SPDA using the EM mouse brain neuronal images (Lee et al., 2015). This dataset contains 4 stacks of EM images (1st: $255 \times 255 \times 168$, 2nd: $512 \times 512 \times 170$, 3rd: $512 \times 512 \times 169$, and 4th: $256 \times 256 \times 121$). Following the practice in (Lee et al., 2015; Shen et al., 2017), we use the 2nd, 3rd, and 4th stacks for model training and the 1st stack for testing. Since the image stacks in this dataset are highly anisotropic (i.e., the voxel spacing along the z-axis is much larger than those along the x- and y-axes), directly applying 3D models with 3D convolutions is not very suitable for highly anisotropic 3D images. Hence, for simplicity, our experiments on this dataset are based on superpixels in the 2D slices of the 3D images and using 2D FCN models, instead of supervoxels and 3D models. We run all experiments 5 times with different random seeds. The average performance across all the runs and their standard deviations are reported in Table 3. Student's t-test suggests that all our improvements are significant. The p-values for V^{Rand}_{Fscore} are: < 0.0001 for U-Net vs U-Net + SPDA, 0.0059 for U-Net + EDDA vs U-Net + SPDA, < 0.0001 for DCN vs DCN + SPDA, and 0.0042 for DCN + EDDA vs DCN + SPDA.

Gland segmentation. This H&E stained microscopy image dataset (Sirinukunwattana et al., 2017) contains 85 training images (37 benign (BN), 48 malignant (MT)) and 60 testing images (33 BN, 27 MT) in part A, and 20 testing images (4 BN, 16 MT) in part B. Table 4 shows the gland segmentation results that demonstrate the effect of SPDA and comparison with the state-of-the-art

Table 3: Comparison results on the neuronal structure segmentation dataset: V^{Rand} scores for evaluating the segmentation quality. DA_{basic}: basic data augmentation operations (random cropping, flipping, and rotation); EDDA: elastic deformation data augmentation in (Ronneberger et al., 2015).

Method	V^{Rand}_{merge}	V^{Rand}_{split}	V^{Rand}_{Fscore}
M^2FCN (Shen et al., 2017)	0.9917	0.9815	0.9866
U-Net (Ronneberger et al., 2015) + DA_{basic}	0.9954 ± 0.0003	0.9879 ± 0.0001	0.9917 ± 0.0001
U-Net + DA_{basic} + EDDA	0.9957 ± 0.0005	0.9931 ± 0.0003	0.9944 ± 0.003
U-Net + DA_{basic} + SPDA	**0.9965 ± 0.0003**	**0.9935 ± 0.0004**	**0.9950 ± 0.0002**
DCN (Chen et al., 2016a) + DA_{basic}	0.9950 ± 0.0003	0.9916 ± 0.0001	0.9933 ± 0.0001
DCN + DA_{basic} + EDDA	0.9980 ± 0.0006	0.9917 ± 0.0001	0.9949 ± 0.0002
DCN + DA_{basic} + SPDA	**0.9987 ± 0.0010**	**0.9921 ± 0.0006**	**0.9954 ± 0.0002**

Table 4: Comparison results on the gland segmentation dataset: The F_1 score and ObjectDice evaluate how well glands are segmented at the instance level, and Object Hausdorff distance evaluates the shape similarity between the segmented objects and ground truth objects.

Method	F_1 Score		ObjectDice		ObjectHausdorff	
	part A	part B	part A	part B	part A	part B
CUMedVision (Chen et al., 2016b)	0.912	0.716	0.897	0.718	45.418	160.347
Multichannel1 (Xu et al., 2016b)	0.858	0.771	0.888	0.815	54.202	129.930
Multichannel2 (Xu et al., 2016a)	0.893	0.843	0.908	0.833	44.129	116.821
MILD-Net (Graham et al., 2018)	0.914	0.844	**0.913**	0.836	**41.54**	105.89
U-Net (Ronneberger et al., 2015) + DA_{basic}	0.89202	0.8087	0.88193	0.83441	51.19	108.25
U-Net + DA_{basic} + SPDA	0.9007	0.83843	0.88429	0.8415	49.95	107.69
DCN (Chen et al., 2016a) + DA_{basic}	0.9071	0.825	0.898	0.826	48.740	126.479
DCN + DA_{basic} + SPDA	**0.918**	**0.860**	**0.913**	**0.858**	42.620	**95.83**

models on this dataset. In particular, using SPDA, DCN can be trained to perform considerably better than the state-of-the-art model (Graham et al., 2018).

4. Conclusions

In this paper, we presented a new data augmentation method using superpixels (or supervoxels), SPDA, for training fully convolutional networks for biomedical image segmentation. Our proposed SPDA method is well motivated, easy to use, compatible with known data augmentation techniques, and can effectively improve the performance of deep learning models for biomedical image segmentation tasks.

References

Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Transac-*

tions on Pattern Analysis and Machine Intelligence, 34(11):2274–2282, 2012.

Antreas Antoniou, Amos Storkey, and Harrison Edwards. Data augmentation generative adversarial networks. *arXiv preprint arXiv:1711.04340*, 2017.

Olivier Chapelle, Jason Weston, Léon Bottou, and Vladimir Vapnik. Vicinal risk minimization. In *Advances in Neural Information Processing Systems*, pages 416–422, 2001.

Hao Chen, Xiaojuan Qi, Jie-Zhi Cheng, Pheng-Ann Heng, et al. Deep contextual networks for neuronal structure segmentation. In *AAAI*, pages 1167–1173, 2016a.

Hao Chen, Xiaojuan Qi, Lequan Yu, and Pheng-Ann Heng. DCAN: Deep contour-aware networks for accurate gland segmentation. In *CVPR*, pages 2487–2496, 2016b.

Hao Chen, Qi Dou, Lequan Yu, Jing Qin, and Pheng-Ann Heng. VoxResNet: Deep voxelwise residual networks for brain segmentation from 3D MR images. *NeuroImage*, 170:446–455, 2018.

Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3D U-Net: Learning dense volumetric segmentation from sparse annotation. In *MICCAI*, pages 424–432, 2016.

Zach Eaton-Rosen, Felix Bragman, Sebastien Ourselin, and M Jorge Cardoso. Improving data augmentation for medical image segmentation. In *MIDL*, 2018.

Simon Graham, Hao Chen, Qi Dou, Pheng-Ann Heng, and Nasir Rajpoot. MILD-Net: Minimal information loss dilated network for gland instance segmentation in colon histology images. *arXiv preprint arXiv:1806.01963*, 2018.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

Solomon Kullback and Richard A Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

Kisuk Lee, Aleksandar Zlateski, Vishwanathan Ashwin, and H Sebastian Seung. Recursive training of 2D-3D convolutional networks for neuronal boundary prediction. In *NIPS*, pages 3573–3581, 2015.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

Danielle F Pace, Adrian V Dalca, Tal Geva, Andrew J Powell, Mehdi H Moghari, and Polina Golland. Interactive whole-heart segmentation in congenital heart disease. In *MICCAI*, pages 80–88, 2015.

Luis Perez and Jason Wang. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*, 2017.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241, 2015.

Wei Shen, Bin Wang, Yuan Jiang, Yan Wang, and Alan Yuille. Multi-stage multi-recursive-input fully convolutional networks for neuronal boundary detection. In *ICCV*, pages 2410–2419, 2017.

Korsuk Sirinukunwattana, Josien PW Pluim, Hao Chen, Xiaojuan Qi, Pheng-Ann Heng, Yun Bo Guo, Li Yang Wang, Bogdan J Matuszewski, Elia Bruni, et al. Gland segmentation in colon histology images: The GlaS challenge contest. *Medical Image Analysis*, 35:489–502, 2017.

Yu-Xiong Wang, Ross Girshick, Martial Hebert, and Bharath Hariharan. Low-shot learning from imaginary data. *arXiv preprint arXiv:1801.05401*, 2018.

Yan Xu, Yang Li, Mingyuan Liu, Yipei Wang, Yubo Fan, Maode Lai, Eric I Chang, et al. Gland instance segmentation by deep multichannel neural networks. *arXiv preprint arXiv:1607.04889*, 2016a.

Yan Xu, Yang Li, Mingyuan Liu, Yipei Wang, Maode Lai, I Eric, and Chao Chang. Gland instance segmentation by deep multichannel side supervision. In *MICCAI*, pages 496–504, 2016b.

Lequan Yu, Jie-Zhi Cheng, Qi Dou, Xin Yang, Hao Chen, Jing Qin, and Pheng-Ann Heng. Automatic 3D cardiovascular MR segmentation with densely-connected volumetric ConvNets. In *MICCAI*, pages 287–295, 2017.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. Mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

Appendix A. Empirical Studies of SPDA-generated Data

In this appendix, two empirical studies are conducted to show: (1) the SPDA-generated data are "near" their original image data in the data space, and (2) the data distribution of the SPDA-augmented training set is "closer" to the distribution of the test data (or true data).

A.1. SPDA-generated data near their original samples

We seek to examine how the SPDA-generated data are spatially close to their corresponding original images. Since $SP(x_i, s)$ and x_i are both in a high dimensional space, comparing them is not a trivial task. One may use a distance metric for measuring the distance between $SP(x_i, s)$ and x_i. However, with different metrics, the meaning of "being different" or "being similar" can be drastically different.

To avoid too much complication in manifold learning or metric learning, we use two common dimensionality reduction methods, standard PCA and t-SNE (Maaten and Hinton, 2008), to help visualize the original image samples and SPDA-generated image samples. Fig. 4 shows visualization results of such samples (both the original and SPDA-generated samples) on the neuronal structure dataset, fungus dataset, and gland dataset (after applying PCA). One may observe that the SPDA-generated data are near/surrounding the original image data, forming a close neighborhood of the original images. In Fig. 5, we provide visualization views of some SPDA-generated image samples using t-SNE (Maaten and Hinton, 2008).

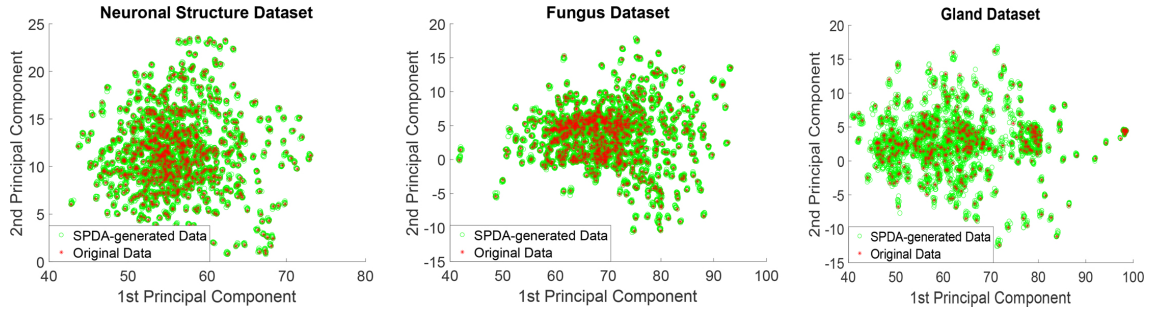

Figure 4: After dimensionality reduction using PCA, each original image sample (red) is surrounded by (or closely adjacent to) its corresponding superpixelized images (green). Zoom-in view would show more details.

A.2. Data distribution comparison

Here we are interested in a basic question: Whether adding SPDA-generated data X_{spda} to the original training set X_{ori} makes the new training set $X_{augmented}$ "closer" to the test data X_{test} in the image representation space.

We utilize variational auto-encoders (VAEs) (Kingma and Welling, 2013) to encode the training images $X = \{x_1, \ldots, x_n\}$ into much lower dimensional representation $Z = \{z_1, \ldots, z_n\}$. On each dimension of the space thus resulted, the data are expected to follow a Gaussian distribution with zero mean and unit variance. This is a standard objective of VAE.

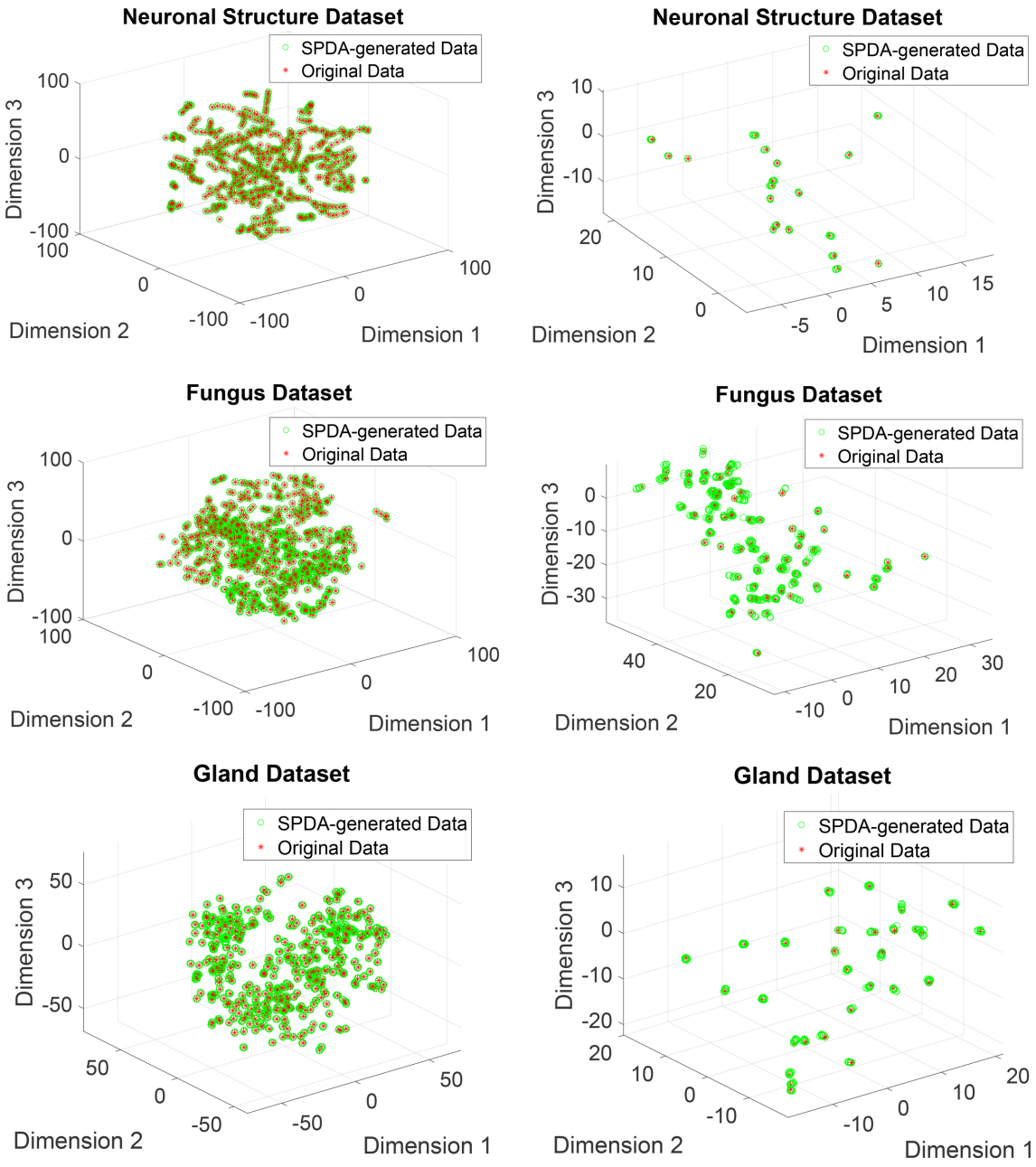

Figure 5: Visualization of some SPDA-generated image samples (green) and the original training samples (red) using t-SNE (Maaten and Hinton, 2008). Left: Overview of the samples; right: zoom-in views. SPDA-generated samples are in a close neighborhood of their corresponding original samples.

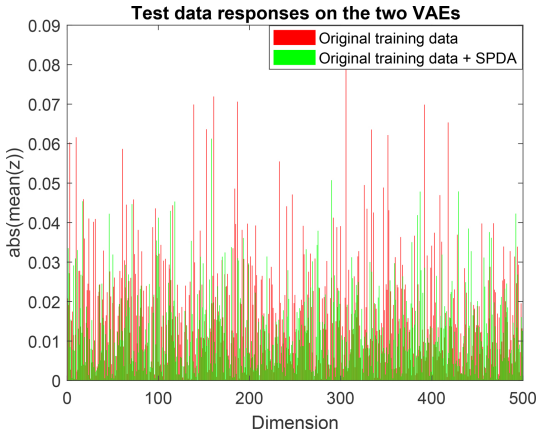

Figure 6: The responses of the test data on the two VAEs trained using the original training data and the SPDA-augmented training data (on the neuronal structure dataset).

To show the effect of SPDA, we train two VAEs: VAE-A is trained using only the original training images X_{ori}, and VAE-B is trained using the SPDA-augmented training set $X_{augmented} = X_{ori} \cup X_{spda}$. These two VAEs are all trained using the same settings; the only difference is their training data. After training, VAE-A is applied to its training data X_{ori} and the test data X_{test}, to obtain Z^A_{ori} and Z^A_{test}. Similarly, VAE-B is applied to its training data $X_{augmented}$ and the test data X_{test}, and $Z^B_{augmented}$ and Z^B_{test} are obtained. We then compare

$$D_{KL}(P(z^A_{test})||P(z^A_{ori})) \tag{6}$$

with

$$D_{KL}(P(z^B_{test})||P(z^B_{augmented})) \tag{7}$$

and compare

$$D_{KL}(P(z^A_{ori})||P(z^A_{test})) \tag{8}$$

with

$$D_{KL}(P(z^B_{augmented})||P(z^B_{test})) \tag{9}$$

where D_{KL} is the Kullback-Leibler divergence (Kullback and Leibler, 1951) and $P(z)$ is the probability distribution of z. The above procedure is applied to the neuronal structure dataset. The results are:

$$D_{KL}(P(z^B_{test})||P(z^B_{augmented})) = 5.5517 < D_{KL}(P(z^A_{test})||P(z^A_{ori})) = 5.8779,$$

and

$$D_{KL}(P(z^B_{augmented})||P(z^B_{test})) = 5.0586 < D_{KL}(P(z^A_{ori})||P(z^A_{test})) = 6.1491.$$

It is clear that SPDA can potentially make the training data distribution closer to the test data/true data distribution in the image representation space. We believe this is a main reason why learning models trained using SPDA-augmented training data can generalize better on test data. To show this observation visually, the absolute values of the averages of Z^A_{test} and Z^B_{test} are shown in Fig 6. One

can see that the values of Z^B_{test} are generally closer to 0 than Z^A_{test}, which means that the distribution of Z^B_{test} is closer to the zero mean Gaussian distribution, and thus is closer to the distribution of $Z^B_{augmented}$.