Semantic Aware Data Augmentation for Cell Nuclei Microscopical Images with Artificial Neural Networks

Alireza Naghizadeh
Rutgers University
Newark, NJ
ar.naghizadeh@rutgers.edu

Hongye Xu
Rutgers University
Newark, NJ
hx123@njms.rutgers.edu

Mohab Mohamed
Rutgers University
Newark, NJ
mm2233@scarletmail.rutgers.edu

Dimitris N. Metaxas
Rutgers University
New Brunswick, NJ
dnm@cs.rutgers.edu

Dongfang Liu
Rutgers University
Newark, NJ
dongfang.liu@rutgers.edu

Abstract

There exists many powerful architectures for object detection and semantic segmentation of both biomedical and natural images. However, a difficulty arises in the ability to create training datasets that are large and well-varied. The importance of this subject is nested in the amount of training data that artificial neural networks need to accurately identify and segment objects in images and the infeasibility of acquiring a sufficient dataset within the biomedical field. This paper introduces a new data augmentation method that generates artificial cell nuclei microscopical images along with their correct semantic segmentation labels. Data augmentation provides a step toward accessing higher generalization capabilities of artificial neural networks. An initial set of segmentation objects is used with Greedy AutoAugment to find the strongest performing augmentation policies. The found policies and the initial set of segmentation objects are then used in the creation of the final artificial images. When comparing the state-of-the-art data augmentation methods with the proposed method, the proposed method is shown to consistently outperform current solutions in the generation of nuclei microscopical images.

1. Introduction

The ability to image, extract and study cells is essential to various research areas within the medical field. Advancements in high-resolution fluorescent microscopy have allowed medical professionals access to a more detailed visualization of cells and their interactions [17]. A prime example is within immunotherapy, where there exists great importance to assess the efficacy of different treatments for fatal illnesses such as cancer, or more contemporarily, HIV/AIDS. Automated analysis of cellular images enables medical researchers to avoid time-consuming quantification, vastly improving the speed at which it takes to quantify reproducible mass data. The development of strong artificial neural networks for semantic segmentation and object detection should naturally accompany the automation process.

However, the large datasets required to train models of Artificial Neural Networks (ANNs) [21] greatly slows down researchers due to the naturally lengthy process of acquiring ground truth images. Creating training data from microscopical images requires expertise and precision to separate nuclei cells. If there is a method to reliably automate the ground truth generation of nuclei cells, it can prevent common human errors. The requirement for experts in the field can be reduced hugely. It saves time, saves resources, and most importantly by generating a huge number of training data, can help the learning of ANNs for related biological fields in a way that is not possible with current technologies. The solution that we are looking for is to create a methodology that not only generates artificially augmented images but is also semantically aware of the objects within the images.

Unfortunately, current methods in machine learning which deal with artificially augmenting training data such as different forms of Generative Adversarial Networks (GANs) [9] do not deal with generating semantically aware labels for the objects. They also have other problems that make them unsuitable for generating training data in medical fields. To properly train GAN networks, we usually need a large set of training data, which is not always available in microscopical nuclei imaging. There are several attempts to solve this problem, such as [11, 26, 38, 37, 41]. DeLiGan parametrizes the latent generative space and learns the model's parameters along with those of GAN [11]. Noguchi and Wang apply the principles of transfer learning to a pretrained generator in order to compensate for the lack of a larger dataset [26]. Another approach is to directly apply commonly-used augmentation techniques such as cropping, flipping, scaling, color jittering, and region masking [13, 12, 8, 15] on the training data to increase its variety.

A more important issue that current models of data augmentation may have with nuclei cell images originates from the nature of this type of data. Given that medical imaging deals with highly sensitive data, uncontrolled, noisy, and unreliable; artificial images can potentially lead to disastrous outcomes. For instance, in natural images, creating unidentified objects that look real is always exciting. However, for immunotherapists that are treating cancer patients, using such training sets may result in not correctly evaluating an immune response within a cancer patient.

To address these problems, we propose semantically aware data augmentation for cell nuclei microscopical images. Our main objective is the generation of artificial images that provide complete semantic segmentation information for the biomedical field. To generate artificial images, we use standard augmentation techniques, which provide the required control over cell objects that are needed to generate accurate cells with their precise semantic information. This method is possible because of the unique forms of cell nuclei microscopical images. In contrast to real-world images with dozens of complex interconnected objects, cell nuclei images have much more simplistic forms with having focus on specific cell objects and mostly recognizable background patterns that separate the cells from one another.

To generate images, we first need an initial set of images that are segmented by experts within the field. We consider two types of cells from the available objects. The first type can be identified as single-cells, which are easily distinguished from other cells. The second type is called multi-cells, which may suffer from low contrast of cell boundaries, background noise (impurities), adhesion, and cell clustering. We show that if these two categories are used to generate images, we can create artificial images that are not distinguishable from natural data. Specifically, normal augmentation techniques are applied to the segmented single-cells and multi-cells. In order to maintain the integrity of the original cells, only a subset of controlled augmentation techniques are used. To apply an augmentation onto the cells, we use an augmentation policy [6, 7, 45]. To find the best policies, a Wasserstein Autoencoder [35] provides a scoring system for Greedy AutoAugment [24, 29], which ranks different combinations of sub-policies [34, 39, 27].

The best performing sub-policies are then used as final policies in creating training images. An image is generated by randomly selecting the number of cells within an image and applying a policy for each cell. The same procedure is used for the available masked images to create artificial images with complete ground truth information. In order to assess the quality of the generated images, we follow a similar approach found in [5]. To evaluate the quality of the images we use the Fréchet Inception Distance (FID) as well as the Kernel Inception Distance (KID), which has shown better support for smaller datasets [3].

2. Proposed Method

Let us define a set of semantic segmented objects $S = (x_0, ..., x_n)$ as an initial set of single-cells and multi-cells, with an infinitely large search space, P, containing different augmentation policies. Our goal is to find the best augmentation techniques that will output the most naturalistic nuclei images while maintaining complete segmentation information. In order to ensure that augmented images maintain natural properties, an AutoEncoder [22, 23, 30, 20] is trained that estimates an identity function, $f(S_i) \approx S_i$ for $i \in [0, n]$. The AutoEncoder gives a high-level evaluation of the quality of the augmented policies and creates the scoring criterion for Greedy AutoAugment [24]. Greedy AutoAugment is an efficient search algorithm that finds the best augmentation policies within an arbitrarily large sample space. After finding the best policies for single-cells and multi-cells objects, the cells accept policies and are placed pseudorandomly in a carefully defined frame.

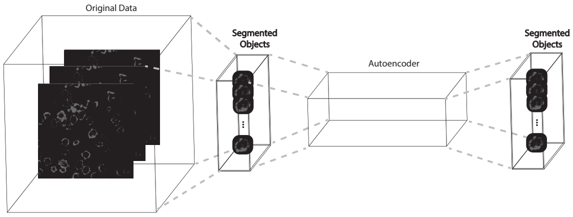


Figure 1: AutoEncoder finds the distribution of cells. It is used as the scoring criterion for Greedy AutoAugment.

2.1. AutoEncoder

The autoencoder attempts to learn a function $f(S_i)$, where S_i is a set of segmented cells that maps S_i to itself. S_i contains all of the segmented objects from the single-cells and multi-cells. This is done to learn the overall distribution of the individual cells. This process is shown in Figure 1. The segmented cells, which are obtained from the original training data, are separated as sampled objects (S_i). An AutoEncoder is used to map S_i to S_i, to find the distribution of S_i. After training for a specific number of epochs, the next step is to use the trained AutoEncoder to create a criterion for the purpose of ranking different augmentation policies.

Table 1: AutoEncoder finds the distribution of cells. It is used as the scoring criterion for Greedy AutoAugment algorithm.

	Technique	Description		Technique	Description
1.	FlipLR	Filliping the image along the vertical axis.	7.	Contrast	Changing the contrast of the image.
2.	FlipUD	Filliping the image along the horizontal axis.	8.	Brightness	Adjusting the brightness of the image.
3.	AutoContrast	Increasing the contrast of the image.	9.	Sharpness	Adjusting the sharpness of the image.
4.	Equalize	Equalizing the histogram of the image.	10.	Smooth	Smoothing the image(Low-pass filtering).
5.	Rotate	Rotating the image by certain degrees.	11.	Resize	Changes the image resolution.
6.	Posterize	Reducing the number of Bits for each pixel.			

To determine the quality of a single policy, we apply the policy to all members of S_i. When a cell is augmented and passed through the AutoEncoder and maps to the same exact cell without applying the policy, its output can evaluate the quality of that policy. With this mechanism, we make sure that the original cells can be changed but maintain their integrity. The purpose of this is when the augmentation takes place, the policy must not alter the cell to an unrecognizable state. In the searching phase, the accuracy of $f(S_i)$ is passed as the scoring criterion for the Greedy Search to rank every explored policy.

2.2. Greedy AutoAugment

A policy is used to perform data augmentation on an image. Each policy has three essential elements, the augmentation technique, the magnitude of the operation, and the probability. Therefore, a search mechanism for finding the best augmentation techniques is a search space that should consider all possible combinations of these three elements. The number of augmentation techniques used in this paper is eleven (see Table 1). The magnitude is the degree to which an operation is applied. For instance, in the rotation augment, the magnitude specifies how much an image is rotated. Finally, the third element specifies the probability of applying the augmentation to the image.

Before searching for the best policy, we have to first discretize the space of probabilities and the magnitudes. The discretization of probabilities uses eleven values within uniform space, and the discretization of magnitudes uses ten values within uniform space. With this setting, the search space is simply $(20 \times 10 \times 11)$. This search space is defined to consider all possible combinations of elements in one sub-policy (the first search layer). If we expand the search space in order to find all the possible combinations for two sub-policies, the search space increases its size to $(20 \times 10 \times 11)^2$ (two search layers). Continuously, we can expand the search space for more layers infinitely. In general, we define $(20 \times 10 \times 11)^l$, where l is the number of layers within the search space.

To reduce the search space, Greedy AutoAugment considers the probability value of one, for all augmentation techniques. It is shown that this design would not hurt the end results [24]. Additionally, instead of considering all possible combinations, only the best combinations from the strongest sub-policies are considered. In other words, a greedy search algorithm is used, and expanding the search layers occurs only when it is required. Accordingly, the new number of the possibilities is defined as follows,

$$\sum_{1}^{k} (t_n \times m_n) \qquad (1)$$

In this notation, k is an arbitrary integer number, which indicates the number of iterations the algorithm is allowed to perform within the search. The t and m variables represent the augmentation technique and magnitude. Respectively t_n and m_n are the maximum values for t and m. To augment the data, a selection of the policies are used among all of the searched policies based on a scoring criterion. With the selected policies, the training data can then be expanded with the new augmented data as much as is required.

2.3. Image Creation

Figure 2 demonstrates the overall process of creating the final augmented images. We processed the original images, along with their respective masks, which come with a set of original images that are semantically labeled by experts within the biomedical field, and then received two types of cell collections, single-cells collection and multi-cells collection. The set of noise collection is manually collected following expert's guidance. Adding noise helps to avoid overfitting within the created datasets. As mentioned before, a multi-cells is a segmentation object consisting of multiple cells with intersecting bounding boxes. In many cases, boundaries are not well-defined, and the use of these ambiguous borders in creating single-cells may lead to unnatural imagery. Although single-cells have well-defined boundaries, the use of these cells for the creation of multi-cells leads to the generation of cells that lack natural properties originally found within the raw data.

In order to address this issue, we distinguish the two different collections and act on them separately. When a multi-cells is placed inside the frame of an artificial image, we solely use the separated collection of multi-cells images.

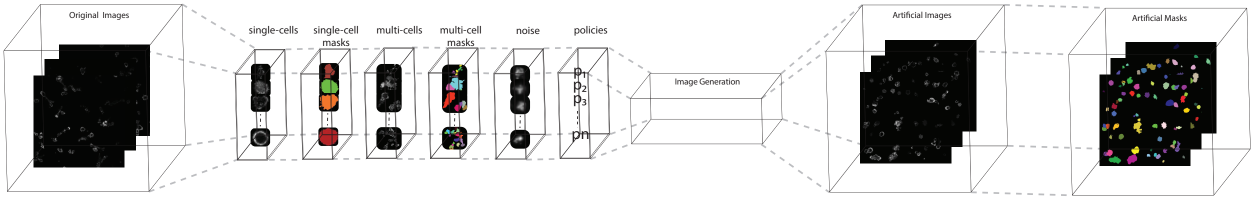


Figure 2: Pipeline describing the method used for image creation. Naturally acquired images are segmented by experts in biological fields. The best policies are found and selected through a greedy search algorithm using different policies scored via an autoencoder. The augmented cells (single-cells and multi-cells) are then placed pseudo-randomly onto a blank image to create an artificial image. The associated masks use the same procedure to create masks for artificial images and maintain the segmentation information of the cells.

On the other hand, if we want to place a single-cell inside the frame of an artificial image, we solely use the collection of single-cell images. To further streamline the process of creating realistic images, all of the single-cells and multi-cells objects that are too close to the boundaries of the original images are ignored. The next step uses these collected objects, along with distribution information, to create the final artificial images. The noise is applied to final images using the same procedure. In Figure 2, this is shown with examples of artificial images and masks.

When creating an image, we mimic the distribution of the number of cells (and noise) within the training data and the ratio of multi-cells to singular cells. For example, let j denote the mode of the distribution representing the number of cells found within our original data, then j policies are selected from the Greedy AutoAugment and applied to j original cells. These newly augmented cells are then applied in a controlled manner onto a blank image, avoiding overlaps while maintaining the ratio between multi and singular cells. The exact procedure is applied to the corresponding masks of the cell objects. Thus, creating an artificial image with natural properties and complete segmentation information. This process is done until we get r images with a matching distribution to the number of cells found within our original data [46].

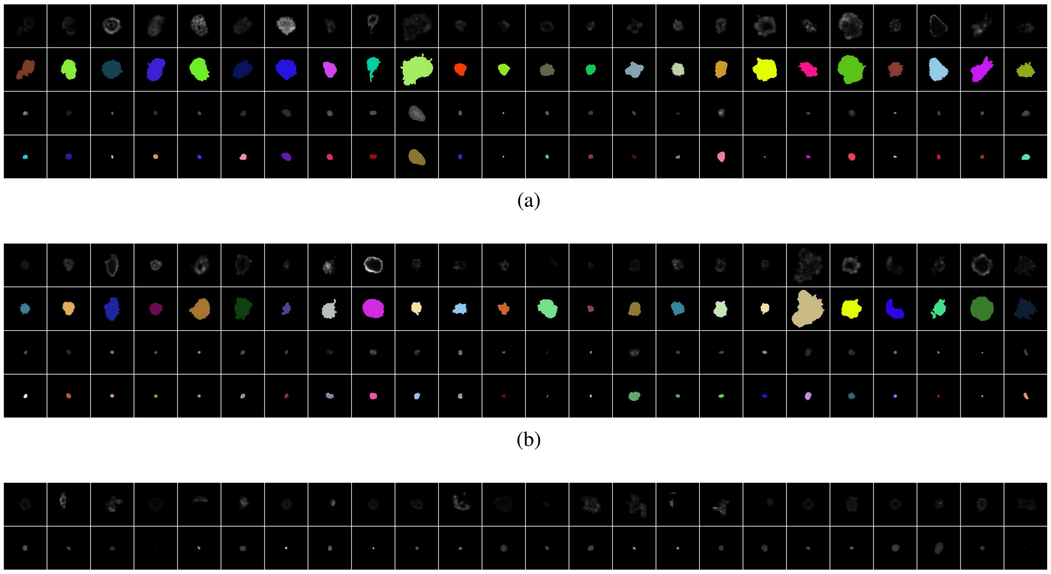
3. Experiments

We use the proposed method, to generate artificially augmented images on four prominent GAN models. The DCGAN is specifically designed to generate synthetic images for large-scale datasets [31, 32]. The small dataset BigGAN (BigGAN-SD) is one of the first serious attempts to introduce GAN for small datasets [4]. Recently, several papers used data augmentation to improve GAN for small datasets [2, 43, 36]. Among these attempts, we selected Data-Efficient GAN, which is more optimized for smaller datasets. There are two variants of Data-Efficient GAN for two state-of-the-art methods, one is StyleGAN2 [14], and the other is BigGAN [4]. In the paper, we call these two variants BigGAN-Diff and StyleGAN2-Diff. While these models are not designed to maintain semantic segmentation information for their image objects, they can still be used to compare the quality of the images created from our method.

Evaluation metrics We use two known evaluation metrics that are used for the GAN networks. Fréchet Inception Distance (FID) calculates the Fréchet distance between two multidimensional Gaussian distributions. This technique compares the distribution of the generated images, along with the mean and variances of the Gaussian distributions between artificial and original images. Kernel Inception Distance (KID) compares the two probability distributions by drawing samples independently from each distribution. This method improves FID and acts as a more reliable and unbiased estimator.

Experiment setup All of our experiments are conducted on K80 NVIDIA graphic cards. As for the embedded deep neural networks for our method, we used Pytorch [28]. For Greedy AutoAugment [24], we used the implementation in [25, 44, 42, 33] and for the AutoEncoder we used the Wasserstein variation [35, 10, 18]. For the experiments, all of the models are compared with their baseline released codes from the authors. We did not alter the suggested configuration from the official papers. To reduce computations, we search the search space of augmentation policies within the maximum number of 20000 policies, in which the first 1000 policies are used. All of the GAN models were trained for 150000 epochs across all experiments.

We mainly use two cell nuclei datasets containing microscopical images obtained from two different biological fields. The CAR-T cell dataset consists of 156 images at a resolution of 1024×1024. This dataset was collected by Rutgers Cancer Institute of New Jersey using microscope system A1R HD25 (Nikon, Japan), which provides a 25 mm field of view. The second dataset is the Kaggle 2018 Data Science Bowl (Bowl-18) [1], which contains around 589 images that were acquired under a variety of conditions



(a)

(b)

(c)

Figure 3: Single-cell image generation on CAR-T and Kaggle datasets; (a) The first two rows and last two rows represent the single-cell images and their masks from CAR-T dataset and Kaggle dataset, respectively. (b) shows the single-cell images and their associated masks that were generated by our method. The first two rows are the single-cell images generated from the CAR-T dataset and their masks. The last two rows are the single-cell images generated from the Kaggle dataset and their masks. (c) shows the single-cell images generated by StyleGan2-Diff. In the generation of single-cell images, StyleGan2-Diff had the best performance among selected GAN models.

and vary in the cell type, magnification, and imaging modality (brightfield vs. fluorescence). A subset of 193 images with consistent attributes is used.

3.1. Single-Cell Image Generation

Single-cell images are the most basic elements in the creation of complete artificial images. To create artificial single-cells, we use extracted segmentations from experts. The cells should then transform from its original state to a new state using an augmented policy. The outcome is an artificial cell that can be used to create artificial images. In this section, we measure the accuracy of these generated cells through the use of the evaluation methods discussed above (FID and KID). In other words, we answer how much applying different augmentation policies affects the quality of single-cells. The results is shown in Table 2. The average scores for two datasets are reported. The numerical values apply to the average scores and are computed over three random sets of sample generation. In the table, the proposed method is referred to as Sem-Aware (semantic-aware).

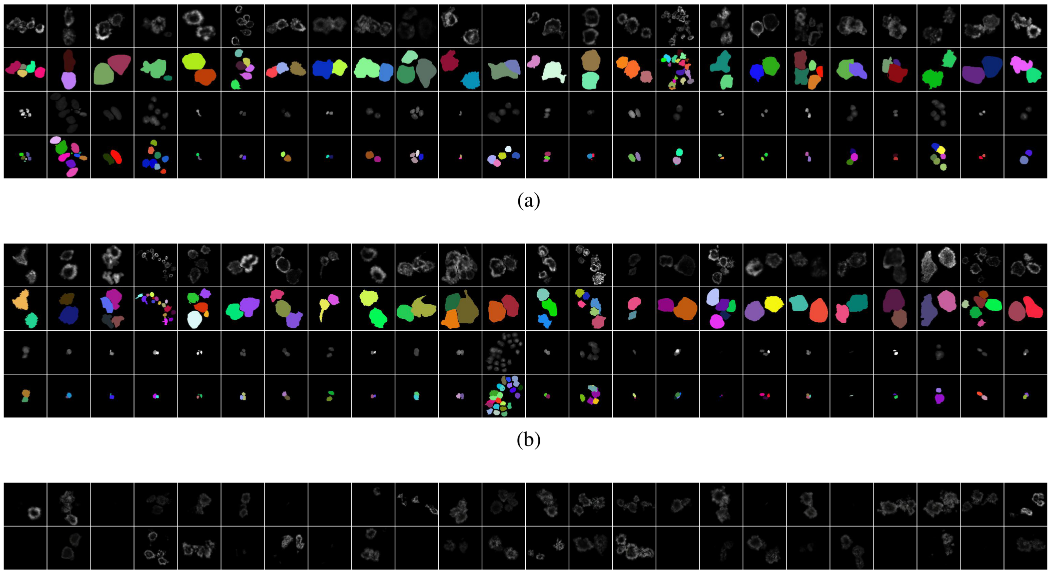
Fréchet Inception Distance The results show that our method performs competitively compared to existing models. The best performing GAN model is StyleGAN2-Diff, which is outperformed by 17.6 with the proposed method using the CAR-T data. When using the Kaggle dataset, StyleGAN2-Diff scores 10.36, which is better than the proposed method.

Model	FID[1]	FID[2]	KID[1]	KID[2]
DCGAN	227.06	235.60	.2187	.2042
BigGAN-SD	381.60	363.57	.4765	.3320
BigGAN-Diff	80.83	262.46	.02591	.4875
StyleGAN2-Diff	31.32	**10.36**	.01173	**.0067**
Sem-Aware	**8.37**	23.32	**.00068**	.0212

Table 2: FID and KID scores of single-cell image generation on CAR-T and Kaggle datasets. [1] represents a score using the CAR-T dataset, [2] represents a score using the Kaggle dataset.

Kernel Inception Distance The KID scores show that the proposed method has a higher score than StyleGAN2-Diff by 0.01105 using the CAR-T data. When using the Kaggle dataset, StyleGAN2-Diff scores .0067, which is better than the proposed method.

Due to the simplified nature of single-cells, most methods performed well in producing strong artificial single-cells. The results for the image generation are presented in Figure 3. Figure 3a shows the original single-cell images and their labels. The first two rows are the images and masks of the CAR-T dataset, and the last two rows



(a)

(b)

(c)

Figure 4: Multi-cells image generation on CAR-T and Kaggle datasets; (a) The first two rows and last two rows represent the multi-cells images and their masks from CAR-T dataset and Kaggle dataset, respectively. (b) shows the multi-cells images and their associated masks that were generated by our method. The first two rows are the multi-cells images generated from the CAR-T dataset and their masks. The last two rows are the multi-cells images generated from the Kaggle dataset and their masks. (c) shows the multi-cells images generated by StyleGan2-Diff. In the generation of multi-cells images, StyleGan2-Diff had the best performance among selected GAN models.

are the images and masks of the Kaggle dataset. Figure 3b shows the single-cell images and their associated masks that were generated by our method. Similar to what was mentioned before, the first two rows are the results of the CAR-T dataset, and the last two rows are the results of the Kaggle dataset. Figure 3c shows the output for StyleGAN2-Diff, which had the best FID/KID scores within the GAN comparisons. As mentioned above, these methods do not produce segmentation results. The proposed method gives competitive, if not better results compared to selected GAN methods and outputs images with a great likeness to the original single-cells. This shows that the controlled environment was successful in the generation of single-cells and preserves the original traits of each natural cell.

3.2. Multi-Cells Image Generation

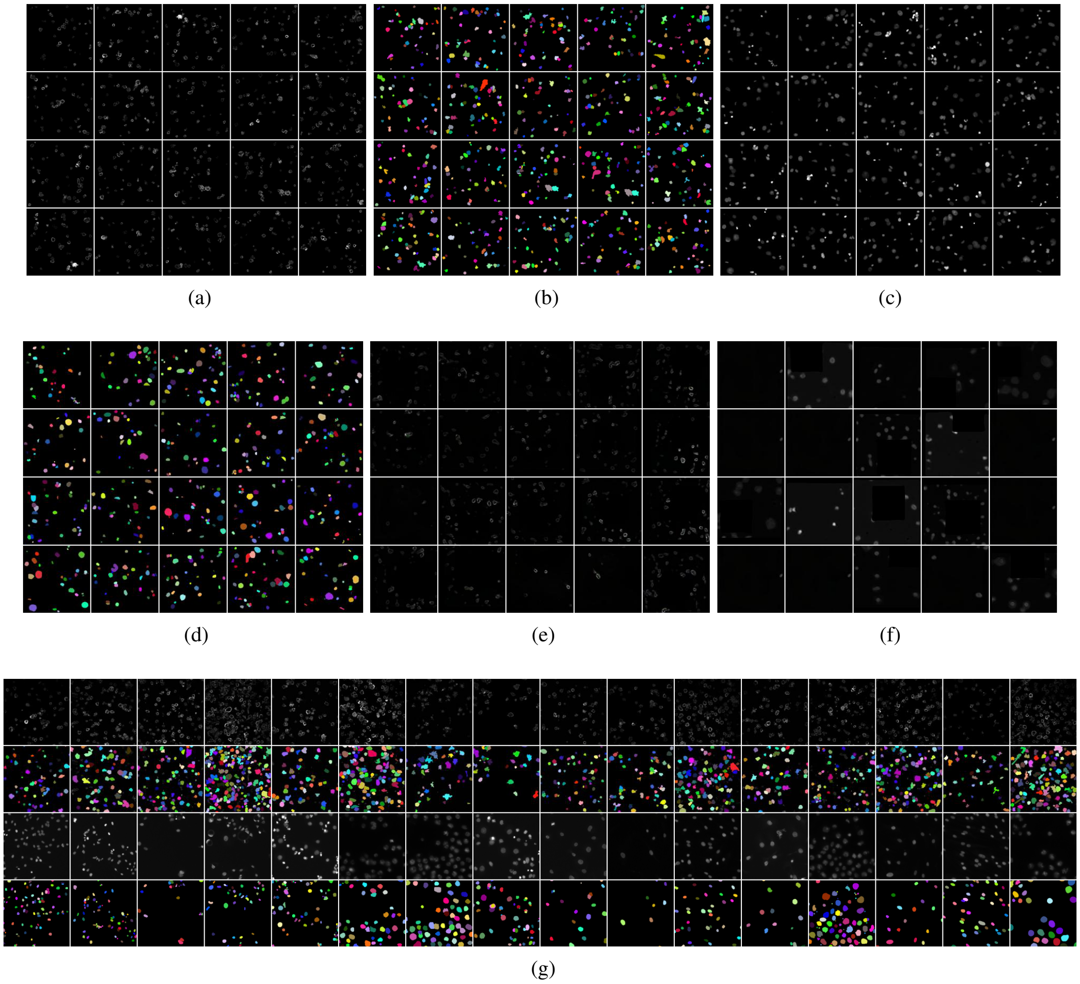
Multi-cells images are another basic element for cell nuclei microscopical images. The artificial generation of multi-cells is more challenging for GAN models because their numbers are generally much lower than single-cells. This should not affect our model, which has minimal dependence on the number of images to produce high-quality cells. Similar to the previous section, we investigate how the proposed method compares to the GAN models used in the creation of realistic images. Respectively, we perform experiments on CAR-T and Kaggle datasets. The scores are reported in Table 3.

Model	FID^1	FID^2	KID^1	KID^2
DCGAN	215.61	404.90	.1911	.2986
BigGAN-SD	369.78	414.81	.3686	.3409
BigGAN-Diff	171.37	126.3	.1300	.1313
StyleGAN2-Diff	86.74	99.04	.0447	.0999
Sem-Aware	**18.33**	**39.93**	**.0011**	**.0732**

Table 3: FID and KID scores of multi-cells image generation on CAR-T and Kaggle datasets. ^{1} represents a score using the CAR-T dataset, ^{2} represents a score using the Kaggle dataset.

Fréchet Inception Distance The results show that our method greatly outperforms the existing models across all parameters. The FID score of the proposed method, surpasses the best performing GAN model StyleGAN2-Diff by 68.41 within the CAR-T dataset and 59.11 within the Kaggle dataset.

Kernel Inception Distance Similar to the FID scores, the data shows the proposed method has a better generation quality for multi-cells. For KID, the best performing GAN model is again StyleGAN2-Diff, which is outperformed



(a) (b) (c)

(d) (e) (f)

(g)

Figure 5: Cell nuclei image generation on CAR-T and Kaggle datasets; (a) and (b) are the CAR-T nuclei images generated by our method and their associated masks. (c) and (d) are the Kaggle nuclei images generated by our method and their associated masks. (e) and (f) represent CAR-T images and Kaggle images generated by StylGAN2-Diff. For reference, the first two rows in (g) show original CAR-T dataset, and the last two rows show original kaggle dataset.

by 0.04365 by our method within the CAR-T dataset and 0.0267 within the Kaggle dataset.

The evaluation scores show the difficulty of generating naturalistic multi-cells due to the many variables that result in cell clustering and adhesion. The results to produce multi-cells using the CAR-T dataset and Kaggle dataset are presented in Figure 4. Similar to the single-cells, samples are provided for observation. Figure 4a shows original multi-cells images and their labels, while Figure 4b shows multi-cells images and their masks generated by our method. Figure 4c shows the output for the method StyleGan2-Diff, which had the best FID/KID scores within the comparisons. The proposed method consistently provides better results compared to GAN for multi-cells. This shows that the controlled environment was again successful in generating new multi-cells while simultaneously preserving the traits found within the original cells.

3.3. Cell Nuclei Image Generation

The proposed method utilizes two sets of images (single-cells and multi-cells), along with recognized noise background, to place them into single frames. The quantity of single-cells and multi-cells are important for creating diverse images. Since we are utilizing small sets of training data, generating artificial data seems more difficult for GANs. The results is shown in Table 4.

Fréchet Inception Distance The proposed method was able to consistently outperform the best performing GAN

Model	FID^1	FID^2	KID^1	KID^2
DCGAN	343.59	435.55	.4563	.3352
BigGAN-SD	460.66	496.38	.5216	.5216
BigGAN-Diff	115.51	297.84	.1313	.3037
StyleGAN2-Diff	109.30	196.92	.0760	.0892
Sem-Aware	**79.57**	**102.99**	**.0657**	**.0716**

Table 4: FID and KID scores of generated cell nuclei images on CAR-T and Kaggle datasets. ^{1} represents a score using the CAR-T dataset, ^{2} represents a score using the Kaggle dataset.

method, StyleGAN2, by 29.73 with the CAR-T dataset and 93.93 with the Kaggle dataset.

Kernel Inception Distance The results remained consistent with the FID values, with the proposed method outperforming StyleGAN2 by 0.97 using the CAR-T dataset and 0.0176 using the Kaggle dataset.

The final results of the creation of complete CAR-T cell images are presented in Figure 5. Figure 5a and Figure 5b show CAR-T images and masks generated by our method. Figure 5c and Figure 5d show Kaggle images and masks generated by our method. Figure 5e and Figure 5f represent CAR-T images and Kaggle images generated by StylGAN2-Diff. Figure 5g shows the real images and masks for reference. The first two rows are the images and masks of the CAR-T dataset, and the last two rows are the images and masks of the Kaggle dataset. The results show the precise segmentation information that our method provides, as well as demonstrating the clear advantage in the creation of high-quality images over alternative GAN solutions.

4. Perfomance on Real Applications

In Table 5 Part 1, we tested our augmentation algorithm with accuracy (AP score) of 11 state-of-the-art detection algorithms. The values on the left side indicate the AP scores from original algorithms, and the values on the right side indicate the AP scores after applying 100 augmented data to the original training sets. The results show that we could consistently improve the results of these algorithms (30 out of 33).

Complex Datasets : While the dark background is needed to create synthetic photo-realistic images, datasets with higher spectral color can still benefit from the method for certain tasks. In Table 5 Part 1 , we report the results for bounding box detection for the Neural cells dataset [40], which has high spectral

Conditional GAN : The GAN methods in their pure form cannot be used to improve the bounding box detection because they do not produce the respective labels which are required for comparisons. In Table 5 Part 2, we use [19] for comparing the method with conditional GAN. By applying 100 extra augmentation data to the original training, we could increase the accuracy by 10.79% and 25.67% for CART and Kaggle. The accuracy here is the mean of the Jaccard scores [16] between pairs of generated and original semantic labels for all test points.

Part 1: The AP scores 11 state-of-the-art bounding box detection algorithms using three datasets.				
Method	Backbone	CART	Kaggle	Neural
multi-stage:				
Faster R-CNN	R-101	49.6/**51.2**	38.6/**44.0**	48.4/**49.7**
Cascade R-CNN	R-101	49.8/**52.5**	**37.5**/37.2	50.7/**50.9**
Grid R-CNN	X-101	26.1/**46.2**	35.0/**40.2**	38.2/**41.0**
Libra R-CNN	X-101	47.6/**48.4**	43.8/**44.5**	50.0/**50.1**
RepPoints	R-101	43.4/**47.5**	38.8/**40.0**	**37.9**/36.7
RepPoints	X-101	45.3/**48.4**	35.8/**38.9**	42.7/**45.9**
one-stage:				
FreeAnchor	R-101	26.6/**47.6**	33.3/**33.4**	48.4/**51.2**
FSAF	X-101	44.1/**50.3**	37.8/**40.6**	48.4/**56.0**
ATSS	R-101	49.9/**52.9**	36.3/**37.2**	**46.9**/46.0
PAA	R-101	47.1/**48.3**	38.5/**46.4**	46.0/**49.4**
GFL	X-101	47.7/**53.1**	37.1/**40.4**	42.1/**45.4**

Part 2: Comparing the results of Multi-Organ Nuclei Segmentation in Histopathology Images (NSeg).			
NSeg-CART	Ours-CART	NSeg-Kaggle	Ours-Kaggle
32.69	**43.48**	52.06	**77.73**

Table 5: Performance results for bounding box detection algorithms and conditional GAN.

Semantic Segmentation : For semantic segmentation, the results of the mixed CAR-T dataset are at least 0.39 % and up to 2.86% higher than results of the real data on AP. It also shows improvement of at least 0.65% and up to 1.18% when compared to the reference CAR-T dataset at IOUs. The details of these experiments are presented in supplementary. The results show that the method could easily improve the accuracy of semantic segmentation of cells.

5. Conclusion

We presented semantically aware data augmentation for the generation of artificial cell nuclei microscopical images along with their correct semantic segmentation labels. An initial set of segmentation objects are used with Greedy AutoAugment to find the best performing policies. The found policies and initial set of segmentation objects are then used to create the final artificial images. The images are compared with state-of-the-art data augmentation methods. The results show that the quality of the proposed method in different stages is on par with the original data and surpasses different GAN models. These observations are confirmed by FID and KID scores. The proposed method is effectively ready to generate artificial cell nuclei microscopical images. In the future, this method can help to better train microscopical images with artificial neural networks.

References

[1] 2018 data science bowl, 2018.

[2] Antreas Antoniou, Amos Storkey, and Harrison Edwards. Data augmentation generative adversarial networks, 2018.

[3] Mikołaj Bińkowski, Danica J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans, 2021.

[4] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis, 2019.

[5] Min Jin Chong and David Forsyth. Effectively unbiased fid and inception score and where to find them, 2020.

[6] Ekin D. Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V. Le. Autoaugment: Learning augmentation policies from data, 2019.

[7] Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. Randaugment: Practical automated data augmentation with a reduced search space, 2019.

[8] Terrance DeVries and Graham Taylor. Dataset augmentation in feature space. 02 2017.

[9] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[10] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans, 2017.

[11] Swaminathan Gurumurthy, Ravi Kiran Sarvadevabhatla, and Venkatesh Babu Radhakrishnan. Deligan : Generative adversarial networks for diverse and limited data, 2017.

[12] Daniel Ho, Eric Liang, Ion Stoica, Pieter Abbeel, and Xi Chen. Population based augmentation: Efficient learning of augmentation policy schedules, 2019.

[13] Hiroshi Inoue. Data augmentation by pairing samples for images classification, 2018.

[14] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan, 2020.

[15] A. Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60:84 – 90, 2012.

[16] Neeraj Kumar, Ruchika Verma, Sanuj Sharma, Surabhi Bhargava, Abhishek Vahadane, and Amit Sethi. A dataset and a technique for generalized nuclear segmentation for computational pathology. *IEEE transactions on medical imaging*, 36(7):1550–1560, 2017.

[17] Scott E. Lillie, David A. Broadway, Nikolai Dontschuk, Sam C. Scholten, Brett C. Johnson, Sebastian Wolf, Stephan Rachel, Lloyd C. L. Hollenberg, and Jean-Philippe Tetienne. Laser modulation of superconductivity in a cryogenic wide-field nitrogen-vacancy microscope. *Nano Letters*, 20(3):1855–1861, Feb 2020.

[18] Erik Linder-Norén. Pytorch-gan. `https://github.com/eriklindernoren/PyTorch-GAN`, 2018.

[19] Faisal Mahmood, Daniel Borders, Richard J Chen, Gregory N McKay, Kevan J Salimian, Alexander Baras, and Nicholas J Durr. Deep adversarial training for multi-organ nuclei segmentation in histopathology images. *IEEE transactions on medical imaging*, 39(11):3257–3267, 2019.

[20] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders, 2016.

[21] B. Mehlig. Artificial neural networks, 2019.

[22] Qinxue Meng, Daniel Catchpoole, David Skillicom, and Paul J. Kennedy. Relational autoencoder for feature extraction. *2017 International Joint Conference on Neural Networks (IJCNN)*, May 2017.

[23] Aratrika Mustafi. Convex smoothed autoencoder-optimal transport model, 2021.

[24] Alireza Naghizadeh, Mohammadsajad Abavisani, and Dimitris N. Metaxas. Greedy autoaugment, 2020.

[25] Alireza Naghizadehn. greedy_auto_augment. `https://github.com/arnaghizadeh/greedy_auto_augment`, 2020.

[26] Atsuhiro Noguchi and Tatsuya Harada. Image generation from small datasets via batch statistics adaptation, 2019.

[27] Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le. Specaugment: A simple data augmentation method for automatic speech recognition. *Interspeech 2019*, Sep 2019.

[28] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[29] Xi Peng, Zhiqiang Tang, Fei Yang, Rogerio Feris, and Dimitris Metaxas. Jointly optimize data augmentation and network training: Adversarial data augmentation in human pose estimation, 2018.

[30] Stanislav Pidhorskyi, Donald Adjeroh, and Gianfranco Doretto. Adversarial latent autoencoders, 2020.

[31] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[32] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks, 2016.

[33] Kim Seonghyeon. stylegan2-pytorch. `https://github.com/rosinality/stylegan2-pytorch`, 2020.

[34] Yinghuan Shi, Tiexin Qin, Yong Liu, Jiwen Lu, Yang Gao, and Dinggang Shen. Automatic data augmentation by learning the deterministic policy, 2019.

[35] Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. Wasserstein auto-encoders, 2019.

[36] Ngoc-Trung Tran, Viet-Hung Tran, Ngoc-Bao Nguyen, Trung-Kien Nguyen, and Ngai-Man Cheung. Towards good practices for data augmentation in gan training. *arXiv preprint arXiv:2006.05338*, 2020.

[37] Yaxing Wang, Abel Gonzalez-Garcia, David Berga, Luis Herranz, Fahad Shahbaz Khan, and Joost van de Weijer. Minegan: effective knowledge transfer from gans to target domains with few images, 2020.

[38] Yaxing Wang, Chenshen Wu, Luis Herranz, Joost van de Weijer, Abel Gonzalez-Garcia, and Bogdan Raducanu. Transferring gans: generating images from limited data, 2018.

[39] Ju Xu, Mengzhang Li, and Zhanxing Zhu. Automatic data augmentation for 3d medical image segmentation, 2020.

[40] Jingru Yi, Pengxiang Wu, Menglin Jiang, Qiaoying Huang, Daniel J Hoeppner, and Dimitris N Metaxas. Attentive neural cell instance segmentation. *Medical image analysis*, 55:228–240, 2019.

[41] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks, 2017.

[42] Shengyu Zhao. data-efficient-gans. `https://github.com/mit-han-lab/data-efficient-gans`, 2020.

[43] Shengyu Zhao, Zhijian Liu, Ji Lin, Jun-Yan Zhu, and Song Han. Differentiable augmentation for data-efficient gan training, 2020.

[44] Shengyu Zhao, Zhijian Liu, Ji Lin, Jun-Yan Zhu, and Song Han. Differentiable augmentation for data-efficient gan training. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

[45] Zhengli Zhao, Zizhao Zhang, Ting Chen, Sameer Singh, and Han Zhang. Image augmentations for gan training, 2020.

[46] X. Zhu, D. Anguelov, and D. Ramanan. Capturing long-tail distributions of object subcategories. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 915–922, 2014.