ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

CVL Computer Vision Lab

Industrial Image Synthesis for Dataset Augmentation

Master's Thesis

Ting-Yu Chen
Department of Computer Science

Advisors: Evangelos Ntavelis
Prof. Dr. Radu Timofte
Supervisor: Prof. Dr. Luc Van Gool

November 15, 2021

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

INDUSTRIAL IMAGE SYNTHESIS FOR DATASET AUGMENTATION

Authored by (in block letters):

For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**
CHEN	TING-YU

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date	**Signature(s)**
Zürich, 15.1.2021	Ting-Yu Chen

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.

Abstract

State-of-the-art generative adversarial networks (GAN) can generate high-quality images with limited training datasets. In this work, we use these GANs to synthesize images for augmenting an industrial dataset and find out whether we can use dataset augmentation to improve the performance of downstream tasks, including classification and segmentation. The industrial dataset has high-resolution images of pump parts with small defects, and we crop the pump images into patches. We propose a defect-pasted technique to create more pump patches with defects for the training of GAN, and our experiments show that it helps GANs generate defects. The classification task is to identify whether a pump patch image has defects, and the segmentation task is to find the exact location of the defects. For classification, we generate pump patches from the StyleGAN2-ADA to over-sample patches with defects. For segmentation, we propose a modification to the SemanticGAN and use our GAN to generate pump patch images/labels for augmentation of the training set. The performance of the classification model is greatly improved after augmenting the dataset with our synthesized pump images. However, our dataset augmentation does not bring a performance boost to the segmentation task. It needs more exploration on how to make GAN generate informative images/labels for the segmentation model.

The other part of our work is minority-class semantic generation because we find that the semantic labels of minority classes are less likely to be generated. We propose a new loss term, distribution loss, to have further control over semantic label generation. This loss term can make our semantic GAN generate more pixels of minority classes, but it causes some artifacts to the synthesized labels in the CelebAMask-HQ dataset.

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my advisors, Evangelos Ntavelis and Dr. Radu Timofte, for their continuous guidance, support, and encouragement during my research. During the research and writing of my master thesis, I have received a lot of assistance and support. I always get great insights and learn a lot from our weekly meetings, which helps me stay on the right track. I especially appreciate Evan's active involvement during the whole time, and I feel blessed to have such an excellent advisor.

I would like to thank Dr. Iason Kastanis and Matthias Höchemer from CSEM for offering the interesting industrial dataset, pumps image dataset, which is proviedd by KNF Flodos. I'm thankful for all the kind colleagues working in the robotics department of CSEM to attend my mid-term presentation and gave me advice from very different perspectives.

I'm very grateful for my parents' financial and emotional support for my master's studies, and it means a lot to me. I want to also thank all my friends and roommates for talking to me and cheering me up. And the cute cats, Siri and Luci, always made me happy in the last two months. Last but not least, I would like to thank my lovely boyfriend, Chun-Ming, for his encouragement and everything he has done for me during some of the stressful weeks. Finally, thanks to God, it's a blessing to have such an outstanding master thesis.

Contents

CONTENTS

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Focus of Work

It has been known that the generative adversarial networks (GAN) need a large amount of data to train to generate good images. Fortunately, some recent studies have enabled GANs to generate images with limited training sets. That makes it easier to achieve dataset augmentation by images generated from the GAN model. Some research has indicated that GAN-based over-sampling in biomedical images can improve classification performance. We want to explore the possibility of GAN-based dataset augmentation in a unique industrial image dataset, the pump part dataset. We first train a state-of-the-art GAN model to generate images of the minority class, pump part with defects. Next, we build a baseline classifier to measure how much performance improvement we can earn from augmenting the dataset by fake images generated from the GAN model.

In addition to generating more images in the minority class for the classification task, our work also includes generating both images and semantic labels to augment the segmentation task's training set. The SemanticGAN[15] can generate images and semantic labels with unlabelled images and a small number of labeled images, and we want to investigate the potential of dataset augmentation for segmentation tasks in the industrial dataset, where the target class, defects, only take up a tiny part of the image. To make the GAN model better at generating the minority classes, we propose some modifications to the SemanticGAN. Next, we train a baseline segmentation model to see if dataset augmentation can boost the segmentation task's performance.

To generalize our research about SemanticGAN, we expand our experiments to the public dataset, CelebAMask-HQ[14], and we find that the SemanticGAN has a hard time generating minority classes. Therefore we attempt to provide more incentives for the generator to generate those minority classes by adding an additional loss term to affect the pixel-wise distribution. We utilize the encoder-generator segmentation model in the SemanticGAN to quantify the effect of our method. Correspondingly, we apply the loss term to our industrial dataset with similar experiment settings and train a segmentation model to evaluate the quality of generated images and labels.

1.2 Thesis Organization

This work consists of two major parts: industrial dataset augmentation and minority semantic class generation, and they are organized independently in two chapters. The chapter on industrial dataset augmentation covers how we improve the quality of synthesized images and labels. It also includes the experiments and

results of the downstream tasks after the dataset augmentation. The other main chapter of minority semantic class generation would mainly focus on the experiments on the CelebAMask-HQ dataset and the details of the loss term we propose, such as how the label distribution changes after adding the loss.

Chapter 2

Related Work

2.1 Generative adversarial networks

Generative adversarial networks[3] (GAN) is designed by Goodfellow et al. in 2014. A GAN involves a generator that generates data points from random noise and a discriminator which evaluates the realness of the synthesized data points. The generator aims to generate data points that look like actual data, and the discriminator tries to distinguish them. By letting the generator and discriminator compete and learn from each other for many iterations, the generator can eventually produce fake data points that are very similar to real ones. This GAN training process can be described by Eq. 1.

$$\min_G \max_D V(G, D) = \mathbb{E}_{\mathbf{x}\sim\mathbf{p}_{\text{data}}}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}[\log(1 - D(G(\mathbf{z})))] \tag{1}$$

Image synthesis can be achieved by using specific architecture for the discriminator and the generator in the GAN framework. Radford et al. proposed deep convolutional generative adversarial networks (DCGAN)[22]. It leverages the power of deep convolutional neural netork[13] and successfully generates good images stably. The architecture of DCGAN uses convolutional and convolutional-transpose layers in the discriminator and the generator, respectively, and applies batch normalization to them. Progressive Growing GAN[9] further improved the training stability by fading in layers with an increasing spatial resolution during training for both discriminator and generator. In addition, they boosted the variation among synthesized images by taking the standard deviation of features on mini-batch and making it an additional feature map for the discriminator. As a result, Progressive Growing GAN has gained great success in generating good quality and high-resolution images. StyleGAN[10] improves the generator of Progressive Growing GAN by applying adaptive instance normalization[7], and StyleGAN2[12] further restructure the progressive-growing architecture to skip generator and a residual discriminator to have even better image quality.

Image synthesis with limited data is what we need for applying GAN models to industrial datasets because data collection and labeling are usually very costly in the industry. The models tend to overfit when the size of the training set is too small. However, performing geometric transforms on training data to prevent overfitting in GAN models may cause leaking, which means that the synthesized images are affected by those transforms. The adaptive discriminator augmentation techniques[11] proposed by Karras et al. solved the leaking problem by applying differentiable augmentation[30] and adaptively adjusting the augment probability based on the level of overfitting. Liu et al. approached the overfitting problem by designing the skip-layer channel-wise excitation module for the generator to have strong gradient flow and restructuring the discriminator to a self-supervised feature encoder trained with extra decoders[17].

Conditional image synthesis allows us to have some control over the synthesized image. Mirza and Osindero introduced conditional GAN[19], which feeds the class label to both discriminator and generator as an additional input layer. With `pix2pix`[8], we could even manipulate the synthesized images on pixel level through conditioning upon pixel-wise label. The multi-scale generator and discriminator architecture proposed by Wang et al. [27] further enabled high-resolution image synthesis conditions on semantic labels, and this work provided a great foundation for many following studies of improving the quality of the synthesized images. Park et al. proposed a new normalization method, spatially-adaptive normalization(SPADE)[21], for the generator to better preserve the semantic information during training. Sushko et al. [26] designed a segmentation-based discriminator to better match the images to the input semantic labels and added 3D noise to increase the variation. Ntavelis et al.[20] suggested to handle the image and semantic labels input by two independent streams in the PatchGAN discriminator to extract more information from the semantic labels.

Semantic image synthesis is a relatively new topic in the field of image generation. Most GAN models focus on generating images, and semantic labels are usually used as input conditions instead of being generated as output results. This state-of-the-art research about semantic image synthesis breaks new ground in this field. Li et al. proposed SemanticGAN[15], which is built on top of the StyleGAN2[12]. They added a semantic branch to the generator of StyleGAN2 and a semantic discriminator, a multi-scale patch discriminator[27], to discriminate the concatenation of images and semantic labels. The synthesized images and semantic labels are generated together from different branches of the same generator model. This SemanticGAN can also be seen as a segmentation model along with an encoder. After the generator is well-trained, they trained an encoder with the labeled data by minimizing the reconstruction loss and dice loss while the generator is frozen. As a result, the SemanticGAN can take any out-of-domain images and produce corresponding semantic labels. Another parallel research, DatasetGAN[29], also suggested an efficient method to annotate images by taking advantage of a strong pre-trained StyleGAN[10] network. With a minimum number of manually labeled images, they trained a style-interpreter to produce corresponding semantic labels of generated images.

2.2 Data augmentation

Data augmentation could significantly improve the network performance by mitigating overfitting, especially when the size of the training set is small[25]. Manipulating the existing dataset by color or geometric transformation, such as flip, rotation, and translation is the most common image processing augmentation. With the development of GAN models, generating new images to augment the dataset is also feasible.

Inner-class imbalance[24] refers that some minority classes have much fewer samples than other majority classes, and that is when GAN models might come in handy. It has been shown that taking generated images from GAN models as dataset augmentation in biomedical image analysis can improve diagnostic sensitivity. Frid-Adar et al.[2] improved the sensitivity and specificity by using DCGAN to generate CT scans of liver lesions. Han et al. proposed condition progressive growing GAN[4] with highly rough bounding box condition to generate brain MR images with tumors, and it increased 10% sensitivity in diagnosis. GAN-based up-sampling can also be useful in other public datasets. For instance, Zhu et al.[32] applied CycleGAN[31] to up-sample minority emotion class and increased the classification accuracy.

Chapter 3

Industrial Dataset Augmentation

The research objective of this part of our work is to explore the possibility of industrial dataset augmentation by GAN models and to experiment whether the dataset augmentation could improve the performance of downstream tasks, including classification and segmentation.

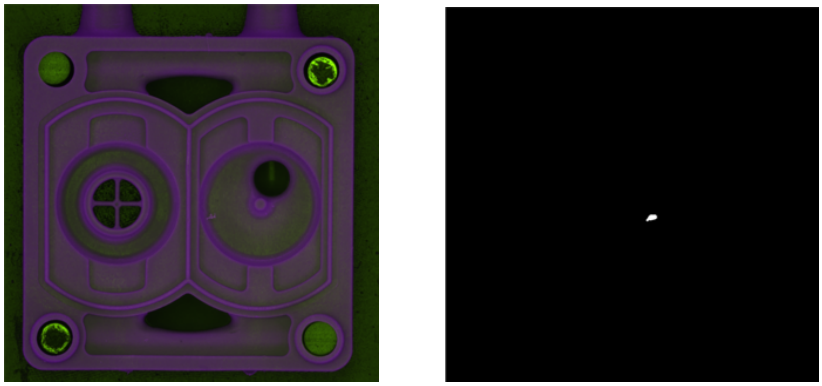
3.1 Dataset

Our industrial dataset, is an image collection of a particular type of part in pumps. We would like to thank KNF Flodos for kindly providing the pumps image dataset. These pump part images have high resolution, which is 3200 by 2224. There are 498 images in the training set and 161 images in the testing set. Each image has its corresponding pixel-wise label, which indicates the location of the defects of that pump part. We only work on one specific type of pump part, and these pictures were taken in a standard process. Therefore, the image structure and content are highly similar in the dataset, except that there is some brightness difference, as shown in Figure 3.1. The primary purpose of the pump dataset is to identify whether there are defects in the pump part.

Figure 3.1: Raw images of the pump part dataset

3.1.1 Pre-processing

It is crucial to retain the original resolution of the pump images as much as possible because the defects are usually tiny relative to the full image. However, it would be challenging to work on images with such resolution. Thus, we crop the raw images into smaller patches by the following pre-processing steps: At first, we center crop the pump part image to the size of 1920 by 1920, which cut off the background and some parts irrelevant to defects. Figure 3.2a exhibits a pump part example after center cropping, and we can get the general idea about the size of defects from this figure. Next, we randomly crop patches of size 256 by 256 from the center cropped pump part images, and the defects would become much more perceivable in the patch images. The patches are resized to 128 by 128 in the following work to accelerate the training process while preserving most of the information. All of our research is conducted on these resized pump patch images instead of full pump images.



(a) Pump Image (b) Semantic label

Figure 3.2: A pump image and its label after center crop

3.2 Methods

We generate pump patch images and image/label pairs with defects by the StyleGAN2-ADA[11] and our modified version of SemanticGAN[15], respectively. It is pretty challenging for the GAN model to correctly generate the defects because they are not very noticeable and only take up a small part of the patch images. We propose defect pasted augmentation to create more pump patch images with defects for better training of the GAN models. In addition, we modify the SemanticGAN structure to make it generate more visible defects in both images and labels. Besides, we present a method to quantify the proximity of distribution of synthesized and actual semantic labels.

3.2.1 Defect Pasted Augmentation

To increase the number of patch images that have defects, we take advantage of the unique property of this industrial dataset and come up with an augmentation method, defect pasted augmentation, which is specifically for datasets with fixed image structure. The idea of our augmentation is to copy and paste, copying the area of defects and pasting it to another good patch image. The destination patch images must cover the same components of the pump part as the source patch images, making the pasted patch images

more realistic. The brightness of destination patch images must also be reasonably close to the source patch images. From Figures 3.3, we could see that the defects are pasted to the correct location with only a few pixels of deviation. With these defect-pasted patch images, we can have more patch images with defects in the training set for the GAN models to improve the visibility and quality of defects in the generated patch images.

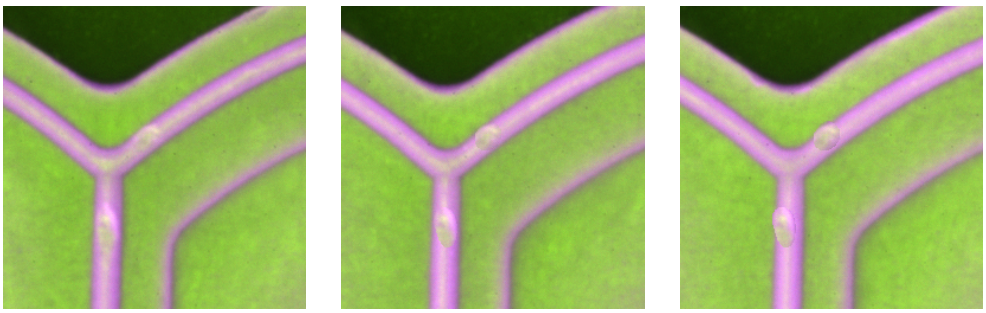


Figure 3.3: Real and defect-pasted pump patch images. The left image is the real patch image with defects, and the middle and right images are other good images after pasting defects from the real one.

3.2.2 Modified Semantic GAN

The authors of SemanticGAN[15] specifically pointed out that they stopped the gradients produced by semantic label discriminator into the generator through the image synthesis branch because they wanted the generated labels to match the images instead of the other way around. In contrast, adjusting the images to match the labels is what we desire in our industrial image/label generation. Hence, we suggest letting the gradient of semantic label discriminator backpropagate to the generator via the image branch in the SemanticGAN. The generator can better synthesize the defects in the patch images to match the generated labels. Our modified semantic GAN is visualized in Figure 3.4. The loss functions of generator, discriminator of image and discriminator of semantics are Eq. 2, Eq. 3, and Eq. 4, respectively. The x and y denote images and labels, their subscript r and f means real and fake, and the D_u and D_l represents unlabelled and labeled data. Although the loss functions remain exactly same as the SemanticGAN, the image generation branch is trained with gradients from both D_{img} and D_{sem} instead of only gradients from D_{img}. Correspondingly, those geometric transformations in the labeled data would leak to the synthesized images. Hence, only horizontal and vertical flips are allowed for the labeled training set after the gradient modification.

$$\mathcal{L}_G = \mathop{\mathbb{E}}_{(x_f,\cdot)=G(z),z\sim p(z)}[\log(1 - D_{img}(x_f))] + \mathop{\mathbb{E}}_{(x_f,y_f)=G(z),z\sim p(z)}[\log(1 - D_{sem}(x_f, y_f))] \tag{2}$$

$$\mathcal{L}_{D_{img}} = \mathop{\mathbb{E}}_{x_r\sim D_u}[\log(D_{img}(x_r))] + \mathop{\mathbb{E}}_{(x_f,\cdot)=G(z),z\ p(z)}[\log(1 - D_{img}(x_f))] \tag{3}$$

$$\mathcal{L}_{D_{sem}} = \mathop{\mathbb{E}}_{(x_r,y_r)\sim D_l}[\log(D_{sem}(x_r, y_r))] + \mathop{\mathbb{E}}_{(x_f,y_f)=G(z),z\sim p(z)}[\log(1 - D_{sem}(x_f, y_f))] \tag{4}$$

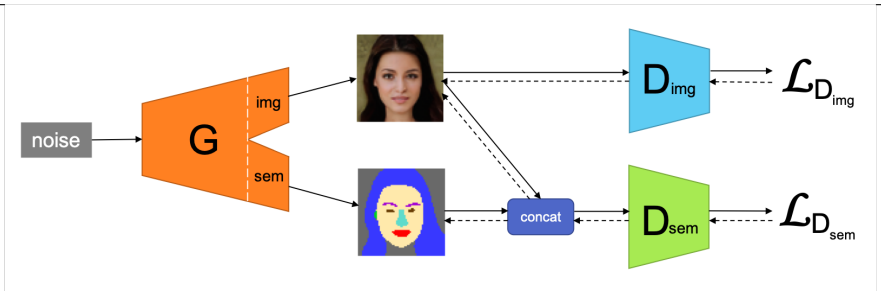


Figure 3.4: Model structure of our modified version of SemanticGAN

3.2.3 Dataset Augmentation

We augment the dataset for two kinds of downstream tasks, classification and segmentation. For the classification task, we want to over-sample the patch images with defects by adding fake images generated from StyleGAN2-ADA[11] to the training set of the classification model. The GAN model would focus on unconditionally generating patch images with defects because there are plenty of healthy patch images. The training set for the StyleGAN2-ADA is real patch images with defects and our defect-pasted patch images. The synthesized patch images can be labeled as having defects for the classification. As for the segmentation task, we generate patch images with defects and their semantic labels via our modified version of SemanticGAN[15]. Likewise, only images and image/label pairs with defects are included in the training set for our GAN model.

To evaluate the effectiveness of dataset augmentation, we train baseline models for both classification and segmentation tasks and find out whether there is a performance improvement after adding our generated patch images to the training set of the baseline models.

3.2.4 Metrics for Synthesized Semantic Labels

Because most semantic labels are annotated by humans manually, and the synthesis of semantic labels is relatively novel, there are no standard metrics to evaluate the quality of semantic labels generated from noise. We propose using a histogram to represent the number of pixels among synthesized semantic labels for each class. We can compute the Wasserstein distance[23] (Earth-Moving distance) between the histogram of actual semantic labels and synthesized ones to evaluate whether specific classes are underrepresented. We use the distance to evaluate how close the distributions of synthesized semantic labels are to the real ones.

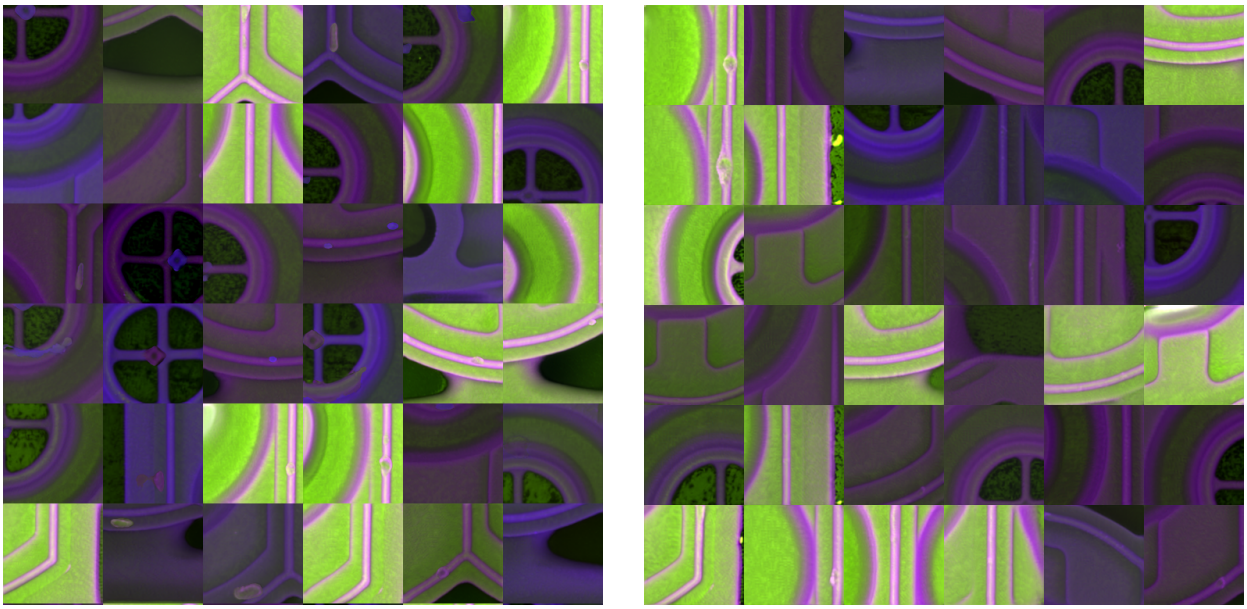
3.3 Experiments and Results

We generate pump patch images with defects by StyleGAN2-ADA and train a classifier to see if appending the synthesized images to the training set can improve the classification performance. Similarly, we generate image and semantic label pairs by our modified semantic GAN and train a segmentation model to investigate if the synthesized image-label pairs can be used as dataset augmentation. The details of model training and results are illustrated in the following sections.

3.3.1 Industrial Image Generation

To train a GAN model for generating pump patch images with evident defects, the training data for the GAN model includes not only pump patch images with defects but also our defect-pasted pump patches. The GAN model would ignore those defects and generate many good pump patches if we train the model with random patch images, with around five percent of images having defects.

The implementation we use for StyleGAN2-ADA is the officially released codes: `https://github.com/NVlabs/stylegan2-ada-pytorch`. There are 6662 patch images with defects and 26648 defect-pasted patch images in the training set, and they are resized from 256 by 256 to 128 by 128. The training configuration is auto, and the model is trained until the Fréchet inception distance (FID)[1] converges to 16.05, after around 18900K images are shown to the models. The synthesized results are shown in Figure 3.5a, and we can see that there are apparent defects in almost every synthesized patch image. Although some of them are slightly dislocated due to the artifacts of defect-pasted pump patches, the quality is generally pretty good. To demonstrate the effectiveness of our defect-pasted augmentation, we also train a model without it to approximately the same number of iterations, and the FID is 19.66. The defects in synthesized patch images are less visible than the ones with defect-pasted augmentation, as shown in Figure 3.5b.



(a) Trained **with** defect-pasted augmentation

(b) Trained **without** defect-pasted augmentation

Figure 3.5: Pump patch images with defects generated from StyleGAN2-ADA

3.3.2 Classification

We train a binary classification model to evaluate if augmenting the dataset by the synthesized patch images can improve the classification performance. The details about the dataset, models, and results are explained below.

Dataset

The training and testing dataset of this task are pump patch images of 128 by 128, as stated in the pre-processing section. We randomly crop 64 patches from each pump image, leading to a highly imbalanced dataset; approximately 5% of the patch images have defects. More specifically, there are 31872 patch images in the training set (1607 of them have defects) and 10304 patch images in the testing set (519 of them have defects).

Classification Model

The architecture of our classification model is residual net[5], and we use the `resnet18` model defined in `torchvison.models`. The model is trained for 80K iterations, and a batch of images is randomly sampled and fed to the model in each iteration. This way, we can ensure that the same number of images are shown to the model after the same number of iterations. In addition, validation-based early stopping is applied during the training. We use 20% of the training set as the validation set and keep track of the model with the best F1-score in the validation set as the final model for testing.

Results

In an experiment, we sample 2000 images from the image pool generated from GAN models, add them to the training set of the classification model, and train the model as described in the previous section. Next, we test the model with the testing set and report the performance. All the experiments are repeated three times, and we re-sample the validation set and the generated images in each repeated experiment.

From Table 3.1, we can observe that augmenting the dataset by synthesized patch images from our method, StyleGAN2-ADA trained with additional defect-pasted patches, can significantly improve the F1-score of classification model. We also experiment on weighted-sampling the patch images with defects during training; that is, the probability of sampling a pump patch with defects is increased and reaches the same probability after adding generated patches to the training set. Our augmentation method outperforms the classifier with weighted sampling. It indicates that the synthesized pump patch images provide some new information for the classifier and help it decide whether a pump patch has defects. Furthermore, the comparison between StyleGAN2-ADA and ours reveals the effectiveness of defect-pasted augmentation for GAN training. More results with different size of augmentation are listed in Appendix A.

Augmentation Method	Accuracy	Precision	Recall	F1-Score
Baseline (None)	**0.9401**	0.2837	0.1194	0.1668
Weighted-Sampling (None)	0.9365	0.2854	0.1730	0.2152
Defect-Pasted	0.9154	0.2217	0.2588	0.2317
StyleGAN2-ada	0.9313	0.3105	0.2960	0.3030
Ours	0.9279	**0.3201**	**0.3699**	**0.3402**

Table 3.1: Mean testing results for pump patch classification with different dataset augmentation methods.

3.3.3 Industrial Image and Label Generation

The images and labels of pump patch with defects are generated from our modified semantic GAN, as mentioned in section 3.2.2. Our implementation is based on the official code: `https://github.com`

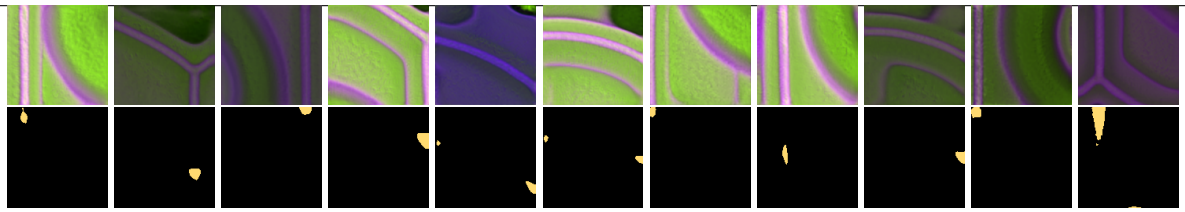


Figure 3.6: Synthesized pump patch images and labels generated from SemanticGAN

`/nv-tlabs/semanticGAN_code`. It is worth noting that we find an engineering bug in the official implementation: the regularization for image discriminator is not properly back propagated, and that makes the SemanticGAN unable to generate reasonable pump patch image. The original SemanticGAN in the following paragraphs refers to the one after fixing this issue. To make the defects more visible, the labeled training set includes only patch images with defects and we add extra defect-pasted pump patch images to the unlabelled training set of our modified semantic GAN.

The FID is converged after training for 140K iterations with batch size of 8. From the image and label pairs shown in Figure 3.7, we could notice that our model can generate more clear defects in images at corresponding pixels labelled as defects, in comparison with the results of SemanticGAN in Figure 3.6. Furthermore, we measure the pixel distribution in quantitative metrics, including the number of synthesized labels having defects, the mean of pixels of defects and the Earth-Moving distance of defects distribution between real and generated samples in Table 3.2. The results show that our method indeed generates more defects than SemanticGAN. It's even more clear if we look at the histogram in Figure 3.8: our method is better at generating large pieces of defects. However, the real distribution of defects is extremely skewed and has a long tail, and our method cannot model that part either.

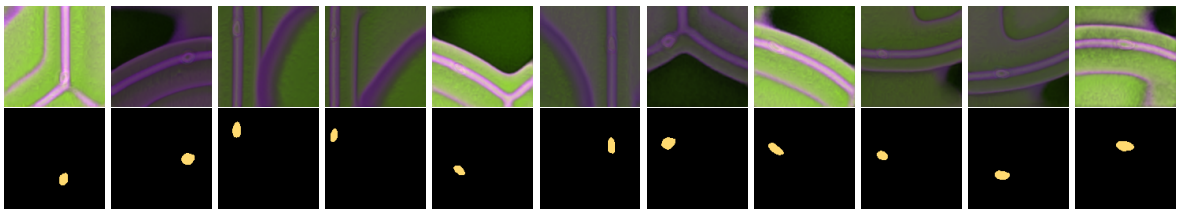
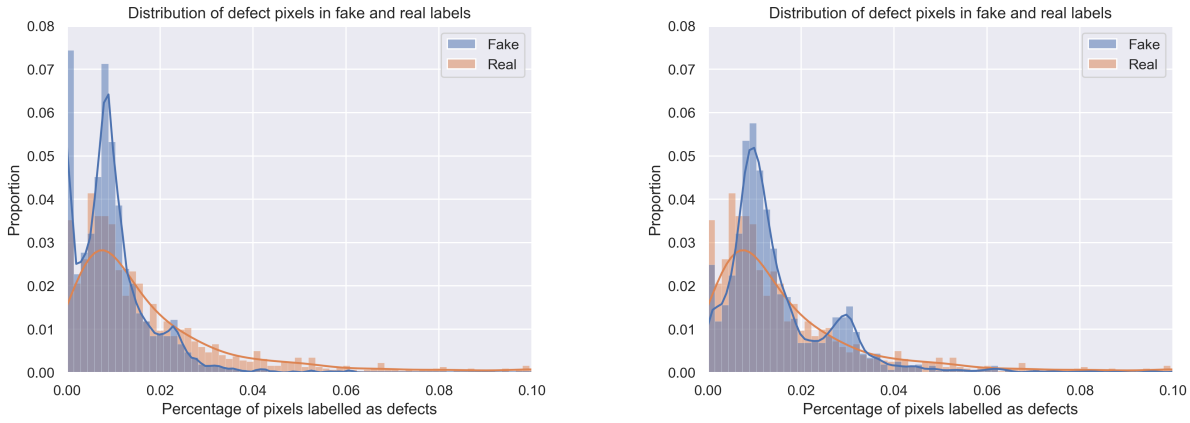


Figure 3.7: Synthesized pump patch images and labels generated from our method

Method	Images with defects(%)	Pixels of defects(%)	Earth-Moving Distance
Real	100	2.52	N.A.
Semantic GAN	90.98	0.96	0.016
Ours	99.38	1.56	0.011

Table 3.2: Quantitative metrics of defects in synthesized labels. The smaller Earth-Moving Distance indicates the distribution is closer to the real one.



(a) SemanticGAN

(b) Ours

Figure 3.8: Comparison of defect pixels distribution of synthesized pump patches between Semantic GAN and ours. The Earth-moving distance between real and fake is shown in Table 3.2. Note that the axis is truncated at 0.1 to make the difference clear.

3.3.4 Segmentation

In the same way as the classification task, we train a segmentation model to check whether augmenting the training set by image/label pairs generated from our semantic GAN can improve the segmentation performance. More details about the experiments are illustrated below.

Dataset

The primary goal of this segmentation task is to find the exact location of defects in a patch image of the pump. The training and testing data only include patch images with defects so that the healthy patches would not confuse the segmentation model. We assume that given some pump patches with defects, and we would like to find out the exact location of the defects in these patches. Otherwise, the segmentation model would not segment any defects if 95 % of the training images have no defects. Even though we consider patch images with defects, on average only 2.5% of pixels are labeled as defects. There are 1607 patch image and label pairs in the training set and 519 pairs in the testing set.

Segmentation Model

The segmentation model is built from the Segmentation Models Pytorch (SMP) library[28]. It has an encoder, a squeeze-and-excitation network[6], to extract features from the input image, a decoder, which has a feature pyramid network(FPN)[16] architecture, and a final segmentation head to make the output semantic mask have the dimensions and resolutions we expect. To make the model focus on segmenting pixels of defects, we model it as a single class segmentation model and set 0.9 as the threshold for the output probability. Similar to the classification task, validation-based early stopping is also applied. The model with the best IoU score of defects in the validation set, which is 20% of the training set, is the final model for testing. We train the model for 20K iterations and validate the model for every 400 iterations.

Results

We sample 500 image/label pairs from the collections of images and labels generated from our method and append these synthesized images and labels to the training set of the segmentation model. Then we train and test the model as illustrated in previous paragraphs. Like the classification section, all the experiments are repeated three times, where the train-validation split and generated data points are re-sampled for each experiment. Although the dataset augmentation does not give the segmentation model performance improvement, the mean IoU score after augmenting the data by our method is at least as good as the baseline model, as indicated in Table 3.3. In contrast, the synthesized image/label pairs of original SemanticGAN even introduce some noise to the segmentation model and slightly worsen the performance. Some samples of segmentation results are presented in Figure 3.9. More results with different size of augmentation are included in Appendix A.

Method	IoU of defects	IoU of healthy part	mean IoU
Baseline	0.5246	0.9869	0.7558
SemanticGAN	0.5057	0.9854	0.7456
Ours	0.5234	0.9866	0.7550

Table 3.3: Mean testing results for pump patch segmentation with different dataset augmentation.

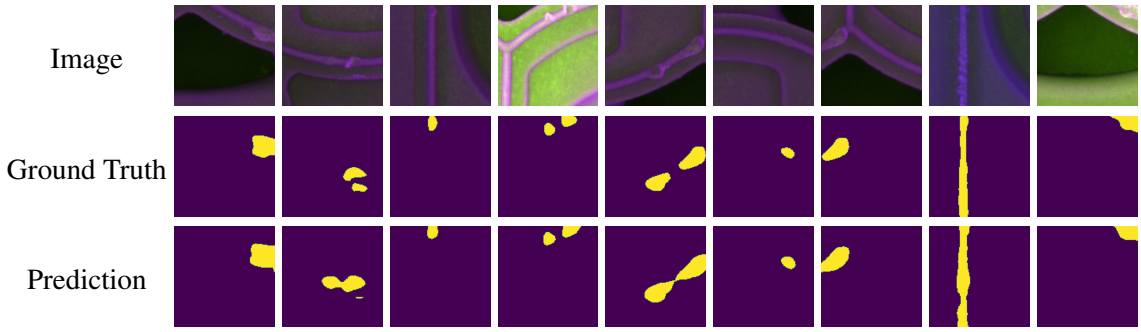


Figure 3.9: Qualitative examples of segmentation prediction of pump patch images with defects. The segmentation model is trained with dataset augmented by our method.

3.4 Discussion

We show that using the GAN model to augment the dataset and over-sample minority class for classification has great potential thanks to this research on GAN models with limited data. The generated images not only over-sample the minority classes but also give some new information to the classifier. However, it still needs more exploration when it comes to generating both images and semantic labels for segmentation tasks.

Our experiments evidence that the StyleGAN2-ADA can generate defects without additional augmented images, and GAN models need to generate images in great detail for industrial application. Although we significantly improved the pump patch classification task, it is not the best from the whole perspective. The baseline classification model is not the best one, and we may get some further improvement if training on more advanced models. However, it is still far from the standard of real-world industrial application.

Ideally, we should generate the full pump image without missing the details, such as defects, but it is nearly infeasible under the limited GPU memory. Besides, generating defects in this scale will become even more complicated, where the defects account for less than 0.05% of pixels. It's still very challenging for the GAN to pay attention to such details during generation. Nevertheless, our work provides a solid basis for more research about generating specific details from GAN models.

Most of the previous dataset augmentation studies generate images for the classification task, and SemanticGAN makes it possible to generate image and semantic label pairs for the segmentation task. Nevertheless, it is likely for the synthesized data points to confuse the segmentation model and deteriorate the performance when the synthesized images and labels are not matched. Even if the augmented dataset is perfectly aligned, it cannot improve the performance when the extra data points cannot provide new information for the segmentation model. There is still more work required to generate relevant and meaningful image and semantic labels for the segmentation model.

Chapter 4

Minority Semantic Class Generation

In our experiments in generating patch images of the pump part, the GAN model tends to ignore the defects if the patch images with defects only account for a small part of the dataset. For instance, it is tough for the SemanticGAN[15] to generate evident defects when only 5% of training patch images have defects. To generalize our research, we investigate this problem on a public dataset and propose a possible method.

4.1 Dataset

CelebAMask-HQ ataset[14] has 30K face images selected from the CelebA Dataset[18] and their corresponding pixel-wise semantic labels for 19 different classes, including facial components and accessories. We merge these 19 classes into ten classes: skin, nose, eyes, eyebrows, ears, mouth, hair, hat, eyeglass, and earrings. The SemanticGAN research mainly focused on the facial components and did not consider those accessories classes. It is fascinating to see how the SemanticGAN deals with these accessories classes, which are relatively rare compared to facial components. To be more specific, only less than 30% of face images have earrings, and around 5% have glasses or hats. Without loss of generality, we use the resolution of 64 by 64 for experiments to explore as much as possible under limited time and computing resources.

4.2 Methods

In the SemanticGAN, the semantic labels and images are generated together, and therefore we can have more information about the synthesized images from their corresponding semantic labels. We wonder if we can use this new information to manipulate the generation of semantic labels and images.

4.2.1 Semantic Gradients for Image generation

The earrings on faces and the defects on pumps have very similar characteristics. They are small relative to the whole image and have different shapes. Based on the results of the industrial image/label generation, we use the same modified semantic GAN structure as explained in section 3.2.2 to make the model generate earrings and other accessories in semantic labels and their corresponding pixels in images. That is, the gradient of the semantic label discriminator would backpropagate to the image branch of generator, as illustrated in Figure 3.4

4.2.2 Distribution Loss

Those minority classes seem to be more likely to be ignored by the SemanticGAN, and thus we would like to add a loss term to make the GAN more aware of the minority classes. We assume that a Gaussian distribution can model the number of pixels for each class among the dataset. The idea is to minimize the distance between the Gaussian distribution of generated semantic pixels and real ones. We compute the mean and standard deviation of pixel percentages for each class in the batch data during training to get the distribution of generated samples. The statistics for actual labeled data are pre-computed. We can compute the KL divergence between the Gaussian distributions of actual semantic labels and generated ones, and that is the distribution loss, which is minimized by the generator.

It is imperative to have an appropriate lambda for this distribution loss. If the lambda is too large, the distribution loss will dominate the generator's loss term, and the synthesized labels will have many artifacts. Another possible solution to the artifacts is decaying the lambda during training so that the artifacts can fade away. The distribution loss term for a single class is summarized in Eq. 5. Since this loss term is class independent, it is possible to apply to any subset of the segmentation classes, and the loss would become the mean of the distribution loss of all selected classes.

$$\mathcal{L}_d = \lambda_d D_{KL}(\mathcal{N}(\mu_r, \sigma_r^2)||\mathcal{N}(\mu_f, \sigma_f^2)) \tag{5}$$

4.3 Experiments and Results

To evaluate the impact of our distribution loss, we consider both quantitative and qualitative results of the synthesized images and labels. We further incorporate the encoder in the SemanticGAN to form a segmentation model to see how the distribution loss affects the segmentation performance.

In addition, we wonder how the distribution loss term would influence the image generation for the industrial dataset, pump parts. Therefore, we design an experiment setting similar to the earring in the CelebAMask-HQ[14] dataset to find out how the distribution loss would guide the generation of pump patches. More details and results are written in the following section.

4.3.1 Image Generation

The images and labels are generated from our modified semantic GAN, and we conduct experiments on the generation with and without distribution loss. The training parameters are the same as the original SemanticGAN, except we use 64 by 64 image size to speed up the training. We also add three extra accessories classes, earrings, glasses, and hats. There are 28000 unlabelled and 1500 labeled images in the training set and 500 labeled images in the testing set. The FID would converge after 110K iterations.

We experiment with adding the distribution loss on a single class and all of the classes. For the single class distribution loss, we notice that it is not effective when the presence of the class is too infrequent, such as the class of glasses and hat in the dataset. The synthesized semantic labels would have more noise and artifacts when imposing the loss on multiple classes. It needs a decay weighting schema to fix this issue. In the following paragraphs, we will focus on presenting the results of adding distribution loss to the class of earrings and all classes with a decaying weight.

When the FID converges, but the percentage of pixels labeled as earrings in synthesized semantic labels does not converge. However, in most checkpoints, our method has advantages over the original semantic GAN in terms of earring pixels generation. If we compare the models of the best FID, as shown in Table 4.1, we can observe that the distribution loss is effective in increasing the percentage of earring pixels in the

synthesized results. If we compare the statistics with the real labeled data, the percentage of images with earrings of our method is higher than the real data. It indicates that our model generates small pieces of earrings in many images instead of large pieces in a few images to minimize the distribution loss.

Although the distribution loss term of the earring class makes the model generate more pixels of earrings without compromising the FID, it also disturbs the generation of other classes and causes some noise in the synthesized labels. Interestingly, the artifacts in the labels do not appear in the corresponding images, as shown in Figure 4.3. As for the experiment on distribution loss for all classes, the decaying lambda also seems to wash away the effect of distribution loss (Figure 4.4). If considering both minority class distribution and quality of semantic labels in generated data, our method without distribution loss have the best overall results. It can better generate earring labels without impairing synthesis of other classes, as demonstrated in Figure 4.2. More details about the semantic statistics of other classes are included in the Appendix B.

Method	Image with earrings(%)	Pixels of earrings(%)	FID
Real	34.45	0.31	N.A
SemanticGAN	15.15	0.04	8.12
Ours w/o $\mathcal{L}_d$	42.66	0.13	8.79
Ours w/ $\mathcal{L}_{d-all}$	35.67	0.03	8.31
Ours w/ $\mathcal{L}_{d-earring}$	47.34	0.19	8.04

Table 4.1: Distribution of earring pixels in synthesized labels of different GAN models.

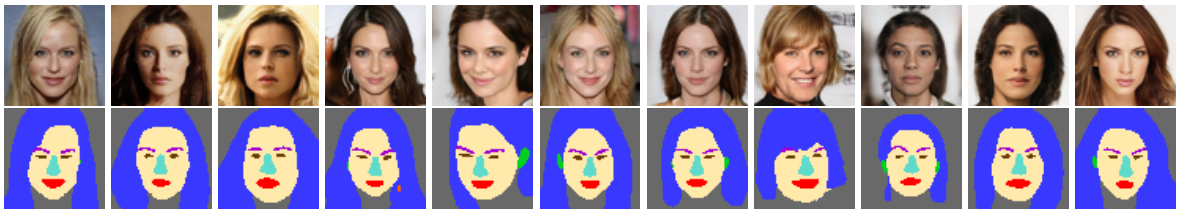


Figure 4.1: Synthesized face images and semantic labels generated from original SemanticGAN

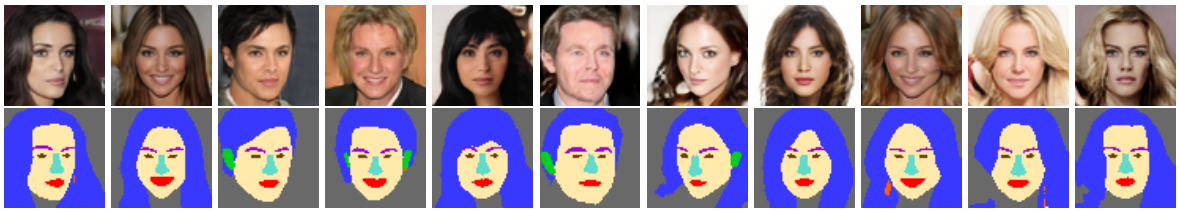


Figure 4.2: Synthesized face images and semantic labels generated from our method **without** distribution loss

4.3.2 Segmentation by Encoder

The second part of the SemanticGAN is to train an encoder to encode any images to input vectors of the pre-trained generator, and use the encoder-generator structure as a segmentation model. If our generator can generate labels of the minority classes, the performance of this encoder-generator segmentation model

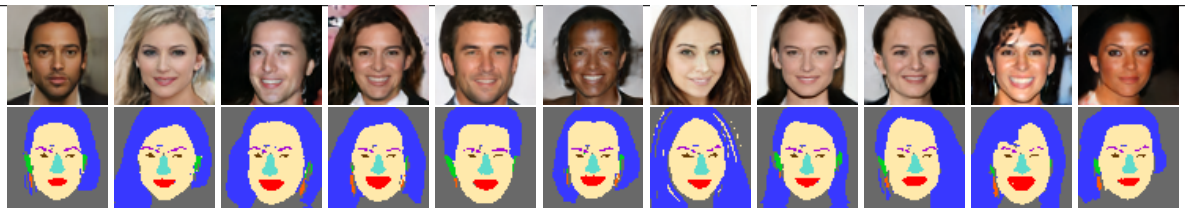


Figure 4.3: Synthesized face images and semantic labels generated from our method with distribution loss on class earring

Figure 4.4: Synthesized face images and semantic labels generated from our method with distribution loss on all classes

should be able to handle minority class segmentation better. However, even if the generator can generate more pixels of earrings from randomly sampled noise, it is not necessarily reflected in the testing mean IoU score of the earring of the encoder-generator segmentation model. Although the distribution loss could not contribute directly to the segmentation performance, backpropagation of semantic gradients to the image branch of generator can improve the mean IoU score for almost every class, as demonstrated in Table 4.2.

Method	Skin	Hair	Eyes	Ears	Brows	Nose	Mouth	Earrings	Glasses	Hat
SemanticGAN	0.80	0.72	0.44	0.26	0.45	0.72	0.65	0.0119	0.0007	0.0007
Ours w/o $\mathcal{L}_d$	0.83	0.76	**0.59**	**0.29**	**0.48**	0.78	0.76	**0.0198**	**0.0028**	0.0003
Ours w/ $\mathcal{L}_{d-all}$	0.83	0.76	0.49	0.25	0.29	0.81	**0.77**	0.0066	0.0002	0.0001
Ours w/ $\mathcal{L}_{d-earring}$	**0.84**	0.76	0.56	0.25	0.39	0.81	0.76	0.0055	0.0001	**0.0016**

Table 4.2: Testing mean IoU for each classes of segmentation by encoder-generator segmentation model

4.3.3 Industrial Image Generation

The previous experiments show that the distribution loss may cause some artifacts and noise in the synthesized semantic labels. There are only two classes in our industrial image dataset, and the generator can concentrate on generating defects without worrying about other labels. In the last chapter, we train our GAN with only pump patch images having defects, but we would like to know if the distribution loss can help when only part of the patches have defects. Thus, we design an experiment setting where half of the pump patches in the dataset have defects, and the other half are healthy patches to assess the effectiveness of distribution loss in this hypothetical scenario. There are 13324 unlabelled images and 3228 labeled images in the training set for our GAN, and 1021 patch images are reserved for the testing set. Instead of making the distribution of defect pixels close to the actual distribution, we want to generate as many defects as possible

for our pump dataset. We double the mean and variance for our target Gaussian distribution to have more defects in the synthesized images and labels.

Our method can greatly change the distribution of defects without causing artifacts in synthesized labels and without compromising the FID, and it outperforms SemanticGAN in terms of defects generation, as demonstrated in Table 4.3. Figure 4.5 and 4.6 clearly show that our method can generate defects with better quality than SemanticGAN. The patterns of defects are very similar to the patterns of earrings in terms of distribution statistics. We haven't reached the same percentages of defect pixels, but we have a lot more images with defects than the real data. Because large defects are less common in the dataset, it's harder for the model to learn to generate large defects. It turns to generate small defects in more pump patches to minimize the loss. More statistics for different checkpoints and the histogram of pixel distribution can be found in Appendix C.

Method	Image with defects(%)	Pixels of defects(%)	FID
Real	48.70	1.26	N.A.
SemanticGAN	46.49	0.31	49.41
Ours w/o $\mathcal{L}_d$	56.48	0.52	48.35
Ours w/ $\mathcal{L}_d$	73.19	0.73	45.68

Table 4.3: Distribution of defects in semantic labels generated from different GAN models.

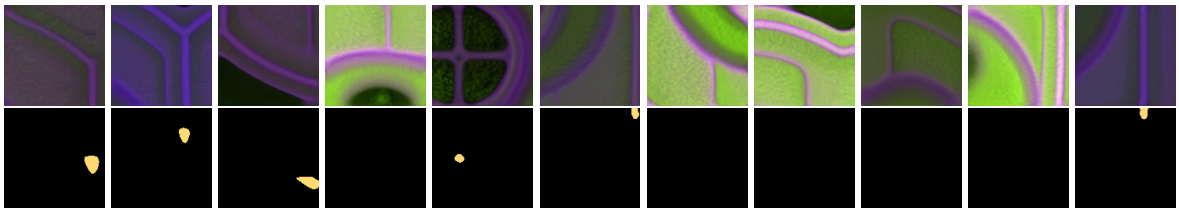


Figure 4.5: Synthesized pump patch image and labels generated from SemanticGAN

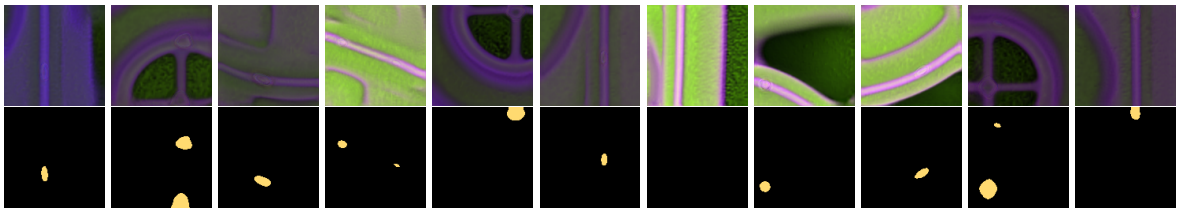


Figure 4.6: Synthesized pump patch image and labels generated from our method **with** distribution loss

4.3.4 Industrial Image Segmentation

The encoder-generator segmentation model from SemanticGAN does not give good mIoU performance in defects segmentation for pump part patch images. We train another segmentation network with the same architecture for defect segmentation from section 3.3.4 to evaluate if the defects in the generated labels match the images after manipulating the generation by our distribution loss. Since this is an extension of the hypothetical scenario from the previous section, we use the same labeled data set for the GAN. The training

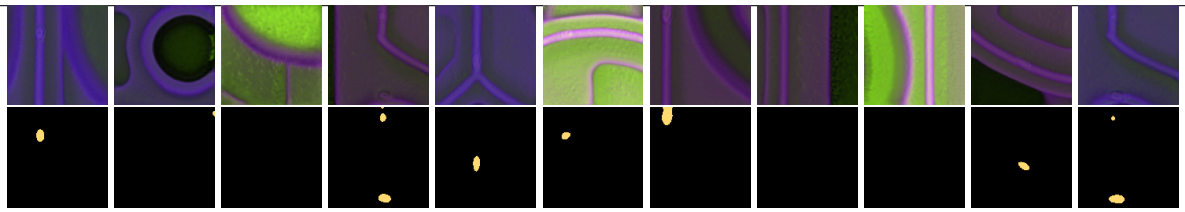


Figure 4.7: Synthesized pump patch image and labels generated from our method **without** distribution loss

set is also roughly half healthy pump patch images and half unhealthy pump patch images with defects, and the testing set has a similar distribution.

Suppose the segmentation model can produce comparable results after adding generated pump patch images with defects. In that case, it indicates that the synthesized defects are pretty similar to the real ones for the segmentation model. In each experiment, we sample 500 pump patch image/label pairs with defects from the data points synthesized by our method and include them in the training set of the segmentation model. All the experiments are repeated three times, and the mean testing results are presented in Table 4.4.

To accurately measure the mIoU score of defects, we consider only those cases whose target labels are non-zero. Besides, we compute the recall and precision to take the false negative and false positive cases into account. The definition of a confusion matrix is slightly different here: A pump patch with pixels labeled as defects is defined as positive. It is true positive only when the predicted and target defects are overlapped; otherwise, it is a false positive. It is a true negative only when our model predicts no defect pixels; otherwise, it is a false negative.

From the results in Table 4.4, we can observe that we have marginal improvement for the mIoU of defects and recall after dataset augmentation. It indicates that the pump images defects generated from our method may provide some new information for the segmentation model. In addition, it also implies that our generated data points are comparable to the real data and that the distribution loss does not adversely affect the quality of generated defects.

Augmentation Method	mIoU of defects	Accuracy	Precision	Recall	F1-score
Baseline (None)	0.4844	0.8480	0.7943	0.9191	0.8521
SemanticGAN	0.4696	0.8382	0.7857	0.9114	0.8439
Ours w/o $\mathcal{L}_d$	0.4927	0.8451	0.7927	0.9162	0.8512
Ours w/ $\mathcal{L}_d$	0.4921	0.8376	0.7741	0.9315	0.8455

Table 4.4: Mean testing results for pump patch segmentation in the hypothetical scenario where pump patches with or without defects are around half and half.

4.4 Discussion

Our experiments on defects generation of the pump dataset and earrings generations of the CelebAMask-HQ dataset show that our modification to SemanticGAN, including semantic gradients for images and distribution loss, make the GAN generate more pixels of minority class. Although the distribution loss causes some artifacts in the semantic labels for the CelebAMask-HQ dataset, the concept of motivating the GAN model to generate minority class labels is still promising. Our experiments of applying the distribution loss

to our industrial dataset demonstrate that this loss term can change the defect distribution in the generated semantic labels while generating corresponding defects in images. This additional variation introduced by the distribution loss may make the generated data points more informative for the downstream task.

There are many possible improvement for our method, such as applying heuristic weighting schema to the distribution loss or incorporating other loss terms to smooth the labels. The other main challenge is to ensure that the corresponding image pixels are accurately generated according to the semantic labels. In our experiment of earrings generation in the CelebAMask-HQ dataset, it is hard to tell whether the earrings are synthesized correctly in the images because any artifacts can look like earrings. There are no perfect metrics to evaluate the quality of these details in generated images. Replacing the image/label pair discriminator with other strong discriminators to enhance the alignment between generated images and labels may help the generator produce better quality on minority classes. If we can tackle these challenges, this may further contribute to the problem of minority classes segmentation.

Chapter 5

Conclusion

It's still challenging for the GAN to pay attention to small details of the images, such as defects, especially when the size of training set is limited. Our defect-pasted augmentation is proved to be effective in encouraging GANs to generate more defects from both image generation experiments of StyleGAN2-ADA and SemanticGAN.

Our research shows that dataset augmentation for industrial dataset by state-of-the-art GAN models is completely feasible. The GAN models can generate images that are almost indistinguishable from real images. However, the synthesized images don't always provide new information for the downstream tasks. Like all the previous work on GAN-based dataset augmentation, our results also demonstrate that oversampling the minority classes by images generated from GAN models can improve the classification performance. Unexpectedly, adding extra synthesized images and labels aren't very informative for the segmentation model. The idea of generating both images and labels for dataset augmentation still has a lot of potential, but we just need more studies on how to add meaningful variation to the synthesized images and labels.

Extended from our research on generating images and labels, we would like to have further control over the distribution of synthesized labels by adding a distribution loss. This loss term can make the GAN generate more pixels of targeted minority class. However, this loss term causes some artifacts in the semantic labels when there are many classes, as shown in our experiments on the CelebAMask-HQ dataset. As for the pump part dataset, our method generates high-quality semantic labels without making artifacts and generate images correspond to the labels. The segmentation experiments indicates that the generated defects after applying distribution loss are not only comparable to the real defects but also possibly provide additional information for the segmentation model.

Appendix A

Supplementary Material for Industrial Dataset Augmentation

This appendix will supplement the industrial dataset augmentation chapter. The following tables present the performance of downstream tasks when we add different number of images generated by our method to the training set. The results in Table A.1 demonstrate that the recall will increase when more pump patch images with defects are added to the training set because all augmented images are with defects. From Table A.2, we can conclude that it causes a drop in performance because the model starts to overfit the generated fake image/label pairs, after adding 1500 generated image when the real/fake ratio is approximately 1.

Size	Accuracy	Precision	Recall	F1-score
0	**0.9401**	0.2837	0.1194	0.1668
1000	0.9273	0.2968	0.3192	**0.3603**
2000	0.9279	**0.3201**	0.3699	0.3402
4000	0.9169	0.2884	0.4348	0.3449
6000	0.8931	0.2265	**0.4503**	0.2978

Table A.1: Pump patch classification results with different size of dataset augmentation of our method.

Size	IoU of defects	IoU of healthy part	mean IoU
0	0.5246	0.9869	0.7558
500	0.5234	0.9866	0.7550
1000	0.5109	0.9852	0.7481
1500	0.3453	0.9510	0.6481

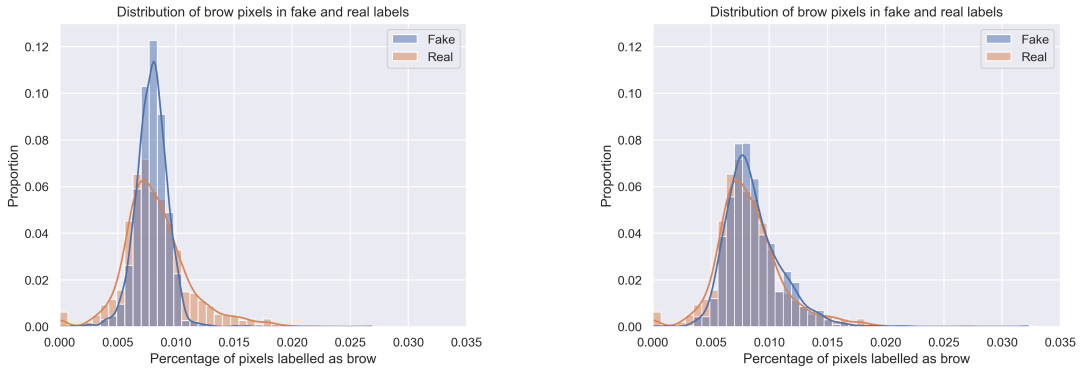
Table A.2: Pump patch defects segmentation results with different different size of dataset augmentation of our method.

APPENDIX A. SUPPLEMENTARY MATERIAL FOR INDUSTRIAL DATASET AUGMENTATION

Appendix B

Supplementary Material for Minority Semantic Class Generation: CelebAMask-HQ Dataset

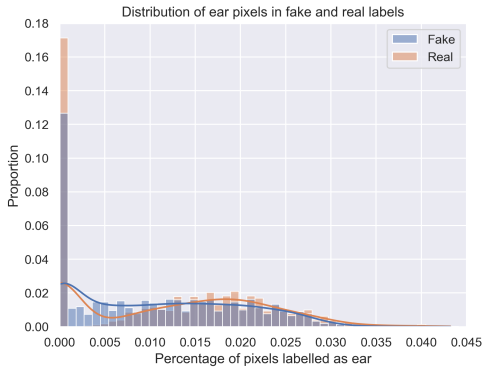
This appendix provides more information about the pixel distribution of more classes in synthesized labels of CelebAMask-HQ from our GAN models, which are demonstrated by the histograms of pixels for other classes.



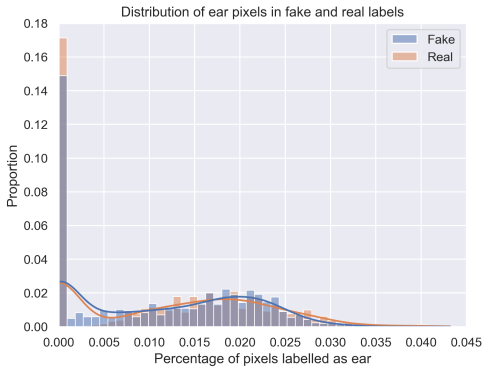
(a) SemanticGAN (b) Ours w/o $\mathcal{L}_d$

Figure B.1: Pixel distribution of class **brow** in synthesized labels

APPENDIX B. SUPPLEMENTARY MATERIAL FOR MINORITY SEMANTIC CLASS GENERATION: CELEBAMASK-HQ DATASET

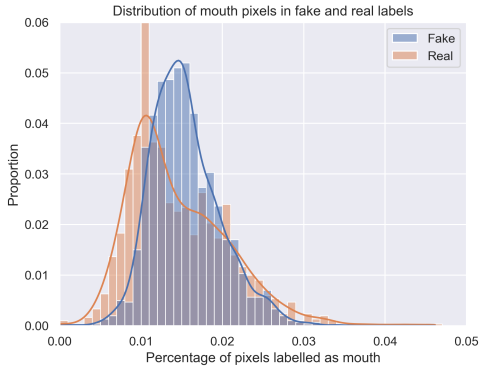


(a) SemanticGAN

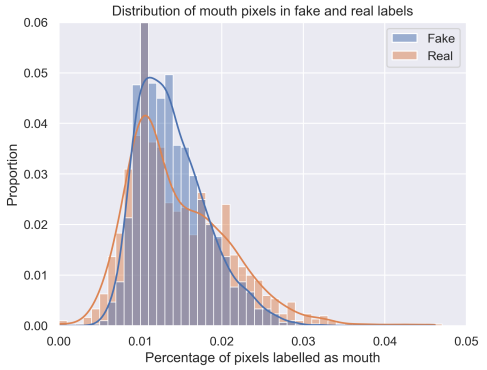


(b) Ours w/o $\mathcal{L}_d$

Figure B.2: Pixel distribution of class **ear** in synthesized labels

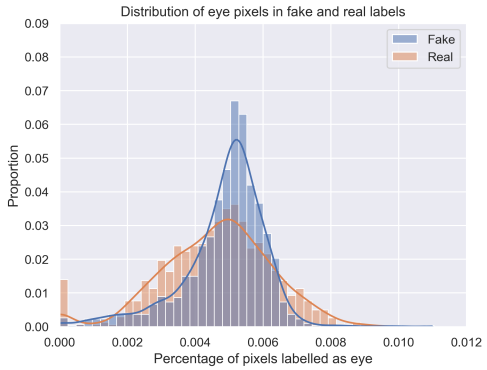


(a) SemanticGAN

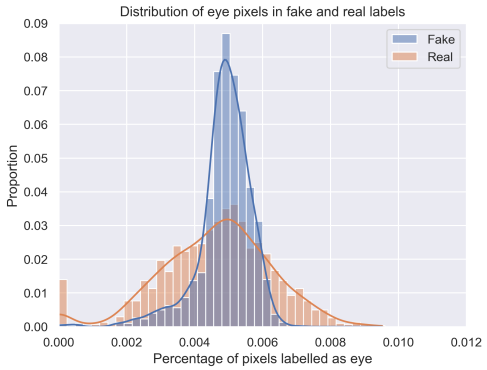


(b) Ours w/o $\mathcal{L}_d$

Figure B.3: Pixel distribution of class **mouth** in synthesized labels

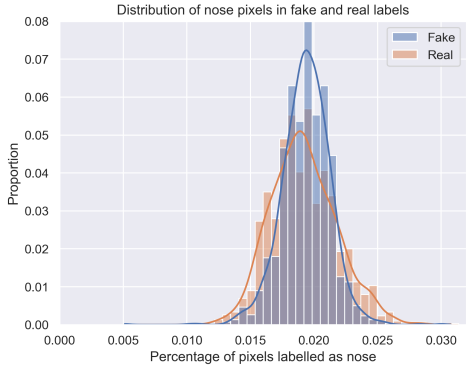


(a) SemanticGAN



(b) Ours w/o $\mathcal{L}_d$

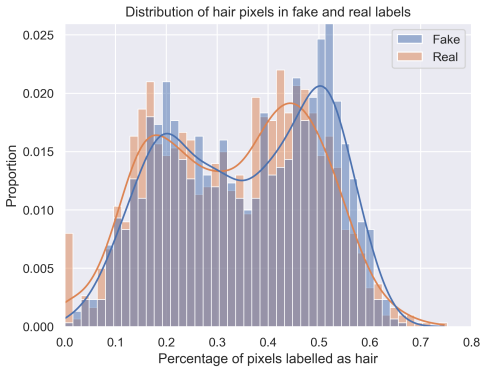
Figure B.4: Pixel distribution of class **eye** in synthesized labels



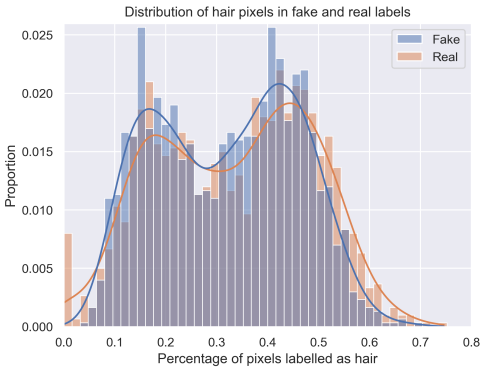
(a) SemanticGAN

(b) Ours w/o $\mathcal{L}_d$

Figure B.5: Pixel distribution of class **nose** in synthesized labels



(a) SemanticGAN



(b) Ours w/o $\mathcal{L}_d$

Figure B.6: Pixel distribution of class **hair** in synthesized labels

APPENDIX B. SUPPLEMENTARY MATERIAL FOR MINORITY SEMANTIC CLASS GENERATION: CELEBAMASK-HQ DATASET

Appendix C

Supplementary Material for Minority Semantic Class Generation: Pump part Dataset

This appendix provides more information about the defect pixels distribution in synthesized labels of our GAN models, including the statistics of more checkpoints (Table C.1 and C.2), which indicate that our method with distribution loss has stable and better statistics for all checkpoints, and the histogram of defect pixels (Figure C.1), showing the distribution change by the loss term.

Method	136K	138K	140K	142K	144K
SemanticGAN	0.56	0.30	0.31	0.45	0.29
Ours w/o $\mathcal{L}_d$	0.60	0.52	**0.66**	**0.65**	0.52
Ours w/ $\mathcal{L}_d$	**0.63**	**0.69**	0.65	0.61	**0.73**

Table C.1: Percentage of defect pixels in synthesized pump patch labels of different checkpoint models.

Method	136K	138K	140K	142K	144K
SemanticGAN	65.63	40.53	46.49	54.24	40.34
Ours w/o $\mathcal{L}_d$	57.46	56.47	61.86	59.83	56.48
Ours w/ $\mathcal{L}_d$	**69.51**	**63.76**	**66.96**	**67.59**	**73.19**

Table C.2: Percentage of synthesized pump patch labels with defects of different checkpoint models.

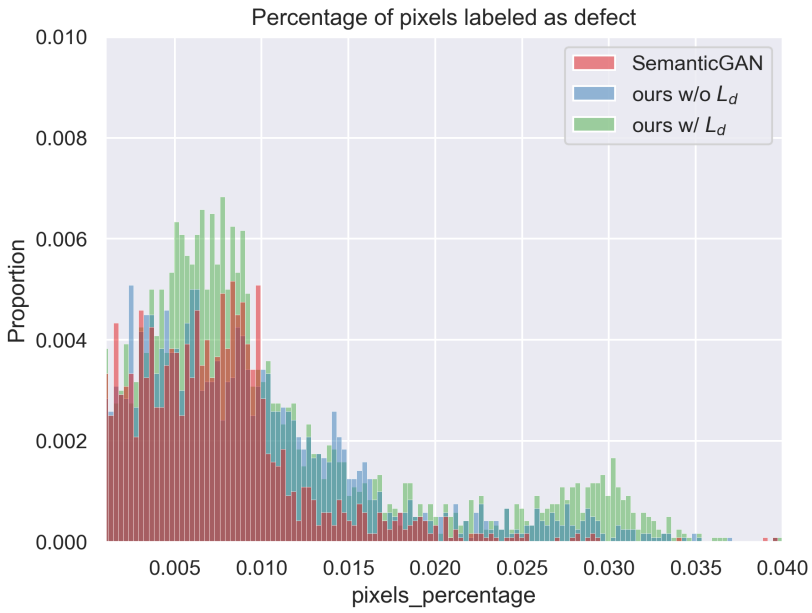


Figure C.1: Percentage of pixels labeled as defects in generated labels.

Bibliography

[1] D.C Dowson and B.V Landau. The fréchet distance between multivariate normal distributions. *Journal of Multivariate Analysis*, 12(3):450–455, 1982. ISSN 0047-259X. doi: https://doi.org/10.1016/0047-259X(82)90077-X. URL https://www.sciencedirect.com/science/article/pii/0047259X8290077X.

[2] Maayan Frid-Adar, Idit Diamant, Eyal Klang, Michal Amitai, Jacob Goldberger, and Hayit Greenspan. Gan-based synthetic medical image augmentation for increased cnn performance in liver lesion classification. *Neurocomputing*, 321:321–331, Dec 2018. ISSN 0925-2312. doi: 10.1016/j.neucom.2018.09.013. URL http://dx.doi.org/10.1016/j.neucom.2018.09.013.

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

[4] Changhee Han, Kohei Murao, Tomoyuki Noguchi, Yusuke Kawata, Fumiya Uchiyama, Leonardo Rundo, Hideki Nakayama, and Shin'ichi Satoh. Learning more with less. *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, Nov 2019. doi: 10.1145/3357384.3357890. URL http://dx.doi.org/10.1145/3357384.3357890.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[6] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu. Squeeze-and-excitation networks, 2019.

[7] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization, 2017.

[8] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks, 2018.

[9] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation, 2018.

[10] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks, 2019.

[11] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In *Proc. NeurIPS*, 2020.

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

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[14] Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. Maskgan: Towards diverse and interactive facial image manipulation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[15] Daiqing Li, Junlin Yang, Karsten Kreis, Antonio Torralba, and Sanja Fidler. Semantic segmentation with generative models: Semi-supervised learning and strong out-of-domain generalization. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[16] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection, 2017.

[17] Bingchen Liu, Yizhe Zhu, Kunpeng Song, and Ahmed Elgammal. Towards faster and stabilized gan training for high-fidelity few-shot image synthesis, 2021.

[18] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[19] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets, 2014.

[20] Evangelos Ntavelis, Andrés Romero, Iason Kastanis, Luc Van Gool, and Radu Timofte. SESAME: Semantic Editing of Scenes by Adding, Manipulating or Erasing Objects. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 394–411. Springer International Publishing, 2020. ISBN 978-3-030-58542-6.

[21] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

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

[23] Aaditya Ramdas, Nicolas Garcia, and Marco Cuturi. On wasserstein two sample testing and related families of nonparametric tests, 2015.

[24] Vignesh Sampath, Iñaki Maurtua, Juan José Aguilar Martín, and Aitor Gutierrez. A survey on generative adversarial networks for imbalance problems in computer vision tasks. *Journal of Big Data*, 8(1): 27, 2021. doi: 10.1186/s40537-021-00414-0. URL https://doi.org/10.1186/s40537-021-00414-0.

[25] Connor Shorten and Taghi M. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):60, 2019. doi: 10.1186/s40537-019-0197-0. URL https://doi.org/10.1186/s40537-019-0197-0.

[26] Vadim Sushko, Edgar Schönfeld, Dan Zhang, Juergen Gall, Bernt Schiele, and Anna Khoreva. You only need adversarial supervision for semantic image synthesis, 2021.

[27] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans, 2018.

[28] Pavel Yakubovskiy. Segmentation models pytorch. `https://github.com/qubvel/segmentation_models.pytorch`, 2020.

[29] Yuxuan Zhang, Huan Ling, Jun Gao, Kangxue Yin, Jean-Francois Lafleche, Adela Barriuso, Antonio Torralba, and Sanja Fidler. Datasetgan: Efficient labeled data factory with minimal human effort. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10145–10155, 2021.

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

[31] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks, 2020.

[32] Xinyue Zhu, Yifan Liu, Zengchang Qin, and Jiahong Li. Data augmentation in emotion classification using generative adversarial networks, 2017.