Repurposing GANs for One-shot Semantic Part Segmentation

Nontawat Tritrong* Pitchaporn Rewatbowornwong* Supasorn Suwajanakorn
VISTEC, Thailand
{nontawat.t_s19, pitchaporn.r_s18, supasorn.s}@vistec.ac.th

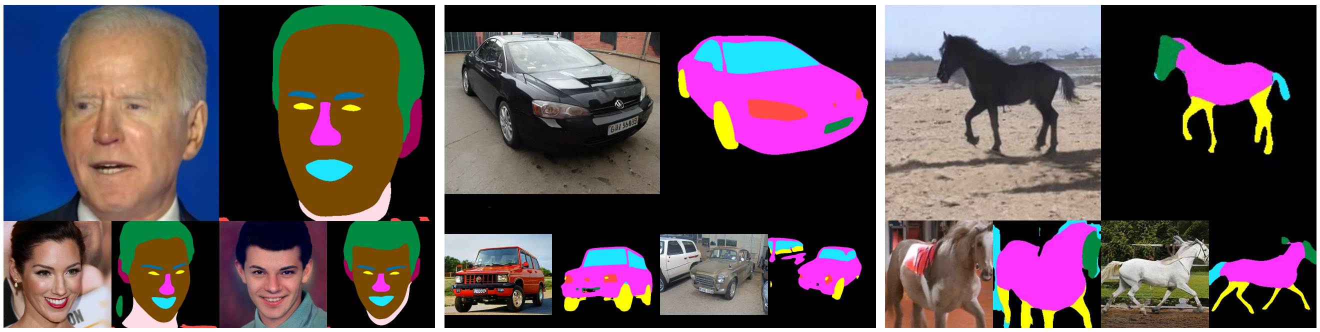


Figure 1: One-shot segmentation results. In each task, our segmentation network is given only one example of part labels.

Abstract

While GANs have shown success in realistic image generation, the idea of using GANs for other tasks unrelated to synthesis is underexplored. Do GANs learn meaningful structural parts of objects during their attempt to reproduce those objects? In this work, we test this hypothesis and propose a simple and effective approach based on GANs for semantic part segmentation that requires as few as one label example along with an unlabeled dataset. Our key idea is to leverage a trained GAN to extract a pixel-wise representation from the input image and use it as feature vectors for a segmentation network. Our experiments demonstrate that this GAN-derived representation is "readily discriminative" and produces surprisingly good results that are comparable to those from supervised baselines trained with significantly more labels. We believe this novel repurposing of GANs underlies a new class of unsupervised representation learning, which can generalize to many other tasks. More results are available at https://RepurposeGANs.github.io/.

1. Introduction

After seeing what an elephant trunk looks like for the first time, a young child can identify this conspicuous part for the whole herd. This key capability in humans is still a fundamental challenge in computer vision. That is, how can a machine learn to identify an object or its parts by seeing only one or few examples? A kid does, however, have access to prior visual information learned constantly throughout the years, and he or she could quickly learn to identify human ears perhaps by utilizing the experience of seeing many faces before. In this paper, we tackle a problem inspired by this scenario. Given a large photo collection of human faces, or any other object classes, our goal is to identify the pixels corresponding to each semantic part for unseen face images given *very few* images with part annotations.

This problem setup is different from the typical definition of few-shot learning, which describes a problem where a learning algorithm trained with many object classes needs to classify or operate on new classes with few supervised examples of those new classes. In contrast, our novel few-shot setup involves a single object class with few annotated examples and no other training data from any other classes. Many methods are proposed in this area of few-shot learning, and the general idea is to apply prior knowledge learned externally to the few-shot task. Examples include meta learning [40] and prototype representation [31, 51] which extract information from annotations of non-target classes or image-level annotations to be used as prior knowledge. However, most of these approaches still learn from some supervised task that requires expensive labels or part annotations. In this work, we introduce a new direction that

*Authors contributed equally to this work.

uses a generative model, specifically a generative adversarial network (GAN) [19], to learn this prior knowledge from zero labels and apply it to semantic segmentation.

GANs have been highly successful in modeling the data distribution and generating realistic images [25, 26, 4]. We hypothesize that GANs need to learn meaningful structural information of objects in order to synthesize them correctly, and the generative computations required to synthesize different parts of object could provide useful discriminative information for other tasks [2, 36]. Our main contribution is a method that leverages a trained GAN to extract meaningful pixel-wise representations from images. These representations can then be used directly for semantic part segmentation. Our experiments show that GANs are incredibly effective for learning such representations and can achieve surprisingly good segmentation results with only one example label (see Figure 1). To our knowledge, this is the first time such high-quality results are achieved on one-shot part segmentation.

Despite its remarkable results, this core idea alone heavily relies on time-consuming latent optimization and requires the test image to lie close to the image distribution learned by GANs. In this paper, we also demonstrate a simple extension, called auto-shot segmentation, that can bypass the latent optimization, leading to faster and more efficient predictions. And importantly, by performing geometric data augmentation during auto-shot training, we can segment multiple objects with different sizes and orientations all at once—a real-world scenario unseen during training.

To summarize, our main contribution is a novel use of GANs for unsupervised pixel-wise representation learning, which achieves surprising and unprecedented performance on few-shot semantic part segmentation. Our findings reveal that such a representation is readily discriminative. We also demonstrate how to extend the main idea to real-world scenarios to address some of the domain gap between the GAN's training data and real-world images.

2. Related Work

Representation Learning The goal of representation learning is to capture the underlying information from raw data that is useful and more convenient to process for downstream tasks. Many approaches learn these representations from solving one task and employ them to help improve the performance on another task [12, 41, 18, 39, 10, 23]. Recent studies have demonstrated that representations learned through a self-supervised task can boost the performance of supervised tasks such as classification and segmentation. Examples of these self-supervision tasks include spatial relative position prediction [11, 33], image colorization [28], and image transformation classification [25, 15]. In contrast, our work explores a representation learned from a generative task, i.e., image synthesis, and extracts feature vectors at the pixel level that is more effective for segmentation problems.

Generative Models Deep generative models have shown promising results in modeling image distribution, thus enabling synthesis of realistic images. There are several classes of visual generative models which include autoregressive models [34, 48], autoencoders based on encoder-decoder architectures such as VAE and its variants [27, 21, 6, 47], and generative adversarial networks (GANs) [19]. Currently, GANs are best in class in image synthesis and have been applied to many other tasks such as image completion [35] and image-to-image translation [24, 58]. State-of-the-art GANs, such as StyleGAN2 [26] and BigGAN [4], can generate extremely realistic images at high resolution. Our work employs GANs for representation learning, and we provide a study that shows the effectiveness of GANs over alternative models.

Motivated by the impressive results from GANs, numerous studies attempt to understand and interpret the internal representations of GANs. GANs dissection [3] applies an external segmentation model to find the relationship between feature maps and output objects, which also allows adding and removing objects in the output image. Suzuki et al. [44] show that interchanging activations between images can result in interchanging of objects in the output image. Edo et al. [9] use clustering to find distinctive groups of feature maps and allow spatially localized part editing. Tsutsui et al. [46] improve one-shot image recognition by combining images synthesized by GANs with the original training images. [13] uses representations learned from BigGAN and achieves state-of-the-art performance on unsupervised representation learning on ImageNet [38].

Some other studies analyze GANs through manipulation of the latent code and attempt to make GANs' internal representations more interpretable. Chen et al.[7] use mutual information to force the network to store human-interpretable attributes in their latent code. Shu et al.[42] solve a similar problem via an additional encoder network, which allows users to have control over the generated results. AttGAN [20] exploits an external classifier network to enable attribute editing. Voynov & Babenko [49] propose an unsupervised method to discover interpretable directions in the latent space. In this paper, we leverage the insight that GANs internal representations are tightly coupled to the generated output and that they can hold useful semantic information.

Semantic Part Segmentation Semantic part segmentation aims to segment parts within an object as opposed to objects within a scene as in semantic segmentation. This problem can be more challenging because two parts sometimes do not have a visible boundary between them, such as nose and face. Considerable progress has been made in semantic part segmentation [50, 52, 45, 32], but these techniques demand a vast number of pixel-wise annotations.

To avoid using pixel-wise annotations, some approaches rely instead on other kinds of annotations that are cheaper to obtain, such as keypoints [16], body poses [54], or edge maps [56]. However, they are often inflexible and only work on some specific domains, such as human body parts. Some other attempts forgo the annotations completely with self-supervised techniques. For example, [22] uses equivariance, geometric, and semantic consistency constraints to train a segmentation network, and [29, 53] exploit motion information from videos. One main drawback of these unsupervised methods is that there is little control over the partition of object parts, which can lead to arbitrary segmentation. Unlike these approaches, our method allows complete control over the partition of object parts by requiring only few annotated examples.

Few-shot Semantic Segmentation Past research has attempted to solve segmentation with few annotations. A meta learning approach [40] first trains a segmentation network on an annotated dataset then fine-tunes the network parameters on one annotation of the target class. Prototypical methods [14, 31, 51] use a support set to learn a prototype vector for each object class. Both meta learning and prototypical methods construct two training branches where the support branch is trained on annotations of non-target classes or image-level annotations, and the query branch then takes an input image as well as the extracted feature to predict segmentation masks. Similarity guidance network [55] masks off the background in the support image, then finds the pixels in the query branch with similar foreground features. Some work [43, 5] segment objects in all video frames with only the first frame annotated. Nonetheless, these methods have not shown success in semantic part segmentation. Meta learning requires annotation masks of similar object classes, and hence learning part-specific prototypes is not viable. Leveraging the information from the support set is also difficult due to the lack of part-level annotations. In contrast, our representation extracted from GANs contains part-level information and can be learned without supervision.

3. Approach

Our problem concerns semantic part segmentation with the following novel setup. Given a set of unlabeled images and a few images (1-10) with part annotations from a single object class, our goal is to part-segment an unseen object from the same class. These part annotations can be specified by the user with binary masks. Note that semantic part segmentation can also be considered as an n-way *per-pixel* classification problem where n is the number of parts.

This problem would become trivial if there existed a function f that maps each pixel value, which by itself lacks semantic meaning, to its own feature vector that contains

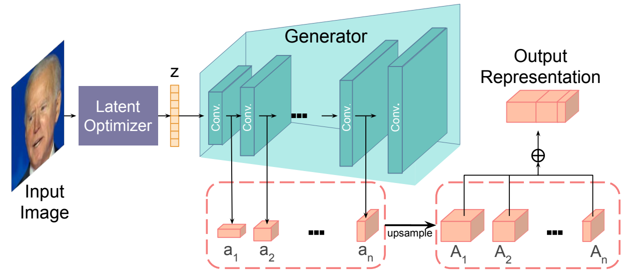


Figure 2: **Representation extraction** To extract a representation from an image, we embed the image into the latent space of GAN by optimizing for the latent z that reproduces the input image. z is then fed to the generator and we collect multiple activation maps $a_1, a_2, ..., a_n$ of dimensions $(h_1, w_1, c_1), ..., (h_n, w_n, c_n)$. Each of these maps is upsampled to A_i with dimension (h_n, w_n, c_i). The representation is a concatenation of all A_i along the channel dimension.

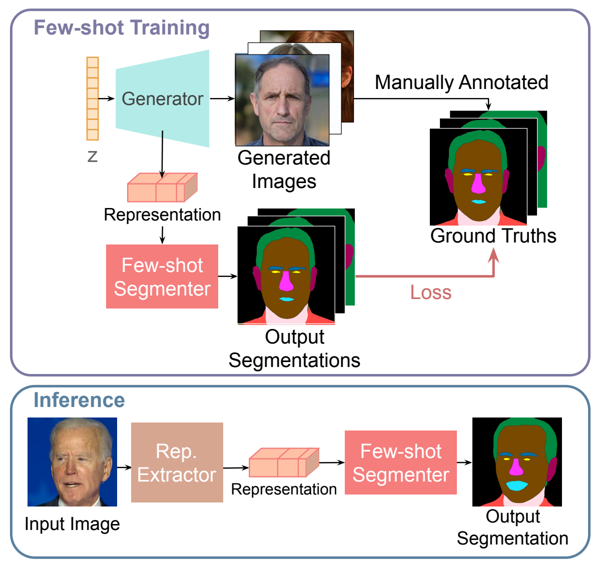


Figure 3: **Few-shot segmentation pipeline** For training, we use a trained GAN to generate a few images along with their representations by feeding random latent codes. Then, we manually annotate these images and train our few-shot segmenter to output segmentation maps that match our annotated masks. For inference, we extract a representation from a test image (Figure 2) then input it to the few-shot segmenter to obtain a segmentation map.

discriminative information for part classification. We propose to derive such a function from a GAN trained to synthesize images of the target class. In the following sections, we will explain how a GAN is utilized for this task, how to use the computed per-pixel features for segmentation, and finally a simple extension that allows segmentation without requiring a GAN or its expensive mapping during inference.

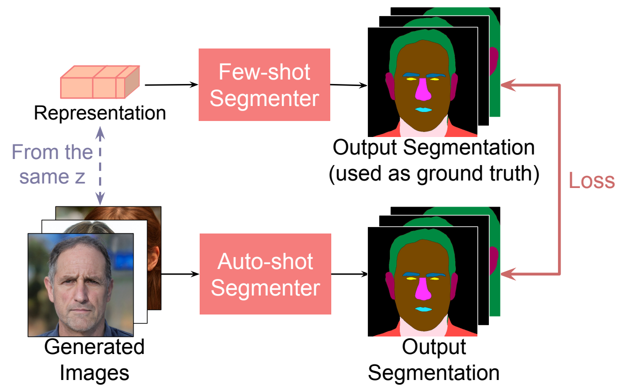


Figure 4: **Auto-shot segmentation pipeline** during training, the auto-shot segmenter uses generated images from the trained GAN as input and segmentation masks predicted by the few-shot segmenter as ground truth.

3.1. Representation Extraction from GANs

Using GANs as a mapping function is not straightforward simply because GANs take as input a random latent code, not the image pixels to be mapped. To understand our process, first consider a typical scenario where we generate an image by feeding a random latent code to a convolutional-based GAN. In this case, the synthesized output image is constructed by the generator through a series of spatial convolutions, and each output pixel is a result of a unique generative computation that can be traced back through each convolution layer down to the initial latent code.

Our key idea is to use these unique computational "paths" for feature representation. Generally, the computational path for generating a pixel is a directed acyclic graph with nodes representing the network parameters or the input latent code involved in the computation of that pixel. However, in our work these nodes represent activation values, and we simply represent the path with a single sequence of activations from all layers within the generator that are spatially aligned with that pixel. In particular, as shown in Figure 2, we extract the activation map from every layer (or some subset of layers) of the generator, $a_1, a_2, \ldots, a_n$, each with dimension (h_i, w_i, c_i), and compute our pixel-wise representation as

$$F = \mathbb{U}(a_1) \oplus_c \mathbb{U}(a_2) \oplus_c ... \oplus_c \mathbb{U}(a_n) \tag{1}$$

where $\mathbb{U}(\cdot)$ spatially upsamples the input to the size of the largest activation map (h_n, w_n) and $\oplus_c$ is a concatenation along the channel dimension. This process maps each 3-dimensional RGB pixel to a $C-$dimensional feature vector, where $C = \sum_{i=1}^{n} c_i$.

Normally, this extraction process only works for images that are synthesized by the generator and cannot be used directly for real test images. However, given any test image, one can optimize for a latent code that generates that given test image with any gradient-based optimization or with more sophisticated schemes [26, 1, 17]. The resulting latent code then allows the feature map to be constructed in a similar manner.

3.2. Segmentation with Extracted Representation

To solve few-shot segmentation, we first train a GAN on images of our target class and generate k random images by feeding it random latent codes. Then, we compute the feature maps and manually annotate object parts for these k images. The k feature maps and annotations together form our supervised training pairs which can be used to train a segmentation model, such as a multilayer perceptron or a convolutional network (see Figure 3). To segment a test image, we compute a pixel-wise feature map for the test image using the aforementioned latent code optimization and feed it to our trained segmentation network.

3.3. Extension: Auto-shot Segmentation Network

Computing pixel-wise feature vectors using a GAN does have a number of restrictions. First, the test image needs to lie close to the image distribution modeled by the GAN; otherwise, the latent optimization may fail to reproduce the test image, leading to poor feature vectors. This constrain can be severely restricting for classes like human face because the input image has to contain exactly one face aligned and centered similar to the trainset. Second, relying on a GAN to generate feature vectors through the latent optimization process is expensive and time-consuming as it requires multiple forward-backward passes through the generator.

To overcome these limitations, we use our trained GAN to synthesize a large set of images and predict segmentation maps for those images using our network to form paired training data. To retain all the probability information from our network's prediction process, each pixel in our segmentation maps is represented by a set of logit values of all part labels rather than a single part ID. That is, we do not apply softmax and argmax to produce the segmentation maps. With this training data, we train another network, e.g. a UNet [37], to solve the segmentation in a single network pass from raw images without relying on a GAN or its feature mapping (see Figure 4). We call this process auto-shot segmentation. Additionally, we apply data augmentation that allows detection of objects at different scales and orientations. Interestingly, we demonstrate how this simple approach can successfully segment multiple object instances at the same time with good quality in our experiments and supplementary video.

4. Implementation

Our training pipeline starts by training a GAN on a dataset of the target class. Then, we use the trained GAN to generate a few images along with their pixel-wise representations (Section 3.1) and manually annotate these images with the desired part segmentation. Finally, we train a few-shot segmentation network that takes as input the pixel-wise representation to predict an output segmentation. For the auto-shot segmentation, we use the same GAN to generate a large dataset of images and use our trained few-shot network to predict segmentation maps for those images. These generated images and their corresponding segmentation maps are then used to train the auto-shot segmentation network.

4.1. Generative Adversarial Network

We use StyleGAN2 [26] for our pipeline. StyleGAN2 has 9 pairs of convolution layers with activation outputs of sizes: 4^2, 8^2, 16^2, 32^2, 64^2, 128^2, 256^2, 512^2, and 1024^2. For feature extraction, we use all pairs except the last which generates the output image. We also use StyleGAN2's projection method proposed in their paper to embed an image into the latent space (latent code optimization explained in Section 3.1).

4.2. Few-shot Segmentation Network

The few-shot network takes the C-channel pixel-wise representation as input and outputs a segmentation map. We explore 2 different architectures: fully convolutional networks (CNN) and multilayer perceptrons (MLP).

CNN: We first use a linear embedding layer (1x1 convolution) to reduce the input dimension from C to 128, followed by 8 convolutional layers with a kernel size of 3 and dilation rates of 2, 4, 8, 1, 2, 4, 8, and 1. The dimensions of the output channels are: 64 for the first 6 layers, 32, and the number of classes. All layers except the output layer use leaky ReLU activation functions.

MLP: We use a 2-layer MLP with 2,000 and 200 hidden nodes for the first and second layers. All layers except the output layer use ReLU activation functions.

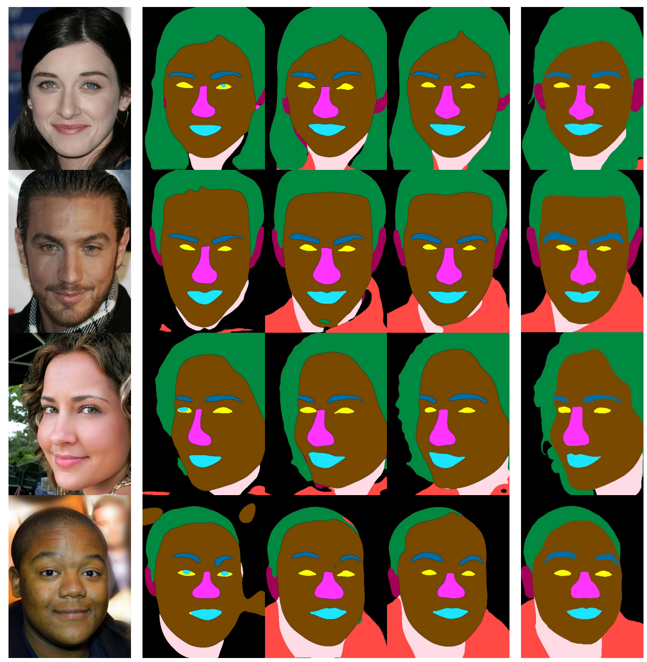
Both MLP and CNN were trained for 1,000 epochs with a cross-entropy loss and a weight decay of 0.001 using Adam optimizer. Our initial learning rate is 0.001 with a decay factor of 0.9 every 50 epochs.

4.3. Auto-shot Segmentation Network

This network is trained with GAN's generated images and their corresponding segmentation maps from the few-shot network. We adopt a UNet architecture described in our supplementary. Additionally, we perform the following data augmentation on this training set: 1) random horizontal flips, 2) random scales between 0.5 and 2, 3) random rotations between -10 and 10 degree, 4) random vertical and horizontal translations between 0% and 50% of the image size. This network is trained for 300 epochs using Adam optimizer with an initial learning rate of 0.001 and a decay factor of 0.1 when the validation score does not decrease within 20 epochs.

Table 1: Weighted IOU scores on few-shot human face segmentation.

Segmentation Network	Shots	3-class	10-class
CNN	1	71.7	77.9
	5	82.1	83.9
	10	**83.5**	**85.2**
MLP	1	75.3	74.1
	5	77.8	79.6
	10	77.2	77.2



Input 1-shot 5-shot 10-shot Ground Truth

Figure 5: Few-shot face segmentation results on CelebAMASK-HQ.

5. Experiments

We perform the following experiments in this section: 1) we evaluate the performance of our few-shot and auto-shot segmenters on 3 object classes and compare them to baselines, 2) we evaluate alternative structures of the few-shot segmentation network. Additionally in our supplementary material, 3) we show segmentation results on videos, 4) we study whether the choice of layers used for feature extraction affects the segmentation performance, 5) we test whether our method can segment parts that have arbitrary or

Table 2: IOU scores of our 10-shot vs auto-shot segmenters on 10-class face segmentation. The auto-shot segmenter is trained with a dataset generated by the 10-shot segmenter. Both techniques have similar performance, which demonstrates the effectiveness of the dataset generation and auto-shot training process.

Network	Weighted IOU	Eyes	Mouth	Nose	Face	Clothes	Hair	Eyebrows	Ears	Neck	BG
10-shot segmenter	85.2	74.0	84.6	82.9	90.0	23.6	79.2	63.1	27.0	73.6	84.2
Auto-shot segmenter	84.5	75.4	86.5	84.6	90.0	15.5	84.0	68.2	37.3	72.8	84.7

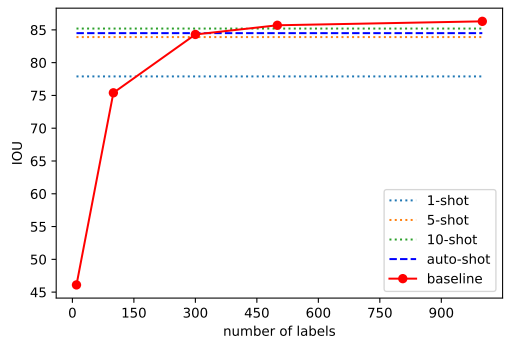


Figure 6: 10-class face segmentation results of supervised baseline and the number of segmentation labels used. Our few-shot segmentation results are shown in dot line for comparison. Supervised baseline consumes over 100 annotations to surpass our 1-shot segmenter, and around 500 annotation to reach same-level of IOU on our 10-shot segmenter.

Table 3: Per-class IOU scores on 3-class human face segmentation.

	Weighted IOU	Eyes	Mouth	Nose
1-shot	71.7	57.8	71.1	76.0
5-shot	82.1	73.6	84.0	82.1
10-shot	83.5	75.9	85.3	82.7

Table 4: IOU scores on PASCAL-Parts car segmentation.

Model	Body	Plate	Light	Wheel	Window	BG	Average
CNN[45]	73.4	41.7	42.2	66.3	61	67.4	58.7
CNN+CRF[45]	75.4	35.8	36.1	64.3	61.8	68.7	57
Ours (Auto-shot)	75.5	17.8	29.3	57.2	62.4	70.7	52.2
OMPS[57]	86.3	50.5	55.1	75.5	65.2	-	66.5
Ours (Auto-shot) w/o bg	76.4	17.5	29.3	52.5	64.1	-	47.9

Table 5: IOU scores on PASCAL-Parts horse segmentation. “-” indicates no available result.

Model	Head	Neck	Torso	Neck+Torso	Legs	Tail	BG
Shape+Appearance[50]	47.2	-	-	66.7	38.2	-	-
CNN+CRF[45]	55.0	34.2	52.4	-	46.8	37.2	76.0
Ours (Auto-shot)	50.1	-	-	70.5	49.6	19.9	81.6

unusual shapes that do not correspond to any semantic parts, 6) we explore and evaluate features learned from other generative models, such as VAE, or from other supervised and self-supervised learning methods and compare against our features learned from GANs.

Table 6: Average IOU scores on PASCAL-Parts horse segmentation.

Model	RefineNet [16]	Pose-Guided [32]	Ours(Auto-shot)
Horse IOU	36.9	60.2	53.1

5.1. Semantic Part Segmentation

We evaluate segmentation performance of the few-shot and auto-shot segmenters on 3 object classes: human face, car, and horse.

Datasets: To train the few-shot segmenter, we use face images and annotated segmentation masks from CelebAMask-HQ [30]. For horse and car, we use images generated by pretrained StyleGAN2s and manually annotate them ourselves. For the auto-shot segmenter, we use 5,000 images generated from each GAN trained on each object class and the predicted annotations from the few-shot segmenter. Our models are evaluated on CelebAMask-HQ for faces and PASCAL-Part dataset [8] for car and horse.

Evaluation metric: We use intersect-over-union (IOU) to evaluate individual object parts and report weighted IOU scores, where the weight of each class is the ratio of the number of ground-truth pixels belonging to the class to the total number of pixels.

5.1.1 Human Face Part Segmentation

We perform experiments with 12 combinations of settings that vary 1) the architecture of the few-shot segmenter (CNN or MLP), 2) the number of part classes (3 or 10), and 3) the number of examples with part annotations (1-shot, 5-shot, 10-shot). One interesting finding in Table 1 is that the MLP segmenter, which can only look at the features of individual pixels to make per-pixel predictions, performs well and almost similarly to the CNN segmenter that has a wide receptive field and can exploit structure priors for predicting a segmentation map.

Table 2 shows IOU scores of the 10-shot segmenter and auto-shot segmenter on 10-class face segmentation. Surprisingly, the auto-shot segmenter achieves similar IOU scores to those of 10-shot segmenter in all classes except for clothes, even though it relies only on the dataset generated by the 10-shot segmenter. Note that the few-shot network also performs poorly on clothes relative to other parts,

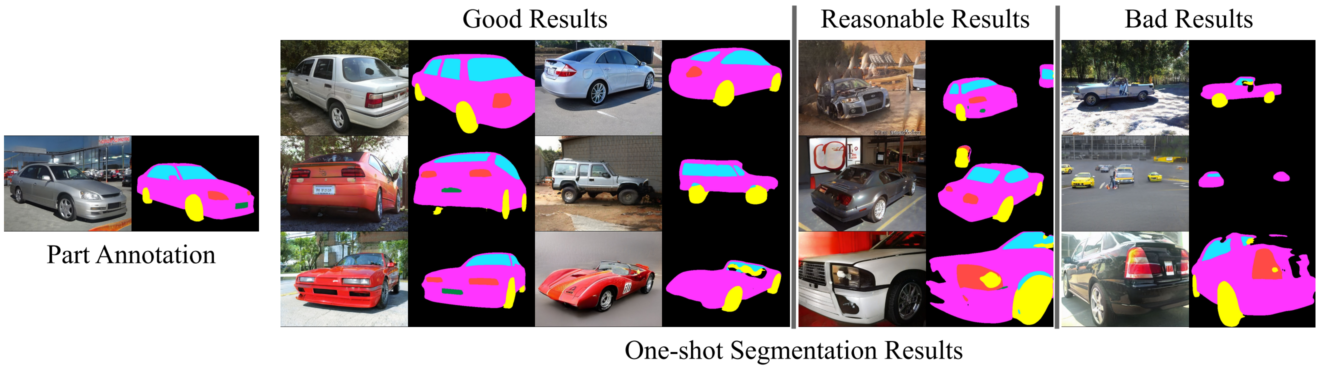


Figure 7: One-shot car part segmentation results on GAN's generated images. The segmentation network can segment car images from varied points of view even though it is trained on annotations of one car from one angle. However, there are some failure cases when the cars appear unusually big or small, or when GANs generate unrealistic cars.

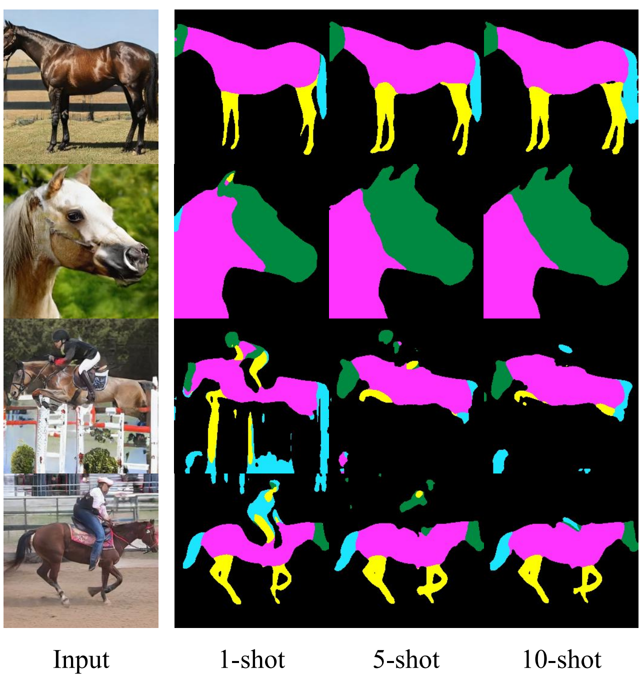


Figure 8: Results on few-shot horse part segmentation from GAN's generated images. Compared to cars and faces with good 1-shot results, horses need more labels. 1-shot horse segmentation often mistakes the rider as a part of horse.

which could be due to the large variation in clothing. The auto-shot segmenter can also segment unaligned images at various scales due to the data augmentation during training. Figure 6 shows a 10-class segmentation comparison between our few-shot and auto-shot segmenters and a supervised baseline which uses the same architecture as the auto-shot segmenter and is trained on ground-truth masks from CelebAMask-HQ with varying numbers of labels. Our few-shot segmenter trained with a single label produces a comparable IOU score to the supervised baseline trained with about 150 labels. And with 10 labels, both of our segmenters match the baseline performance with 500 labels. Qualitative and quantitative results for the CNN-based few-shot network are shown in Figure 5 and Table 3.

5.1.2 Car Part Segmentation

Unlike well-aligned face images in CelebA-HQ, car images in PASCAL-Part have larger variations in pose and appearance. Despite this challenge, our one-shot segmenter produces good segmentation results and can identify wheels, windows, and the license plate shown in Figure 7. We compare our method to DeepCNN-DenseCRF [45] and the Ordinal Multitask Part Segmentation [57] on the car class in PASCAL-Part. Details on the experiment setup can be found in our supplementary material. Table 4 shows our results using the auto-shot segmenter trained on a 10-shot dataset (dataset generated from our few-shot segmenter with 10 labels), which compares favorably to the fully supervised baselines. Note that we compare to [57] by excluding the background class similarly to how their scores were reported.

5.1.3 Horse Part Segmentation

Horse segmentation is more challenging than the other two because horses are non-rigid and can appear in many poses such as standing or jumping. Also, the boundaries between legs and body are not clearly visible. Our one-shot segmenter has lower performance compared to those of faces and cars; however, the result improves significantly with a few more annotations as shown in Figure 8. In Table 5, we compare our auto-shot segmentation (also learned from dataset by 10-shot segmenter) IOU scores on each class to Shape+Appearance [50] and CNN+CRF [45]. Table 6

Figure 9: Some examples of auto-shot segmentation trained with datasets generated by 10-shot segmenters on CelebAMask-HQ, PASCAL-Part Car and PASCAL-Part horse. The pretrained StyleGAN2 for each class was trained on FFHQ, LSUN-Car and LSUN-Horse.

Table 7: IOU scores on 1-shot face segmentation of different size of few-shot segmenters.

Model	Size	Weighted IOU
MLP	0 hidden layers	74.0
	1 hidden layer	72.2
	2 hidden layers	74.1
CNN	S	73.4
	M	75.2
	L	77.9

Figure 10: Results on 1-shot segmentation of MLP-based segmentors containing 0, 1, or 2 hidden layers.

shows the overall IOU scores of our auto-shot segmenter, RefineNet [16], and Pose-Guided Knowledge Transfer [32]. The score of RefineNet is taken from [32]. Our method surpasses RefineNet, and our IOU is slightly lower than Pose-Guided Knowledge Transfer [32] which is a fully-supervised method trained with over 300 annotated images and additional annotated keypoints. Experimental details can be found in our supplementary material. Auto-shot segmentation results are presented in Figure 9.

5.2. Analysis on GAN-derived Representation

One desirable property of a good representation is that it should contain meaningful semantic information in a readily discriminative form. We could test this by evaluating how well a simple linear classifier or smaller networks with limited capability perform given our representation as input.

We evaluate several architectures on one-shot face segmentation: 3 sizes of multilayer perceptrons: i) 0 hidden layers, ii) 1 hidden layer with 2000 nodes, and iii) 2 hidden layers with 2000 and 200 nodes, as well as small, medium, and large convolutional networks described in the supplementary material (Table C - E). We found that a linear classifier (0 hidden-layer) gives reasonable segmentation masks shown in Figure 10; however, a non-linear MLP classifier (2 hidden layers) is needed to obtain more accurate boundaries in complex areas such as hair. As shown in Table 7, L-network obtains the highest IOU score, although smaller networks or even a linear classifier do not perform significantly worse with IOU differences of only around 2.7-5.7.

6. Conclusion

We present a simple and powerful approach that repurposes GANs, used predominantly for synthesis, for few-shot semantic part segmentation. Our novelty lies in the unconventional use of *readily discriminative* pixel-wise representation extracted from the generative processes of GANs. Our approach achieves promising and unprecedented performance that allows part segmentation given very few annotations and is competitive with fully-supervised baselines that require 10-50× more label examples. We also propose a more efficient extension to our segmentation pipeline that bypasses the required latent optimization and generalizes better to real-world scenarios with multiple objects of varying sizes and orientations. We believe this novel use of GANs for unsupervised representation learning can serve as an effective and generic "upstream" task in transfer learning for problems that involve reasoning about object parts, scene semantics, or make pixel-level predictions.

Acknowledgements

This research was supported by PTT public company limited and SCB public company limited.

References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *Proceedings of the IEEE international conference on computer vision*, pages 4432–4441, 2019. 4

[2] Daniel De Freitas Adiwardana, Akihiro Matsukawa, and Jay Whang. Using generative models for semi-supervised learning. In *Medical image computing and computer-assisted intervention–MICCAI*, volume 2016, pages 106–14, 2016. 2

[3] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Bolei Zhou, Joshua B Tenenbaum, William T Freeman, and Antonio Torralba. Gan dissection: Visualizing and understanding generative adversarial networks. *arXiv preprint arXiv:1811.10597*, 2018. 2

[4] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018. 2

[5] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 221–230, 2017. 3

[6] Ricky TQ Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. In *Advances in Neural Information Processing Systems*, pages 2610–2620, 2018. 2

[7] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*, pages 2172–2180, 2016. 2

[8] Xianjie Chen, Roozbeh Mottaghi, Xiaobai Liu, Sanja Fidler, Raquel Urtasun, and Alan Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1971–1978, 2014. 6

[9] Edo Collins, Raja Bala, Bob Price, and Sabine Susstrunk. Editing in style: Uncovering the local semantics of gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5771–5780, 2020. 2

[10] Jifeng Dai, Kaiming He, and Jian Sun. Instance-aware semantic segmentation via multi-task network cascades. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3150–3158, 2016. 2

[11] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pages 1422–1430, 2015. 2

[12] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. A deep convolutional activation feature for generic visual recognition. *UC Berkeley & ICSI, Berkeley, CA, USA*. 2

[13] Jeff Donahue and Karen Simonyan. Large scale adversarial representation learning. In *Advances in Neural Information Processing Systems*, pages 10542–10552, 2019. 2

[14] Nanqing Dong and Eric P Xing. Few-shot semantic segmentation with prototype learning. In *BMVC*, volume 3, 2018. 3

[15] Alexey Dosovitskiy, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with convolutional neural networks. In *Advances in neural information processing systems*, pages 766–774, 2014. 2

[16] Hao-Shu Fang, Guansong Lu, Xiaolin Fang, Jianwen Xie, Yu-Wing Tai, and Cewu Lu. Weakly and semi supervised human body part parsing via pose-guided knowledge transfer. *arXiv preprint arXiv:1805.04310*, 2018. 3, 6, 8

[17] Aviv Gabbay and Yedid Hoshen. Style generator inversion for image enhancement and animation. *arXiv preprint arXiv:1906.11880*, 2019. 4

[18] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014. 2

[19] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 2

[20] Zhenliang He, Wangmeng Zuo, Meina Kan, Shiguang Shan, and Xilin Chen. Attgan: Facial attribute editing by only changing what you want. *IEEE Transactions on Image Processing*, 28(11):5464–5478, 2019. 2

[21] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. 2016. 2

[22] Wei-Chih Hung, Varun Jampani, Sifei Liu, Pavlo Molchanov, Ming-Hsuan Yang, and Jan Kautz. Scops: Self-supervised co-part segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 869–878, 2019. 3

[23] Vladimir Iglovikov and Alexey Shvets. Ternausnet: U-net with vgg11 encoder pre-trained on imagenet for image segmentation. *arXiv preprint arXiv:1801.05746*, 2018. 2

[24] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *CVPR*, 2017. 2

[25] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 2

[26] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020. 2, 4, 5

[27] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2

[28] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Colorization as a proxy task for visual

understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6874–6883, 2017. 2

[29] Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, Nicu Sebe, et al. Motion-supervised co-part segmentation. *arXiv preprint arXiv:2004.03234*, 2020. 3

[30] 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. 6

[31] Yongfei Liu, Xiangyi Zhang, Songyang Zhang, and Xuming He. Part-aware prototype network for few-shot semantic segmentation. *arXiv preprint arXiv:2007.06309*, 2020. 1, 3

[32] Shujon Naha, Qingyang Xiao, Prianka Banik, Md Alimoor Reza, and David J Crandall. Pose-guided knowledge transfer for object part segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 906–907, 2020. 2, 6, 8

[33] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision*, pages 69–84. Springer, 2016. 2

[34] Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*, 2016. 2

[35] Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei Efros. Context encoders: Feature learning by inpainting. In *Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[36] Mehdi Rezagholiradeh and Md Akmal Haidar. Reg-gan: Semi-supervised learning based on generative adversarial networks for regression. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2806–2810. IEEE, 2018. 2

[37] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 4

[38] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 2

[39] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013. 2

[40] Amirreza Shaban, Shray Bansal, Zhen Liu, Irfan Essa, and Byron Boots. One-shot learning for semantic segmentation. *arXiv preprint arXiv:1709.03410*, 2017. 1, 3

[41] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 806–813, 2014. 2

[42] Zhixin Shu, Ersin Yumer, Sunil Hadap, Kalyan Sunkavalli, Eli Shechtman, and Dimitris Samaras. Neural face editing with intrinsic image disentangling. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5541–5550, 2017. 2

[43] Mennatullah Siam, Naren Doraiswamy, Boris N Oreshkin, Hengshuai Yao, and Martin Jagersand. One-shot weakly supervised video object segmentation. *arXiv preprint arXiv:1912.08936*, 2019. 3

[44] Ryohei Suzuki, Masanori Koyama, Takeru Miyato, Taizan Yonetsuji, and Huachun Zhu. Spatially controllable image synthesis with internal representation collaging. *arXiv preprint arXiv:1811.10153*, 2018. 2

[45] Stavros Tsogkas, Iasonas Kokkinos, George Papandreou, and Andrea Vedaldi. Deep learning for semantic part segmentation with high-level guidance. *arXiv preprint arXiv:1505.02438*, 2015. 2, 6, 7

[46] Satoshi Tsutsui, Yanwei Fu, and David Crandall. Meta-reinforced synthetic data for one-shot fine-grained visual recognition. In *Advances in Neural Information Processing Systems*, pages 3063–3072, 2019. 2

[47] Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder. *arXiv preprint arXiv:2007.03898*, 2020. 2

[48] Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. In *Advances in neural information processing systems*, pages 4790–4798, 2016. 2

[49] Andrey Voynov and Artem Babenko. Unsupervised discovery of interpretable directions in the gan latent space. In *International Conference on Machine Learning*, pages 9786–9796. PMLR, 2020. 2

[50] Jianyu Wang and Alan L Yuille. Semantic part segmentation using compositional model combining shape and appearance. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1788–1797, 2015. 2, 6, 7

[51] Kaixin Wang, Jun Hao Liew, Yingtian Zou, Daquan Zhou, and Jiashi Feng. Panet: Few-shot image semantic segmentation with prototype alignment. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9197–9206, 2019. 1, 3

[52] Peng Wang, Xiaohui Shen, Zhe Lin, Scott Cohen, Brian Price, and Alan L Yuille. Joint object and part segmentation using deep learned potentials. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1573–1581, 2015. 2

[53] Zhenjia Xu, Zhijian Liu, Chen Sun, Kevin Murphy, William T Freeman, Joshua B Tenenbaum, and Jiajun Wu. Unsupervised discovery of parts, structure, and dynamics. *arXiv preprint arXiv:1903.05136*, 2019. 3

[54] Zhengyuan Yang, Yuncheng Li, Linjie Yang, Ning Zhang, and Jiebo Luo. Weakly supervised body part parsing with pose based part priors. *arXiv preprint arXiv:1907.13051*, 2019. 3

[55] Xiaolin Zhang, Yunchao Wei, Yi Yang, and Thomas S Huang. Sg-one: Similarity guidance network for one-shot semantic segmentation. *IEEE Transactions on Cybernetics*, 2020. 3

[56] Ziwei Zhang, Chi Su, Liang Zheng, and Xiaodong Xie. Correlating edge, pose with parsing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8900–8909, 2020. 3

[57] Yifan Zhao, Jia Li, Yu Zhang, Yafei Song, and Yonghong Tian. Ordinal multi-task part segmentation with recurrent prior generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 6, 7

[58] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017. 2