This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Network-free, unsupervised semantic segmentation with synthetic images

Qianli Feng Raghudeep Gadde Wentong Liao Eduard Ramon Aleix Martinez
Amazon

Figure 1. Semantic segmentation results of scenes and single objects. Top row: Segmentation of couches (blue mask) and coffee tables (orange mask) in living room scene images. Middle row: Foreground-background face segmentation as well as fine-grained semantic segmentation of facial components. Bottom row: Foreground-background segmentation of animals.

Abstract

We derive a method that yields highly accurate semantic segmentation maps without the use of any additional neural network, layers, manually annotated training data, or supervised training. Our method is based on the observation that the correlation of a set of pixels belonging to the same semantic segment do not change when generating synthetic variants of an image using the style mixing approach in GANs. We show how we can use GAN inversion to accurately semantically segment synthetic and real photos as well as generate large training image-semantic segmentation mask pairs for downstream tasks.

1. Introduction

Semantic segmentation is a computer vision problem with countless important applications, including self-driving cars, medical image analysis, and image content generation and editing [5, 11, 12, 19]. Yet, attaining accurate semantic segmentation masks remains an open problem [18, 28]. A recent proposed solution is to synthesize large training data-sets of photo-realistic images and their masks using generative models like Generative Adversarial Networks (GANs) [1, 6, 18, 19, 28]. However, these methods require 1. adding and training an extra neural network to synthesize the mask, increasing model and training complexity, and 2. very costly pixel-wise human annotations on a large set of training images for every type of object and

scene of interest [19, 28].

Here, we propose a new algorithm that does not require the addition of any extra network, costly pixel-wise human annotations, or supervised training. Our key observation is that the correlation of a set of pixels belonging to the same semantic segment do not change when generating synthetic variants of an image using the style mixing approach [14]. This allows us to derive an unsupervised algorithm to generate highly accurate semantic segmentation masks without the need to incorporate new nets or layers to existing ones or the need to re-train them. We show how our algorithm can be used to semantically segment real photos, generate synthetic data to successfully train semantic segmentation algorithms, and create semantic segmentation masks for applications like style mixing Fig. 1.

In recent years, a number of works [1, 4, 6, 19, 21, 24, 28] have emerged to address the problem of semantic segmentation with synthetic images. The difference of this solution, compared to a classical semantic segmentation methods on photos, is that we can take advantage of the rich semantic structured in models like StyleGAN2. This, combined with cheap photo-realistic image synthesis at scale, provides the possibility to synthesize large training sets with their semantic masks to train semantic segmentation algorithms at low cost while attaining better or state-of-the-art results [4].

2. Related Works

The works most relevant to this study include supervised and unsupervised method that do semantic segmentation on synthetic images (fine-grained semantic masks and/or foreground vs. background extraction).

One of the first attempts to do semantic segmentation on synthetic images is DatasetGAN [18, 28]. DatasetGAN is a few-shot fully supervised solution, where a small MLP network is trained on the activation of a StyleGAN synthesis networks to regress a fully annotated fine-grained segmentation mask. DatasetGAN still requires pixel-wise human annotations though. A number of efforts have been made to remove this human annotation requirement, e.g., Labels4Free [1] and FurryGAN [4]. Both of these approaches use an independent masking network that's trained unsupervisedly to discriminate foreground vs background. Unfortunately, extensions to a full, fine-grained semantic map is not available and unclear how to achieve it. A potential solution is to use unsupervised clustering on a CLIP-based map [21] but this leads to inaccurate segmentations.

Another characteristic of existing methods is that the algorithms operate on intermediate features of the StyleGAN generators, introducing additional dependencies on the generator architectures and the trained weights. Thus, whenever the pre-trained generator is updated, either with new weights or new architectures, it is often necessary to reconfigure the masking network branch correspondingly, followed by re-training. The high dimensionality of the intermediate features also poses significant computational demand, which has to be resolved by often more costly machines, or segmentation at lower resolution.

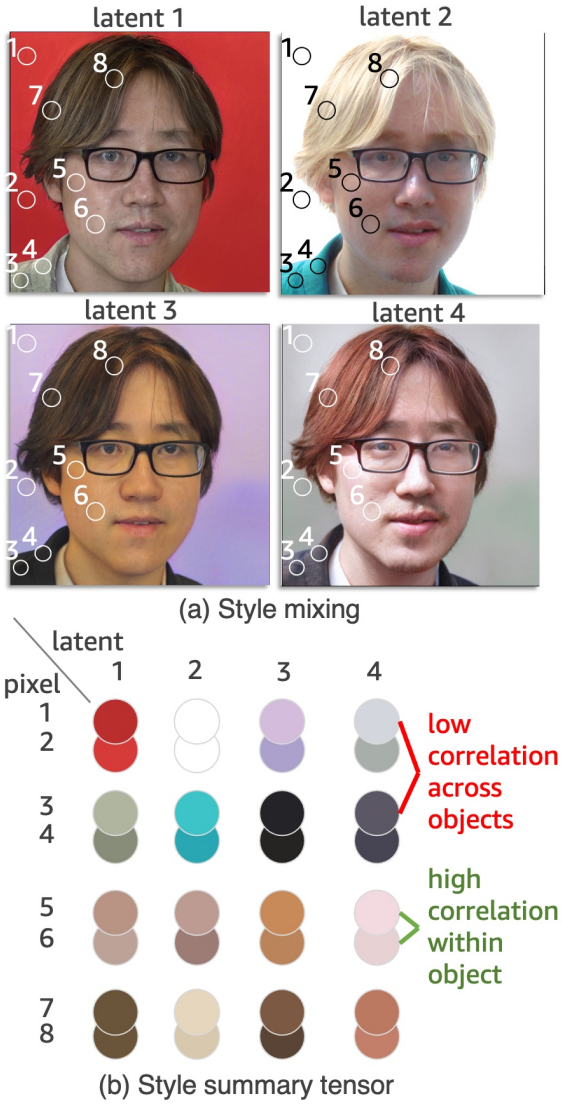

Figure 2. The key insight of our paper is to use generative model's editing techniques like style mixing in StyleGAN2 to identify image segments that co-vary vs segments that do not (mixing cutoff $c = 8$). Notice that across style-mixed images, pixels vary consistently within the same semantic segment but differently across them.

In contrast, the method we derive below achieves highly accurate semantic segmentation maps in a fully unsupervised way without the need of adding any new net/layers, re-train any components of the existing models, or the use of human annotations. As our method operate on raw pixels, it also gives flexibility and adaptability to new models,

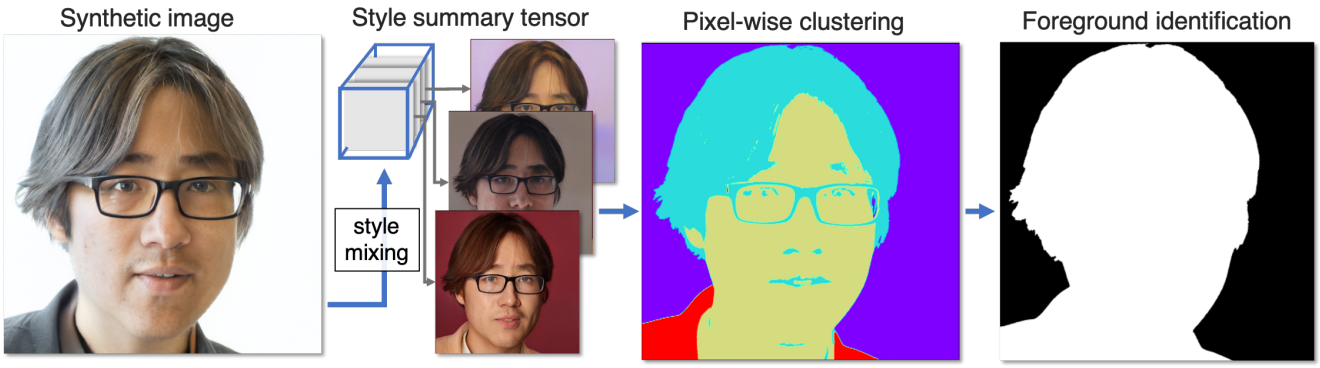

Figure 3. Overview of our method. Given a photo (and its synthetic version obtained with GAN inversion) or a synthetic image generated by StyleGAN2, we first construct a style summary tensor by concatenating style-mixed images. Unsupervised pixel-wise clustering is then applied on the style summary tensor, yielding semantic specific masks. These can be further combined to create the desirable semantic segment.

reducing computational and operational cost.

3. Method

This section provides detailed derivation of the proposed algorithm, Fig. 2 and Fig. 3.

3.1. Prelimiaries on style mixing

Style mixing is a technique first proposed in StyleGAN [15] as a regularization during training, but was later adopted as a method for synthesizing synthetic image variants, Fig. 2(a).

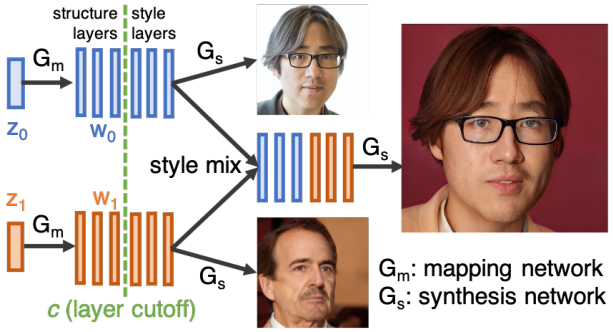

Figure 4. Style mixing process of StyleGAN2.

More formally, a StyleGAN generator $G(\cdot) = G_s(G_m(\cdot))$ is composed by two sub-networks G_m the mapping network and G_s the synthesis network. G_m maps from input latent $\mathbf{z}$ to intermediate latent $\mathbf{w} \in \mathbb{R}^{d \times l}$, where d and l are the latent dimensions and number of modulated layers in G_s respectively. G_s then maps $\mathbf{w}$ to the image space $\mathbf{X} \in \mathbb{R}^{w \times h \times 3}$.

As shown in Fig. 4, style mixing operates on two latent codes $\mathbf{z}_0, \mathbf{z}_1$ for a trained generator. Given a layer cutoff $c \in \{0, ..., l\}$, a new code $\mathbf{w}_{01}$ is generated by concatenating $\mathbf{w}_0$ before layer c and $\mathbf{w}_1$ after layer c. The style-mixed image is then given by $\mathbf{X}_{01} = G_s(\mathbf{w}_{01})$.

This process is called style mixing, as the image $\mathbf{X}_{01}$ is a mix of $\mathbf{X}_0 = G_s(\mathbf{w}_0)$ and $\mathbf{X}_1 = G_s(\mathbf{w}_1)$. The level of combination depends on the cutoff c. From 0 to l, the style-mixed image $\mathbf{X}_{01}$ will change from $\mathbf{X}_1$ to $\mathbf{X}_0$. As illustrated in Fig. 5, when c increases, $\mathbf{X}_{01}$ becomes closer to $\mathbf{X}_0$ with the mixing happens in the orders of [high-level (pose, identity, etc.)] $\rightarrow$ [low-level (texture, color style, color shift, etc.)].

In the rest of the paper, $\mathbf{w}_0$ will be referred as the structure latent, and $\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_n$ the style latent(s), $\mathbf{X}_{01}, \mathbf{X}_{02}, ..., \mathbf{X}_{0n}$ style-mixed images.

3.2. Semantic clustering through Style Mixing

Given a synthetic image $\mathbf{X}_0 \in \mathbf{R}^{w \times h \times 3}$ generated by its latent code $\mathbf{z}_0$ for a trained StyleGAN generator and a content of interest o, we wish to find a binary mask $\mathbf{Y}_0 \in \mathbf{R}^{w \times h}$ for o such that each element y_{ij} follows:

$$y_{ij} = \begin{cases} 1 & \text{if pixel } [i, j] \text{ belongs to } o, \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

We will first address when o is the image foreground and extending to object-wise masking in Sec. 3.3.

To generate high-precision object mask without training, the key question is which pixels in $\mathbf{X}_0$ belongs to the same semantic segment. We note that this can be readily achieved by leveraging the properties of style mixing in StyleGAN generators.

How can style mixing help cluster pixels by objects? We observe that with a properly selected c, style-mixed images generally maintain all the semantic structure of the original image, Fig. 5. This allows pixel-wise color correlation

Figure 5. Changing c changes the level of style mixing.

across different style-mixed images serves as an surrogate of their semantic categories.

Specifically, for a given query image $\mathbf{X}_0$, n style-mixed images $\mathbf{X}_1$,...,$\mathbf{X}_n$ are generated. First, we construct a *style summary tensor* $\mathbf{X}_s \in \mathbf{R}^{w\times h\times 3n}$ by concatenating style-mixed images at each pixel location, Fig. 3 columns 1-2. Second, K-means clustering is applied on $\mathbf{X}_s$ across pixels with $3n$-dimension style summary features. That is,

$$\mathbf{Y}' = \text{kmeans}(\mathbf{X}_s, k), \tag{2}$$

where k is the number of clusters and $\mathbf{Y}' \in [1, ..., k]^{w\times h}$ is the $w \times h$ cluster assignment map of $\mathbf{X}_0$ whose elements $\in \{1, ..., k\}$. This creates pixel sets where within-cluster pixels change their color similarly, and across-cluster pixels display much wider, random changes, Fig. 2(b). Hence, $\mathbf{Y}'$ is a surrogate of the desirable semantic segmentation map of $\mathbf{X}_0$, Fig. 3 column 3.

At this stage, the cluster map $\mathbf{Y}'$ is intermediate as we do not know which clusters correspond to the content of interest o. Thus, foreground identification has to be performed to map $\mathbf{Y}'$ to $\mathbf{Y}$. Many algorithms can be used to decide the foreground cluster. In this paper, we give examples on two methods: 1. corner minority, 2. saliency.

Corner minority approach. For a given bounding box around an object and the intermediate mask $\mathbf{Y}'$, we can generally assume that the object is locate at the center of the bounding box and the four corners are mostly occupied by background pixels. Thus, we examine a $b \times b$ area in the 4 corners of $\mathbf{Y}'$. For a pre-defined threshold θ_{corner}, if a cluster occupies greater or equal to round$(\theta_{\text{corner}}\, b^2)$ pixels, then it is a background cluster. We iterate through all k clusters. Pixels within background clusters are assigned to 0, otherwise 1, yielding the final binary mask $\mathbf{Y}$.

This method is simple, but particularly effective against rigid, convex objects without substantial shape variation across images.

Saliency approach. An alternative foreground detection algorithm utilizes a saliency map $\mathbf{S} \in \mathbb{R}^{w\times h}$. Each element $s_{ij} \in [0, 1]$ in $\mathbf{S}$ approximates how likely it is for a pixel to belong to o in $\mathbf{Y}'$. Given a predefined threshold θ_{saliency} and cluster index $m \in \{1, ..., k\}$, the foreground clusters can be identified by examining the average saliency for all the pixels within the cluster m. Specifically, cluster m belongs to o if,

$$\frac{1}{N}\sum_{i,j} \mathbf{S}(i, j) > \theta_{\text{saliency}}, \text{for all } i, j \in \{\mathbf{Y}' = m\}, \tag{3}$$

where N is the number of pixels belonging to cluster m.

For most single convex objects, a pre-defined Gaussian heat map peaked at the center of the image is sufficiently good as the saliency map, which is what we use in our experiments for human and animal faces.

3.3. Object and instance segmentation

In this section we extend our method to the object level. This is crucial for complex scenes. In scenes the foreground is not always consistently defined by a type of object, nor is its appearance and alignment [12]. This complexity poses special challenges for existing unsupervised segmentation algorithms on synthetic images, as most of them rely on a stable foreground-background decomposition.

To extend our method to object- or even instance-level, one only needs to apply the aforementioned algorithm to the pixels within the bounding box of the object/instance ($\mathbf{X}_o$ is an image crop instead of full image). With the current significant progress on pre-trained object detector [10, 25] and zero-shot object detector [26], one can obtain high-performance model on a wide range of object categories. In this study, we use GLIP [26] for its accuracy on zero-shot object detection.

3.4. Assumptions and limitations

Not relying on an additional network branch to perform the task of segmentation makes our method clearer in its assumption and limitation, as well as higher interpretability when the program fails. Here, we provide an analysis of our algorithm to provide initial applicability assessment and troubleshooting directions.

Our method currently uses images from StyleGAN2 models. As the nature of our algorithm relies on the property of style mixing, directly applying the method on synthetic images generated by other architectures that significantly different from StyleGAN is not straightforward. This

is specially true for generative models where style mixing (or variants) cannot be used. This can be resolved indirectly by using a StyleGAN model of the same domain with an accurate GAN inversion algorithm like [3, 11, 22]. On the other hand, one should notice that our method relies on the style mixing property on a set of images. StyleGAN is currently a straightforward source of such images. With the recent development such as ControlNet [27], one could generalize our method to diffusion models as well.

Our foreground modeling only works when the corner background assumption is met or the saliency map can approximately reflect the actual content of interest. When the interested object violate those assumptions, one will find our foreground identification fails and a custom foreground heuristic has to be used. We have not seen this in any of our experimental results but one can always compile a scene where these assumptions are violated. We will provide more analysis on the failure mode of our method in Sec. 5.

4. Experiments

We report the results of our approach in three different applications and provide comparative results against state-of-the-art methods.

4.1. Implementation details

Due to its simplicity and memory efficiency, we run our algorithm on the native resolution of the pre-trained StyleGAN2 generators, that is 1024×1024 pixels for faces and scenes, 512×512 for horses and the face of the other animals. In all experiments, we set $n = 50$ for the style summary tensor described above. For facial images, the number of clusters k in K-means is set to 3 for foreground segmentation and 8 for eyes and mouth segmentation, both using the saliency approach. For horses, animal faces and scenes, k is set to 2 and we use the corner approach.

We use the K-means implementation faiss [13]. Each image takes less than 1 second to process, taking 4GBs of GPU memory on a single A100.

4.2. Synthetic Image Segmentation

We test our algorithm's accuracy at extracting object masks on the FFHQ [14], LSUN-Horses, AFHQ [8], and DeepRooms [12] datasets.

Since synthetic images do not have golden ground truth segmentation from humans, we use off-the-shelf state-of-the-art semantic segmentation algorithm as pseudo-ground truth, which is similar to the testing process used in prior art like Labels4Free [1]. For experiments on FFHQ and LSUN-horses, we use the DeepLabV3+ [7] model trained on the augmented PASCAL-VOC12 dataset. For DeepRooms, we used SwinTransformer [20] pre-trained on the ADE20K dataset [29]. The foreground class is obtained by choosing the mask from the appropriate semantic classes *sofa, table* in our experiments. All the trained models are taken from [9]. For LSUN-horses and DeepRoom, where multiple foreground objects might appears in the images, we first apply GLIP zero-shot object detector [26], then use our algorithm within the detected bounding boxes, as described in Sec. 3.3. We use *"horses"* and *"sofa, coffee table, lamp, side table, rug, in a livingroom"* as GLIP text caption for LSUN-horse and DeepRoom dataset images respectively.

We report comparative results using mIOU (mean Intersection Over Union). For foreground segmentation, we report foreground IOU, background IOU and their average as mIOU. For object-wise semantic segmentation, we report mIOU over object categories and individual object IOUs.

We compare the synthetic semantic segmentation performance of our algorithm with DatasetGAN [28], Labels4Free(L4F) [1], and Semantic In Style (SiS) [21]. For DatasetGAN, we train the model on *stylegan2-ffhq-config-f* generator with the authors' provided annotations on 16 facial images with the official optimization-based inversion algorithm in StyleGAN2 at 512×512 resolution. For both L4F and SiS, we re-train the model at 1024×1024 for a fair comparison to ours. L4F requires around 10k synthetic images to train its Alpha Network while SiS requires 50 images for clustering and 15k images for training its masking branch, Tab. 1.

As shown in Tab. 1 and Tab. 2, our method outperforms the supervised baseline DatasetGAN on synthetic human faces in terms of mIOU. Comparing to the state-of-the-art unsupervised foreground segmentation algorithm, even if with no training data is used to extract across sample information, we are able to achieve similar or better performance.

4.3. Semantic segmentation of real photos

We compare our results against the golden ground-truth given by human annotations. We perform real photo semantic segmentation on the CelebAHQ-Mask [17] dataset. To use our method and other synthetic segmentation algorithms on photos, we use ReStyle encoder [2] to first perform GAN inversion and then apply our algorithm to compute the semantic segmentation mask. Similar to our previous experiments, mIOU is used as the main evaluation metric, see Tab. 1 CelebAMask-HQ columns.

4.4. Synthetic image and segmentation masks as training data

As mentioned above, an important application of synthetic semantic segmentation is to generate training data for semantic segmentation algorithms. We use a similar evaluation framework as in [21, 28]. For this experiment, we

Methods	Training data	supervision	additional network	FFHQ IOU (fg/bg)	FFHQ mIOU	CelabAHQ-Mask IOU (fg/bg)	CelabAHQ-Mask mIOU
DatasetGAN [28]	16	✓	✓	0.83/0.73	0.78	0.87/0.73	0.80
L4F [1]	10k	×	✓	0.92/0.85	**0.88**	0.92/0.80	0.86
SiS [21]	50+15k	×	✓	0.89/0.77	0.83	0.92/0.81	**0.87**
Ours	0	×	×	0.87/0.73	0.80	0.91/0.81	0.86

Table 1. Image segmentation performance on FFHQ (*i.e.*, on synthetic data) and CelebA-Mask-HQ (*i.e.*, on real data). IOU (fg/bg) is the IOU for foreground/background segmentation. mIOU is the average between the IOU (fg) and IOU (bg).

Methods	LSUN-Horse IOU (horse-fg/bg)	LSUN-Horse mIOU	DeepRoom-livingroom IOU (sofa-fg/bg)	DeepRoom-livingroom mIOU	DeepRoom-livingroom IOU (table-fg/bg)	DeepRoom-livingroom mIOU
L4F [1]	0.51/0.73	0.62	×	×	×	×
SiS [21]	0.44/0.78	0.61	×	×	×	×
Ours	0.64/0.89	**0.77**	**0.88/0.97**	**0.93**	**0.14/0.96**	**0.55**

Table 2. Semantic segmentation performance on LSUN-horses, and DeepRoom-livingroom datasets, all with synthetic images and DeepLabV3 as psuedo ground-truth. ×: method not easily extendable to segment the target class.

Methods	# manual gt	IOU fg	IOU bg	mIOU fg/bg	Trimap IOU fg	Trimap IOU bg	Trimap mIOU fg/bg
U-net [23]	1000	0.95	0.87	**0.91**	0.53	0.45	**0.49**
w/ DatasetGAN [28]	16	0.90	0.79	0.84	0.43	0.39	0.41
w/ L4F [1]	0	0.92	0.82	0.87	0.43	0.38	0.41
w/ SiS [21]	0	0.92	0.80	0.86	0.45	0.33	0.39
w/ Ours	0	0.92	0.82	**0.87**	0.42	0.43	**0.42**

Table 3. Using synthetic data as training data for image segmentation. Trained on images generated from FFHQ model, test on CelebA-Mask-HQ (real data). The supervised segmentation method is DeepLabV3. All synthetic data performances are trained from scratch using synthetic data only. Trimap width is 3 pixels.

generate a synthetic segmentation dataset for faces using the FFHQ generator [15]. Using the image and pixel-wise labels we train a U-Net [23] from scratch, for 40K iterations, to evaluate on photos from the test partition of the CelebA-Mask-HQ dataset. Models are trained using the public codebase from [9]. In addition to the standard mIOU computed using entire images, we also report mIOU computed on a Trimap of width 3 pixels following [16]. Such a metric focuses on performance along the boundary pixels. The more precise the boundary is, the higher the Trimap mIOU. We report our results in Tab. 3.

4.5. Qualitative results

In this section, we provide qualitative results of our segmentation. Fig. 6 shows facial foreground segmentation in fine details. Fig. 8 shows alpha composition between the original images and distinct style-mixed images with our masks as alpha channel. We directly use the hard binary mask without any feathering or Gaussian blur. These results show the quality of our masks, since an inaccurate mask leads to obvious artifact in the image composition that can be readily detected by humans. We provide these visualizations on FFHQ, AFHQ-wild, and DeepRooms in Fig. 8.

c	RGB k=2	RGB k=4	RGB k=10	LAB k=2	LAB k=4	LAB k=10
4	0.66	0.73	0.67	0.70	0.70	0.66
6	0.74	0.75	0.68	0.74	0.79	0.69
8	0.76	0.76	0.67	0.76	0.78	0.69
12	0.44	0.58	0.68	0.45	0.61	0.69
14	0.45	0.59	0.66	0.51	0.60	0.62

Table 4. Ablation study on 500 randomly selected FFHQ images, measured in mIOU.

4.6. Ablation Study

To test the effect of the parameters in our algorithm, we perform a series of ablation studies. These correspond to the layer cutoff c, the color space of the style summary tensor, and the number of clusters k.

Effect of cutoff c.

As described in Sec. 3.1, the higher the c, the closer the style-mixed image is to the original $\mathbf{X}_0$. Thus, if c is too

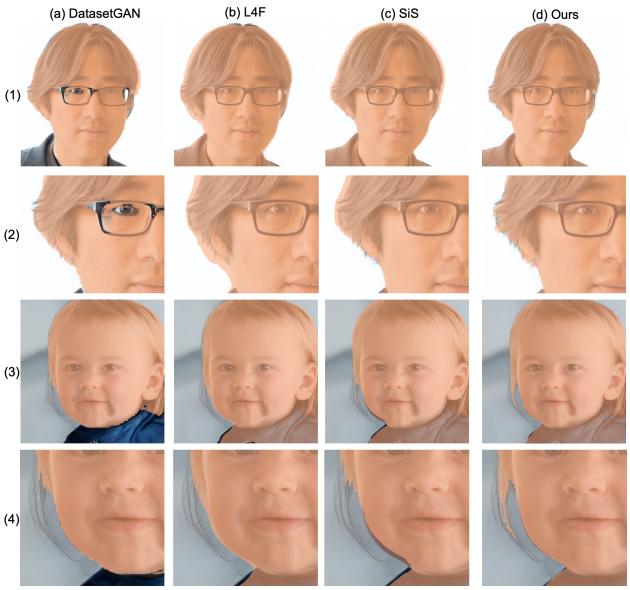

Figure 6. Segmentation details of people from CelebAMask-HQ photos and FFHQ synthetic images. Details zoomed in to compare the segmentation precision.

high, the style summary tensor might lack the diversity and style disentanglement between objects needed to yield accurate results. If c is too low, then the tensor might contain images of very different structures, losing their spatial and semantic correspondences. In these cases, the performance of clustering will be negatively affected. In Tab. 4, we compare segmentation performance with c ranging from 4 to 12. The mIOU is generally the worst when $c = 4$ and $c = 12$ while holding k constant.

Color space

The choice of color space also affects the performance of the clustering algorithm. We observe that when constructing the style summary tensor in RGB space, the K-means clustering might be overly focused on the highlight of the objects, especially when reflective surfaces are present. This is because of highlights mostly vary in conjunction with scene illumination, not with semantic segments (objects). Changing the color space to LAB addresses this problem, as we show in Tab. 4.

Number of clusters

The number of clusters k is another parameter of importance. Generally, when the input $\mathbf{X}_0$ is an image cropped from a tight bounding box, $k = 2$ yields very good semantic segmentations. However, when estimating the foreground of a main object class (e.g., a person or a cat) on an entire image, it is likely that the foreground pixels clustered with some background pixels. We wish to have a slightly larger value for k that allows for shading and textural changes in the same semantic segment. For this reson, we selected k=4. We test the choice of k from 2 to 10 on faces. Tab. 4 shows that both $k = 2$ and $k = 10$ give low performance and a better mIOU is indeed achieved when k=4.

5. Failure Cases

There are two potential modes of failure in our method: 1. clustering failure, and 2. foreground failure. The latter we already discussed in the Sec. 3.4. The former happens when a) the style mixing properties does not hold well and/or b) the main assumption of color correlation in style summary tensor does not hold.

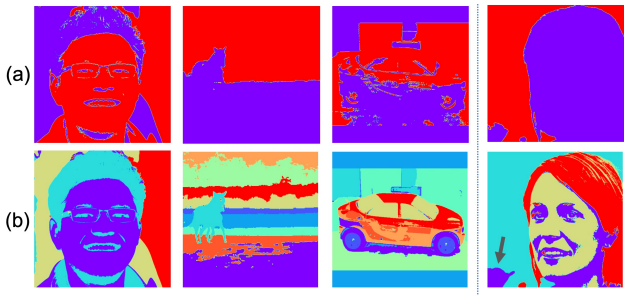

Figure 7. (a) Failure cases for our methods when k is small. (b) Clustering with properly selected k.

Fig. 7 shows examples on the clustering results where this failure happens. Specifically, (a) shows that when k is selected to be too small, significant foreground-background confusion may occur. This type of failure can be mitigated when k is properly selected as in (b). However, even with a proper k, it is possible (though rare) that areas in the background are assigned as foreground (right most column, black arrow).

6. Conclusion

Semantic segmentation is an important problem in computer vision. In the present paper, we have derived a new approach that takes advantage of the consistency of pixel correlations when editing synthetic images with techniques like style mixing in GANs. We have shown how our approach can be used to generate semantic segmentation masks of real photos, create synthetic edits of these photos, and generate synthetic training data for downstream tasks. Comparative results with state-of-the-art algorithms show that the proposed approach yields as or more accurate results than those reported in prior art with the added advantages that our approach does 1. not require adding layers or networks to existing pre-trained generative models, 2. not need to be fine-tuned to each application, and 3. not require any supervision or labelled training data.

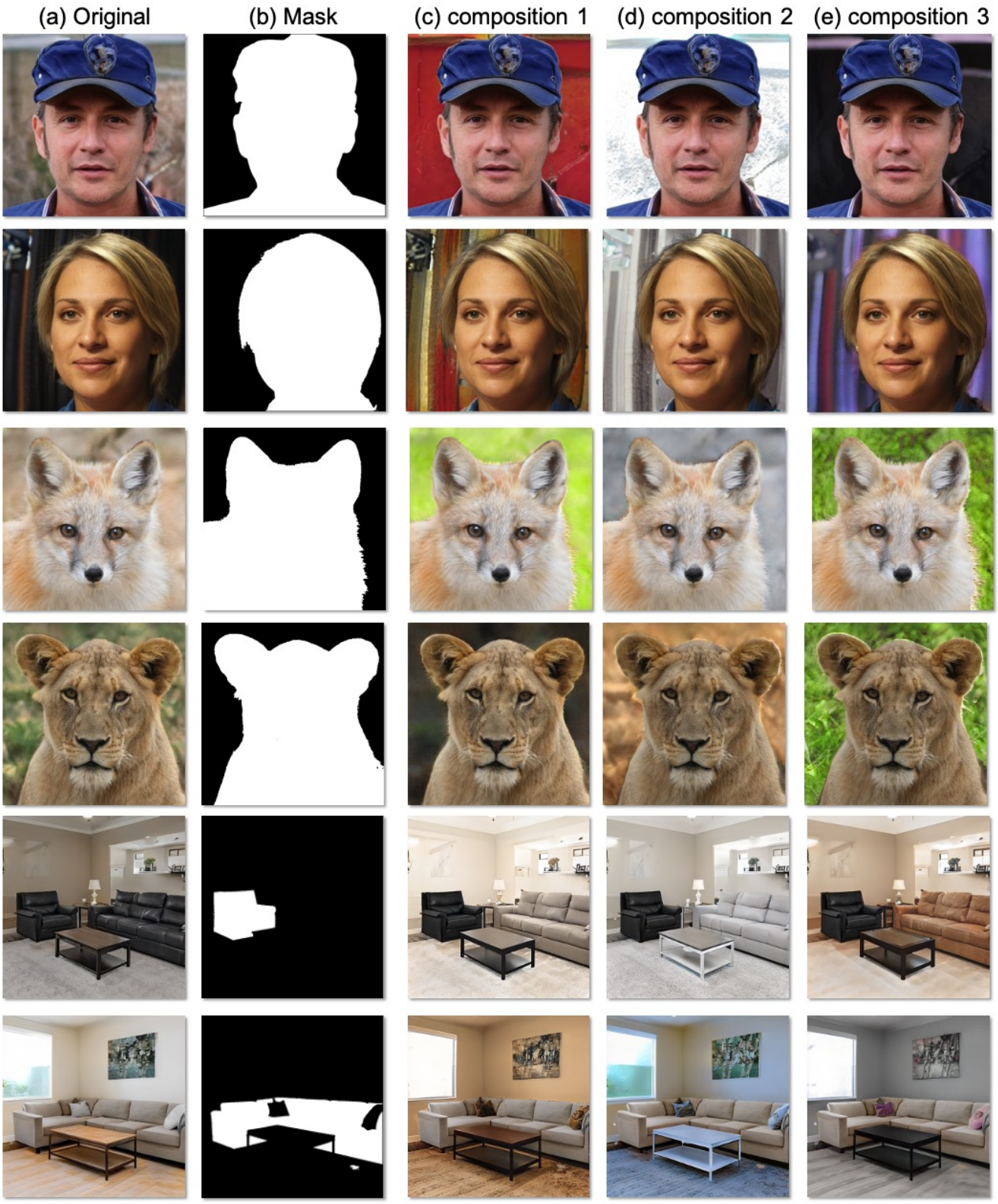

Figure 8. Image composition with our masks on FFHQ, AFHQ-wild and DeepRoom images, blended using hard binary masks with no feathering.

References

[1] Rameen Abdal, Peihao Zhu, Niloy J. Mitra, and Peter Wonka. Labels4free: Unsupervised segmentation using stylegan. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13970–13979, October 2021. 1, 2, 5, 6

[2] Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or. Restyle: A residual-based stylegan encoder via iterative refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021. 5

[3] Yuval Alaluf, Omer Tov, Ron Mokady, Rinon Gal, and Amit H. Bermano. Hyperstyle: Stylegan inversion with hypernetworks for real image editing, 2021. 5

[4] Jeongmin Bae, Mingi Kwon, and Youngjung Uh. Furrygan: High quality foreground-aware image synthesis, 2022. 2

[5] Amin Banitalebi-Dehkordi and Yong Zhang. Repaint: Improving the generalization of down-stream visual tasks by generating multiple instances of training examples. *arXiv preprint arXiv:2110.10366*, 2021. 1

[6] Adam Bielski and Paolo Favaro. *Emergence of Object Segmentation in Perturbed Generative Models*. Curran Associates Inc., Red Hook, NY, USA, 2019. 1, 2

[7] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018. 5

[8] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8188–8197, 2020. 5

[9] MMSegmentation Contributors. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. https://github.com/open-mmlab/mmsegmentation, 2020. 5, 6

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021. 4

[11] Qianli Feng, Viraj Shah, Raghudeep Gadde, Pietro Perona, and Aleix Martinez. Near perfect gan inversion. *arXiv preprint arXiv:2202.11833*, 2022. 1, 5

[12] R. Gadde, Q. Feng, and A. M. Martinez. Detail me more: Improving gan's photo-realism of complex scenes. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13930–13939, Los Alamitos, CA, USA, oct 2021. IEEE Computer Society. 1, 4, 5

[13] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019. 5

[14] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *CoRR*, abs/1812.04948, 2018. 2, 5

[15] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks, 2018. 3, 6

[16] Pushmeet Kohli, Lubor Ladicky, and Philip H. S. Torr. Robust higher order potentials for enforcing label consistency. *Int. J. Comput. Vis.*, 82(3):302–324, 2009. 6

[17] 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. 5

[18] Daiqing Li, Huan Ling, Seung Wook Kim, Karsten Kreis, Adela Barriuso, Sanja Fidler, and Antonio Torralba. Bigdatasetgan: Synthesizing imagenet with pixel-wise annotations. 2022. 1, 2

[19] Huan Ling, Karsten Kreis, Daiqing Li, Seung Wook Kim, Antonio Torralba, and Sanja Fidler. Editgan: High-precision semantic image editing. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 1, 2

[20] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 5

[21] Daniil Pakhomov, Sanchit Hira, Narayani Wagle, Kemar E Green, and Nassir Navab. Segmentation in style: Unsupervised semantic image segmentation with stylegan and clip. *arXiv preprint arXiv:2107.12518*, 2021. 2, 5, 6

[22] Daniel Roich, Ron Mokady, Amit H Bermano, and Daniel Cohen-Or. Pivotal tuning for latent-based editing of real images. *ACM Trans. Graph.*, 2021. 5

[23] 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. 6

[24] Andrey Voynov, Stanislav Morozov, and Artem Babenko. Object segmentation without labels with large-scale generative models. In *International Conference on Machine Learning*, pages 10596–10606. PMLR, 2021. 2

[25] Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. Lit: Zero-shot transfer with locked-image text tuning. *CVPR*, 2022. 4

[26] Haotian* Zhang, Pengchuan* Zhang, Xiaowei Hu, Yen-Chun Chen, Liunian Harold Li, Xiyang Dai, Lijuan Wang, Lu Yuan, Jenq-Neng Hwang, and Jianfeng Gao. Glipv2: Unifying localization and vision-language understanding. *arXiv preprint arXiv:2206.05836*, 2022. 4, 5

[27] Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023. 5

[28] 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 *CVPR*, 2021. 1, 2, 5, 6

[29] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 5