Spatially-Adaptive Normalization for Human Face Synthesis

Yunbai Zhang: yz3386@columbia.edu
Shiyuan Li: sl4398@columbia.edu
Jiao Zhou: jz3071@columbia.edu

Abstract

The new conditional normalization method called Spatially-Adaptive Normalization(SPADE) has been proved as a good fit for landscape, cityscape image synthesis. Since the three datasets the author used did not involve that of "human face", we suspect that SPADE normalization is not suitable for this kind of dataset. Face synthesis is also an important area in artificial intelligence, so our task is to improve the SPADE model to make it applicable to human face dateset. The first step we do is to do image segmentation. We implement the model by CelebAMask-HQ on our dataset to identify the face location and attributes. Then we use the segmented images as input to the SPADE model and do conditional normalization.

1. Introduction

Conditional image synthesis is a method that generating images depends on input data. Recently, some researchers use neural network to normalize without requiring external data. During this process, the learned affine parameters are used to control the global style of the output and so are uniformly across the spatial coordinates. [1] As a result, this may cause semantic information washing away. Compare to the unconditional normalization method, the parameters for SPADE have already "gained" enough information about the label layout, it does not need to feed the segmentation map to the first layer. Therefore, it would maintain more semantic information than transitional normalization layers, and so it would better fit tasks of image synthesis.

2. Problem formulation and goal

One goal of our project is to use generative adversarial networks (GANs) which aims for synthesizing human face images. Suppose the condition that if we only have one segmentation mask with a single label as input, the convolution outputs will be uniform with different labels having different uniform values.[1] The normalized activation will become all zeros, and the feature of human faces will largely lose. Thus, We plan to introduce SPADE method which can maintain semantic face feature information. In addition, we learned that the traditional omission of noise leads to featureless "painterly" look. We tried to avoid it on our generator.

3. Methods

3.1 Method Description Suppose that $\mathbf{m} \in L^{H \times W}$ to be a semantic mask, where L denotes semantic labels, and H and W are the image height and width. Our task is to convert an input segmentation mask m to a realistic image. [1] Let H^i and W^i denote the height and width of the activation map. The method of computation for SPADE is similar to that of Batch Normalization, in which the activation is normalized in channel pairwisely. And the activation value

$$\gamma^i_{c,y,x}(\mathbf{m})\frac{h^i_{n,c,y,x}-\mu^i_c}{\sigma^i_c} + \beta^i_{c,y,x}(\mathbf{m})$$

,

where the mean and variance in the channel c is given by $\mu^i_c = \frac{1}{NH^iW^i}\sum_{n,y,x} h^i_{n,c,y,x}$

and $\sigma^i_c = \sqrt{\frac{1}{NH^iW^i}\sum_{n,y,x}\left(h^i_{n,c,y,x}\right)^2 - \left(\mu^i_c\right)^2}$ [1]

Different from Batch-norm, the method SPADE depends on the input segmentation mask and the location.

3.2 Data Description and preprocessing We used a new dataset of human faces, Flickr-Faces-HQ (FFHQ), consisting of 70,000 high-quality images at 10242 resolution. The dataset includes vastly more variation than CELEBA-HQ in terms of age, ethnicity and image background, and also has much better coverage of accessories such as eyeglasses, sunglasses, hats, etc [2]. We split the whole dataset into two parts, 20% as test data, 80 % as training data. Due to limited time for training, we randomly sub-sample 10000 images as input

3.3 Image Segmentation In order to implement SPADE method, we need to use the segmented pictures to do image synthesis. So, we firstly need to transform our original images to segmented ones. We use $pix2pix$ with $cycle-GAN$ model to translate the original image to a segmentation map. There are 19 classes for our segmentation model, and the classes table is shown on Figure 5. The accuracy for this model is 93.41 percent. [3]. For details, after training this

Figure 1: Segmentation of hair

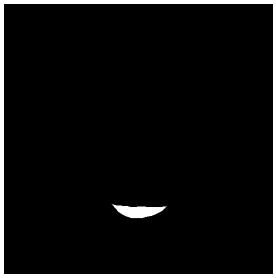

Figure 2: Segmentation of mouth

Figure 3: Original image

Figure 4: Segmentation map

model, we label each part of face as Figure 1 and Figure 2 show. And then we synthesis each part to a segmentation image with different colors. Figure 3 and Figure 4 show one example of this transformation.

3.4 Training Process

Generator After the linear mapping, we did some data pre-processing, which

Unlike the state of art generator method pix2pixHD which includes both down-sampling layers and up-sampling layers, our SPADE generator can achieve a better result by removing the down-sampling layers. We start by using a random vector as our input. Then we use an encoder to process a real image into this vector, which then is fed to our SPADE generator. The encoder and generator form a variational autoencoder. This encoder has the ability to capture the style of the image. Then the generator will use the encoded style and the segmentation mask through the SPADE to create an image. Finally, we plan to add some random noises and novel mixing regularization to make neighboring styles less correlated and our image more realistic. When we train the model, we simply use two layers of convolutional neural network for each residual block Figure 6 shows how we use the $SPADE$ method in training our model. When we run the code of the $SPADE$ method to our segmented face images, we firstly try the learning rates for the generator and discriminator are set to 0.0001 and 0.0004, respectively, as shown in the paper. We get a result shown in Figure 7 and Figure 8. After we get the result, we try different learning rates for the generator and discriminator shown in table 1.

Label list		
0: 'background'	1: 'skin'	2: 'nose'
3: 'eye_g'	4: 'l_eye'	5: 'r_eye'
6: 'l_brow'	7: 'r_brow'	8: 'l_ear'
9: 'r_ear'	10: 'mouth'	11: 'u_lip'
12: 'l_lip'	13: 'hair'	14: 'hat'
15: 'ear_r'	16: 'neck_l'	17: 'neck'
18: 'cloth'		

Figure 5: Label List of segmentation map

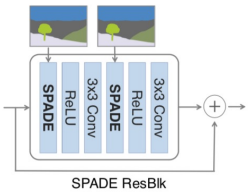

Figure 6: A residual block with SPADE method [1]

Generator	Discriminator
0.0004	0.0006
0.0008	0.0008
0.001	0.0012

Table 1: Different learning rate for generator and discriminator

Performance Criteria. We use mean Intersection-over-Union(mIoU) to measure the segmentation accuracy, and we use Frechet Inception Distance(FID) to measure distance between the distributions of synthesized results and the distribution of real images.[1] Here is the formula for mIoU criteria: $IoU = \frac{target \cap predictor}{target \cup predictor}$ and $FID = \|\mu_r - \mu_g\|^2 + Tr\left(\Sigma_r + \Sigma_g - 2\left(\Sigma_r \Sigma_g\right)^{1/2}\right)$, where lower FID score corresponds to higher similarity between the ground truth and the predictor.

5.Result

Figure 7: Original image

Figure 8: Segmented image

Figure 9: Synthesized image

In the process of segmentation, we use the pre-trained model provided by CelebAMask-HQ. Then we feed the segmented images as input, use SPADE normalization and generate the synthesized output. During the training process, we use 1000 segmented images as input and set our training epoch to be 10 with 1000 iterations per epoch.

6. References

[1] Taesung Park, Ming-Yu Liu, Ting-Chun, Wangun,Yan Zhu; Semantic Image Synthesis with Spatially-Adaptive Normalization. *IEEE: Computer Vision and Pattern Recognition*. 2019

[2] Karras, T., Laine, S. Aila, T. A Style-Based Generator Architecture for Generative Adversarial Networks. *Neural and Evolutionary Computing (cs.NE)*. 2018

[3] Ziwei Liu, Ping Luo, Xiaogang Wang and Xiaoou Tang; "Deep Learning Face Attributes in the Wild" *IEEE International Conference on Computer Vision (ICCV)*. 2015