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 to be a good fit for landscape and cityscape image synthesis. Since neither of the the three datasets the authors used involved "human faces", we suspect that vanilla SPADE normalization is not suitable to work with "human faces". 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.

1.1 Unconditional Normalization layers There are some popular unconditional normalization methods, such as Batch Normalization and Instance Normalization. Let β to be a mini-batch with size m. For each activation function, we will normalize each scalar feature for each dimension.[5] Differently from Batch Normalization, that is normalizing all images across the batch and spatial locations, Instance Normalization normalizes each batch independently [6]

1.2 Conditional Normalization layers Intuitively, different from unconditional normalization layers, the conditional one depends also on external data. Conditional instance normalization only acts on the scaling and shifting parameters, for the task of image synthesis, it need fewer parameters than unconditional normalization model. [7]

1.3 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.

2. Methods

2.1 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 [1]

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

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

Figure 1: Segmentation of hair (left) and Segmentation of mouth (right)

Figure 2: Original image (left) and Segmentation image (right)

2.2 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.

2.3 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 $2D$ convolutional network to translate the original image to a segmentation map. For each block, we firstly use Batch Normalization followed by $Relu$ function and a convolutional layer. We have total 3 blocks. 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 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.

2.4 Training Process

2.4.1 Generator After the linear mapping, we did some reshaping such that it could fit the SPADE ResBlock. For each layer, we use a SPADE normalization followed by a ReLU activation and 3*3 convolutional layer with k filters. We repeat this process several times to finish SPADE ResBlk construction.

2.4.2 Discriminator For the discriminator, we apply the method that is used in the pix2pixHD paper, which is a multi-scale design. We first downsample the real and synthesized high-resolution images by a factor of 2 and 4 to create an image pyramid of 3 scales.[4] Then it takes the segmentation map and the synthesized image from the generator as input. In each layer, it follows by a 4*4 convolutional layer, an instance normalization and LReLU activation function. For the last layer, corresponding to Patch-GAN, we only use convolutional layer.

2.4.3 Encoder The image encoder consists of several stride-2 convolutional layers with 3*3 filters. The last layer is two linear one, in which we can get the mean vector μ and the variance vector of σ

2.4.3 Learning Objective When training the proposed framework with the image encoder, we include a KL Divergence loss: $\mathcal{L}_{\mathrm{KLD}} = \mathcal{D}_{\mathrm{KL}}(q(\mathbf{z}|\mathbf{x})\|p(\mathbf{z}))$, where the prior distribution $p(z)$, and z is from standard normal distribution and q is determined by a mean vector and a variance vector. For this case, we weight for the KL-Divergence loss is 0.05

2.4.4 Training process summary 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 10000 segmented images as input and set our training epoch to be 2 with 1000 iterations per epoch.

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 shows how we use the SPADE method in training our model.

2.4.5 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 cor-

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 3: Label List of segmentation map

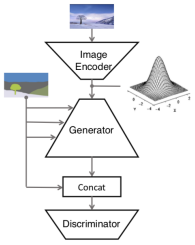

Figure 4: A residual block with SPADE method [1]

responds to higher similarity between the ground truth and the predictor.

2.4.6 Testing Process. For the testing part, we first segment the original image, feed the segmented image to the training model, and then generate a synthesized image. The second step is that we use the new synthesized image as input and implement the segmentation algorithm to generate a new segmented image. Our test algorithm is that we use the mIoU and FID as performance criteria and test whether these two segmented image are similar enough. The reason we use this algorithm is that it is hard to compare accuracy between the original one and the synthesis one, but if we use segmentation images, it is easy for us to detect the attributes and location of facial features by implementing features classification.

Here is our test algorithm architecture.

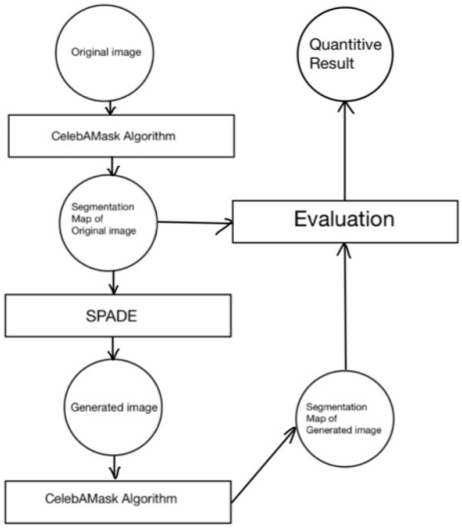

Figure 5: Testing algorithm

3.Result

Figure 6: Original image (Left), segmentation image (middle) and segmentation image generated by new synthesized image(right)

From the image above, we can see that the segmented image generated by the original one is very closed to that of generated by the synthesized one.

Accuracy	mIoU	FID
0.9139	0.7213	0.233

Compare to the model that implemented on the Cityscapes and COCO-Stuff dataset, the accuracy and mIoU for human face dateset is much higher. That means the SPADE normalization can well-implement on human face dataset.

This is our github link: https://github.com/summerdeeplearning/deep-learning

5. 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

[4] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, Bryan Catanzaro; "High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs" *Computer Vision and Pattern Recognition*. 2018

[5] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International *Conference on Machine Learning (ICML)*. 2015.

[6] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Instance nor- malization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*. 2016

[7] V. Dumoulin, J. Shlens, and M. Kudlur. A learned repre- sentation for artistic style. *International Conference on Learning Representations (ICLR)*. 2016.