Content-Consistent Generation of Realistic Eyes with Style

Marcel Bühler [1], Seonwook Park[1], Shalini De Mello[2], Xucong Zhang[1], Otmar Hilliges[1]
[1]ETH Zürich, [2]NVIDIA
{buehlmar,spark, zhangxuc, otmarh}@ethz.ch; shalinig@nvidia.com

Abstract

Accurately labeled real-world training data can be scarce, and hence recent works adapt, modify or generate images to boost target datasets. However, retaining relevant details from input data in the generated images is challenging and failure could be critical to the performance on the final task. In this work, we synthesize person-specific eye images that satisfy a given semantic segmentation mask (content), while following the style of a specified person from only a few reference images. We introduce two approaches, (a) one used to win the OpenEDS Synthetic Eye Generation Challenge at ICCV 2019, and (b) a principled approach to solving the problem involving simultaneous injection of style and content information at multiple scales. Our implementation is available at https://github.com/mcbuehler/Seg2Eye.

1. Introduction

Recent generative models are capable of synthesizing realistic images by using adversarial methods. However, realism is not the only requirement for computer vision research, as content and style can play important roles for specific tasks, such as regression tasks which require high accuracy, e.g. hand joints regression and eye gaze estimation. In this paper, we study the task of generating realistic near-eye images while preserving the content defined by a semantic segmentation mask, and style defined by a few images from a target person. We propose two methods to tackle this task. Our first method uses image refinement and is the winning solution of the OpenEDS Synthetic Eye Generation Challenge 2019[1], and our second method is a novel architecture for ensuring preservation of desired content and style. However, due to optimizing for a very specific error metric, the generated images show blurry regions. Therefore, we propose another more principled method for image synthesis that produces realistic high-quality images that still satisfy both content and style, and furthermore allows for an interpolation between styles. This method, Seg2Eye (Style and Semantic Segmentation preserving GAN), uses content-preserving spatially adaptive normalization blocks (SPADE) [16] alongside style-preserving adaptive instance normalization layers (AdaIN) [7, 8, 10], to inject both content and style information at different feature map scales. It is simple yet highly effective when applied in conjunction with a style consistency loss. In addition, style injection in Seg2Eye is performed from latent embeddings of multiple reference style images from the target person. This allows for control as well as a sampling from the learned latent space for synthesizing entirely new people (see Figure 1).

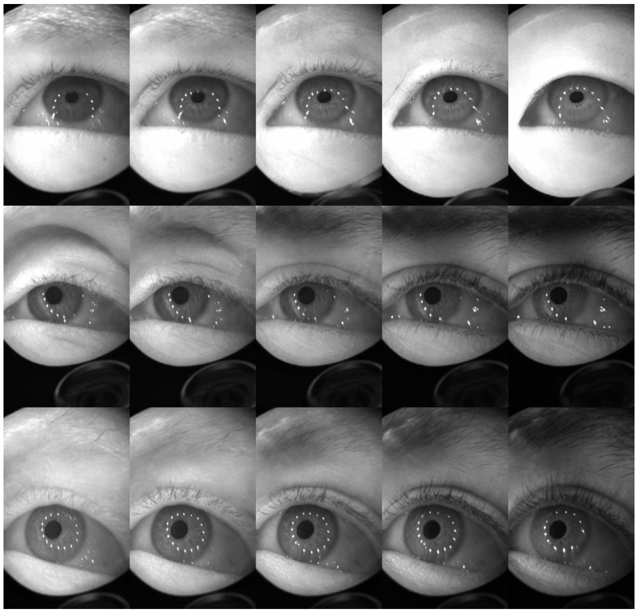

Figure 1: Walking the style latent space in our proposed method, Seg2Eye. We extract latent style codes from two people and show the decodings of their linear interpolation.

2. Related Work

Gaze Estimation. Recent works in gaze estimation modify existing synthetic or real eye images through domain randomization [15], style transfer [20, 19, 12] and gaze redirection [3, 22, 23, 6, 14] to yield data for training more ro-

[1] https://research.fb.com/programs/openeds-challenge

bust and accurate models. However, the style transfer methods can be poor in preserving content (i.e. eye shape), and re-direction methods can struggle with extrapolating from gaze directions available during training.

Image Translation. Prior art in image translation often train domain-specific models for cross-domain style transfer [9, 21], use Adaptive Instance Normalization (AdaIN) to allow real-time control of visual style [13, 7], and more recently inject style [10] or content information (via SPADE blocks [16]) at multiple feature map scales. Our proposed Seg2Eye approach combines the power of AdaIN and SPADE blocks for control of the style and content of generated eye images respectively.

3. Dataset

The provided dataset for the challenge, OpenEDS [4], is a collection of near infra-red eye images from 152 people captured by a virtual-reality headset (similar to [11]) and includes segmentation map annotations for pupil, iris and sclera regions for a subset of $12,759$ images and corneal topography data for 143 of the 152 participants. The unlabeled dataset has two subsets: a "generative" subset with $252,690$ images, and a "sequence" dataset with $91,200$ frames (from 1.5 seconds videos collected at 200Hz).

4. Method

In this section, we describe how we won the OpenEDS challenge, and our suggested approach to generating realistic eye images while preserving both content and style.

4.1. Eye Segmentation and Similarity Ranking

The OpenEDS dataset exhibits two modes, which we assume to come from left vs. right eyes. Feeding a left image in order to generate a synthetic image of the right eye impacts performance due to sources of high error (glints, black regions from the head mount) not being apparent in segmentation maps nor consistently in the unlabeled images. Hence, there is a need to carefully select images that come from the same mode. In addition, the more similar the selected unlabeled image is to the target image, the easier it is for the network to produce high-quality output.

In order to find unlabeled images that are similar to a segmentation mask, we train a DeepLab v3+ network [1, 2] to predict pseudo-labels (segmentation masks) on the unlabeled dataset and compare them to the target segmentation mask. We use mean squared error as similarity measure. The unlabeled images are then ranked by the similarity of their predicted segmentation mask with the target segmentation mask. We found the matching to perform better when coloring the segmentation masks by the mean value of the respective region across all persons in the training set.

The output of this step is a ranked list of unlabeled images of the same person for each segmentation mask. This ranking is later used to either modify a similar image to satisfy a test segmentation mask (Section 4.2), or generate a new image (Section 4.3).

4.2. Refiner Network

Learning a Residual Map. The refiner $\mathcal{R}_\Theta$ is a DeepLab v3+ network [2] with modified inputs and a different loss function. It learns a residual map from the target segmentation map M_T, a similar reference image I from the same mode and its pseudo-label M_I. The predicted residual map is added to the reference image I in order to produce the final output image $\hat{I}$. We train the refiner end-to-end with mean L2 error as the optimization objective.

$$\hat{T} = I + \mathcal{R}_\Theta(M_T, M_I, I) \tag{1}$$

4.3. Seg2Eye Network

While the refinement approach (Section 4.2) wins the challenge the images produced are not comparable in visual fidelity to the near IR eye images in the dataset (Figure 3). Furthermore, the previous approach is unable to synthesize new styles (or person identities). Hence, we propose a generative adversarial network approach, which learns latent style embeddings of unlabeled images and merges them in order to produce photo-realistic outputs.

The basis of our method is a mixture between GauGAN [16] and StyleGAN [10]. GauGAN is a recently proposed generative adversarial network (GAN) with strong segmentation mask (content) consistency, whereas StyleGAN learns a suitable latent representation of style and injects it via AdaIN [7].

Stylizing the Output. Figure 2c illustrates the information flow from the encoder to the generator. We calculate the latent style code by sampling a set of images $I^{(1)}, I^{(2)}, ..., I^{(k)}$ ($k \in \mathbb{N}$) from a specified target person and embed them via an encoder network $\mathbf{s}^{(i)} = E_{style}(I^{(i)})$. The style codes are aggregated by taking the element-wise maximum: $\mathbf{s}_i = \max_{1 \leq j \leq k} \mathbf{s}_i^{(j)}$.

This is inspired by the set-based face recognition literature [17, 18], where variations in appearance and head pose in the real world and consequent loss of information can be mitigated by merging information from multiple images of the same person. Our style encoder uses spectral instance normalization in order to preserve style information [7, 13]. The aggregated style code is used to calculate the parameters of AdaIN in the generator blocks. In this way, we can create a realistic eye image of a person with just a few unlabeled eye images and perform latent walks in style space.

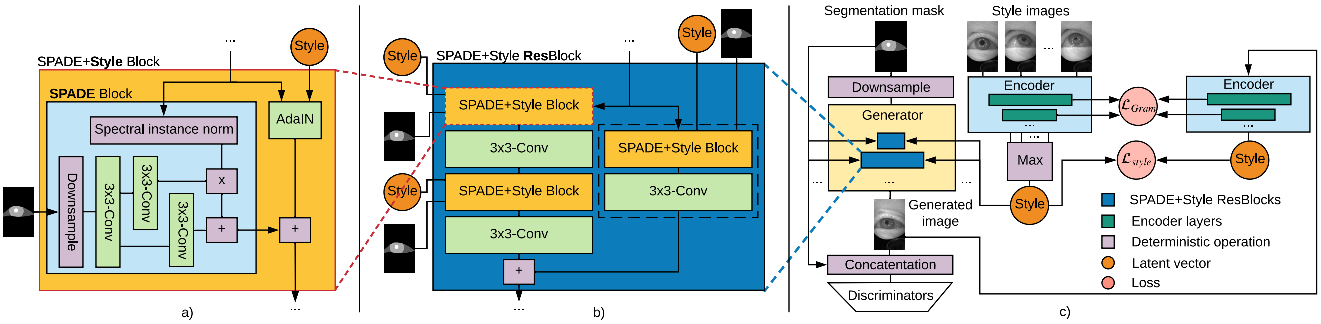

Figure 2: Seg2Eye architecture. (a) describes our novel SPADE+Style Block, which combines Adaptive Instance Normalization (AdaIN) and Spatially Adaptive Normalization (SPADE) to allow for simultaneous style and content injection into the generator at multiple scales. (b) shows how the SPADE+Style Block is used in our ResBlocks, and (c) shows the overall architecture. The generator takes a downsampled segmentation mask and a latent style code as input and produces a synthetic image. The style code is calculated from several style images. We illustrate losses for style consistency, but omit the discriminator losses for brevity.

Content Preservation. In order to produce images following the input segmentation mask, our generators repeatedly apply spatially adaptive normalization (SPADE) [16]. In comparison to batch or instance norm, where the adaptive parameters modify channels of feature maps as a whole, the SPADE layer learns a scale and offset for each element in the feature map based on the reference segmentation mask.

Generator Architecture. Our generator model is an extension of GauGAN [16], a model that applies spatially adaptive normalization (SPADE) to generate synthetic images given a semantic segmentation mask. We modify the SPADE block to inject both content and style. Our combined normalization, the SPADE+Style Block, takes three inputs: a semantic segmentation map M, a style vector $\mathbf{s}$ and the actual feature maps X. The input feature maps X take two different paths and are merged again by addition before leaving the block. One path is an AdaIN [7] layer where the parameters are computed from a style vector. The AdaIN layer applies a learned affine transform to adjust the dimensionality of the style vector to the correct number of channels. The other path is a SPADE block as described by [16], but we apply spectral instance norm instead of synchronized batch norm. The output is divided by two, which we found to stabilize the training. In short, we compute SPADE+STYLE$(X) = ($SPADE$(X) +$ AdaIN$(X))/2$.

Following GauGAN, our generator starts from a small segmentation map and then doubles feature map resolution in a series of SPADE+Style ResBlocks. Each SPADE+Style ResBlock consists of two SPADE+Style Blocks and a residual connection. Figure 2 shows the elements of a SPADE+Style (a) Block and (b) ResBlock.

Training. We train Seg2Eye on paired labeled samples from the training subset of OpenEDS, and sample style images from the top 200 images from the similarity ranking stage (Section 4.1). Further implementation information can be found in our supplementary materials.

4.3.1 Objective Function

Discriminator Losses. The adversarial loss $\mathcal{L}_{\mathcal{GAN}}$ follows GauGAN [16] and thus takes as input the concatenation of the semantic segmentation mask and the generated image. We also apply an L1 consistency loss on the discriminator feature maps. Concretely, let $F_D^{(i)}(\cdot)$ extract the feature map of the discriminator layer i and let I and $\hat{I}$ be the real and generated image. Our feature map consistency loss is computed as the sum of the intermediate feature map consistency terms. Let m be the number of feature maps in the discriminator. We compute the loss as

$$\mathcal{L}_{\mathcal{D}_\mathcal{F}} = \Sigma_{i=2}^{m} ||F_D^{(i)}(\hat{I}) - F_D^{(i)}(I)||_1 \quad (2)$$

Pixel-Matching Loss. As we have paired training samples, we apply a simple L2 consistency loss on the generated vs. target image: $\mathcal{L}_{L2}||\hat{I} - I||_2$.

Style Consistency Loss. We compute the style consistency loss by passing the generated image $\hat{I}$ through the style encoder E_{style}. Let $\mathbf{s}$ be the aggregated style vectors as described above and $\hat{\mathbf{s}} = E_{style}(\hat{I})$ be the style vector for the generated image $\hat{I}$. We compute the latent style code consistency loss as $\mathcal{L}_{style} = ||\mathbf{s} - \hat{\mathbf{s}}||_2$. In addition, we compute a consistency loss on the Gram matrix of the feature maps of the encoder.

Let $F_E^{(i)}(\cdot)$ extract the i-th feature map of the encoder E_{style}. We calculate the encoder Gram matrix consistency loss as

$$\mathcal{L}_{Gram} = \Sigma_{i=2}^{m} ||Gr(F_E^{(i)}(\hat{I})) - Gr(F_E^{(i)}(I))||_1 \quad (3)$$

where m is the number of feature maps in the encoder and $Gr(\cdot)$ is the function to compute the Gram matrix [5].

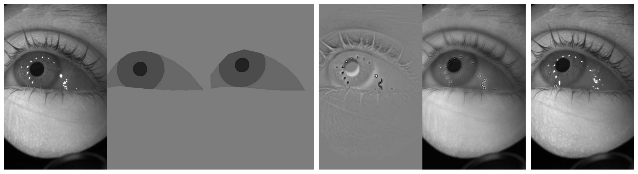

Figure 3: Qualitative outputs of the Refiner Network. The left three columns show the input reference image, pseudo-label and target label. The three columns on the right show the learned residual map, the final predicted image and the target image. We see that the modified regions are blurry due to optimizing for the L2 score.

Full Training Objective. The full training objective for the generator G (given a discriminator D) is written as:

$$\begin{aligned}\mathcal{L}_G =& \lambda_{GAN}\mathcal{L}_{GAN} + \lambda_{D_F}\mathcal{L}_{D_F} + \lambda_{L2}\mathcal{L}_{L2} \\ &+ \lambda_{style}\mathcal{L}_{style} + \lambda_{Gram}\mathcal{L}_{Gram}\end{aligned} \tag{4}$$

with $\lambda_{GAN} = \lambda_{D_{FM}} = 10$, $\lambda_{L2} = 15$, $\lambda_{style} = 0.5$ and $\lambda_{Gram} = 10^4$.

5. Results

The per-image objective function of the OpenEDS Synthetic Eye Generation Challenge[2] is given as: $\frac{1}{HW}\sqrt{\Sigma_i^H \Sigma_j^W (\hat{I}_{ij} - I_{ij})^2}$. For this challenge, we developed multiple models, two of which we present in this paper. Section 5.1 describes the results that optimizes for the target metric and wins the challenge and Section 5.2 talks about an alternative solution to the problem.

5.1. Refiner Network

Our refinement model (cf. Section 4.2) achieved the lowest score of all teams at 25.23. The scores of the second (PAU) and third teams (tomcarrot) were 27.69 and 33.79, respectively. The baseline provided by the challenge organizers was 59.25. Although the refiner network could win the challenge, we found that its generated images contain blurry regions and ghosting effects (see Figure 3). We believe that the reason for these effects lie in the L2 optimization objective, which encourages “washed out” regions.

5.2. Seg2Eye

The original GauGAN architecture starts from just a single downsampled style image, whose style is supposed to be preserved. When directly applied to the OpenEDS dataset, we found that all output images followed the same style, i.e., a complete lack of style preservation. We introduced our novel SPADE+Style blocks to incorporate style information via AdaIN, specifically allowing for the merging of style information from several images via a style encoder and max-reduction. We found that this improves model performance in terms of both L2 score and visual quality. However, without any additional encouragement, the network did not learn a consistent latent style space that allowed for interpolation in the style latent space. For this reason, we added a style consistency loss on both the latent code and the Gram matrix of the encoder feature maps. In this scenario, style was applied considerably better and we could perform latent space walks as shown in Figure 1

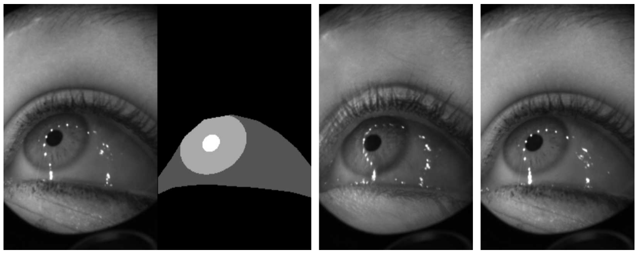

Figure 4: Qualitative outputs of Seg2Eye. From left to right are (1) one of the style image inputs, (2) target segmentation mask input, (3) generated image, and (4) target real image taken from the validation set. The generated image closely follows the segmentation mask and input style.

The final Seg2Eye approach does not achieve the lowest score of all approaches, but produces realistic images of high-perceptual quality. Figure 4 illustrates some example in- and outputs. It can be seen that visual fidelity is vastly improved compared to the Refiner Network. Additionally, Figure 1 shows a style interpolation between two people, demonstrating that a good understanding of style has been met. We also believe that this is more in line with the expected final usage of such a method in generating vast amounts of training data in a controlled manner.

6. Conclusion

In this paper, we described both the winning approach to the OpenEDS Synthetic Eye Generation Challenge and a principled approach to generating person-specific eyes given a semantic segmentation map. Our method, Seg2Eye, is inspired by [10] and suggests a modification to spatially adaptive normalization as introduced by [16] that leads to a a more consistent application of style.

In future work, one should explore different ways to combine the content and style information in the Seg2Eye generator, or combine style information from multiple style images. Furthermore, to truly take advantage of Seg2Eye, an interpretable latent space of eye shapes should be learned such that plausible and high fidelity IR images of eyes can be created from entirely unseen people.

Acknowledgements. This work was supported in part by the ERC Grant OPTINT (StG-2016-717054).

[2] http://evalai.cloudcv.org/web/challenges/challenge-page/354/evaluation

References

[1] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *TPAMI*, 40(4):834–848, 2017. 2

[2] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 2

[3] Yaroslav Ganin, Daniil Kononenko, Diana Sungatullina, and Victor Lempitsky. Deepwarp: Photorealistic image resynthesis for gaze manipulation. In *ECCV*, pages 311–326, 2016. 1

[4] Stephan J. Garbin, Yiru Shen, Immo Schuetz, Robert Cavin, Gregory Hughes, and Sachin S. Talathi. Openeds: Open eye dataset. *CoRR*, abs/1905.03702, 2019. 2

[5] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *CVPR*, pages 2414–2423, 2016. 3

[6] Zhe He, Adrian Spurr, Xucong Zhang, and Otmar Hilliges. Photo-realistic monocular gaze redirection using generative adversarial networks. In *ICCV*, 2019. 1

[7] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017. 1, 2, 3

[8] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *ECCV*, September 2018. 1

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

[10] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *CoRR*, abs/1812.04948, 2018. 1, 2, 4

[11] Joohwan Kim, Michael Stengel, Alexander Majercik, Shalini De Mello, David Dunn, Samuli Laine, Morgan McGuire, and David Luebke. Nvgaze: An anatomically-informed dataset for low-latency, near-eye gaze estimation. In *CHI*, 2019. 2

[12] Kangwook Lee, Hoon Kim, and Changho Suh. Simulated+unsupervised learning with adaptive data generation and bidirectional mappings. In *ICLR*, 2018. 1

[13] Yanghao Li, Naiyan Wang, Jiaying Liu, and Xiaodi Hou. Demystifying neural style transfer. In *IJCAI*, pages 2230–2236, 2017. 2

[14] Seonwook Park, Shalini De Mello, Pavlo Molchanov, Umar Iqbal, Otmar Hilliges, and Jan Kautz. Few-shot adaptive gaze estimation. In *ICCV*, 2019. 1

[15] Seonwook Park, Xucong Zhang, Andreas Bulling, and Otmar Hilliges. Learning to find eye region landmarks for remote gaze estimation in unconstrained settings. In *ACM Symposium on Eye Tracking Research and Applications (ETRA)*, 2018. 1

[16] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *CVPR*, 2019. 1, 2, 3, 4

[17] Omkar M Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. In *BMVC*, 2015. 2

[18] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015. 2

[19] Matan Sela, Pingmei Xu, Junfeng He, Vidhya Navalpakkam, and Dmitry Lagun. Gazegan - unpaired adversarial image generation for gaze estimation. *CoRR*, abs/1711.09767, 2017. 1

[20] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Joshua Susskind, Wenda Wang, and Russell Webb. Learning from simulated and unsupervised images through adversarial training. In *CVPR*, July 2017. 1

[21] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *CVPR*, June 2018. 2

[22] Erroll Wood, Tadas Baltrušaitis, Louis-Philippe Morency, Peter Robinson, and Andreas Bulling. Gazedirector: Fully articulated eye gaze redirection in video. In *Computer Graphics Forum*, volume 37, pages 217–225. Wiley Online Library, 2018. 1

[23] Yu Yu, Gang Liu, and Jean-Marc Odobez. Improving few-shot user-specific gaze adaptation via gaze redirection synthesis. In *CVPR*, June 2019. 1