Facial Image Colorization and Editing with GAN Inversion

Furkan Güzelant Melih Coşğun Umut Başer

***Abstract*—Facial image colorization has a number of useful applications, including the reconstruction of old images. By extending this problem, we also explored the manipulation of the reconstructed image and added the ability to colorize an image based on a color histogram of another image. Our methodology employs a StyleGAN-based approach for editing and colorizing images.**

***Index Terms*—Image Colorization, Image Editing, Color Histogram Conditioned Image Colorization**

I. Introduction

IMAGE colorization brings black-and-white pictures to life by adding vibrant colors. This project explores using advanced technology, like deep neural networks, to make grayscale images colorful. By blending artificial intelligence with human creativity, we aim to enhance visual storytelling and restoration. Using a pretrained StyleGAN2 [1] model for face generation, we aim to colorize grayscale face images to obtain realistic colored images.

In today's digital world, deep learning has transformed image editing. By using neural networks, we can now enhance, manipulate, and transform images with unparalleled accuracy and speed. This project also explores the power of deep learning in image editing, applying transformations such as altering smiles and adding glasses.

In the initial phase of this project, we have investigated the diverse architectural approaches, including Pixel2Style2Pixel [2] and High Fidelity GAN Inversion [3], and employed various techniques to enhance the colorization of a given 256x256 pixel grayscale input image. We have employed two distinct approaches to address the issue of image colorization. The first approach involved histogram-conditioned image colorization, while the second was unconditional image colorization. In the unconditional image colorization process, we leveraged the pSp and HFGI architectures with certain modifications. We have attempted to train the encoders with different losses and have presented our results in the Results section. With regard to conditioned image colorization, we have attempted to feed the color histogram information into the architecture in different ways. Finally, for the purpose of editing a colorized image, we have used the output latent codes of our colorization techniques and edited the attributes of the images before feeding them into the StyleGAN decoder by using our calculated attribute directions.

II. Related Work

A. Image Colorization

Image colorization is the task of adding color to grayscale images. While this task is seemingly straightforward, it contains challenges due to the limited information inherent in a grayscale image. Inferring plausible colors is a non-trivial problem as a grayscale pixel can be filled with a wide range of potential colors, and many traditional methods rely on user interaction by manually specifying the color information for regions, which is both time-consuming and tedious.

However, recent deep learning works, particularly those using Generative Adversarial Networks (GANs), allowed visually compelling automatic colorization since GANs can learn the complex mappings from grayscale to color images.

Image Colorization with Generative Adversarial Networks: Nazeri et al. [4] introduce a GAN-based approach for image colorization tasks. The authors used DCGAN guidelines and modified the architecture to be a conditional GAN.
The authors provide a comparative analysis between their results and a baseline network derived from the U-Net architecture, showing that a GAN-based approach can produce better-colorized images. The authors share the results of their network for multiple datasets to show that the method applies to many domains. However, the authors also mention a drawback of GANs: the tendency to colorize objects with the most frequently seen colors.

PalGAN: Image Colorization with Palette Generative Adversarial Networks: Wang et al. [5] propose to use a two-stage process for image colorization. In the first stage, a color palette is estimated from the grayscale image, and in the second stage, pixel-wise color assignment is performed by a model conditioned on the estimated color palette. The authors introduce a chromatic attention module that aligns the color affinity with semantics and local intensity details to avoid color bleeding effects. The evaluation done by the authors shows that PalGAN outperforms prior methods in visual quality and perceptual metrics in ImageNet and COCO-Stuff benchmarks. A vital advantage of this method comes from using a color palette for conditioning. This allows controllable colorization by the manipulation of color palette probabilities. Hence, it allows for the mitigation of GAN's training set bias.

B. Image Editing

While image colorization only adds color information to the image, image editing covers a broader range of manipulations, including changing facial attributes, changing an object's shape, or converting an image to a different style. Generating semantically consistent edited images is a more complex task. GANs, by learning semantic features and encoding them in the latent space, allow semantically consistent image editing. However, to edit an image in the GAN's latent space, the corresponding latent code of the image is needed.

Interpreting the Latent Space of GANs for Semantic Face Editing: Shen et al. [6] introduce the InterfaceGAN framework to interpret and manipulate real images. They built their idea on the fact that latent spaces of GANs learn disentangled representations of various semantic features. They demonstrate how linear transformations on latent code allow control of the facial attributes. The framework uses Support Vector Machines (SVMs) to identify facial attribute semantic boundaries. The authors extensively experiment with the assumptions of linear separability of the facial attributes in the latent space. And demonstrates real image editing using GAN Inversion to edit real images.

High-Fidelity GAN Inversion for Image Attribute Editing: Wang et al. [3] introduces Distortion Consultation Inversion (DCI), which uses a distortion map to enhance reconstruction fidelity while maintaining editability. DCI addresses the trade-off between reconstruction quality and editability by fusing low-rate latent code with high-rate latent maps from the distortion map. Additionally, the authors use Adaptive Distortion Alignment (ADA) to align the edited image with the distortion map to preserve high-level details during editing. The framework applies edit operations on the low-rate latent code and uses consultation to recover lost details. Experiments conducted by the authors show that HFGI can preserve high-level details better than existing editing approaches.

III. Method

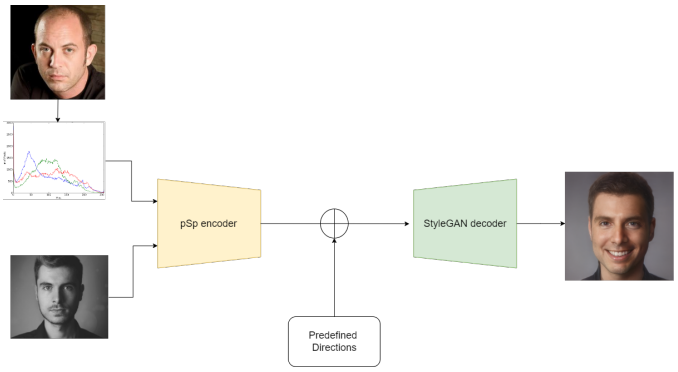


Fig. 1. The diagram illustrates the overall process of the proposed methodology. The architecture takes grayscale images, color histograms of colored images, and desired attribute directions as inputs.

To colorize and edit a given input grayscale image, it is necessary to recognize that these processes are independent of each other. Therefore, our methodology has been divided into three steps. In the first step, we employ a pixel2style2pixel encoder architecture with some modifications to obtain a colored version of a given grayscale image. In the second step, we utilize the image editing pipeline from InterFaceGAN. Finally, in the third and final step, we color the images with a given color histogram of another image. Overall pipeline of our methodology is shown in Figure 1.

A. Dataset

We are using CelebA-HQ [7] dataset, which contains 30000 high resolution human face images. The original dataset is 1024x1024 pixels but we are using a resized 256x256 pixels variant[1]. The dataset is biased towards people under 35 years old, also there is an over-representation of white ethnicity, constitutes 70% of the dataset [8]. However, it is a highly used dataset, and pretrained models are publicly available. Therefore, we decided to use it as our dataset.

B. Colorization using encoder

For our colorization task, we've implemented an encoder-based methodology to enhance our results. The process begins by encoding the features of a grayscale image using two distinct encoders. Initially, we employ the Pixel2Style2Pixel (pSp) [2] encoder to derive low-rate (18x512) latent vectors. These vectors serve as the foundation for our colorization process. However, to further refine the colorization and ensure fidelity to the original image, we incorporate a consultation encoder derived from the High Fidelity Gan Inversion (HFGI) [3] model. This consultation encoder produces high-rate latent vectors that capture finer details, compensating for any loss incurred during the initial encoding stage with the pSp encoder.

With the encoded features in hand, we leverage a pre-trained StyleGAN generator to generate the colorized image. By feeding the encoded features into the generator, we effectively infuse the grayscale image with color information, resulting in a realistic colorization.

To evaluate the fidelity and quality of our colorization process, we utilize a combination of losses to guide the encoder during training. The pixel-wise L2 loss ensures that the output image closely matches the input image at a pixel level, promoting overall fidelity and better color selection.

Pixel-wise L2 Loss:

$$L_2(x) = \|x - \text{pSp}(x)\|_2^2 \tag{1}$$

Additionally, to capture perceptual similarities, we employ the LPIPS loss, which has demonstrated superior image quality preservation compared to the standard perceptual loss.

LPIPS Loss:

$$L_{\text{LPIPS}}(x) = \|F(x) - F(\text{pSp}(x))\|_2^2 \tag{2}$$

where $F(\cdot)$ denotes the perceptual feature extractor.

To address the challenge of encoding facial images by preserving the identity of the image, we employ a specialized recognition loss that measures the cosine similarity between the output image and the original input.

Identity Loss:

$$L_{\text{ID}}(x) = 1 - \langle R(x), R(\text{pSp}(x)) \rangle \tag{3}$$

where $\langle \cdot, \cdot \rangle$ denotes the cosine similarity between the output image and its source.

The total loss is calculated by multiplying the loss functions with constant coefficients:

Total Loss:

$$L(x) = \lambda_1 L_2(x) + \lambda_2 L_{\text{LPIPS}}(x) + \lambda_3 L_{\text{ID}}(x) \tag{4}$$

[1]Obtained from: https://www.kaggle.com/datasets/badasstechie/celebahq-resized-256x256

By optimizing these loss functions, our encoder learns to refine style features to accurately incorporate color information, ensuring high-quality colorization outputs. The original pSp [2] architecture employs an additional regularization loss function to encourage the encoder to output latent vectors that are more closely aligned with the average latent vector.

Regularization Loss:

$$L_{\text{reg}}(x) = \|E(x) - \overline{w}\|_2 \tag{5}$$

Nevertheless, in the methodology that we employed, we did not utilize this loss function and set the constant coefficient for this function to 0. Our architecture combines the strengths of multiple encoders and leverages a sophisticated generator model to achieve good results in grayscale image colorization, faithfully preserving details and enhancing visual appeal.

C. Colorization with histogram condition

In our colorization pipeline, we've aimed to condition the colorization process using image histogram information. This technique aims to enhance the fidelity and accuracy of colorization by leveraging the statistical distribution of colors in the target image.

To implement this, we begin by computing the color histogram for each channel of the target image. This histogram encapsulates valuable information about the distribution and intensity of colors present in the image. Next, we implemented a convolutional network designed to extract relevant features from these histograms. By processing the histograms through this network, we obtain a condensed representation of the color distribution, which captures important characteristics for guiding the colorization process. To integrate this histogram information into our colorization framework, we concatenate the output of the convolutional network with the mid-layer output of the pSp encoder. This augmented representation is then fed into the last layers of the pre-trained StyleGAN [1] generator. These final layers of StyleGAN are crucial for capturing fine details and nuances, including color information. Therefore, our objective is to train the network to discern the intricate relationship between the color histogram and pixel colors, thereby enabling it to generate face images that are conditioned on this histogram information. By understanding and learning from the statistical distribution of colors in the histogram, the network gains insights into how different colors are distributed across the image. This understanding allows the network to make informed decisions about the colorization of individual pixels, ensuring that the generated face images accurately reflect the characteristics of the input histogram.

We also introduced a new loss function to our network called histogram loss, which is measured by calculating the absolute difference between the output and target images. This loss ensures that the colorization aligns with the color histogram of the target images.

D. Image editing

The image editing step of the pipeline can be isolated from the previous steps. Editing occurs on a given latent code of the colorized version of an image that is generated from the pSp encoder [2]. To edit an image with a desired attribute, we sum the output of the pSp encoder with the desired attribute's direction to obtain a new code in the GAN latent space. The power of the attribute is modified by a scalar coefficient, which is multiplied with the desired attribute's direction. To calculate the directions of the given set of attributes, we followed the steps outlined in the InterFaceGAN [6].

The initial step involves training an attribute classifier on an auxiliary dataset with a given set of attributes as labels. In the second step, a substantial number of randomly generated images are saved with their StyleGAN [1] latent codes. Subsequently, the generated images are provided as inputs to the classifier, and the top results for each attribute are saved with the latent codes. Finally, the latent codes and their corresponding attributes are used to train an SVM, which identifies the directions of the attributes in the StyleGAN latent space.

IV. Experiments

A. Training pSp encoder for reconstructing grayscale images

In our initial experiment, we focused on training the Pixel2Style2Pixel encoder to reconstruct grayscale images. This involved feeding grayscale face images into the network and assessing the discrepancy between the input image and the one generated using StyleGAN. Through this process, the encoder learned to capture the stylistic features inherent in grayscale images. However, in this experiment, our primary goal was not colorization per se, but rather to harness the encoder's ability to extract high-level features such as pose and facial structure from grayscale images. The colorization aspect was added by randomly initializing latent vectors corresponding to the final six layers of StyleGAN, which are pivotal for capturing fine details, including color schemes, in the generated images.

B. Training pSp encoder with RGB target images

As our second experiment on image colorization, we embarked on training the pSp [2] encoder using RGB target images. Our methodology in this experiment involved directing the model to calculate the loss between the colorized version of the input grayscale image and the colorized image generated by the pre-trained StyleGAN generator. This approach was carefully designed to prompt the pSp encoder to learn and seamlessly integrate color information into the encoded style features.

To achieve this, we employed a combination of loss functions, specifically optimizing the L2 and identity loss functions. By optimizing these loss functions, we encouraged the pSp encoder to refine its understanding of color nuances and embed them into the encoded style features. This process effectively encouraged the encoder to learn the intricate relationships between grayscale input images and their corresponding colorized counterparts. Through this experiment, we aimed to equip the pSp encoder with the capability to generate colorized images that closely resembled the target RGB images. By leveraging the power of loss optimization, we sought to enhance the realism of the colorization process, ensuring that

the generated images not only retained the structural details of the original grayscale images but also exhibited vibrant and accurate color representations.

C. Training Consultation Encoder of HFGI

To refine the performance of our colorization network, we focused on experimenting with the consultation encoder of the High-Fidelity GAN Inversion (HFGI) [3] model. Our objective was to augment the reconstruction capabilities of our network by harnessing high-rate latent codes extracted from the difference between the input grayscale image and the image generated using low-rate latent codes obtained from the pSp encoder. To execute this experiment, we utilized the pSp encoder trained in our previous experiment and trained the consultation encoder while keeping other models frozen. This approach allowed us to focus on enhancing image-specific details through the extraction of high-rate latent maps from the consultation encoder of HFGI, while concurrently leveraging the low-rate latent codes from the pSp encoder to govern the style features and color scheme.

By integrating the consultation encoder into our colorization network, we aimed to enrich the reconstruction process with finer details, thereby elevating the overall fidelity of the colorized outputs. This synergistic approach is followed to ensure that our network could capture both global style features and intricate image-specific nuances, resulting in colorized images that not only retained the essence of the original grayscale inputs but also exhibited enhanced visual quality and coherence.

D. Colorizing image with histogram condition

We experimented with various color histogram conditioning methods. We conducted experiments on the calculation of the color histogram, the layer that the conditioned feature is fed, and the way the conditioned feature and feature maps are combined.

Firstly, we experimented with 2D color histograms for a given image by calculating them for different combinations of color channels, that is, Red-Green, Red-Blue, and Green-Blue. We employed a network composed of transposed convolution layers to extract features from the color histogram. Then, we fed the output of this network to some mid-layer output feature map of the pSp encoder. We experimented with various resolutions of conditioned features that correspond to the different layers of the pSp encoder. Additionally, we explored augmentation techniques like addition and concatenation, integrating the conditioned features with the feature maps of the pSp encoder.

Furthermore, we explored the utilization of 1D color histogram vectors for each channel of the RGB image. This approach involved first deriving these histogram vectors and then flattening and repeating them to align with the size of a mid-layer output of the pSp encoder. Once the histograms were appropriately sized, we concatenated them with the feature maps and passed the combined data through a 1x1 convolution layer. This step aimed to integrate the color histogram information directly into the feature space of the pSp encoder, thereby enabling the model to leverage both structural and color-related cues for more accurate and nuanced colorization.

E. Image editing experiments

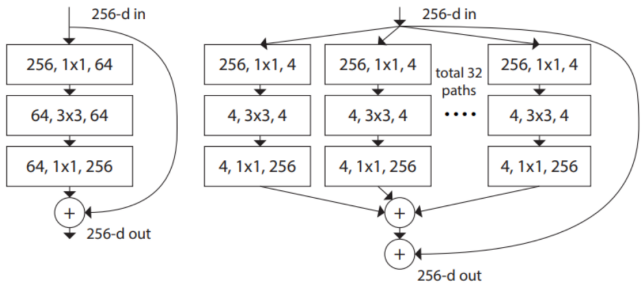


Fig. 2. Difference between ResNeXt's residual block and ResNet's residual block. The left figure is original ResNet.

In order to perform image editing, we proceeded in accordance with the instructions provided by the InterFaceGAN. Initially, we trained an attribute classifier on the CelebFaces Attributes dataset. We focused on the following eight attributes for a given human face: bald, black hair, blond hair, eyeglasses, goatee, male, mustache, and smiling.

We utilized the ResNeXt architecture [9] as our classifier architecture. ResNeXt is a modified version of the original ResNet architecture [10]. It employs the same residual connection network concept but modifies the residual block of the ResNet by incorporating different sized convolutional layers. The differences between residual blocks are shown in Figure 2. Following the training of the attribute classifier network, we achieved an accuracy of 96.4% on the test data.

As the second step in the editing process, we employed the StyleGAN implementation utilized by Pixel2Style2Pixel [2], which incorporates pretrained StyleGAN weights. These weights were also utilized to train our pSp encoder. A total of 500,000 images were generated with random latent vectors, and the images were saved with their corresponding latent codes in the StyleGAN latent space. It is important to note that during the generation of random images from StyleGAN, we did not permit the use of random noise to stay loyal to our latent codes.

In the third step of the process, the trained attribute classifier was utilized to classify the attribute scores for each generated image. The resulting attribute scores were then saved in association with the corresponding latent codes of the images.

As the final step in the process of identifying editable directions, we selected the top 10,000 images with the highest confidence scores for each attribute. These images were then used to train a support vector machine (SVM) classification model. The obtained directions were used to sum the desired attribute with the output of the pSp encoder, effectively moving the latent code in that direction. A scalar coefficient was defined to determine the strength of the given attribute's direction. Finally, the modified latent code is provided to the pretrained StyleGAN model, which generates the edited image.

Method	LPIPS	FID[2]	L2
pSp Based	.32760	107.028	.07736
HFGI Based	.17842	106.357	.02904
pSp 2D Histogram Conditioned	.20662	131.872	.04157

TABLE I
QUANTIATIVE RESULTS FOR IMAGE COLORIZATION.

V. RESULTS

A. Image colorization results

Table III provides a comprehensive overview of the results obtained from our colorization experiments. The images labeled "pSp GS" showcase the outcome of our initial experiment, wherein the pSp encoder was trained using grayscale images for reconstruction, followed by colorization via random initialization of latent vectors corresponding to StyleGAN's final layers. This experiment highlighted that while random initialization of final latent vectors led to alterations in color schemes, it also induced changes in facial structure and shape. This observation underscores the importance of latent vector initialization in influencing not just colorization but also broader image characteristics. In contrast, the results from our second experiment, denoted as "pSp RGB," reveal a different approach. Here, the encoder was trained using RGB target images, leveraging computed losses to guide the colorization process toward realism. Despite the encoder's improved understanding of colorization, the resulting outputs fell short in terms of reconstruction performance.

The images labeled "HFGI" represent the result of our third experiment, wherein we integrated the consultation encoder to enhance reconstruction performance during colorization. This augmentation yielded output images more closely resembling target images in both color and shape. As evidenced by Table I, the pixel-wise distance and perceptual loss between the output and target images were notably reduced, affirming the efficacy of the consultation encoder in refining colorization outcomes.

Furthermore, the experiment denoted as "pSp 2D Histogram Conditioned" underscores our exploration into color histogram conditioning. In this experiment, we generate the colorized image by conditioning it on its color histogram. Table I showcases the better performance of the color histogram-conditioned pSp encoder compared to its non-conditioned counterpart. This observation underscores the significance of color histogram information to augment the encoder's understanding of colorization tasks, leading to more accurate and realistic outputs.

Table IV presents the results of our image colorization experiment using various condition images. The findings show that the colorized images prominently feature the most dominant color from the condition image, particularly in the backgrounds. However, conditioning on the color histogram results in some loss of detail. On the other hand, the condition images only affect the color scheme of the output images, preserving the encoded face structures and shapes as desired.

[2]Calculated using FFHQ Dataset with 2000 images

B. Image editing results

Table II presents the visual results of an image editing task. The input grayscale image is colorized via our colorization method, and the edited images for different attributes are obtained in an interpolation using a coefficient between [0, 30]. Two key observations can be made from the visual results. First, the inversion robustness of the image is inversely correlated with the strength coefficient of the attributes. A second observation is that some attribute directions are correlated with each other, even though this correlation does not directly occur in real life. For instance, Table II shows that the baldness attribute is correlated with the goatee attribute. Additionally, it can be observed that the baldness and eyeglasses attributes age the image when the strength is increased.

VI. CONCLUSION

In the context of this project, we have investigated a range of facial image colorization techniques with varying architectural approaches. Our methodologies utilize the StyleGAN latent space for colorizing and editing images. We have employed advanced encoder architectures, including Pixel2Style2Pixel and High Fidelity GAN Inversion, which have yielded promising results in colorizing grayscale images. In addition, we have developed a method for adding a condition to the colorization process, utilizing another image's color histogram to provide greater control to the user. We have modified the pSp architecture and explored various approaches for incorporating condition information into the training of the encoder. Furthermore, we have implemented the ability to edit the attributes of the output colorized image by modifying its latent code within the StyleGAN latent space. In summary, we have explored a range of methodologies to achieve our objective. We believe that our work demonstrates an effective approach to restoring and modifying historical images through the use of generative networks.

REFERENCES

[1] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of stylegan," 2019.

[2] E. Richardson, Y. Alaluf, O. Patashnik, Y. Nitzan, Y. Azar, S. Shapiro, and D. Cohen-Or, "Encoding in style: a stylegan encoder for image-to-image translation," 2020.

[3] T. Wang, Y. Zhang, Y. Fan, J. Wang, and Q. Chen, "High-fidelity gan inversion for image attribute editing," 2021.

[4] K. Nazeri, E. Ng, and M. Ebrahimi, *Image Colorization Using Generative Adversarial Networks*, p. 85–94. Springer International Publishing, 2018.

[5] Y. Wang, M. Xia, L. Qi, J. Shao, and Y. Qiao, "Palgan: Image colorization with palette generative adversarial networks," 2022.

[6] Y. Shen, J. Gu, X. Tang, and B. Zhou, "Interpreting the latent space of gans for semantic face editing," 2020.

[7] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," 2018.

[8] H. Zhu, W. Wu, W. Zhu, L. Jiang, S. Tang, L. Zhang, Z. Liu, and C. C. Loy, "Celebv-hq: A large-scale video facial attributes dataset," 07 2022.

[9] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," 2016.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

APPENDIX

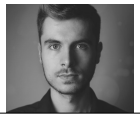
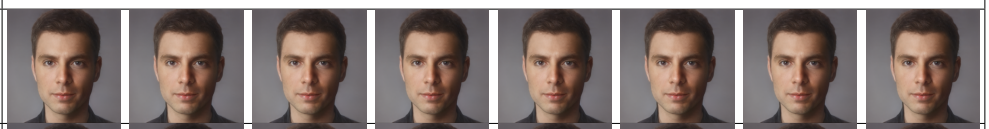
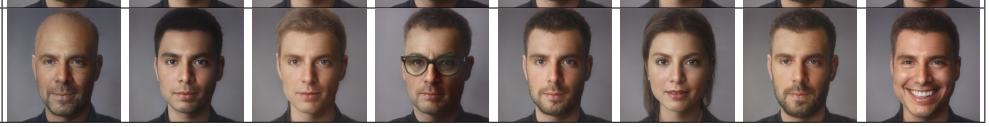
	Attributes							
	Bald	Black Hair	Blonde Hair	Eyeglasses	Goatee	Male	Mustache	Smiling
Coefficient = 0.0								
Coefficient = 10.0								
Coefficient = 20.0								
Coefficient = 30.0								

TABLE II

THE RESULTS OF THE COLORIZATION AND EDITING OF AN INPUT GRAYSCALE IMAGE (TOP LEFT). THE ROWS REPRESENT THE SCALE OF THE ATTRIBUTE DIRECTION WITH A COEFFICIENT. THE COEFFICIENT IS EQUAL TO ZERO WHEN THE IMAGE HAS NOT BEEN EDITED, BUT ONLY COLORIZED.

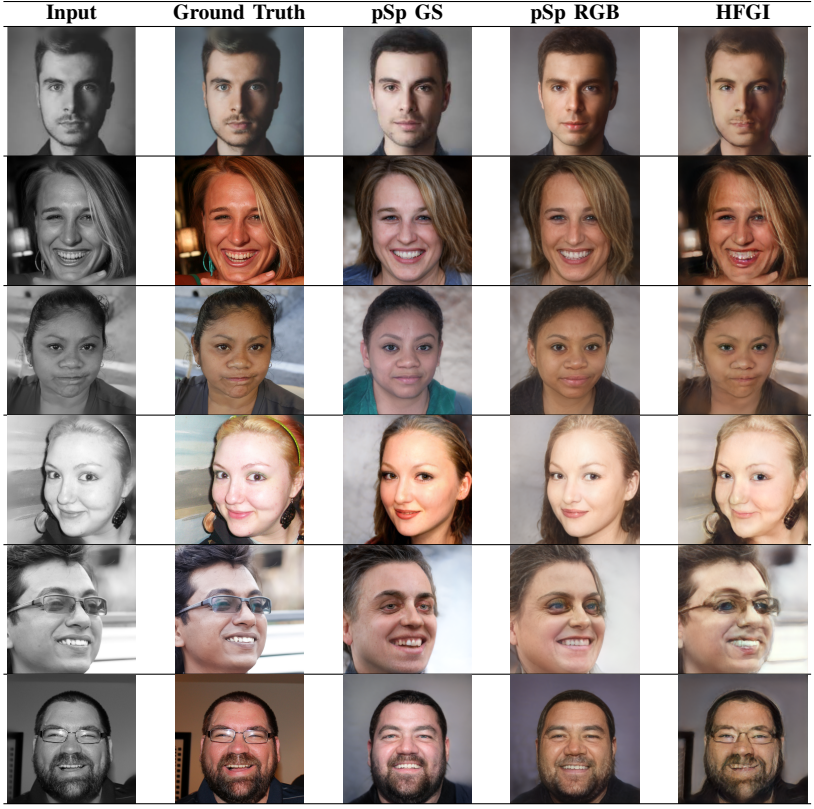
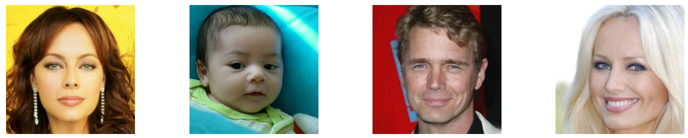
Input	**Ground Truth**	**pSp GS**	**pSp RGB**	**HFGI**
				

TABLE III

COLORIZATION RESULT

		Condition Image 1	Condition Image 2	Condition Image 3	Condition Image 4



Input	Ground Truth	Conditioned Image 1	Conditioned Image 2	Conditioned Image 3	Conditioned Image 4

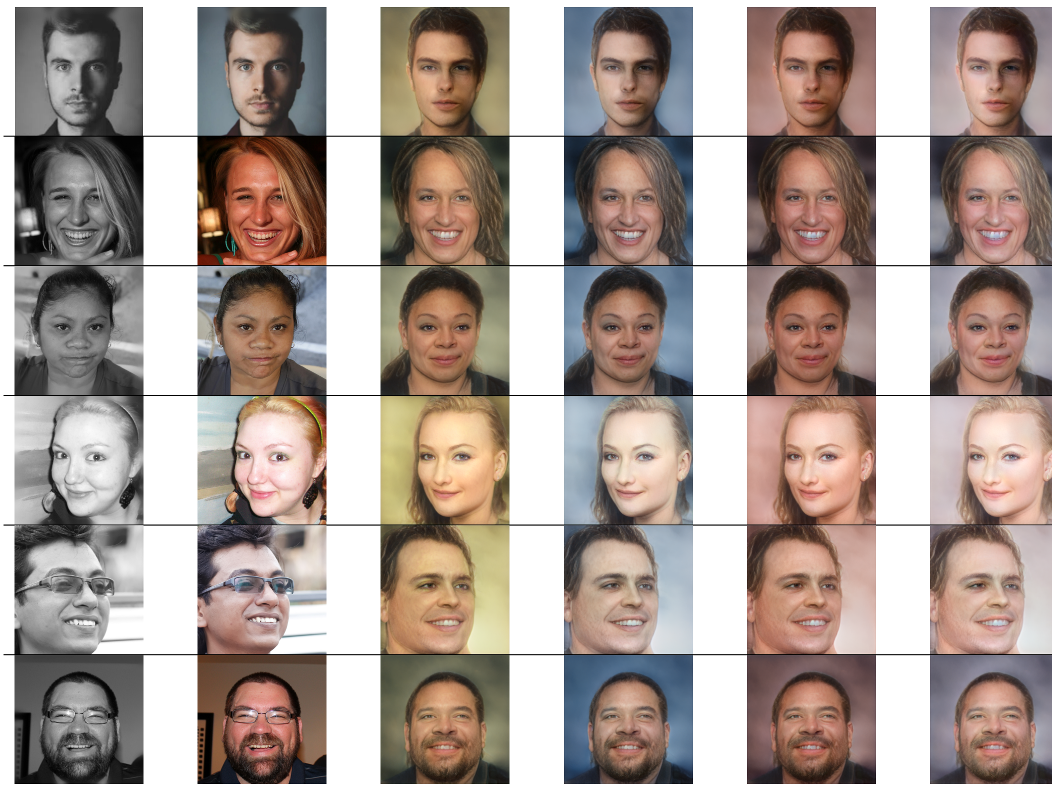


TABLE IV
CONDITIONED COLORIZATION RESULTS