Facial Features Manipulation Using Free Form Input

Student: Andrey Gusev
Advisor: Maksim Artemev

PicsArt

Motivation

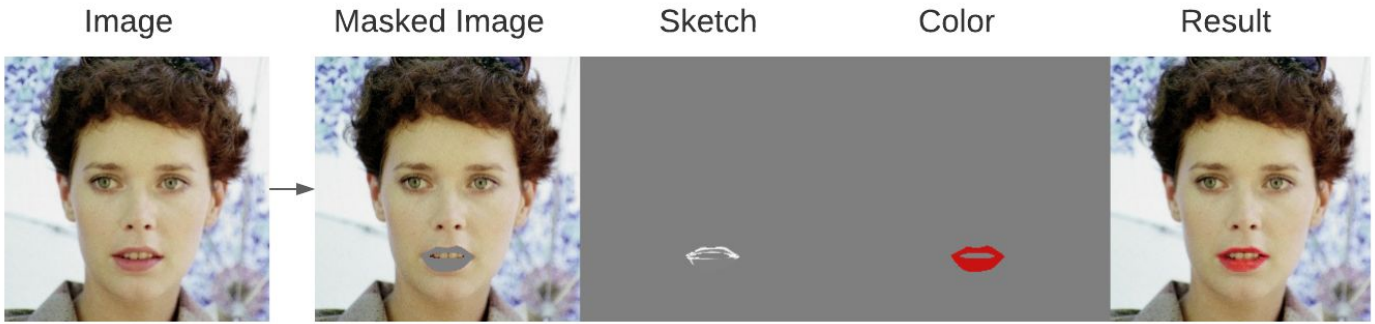
- Many researchers proposed to vary predefined facial attributes:
  - StyleFlow [Abdal et al. 2020]
- A very broad unexplored field of free-form input:
  - DeepFacePencil [Li et al. 2020]
  - MaskGAN [Lee et al. 2020]
  - SC-FEGAN [Jo et al. 2019]

PicsArt

Problem statement

Goal:

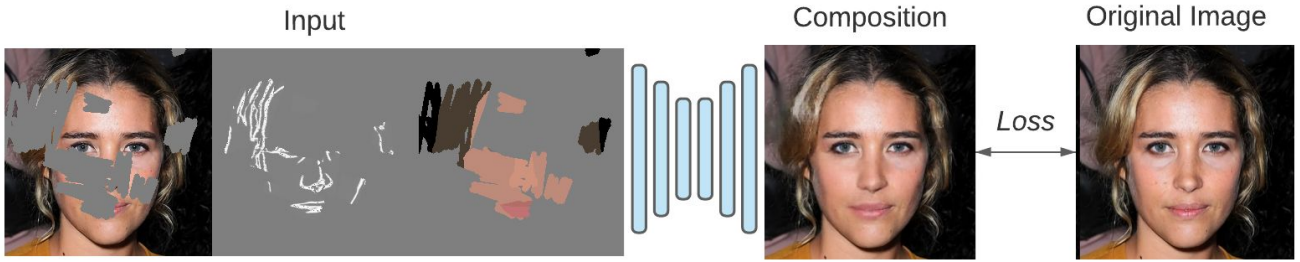
Develop a model for free-form facial attributes manimulation



Problem statement

Goal:

Develop a model for free-form facial attributes manimulation



PicsArt

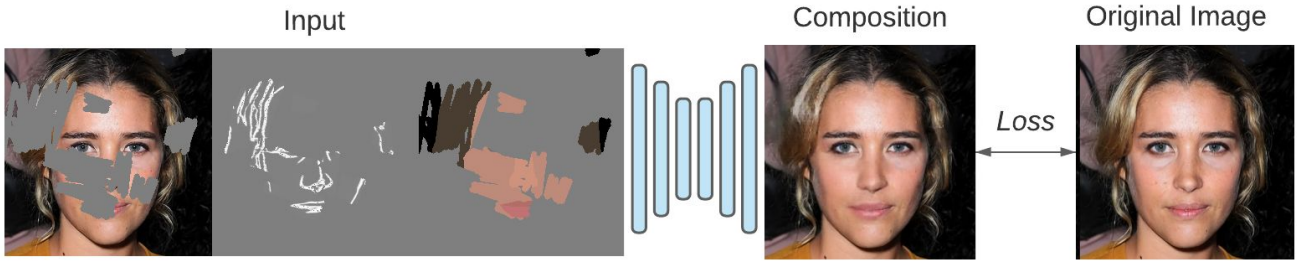
Problem statement

Goal:

Develop a model for free-form facial attributes manimulation

Tasks:

1. Prepare data and implement mask simulation algorithm



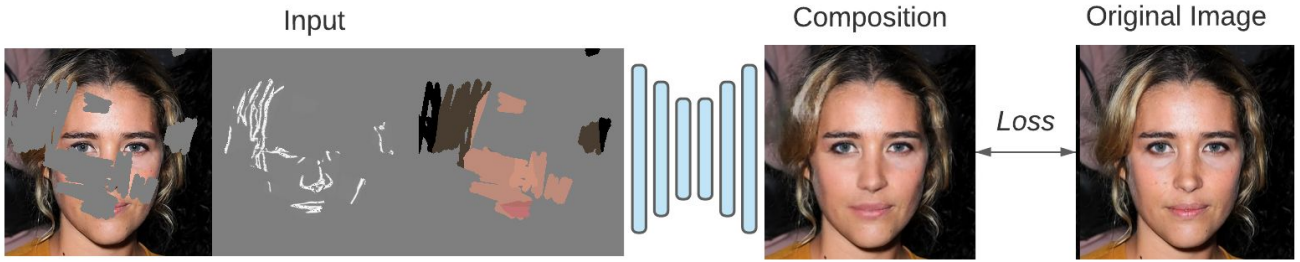
Problem statement

Goal:

Develop a model for free-form facial attributes manimulation

Tasks:

1. Prepare data and implement mask simulation algorithm
2. Design image inpainting model



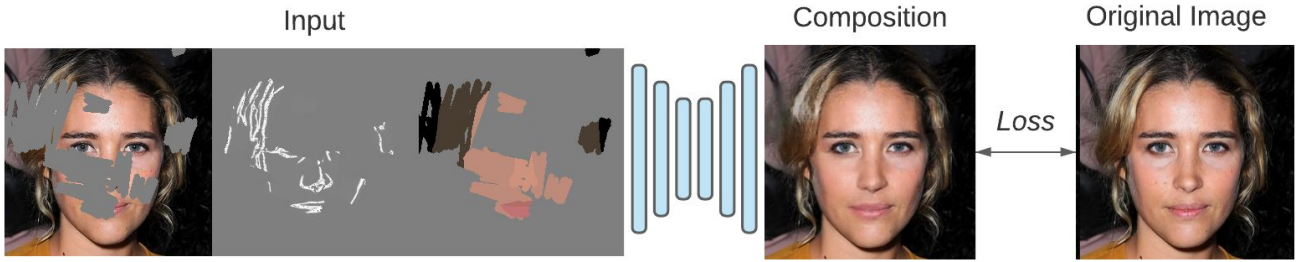
Problem statement

Goal:

Develop a model for free-form facial attributes manimulation

Tasks:

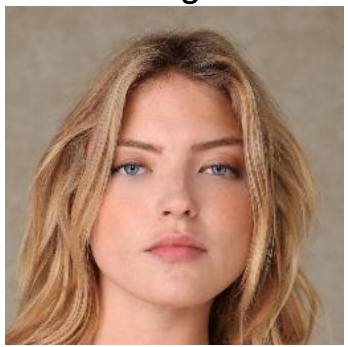
1. Prepare data and implement mask simulation algorithm
2. Design image inpainting model
3. Validate the trained model qualitatively on synthetic and real user inputs



Dataset

1. FFHQ [Karras et al. 2019]
2. LaPa [Lia et al. 2020]
3. CelebAMask-HQ [Lee et al. 2020]

Image



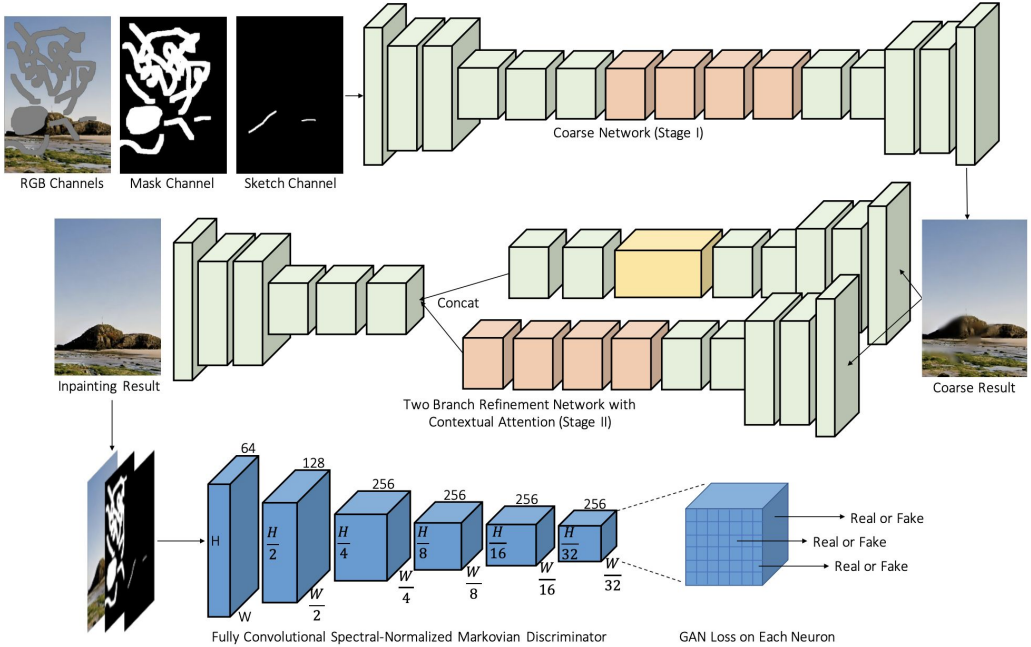
Sketch

Color

Mask generation algorithm

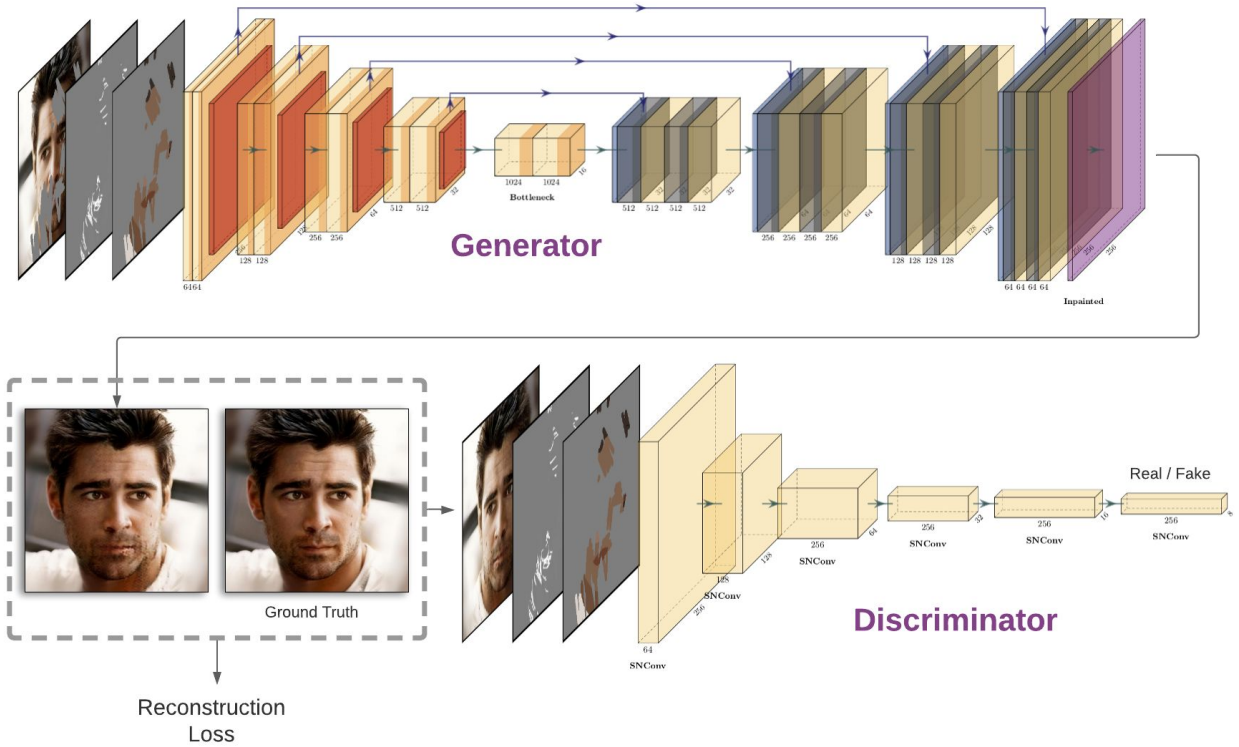
$N_{areas} \sim \mathcal{U}\{\textbf{minAreas maxAreas}\}$;
for $i = 1 \dots N_{areas}$ **do**
  $x_0 \sim \mathcal{N}(H/2, H/4)$;
  $y_0 \sim \mathcal{N}(W/2, W/4)$;
  $\ell \sim \mathcal{U}(0, H/4)$;
  $\rho \sim \mathcal{U}(0, 2\pi)$;
  $N_{actions} \sim \mathcal{U}\{1, \textbf{maxActions}\}$;
  for $j = 1 \dots N_{actions}$ **do**
    $\Delta_\ell \sim \mathcal{N}(0, 1)$;
    $\Delta_\rho \sim \mathcal{N}(0, \pi/20)$;
    $x_j := x_{j-1} + (\ell + \Delta_\ell) \cdot \sin(\rho + \Delta_\rho)$;
    $y_j := y_{j-1} + (\ell + \Delta_\ell) \cdot \cos(\rho + \Delta_\rho)$;
    `Draw line` $(x_{j-1}, y_{j-1}) \rightarrow (x_j, y_j)$

DeepFillv2



Jiahui Yu et al. “Free-form image inpainting with gated convolution” [2019]

Architecture



PicsArt

Architecture

Generator:

- U-Net
- Gated convolutions
- Outermost `ConvT → UpSample + Conv`

Discriminator:

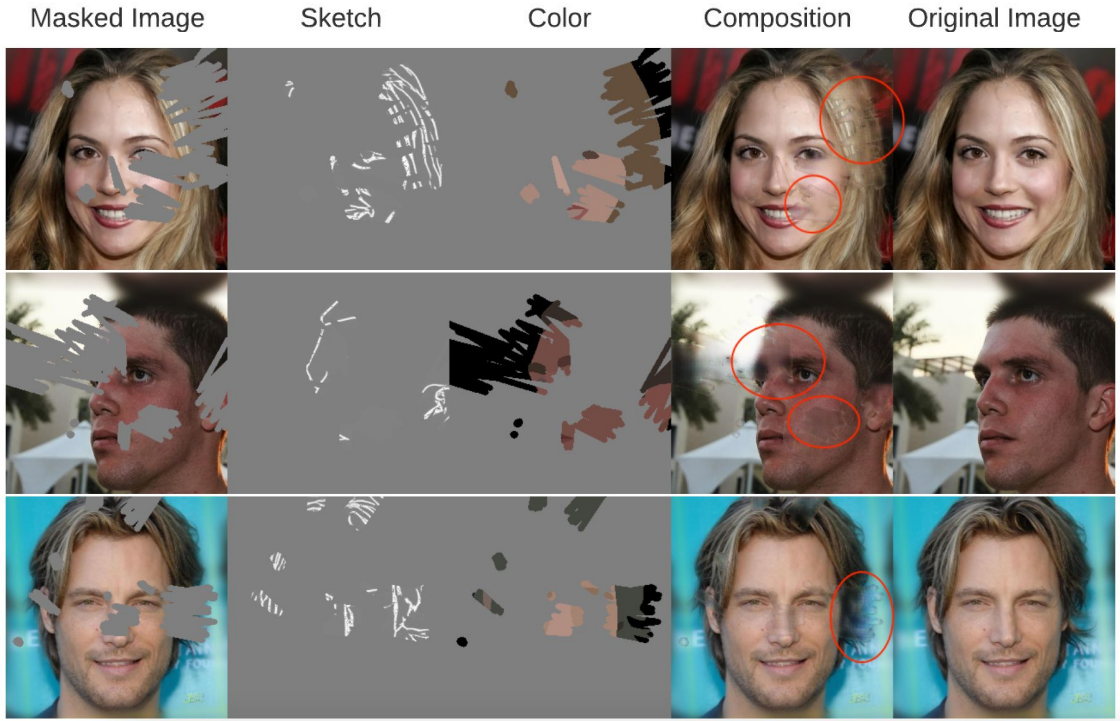
- PatchGAN
- Spectral Normalization

PicsArt

Loss

$$L_D = \mathsf{E}[1 - D(I_{real})] + \mathsf{E}[1 + D(I_{comp})]$$
$$L_G = -\mathsf{E}[D(I_{comp})] + L_{recon}$$
$$L_{recon} = L_1(I_{real}, I_{comp})$$

Droplet artifacts



PicsArt

Perceptual component

$$L_D = \mathsf{E}[a(1 - D(I_{real}))] + \mathsf{E}[a(1 + D(I_{comp}))]$$
$$L_G = -\mathsf{E}[D(I_{comp})] + L_{recon}$$
$$L_{recon} = L_1(I_{real}, I_{comp}) + \gamma L_{percept}(I_{real}, I_{comp})$$

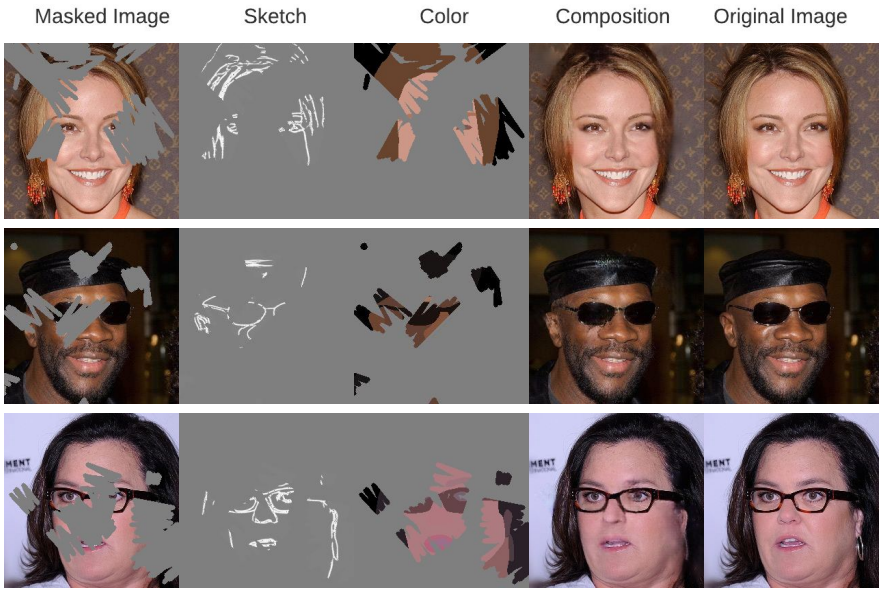
Perceptual component

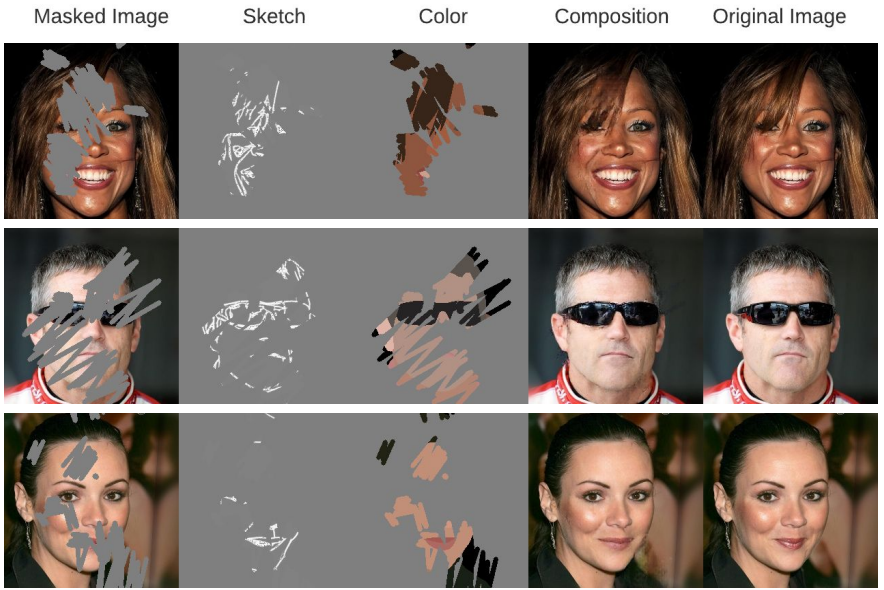
$$L_D = \mathsf{E}[a(1 - D(I_{real}))] + \mathsf{E}[a(1 + D(I_{comp}))]$$
$$L_G = -\mathsf{E}[D(I_{comp})] + L_{recon}$$
$$L_{recon} = L_1(I_{real}, I_{comp}) + \underbrace{\gamma L_{percept}(I_{real}, I_{comp})}_{VGG-16\ (StyleGAN2ada)}$$

Tero Karras et al. “Training generative adversarial networks with limited data”. [2020]

PicsArt

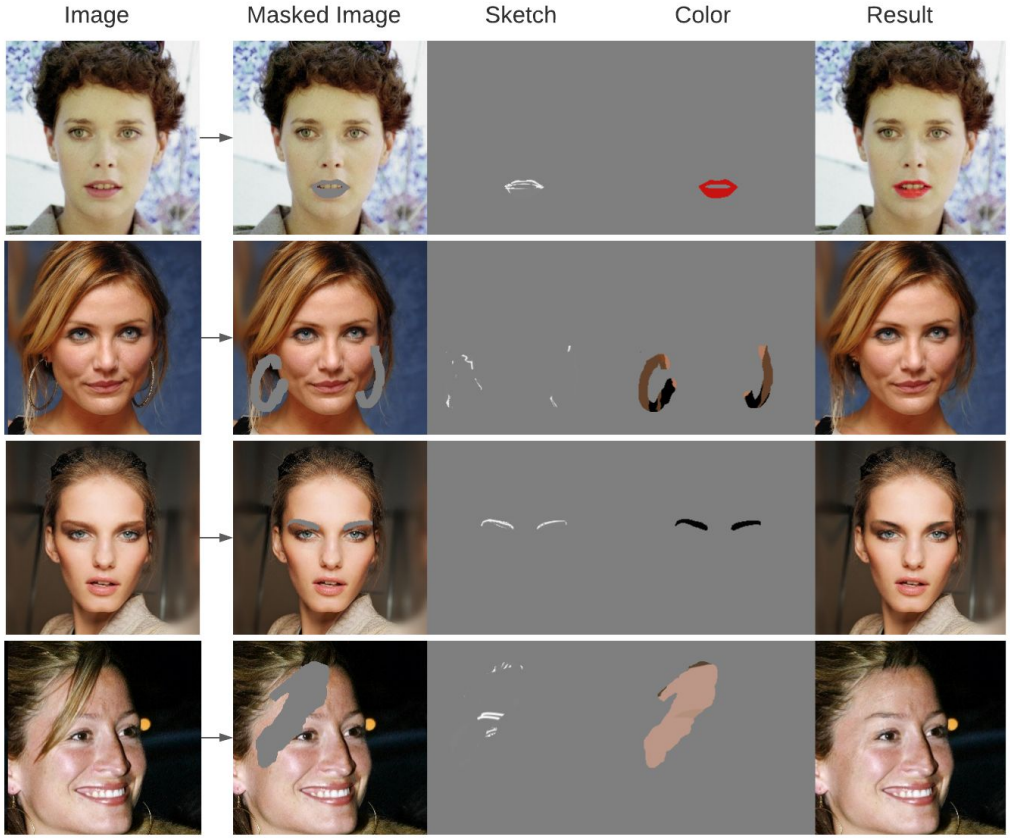
Synthetic input





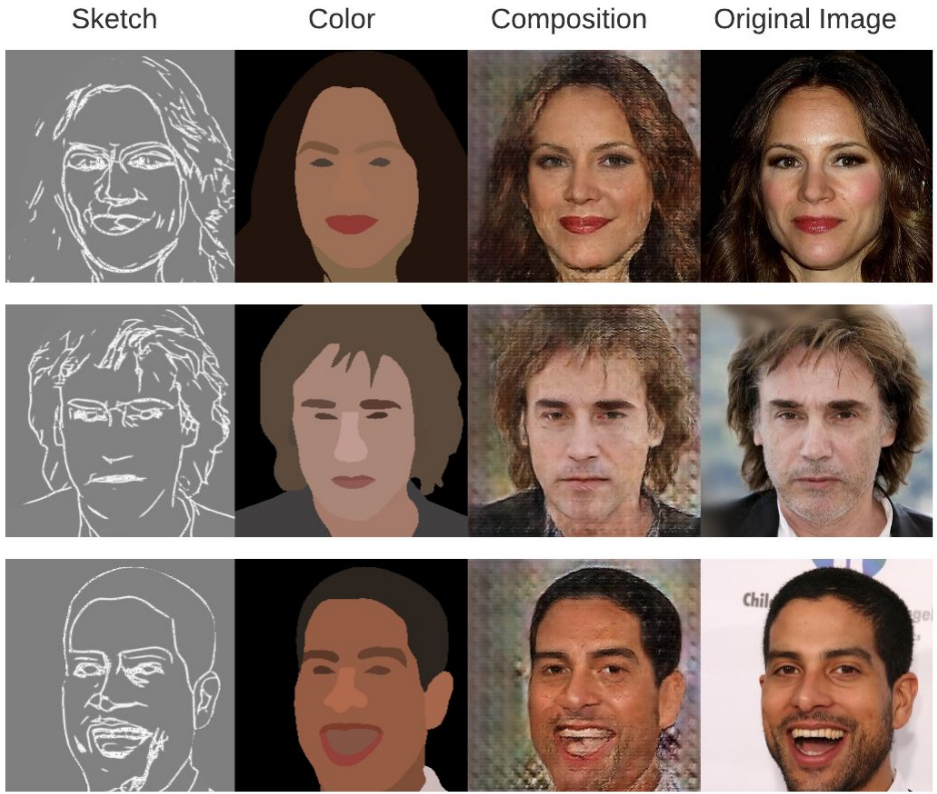
PicsArt

User input



PicsArt

Fullfill experiment



PicsArt

Quantitative analysis

	PSNR(↑)	SSIM(↑)	L2%(↓)
PatchMatch	21.6649	0.9157	0.95
DeepFIllv1	23.7554	0.9145	0.52
SC-FEGAN	29.4912	0.9543	0.2
Ours	31.3779	0.9322	0.18

The first two methods take as input masked images only, while we supplement our model with sketch and color information. The metrics values for PatchMatch, DeepFillv1, SC-FEGAN were taken from the paper [Jo and Park 2019].

PicsArt

Contributions

- Proposed data preparation and mask simulation algorithms;

- Developed model capable of dealing with free-form input layers and producing realistic and faithful compositions;

- The proposed method forms a basis for further research in the field of free-form image editing. All-stages source code is publicly available at GitHub:

  https://github.com/angusev/SCGAN

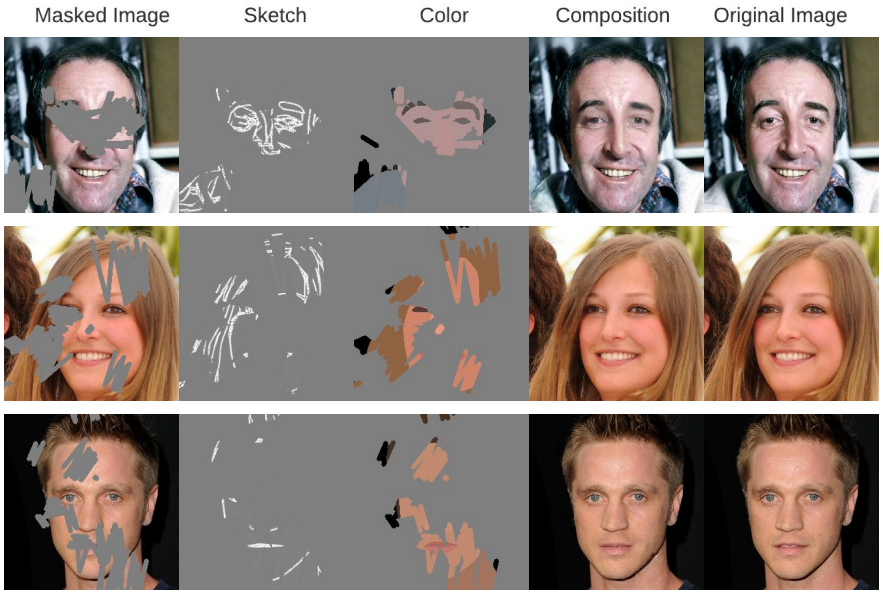
PicsArt

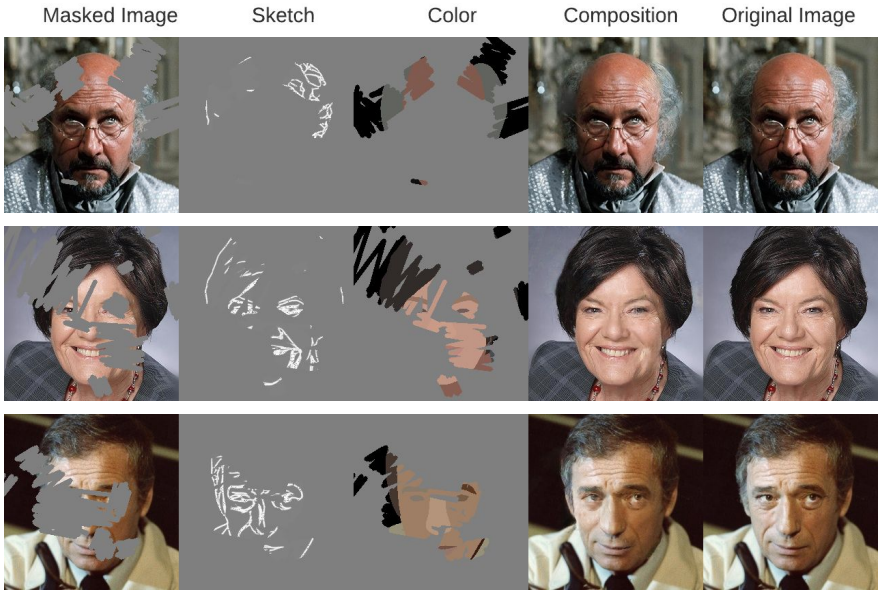
Further research

- Collect user input example to enhance mask generation algorithm
- Train on real data
- Compress the model to port on device
- Try different free-form input options and reinforce input by additional conditions

PicsArt

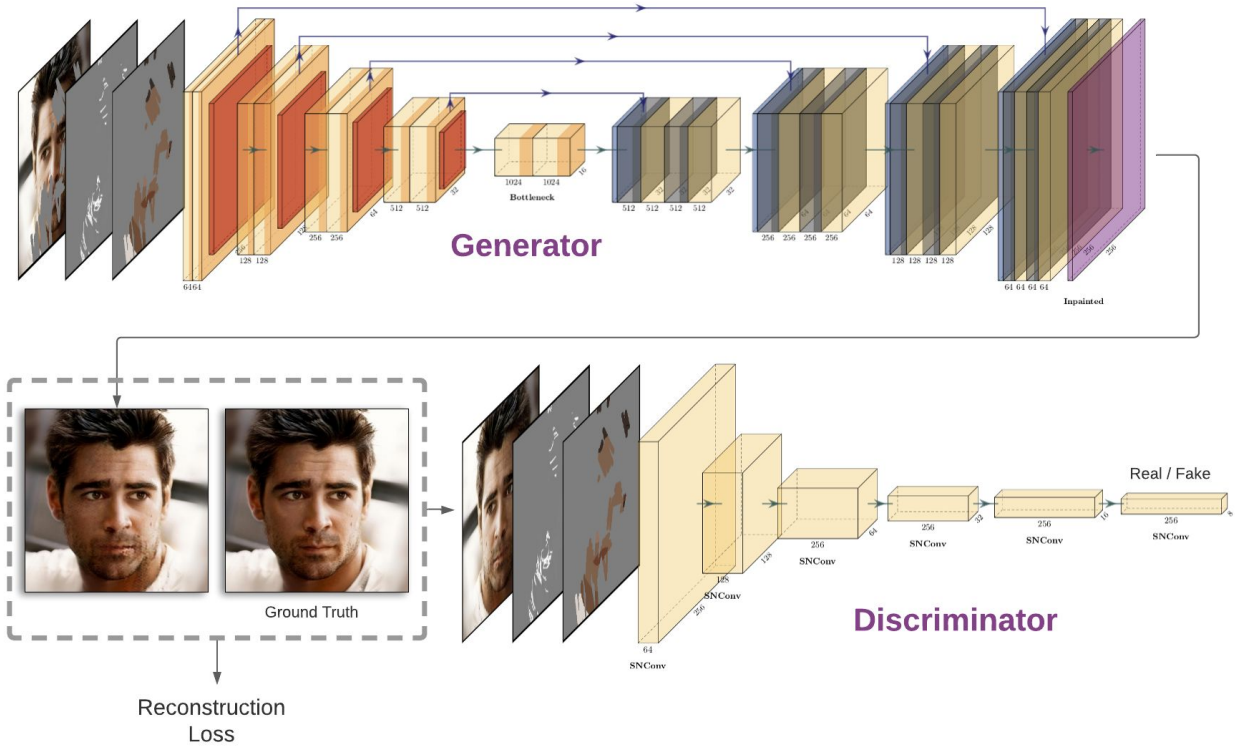
More examples





PicsArt

Architecture



PicsArt

Perceptual component

$$L_D = \mathsf{E}[a(1 - D(I_{real}))] + \mathsf{E}[a(1 + D(I_{comp}))]$$
$$L_G = -\mathsf{E}[D(I_{comp})] + L_{recon}$$
$$L_{recon} = L_1(I_{real}, I_{comp}) + \underbrace{\gamma L_{percept}(I_{real}, I_{comp})}_{VGG-16\ (StyleGAN2ada)}$$

Tero Karras et al. “Training generative adversarial networks with limited data”. [2020]

PicsArt