Deep learning applications for the Face Restoration task

Anton Yatsenko
Faculty of Computer Science
National Research University Higher School of Economics
aiyatsenko@edu.hse.ru

Diffusion is a powerful image processing tool that allows you to improve the quality and restore details in images. The aim of the study is to determine effective approaches to the use of diffusion and other methods to improve the quality of facial reconstruction in images. The results obtained can be useful for developing new image processing methods and improving facial image restoration technologies.

I. INTRODUCTION

Our research focuses on the automatic quality improvement of a photo. This and similar problems are traditionally solved by generative methods such as GAN or diffusion models. The solution to this problem is to collect the most important picture features and after that generate a new picture that is very similar to the original image but with high-quality resolution. To solve this problem, we investigate the existing methods, analyze their advantages and disadvantages, and then try to combine all of them to create a high-quality and fast method for hair editing. Our research is based on several current SOTA approaches GFP-GAN and DDMs.

II. LITERATURE REVIEW

Before we move on to the articles, we need to explain what diffusion models and GANs are. Diffusion model uses Markov's chain to slowly add noise to image in the forward way and then learns how to remove this noise correctly on the backward way. By doing that on a lot of data we can generate completely new pictures with high quality detalization. GANs on the other hand, have different approach to such problem. Generative Adversarial Networks consist of two networks: Discriminator and Generator. Discriminator is a classifier that identifies real data from the fake data created by Generator. Generator learns to create fake images by incorporating feedback from the discriminator. GANs are much faster than diffusion models, but diffusion models can achieve better quality.

A. GFP-GAN

This network consist of 2 modules: degradation removal module and pre-trained face GAN. This modules are connected by a direct code mapping and several Channel-Split Spatial Feature Transform layers to get spatial feature transform on one split, and the rest are kept as identity. Degradation removal module is a U-Net network which implements L1 loss between the restored images at each resolution scale. We took F-latent vector which has the most important features and pass it through dense layers to the pre-trained GANs. That's why we will not get something weird at the end because we used vector has information about original image. Then the authors use spatial features generated by a U-Net model to modulate the features produced by a GANs. Preserving spatial information from the input images is important for maintaining fidelity in face restoration, as it allows for the preservation of local characteristics and adaptive restoration at different locations on the face. To achieve this, they employ a Spatial Feature Transform technique, which generates affine transformation parameters for spatial-wise feature modulation. The learning objective of training GFP-GAN consists of: 1) reconstruction loss that constraints the outputs y^ close to the ground-truth y, 2) adversarial loss 3) proposed facial component loss to further enhance facial details, and 4) identity preserving loss.
Reconstruction loss has this formula:

$$\mathcal{L}_{rec} = \lambda_{l1}\|\hat{\boldsymbol{y}} - \boldsymbol{y}\|_1 + \lambda_{\text{per}}\,\|\phi(\hat{\boldsymbol{y}}) - \phi(\boldsymbol{y})\|_1$$

where ϕ is the pretrained VGG-19 network.
Adversarial loss is used to favor the solutions in the natural image manifold and generate realistic textures:

$$\mathcal{L}_{adv} = -\lambda_{adv}\mathbb{E}_{\hat{\boldsymbol{y}}}\,\text{softplus}(D(\hat{\boldsymbol{y}}))$$

where D denotes the discriminator and λ_{adv} represents the adversarial loss weight.

Proposed facial component loss:
For each region of a face, a small discriminator decides whether the restored patches are real. Interesting thing presented was an incorporation of a feature style loss based on the learned discriminators. Different from previous feature matching loss with spatial-wise constraints the feature style loss attempts to match the Gram matrix statistics of real and restored patches. Gram matrix calculates the feature correlations and usually effectively captures texture information.

$$\begin{aligned}\mathcal{L}_{\text{comp}} = &\sum_{\text{ROI}} \lambda_{\text{local}}\,\mathbb{E}_{\hat{\boldsymbol{y}}_{\text{ROI}}}\left[\log\left(1 - D_{\text{ROI}}\left(\hat{\boldsymbol{y}}_{\text{ROI}}\right)\right)\right] + \\ &\lambda_{fs}\left\|\text{Gram}\left(\psi\left(\hat{\boldsymbol{y}}_{\text{ROI}}\right)\right) - \text{Gram}\left(\psi\left(\boldsymbol{y}_{\text{ROI}}\right)\right)\right\|_1\end{aligned}$$

where ROI is region of interest from the component collection { left-eye, right-eye, mouth }.D_{ROI} is the

local discriminator for each region. ψ denotes the multi-resolution features from the learned discriminators. λ_{local} and λ_{fs} represent the loss weights of local discriminative loss and feature style loss, respectively.

Identity Preserving Loss
The identity preserving loss enforces the restored result to have a small distance with the ground truth in the compact deep feature space:

$$\mathcal{L}_{id} = \lambda_{id} \|\eta(\hat{\boldsymbol{y}}) - \eta(\boldsymbol{y})\|_1$$

where η represents face feature extractor, i.e. ArcFace in author's implementation. λ_{id} denotes the weight of identity preserving loss.
At the end this model gets the best LPIPS metric, which shows how simillar input image and result. Also it achieves great accuracy in the smallest details such as hair, pupils, and teeth in comparison with other models.

B. IDM

The main problems of face restoration task arise from data corruption. Because of low quality images, previous model designs are unsatisfactory in balancing degradation removal and detail refinement. So authors wanted to solve this problem by creating such DDM which will automatically clean the data without expensive labor annotation and pave the path for authentic restoration and accurate evaluation. Authors idea is to upgrade diffusion model with 2 separate processes which are called Intrinsic Iterative Learning and Extrinsic Iterative Learning.

Intrinsic Iterative Learning
DDMs are designed to learn a data distribution $q(x_0)$ by defining a chain of latent variable models $p_\theta(x_{0:T}) = p(x_T) \Pi_{t=1}^{T} p_\theta(x_{t-1} \mid x_t)$,

$$x_T \to x_{T-1} \to \ldots \to x_1 \to x_0$$

where each timestep's example x_t has the same dimensionality $x_t \in \mathbb{R}^D$. Usually, the chain starts from a standard Gaussian distribution $x_T \sim \mathcal{N}\left(0, I^D\right)$ and only the final sample x_0 is stored.
The forward diffusion process that can be simplified into a specification of the true posterior distribution

$$q(x_t \mid x_0) = \mathcal{N}\left(x_t \mid \sqrt{\gamma_t} x_0, (1-\gamma_t) I\right)$$

where γ_t defines the noise schedule. We thus learn the reverse chain with a "denoiser" model f_θ which takes both source image x_d and a intermediate noisy target image x_t by comparing it $p_\theta(x_{t-1} \mid x_t, x_d)$ with the tractable posterior with $q(x_{t-1} \mid x_t, x_0)$. The aim to optimize the following objective that estimates the noise eps and authors modify the default DDM loss:

$$\mathcal{L} = \mathbb{E}_{(x,x_d)} \mathbb{E}_\gamma \| f_\theta\left(x_d, \hat{x}, \gamma\right)\big) - x_0\big) \|_p^p$$

This loss checks if model has learned to detect generated noise correctly. The formulation is more efficient in both training and inference since the sequence of network approximates x0 at each time step starting from different amount of noises. As for the reverse diffusion process during the inference stage, model follows the Langevin dynamics,

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{1-\alpha_t}{\sqrt{1-\gamma_t}} f_\theta\left(x_t, x_d, \gamma_t\right) \right) + \sqrt{1-\alpha_t} \epsilon_t$$

where α_t is hyper-parameter related to $\gamma_t = \Pi_{i=1}^{t} \alpha_i$. Such process iteratively upgrade input image to high quiality.

Extrinsic Iterative Learning
The quality of dataset proven to be a key factor in final results of the model. To avoid this problem authors propose new solution. Thanks to the capability of authentic restoration of DDMs, the solution can automatically restore the training data without damaging the data, especially for those whose quality are already high and facial details are ineligible.
After learning the restoration model f_θ, we can apply it to all the training data to produce a new high-quality image domain P_{X^*},

$$x^* = f_\theta(x), \quad x \sim P_X.$$

They term the method as extrinsic iterative learning because it produces a higher quality data for another iteration of restoration model training, which is different from the internal chained process in DDMs.
Then, two DDMs combined together. It works because the only difference of learning f_θ and f_{θ^*} is the target distribution P_X and P_{X^*}. With proper tuning, the pool of P_X can be gradually replaced and filled by new data from P_{X^*}.
So the algorithm consist of intrinsic and extrinsic iterations. Intrinsic iterations are conditioned on the synthetically degraded samples from original data. The extrinsic iteration consists of intrinsic iterations of learning. The resulting model from the first intrinsic iteration is used to enhance the training data only which will then be used as the target data for the next intrinsic iteration.

Authors implementation is original an DDM with added attention layers in lower resolution (32×32 to 8×8) blocks.
As the results, this method consistently outperforms GFP-GAN, RestoreFormer and CodeFormer as baselines across PSNR, SSIM, LPIPS, Arcface identity score metrics.

III. MODEL TO RUN

I checked GFP-GAN because it is considered to be State of Art model. I used FFHQ as training and Celeba-Hq as test datasets because this datasets are

presented in both articles. The learning rate was set to 2×10^{-3}. Methods were taken from the article: LPIPS, FID, NIQE. I got this scores:

Methods	LPIPS ↓	FID ↓	NIQE ↓
GFP-GAN (article)	0.3646	42.62	4.077
GFP-GAN (mine)	**0.3684**	**43.92**	**5.023**

The results are very similar. Visually, images get better quality in the article.

IV. CONCLUSION

GFP-GAN can successfully improve image quality. However, the IDM article promises better results, so in the next step of the project it is reasonable to implement code of the model and compare results with it.