Blind Drifting: Diffusion models with a linear SDE drift term for blind image restoration tasks

Simon Welker[*,†], **Henry N. Chapman**[†], **Timo Gerkmann**[*]
[*] Signal Processing (SP), Universität Hamburg, Germany,
[†] Center for Free-Electron Lasers (CFEL), DESY, Hamburg, Germany
simon.welker@uni-hamburg.de, henry.chapman@cfel.de, timo.gerkmann@uni-hamburg.de

Abstract

In this work, we utilize the high-fidelity generation abilities of diffusion models to solve blind image restoration tasks, using JPEG artifact removal at high compression levels as an example. We propose a simple modification of the forward stochastic differential equation (SDE) of diffusion models to adapt them to such tasks. Comparing our approach against a regression baseline with the same network architecture, we show that our approach can escape the baseline's tendency to generate blurry images and recovers the distribution of clean images significantly more faithfully, while also only requiring a dataset of clean/corrupted image pairs and no knowledge about the corruption operation. By utilizing the idea that the distributions of clean and corrupted images are much closer to each other than to a Gaussian prior, our approach requires only low levels of added noise, and thus needs comparatively few sampling steps even without further optimizations.

1 Introduction

Diffusion models have taken the world of machine learning by storm due to their unprecedented ability to generate high-fidelity images and great flexibility to condition on a variety of user inputs. Previous works on diffusion models have largely concentrated on unconditional and conditional image generation. Recently, Bansal et al. [1] and Daras et al. [2] have proposed to extend and improve diffusion models by adding known deterministic corruptions, but have only evaluated their ideas for unconditional generation tasks, rather than for faithfully inverting corruptions to restore plausible original images. Here, we try to go a different route, and explicitly modify and train diffusion models to restore plausible images from corrupted ones.

In contrast to other recent works [2, 3], our approach does not require the underlying corruption operator to be known, linear, nor differentiable, and instead requires only a dataset of (clean image, corrupted image) pairs. We also propose an alternative way of combining a deterministic corruption with noise. In this work, we consider JPEG compression with low quality levels (10–20 %), as an example corruption with a nonlinear and nondifferentiable corruption operator that is unknown to the restoration procedure at inference time. We propose a simple modification of the forward stochastic differential equation (SDE) of diffusion models to adapt them to such tasks, building upon previous work from the speech processing literature [4, 5] but extending their modification to a more general form and newly applying these ideas to a nonlinear inverse problem in the image domain.

DLDE Workshop in the 36th Conference on Neural Information Processing Systems (NeurIPS 2022).

2 Methods

2.1 A family of task-adapted linear SDEs

Following [6], the forward process of a diffusion model can be interpreted as a dynamical system following a stochastic differential equation

$$\mathrm{d}\mathbf{x}_t = f(\mathbf{x}_t, t)\mathrm{d}t + g(\mathbf{x}_t, t)\mathrm{d}\mathbf{w} \tag{1}$$

where in this work, $\mathbf{x} \in \mathbb{R}^{C\times H\times W}$ is the current image and $\mathbf{w}$ is a standard Wiener process of the same dimensionality as $\mathbf{x}$, and the process runs forward from $t = t_\varepsilon$ until $t = T := 1$, with $t_\varepsilon \gtrapprox 0$ for numerical reasons [6]. Each image in the training dataset then represents the initial value $\mathbf{x}_0$ of a particular realization of this SDE. Song et al. [6] showed that previous diffusion models in the discrete-time domain can be interpreted to follow either the so-called *Variance Exploding (VE) SDE* or the so-called *Variance Preserving (VP) SDE*. Both SDEs have the aim of progressively turning images into Gaussian white noise, thereby turning the intractable image distribution into a tractable prior. To generate images, one then samples from this prior and numerically solves the corresponding *reverse SDE* [7],

$$\mathrm{d}\mathbf{x}_t = [-f(\mathbf{x}_t, t) + g(t)^2\nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t, t)]\mathrm{d}t + g(\mathbf{x}_t, t)\mathrm{d}\bar{\mathbf{w}} \tag{2}$$

where the only unknown term is the *score* $\nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t, t)$. A deep neural network called a *score network* $S_\theta(\mathbf{x}_t, t)$ is then trained to estimate this score, given the current process state $\mathbf{x}$ and time t.

Rather than turning the clean image distribution into pure noise, here we aim to turn the clean image distribution into a noisy version of the corrupted image distribution. This has two purposes:

1. Instead of pure noise, this uses the corrupted image (plus tractable noise) as the initial value of the reverse SDE, thus achieving the task adaptation through the formulation of the process itself, as opposed to only providing the corrupted image as conditioning information.
2. Since the added Gaussian noise is white, it functions as a continual source of all possible spatial frequencies throughout the reverse process. The trained score model then filters these frequencies appropriately, generating plausible clean image estimates without a loss of high-frequency detail.

Note that the distribution of noisy corrupted images is still "tractable" for the purposes of the restoration task, as a sample from it can be drawn by taking a corrupted image and adding Gaussian noise. We now propose the following family of linear forward SDEs to realize our idea:

$$f(\mathbf{x}_t, t) = \gamma t^\alpha(\mathbf{y} - \mathbf{x}_t)\,, \quad g(t) = \nu\left(\frac{\sigma_{\max}}{\sigma_{\min}}\right)^{2t}\,, \tag{3}$$

where $\mathbf{y}$ is the corrupted image corresponding to $\mathbf{x}_0$, γ is a stiffness hyperparameter controlling how strongly $\mathbf{x}_t$ is pulled towards $\mathbf{y}$, and $\alpha \in \mathbb{R}_{\geq 0}$ controls the shape of the curve pulling $\mathbf{x}_t$ towards $\mathbf{y}$. ν is a normalization factor determined to ensure that $\sigma_T \approx \sigma_{\max}$, where σ_t is the closed-form variance of the Gaussian process described by (3). Intuitively, our family of SDEs combines the diffusion g of the VE SDE with an added drift term f that pulls $\mathbf{x}_t$ towards the corrupted image $\mathbf{y}$.

2.2 The two considered SDEs

In this work, we will only consider $\alpha \in \{0, 1\}$ for simplicity. The case of $\alpha = 0$ was previously proposed for similar tasks in the speech processing literature [4], and we refer to it as the *Ornstein-Uhlenbeck Variance Exploding* (OUVE) SDE. We also newly propose using $\alpha = 1$, which we call the *t-squared Decay Variance Exploding* (TSDVE) SDE. To allow for efficient forward sampling in order to perform denoising score matching [6], we determine closed-form expressions of the mean $\boldsymbol{\mu}_t$ and variance σ_t of the Gaussian processes described by each SDE, see for instance [8]. Since the SDEs are linear in the state $\mathbf{x}_t$, the means of the two SDEs follow

$$\boldsymbol{\mu}_t^{\text{OUVE}} = e^{-\gamma t}\mathbf{x}_0 + (1 - e^{-\gamma t})\mathbf{y}\,, \qquad \boldsymbol{\mu}_t^{\text{TSDVE}} = e^{-\gamma\frac{t^2}{2}}\mathbf{x}_0 + (1 - e^{-\gamma\frac{t^2}{2}})\mathbf{y}\,. \tag{4}$$

Both expressions describe a linear interpolation between $\mathbf{x}_0$ and $\mathbf{y}$, with the interpolation parameter controlled by an exponential (OUVE) or half-Gaussian-shaped (TSDVE) decay over time. We relegate the somewhat involved closed-form expressions for the variance to Appendix A.

For both SDEs, $\boldsymbol{\mu}_t \neq \mathbf{y}$ for all finite t, which may seem like an issue since the aim was to have the process move towards $\mathbf{y}$. However, letting $\mathbf{z} \sim \mathcal{N}(0, I)$, it is only required that the distributions of $(\boldsymbol{\mu}_T + \sigma_T \mathbf{z})$ and $(\mathbf{y} + \sigma_T \mathbf{z})$ are similar, so that the latter can function as a plausible initial value for the reverse sampling process. We can control how well the distributions of these two expressions match, either by increasing the stiffness γ at the cost of potentially destabilizing the reverse process, or by increasing σ_{max} to further smooth the density functions of both distributions at the cost of more reverse iterations. Here, we choose a set of parameters that empirically work well, and leave further optimization of them to future work.

2.3 Dataset

For $\mathbf{x}_0$, we use the CelebA-HQ dataset [9], resized to 256x256 and split into 24000 images for training, 1500 for validation and 4500 for testing. To generate $\mathbf{y}$, we first randomly sample a JPEG quality value from 0 to 30 for each image $\mathbf{x}_0$ and training iteration, and then apply JPEG compression to $\mathbf{x}_0$. During evaluation, we set the JPEG quality values to a constant value across all compared images and models (but do not provide this quality value to any model).

2.4 Network training and process parameterization

We utilize the NCSN++ architecture [6], both for the regression baseline and as a score network. We train the regression baseline R_θ to recover $\mathbf{x}_0$ given $\mathbf{y}$ via a simple L_2 loss:

$$\theta^* = \arg\min_\theta \mathbb{E}_{(\mathbf{x}_0, \mathbf{y})} \left[\| R_\theta(\mathbf{y}) - \mathbf{x}_0 \|_2^2 \right] , \tag{5}$$

where we pass a constant "dummy" value of $t = 1$ to the time embedding layers of NCSN++ to avoid making any changes to the DNN that may affect the qualitative behavior of each layer. To train the score models S_θ, we use the idea that diffusion models based on such linear SDEs can still be trained by the same denoising score matching target as in [6]:

$$\theta^* = \arg\min_\theta \mathbb{E}_{t,(\mathbf{x}_0,\mathbf{y}),\mathbf{x}_t|(\mathbf{x}_0,\mathbf{y},t)} \left[\| S_\theta(\mathbf{x}_t, \mathbf{y}, t) + \mathbf{z} \|_2^2 \right] , \quad \mathbf{x}_t = \boldsymbol{\mu}_t + \sigma_t \mathbf{z} \tag{6}$$

with $\mathbf{z} \sim \mathcal{N}(0, I)$, and image pairs $(\mathbf{x}_0, \mathbf{y})$ sampled from the dataset. The only changes in this objective are that the expectation is also calculated over $\mathbf{y}$, and that $\mathbf{y}$ is provided to the score network. We provide $\mathbf{y}$ as an input to S_θ by concatenating $\mathbf{x}_t$ and $\mathbf{y}$ along the channel dimension, for which we change the number of input channels of NCSN++ from 3 to 6. For training, we use the *AdamW* optimizer [10], and set the hyperparameters of the process and training as listed in Table 1. For evaluation, we use each method's checkpoint with minimum loss on the validation set. In line with the diffusion model literature [6], we update an exponential moving average (decay of 0.999) of all network parameters after each training step, and use these parameters for evaluation.

SDE	α	γ	σ_{max}	σ_{min}	t_ε	N
OUVE	0	1	0.3	0.01	0.01	100
TSDVE	1	2	0.3	0.01	0.01	100

(a) Process parameterization

Learning rate	0.0002
Batch size per GPU	6
Number of GPUs	2
Max. epochs	100

(b) Training hyperparameters

Table 1: Parameters for the SDEs (a) and network training (b)

3 Results and Discussion

We generate reconstructions from our diffusion models with the Euler-Maruyama sampler discretized to $N = 100$ steps, and retrieve reconstructions from the regression baseline via a single pass. In Figure 1, we compare the regression baseline against our SDE-based diffusion models for JPEG quality level 10. Qualitatively, the baseline reconstructs major features well, but fails to produce plausible high-frequency details. This is particularly visible for hair, facial hair and skin textures, and results in a painting-like look. In contrast, our approach subjectively results much more

Figure 1: Example images for JPEG quality level 10, comparing the regression baseline and both proposed SDEs (OUVE, TSDVE) against the ground truth and corrupted images. The baseline reconstructions exhibit a blurry, painting-like quality (best visible when zoomed in).

natural-looking images and, for both proposed SDEs, shows a remarkable ability to reconstruct natural skin and hair textures that are plausible given the corrupted image. Further example images can be found in Appendix B. Both SDEs generally result in perceptually very similar images.

In Table 2, we compare the distributions of the corrupted and reconstructed images of each method against the distribution of the ground-truth images, using the metrics FID [11] and KID [12]. We also list the average SSIM [13] and LPIPS [14] values, and evaluate all metrics on our test set of 4500 images. Judging from the distribution-based metrics, both of our diffusion-based approaches model the clean image distribution of CelebA-HQ significantly more faithfully than the baseline, which even performs worse than the corrupted (compressed) images in this regard. On the other hand, the baseline achieves a significant SSIM improvement whereas our proposed models do not. This is to be somewhat expected due to the generative nature of our approach. We argue that SSIM does not match human perception well here: the blurry look of the baseline images is clearly visible, but does not seem to be strongly penalized. Indeed, for the perceptual LPIPS metric our methods consistently outperform the baseline. Both proposed SDEs perform very similarly in all regards.

	KID	FID	LPIPS	SSIM
Corrupted	22.53	36.26	0.20	0.82
Baseline	38.18	45.92	0.13	**0.90**
TSDVE	**2.32**	15.72	**0.08**	0.83
OUVE	2.37	**15.69**	**0.08**	0.83

(a) JPEG quality level 10

	KID	FID	LPIPS	SSIM
Corrupted	8.63	21.17	0.08	0.90
Baseline	27.34	35.71	0.08	**0.94**
TSDVE	0.59	12.99	**0.05**	0.89
OUVE	**0.57**	**12.97**	**0.05**	0.89

(b) JPEG quality level 20

Table 2: Distribution-based metrics (KID [12], FID [11]), average SSIM [13] and LPIPS [14], comparing corrupted and reconstructed test images to the ground truth for two JPEG quality levels. KID scores are multiplied by 1000 for readability. Lower is better for all metrics except SSIM. Best values are listed in bold.

4 Conclusion

Based on work from the speech processing literature, we propose a simple change to SDE-based diffusion models to adapt them to image restoration tasks. Our approach does not require the corruption operator to be available in closed form and does not impose strong restrictions on its nature, but only requires a dataset of paired images. Compared to a regression baseline using the same architecture, our approach restores images of perceptually higher quality and models the ground-truth image distribution significantly more faithfully. While currently requiring 100x more DNN passes than the baseline, we expect that recent progress on efficient sampling for diffusion models will allow this number to drastically decrease and make the approach competitive in terms of runtime.

Acknowledgments and Disclosure of Funding

We acknowledge the support by DASHH (Data Science in Hamburg - HELMHOLTZ Graduate School for the Structure of Matter) with the Grant-No. HIDSS-0002.

References

[1] A. Bansal, E. Borgnia, H.-M. Chu, J. S. Li, H. Kazemi, F. Huang, M. Goldblum, J. Geiping, and T. Goldstein, "Cold diffusion: Inverting arbitrary image transforms without noise," *arXiv preprint arXiv:2208.09392*, 2022.

[2] G. Daras, M. Delbracio, H. Talebi, A. G. Dimakis, and P. Milanfar, "Soft diffusion: Score matching for general corruptions," *arXiv preprint arXiv:2209.05442*, 2022.

[3] B. Kawar, M. Elad, S. Ermon, and J. Song, "Denoising diffusion restoration models," in *ICLR Workshop on Deep Generative Models for Highly Structured Data*, 2022.

[4] S. Welker, J. Richter, and T. Gerkmann, "Speech enhancement with score-based generative models in the complex STFT domain," *ISCA Interspeech*, 2022.

[5] J. Richter, S. Welker, J.-M. Lemercier, B. Lay, and T. Gerkmann, "Speech enhancement and dereverberation with diffusion-based generative models," *arXiv preprint arXiv:2208.05830*, 2022.

[6] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," *Int. Conf. on Learning Representations (ICLR)*, 2021.

[7] B. D. Anderson, "Reverse-time diffusion equation models," *Stochastic Processes and their Applications*, vol. 12, no. 3, pp. 313–326, 1982.

[8] S. Särkkä and A. Solin, *Applied Stochastic Differential Equations*. Cambridge University Press, 2019.

[9] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," in *Int. Conf. on Learning Representations (ICLR)*, 2018.

[10] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Int. Conf. on Learning Representations (ICLR)*, 2019.

[11] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local Nash equilibrium," in *Advances in Neural Inf. Proc. Systems (NeurIPS)*, vol. 30, 2017.

[12] M. Bińkowski, D. J. Sutherland, M. Arbel, and A. Gretton, "Demystifying MMD GANs," in *Int. Conf. on Learning Representations (ICLR)*, 2018.

[13] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. on Image Proc.*, vol. 13, no. 4, pp. 600–612, 2004.

[14] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *CVPR*, 2018.

A Closed-form variance solutions of the OUVE and TSDVE SDEs

In the following, we provide the expressions for the diffusion normalization factors ν and the solved variance of the Gaussian process, for the OUVE SDE and the TSDVE SDE. We utilized the software Mathematica 12.1 to solve the mean and variance ODEs [8] for the initial value $\sigma_0 = 0$. This choice is in contrast to [6], where the initial value $\sigma_0 = \sigma_{\text{min}}$ was used. Our reasoning for this change is that due to the influence of our added drift terms, the choice by Song et al. may result in nonmonotonous functions σ_t, particularly depending on the choice of γ and the relative scale of σ_{max} in comparison to σ_{min}. We admit that the convenient name of σ_{min} may be misleading under this assumption, since it is not a "minimum sigma" under our assumptions. Nonetheless, we keep this parameter as a way to control the shape of the variance curve, and also keep its name in line with the previous literature.

For the OUVE SDE, we follow [6] to determine an approximate normalization factor:

$$\nu_{\text{OUVE}} = \sqrt{2\left(\gamma + \log\left(\frac{\sigma_{\text{max}}}{\sigma_{\text{min}}}\right)\right)}, \tag{7}$$

resulting in the variance

$$\sigma^2_{t,\text{OUVE}} = \sigma^2_{\text{min}}\left(\left(\frac{\sigma_{\text{max}}}{\sigma_{\text{min}}}\right)^{2t} - e^{-2\gamma t}\right). \tag{8}$$

One may observe that the condition $\sigma_1 = \sigma_{\text{max}}$ is only approximately fulfilled here, but the error is small when σ_{min} is small and γ is reasonably large: for our parameter choice, it is equal to

$$0.01^2(-e^{-2\cdot 1\cdot 1}) \approx -1.4 \times 10^{-5} \tag{9}$$

and we therefore use this factor ν_{OUVE} due to its relative simplicity.

For the TSDVE SDE, we first solved for the unnormalized variance expression using $\nu = 1$:

$$\sigma^2_{t,\text{TSDVE,unnorm}} = \frac{\sigma^2_{\text{min}}}{\sqrt{\gamma}} e^{-\gamma t^2}\left(e^{\gamma t^2 + 2t\log\left(\frac{\sigma_{\text{max}}}{\sigma_{\text{min}}}\right)} D\left(\frac{t\gamma + \log\left(\frac{\sigma_{\text{max}}}{\sigma_{\text{min}}}\right)}{\sqrt{\gamma}}\right) - D\left(\frac{\log\left(\frac{\sigma_{\text{max}}}{\sigma_{\text{min}}}\right)}{\sqrt{\gamma}}\right)\right) \tag{10}$$

where D is the Dawson function. Solving for ν such that $\nu^2\sigma^2_T = \sigma_{\text{max}}$ exactly, we retrieve

$$\nu_{\text{TSDVE}} = \sqrt{\frac{\sigma^2_{\text{max}}\sqrt{\gamma}}{e^{-\gamma}\left(e^{\gamma}\sigma^2_{\text{max}} D\left(\frac{\gamma + \log\left(\frac{\sigma_{\text{max}}}{\sigma_{\text{min}}}\right)}{\sqrt{\gamma}}\right) - \sigma^2_{\text{min}} D\left(\frac{\log\left(\frac{\sigma_{\text{max}}}{\sigma_{\text{min}}}\right)}{\sqrt{\gamma}}\right)\right)}}. \tag{11}$$

We then multiply the diffusion term $g(t)$ of the forward SDE by this ν_{TSDVE}, finally resulting in

$$g(t) := \nu_{\text{TSDVE}}\left(\frac{\sigma_{\text{max}}}{\sigma_{\text{min}}}\right)^{2t} \tag{12}$$

$$\implies \sigma^2_{t,\text{TSDVE}} = \nu^2_{\text{TSDVE}} \cdot \sigma^2_{t,\text{TSDVE,unnorm}} \tag{13}$$

Note that we are now already dealing with unwieldy functions such as the Dawson function just by setting $\alpha = 1$ (TSDVE SDE). This was the principal reason for us to not investigate $\alpha = 2$ and higher, though exploring a solution or approximation of σ_t for these cases may be helpful: Increasing α keeps the process mean close to $\mathbf{x}_0$ for longer early in the forward process, which may help the reverse process to result in higher-quality reconstructions; however, in practice we did not find significant differences between $\alpha = 0$ and $\alpha = 1$ in this work.

B More example images

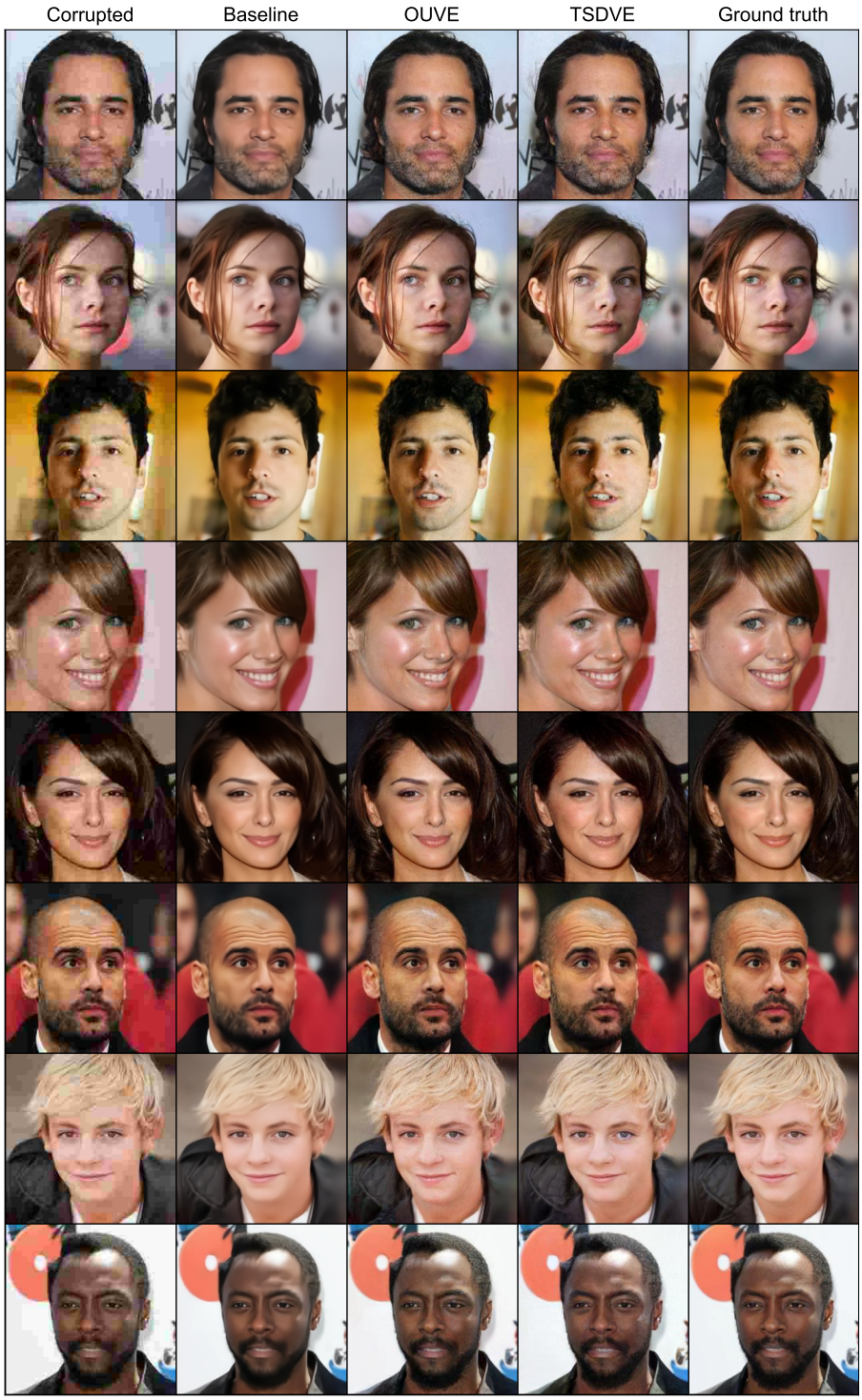

Figure 2: Further example images for the restoration task with JPEG quality level 10.

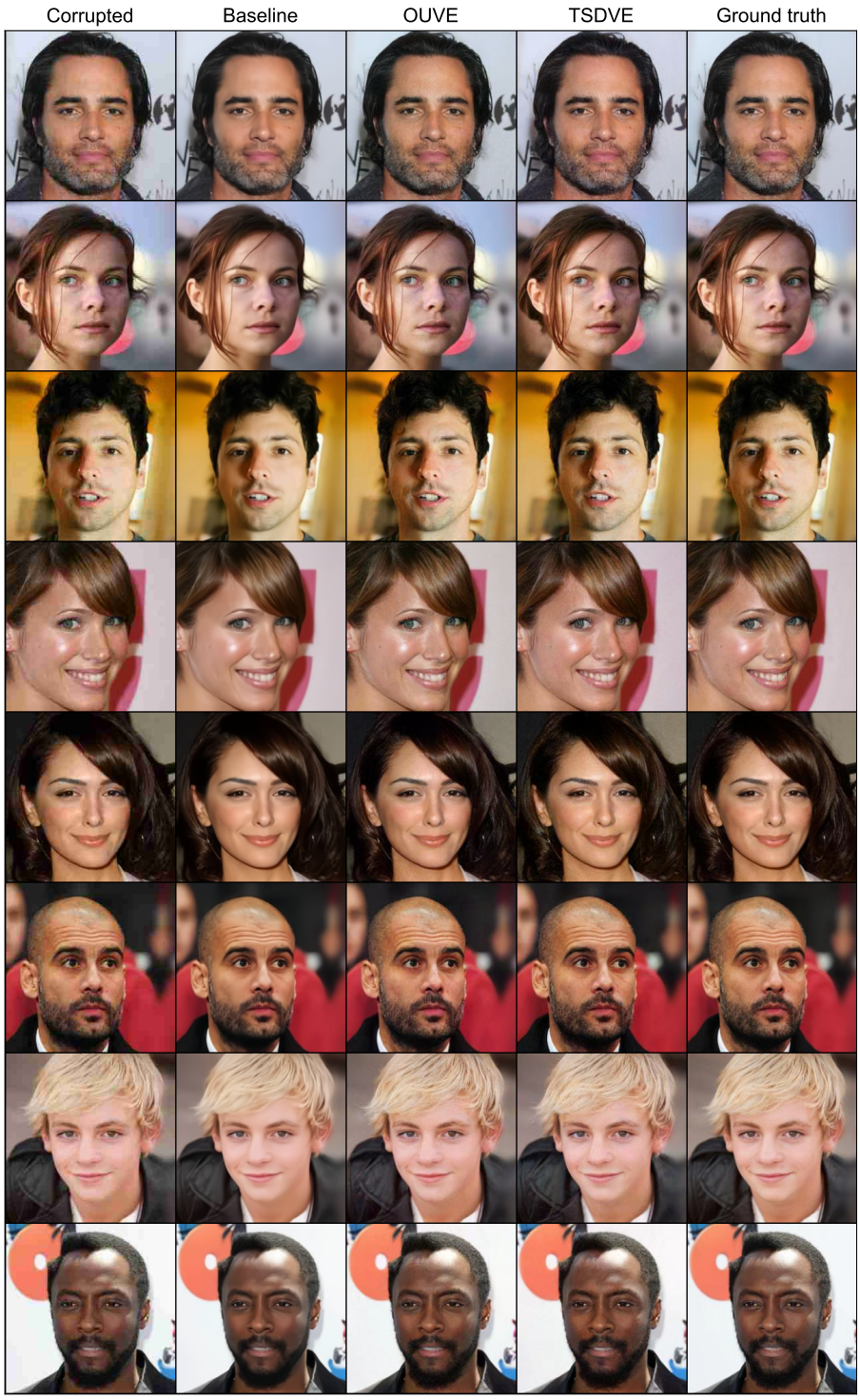

Figure 3: Further example images for the restoration task with JPEG quality level 20.