UGPNet: Universal Generative Prior for Image Restoration
– Supplementary Material –

Hwayoon Lee[1,*] Kyoungkook Kang[2] Hyeongmin Lee[2] Seung-Hwan Baek[2] Sunghyun Cho[2]

[1]GENGENAI
hwayoon.lee@gengen.ai

[2]POSTECH
{kkang831, hmin970922, shwbaek, s.cho}@postech.ac.kr

Figure 1. Qualitative comparison with other generative prior-based methods on out-of-distribution images. In all the examples, only UGPNet succeeds in robust image restoration without noticeable artifacts. The source images are collected from the internet[1].

In this supplementary material, we present:

- Detailed architecture of the restoration module,
- Detailed architecture of the synthesis module,
- Mathematical definitions of the losses,
- Implementation details,
- Discussion on the impact of the pretrained networks,
- Additional comparisons on the robustness to out-of-distribution images,
- Additional comparisons on denoising and deblurring of natural images, and
- Additional qualitative comparisons on denoising, deblurring, and super-resolution.

S.1. Architecture of Restoration Module

We first describe the detailed architecture of the restoration module, which consists of a structure encoder R_{se}, and a merging network R_{mg}. The structure encoder consists of one ConvBlock, and the merging network R_{mg} consists of two ConvBlocks and one ToImage layer. A ConvBlock is composed of three 3×3 convolution layers, each of which is followed by a Leaky ReLU activation layer. A ToImage

*This work was done at POSTECH.
[1] https://unsplash.com/

layer has a 3×3 convolution layer followed by a Leaky ReLU activation layer. For the convolution layers of the structure encoder, we use the same number of channels as the last layer feature map of regression network R. For the convolution layers of the merging network, we use 64 channels.

S.2. Architecture of Synthesis Module

Fig. 2 illustrates the network architecture of the synthesis module. Our encoder E takes a regressed image x_{reg} as input and estimates a latent code in the $\mathcal{F}/\mathcal{W}^+$ space [6], which is composed of an intermediate feature map $f_{16\times16}$ and 12 w vectors. Then, the generator G synthesizes a perceptually-realistic image from the estimated latent code.

Here, we present a detailed description of the encoder network. The main path of the encoder E estimates the latent code of spatial resolution 16×16 in the $\mathcal{F}$ space, and its architecture is shown in Tab. 1(a). In the table, each EncoderBlock consists of three ConvBlocks, and each ConvBlock has a 3×3 convolution layer, a Batch Normalization layer, and a Leaky ReLU activation layer. We use average pooling to halve the spatial resolution of intermediate feature maps.

The intermediate feature map after the fourth Average Pooling layer is passed to a single map2style network proposed by pSp [11]. After that, the feature map is passed to 12 fully-connected layers in order to estimate the latent code in the $\mathcal{W}^+$ space. As mentioned in the main paper, we use a single map2style network to estimate w vectors, which significantly reduces computational overhead compared to the original work [11] that estimates each w vector with the corresponding map2style network. Tab. 1(b) presents the architecture of the map2style network. In the table, each ConvBlock consists of a 3×3 convolution layer with stride 2 followed by a Leaky ReLU activation layer.

S.3. Mathematical Definitions of Loss Functions

To train the synthesis module, we use a weighted sum of $\mathcal{L}_1$, $\mathcal{L}_{per}$, and $\mathcal{L}_{adv}$. $\mathcal{L}_1(x)$ is defined as $\|x - x_{gt}\|_1$. $\mathcal{L}_{per}(x)$ is defined as $\|\phi(x) - \phi(x_{gt})\|^2$, where ϕ is an LPIPS [18] network. These reconstruction losses ($\mathcal{L}_1$, $\mathcal{L}_{per}$) encourage the synthesis module to reconstruct the image accurately. For the synthesis module to produce a realistic image, we employ an adversarial loss, $\mathcal{L}_{adv}(x)$. Specifically, we adopt the non-saturating loss of StyleGAN2 [8], which is defined as:

$$\mathcal{L}_{adv}(x) = -\mathbb{E}_x \left[\texttt{softplus}(D(x)) \right] \tag{1}$$

where D is a discriminator.

To train the fusion module, we use a weighted sum of $\mathcal{L}_1$, $\mathcal{L}_{per}$, and $\mathcal{L}_{cf}$. $\mathcal{L}_{cf}$ is a patch-wise contextual loss [10] between x_{syn} and $\hat{x}$. It maximizes the contextual similarity between images. The images x and y can be represented as collections of perceptual feature vectors $\{x_i\}$ and $\{y_j\}$, respectively, where i and j are feature indices. The contextual similarity between two feature points x_i and y_j is then

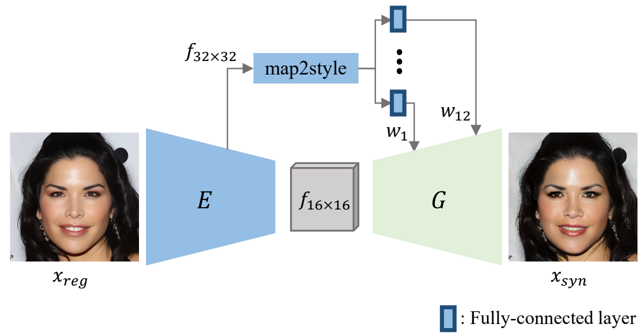


Figure 2. The synthesis module of UGPNet consists of an encoder E and a generator G. The encoder estimates the latent code of spatial resolution 16×16 in the $\mathcal{F}$ space in a feed-forward manner. Then, a single map2style network takes the intermediate feature map of size 32×32. The feature map is passed to additional 12 fully-connected layers after the map2style network, to estimate the latent code in $\mathcal{W}^+$.

I (3ch)
EncoderBlock (16ch)
Average Pooling
EncoderBlock (32ch)
Average Pooling
EncoderBlock (64ch)
Average Pooling
EncoderBlock (128ch)
Average Pooling
EncoderBlock (256ch)
Average Pooling
EncoderBlock (512ch)
1×1 Conv (512ch)

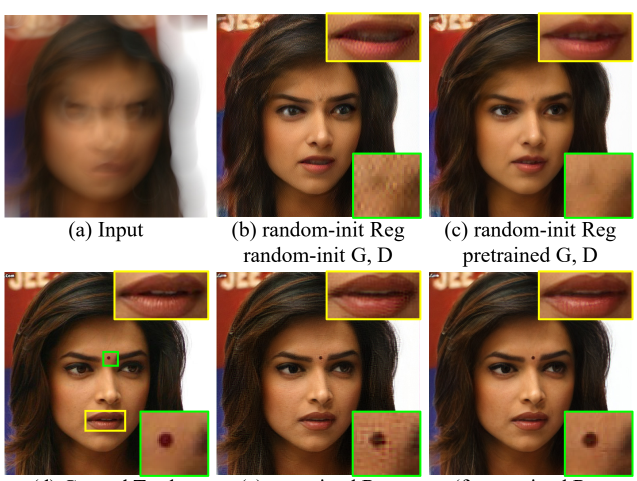
(a) Encoder E

Input (128ch)
ConvBlock (512ch)
ConvBlock (512ch)
ConvBlock (512ch)
ConvBlock (512ch)
ConvBlock (512ch)

(b) map2style network

Table 1. (a) Detailed architecture of our encoder E. Each EncoderBlock consists of three ConvBlocks. Each ConvBlock consists of a 3×3 convolution layer, a Batch Normalization layer, and a Leaky ReLU activation layer. The intermediate feature map after the fourth Average Pooling layer (marked as yellow) is passed to a single map2style network. (b) Detailed architecture of the map2style network. Each ConvBlock has a 3×3 convolution layer with stride 2 and a Leaky ReLU activation layer.

Figure 3. Qualitative comparison of denoising and deblurring on natural images [17].



(a) Input
(b) random-init Reg random-init G, D
(c) random-init Reg pretrained G, D
(d) Ground Truth
(e) pretrained Reg random-init G, D
(f) pretrained Reg pretrained G, D

Figure 4. We investigate the impact of the pretrained knowledge in the restoration and synthesis modules. Without the pretrained regression network ((b) and (c)), UGPNet fails to recover image structure (modified mouth and missing spot), and without the pretrained GAN model ((b) and (e)), UGPNet fails to generate realistic texture, resulting in artifacts.

defined as follows:

$$\mathrm{CX}_{ij} = \exp\left(\frac{1-\tilde{d}_{ij}}{h}\right) / \sum_k \exp\left(\frac{1-\tilde{d}_{ik}}{h}\right) \quad (2)$$

where $\tilde{d_{ij}}$ is the normalized cosine distance between feature points x_i and y_j, and h is a bandwidth parameter. Then, the contextual similarity between two images x, y can be defined as:

$$\mathrm{CX}(x,y) = \frac{1}{N}\sum_j \max_i \mathrm{CX}_{ij} \quad (3)$$

where N is the number of feature points. Finally, the contextual loss is defined as:

$$\mathcal{L}_{\mathrm{CX}}(x,y) = -\log\left(\mathrm{CX}\left(\phi(x),\phi(y)\right)\right) \quad (4)$$

where ϕ is the perceptual network for extracting perceptual features. In our implementation, the `relu3_4` layer of VGG19 [12] was used as ϕ. $\mathcal{L}_{cf}$ applies $\mathcal{L}_{\mathrm{CX}}$ between $\hat{x}$ and x_{syn} in a patch-wise manner to compensate for the potential misalignment.

S.4. Implementation Details

In all the training stages, we use a batch size of 8 and use the Adam optimizer [9] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. For the balancing weights in the loss functions, we use $\lambda_{per} = 10$, $\lambda_{adv} = 0.3$, and $\lambda_{cf} = 0.05$. The learning rate of 10^{-4} and the number of iterations of 20,000 are used for training the restoration module and we use the best model based on the PSNR score over 1,000 images of the CelebA-HQ dataset [7]. For the synthesis module, we set the learning rate to 10^{-4} for the encoder and generator, and 2.5×10^{-5} for the discriminator, and the number of iterations to 40,000. For the fusion module, we set the initial learning rate to 10^{-3} and reduce it by a factor of 0.1 at the 8,000th iteration. We use the best model based on the PSNR score during 40,000 iterations.

S.5. Exploiting Pretrained Networks

We investigate the impact of the knowledge learned in the pretrained networks within the restoration module and the synthesis module. Specifically, we compare all variations of UGPNet initialized with pretrained weights or with random weights in each module on deblurring, as shown in Fig. 4. Without pretrained weights of the regression network and the generative network, UGPNet has difficulty in restoring faithful structure and generating realistic textures.

S.6. Additional Comparisons of Restoration of Out-of-Distribution Images

As demonstrated in the main paper, UGPNet is robust against catastrophic failures for images outside the training distribution of the generative prior. Here, we present additional qualitative comparisons against state-of-the-art generative prior-based methods (GFP-GAN [13], GPEN [16], and VQFR [4]) in Fig. 1. In all the examples, only UGPNet recovers authentic image structures, whereas all the other models produce severe artifacts.

S.7. Additional Comparisons of Denoising and Deblurring on Natural Images

We train UGPNet on the LSUN-Church [17] dataset to validate its applicability on natural images. For denoising, we synthesize noisy images by adding Gaussian ($\mu = 0$, $\sigma = 0.3$) and Poisson noise ($k = 30$). For deblurring, we apply random motion blur sampled from 2,000 motion blur kernels of size 51×51. We provide additional qualitative comparisons on denoising and deblurring with NAFNet [2] trained on the same dataset in Fig. 3. As shown in the figure, UGPNet achieves more realistic high-frequency details compared to the state-of-the-art work.

S.8. Additional Qualitative Comparisons

We provide additional qualitative comparisons against recent learning-based algorithms in Fig. 5 and Fig. 6 for denoising, Fig. 7 and Fig. 8 for deblurring, and Fig. 9 and Fig. 10 for super-resolution. In the figures, UGPNet universally succeeds in high-quality image restoration for all the tasks. In the case of denoising and deblurring, UGPNet outperforms all the regression-based methods in terms of realistic high-frequency detail generation and all the generative prior-based methods in terms of faithful recovery of authentic image structures. In the case of super-resolution, UGPNet is superior to the regression-based methods and shows comparable performance to generative prior-based methods.

References

[1] Kelvin CK Chan, Xintao Wang, Xiangyu Xu, Jinwei Gu, and Chen Change Loy. Glean: Generative latent bank for large-factor image super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14245–14254, 2021. 11

[2] Liangyu Chen, Xiaojie Chu, Xiangyu Zhang, and Jian Sun. Simple baselines for image restoration. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VII*, pages 17–33. Springer, 2022. 4, 6, 8

[3] Liangyu Chen, Xin Lu, Jie Zhang, Xiaojie Chu, and Chengpeng Chen. Hinet: Half instance normalization network for image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 182–192, 2021. 6, 8

[4] Yuchao Gu, Xintao Wang, Liangbin Xie, Chao Dong, Gen Li, Ying Shan, and Ming-Ming Cheng. Vqfr: Blind face restoration with vector-quantized dictionary and parallel decoder. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVIII*, pages 126–143. Springer, 2022. 4, 7, 9, 11

[5] Jingwen He, Wu Shi, Kai Chen, Lean Fu, and Chao Dong. Gcfsr: a generative and controllable face super resolution method without facial and gan priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1889–1898, 2022. 11

[6] Kyoungkook Kang, Seongtae Kim, and Sunghyun Cho. Gan inversion for out-of-range images with geometric transformations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13941–13949, 2021. 2

[7] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. 4

[8] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020. 2

[9] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 3

[10] Roey Mechrez, Itamar Talmi, and Lihi Zelnik-Manor. The contextual loss for image transformation with non-aligned data. In *Proceedings of the European conference on computer vision (ECCV)*, pages 768–783, 2018. 2

[11] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2287–2296, 2021. 2

[12] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 3

[13] Xintao Wang, Yu Li, Honglun Zhang, and Ying Shan. Towards real-world blind face restoration with generative facial prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9168–9178, 2021. 4, 7, 9, 11

[14] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European conference on computer vision (ECCV) workshops*, pages 0–0, 2018. 10

[15] Zhendong Wang, Xiaodong Cun, Jianmin Bao, Wengang Zhou, Jianzhuang Liu, and Houqiang Li. Uformer: A general u-shaped transformer for image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17683–17693, 2022. 6, 8

[16] Tao Yang, Peiran Ren, Xuansong Xie, and Lei Zhang. Gan prior embedded network for blind face restoration in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 672–681, 2021. 4, 7, 9, 11

[17] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 3, 4

[18] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 2

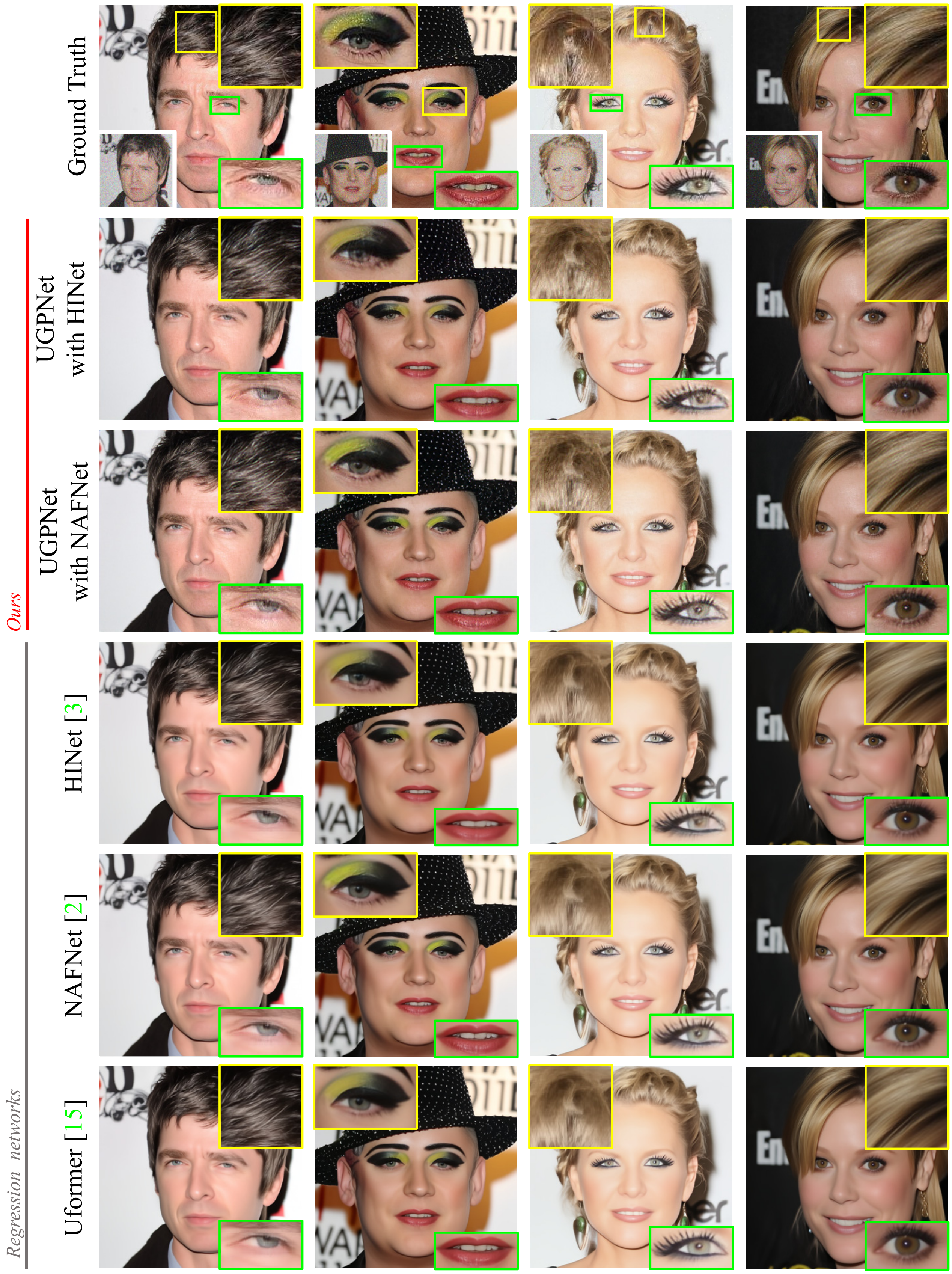


Figure 5. Qualitative comparison on denoising with recent regression-based methods including Uformer [15], NAFNet [2] and HINet [3]. The insets at the bottom of the ground-truth images are input degraded images.

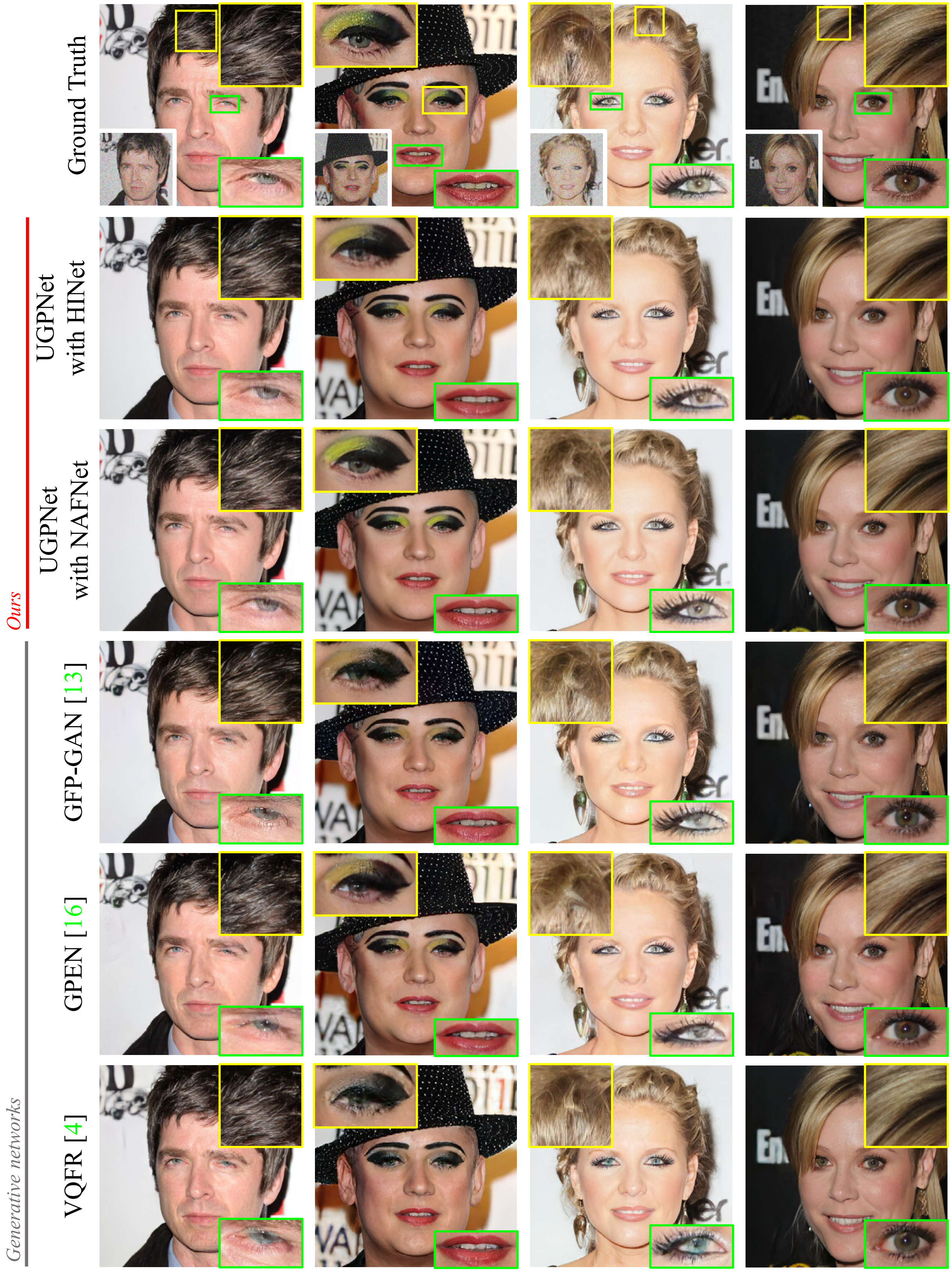


Figure 6. Qualitative comparison on denoising with recent generative prior-based methods including GFP-GAN [13], GPEN [16], and VQFR [4]. The insets at the bottom of the ground-truth images are input degraded images.

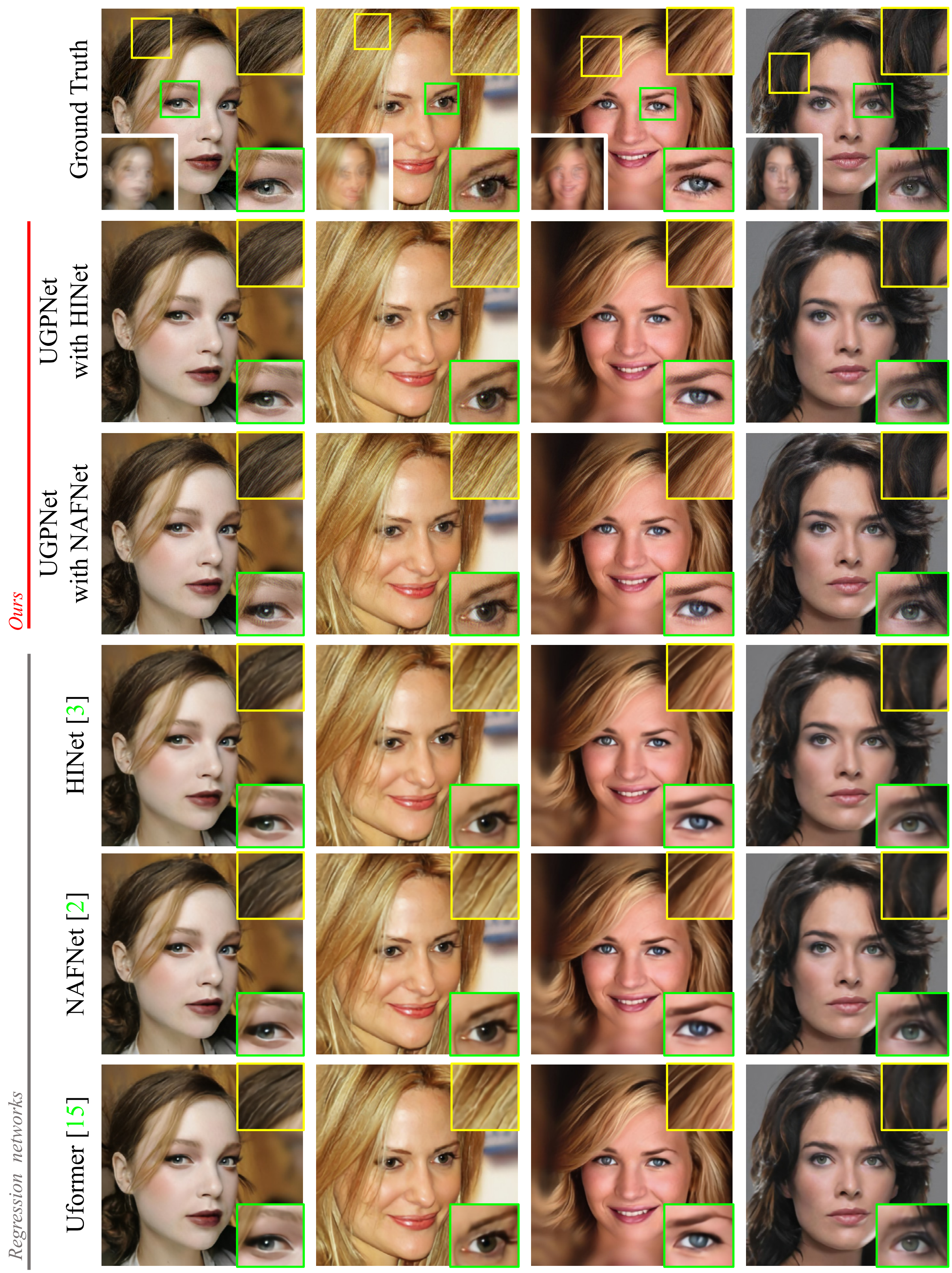


Figure 7. Qualitative comparison on deblurring with recent regression methods including Uformer [15], NAFNet [2] and HINet [3]. The insets at the bottom of the ground-truth images are input degraded images.

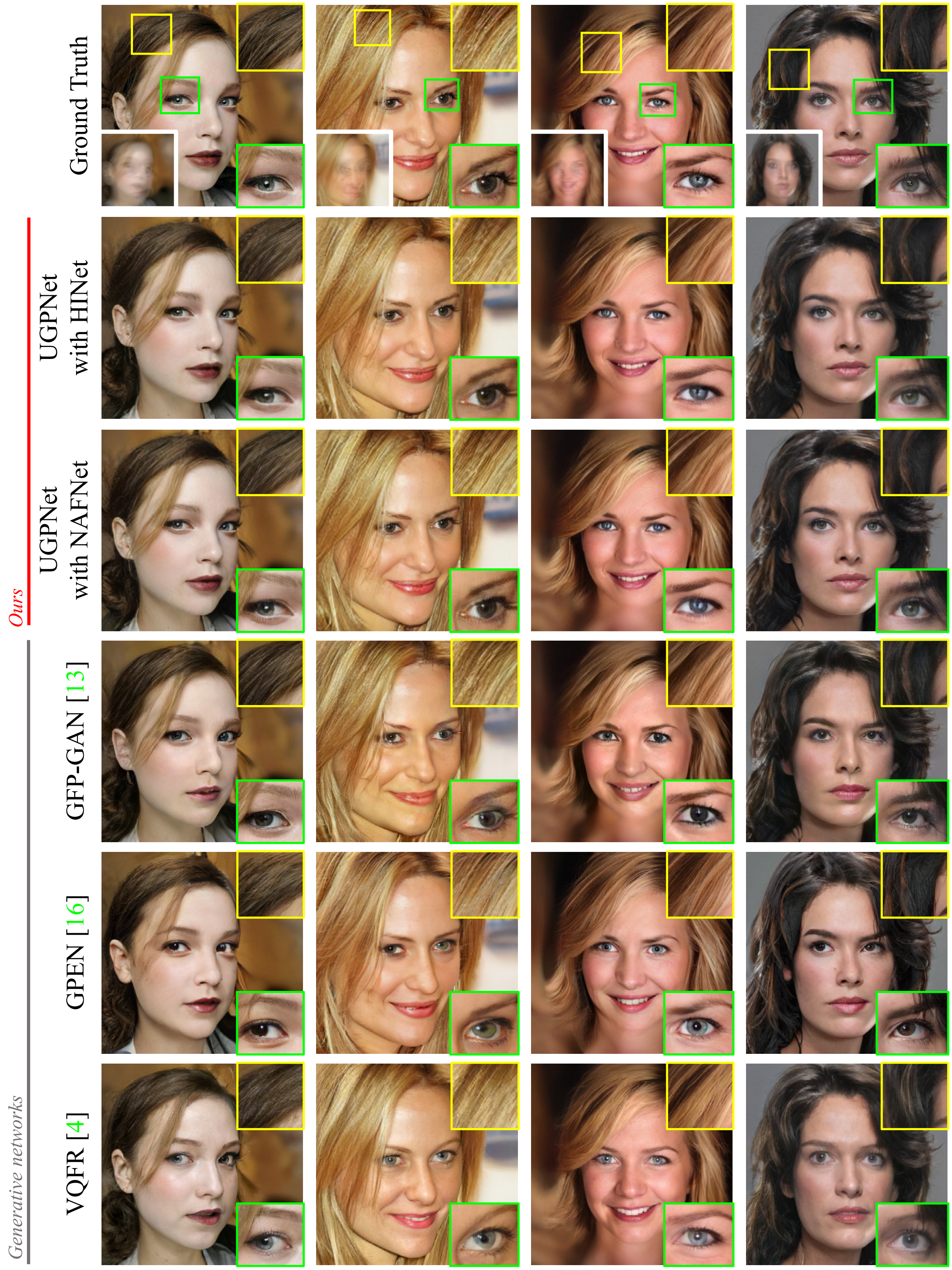


Figure 8. Qualitative comparison on deblurring with recent generative prior-based methods including GFP-GAN [13], GPEN [16], and VQFR [4]. The insets at the bottom of the ground-truth images are input degraded images.

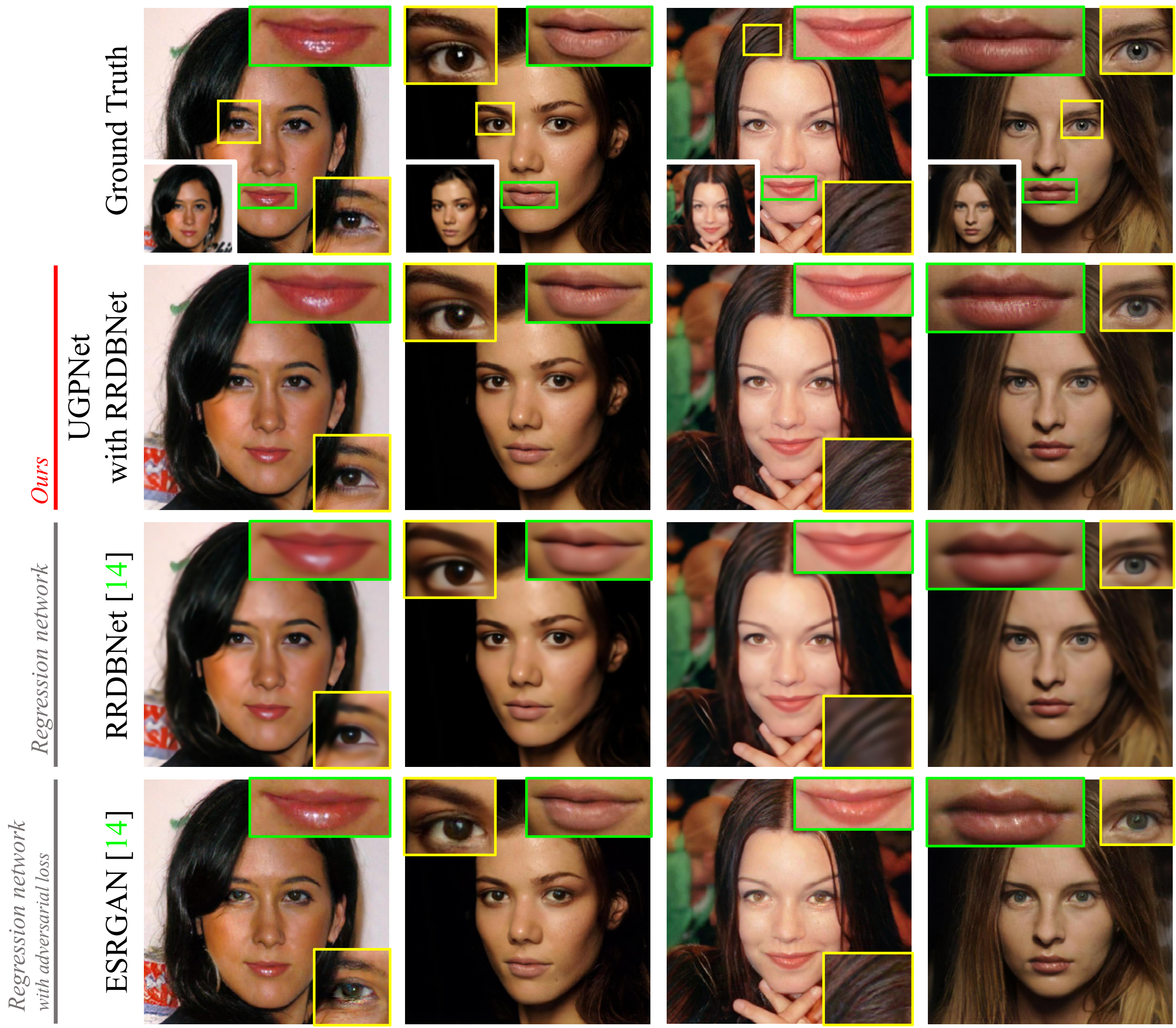


Figure 9. Qualitative comparison on super-resolution with recent regression-based methods including RRDBNet [14] and ESRGAN [14]. The insets at the bottom of the ground-truth images are input degraded images.

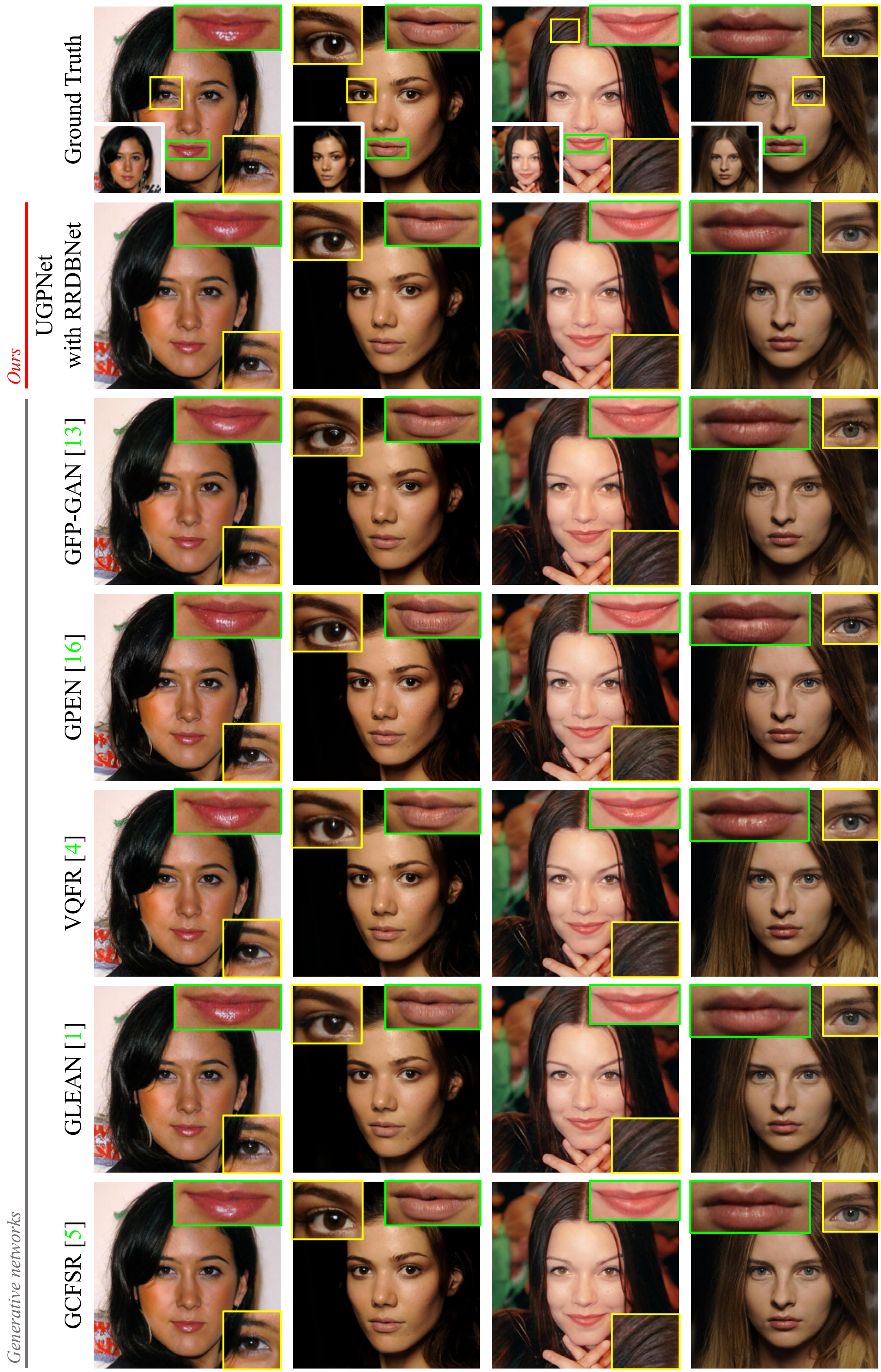


Figure 10. Qualitative comparison on super-resolution with recent generative prior-based methods including GFP-GAN [13], GPEN [16], VQFR [4], GLEAN [1], and GCFSR [5]. The insets at the bottom of the ground-truth images are input degraded images.