GIFNETS: DIFFERENTIABLE GIF ENCODING FRAMEWORK

SUPPLEMENTARY MATERIALS

Innfarn Yoo, Xiyang Luo, Yilin Wang, Feng Yang, and Peyman Milanfar
Google Research - Mountain View, California
[innfarn, xyluo, yilin, fengyang, milanfar]@google.com

This supplementary material provides the details of our neural network architectures (Section 1), training parameters (Section 2), and more examples (Section 3).

1 Architectures

PaletteNet uses InceptionV2 as a backbone network. The last layer of InceptionV2 is connected to a fully connected layer with $N_p \times 3$ outputs. The hyperbolic tangent activation function ($tanh$) is used to limit the output range to $[-1, 1]$. PointNet is also tested as a backbone network for PaletteNet as a comparison to InceptionV2. Same as InceptionV2, the last layer of PointNet's segmentation architecture is connected to a fully connected layer with $N_p \times 3$ outputs.

For DitherNet, a variation of U-Net architecture is used. Five U-Net blocks are used, and the U-Net blocks have (16, 32, 32, 64, 64) output channels with 2×2 max pooling layers in between (16, 32, 64) layers. Up-sampling layers are the reverse of the down-sampling layers. In between down-sampling and up-sampling layers, there are two bottleneck layers, which have (512, 512) output elements respectively. In the final layer, 1×1 convolution layer is applied with the hyperbolic tangent activation function to produce the final output of DitherNet.

BandingNet uses four blocks of convolution layers followed by max pooling layers. The convolution layers have (32, 64, 64, 64) output channels respectively, and the max pooling layers uses 4×4 stride. Global pooling is applied before connecting it to fully connected layers, which have (256, 1) elements. The Sigmoid activation function is used for the last layer to ensure output value range, $[0, 1]$.

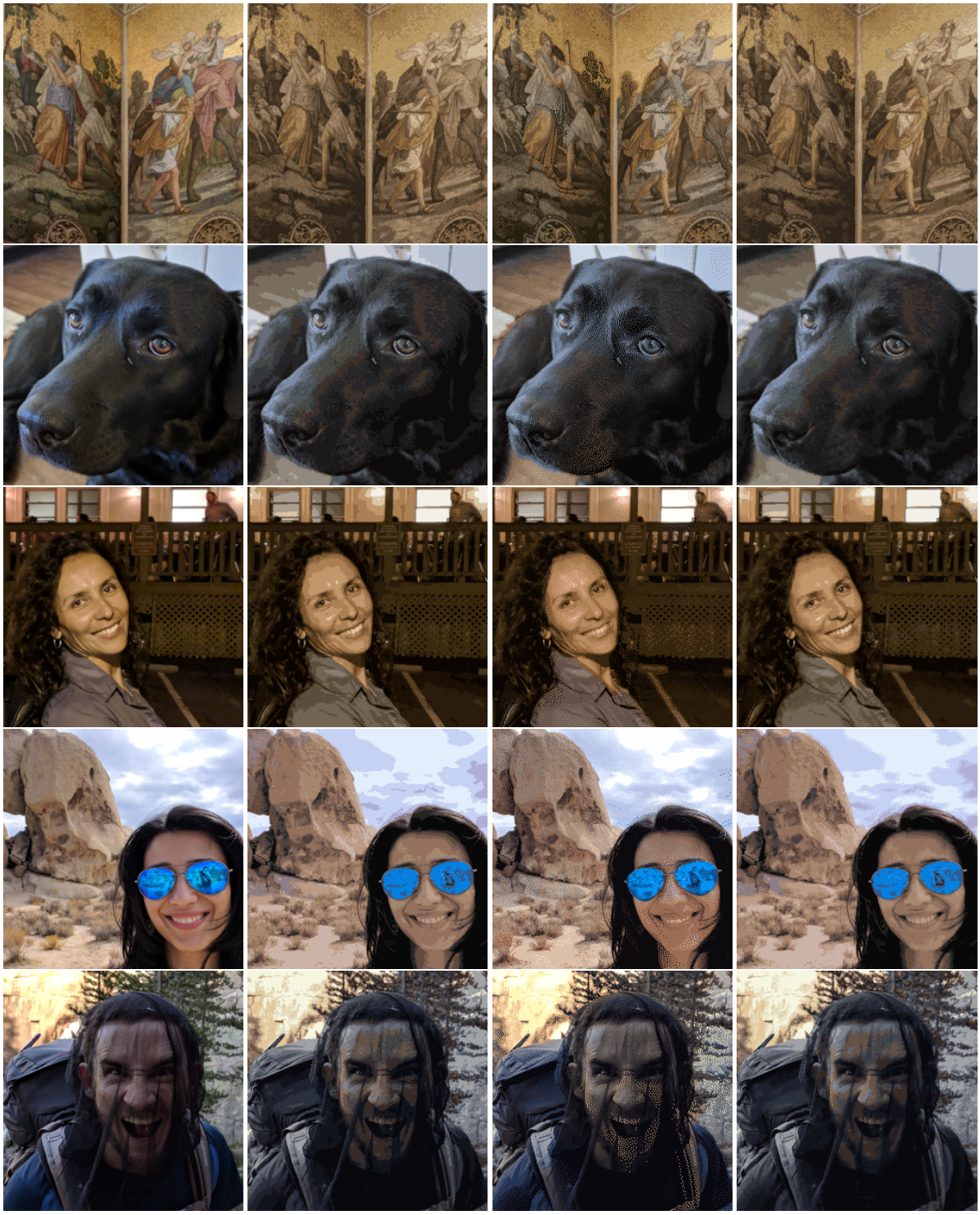
2 Training Parameters

For training PaletteNet, we apply three types of data augmentation: Gaussian noise (0.1), random hue (0.4), random contrast (3.0) to improve the generalization and robustness of our model. We train for a total of 1 million steps, where the Adam optimizer is used with a learning rate of 0.001, and a decay factor of 0.8 and decay step of 100k.

We use different parameters for training DitherNet depending on palette number $N_p = \{16, 32, 64, 128, 256\}$. For low number of palette ($N_p = \{16, 32, 64\}$), we use the following weights: $\lambda = 0.01$, $\gamma = 0.3$, $\delta = 0.05$, $\theta = 0.001$ for the VGG loss, and $\theta = 0.0001$ for NIMA loss (NIMA score range is [0, 10]). For soft projection, we use temperature $T = 0.001$ to have a very close approximation to hard projection. For $N_p = \{128, 256\}$, we change λ to 0.2 and δ to 0.025 in the parameters above. The Adam optimizer is used with a learning rate of 0.0001, decay rate of 0.96, decay step of 20k, and batch size of 4.

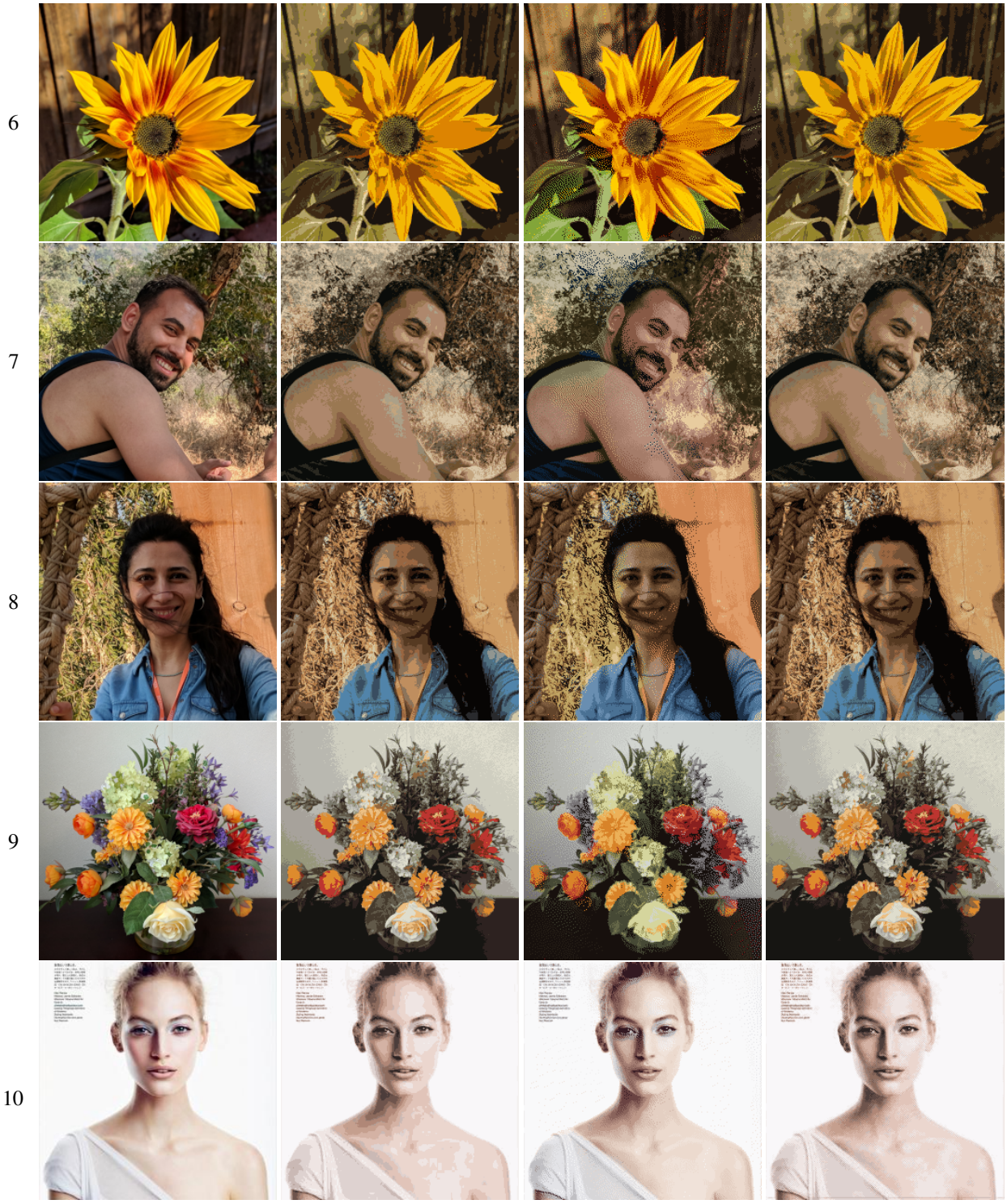
3 More Examples

Figures from 1 to 10 show the comparisons of (a) original images, (b) quantized images with PaletteNet, (c) Median-cut + Floyd-Steinberg images, and (d) PaletteNet + DitherNet images (our final method) with different numbers of palette colors $N_p = \{16, 32, 64, 128, 256\}$. Side-by-side comparison between Floyd-Steinberg and our method are shown in Figure 11.



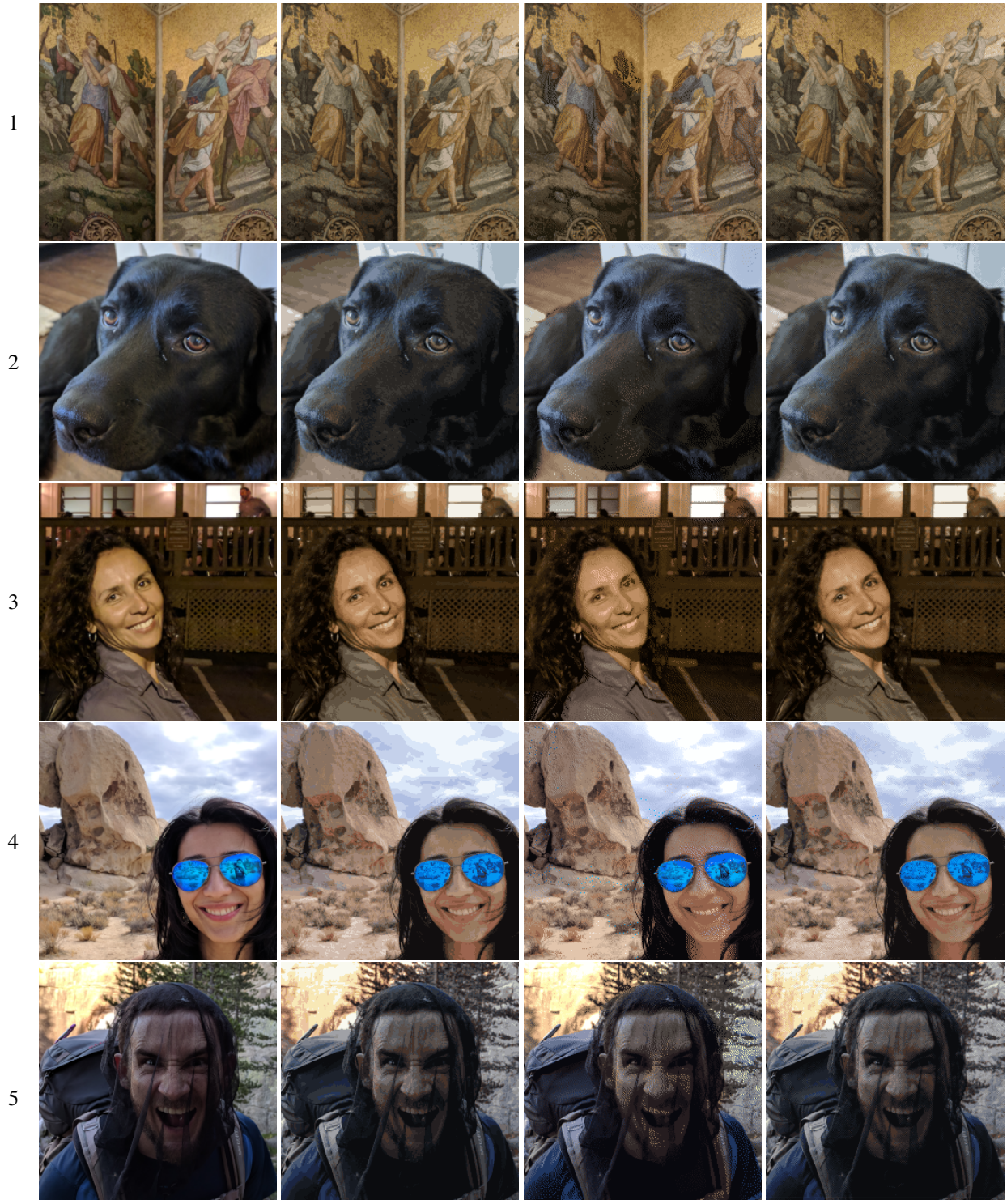
(a) Original (b) Palette (c) FS Dither (d) Ours

Figure 1: Example set 1 with $N_p = 16$



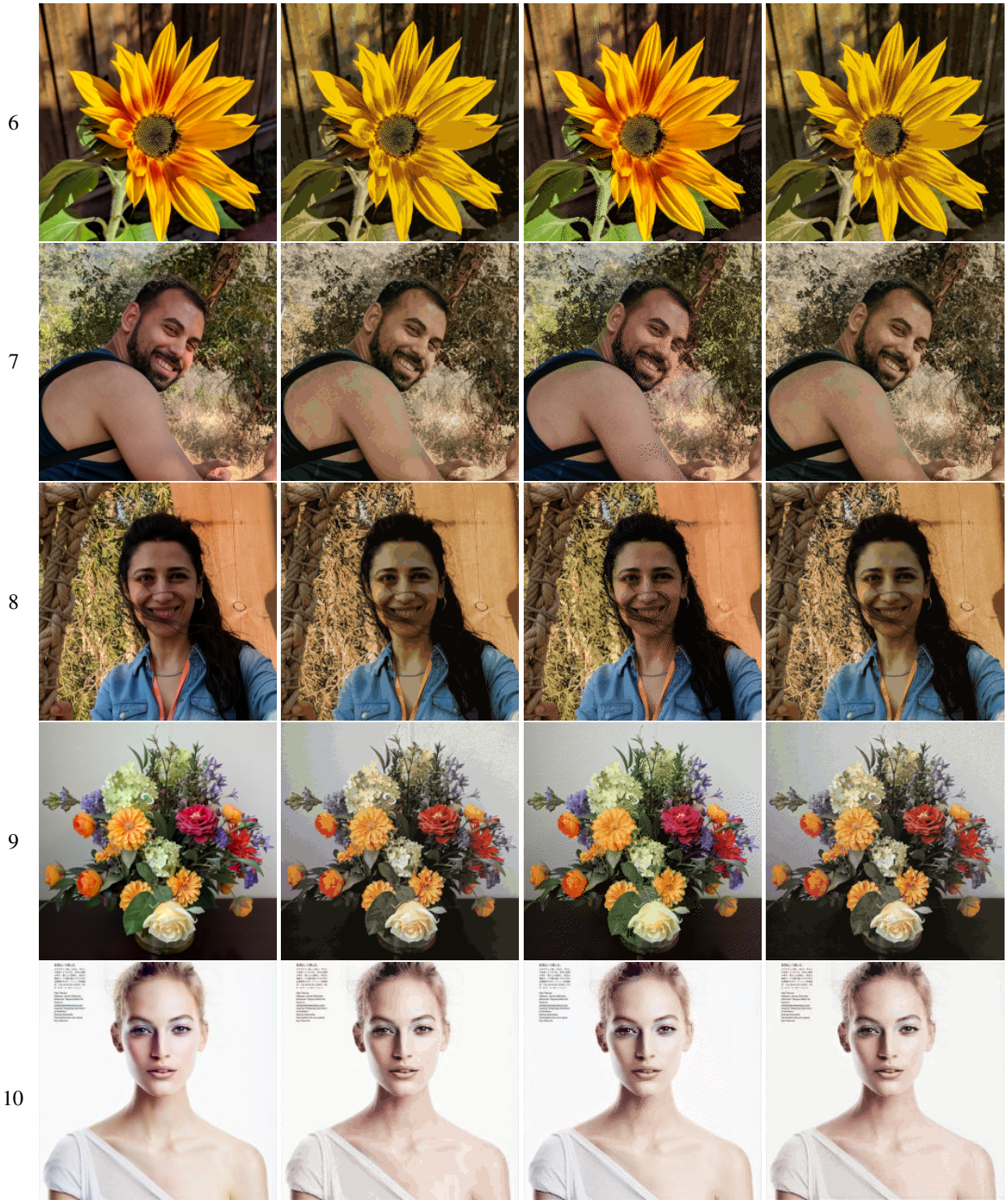
(a) Original (b) Palette (c) FS Dither (d) Ours

Figure 2: Example set 2 with $N_p = 16$



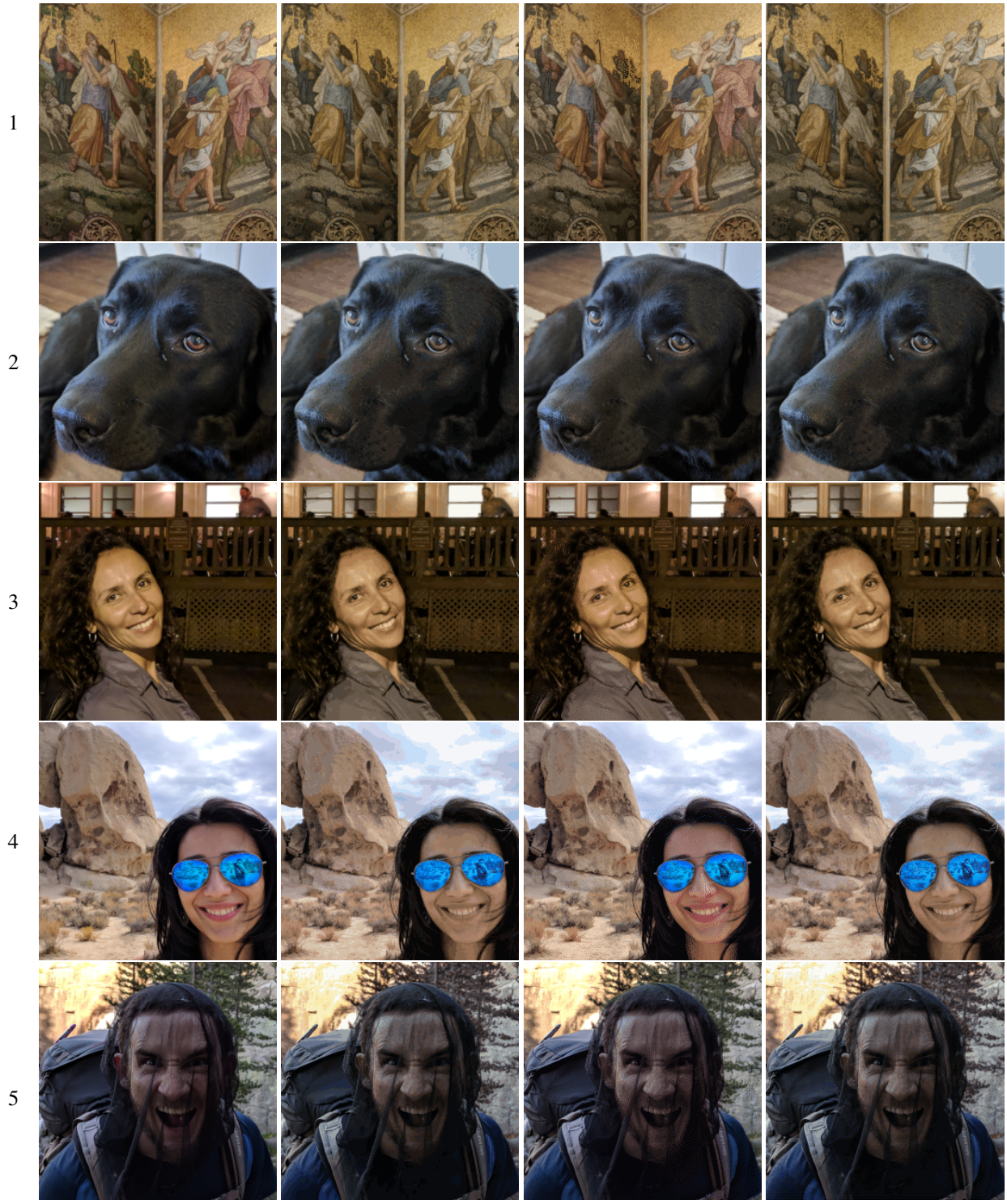
(a) Original (b) Palette (c) FS Dither (d) Ours

Figure 3: Example set 1 with $N_p = 32$



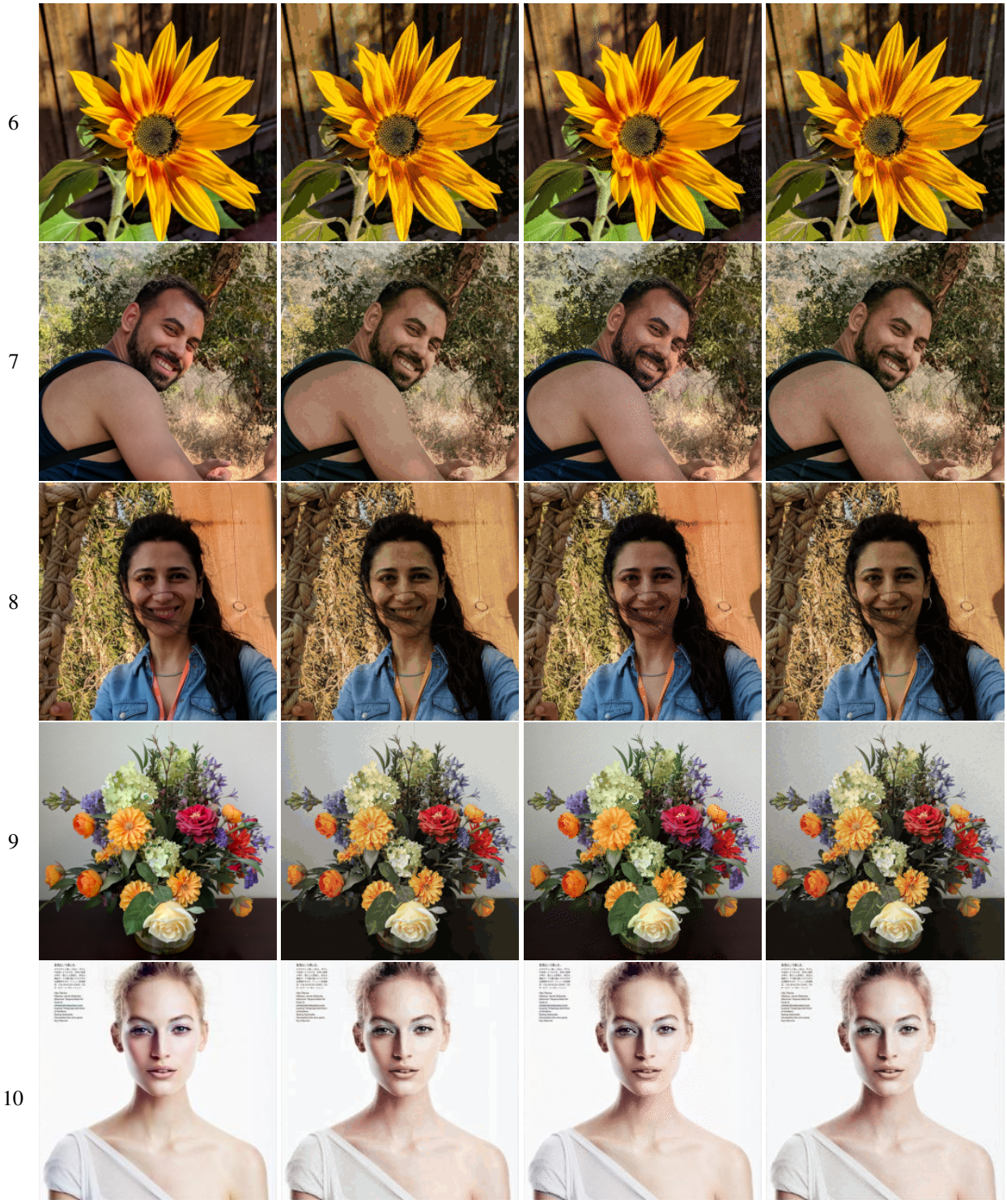
(a) Original (b) Palette (c) FS Dither (d) Ours

Figure 4: Example set 2 with $N_p = 32$



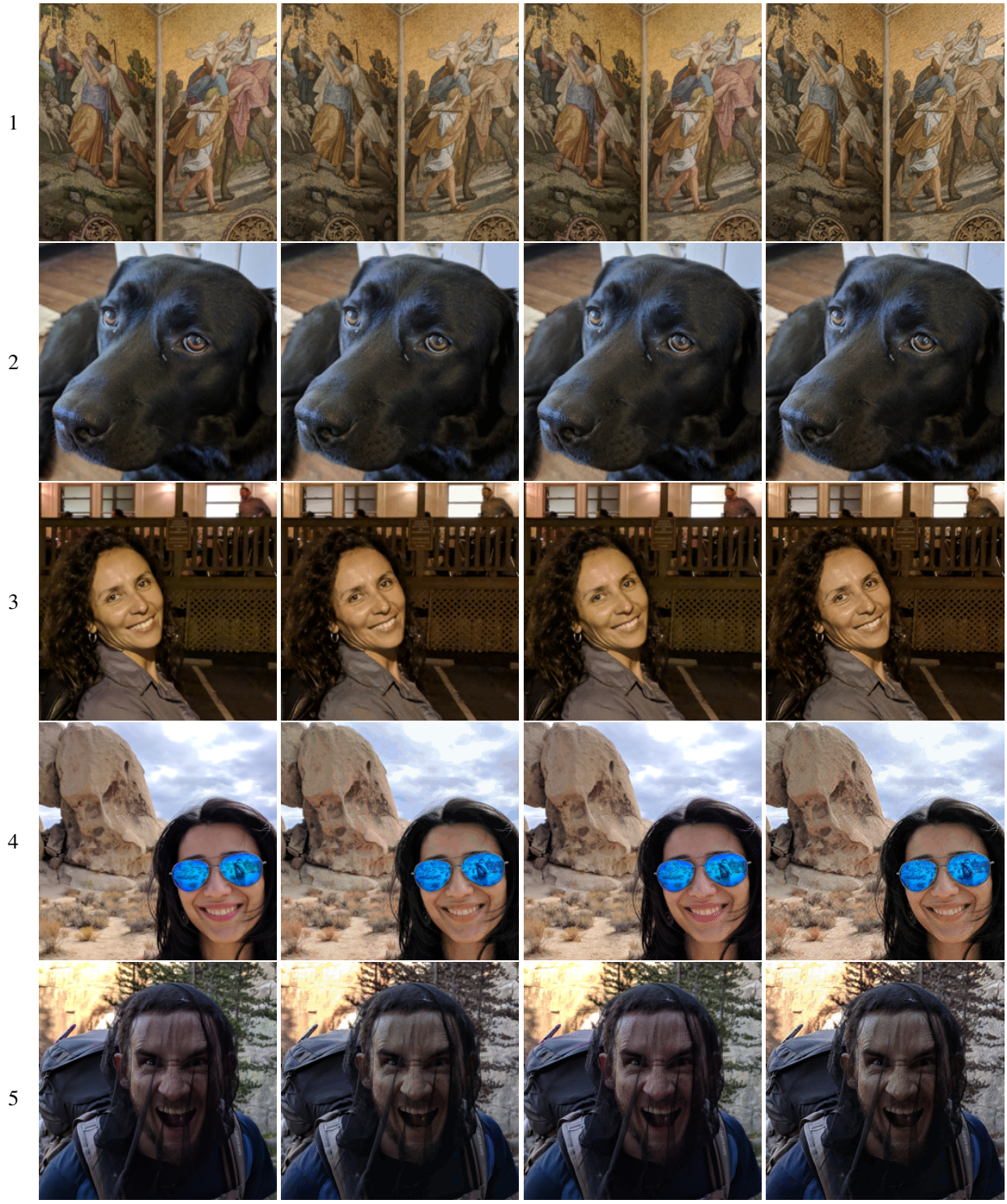
(a) Original (b) Palette (c) FS Dither (d) Ours

Figure 5: Example set 1 with $N_p = 64$



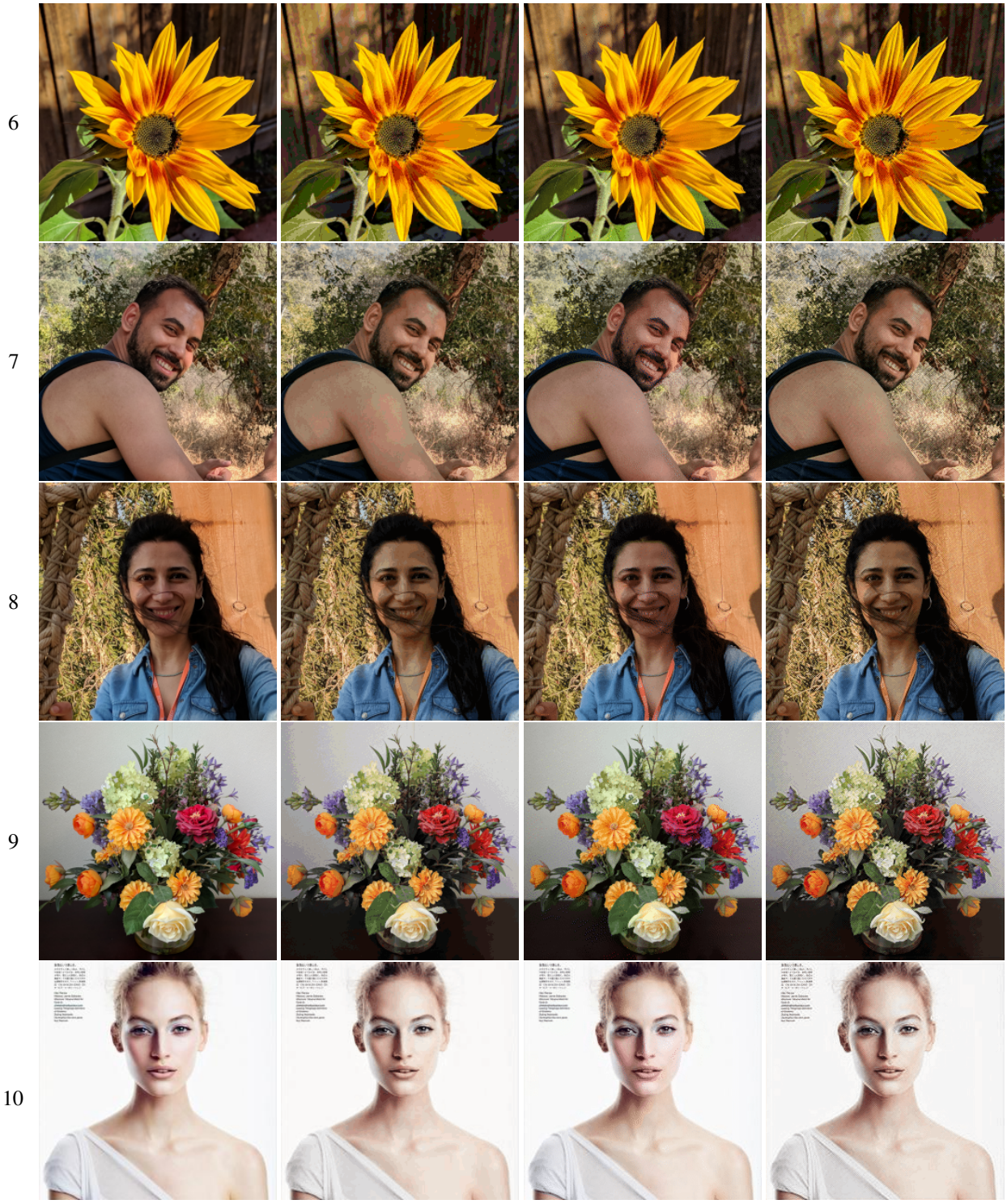
(a) Original (b) Palette (c) FS Dither (d) Ours

Figure 6: Example set 2 with $N_p = 64$



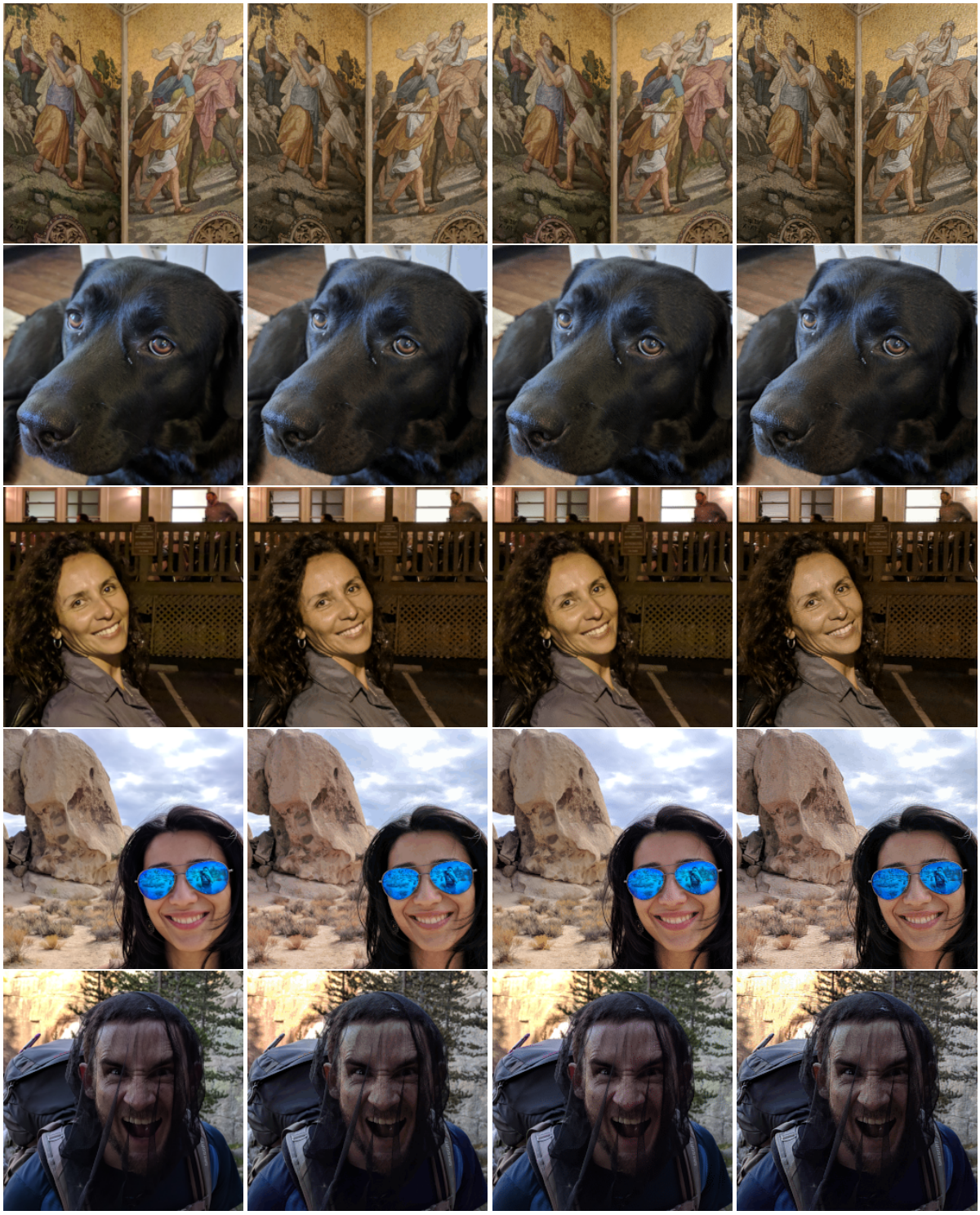
(a) Original (b) Palette (c) FS Dither (d) Ours

Figure 7: Example set 1 with $N_p = 128$



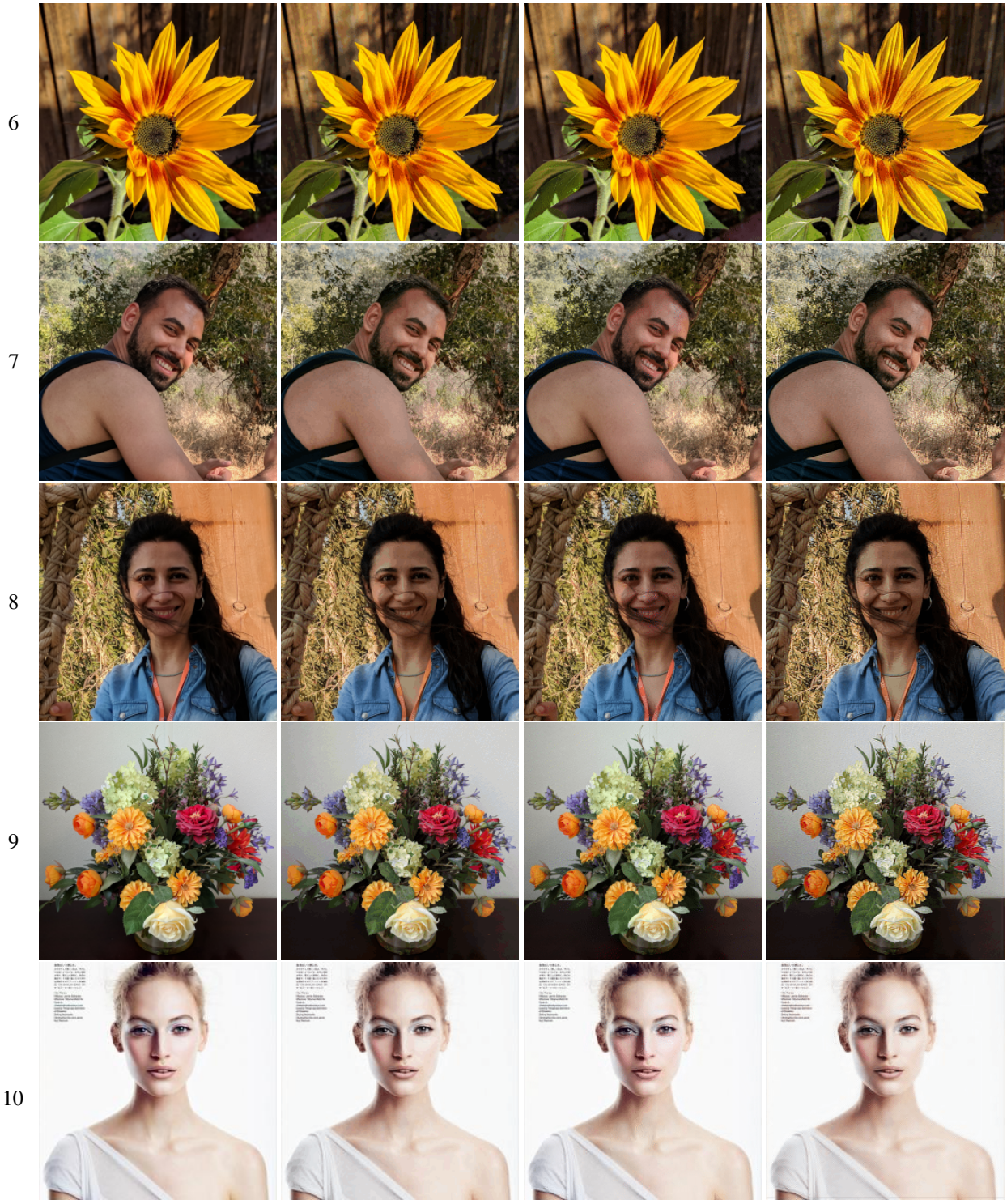
(a) Original (b) Palette (c) FS Dither (d) Ours

Figure 8: Example set 2 with $N_p = 128$



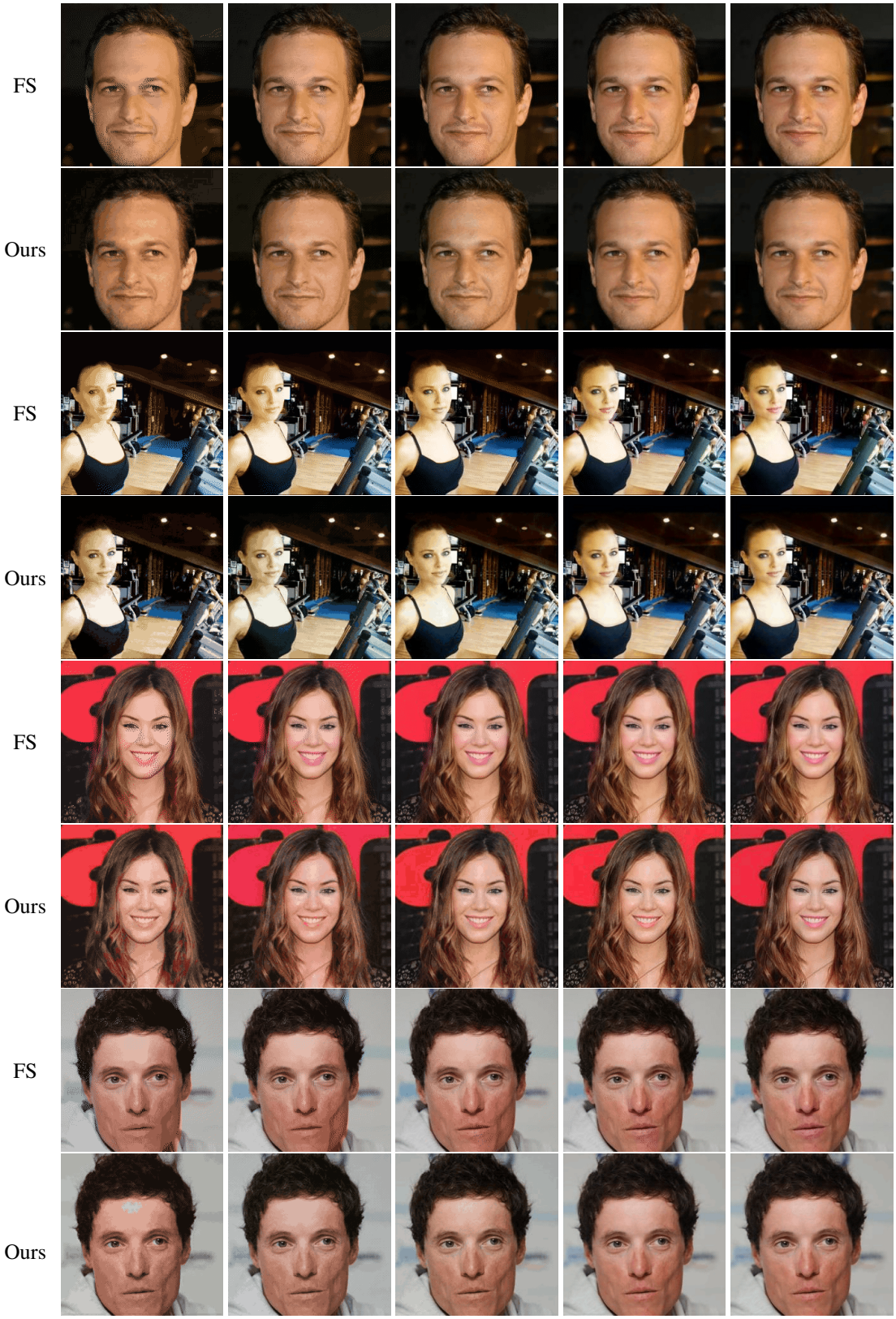
(a) Original (b) Palette (c) FS Dither (d) Ours

Figure 9: Example set 1 with $N_p = 256$



(a) Original (b) Palette (c) FS Dither (d) Ours

Figure 10: Example set 2 with $N_p = 256$



(a) 16 Palette (b) 32 Palette (c) 64 Palette (d) 128 Palette (e) 256 Palette

Figure 11: Side-by-side comparison.