# Supplementary material for “Inverting Visual Representations with Convolutional Networks”

Alexey Dosovitskiy Thomas Brox
University of Freiburg
Freiburg im Breisgau, Germany
{dosovits,brox}@cs.uni-freiburg.de

**Network architectures** Table 1 shows the architecture of AlexNet. Tables 2-6 show the architectures of networks we used for inverting different features. After each fully connected and convolutional layer there is always a leaky ReLU nonlinearity. Networks for inverting HOG and LBP have two streams. Stream A compresses the input features spatially and accumulates information over large regions. We found this crucial to get good estimates of the overall brightness of the image. Stream B does not compress spatially and hence can better preserve fine local details. At one points the outputs of the two streams are concatenated and processed jointly, denoted by “J”. K stands for kernel size, S for stride.

**Shallow features details** As mentioned, in the paper, for all three methods we use implementations from the *VLFeat* library [2] with the default settings. We use the Felzenszwalb et al. version of HOG with cell size 8. For SIFT we used 3 levels per octave, the first octave was 0 (corresponding to full resolution), the number of octaves was set automatically, effectively searching keypoints of all possible sizes.

The LBP version we used works with $3 \times 3$ pixel neighborhoods. Each of the 8 non-central bits is equal to one if the corresponding pixel is brighter than the central one. All possible 256 patterns are quantized into 58 patterns. These include 56 patterns with exactly one transition from 0 to 1 when going around the central pixel, plus one quantized pattern comprising two uniform patterns, plus one quantized pattern containing all other patterns. The quantized LBP patterns are then grouped into local histograms over cells of $16 \times 16$ pixels.

**Experiments: shallow representations** Figure 1 shows several images and their reconstructions from HOG, SIFT and LBP. HOG allows for the best reconstruction, SIFT slightly worse, LBP yet slightly worse. Colors are often reconstructed correctly, but sometimes are wrong, for ex-

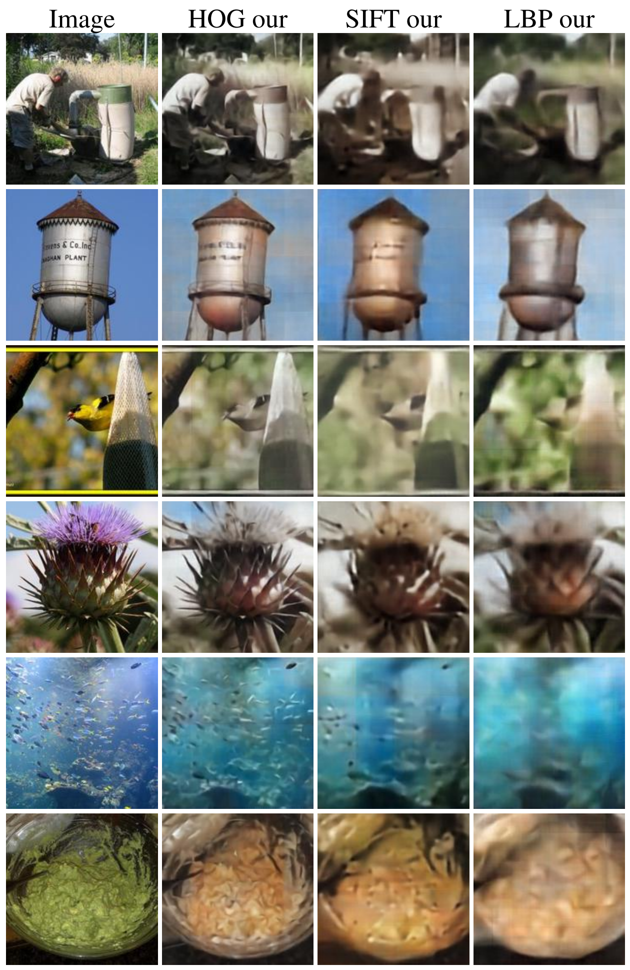

Figure 1: Inversion of shallow image representations.

| layer | CONV1 | | CONV2 | | CONV3 | CONV4 | CONV5 | | FC6 | | FC7 | | FC8 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| processing steps | conv1 relu1 | mpool1 norm1 | conv2 relu2 | mpool2 norm2 | conv3 relu3 | conv4 relu4 | conv5 relu5 | mpool5 | fc6 relu6 | drop6 | fc7 relu7 | drop7 | fc8 |
| out size | 55 | 27 | 27 | 13 | 13 | 13 | 13 | 6 | 1 | 1 | 1 | 1 | 1 |
| out channels | 96 | 96 | 256 | 256 | 384 | 384 | 256 | 256 | 4096 | 4096 | 4096 | 4096 | 1000 |

Table 1: Summary of the AlexNet network. Input image size is $227 \times 227$.

ample in the last row. Interestingly, all network typically agree on estimated colors.

**Experiments: AlexNet** We show here several additional figures similar to ones from the main paper. Reconstructions from different layers of AlexNet are shown in Figure 2. Figure 3 shows results illustrating the 'dark knowledge' hypothesis, similar to Figure 8 from the main paper. We reconstruct from all FC8 features, as well as from only 5 largest ones or all except the 5 largest ones. It turns out that the top 5 activations are not very important.

Figure 4 shows images generated by activating single neurons in different layers and setting all other neurons to zero. Particularly interpretable are images generated this way from FC8. Every FC8 neuron corresponds to a class. Hence the image generated from the activation of, say, "apple" neuron, could be expected to be a stereotypical apple.

| Layer | Input | InSize | K | S | OutSize |
|---|---|---|---|---|---|
| convA1 | HOG | $32\times32\times31$ | 5 | 2 | $16\times16\times256$ |
| convA2 | convA1 | $16\times16\times256$ | 5 | 2 | $8\times8\times512$ |
| convA3 | convA2 | $8\times8\times512$ | 3 | 2 | $4\times4\times1024$ |
| upconvA1 | convA3 | $4\times4\times1024$ | 4 | 2 | $8\times8\times512$ |
| upconvA2 | upconvA1 | $8\times8\times512$ | 4 | 2 | $16\times16\times256$ |
| upconvA3 | upconvA2 | $16\times16\times256$ | 4 | 2 | $32\times32\times128$ |
| convB1 | HOG | $32\times32\times31$ | 5 | 1 | $32\times32\times128$ |
| convB2 | convB1 | $32\times32\times128$ | 3 | 1 | $32\times32\times128$ |
| convJ1 | {upconvA3, convB2} | $32\times32\times256$ | 3 | 1 | $32\times32\times256$ |
| convJ2 | convJ1 | $32\times32\times256$ | 3 | 1 | $32\times32\times128$ |
| upconvJ4 | convJ2 | $32\times32\times128$ | 4 | 2 | $64\times64\times64$ |
| upconvJ5 | upconvJ4 | $64\times64\times64$ | 4 | 2 | $128\times128\times32$ |
| upconvJ6 | upconvJ5 | $128\times128\times32$ | 4 | 2 | $256\times256\times3$ |

Table 2: Network for reconstructing from HOG features.

| Layer | Input | InSize | K | S | OutSize |
|---|---|---|---|---|---|
| conv1 | SIFT | $64\times64\times133$ | 5 | 2 | $32\times32\times256$ |
| conv2 | conv1 | $32\times32\times256$ | 3 | 2 | $16\times16\times512$ |
| conv3 | conv2 | $16\times16\times512$ | 3 | 2 | $8\times8\times1024$ |
| conv4 | conv3 | $8\times8\times1024$ | 3 | 2 | $4\times4\times2048$ |
| conv5 | conv4 | $4\times4\times2048$ | 3 | 1 | $4\times4\times2048$ |
| conv6 | conv5 | $4\times4\times2048$ | 3 | 1 | $4\times4\times1024$ |
| upconv1 | conv6 | $4\times4\times1024$ | 4 | 2 | $8\times8\times512$ |
| upconv2 | upconv1 | $8\times8\times512$ | 4 | 2 | $16\times16\times256$ |
| upconv3 | upconv2 | $16\times16\times256$ | 4 | 2 | $32\times32\times128$ |
| upconv4 | upconv3 | $32\times32\times128$ | 4 | 2 | $64\times64\times64$ |
| upconv5 | upconv4 | $64\times64\times64$ | 4 | 2 | $128\times128\times32$ |
| upconv6 | upconv5 | $128\times128\times32$ | 4 | 2 | $256\times256\times3$ |

Table 3: Network for reconstructing from SIFT features.

| Layer | Input | InSize | K | S | OutSize |
|---|---|---|---|---|---|
| convA1 | LBP | $16\times16\times58$ | 5 | 2 | $8\times8\times256$ |
| convA2 | convA1 | $8\times8\times256$ | 5 | 2 | $4\times4\times512$ |
| convA3 | convA2 | $4\times4\times512$ | 3 | 1 | $4\times4\times1024$ |
| upconvA1 | convA3 | $4\times4\times1024$ | 4 | 2 | $8\times8\times512$ |
| upconvA2 | upconvA1 | $8\times8\times512$ | 4 | 2 | $16\times16\times256$ |
| convB1 | LBP | $16\times16\times58$ | 5 | 1 | $16\times16\times128$ |
| convB2 | convB1 | $16\times16\times128$ | 3 | 1 | $16\times16\times128$ |
| convJ1 | {upconvA2, convB2} | $16\times16\times384$ | 3 | 1 | $16\times16\times256$ |
| convJ2 | convJ1 | $16\times16\times256$ | 3 | 1 | $16\times16\times128$ |
| upconvJ3 | convJ2 | $16\times16\times128$ | 4 | 2 | $32\times32\times128$ |
| upconvJ4 | upconvJ3 | $32\times32\times128$ | 4 | 2 | $64\times64\times64$ |
| upconvJ5 | upconvJ4 | $64\times64\times64$ | 4 | 2 | $128\times128\times32$ |
| upconvJ6 | upconvJ5 | $128\times128\times32$ | 4 | 2 | $256\times256\times3$ |

Table 4: Network for reconstructing from LBP features.

| Layer | Input | InSize | K | S | OutSize |
|---|---|---|---|---|---|
| conv1 | AlexNet-CONV5 | $6\times6\times256$ | 3 | 1 | $6\times6\times256$ |
| conv2 | conv1 | $6\times6\times256$ | 3 | 1 | $6\times6\times256$ |
| conv3 | conv2 | $6\times6\times256$ | 3 | 1 | $6\times6\times256$ |
| upconv1 | conv3 | $6\times6\times256$ | 5 | 2 | $12\times12\times256$ |
| upconv2 | upconv1 | $12\times12\times256$ | 5 | 2 | $24\times24\times128$ |
| upconv3 | upconv2 | $24\times24\times128$ | 5 | 2 | $48\times48\times64$ |
| upconv4 | upconv3 | $48\times48\times64$ | 5 | 2 | $96\times96\times32$ |
| upconv5 | upconv4 | $96\times96\times32$ | 5 | 2 | $192\times192\times3$ |

Table 5: Network for reconstructing from AlexNet CONV5 features.

| Layer | Input | InSize | K | S | OutSize |
|---|---|---|---|---|---|
| fc1 | AlexNet-FC8 | 1000 | – | – | 4096 |
| fc2 | fc1 | 4096 | – | – | 4096 |
| fc3 | fc2 | 4096 | – | – | 4096 |
| reshape | fc3 | 4096 | – | – | $4\times4\times256$ |
| upconv1 | reshape | $4\times4\times256$ | 5 | 2 | $8\times8\times256$ |
| upconv2 | upconv1 | $8\times8\times256$ | 5 | 2 | $16\times16\times128$ |
| upconv3 | upconv2 | $16\times16\times128$ | 5 | 2 | $32\times32\times64$ |
| upconv4 | upconv3 | $32\times32\times64$ | 5 | 2 | $64\times64\times32$ |
| upconv5 | upconv4 | $64\times64\times32$ | 5 | 2 | $128\times128\times3$ |

Table 6: Network for reconstructing from AlexNet FC8 features.

What we observe looks rather like it might be the average of all images of the class. For some classes the reconstructions are somewhat interpretable, for others – not so much.

Qualitative comparison of reconstructions with our method to the reconstructions of [1] and the results with AlexNet-based autoencoders is given in Figure 5 .

Reconstructions from feature vectors obtained by interpolating between feature vectors of two images are shown in Figure 6 , both for fixed AlexNet and autoencoder training. More examples of such interpolations with fixed AlexNet are shown in Figure 7 .

As described in section 5.5 of the main paper, we tried two different distributions for sampling random feature activations: a histogram-based and a truncated Gaussian. Figure 8 shows the results with fixed AlexNet network and truncated Gaussian distribution. Figures 9 and 10 show images generated with autoencoder-trained networks. Note that images generated from autoencoders look much less realistic than images generated with a network with fixed AlexNet weights. This indicates that reconstructing from AlexNet features requires a strong natural image prior.

## References

[1] A. Mahendran and A. Vedaldi. Understanding deep image representations by inverting them. In *CVPR*, 2015. 3, 5

[2] A. Vedaldi and B. Fulkerson. Vlfeat: an open and portable library of computer vision algorithms. In *International Conference on Multimedia*, pages 1469–1472, 2010. 1

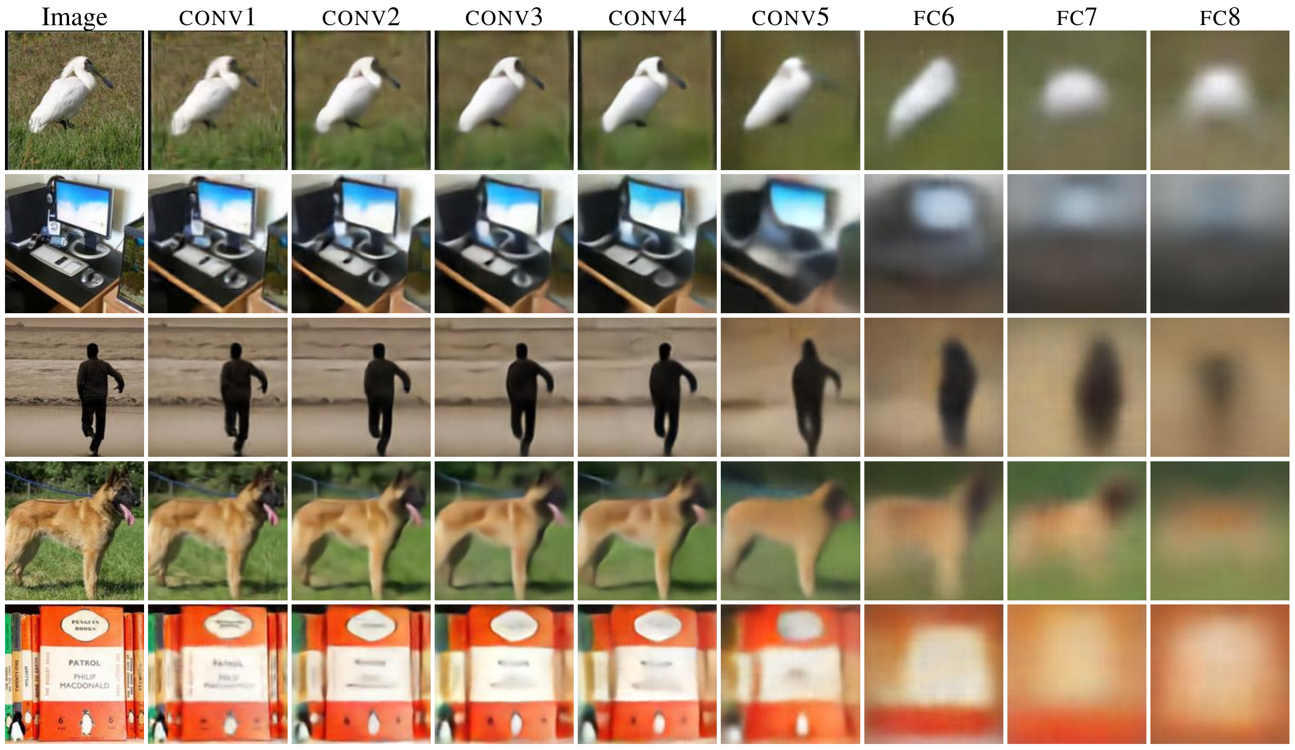

Figure 2: Reconstructions from different layers of AlexNet.

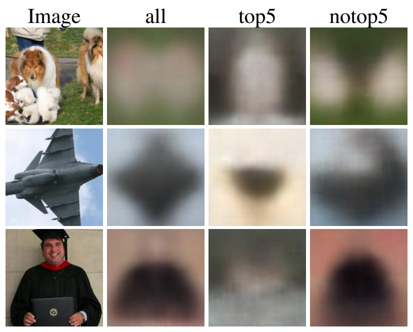

Figure 3: Left to right: input image, reconstruction from fc8, reconstruction from 5 largest activations in FC8, reconstruction from all FC8 activations except 5 largest ones.

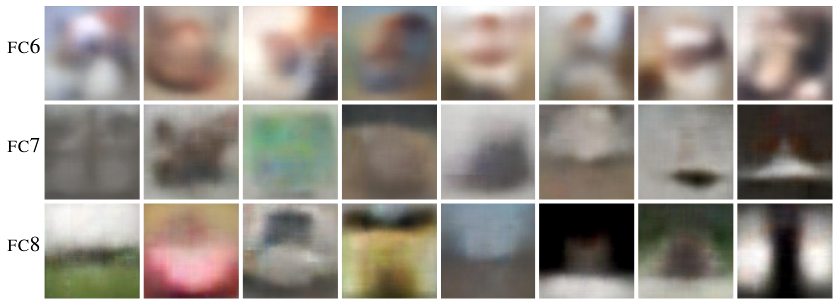

Figure 4: Reconstructions from single neuron activations in the fully connected layers of AlexNet. The FC8 neurons correspond to classes, left to right: kite, convertible, desktop computer, school bus, street sign, soup bowl, bell pepper, soccer ball.

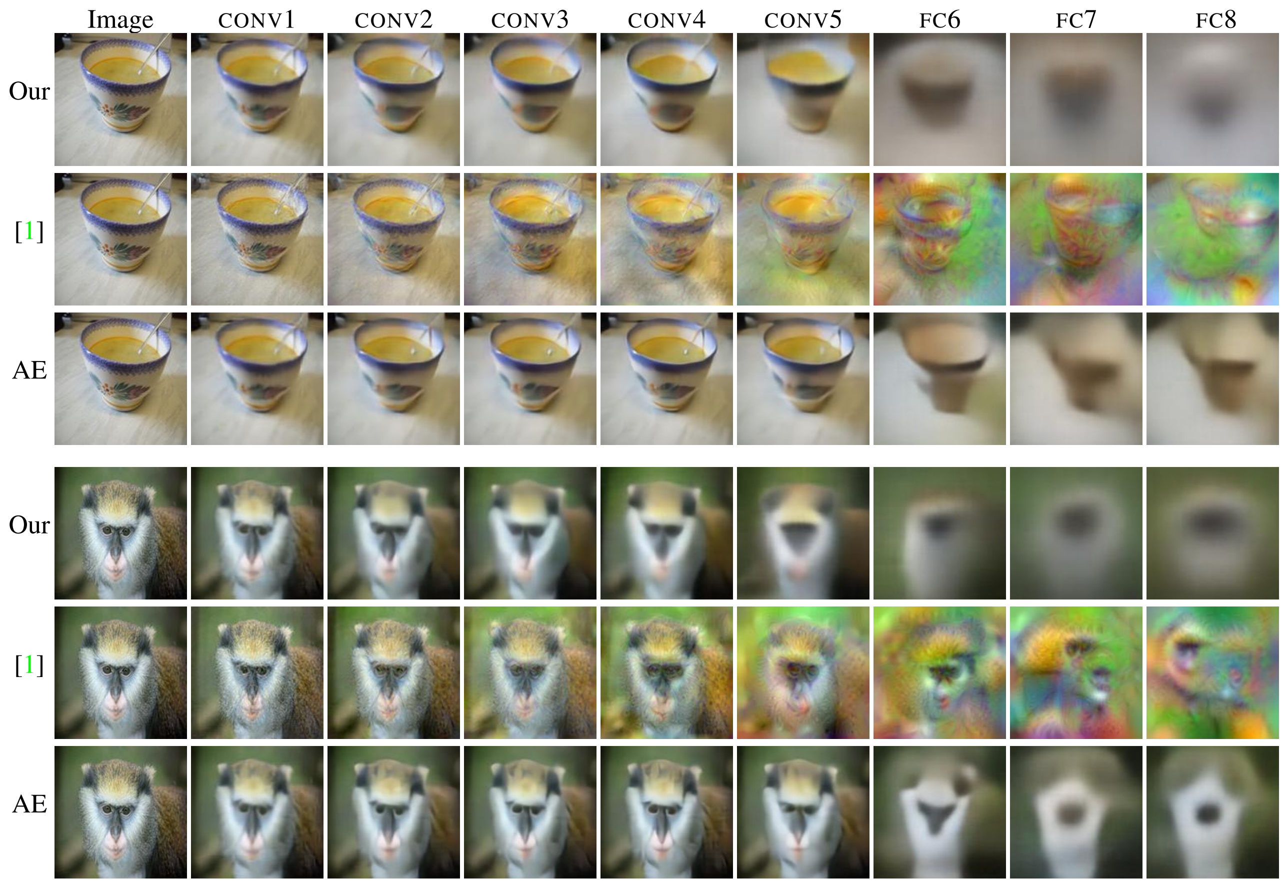

Figure 5: Reconstructions from different layers of AlexNet with our method and [1].

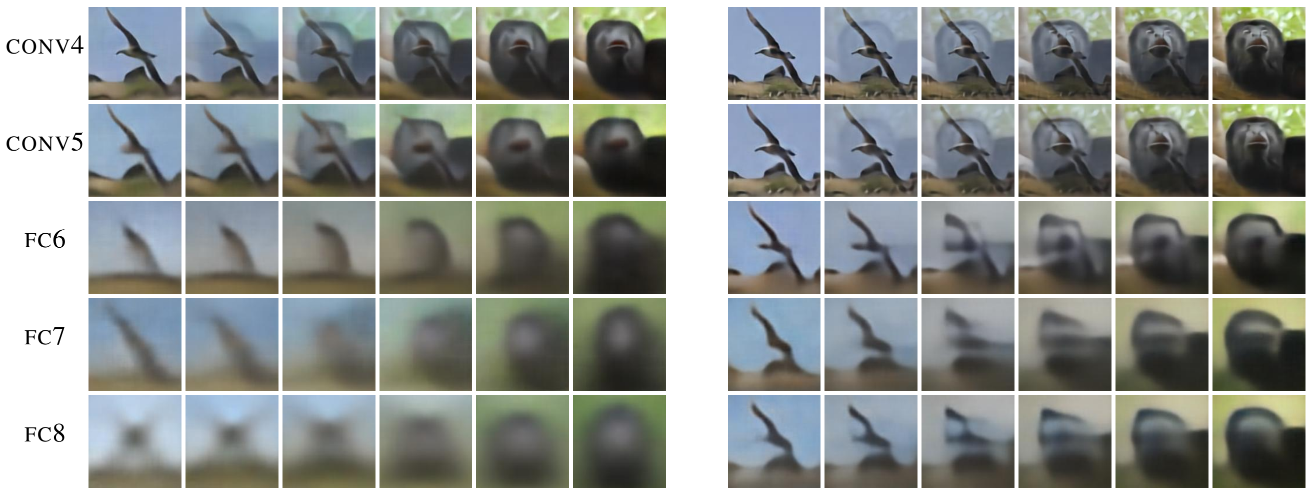

Figure 6: Interpolation between the features of two images. Left: AlexNet weights fixed, right: autoencoder.

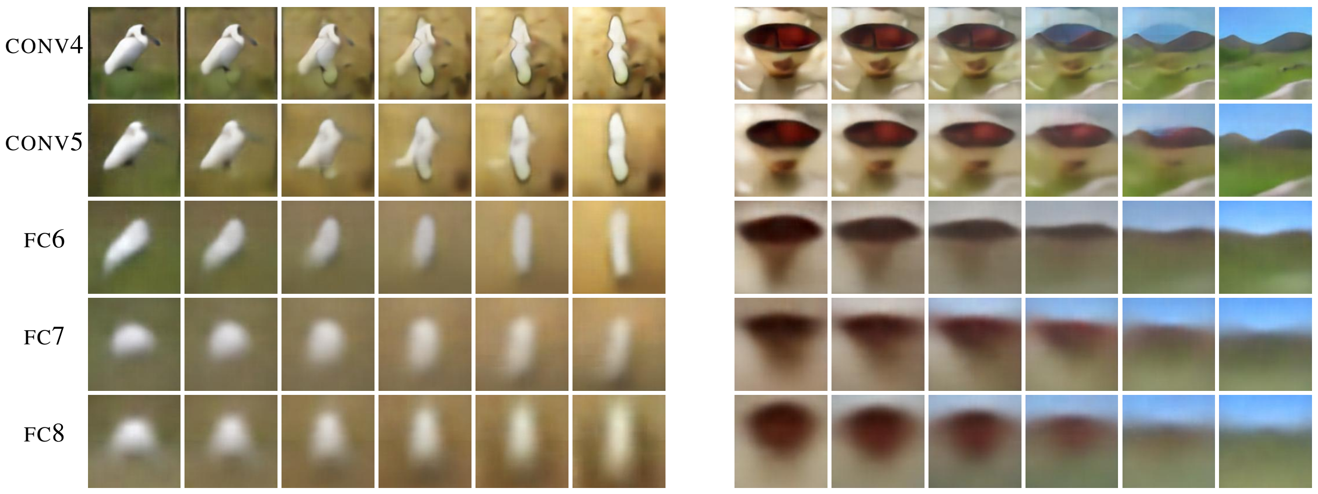

Figure 7: More interpolations between the features of two images with fixed AlexNet weights.

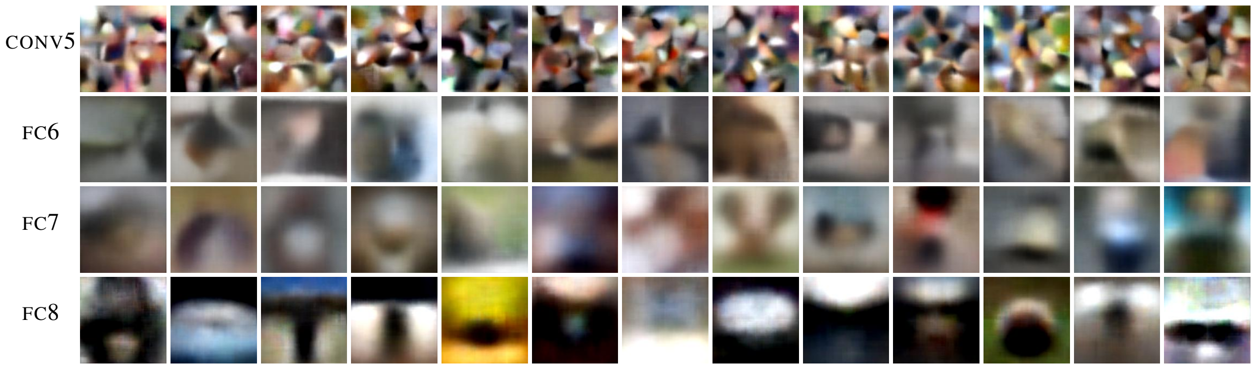

Figure 8: Images generated from random feature vectors of top layers of AlexNet with the simpler truncated Gaussian distribution (see section 5.5 of the main paper).

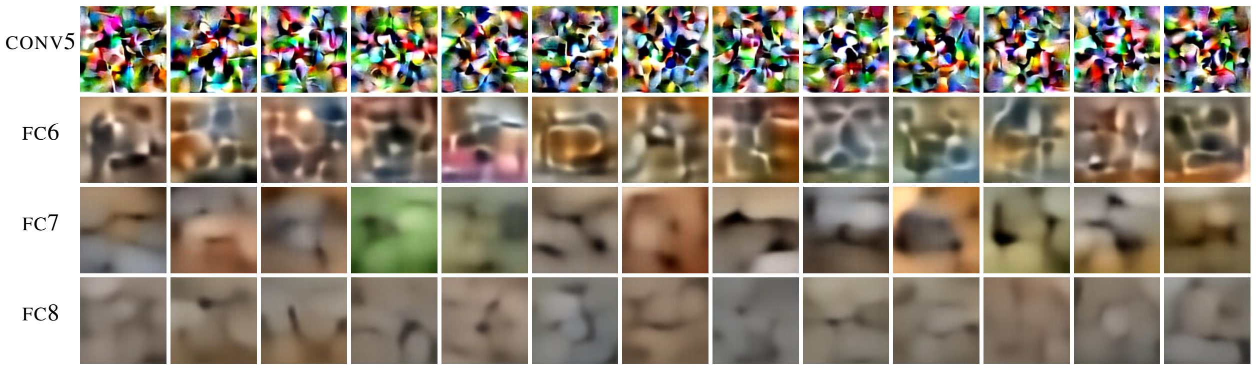

Figure 9: Images generated from random feature vectors of top layers of AlexNet-based autoencoders with the histogram-based distribution (see section 5.5 of the main paper).

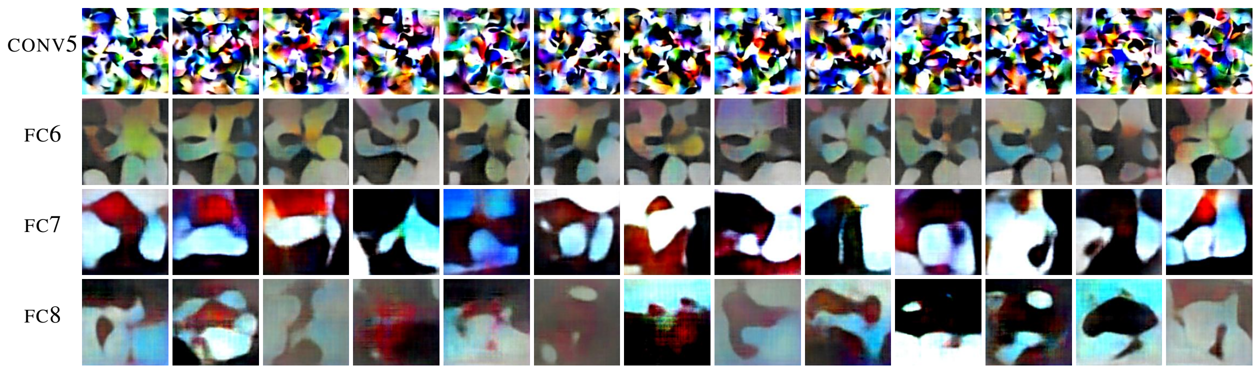

Figure 10: Images generated from random feature vectors of top layers of AlexNet-based autoencoders with the simpler truncated Gaussian distribution (see section 5.5 of the main paper).