PRANC: Pseudo RAndom Networks for Compacting deep models (Supplementary material)

Appendix

Orthogonality and norm of basis networks:

In the main submission, we mentioned that random basis networks are almost perpendicular to each other in high dimensional spaces. To show this, we generate 1000 random vectors at the d dimensional space (varying d from 10 to 1000), and plot the histogram of their ℓ_2 norm and pairwise cosine similarities in Figures 1 and 2 respectively. We also run this experiment 100 times, calculate the maximum cosine similarity for each run, and plot the histogram of maximum values in Figure 3. As expected, at higher number of dimensions, the cosine similarity gets closer to 0 and the norm gets closer to 1. This empirically suggests that at higher dimensions, the random basis are close to orthonormal basis. Please note that our method does not require orthonormal basis to work.

Reconstruction using a subset of basis:

Figure 5 shows the distribution of alpha values for a few images from Kodak dataset. As expected, the alpha values vary across the basis models. This motivated us to study "what if we use only a subset of basis models instead of all of them?".

For image classification, we do this experiment by selecting random $p\%$ of the basis with varying p. We repeat this 4 times. As another selection method, we sort alpha values based on their absolute values and use the top $p\%$ of them. As shown in Figure 4, in the random selection, we need most of the basis to be able to retrieve a reasonable accuracy while for the sorted case, a small percentage of the basis models is sufficient to get a reasonable accuracy. This is an interesting observation that somehow loosely suggests that the loss landscape of the optimization for alpha values is reasonably smooth. This is intentionally a vague statement since it needs further investigation as future work.

For image compression, we perform the same experiment and show reasonable reconstructed images with a subset of alpha values with the largest absolute values. Since we have a different set of alphas for each layer of MLP in image compression, we sort absolute values and select top $p\%$ of each layer with higher values. We vary p and visualize the reconstructed images for each p value in Figure 7, 8 and 9. Additionally, we report the effect of p in PSNR and MS-SSIM for Kodak dataset in Table 1. Similar to our observation in mage classification, we observe that images with $p = 70\%$ has acceptable quality.

Aside from better understanding the method, this observation can enable communicating a deep model or an image progressively where the sender sends a subset of alpha values (most important ones) first and then gradually sends the rest to improve the model accuracy or the image quality. For image compression, this method is somehow similar to the application of Progressive JPEG where it is hand-crafted to send the low frequency components first. Please note that to use this in practice, this progressive version of our method has some extra-overhead since we also need to communicate which alpha values are sent at each step (e.g., using a bit for each basis). Studying this in more detail is left for future work.

Details of Image Compression:

As described in the main submission, in image compression experiments, in order to change the bit-per-pixel (bpp), we fix the network architecture and vary the number of α values per layer. We report the number of α values per layer for each bpp in Table 2.

Compression to Advanced Image Compression methods:

We compare our method with more advanced codecs (e.g., BPG, VTM) and learned-based image compression methods. Results for CLIC-Mobile are in Table 3 and the results for Kodak are in Figure 6. Note that advanced codecs are heavily hand-crafted by a large community. And, learned-based methods utilize a training dataset to learn a good code (similar to auto-encoder), hence, they may not be able to compress a single image without having access to a corpus of images. In contrast, our method can compress a single image without using a population of images. Moreover, for the same reason, our method cannot be biased towards the popular cases (head of distribution). Obviously, our method has other biases (e.g., what can be represented with INR) that needs to be studied as the future work.

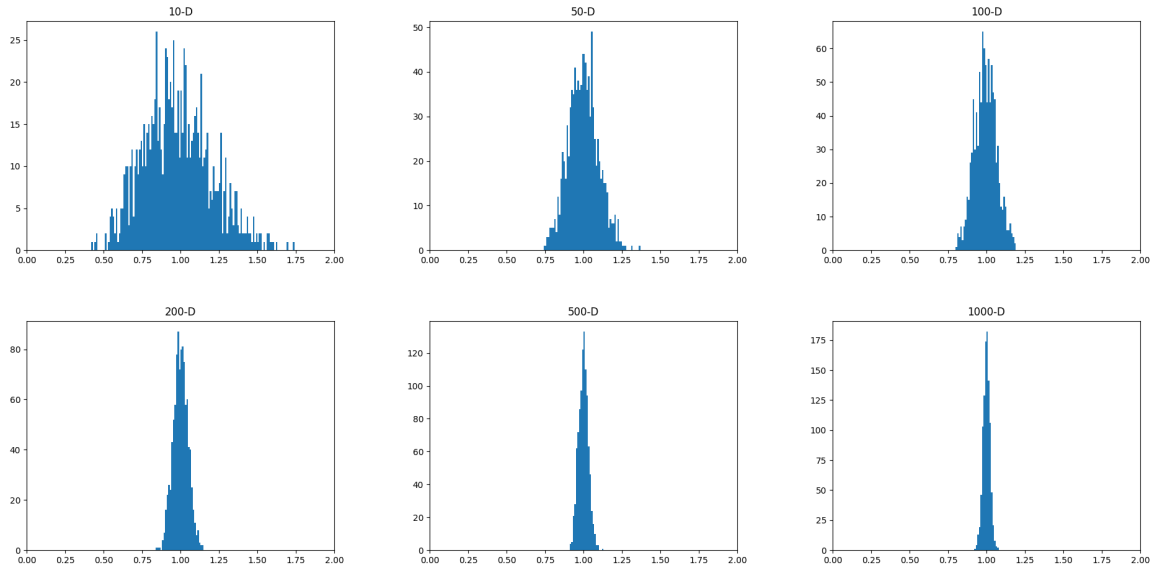


Figure 1. Histogram of ℓ_2 norm of 1000 randomly generated d dimensional vectors. When increasing d the norm approaches 1.

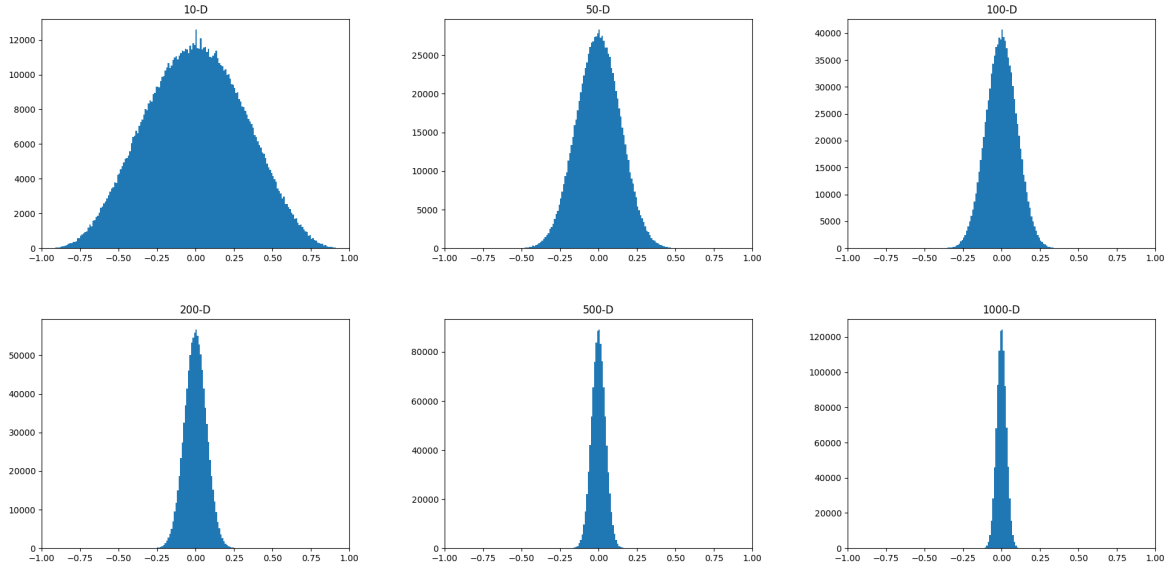


Figure 2. Histogram of pairwise cosine similarity of 1000 randomly generated d dimensional vectors. When increasing d the cosine similarity approaches 0.

Visual Comparison to JPEG:

Similar to Figure 5 of main submission, we include more visual comparison to JPEG in Figures 10 through 13 (for Kodak dataset with 512×768 resolution) and Figures 14 and 15 (for CLIC-Mobile dataset with 1512×2016 or 2016×1512 resolution). Moreover, in Figures 16 through 18, we include visual comparison in chest x-ray dataset with 1024×1024 resolution.

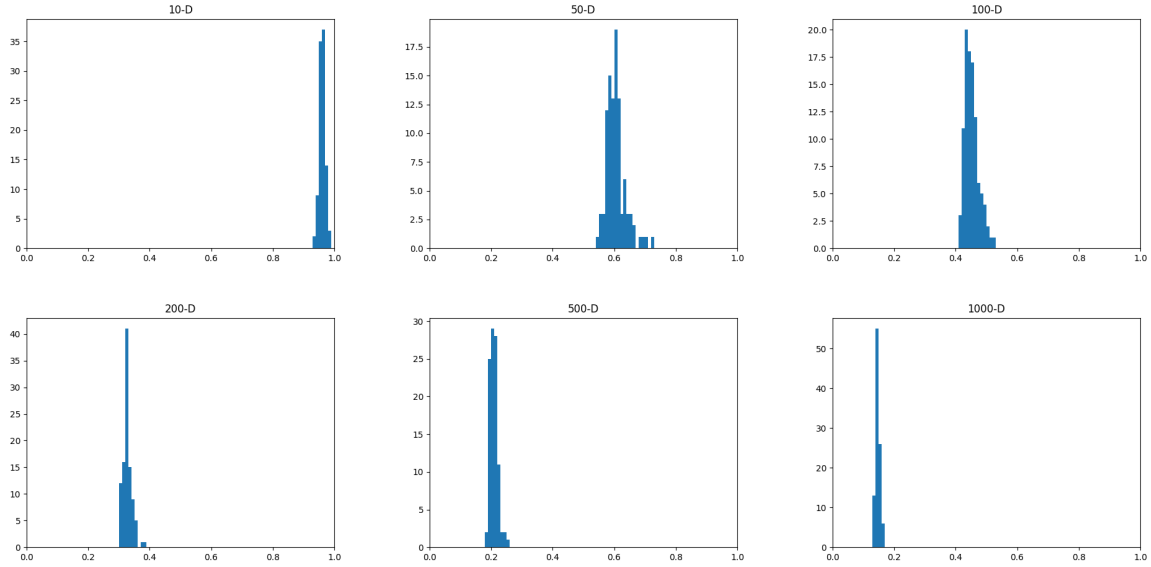


Figure 3. Histogram of maximum of pairwise cosine similarity of 1000 randomly generated d dimensional vectors over 100 trials. When increasing d, the maximum of cosine similarity approaches 0.

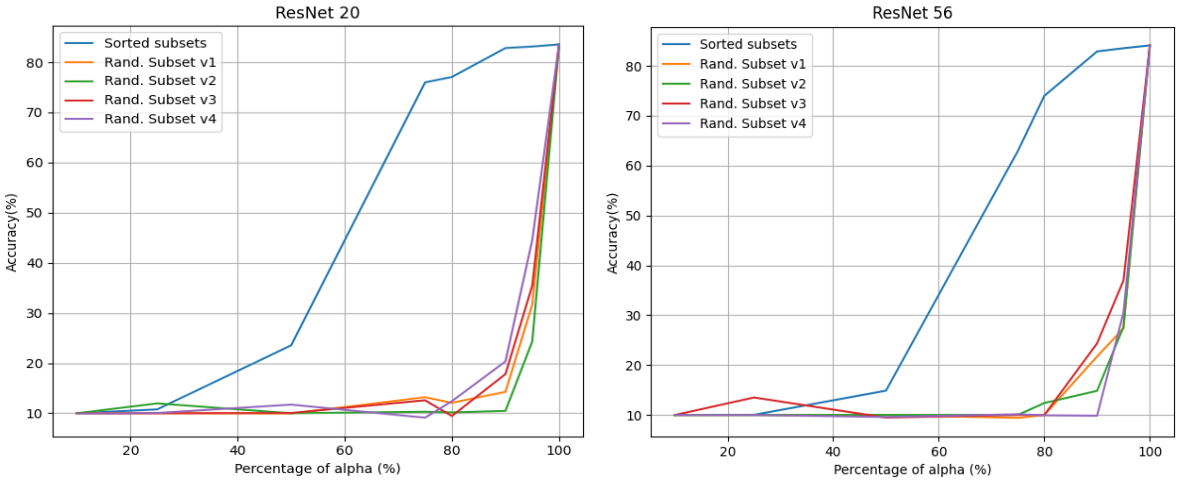


Figure 4. Effect of using only $p\%$ of basis models selected randomly (4 times) or selected based on highest absolute values of alphas.

Table 1. **Effect of keeping $p\%$ of α with highest absolute value:**

percentile ($p\%$)	10	20	30	40	50	60	70	80	90	100
bpp	0.07	0.14	0.22	0.29	0.36	0.43	0.50	0.58	0.65	0.72
PSNR	11.94	12.3	13.4	14.98	17.01	19.51	22.71	26.92	31.85	33.64
MS-SSIM	0.10	0.12	0.18	0.28	0.41	0.55	0.71	0.86	0.96	0.97

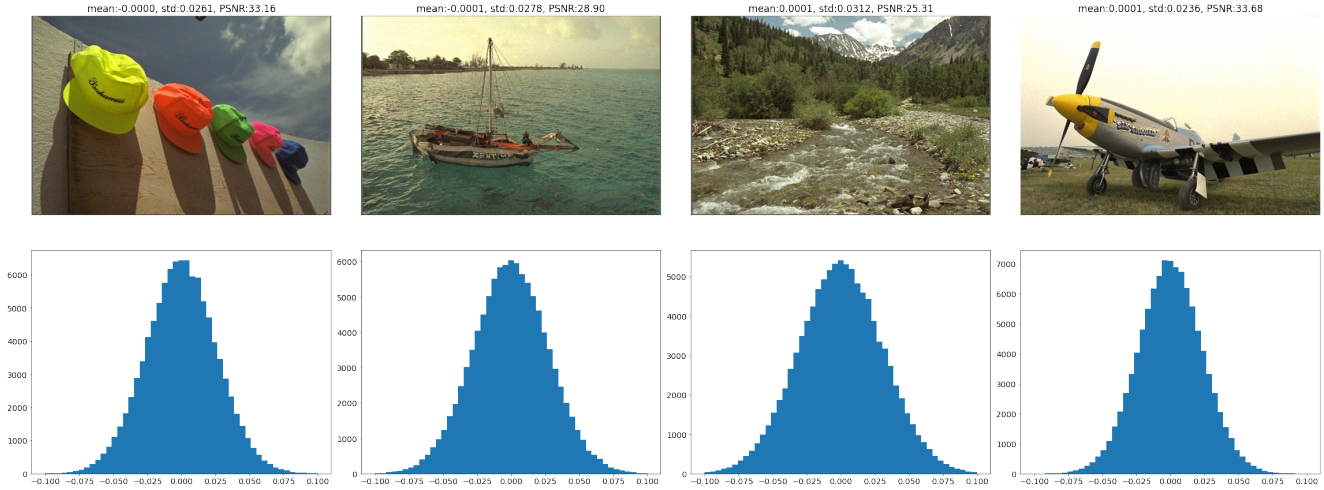


Figure 5. **Distribution of alphas:** We plot the distribution of alpha values for a few Kodak images.

Table 2. **Details of image compression models:** We report the number of α values per layer for each bpp. We use MLP with hidden dimension of 256 for Kodak dataset at bpp of 0.18 (first row), and hidden dimension of 512 for all other experiments. All settings use Fourier mapping of size 512. We use fewer number of alpha values for the last layer since the last layer has fewer number of weights as it goes from hidden layer to only 3 dimensions (RGB values). Also, for the first row, we use more number of alpha values for first layer since it has more number of weights (512×256).

Dataset	bpp	Layer-1	Layer-2	Layer-3	Layer-4	Layer-5	Total (αs)
4*Kodak	0.18	10k	7.5k	7.5k	7.5k	2k	34.5k
	0.31	15k	15k	15k	15k	2k	57k
	0.52	10k	30k	30k	30k	2k	102k
	0.72	20k	40k	40k	40k	2k	142k
CLIC-Mobile	0.12	45k	45k	45k	45k	2k	182k
Chest x-ray	0.15	20k	20k	20k	20k	2k	82k

Table 3. **CLIC-mobile Dataset Image Compression:** Similar to Table 6 of the main submission, we include comparison to advanced codecs like BPG and VTM. We also compare with some learning-based image compression methods (e.g., MBT, CST, BMS).

Model	bpp	PSNR	MS-SSIM
VTM	0.183	33.07	0.964
CST	0.146	31.85	0.957
MBT	0.146	31.62	0.955
BPG	0.128	30.65	0.942
WebP	0.185	30.07	0.940
BMS	0.113	29.38	0.936
JPEG2000	0.126	29.40	0.918
JPEG	0.195	24.82	0.836
Trained INR	0.125	26.93	0.864
PRANC (ours)	0.119	29.71	0.920

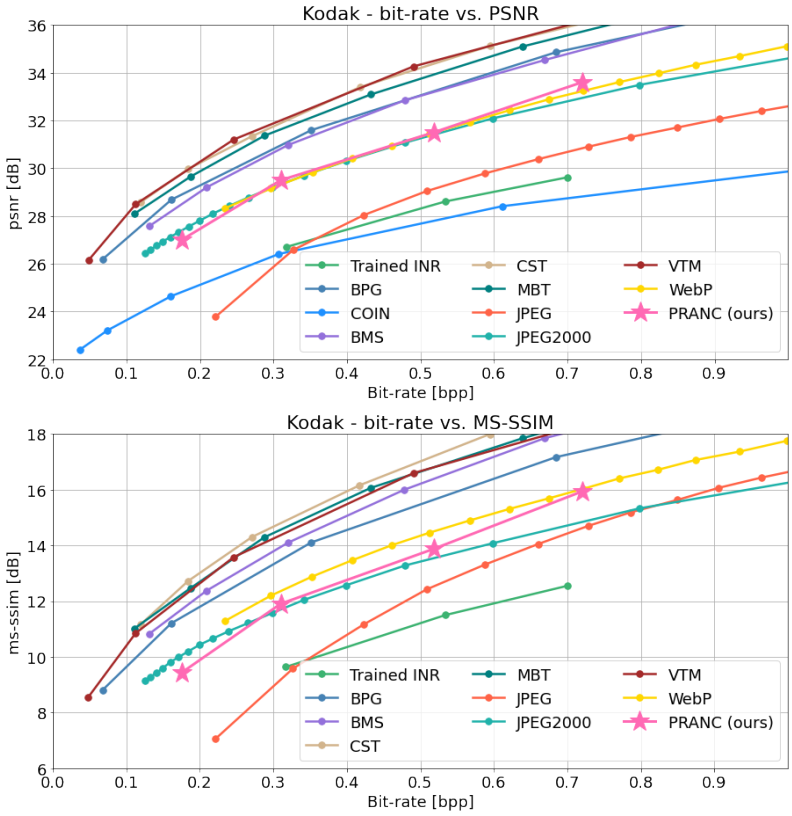


Figure 6. **Kodak Dataset Image Compression:** Similar to Figure 4 of the main submission, we include comparison to advanced codecs like BPG and VTM. We also compared with learned-based image compression (e.g., MBT, CST, BMS).

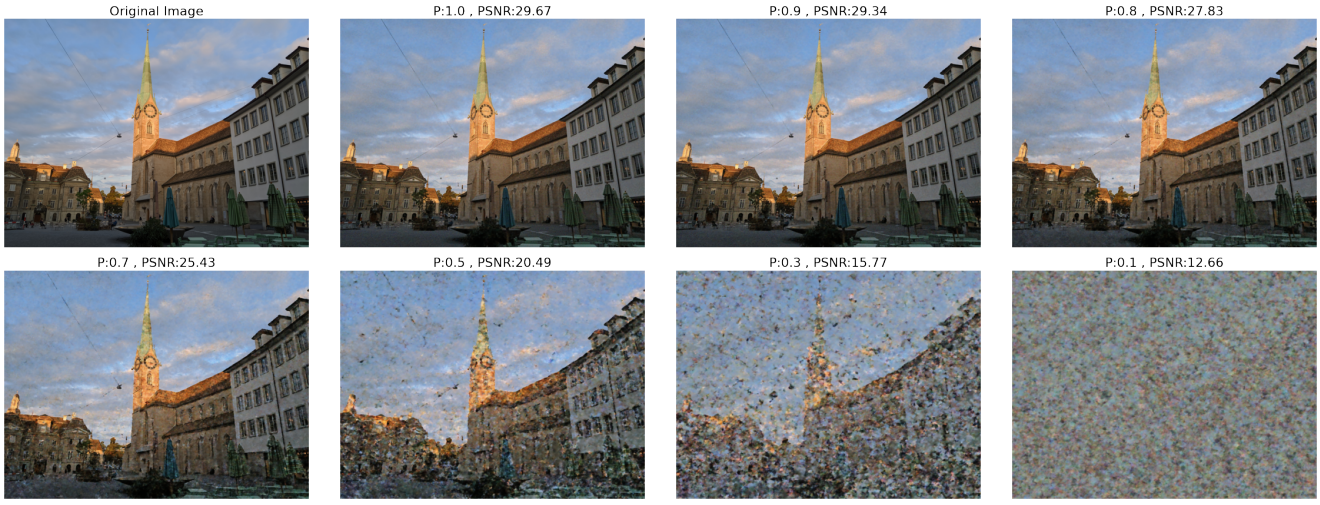


Figure 7. **Effect of keeping $p\%$ of basis models with the highest absolute alpha values.** We get reasonable images with smaller subset of basis models.

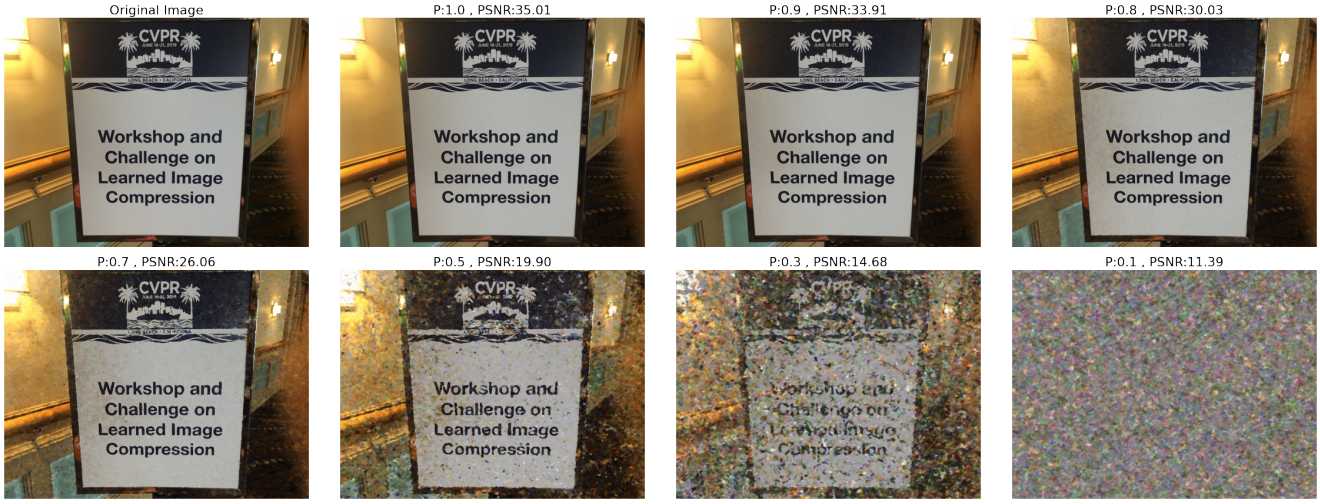


Figure 8. See Figure 7.

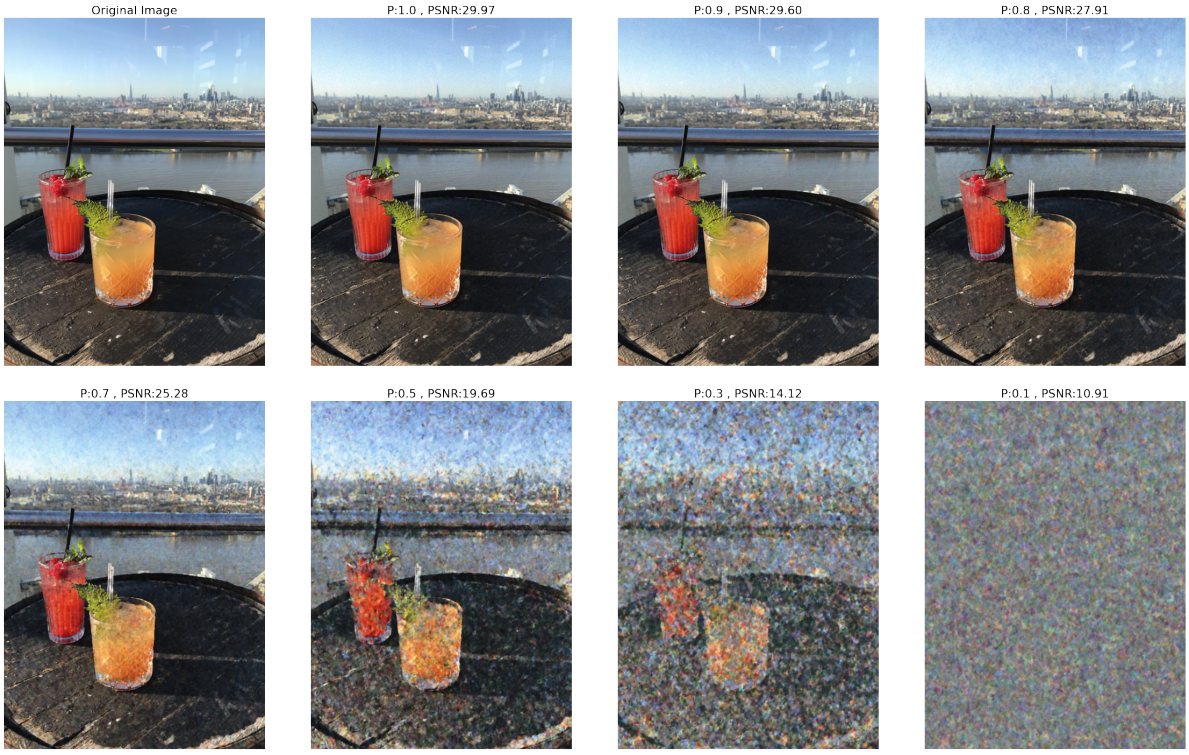
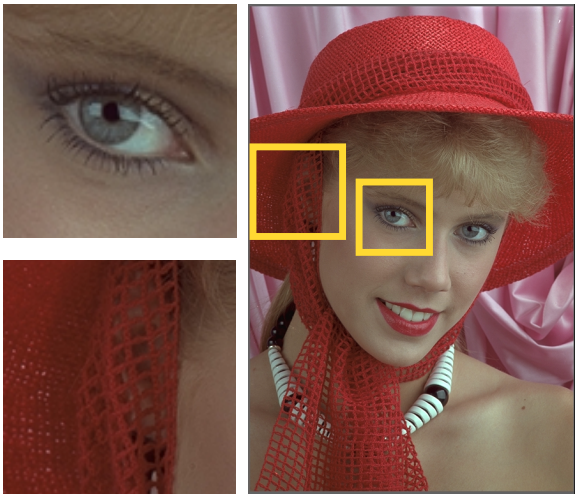
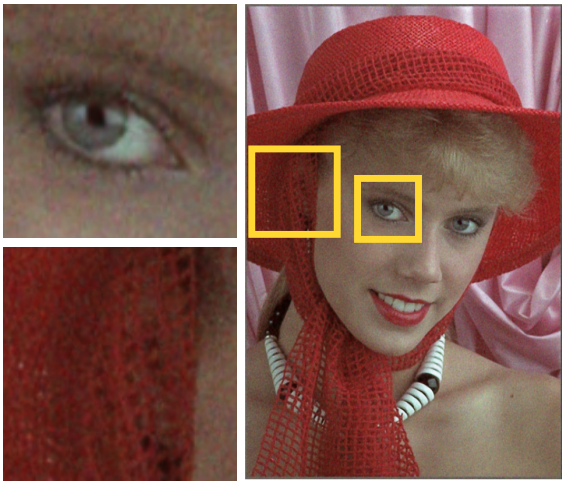


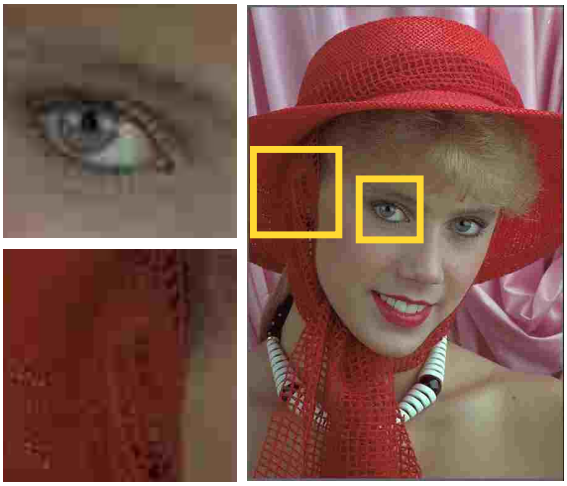
Figure 9. See Figure 7.



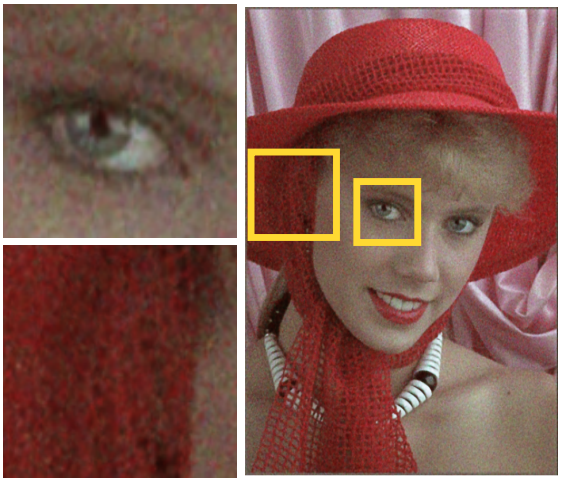
Original image



Ours (bpp=0.31, PSNR=30.62)



JPEG (bpp=0.31, PSNR=28.68)



Ours (bpp=0.17, PSNR=28.18)

JPEG (bpp=0.17, PSNR=22.47)

Figure 10. **Kodak visualization**. We compare PRANC and JPEG on image 4 of Kodak dataset at bpp=0.31 and 0.17

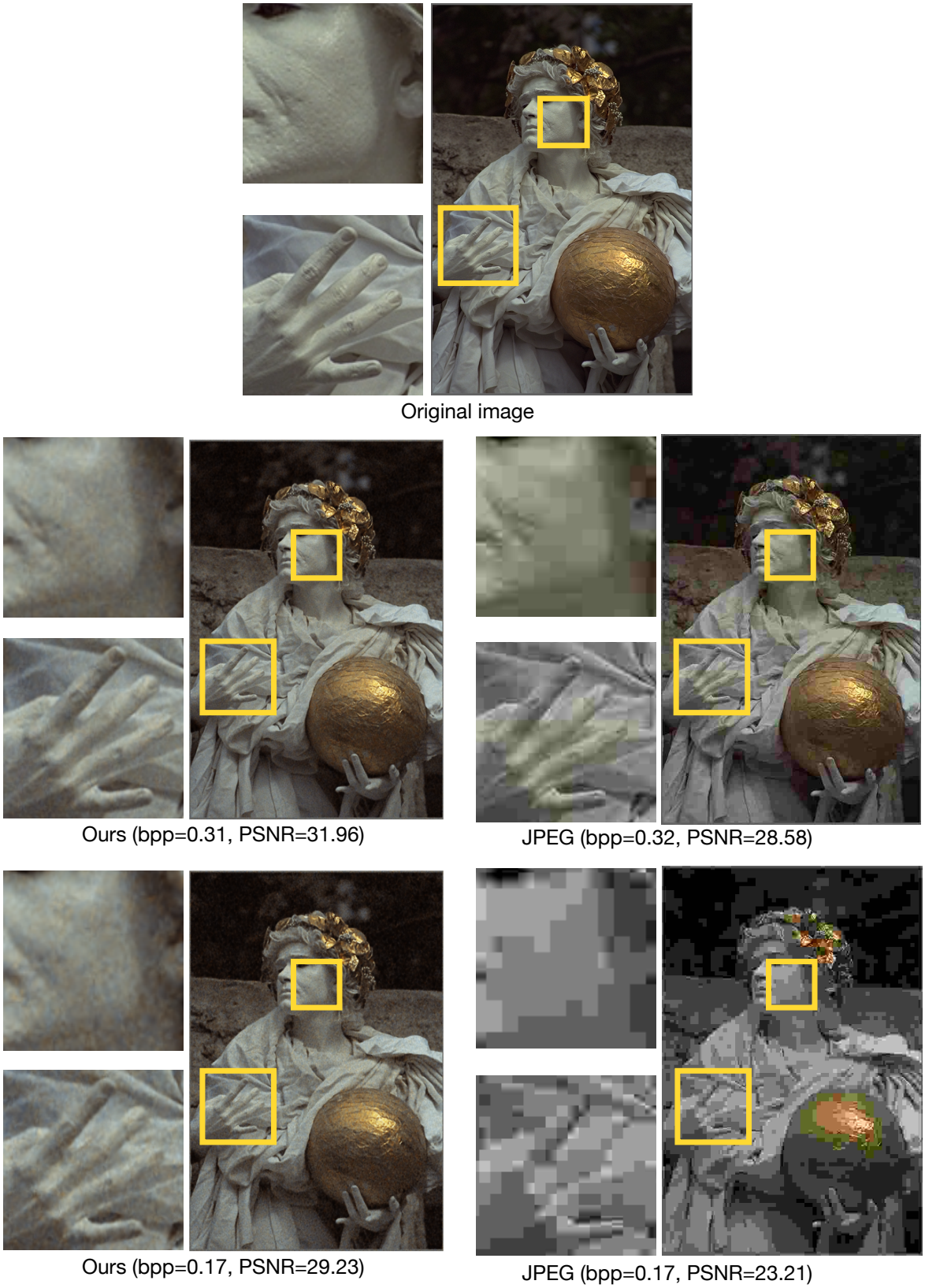


Figure 11. **Kodak visualization**. We compare PRANC and JPEG on image 17 of Kodak dataset.

Original image

Ours (bpp=0.31, PSNR=25.57)

JPEG (bpp=0.31, PSNR=21.34)

Ours (bpp=0.17, PSNR=23.3)

JPEG (bpp=0.21, PSNR=19.42)

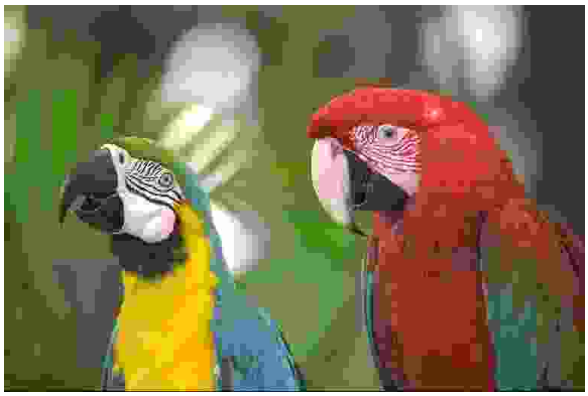
Figure 12. **Kodak visualization**. We compare PRANC and JPEG on image 5 of Kodak dataset.

Original image

Ours (bpp=0.31, PSNR=31.41)

JPEG (bpp=0.32, PSNR=31.16)

Ours (bpp=0.17, PSNR=27.88)



JPEG (bpp=0.17, PSNR=23.64)

Figure 13. **Kodak visualization**. We compare PRANC and JPEG on image 23 of Kodak dataset.

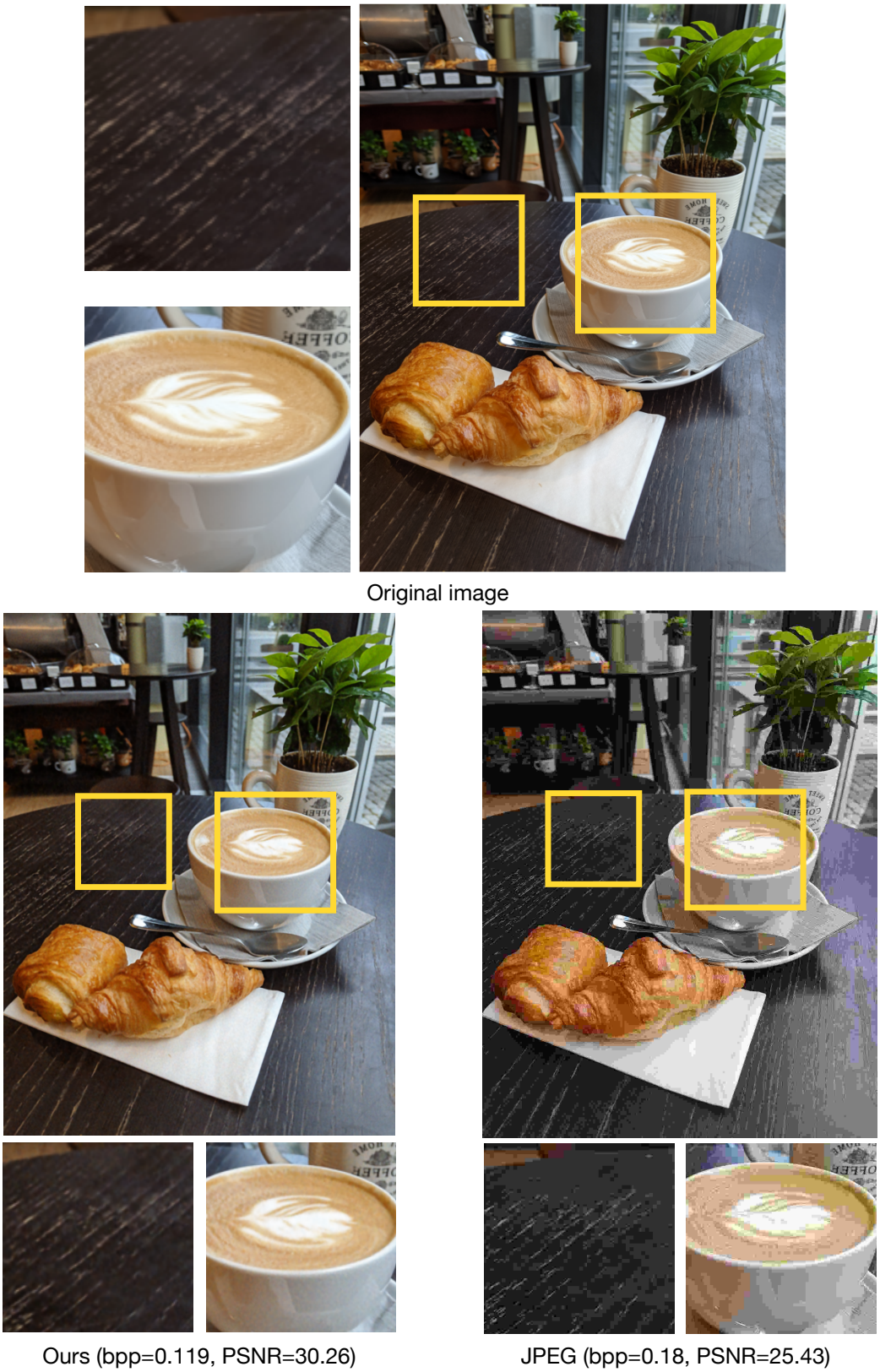
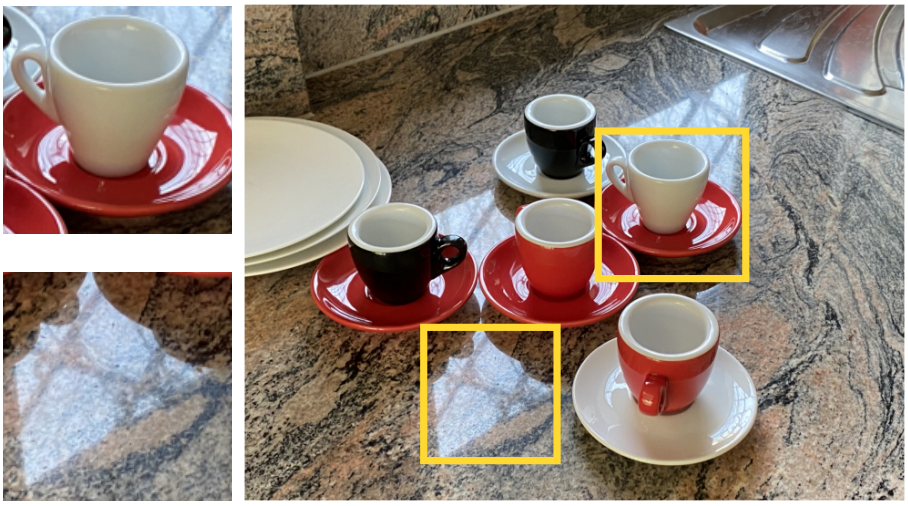
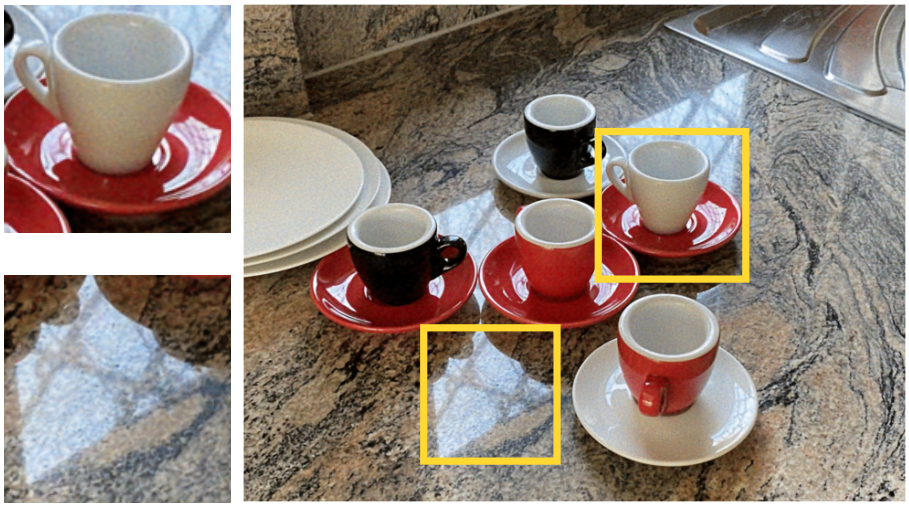


Figure 14. **CLIC visualization**. We compare PRANC at bpp=0.119 with JPEG at bpp=0.18 on a CLIC image.



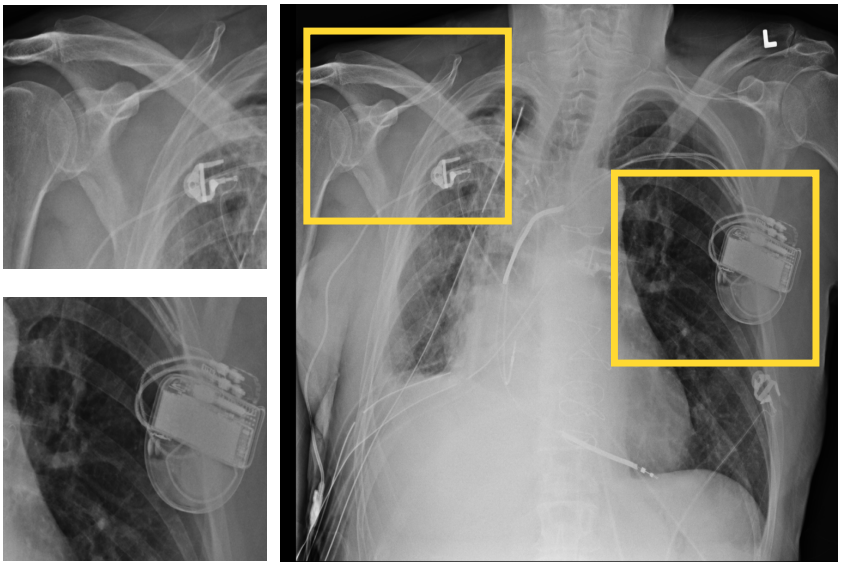
Original image



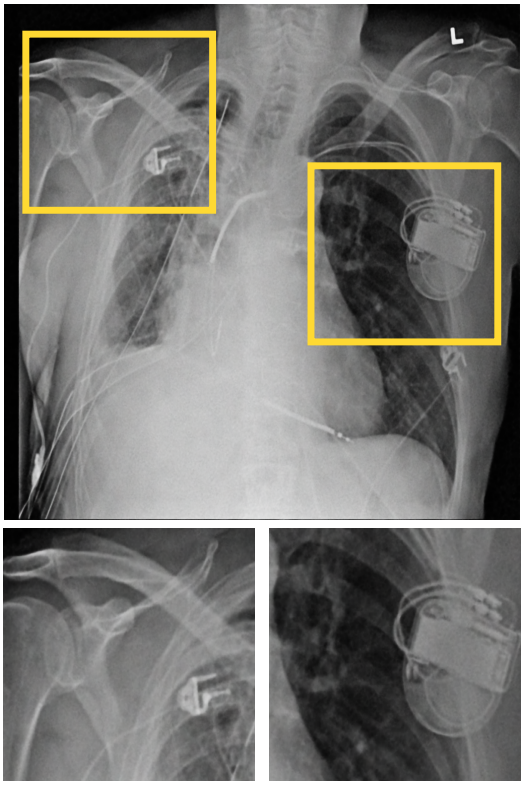
Ours (bpp=0.119, PSNR=28.99)

JPEG (bpp=0.226, PSNR=24.59)

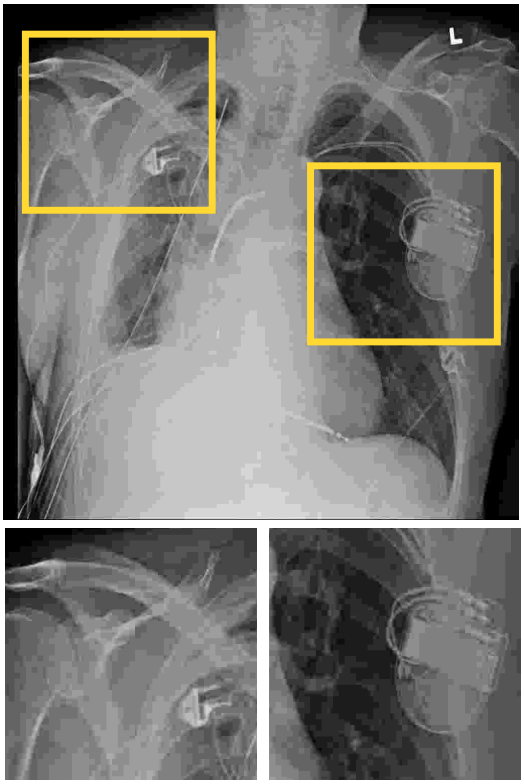
Figure 15. **CLIC visualization**. We compare PRANC at bpp=0.119 with JPEG at bpp=0.226 on a CLIC image.



Original image



Ours (bpp=0.15, PSNR=36.06)



JPEG (bpp=0.15, PSNR=32.11)

Figure 16. **Chest X-ray visualization**. We compare PRANC and JPEG on a Chest X-ray image at bpp=0.15

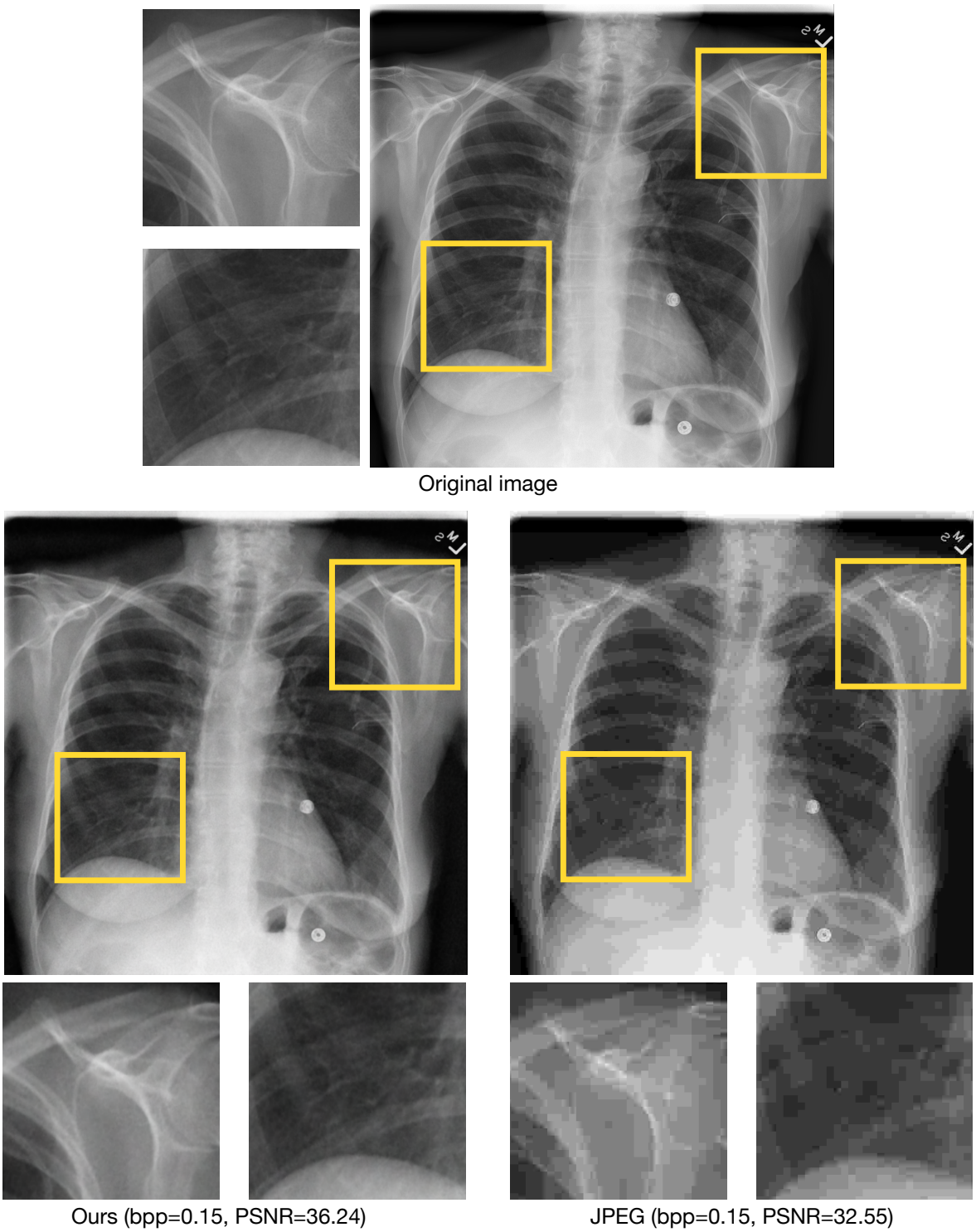
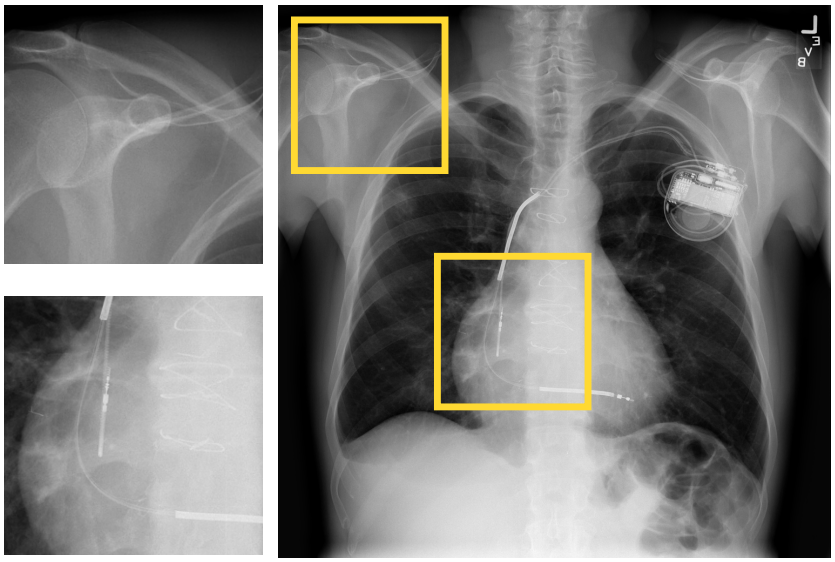
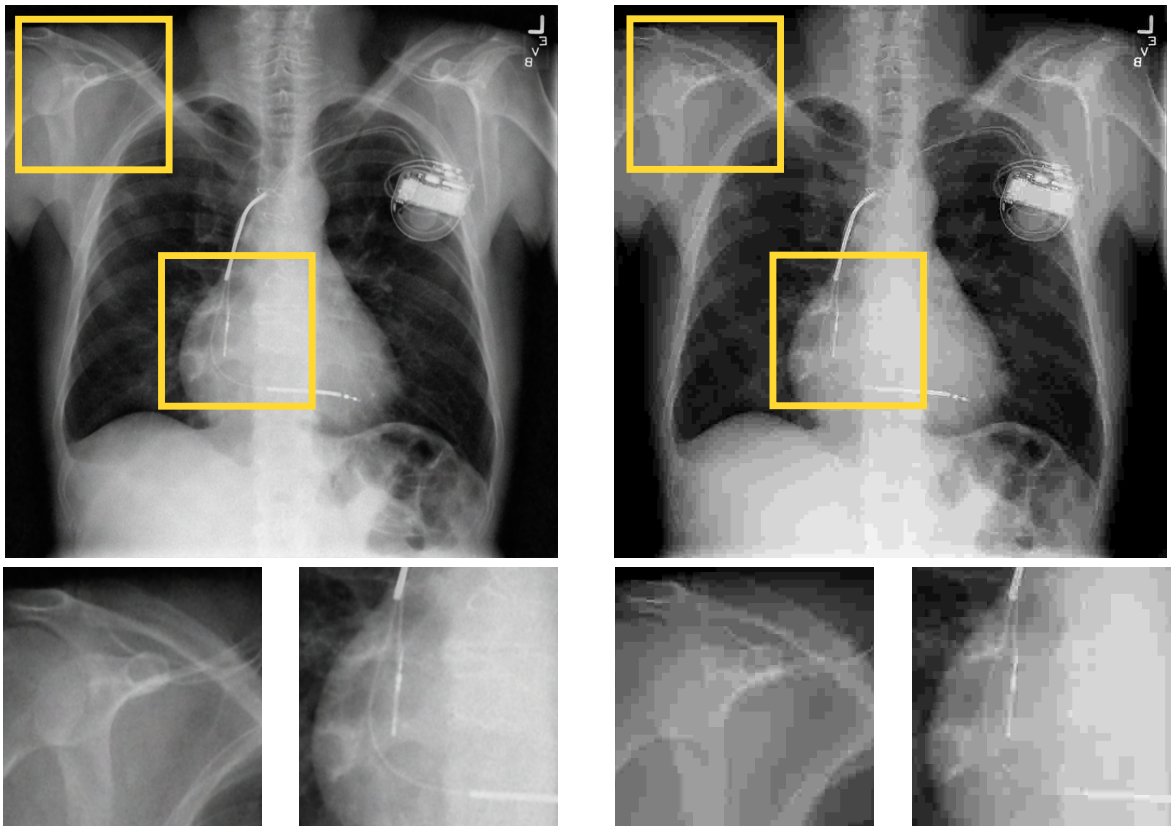


Figure 17. See Figure 16.



Original image



Ours (bpp=0.15, PSNR=35.54)

JPEG (bpp=0.15, PSNR=32.20)

Figure 18. See Figure 16.