Lossy Compression 2

Scenario

(a) High-freq → 0

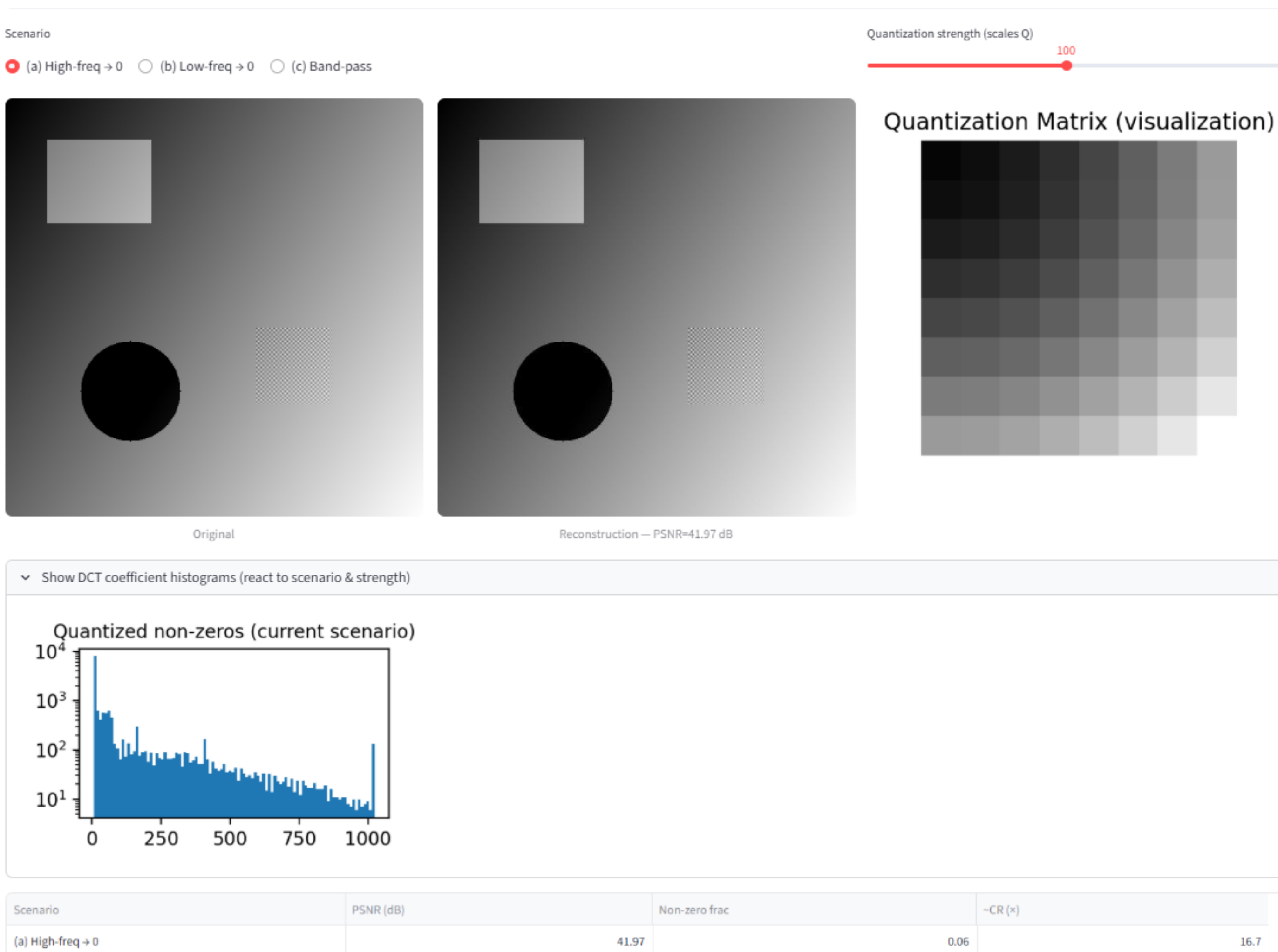


Fig.1

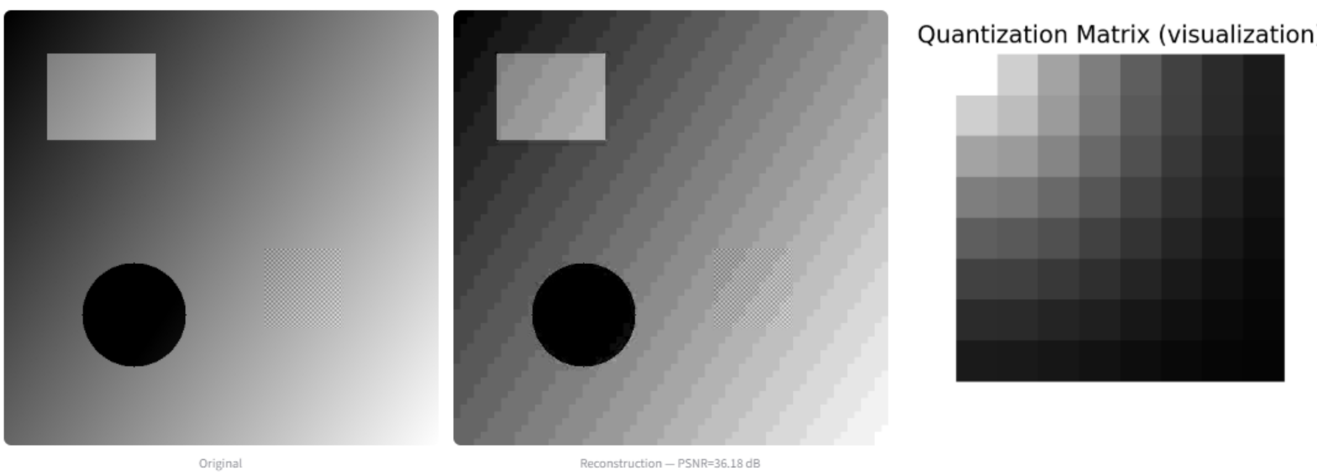
(b) Low-freq → 0

Scenario

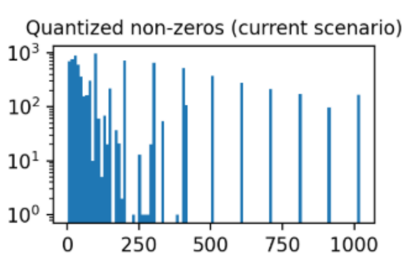
(a) High-freq → 0 (b) Low-freq → 0 (c) Band-pass

Quantization strength (scales Q)

100



Show DCT coefficient histograms (react to scenario & strength)



Scenario	PSNR (dB)	Non-zero frac	~CR (×)
(b) Low-freq → 0	36.18	0.033	30.1

Fig.2

(c) Band-pass

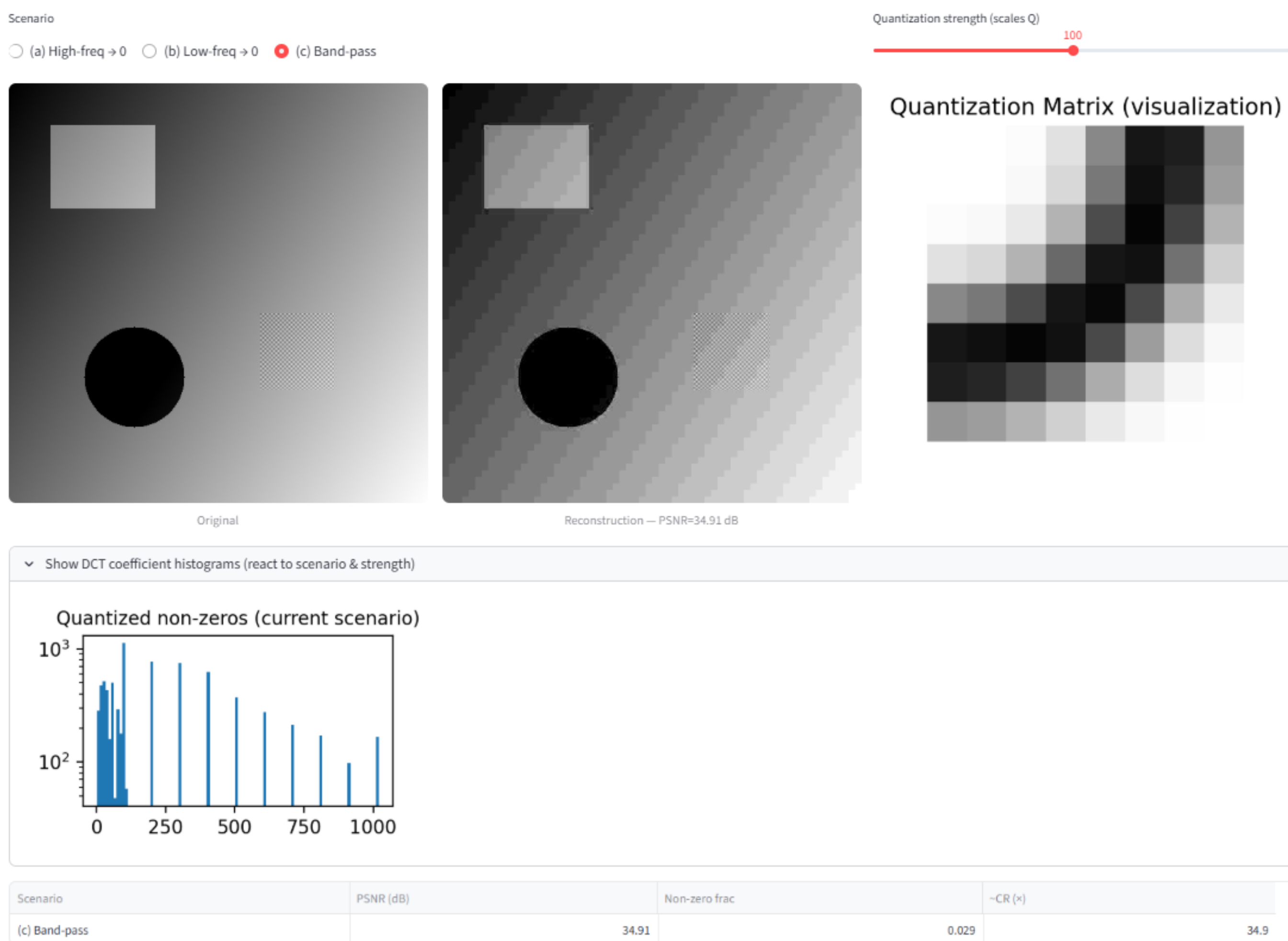


Fig.3

Discussion of Results

In this section I compare the three quantization scenarios that I implemented in the Streamlit app. For all screenshots the **quantization strength slider was set to 100**, so the comparison between the scenarios is fair.

Scenario (a): "Normal" compression – high frequencies go to 0 (Fig. 1)

In Scenario (a) I use a quantization matrix that **keeps low-frequency coefficients** (top-left of each 8×8 DCT block) and **penalizes high-frequency coefficients** (bottom-right) much more. This is similar to what JPEG does.

From Fig. 1 we can see:

- The **overall structure** of the image is almost unchanged.
  The gradient background still looks smooth.
- The bright square and dark circle keep their shapes, but the edges are **slightly softer** than in the original.
- The high-frequency checkerboard area has lost some fine detail and looks a bit blurred.

The PSNR reported by the app for this scenario with strength 100 is:

- **PSNR ≈ 41,97 dB** (from the table in the GUI)

The **non-zero fraction** of DCT coefficients is quite low, which means many coefficients have really become 0 after quantization. So we get **good compression** while still having **high visual quality**.

This behaviour makes sense: our eyes are less sensitive to very fine, high-frequency details, so we can throw away a lot of them without destroying the image.

Scenario (b): only low frequencies go to 0 (Fig. 2)

In Scenario (b): the quantization matrix now **penalizes low-frequency coefficients** and keeps high-frequency ones more. So, we are basically trying to remove the smooth, low-frequency content and keep edges and fine details.

From Fig. 2 we can see:

- The **global brightness and smooth gradient** are heavily distorted. The background does not look like a clean gradient anymore.
- The bright square and dark circle are still visible, but they look more “harsh” and not as natural.
- The checkerboard region still has some detail, because high frequencies are kept, but it sits on top of a strange background.

The PSNR for Scenario (b) at strength 100 is:

- **PSNR ≈ 36,18 dB**

Even if some details are still there, the overall image looks worse than in Scenario (a).
This tells us that **low-frequency information is very important** for human perception: once we destroy it, the image looks wrong, even if high-frequency stuff is preserved.

Scenario (c): high and low frequencies go to 0 (band-pass) (Fig. 3)

Scenario (c) uses a “band-pass” style quantization matrix. Here both very **low** and very **high** frequencies are penalized, and only **mid-frequency coefficients** are kept relatively more.

Looking at Fig. 3:

- The background gradient is again not correct, because low frequencies are damaged.
- The shapes are still visible, and edges are present, but the image looks a bit **flat** and not very natural.

- Some textures and fine details are also smoothed out, since the highest frequencies are also suppressed.

The PSNR for Scenario (c) at strength 100 is:

- **PSNR ≈ 34,91 dB**

This PSNR is the **lowest** of the three scenarios, which matches the visual impression: we are now throwing away information at *both* ends of the frequency spectrum, so the reconstruction gets further from the original.

Overall Comparison

If we compare the three scenarios:

- **Scenario (a)** (high-freq → 0)
  - Best visual quality and usually the **highest PSNR**.
  - Only fine details and noise are heavily compressed.
  - Image looks slightly smoother but still very natural.
- **Scenario (b)** (low-freq → 0)
  - Worse visual quality: background and global contrast look wrong.
  - PSNR is lower than (a).
  - Shows that removing low frequencies is a bad idea for normal images.
- **Scenario (c)** (band-pass)
  - Keeps mainly mid-frequencies, so edges exist but the image looks unnatural.
  - PSNR is the lowest of the three, which fits the visual result.

The **main takeaway** from these experiments is that:

For natural images, we should keep the **low-frequency content** with high precision and compress the **high-frequency content** more aggressively.

This is exactly what real formats like JPEG do. Scenario (a) therefore behaves most like a realistic lossy compression scheme, and the PSNR values and screenshots support that.