EE669 Homework #3

Yifan Wang
wang608@usc.edu

October 12, 2019

1 Problem 1: Scalar Quantization

1.1 Written Questions

(1)

Maximum possible value is 7, $(2^3 - 1)$, since it's a 3 bit image.

(2)

There are 2 quantization levels.

$$Init : x_0^{(0)} = 0, x_1^{(1)} = 7$$
$$t_0^{(0)} = \frac{x_0^{(0)} + x_1^{(0)}}{2} = 3.5$$
$$x_0^{(1)} = \frac{0 \times 0.125 + 1 \times 0.125 + 2 \times 0.125 + 3 \times 0.125}{0.125 + 0.125 + 0.125 + 0.125} = 1.5$$
$$x_1^{(1)} = \frac{4 \times 0.125 + 5 \times 0.125 + 6 \times 0.125 + 7 \times 0.125}{0.125 + 0.125 + 0.125 + 0.125} = 5.5$$

Since $(t_0^{(1)} = \frac{x_0^{(1)} + x_1^{(1)}}{2} = 3.5) == t_0^{(0)}$, it has already converged. Interval is

$$[0, 3.5], [3.5, 7]$$

Two corresponding reconstruction value is

$$x_0 = 1.5, x_1 = 5.5$$

(3)

5.5	1.5	5.5	1.5
1.5	1.5	1.5	5.5
5.5	1.5	5.5	1.5
5.5	1.5	5.5	5.5

Figure 1: Quantizer result

$$\textbf{MSE} = \frac{1}{N}\sum_{i=1}^{N}(X'(i) - X(i))^2$$

$$= \frac{1}{16}((5.5-5)^2 + (1.5-2)^2 + (5.5-4)^2 + (1.5-0)^2 + (1.5-1)^2 + (1.5-3)^2 + (1.5-2)^2 + (5.5-5)^2 +$$
$$(5.5-4)^2 + (1.5-3)^2 + (5.5-6)^2 + (1.5-2)^2 + (5.5-7)^2 + (1.5-2)^2 + (5.5-7)^2 + (5.5-6)^2)$$
$$= \frac{18}{16} = \mathbf{1.125}$$

$$\textbf{PSNR} = 10log_{10}(\frac{MAX^2}{MSE})$$
$$= 10log_{10}(\frac{7^2}{1.125}) = \textbf{16.39(dB)}$$

1.2 Programming

(1)

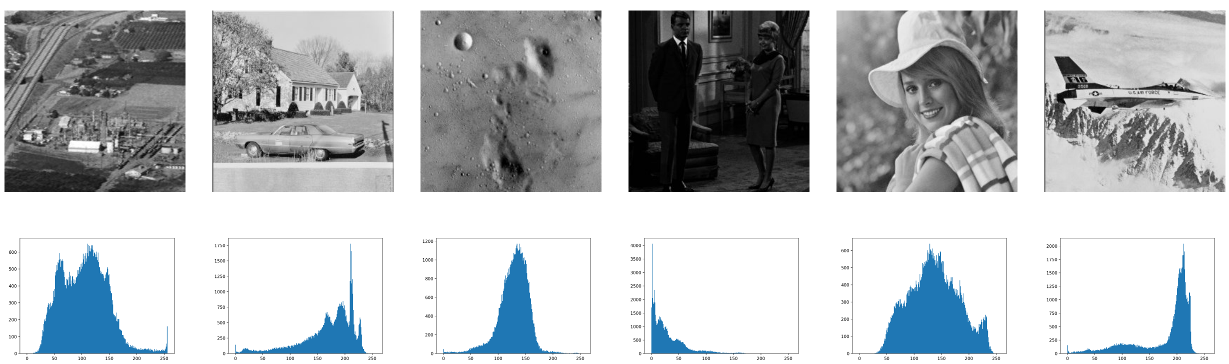

Figure 2: Raw image and it's corresponding histogram

(2)

iter	1	2	3	4	5	6	7	8	9
PSNR(dB)	29.1907	29.4195	29.6287	29.7851	29.9066	29.9883	30.0552	30.0977	30.1329
iter	10	11	12	13	14	15	16	17	18
PSNR(dB)	30.1703	30.1934	30.2256	30.2418	30.2493	30.2592	30.2724	30.2779	30.2827

Table 1: $3-bit$ SQ. PSNR for the training images at each iteration.

iter	1	2	3	4	5	6	7	8	9
PSNR(dB)	38.7820	39.0812	39.2550	39.3550	39.4358	39.5190	39.5979	39.6406	39.6825
iter	10	11	12						
PSNR(dB)	39.7184	39.7435	39.7505						

Table 2: $5-bit$ SQ. PSNR for the training images at each iteration.

			PSNR			
bits	$moon.256$	$chem.256$	$house.256$	$f16.256$	$couple.256$	$elaine.256$
3	31.4855	29.6479	30.0251	30.0518	21.3196	30.4929
5	43.3884	37.9541	39.6154	38.4052	29.8711	41.256

Table 3: SQ. PSNR for quantized image.

For initialization, it is better to initialize each center with a region have similar probability according to the PDF of train data, to make each code word follow a uniform distribution. So that I separate the training data to small region which has same probability, then using the exception of this region as initial center.

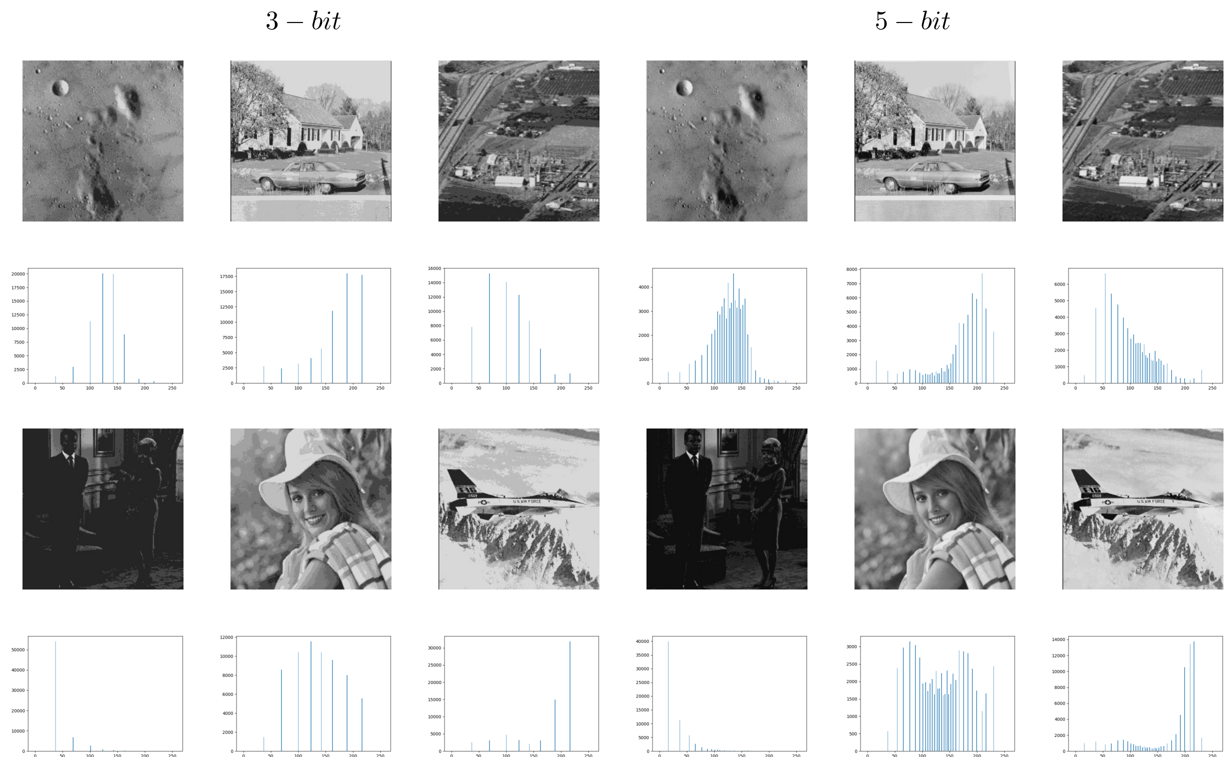

Figure 3: $3-bit$ and $5-bit$ results and corresponding histograms

(3)&**(4)**

Result is shown in $Figure3$. A good quantization is human cannot find much visual difference between the raw image and quantized one. Which means smooth transition between the bright region to dark region, no color blocks. Detail retained is another important factor, after quantization cannot remove too much detail, especially which is obvious to human. Besides, for some extreme cases, super dark or white region can use a low bit quantization, due to the insensitivity of human eye to these parts, which means a higher distortion and low bits may not affect visual effect.

(5)

Entropy	$chem.256$	$house.256$	$moon.256$	$couple.256$	$f16.256$	$elaine.256$
$3-bit$	2.671	2.600	2.300	0.857	2.235	2.869
$5-bit$	4.602	4.430	4.583	2.039	3.802	4.898

Table 4: Entropy for quantized image.

(6)

$chem.256$ $moon.256$ $house.256$ $couple.256$ $elaine.256$ $f16.256$

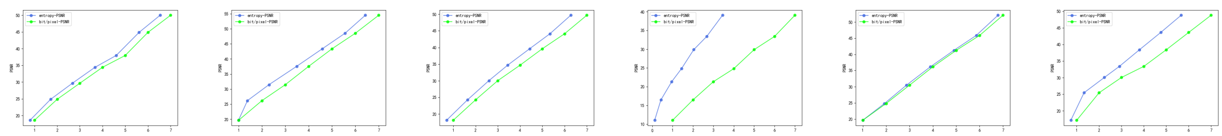

Figure 4: bits/pixel-PSNR curve and entropy-PSNR curve

(7)

Due to the training image, it would give more detail on middle grayscale rather than dark or white region. It would give best visual result for image which have similar histogram as the training data. However, due to the fact that human eyes do not so sensitive to dark or white region. When meeting image whose histogram is left shifted like $couple.256$ is seems relative nice when it is 5 bits. But due to the conflict of histogram, using lower bit would shift the dark region left which gives a super bad visual result than $elaine.256$ whose histogram is highly matched with training data. Similar result would happen to $f16.256$ whose histogram is right shifted. After quantization using lower bits, it would move the white region to left and resulting a worse visual experience.

2 Vector Quantization

2.1 Written Questions

ERROR MAP FOR (1,6)					
2	-5	4	1	3	-5
3	-5	5	-3	4	0
3	1	2	-5	1	-6
3	-4	1	0	2	0
-1	1	5	1	0	-1
1	-3	5	-6	4	-4

ERROR MAP FOR (3,3)					
0	-2	2	4	1	-2
1	-2	3	0	2	3
1	4	0	-2	-1	-3
1	-1	-1	3	0	3
-3	4	3	4	-2	2
-1	0	3	-3	2	-1

ERROR MAP FOR (5,2)					
-2	-1	0	5	-1	-1
-1	-1	1	1	0	4
-1	5	-2	-1	-3	-2
-1	0	-3	4	-2	4
-5	5	1	5	-4	3
-3	1	1	-2	0	0

ERROR MAP FOR (4,0)					
-1	1	1	7	0	1
0	1	2	3	1	6
0	7	-1	1	-2	0
0	2	-2	6	-1	6
-4	7	2	7	-3	5
-2	3	2	0	1	1

Figure 5

Result					
4	0	1	6	4	0
4	0	5	2	3	3
1	6	4	0	4	0
5	2	1	6	1	6
1	6	3	3	1	6
3	3	4	0	5	2

ERROR MAP					
-1	1	4	1	0	1
0	1	1	1	2	3
3	1	-1	1	-2	0
-1	0	1	0	2	0
-1	1	3	4	0	-1
-1	0	2	0	0	0

Figure 6

$$MSE = \frac{91}{36} = 2.528$$
$$PSNR = 12.87(dB)$$

2.2 Programming

(1)

Blocking and its reverse is realized in $VQ.hpp$.

(2) Standard VQ

Empirical entropy and average distortion values for both train and test images are shown in $Table5$. Visualized results is shown in $Figure7$.

Generally, standard vector quantization would give better performance with a larger code book size under same block size. Since more code words mains more bits for the quantized image. While for the same code book size, result of quantization would depend on image context and block size. A larger block trends to have a larger distortion and a worse visual result with mosaic like structure. Lots of detail are lost, especially for image like $chem.256$ which has many fine detail. A larger block would destroy them. The block contents should highly related to image context and resolution so that can standard vector quantization provide a better visual results.

Using a larger code book size, empirical entropy would increase as well. While for the same code book size, entropy would decrease with the increasing of block size. Since a larger block would lead to less information retained in final image (more parts trends to be similar when considering a small block), so that entropy would decrease.

(3) Tree Structured VQ

Empirical entropy and average distortion values for both train and test images are shown in $Table6$. Visualized results is shown in $Figure8$.

Tree structured VQ using statistic results rather than split image and find centroids would result a much better code book which resembles to separate from probability distribution into (almost) small equal probability parts. While other observation is the same as standard VQ

Comparison

For entropy, TSVQ's is less than Standard VQ's, one possible reason for that is TSVQ generates new cell based on statistic results rather than split blocks by local, which would result a more uniform distributed code book, so that entropy is much lower. While quantized result looks almost the same from my point of view. Corresponding distortion would slight decrease as well which is a result of better code book.

		$couple.256$	$elaine.256$	$f16.256$	$chem.256$	$moon.256$	$house.256$
				N=16			
$block = 2$	Average distortion	282.55	332.67	840.539	588.72	333.31	1030.81
	Empirical entropy	2.722	3.353	3.111	3.163	2.562	3.343
$block = 4$	Average distortion	2257.46	3135.91	7558.83	5589.15	2369.82	8255.14
	Empirical entropy	2.687	3.368	3.092	3.196	2.490	3.237
$block = 8$	Average distortion	17019.80	28978.87	56837.185	38373.84	18214.50	53459.81
	Empirical entropy	2.543	2.840	2.675	2.906	1.752	2.758
				N=32			
$block = 2$	Average distortion	188.62	234.87	530.43	462.84	255.76	742.58
	Empirical entropy	3.586	3.986	3.881	3.887	3.092	4.136
$block = 4$	Average distortion	1864.16	2260.66	5465.92	4623.80	1911.43	6821.13
	Empirical entropy	3.568	3.950	3.915	3.924	2.789	4.001
$block = 8$	Average distortion	15801.68	24171.66	52462.71	33680.08	12664.04	49879.40
	Empirical entropy	3.339	3.651	3.542	3.592	2.466	3.632
				N=64			
$block = 2$	Average distortion	129.07	167.79	352.62	352.57	197.79	529.85
	Empirical entropy	4.409	4.831	4.836	4.790	3.956	5.044
$block = 4$	Average distortion	1488.99	1785.98	4495.34	3976.50	1592.82	5812.33
	Empirical entropy	4.468	4.904	4.735	4.830	3.73	4.945
$block = 8$	Average distortion	14857.31	22222.85	50090.76	31430.98	12323.85	48073.58
	Empirical entropy	3.991	4.413	4.427	4.242	3.104	4.430

Table 5: Standard VQ

		$couple.256$	$elaine.256$	$f16.256$	$chem.256$	$moon.256$	$house.256$
				Bit Rate=4			
$block = 2$	Average distortion	349.29	299.33	661.90	527.63	286.31	907.35
	Empirical entropy	2.222	3.549	2.882	3.503	2.794	3.436
$block = 4$	Average distortion	2726.84	2621.41	5048.95	5041.77	2082.04	7072.22
	Empirical entropy	1.838	3.538	3.094	3.594	2.722	3.438
$block = 8$	Average distortion	15834.25	22364.39	34381.02	36643.10	16017.65	48695.61
	Empirical entropy	1.911	3.307	3.102	2.948	1.972	3.158
				Bit Rate=5			
$block = 2$	Average distortion	218.95	221.37	391.00	426.28	261.40	616.22
	Empirical entropy	2.852	3.865	3.718	3.905	2.988	4.249
$block = 4$	Average distortion	2108.25	1637.09	2553.01	4135.25	1929.25	5765.32
	Empirical entropy	2.286	4.192	4.264	4.317	2.881	4.410
$block = 8$	Average distortion	12495.37	14076.00	20827.19	31887.5	11636.03	42186.30
	Empirical entropy	2.162	4.607	3.879	4.170	2.868	4.082
				Bit Rate=6			
$block = 2$	Average distortion	136.64	149.69	149.69	284.57	192.95	365.69
	Empirical entropy	3.387	4.693	4.693	5.229	3.865	5.121
$block = 4$	Average distortion	1342.06	1260.84	1723.76	3645.27	1665.62	5061.79
	Empirical entropy	3.362	4.795	4.725	5.160	3.334	4.989
$block = 8$	Average distortion	10254.59	9130.60	11477.93	29589.38	11100.70	39676.80
	Empirical entropy	2.614	5.333	4.989	4.608	2.987	4.953

Table 6: Tree structured VQ

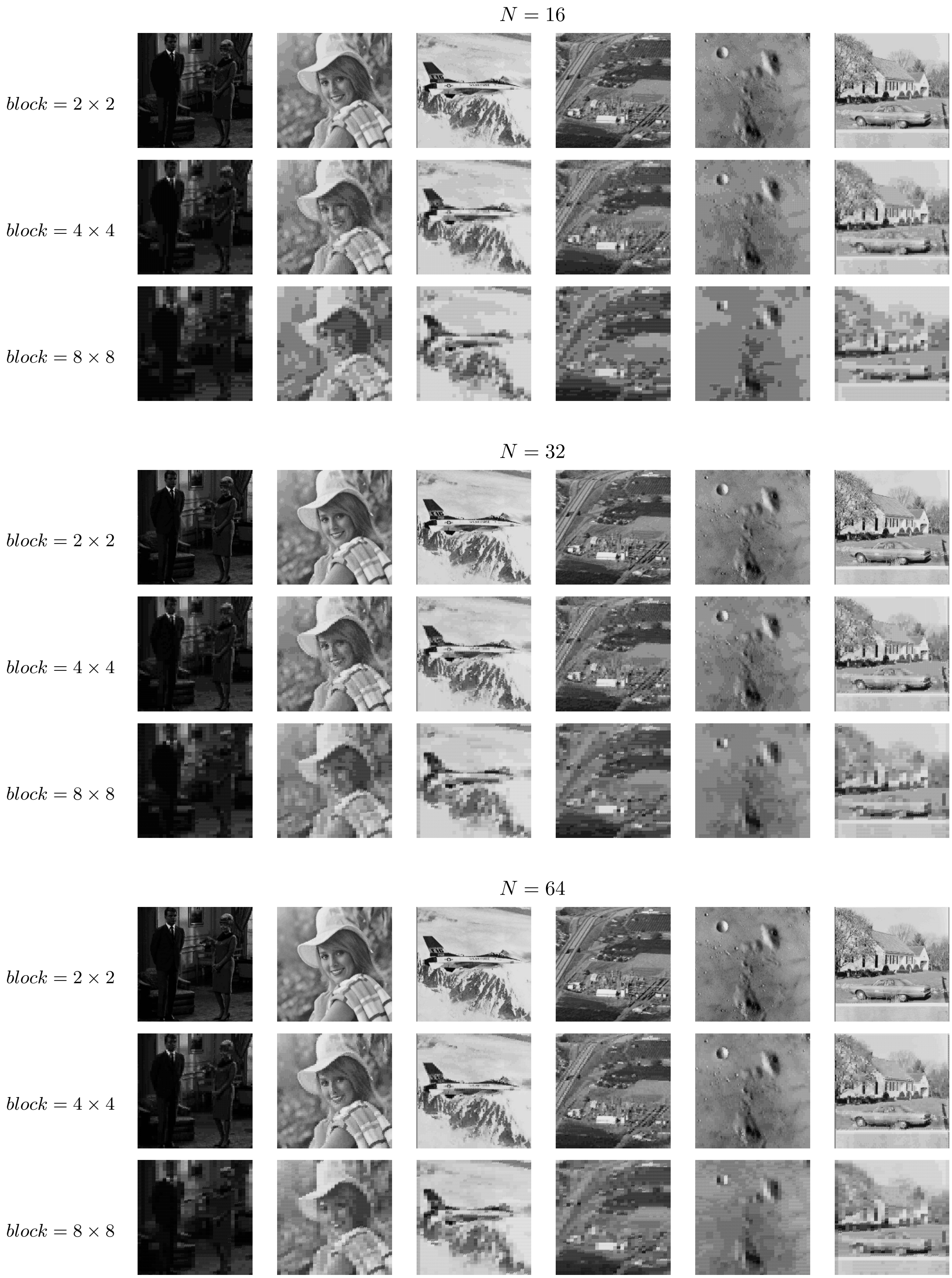

Figure 7: Results quantized by Standard VQ.

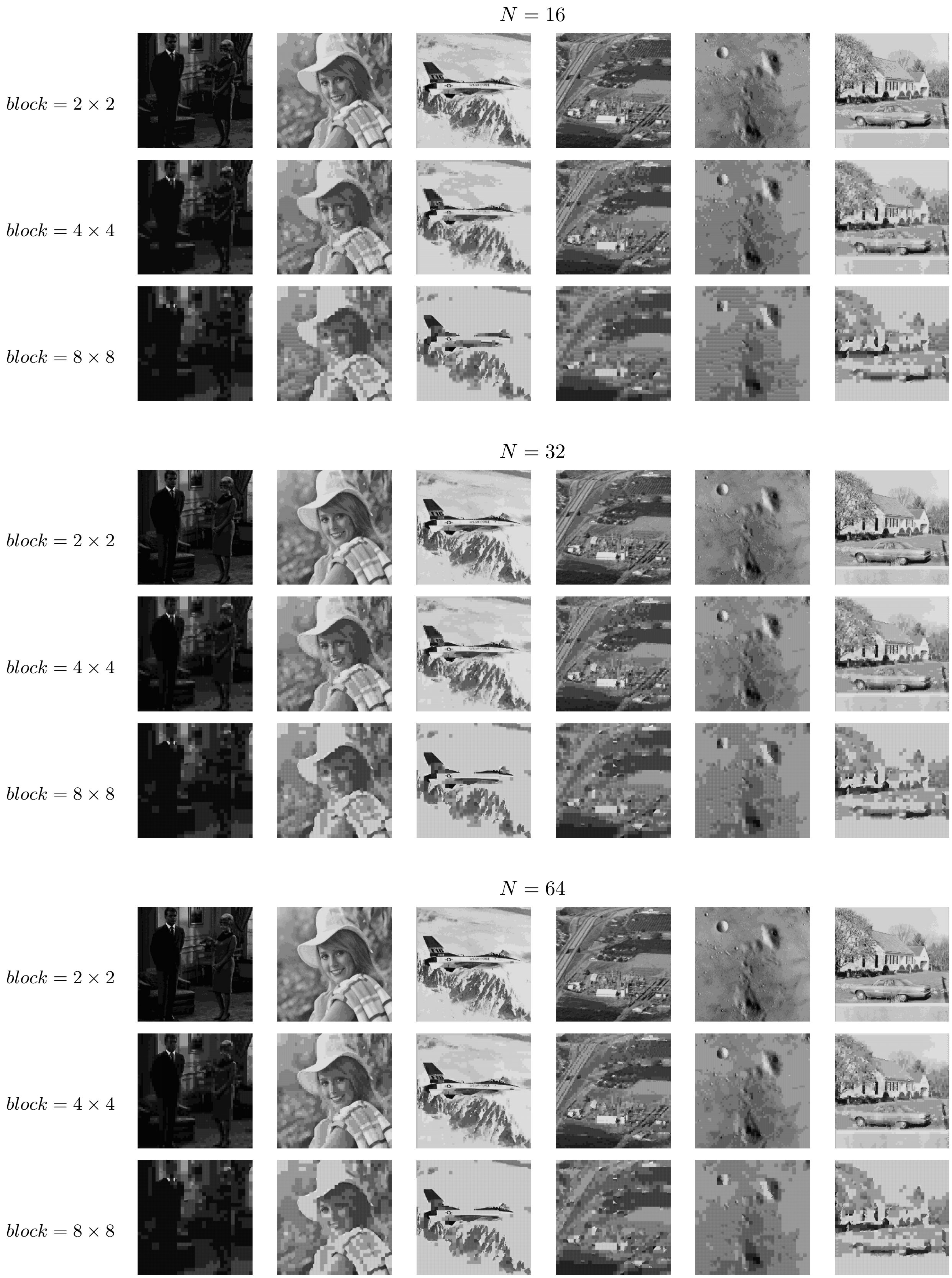

Figure 8: Results quantized by Tree Structured VQ.