ECE 5273
Test 2

Friday, May 15, 2026
10:30 AM - 12:30 PM

Spring 2026 Name: SOLUTION

Dr. Havlicek Student Num: ____________

Directions: This is an open notes test. You may use a clean copy of the course notes as published on the course web site. Other materials are not allowed. You have 120 minutes to complete the test. All work must be your own.

SHOW ALL OF YOUR WORK for maximum partial credit!

GOOD LUCK!

SCORE:

1. (20) ________

2. (20) ________

3. (20) ________

4. (20) ________

5. (20) ________

TOTAL (100):

____

On my honor, I affirm that I have neither given nor received inappropriate aid in the completion of this test.

Name: ____________ Date: ____________

1. **20 pts.** True or False. Mark *True* only if the statement is **always** true.

TRUE	FALSE	
	X	(a) **3 pts.** For large window sizes, the median filter is almost always implemented by pointwise multiplying the DFT's of the appropriately zero padded images. Median Filter is nonlinear
X		(b) **3 pts.** Special zero padding is needed to correctly implement a *zero-phase* 2D LTI filter by pointwise multiplication of DFT's. Notes pp. 5.72-5.75
X		(c) **3 pts.** ResNet is a deep learning network that uses "skip connections" to bypass one or more layers. Notes p. 11DL26-43
X		(d) **3 pts.** Baseline JPEG achieves image compression by performing lossy quantization of block DCT coefficients followed by lossless DPCM, run-length coding, and entropy coding. Notes p. 7.72
	X	(e) **3 pts.** The main reason for using normalized cross-correlation in gray scale template matching algorithms is that the normalized cross-correlation is not affected by scaling and rotation. Notes pp. 8.42, 8.45
X		(f) **3 pts.** The key innovation of the bilateral filter is that the weights for neighboring pixels are computed using *luminance similarity* in addition to the spatial distance weighting that is normally found in an LTI filter. Notes p. 6.79
OH	MY!	(g) **2 pts.** The acronym "DPCM" was invented in 2020 by the US Centers for Disease Control and Prevention as part of a social distancing campaign; it stands for: "**D**on't **P**ass **C**oronavirus, **M**an!"

2. **20 pts.** The 4×4 image **I** shown on the left below has 8-bit pixels in the range $0 \leq I(m,n) \leq 255$. This image is transmitted through a communication channel where it is corrupted by IID additive white Laplacian noise. The received image **J** is shown on the right below.

I =

32	32	32	32
120	120	32	32
120	120	120	32
120	120	120	120

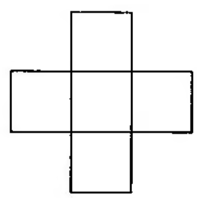
J =

30	32	28	35
125	134	39	36
117	115	107	27
123	120	122	119

(a) **15 pts.** Choose the best OS or morphological filter (MED, AVE, INNER_AVE, ERODE, DILATE, OPEN, CLOSE, *etc.*) to denoise the received image by attenuating the transmission noise. Use the structuring element/window **B** = CROSS(5) shown below. **Show** the denoised image **K** in the space provided at the bottom of this page. Handle edge effects by replication.

Median is best for Laplacian noise.

B =



Note: Work space is provided on the following page.

(b) **5 pts.** Compute the ISNR for the denoised image **K**.

Hint: $\text{ISNR}(\mathbf{K}) = 10\log_{10}\dfrac{\text{MSE}(\mathbf{J})}{\text{MSE}(\mathbf{K})}$

Show the denoised image here:

K =

30	32	32	35
125	115	39	36
117	117	107	36
123	120	120	119

ISNR = 2.1310 dB

Work Space for Problem 2…

$$MSE(J) = \frac{\Sigma(I-J)^2}{16} = \frac{557}{16} = 34.8125$$

$$MSE(K) = \frac{\Sigma(I-K)^2}{16} = \frac{341}{16} = 21.3125$$

$$ISNR = 10\log_{10}\frac{MSE(J)}{MSE(K)} = 10\log_{10}1.63343$$

$$= 2.1310\ dB$$

$J =$

	30	32	28	35	
30	30	32	28	35	35
125	125	134	39	36	36
117	117	115	107	27	27
123	123	120	122	119	119
	123	120	122	119	

$K =$

30	32	32	35
125	115	39	36
117	117	107	36
123	120	120	119

$I - J =$

2	0	4	-3
-5	-14	-7	-4
3	5	13	5
-3	0	-2	1

$I - K =$

2	0	0	-3
-5	5	-7	-4
3	3	13	-4
-3	0	0	1

$(I-J)^2 =$

4	0	16	9
25	196	49	16
9	25	169	25
9	0	4	1

$(I-K)^2 =$

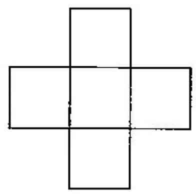
4	0	0	9
25	25	49	16
9	9	169	16
9	0	0	1

3. **20 pts.** The noisy 5×5 image $\mathbf{I}$ shown below has 4-bit pixels in the range $0 \leq I(m,n) \leq 15$:

$\mathbf{I} =$

4	8	12	15	13
11	15	4	9	4
10	5	**8**	2	13
3	9	11	2	4
2	4	14	4	15

Denoising is to be performed by applying a median filter with the window $\mathbf{B}$ = CROSS(5) shown here:

$\mathbf{B} =$ 

Use Delman's fast bit-serial median filtering algorithm to compute the filtered output value for the middle pixel (8) shown in large boldface type above.

Note: you do **not** have to apply the filter at any other pixels! Just apply it at the center pixel *only* to find the filtered output value for that pixel *only*. More workspace is provided on the following page in case you need it.

windowed set is $B \circ I(2,2) = \{4, 5, 8, 2, 11\}$

Notes pp. 6.56 - 6.58:

Original Data	First Change	Second Change	Third Change
0 1 0 0	0 1 0 0	0 1 0 0	0 1 0 0
0 1 0 1	0 1 0 1	0 1 0 1	0 1 0 1
1 0 0 0	1 1 1 1	1 1 1 1	1 1 1 1
0 0 1 0	0 0 1 0	0 0 0 0	0 0 0 0
1 0 1 1	1 1 1 1	1 1 1 1	1 1 1 1
↳ 0	↳ 1	↳ 0	↳ 1

MEDIAN = 0101_b = 5

4. **20 pts**. The 256×256 image $\mathbf{I}$ shown at left below has 8-bit pixels in the range $0 \leq I(m,n) \leq 255$. This image is transmitted through a communication channel where it is corrupted by IID additive white Gaussian noise. The received image $\mathbf{J}$ is at right below.

I

J

The following two filters are applied to perform denoising:

- A 256×256 Gaussian low-pass filter with space constant $\sigma = 1.25$, implemented exactly as described on pages 5.64-5.65 of the course notes,
- A bilateral filter with Gaussian weighting functions having $\sigma_G = 1.0$ and $\sigma_H = 0.67$.

Identify and label the two filtered images shown below:

Bilateral Filter

- Sharp Edges
- No Blur
- Removes Noise Better

Gaussian

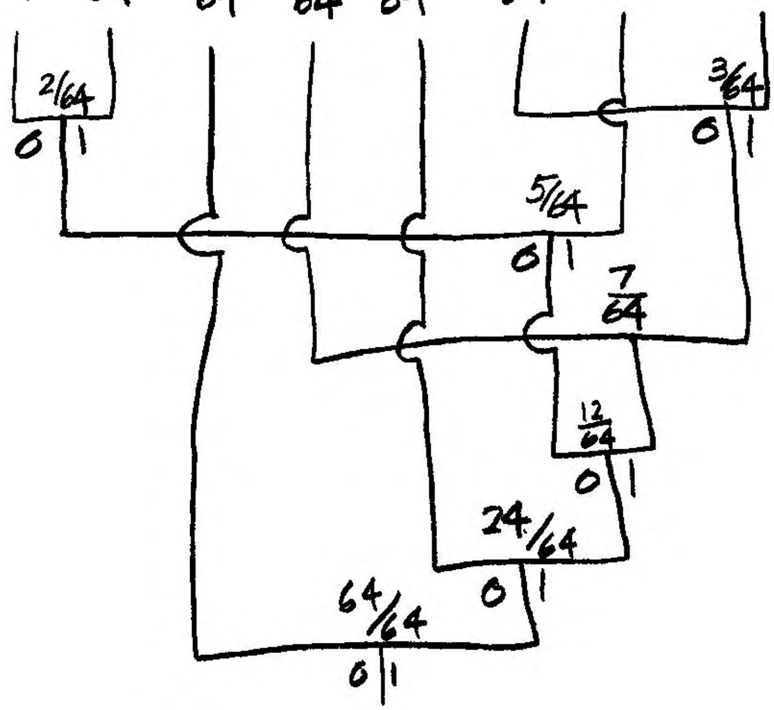
- Blurs Image
- Linear convolution leaves dark border

5. **20 pts.** Gray scale digital images **I** with 3 bits per pixel and gray levels in the range $\{0, 1, \ldots, 7\}$ are modeled as coming from an information source with the following source symbol probabilities (normalized histogram):

k	0	1	2	3	4	5	6	7
$p_{\mathbf{I}}(k)$	$\frac{1}{64}$	$\frac{1}{64}$	$\frac{5}{8}$	$\frac{1}{16}$	$\frac{3}{16}$	$\frac{1}{32}$	$\frac{3}{64}$	$\frac{1}{64}$

(a) **12 pts.** Design a Huffman code to encode these images.

K	0	1	2	3	4	5	6	7
$P_I(K)$	$\frac{1}{64}$	$\frac{1}{64}$	$\frac{5}{8}$	$\frac{1}{16}$	$\frac{3}{16}$	$\frac{1}{32}$	$\frac{3}{64}$	$\frac{1}{64}$
	$\frac{1}{64}$	$\frac{1}{64}$	$\frac{40}{64}$	$\frac{4}{64}$	$\frac{12}{64}$	$\frac{2}{64}$	$\frac{3}{64}$	$\frac{1}{64}$



K	0	1	2	3	4	5	6	7
C(K)	11000	11001	0	1110	10	11110	1101	11111
L[C(K)]	5	5	1	4	2	5	4	5

Problem 5 cont...

(b) **4 pts.** Find the expected BPP (bits per pixel) and CR (compression ratio) for the coded images $C(\mathrm{I})$.

$$BPP(I) = 3$$

$$BPP(\hat{I}) = \underbrace{5\left(\tfrac{1}{64}\right)}_{0} + \underbrace{5\left(\tfrac{1}{64}\right)}_{1} + \underbrace{1\left(\tfrac{40}{64}\right)}_{2} + \underbrace{4\left(\tfrac{4}{64}\right)}_{3} + \underbrace{2\left(\tfrac{12}{64}\right)}_{4} + \underbrace{5(2/64)}_{5} + \underbrace{4(3/64)}_{6} + \underbrace{5\left(\tfrac{1}{64}\right)}_{7}$$

$$= \frac{5+5+40+16+24+10+12+5}{64}$$

$$= \frac{117}{64} = \underline{\underline{1.82813}}$$

$$CR = \frac{BPP(I)}{BPP(\hat{I})} = \frac{3 \cdot 64}{117} = \underline{\underline{1.64103}}$$

(c) **4 pts.** Does your code achieve the theoretical lower bound on BPP for the coded images? Explain why or why not.

<u>No</u> because some of the symbol probabilities are not integer powers of 2...

e.g. $\frac{12}{64}$. //