COMP 322: Midterm 1 Study Guide

Computer Vision and Image Processing · Fall 2026

Prof. Hong Xu

Work through the exercises below as practice for Midterm 1. Solutions are at the end of this document—try each problem before looking.

1. Topics Covered

- **Images as signals:** digital image representation; pixels; images as discrete 2D arrays / signals.
- **Filtering:** linear vs. nonlinear filters; box / averaging filters; Gaussian smoothing; median filtering.
- **Correlation and convolution:** definitions; flipping the kernel; when they coincide; hand computation on small patches.
- **Filter properties:** linearity; commutativity / associativity of convolution; low-pass vs. high-pass; derivative / Sobel-style kernels.
- **Separability:** meaning; computational savings of $1 \times k$ then $k \times 1$ vs. $k \times k$.
- **NumPy for images:** slicing; boolean masking; broadcasting; reading short image-processing snippets.
- **Sampling and aliasing:** subsampling artifacts; low-pass before downsample; image pyramids; Nyquist intuition.
- **Fourier analysis:** sines/cosines as bases; low vs. high spatial frequency; convolution theorem; filtering in the frequency domain.
- **Edge detection:** finite-difference derivatives; noise and derivative of Gaussian; gradient magnitude / orientation; Canny pipeline (NMS, hysteresis); effect of σ.

2. Practice Exercises

1. Why does Canny use two thresholds? Give one failure mode if the high threshold is too high, and one if the low threshold is too low.
2. A 4×4 grayscale patch is
$$\begin{bmatrix} 10 & 2 & 8 & 1 \\ 3 & 9 & 4 & 7 \\ 6 & 0 & 5 & 2 \\ 1 & 8 & 3 & 9 \end{bmatrix}.$$
Ignore boundaries. Compute the 2×2 result of a 3×3 **median** filter (valid centers only).
3. A 1D gradient-magnitude profile across an edge is $[2, 5, 9, 6, 3]$, and orientation says the edge is vertical (so NMS compares left/right neighbors). Which indices survive non-maximum suppression (1-based: positions 1..5)? State a boundary rule if needed.
4. **NumPy.** Assume `H` and `W` are divisible by 4 and `img.shape == (H,W)`. Write loop-free code that replaces every non-overlapping 4×4 block by the block's mean, producing shape `(H//4, W//4)`.

5. List the main stages of the Canny detector in order. For hysteresis, explain the roles of the high and low thresholds. What happens to detected edges if Gaussian σ is increased a lot?

6. Explain the difference between correlation and convolution on a **non-symmetric** kernel in one or two sentences. Then, for $I = \begin{bmatrix} 2 & 1 \\ 0 & 3 \end{bmatrix}$ and $K = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$, compute the output of correlation with K aligned to I, and the output of convolution (correlation with K rotated 180°). Show the flipped kernel.

7. A filter is $K = uv^\top$ with $u, v \in \mathbb{R}^7$. Roughly how many multiply-adds per output pixel for separable filtering vs. full 7×7 convolution (ignore boundaries)?

8. True/False; if false, correct the statement in one sentence.
   (a) Convolution is associative.
   (b) Correlation is always identical to convolution.
   (c) Averaging / box filters are typically high-pass.
   (d) Median filtering is a linear operation.

9. Write a short function that, given RGB `img`, returns a copy where the top half keeps only the red channel (G=B=0) and the bottom half keeps only the blue channel (R=G=0). Then state what happens to a pure green pixel $(0, 255, 0)$ in each half.

10. Why might you prefer a Gaussian blur over a box blur before estimating image derivatives? Give one computational or qualitative reason one might still use a box filter in practice.

11. Why is $\partial_x(G_\sigma * I) = (\partial_x G_\sigma) * I$ the preferred way to estimate derivatives of noisy images?

12. In an image's Fourier representation, what do low spatial frequencies tend to encode vs. high spatial frequencies? Give one example low-pass kernel and one example high-pass / edge kernel from class.

13. State the convolution theorem in the frequency domain (words or notation). List the steps to apply a filter in the frequency domain. Name one artifact of using a sharp ideal low-pass cutoff in frequency.

14. Apply a 3×3 box filter (all weights 1/9) to the center pixel of

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 8 & 1 \\ 1 & 1 & 1 \end{bmatrix}.$$

    What is the output value at the center? Qualitatively, what did the filter do?

15. Why can FFT-based convolution help for very large kernels, while a separable spatial Gaussian of modest width is often still faster in practice?

16. Show that if a kernel K satisfies $K(r, c) = K(-r, -c)$ (180° symmetry about its center), then correlation with K equals convolution with K.

17. Define *signal* as used in this course in one sentence. Then explain why a digital image is a discrete 2D signal, and what a single grayscale pixel stores (photometrically, at a high level).

18. Describe precisely what the following returns (assume `img` is RGB with shape `(H,W,3)`):

```
def mystery(img):
    out = img.copy()
    out[:, : img.shape[1]//2, 1] = 0
    out[:, img.shape[1]//2 :, 2] = 0
    return out
```

19. **NumPy.** `img` is RGB `uint8` with shape `(H,W,3)`. Using broadcasting, compute grayscale $0.3R + 0.6G + 0.1B$ as floats. Then, on a copy of `img`, set every pixel whose grayscale value is < 40 to black $(0, 0, 0)$ using a mask.

20.
```
def pool2(img):
    H, W = img.shape
    out = np.zeros((H//2, W//2))
    for i in range(out.shape[0]):
        for j in range(out.shape[1]):
            out[i,j] = np.mean(img[2*i:2*i+2, 2*j:2*j+2])
    return out
```

   (a) Describe what `pool2` does.

   (b) Compute `pool2` on $\begin{bmatrix} 0 & 2 & 4 & 6 \\ 8 & 10 & 12 & 14 \\ 1 & 1 & 1 & 1 \\ 3 & 3 & 3 & 3 \end{bmatrix}$.

21. The Nyquist limit says that to represent a 1D frequency f without aliasing, you generally need a sampling rate strictly greater than $2f$ (i.e. more than two samples per cycle). Using that reminder: if an image contains fine checkerboard texture near the Nyquist frequency of the pixel grid, why does $2\times$ downsampling without blur often look awful? What fix does class recommend?

22. Let `img` be a NumPy array of shape `(H, W)`.

   (a) Write a slicing expression that downsamples by taking every third row and every third column, starting at $(0, 0)$.

   (b) Using boolean masking only, set all values of `img` that are strictly greater than 200 to 200.

   (c) `a` has shape `(3,1)` and `b` has shape `(1,5)`. What is the shape of `a+b`?

23. Explain separability of a 2D kernel. Give the rough operation-count comparison (ignore boundaries) for filtering an $N \times N$ image with a $k \times k$ kernel vs. two 1D passes of length k.

24. Explain aliasing when shrinking an image by throwing away pixels. What preprocessing should you apply first, and why (frequency language)? What is an image pyramid?

25. Natural-image DFTs often have most energy near DC (center / origin, depending on visualization). Why? If you instead keep only high frequencies and invert, what should you roughly see?

26. For the 1D intensity signal $f = [4, 4, 4, 4, 10, 10]$, compute forward differences $f[i+1] - f[i]$. Where is $|f'|$ largest? If i.i.d. noise is added to f, why are naive differences unreliable, and what remedy uses a derivative of a Gaussian?

27. Salt-and-pepper noise vs. mild Gaussian noise: which of median vs. Gaussian smoothing is usually better for each? How can median filtering hurt thin line drawings?

28. For
$$I = \begin{bmatrix} 3 & 1 & 4 \\ 1 & 5 & 9 \\ 2 & 6 & 5 \end{bmatrix}, \quad K = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix},$$
compute **correlation** and **convolution** at the center of I. Show the 180°-rotated kernel used for convolution.

29. Let `grid = np.arange(1, 65).reshape(8, 8)`. Using only slicing (no `np.flip`), construct: (i) reverse rows; (ii) the 4×4 center; (iii) every other column starting at index 1; (iv) the anti-diagonal (diagonal in reverse order) as a 1D array (advanced indexing allowed for (iv) only).

30. Consider the 1D filter kernel $[1,\ -2,\ 1]$. What do you think this filter does to a signal or image row? Is it primarily low-pass or high-pass? Briefly justify.

31. Let I_x be the horizontal image derivative (how intensity changes as you move right) and I_y be the vertical image derivative (how intensity changes as you move down), each obtained by filtering the image with an appropriate derivative kernel. Write formulas for gradient magnitude and orientation in terms of I_x and I_y. Why does thresholding the gradient magnitude alone usually give thick edges?

32. Correlate the image
$$\begin{bmatrix} 1 & 4 & 2 \\ 0 & 5 & 3 \\ 2 & 1 & 6 \end{bmatrix}$$
with
$$K = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$
at the center pixel only. Name the filter family this K belongs to (e.g. Sobel), and say whether it is low-pass or high-pass.

3. Solutions

1. High marks confident edges; low allows weak continuation of strong contours. High too high: broken / missing edges. Low too low: noisy speckles survive.

2. Valid centers are the inner 2×2 positions. Neighborhoods (sorted medians): center at $(1,1)$: $\{10,2,8,3,9,4,6,0,5\} \to 4$; $(1,2)$: $\{2,8,1,9,4,7,0,5,2\} \to 2$; $(2,1)$: $\{3,9,4,6,0,5,1,8,3\} \to 4$; $(2,2)$: $\{9,4,7,0,5,2,8,3,9\} \to 5$. Result $\begin{bmatrix} 4 & 2 \\ 4 & 5 \end{bmatrix}$.

3. Compare each interior point to neighbors: 5 loses to 9; 9 beats 5 and 6; 6 loses to 9. Ends: if they require a missing neighbor, typically suppressed or special-cased; commonly only index 3 (9) clearly survives.

4.
```
out = img.reshape(H//4, 4, W//4, 4).mean(axis=(1, 3))
```

5. Smooth/derivative-of-Gaussian $\to$ magnitude & orientation $\to$ non-maximum suppression $\to$ hysteresis (high: strong seeds; low: weak candidates kept if connected to strong). Larger σ: smoother, fewer, more blurred/shifted edges; fine detail lost.

6. Convolution = correlation with the 180°-rotated kernel; they differ unless K equals its rotation. Correlation: $1 \cdot 2 + 2 \cdot 1 + 3 \cdot 0 + 4 \cdot 3 = 2 + 2 + 0 + 12 = 16$. Flipped K: $\begin{bmatrix} 4 & 3 \\ 2 & 1 \end{bmatrix}$. Convolution: $4 \cdot 2 + 3 \cdot 1 + 2 \cdot 0 + 1 \cdot 3 = 8 + 3 + 0 + 3 = 14$.

7. Separable: about $7 + 7 = 14$ per pixel (two 1D passes). Full: 49 per pixel.

8. (a) T. (b) F — only when the kernel is unchanged by 180° rotation (centered). (c) F — they are low-pass smoothers. (d) F — median is nonlinear.

9.
```
def half_rb(img):
    out = np.zeros_like(img)
    h = img.shape[0]//2
    out[:h, :, 0] = img[:h, :, 0]
    out[h:, :, 2] = img[h:, :, 2]
    return out
```

   Green $(0, 255, 0)$ becomes black in both halves (neither half keeps G).

10. Gaussian has a smoother frequency response and matches the derivative-of-Gaussian pipeline (less ringing / nicer scale-space behavior). Box filters are cheaper and easy to implement / accumulate.

11. Differentiation amplifies noise; smoothing first (equivalently, correlating with a derivative-of-Gaussian) stabilizes the estimate. Linearity lets the operators commute.

12. Low: smooth shading / large structure. High: edges, texture, noise. Low-pass: Gaussian or box. High-pass: Sobel / simple derivative kernels.

13. Convolution in space $\leftrightarrow$ multiplication in frequency ($\mathcal{F}(f * g) = \mathcal{F}f \cdot \mathcal{F}g$, up to convention). Steps: FFT(image), form FFT(kernel) or mask, multiply, IFFT. Hard cutoffs cause ringing.

14. Eight neighbors are 1 and the center is 8, so the sum is 16 and the mean is $16/9$. The bright spike is strongly smoothed into its surroundings.

15. FFT turns wide spatial kernels into pointwise products (better asymptotic scaling for huge k), but has large constants / padding overhead. Separable Gaussian is only $O(k)$ per pixel with tiny constants.

16. Convolution is correlation with the rotated kernel $K^{\text{rot}}(r,c) = K(-r,-c)$. If $K = K^{\text{rot}}$, the two operations match.

17. A signal is a (multi-dimensional) function containing information about a phenomenon. A camera samples brightness on a grid, so the image is a discrete 2D signal. A grayscale pixel stores a quantized intensity / light measurement at that sample location.

18. Left half: green channel zeroed (red/blue kept). Right half: blue channel zeroed (red/green kept). Spatial split is vertical (by columns).

19.
```
w = np.array([0.3, 0.6, 0.1])
gray = (img.astype(float) * w).sum(-1)
out = img.copy()
out[gray < 40] = (0, 0, 0)
```

20. (a) 2×2 mean downsample (average pooling). (b) $\begin{bmatrix} 5 & 9 \\ 2 & 2 \end{bmatrix}$ because averages are $(0+2+8+10)/4 = 5$, $(4+6+12+14)/4 = 9$, $(1+1+3+3)/4 = 2$, $(1+1+3+3)/4 = 2$.

21. Downsampling halves the representable frequency; content above the new Nyquist aliases. Blur (low-pass) first to remove frequencies the coarser grid cannot carry, then subsample.

22. (a) `img[::3, ::3]`. (b) `img[img > 200] = 200`. (c) `(3,5)`.

23. Separable means $K = uv^\top$. Full 2D: $\Theta(N^2k^2)$ multiply-adds; separable: $\Theta(N^2k)$ for each pass, $\Theta(N^2k)$ overall (constant factor 2).

24. High-frequency content folds into lower frequencies under coarse sampling, creating jagged / Moire artifacts. Low-pass (blur) first to remove frequencies the new grid cannot represent. A pyramid repeatedly blurs and subsamples to build a multi-scale stack.

25. Images vary slowly over large regions $\Rightarrow$ low-frequency energy. High-pass residual looks like edges/texture with little smooth shading.

26. Differences: $[0,0,0,6,0]$. Largest at the step $4 \to 10$. Noise creates spurious jumps; convolve with $\partial_x G_\sigma$ (or blur then differentiate).

27. Median for salt-and-pepper; Gaussian for additive Gaussian noise. Median can erase thin strokes / sharp corners treated like outliers.

28. Correlation: $1\cdot3+0\cdot1+(-1)\cdot4+1\cdot1+0\cdot5+(-1)\cdot9+1\cdot2+0\cdot6+(-1)\cdot5 = 3-4+1-9+2-5 = -12$. Rotate K by 180°: $\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$. Convolution: $(-1)3+0+1\cdot4+(-1)1+0+1\cdot9+(-1)2+0+1\cdot5 = -3+4-1+9-2+5 = 12$.

29. (i) `grid[::-1, :]`; (ii) `grid[2:6, 2:6]`; (iii) `grid[:, 1::2]`; (iv) `grid[np.arange(8), np.arange(7,-1,-1)]`.

30. It responds strongly where the signal bends / changes slope (e.g. peaks, valleys, sharp transitions) and is near zero on flat regions. It is a discrete second-difference / "accentuate change" filter. Primarily **high-pass**: smooth regions are suppressed relative to rapid variation; it amplifies high-frequency content (and noise).

31. $\|\nabla I\| = \sqrt{I_x^2 + I_y^2}$, $\theta = \operatorname{atan2}(I_y, I_x)$. The magnitude forms a ridge around the true boundary; without NMS it stays thick.

32. Center correlation: $1 \cdot 1 + 2 \cdot 4 + 1 \cdot 2 + 0 + 0 + 0 + (-1) \cdot 2 + (-2) \cdot 1 + (-1) \cdot 6 = 1 + 8 + 2 - 2 - 2 - 6 = 1$. This is a (horizontal-edge / y-derivative) Sobel-style kernel: high-pass / derivative.

End of study guide.