1 Image Processing and Convolution

1.1 Preparation: Understanding Convolution

Think of convolution as a way to find patterns in an image by looking through a small window (kernel). As we slide this window across the image, we:

1. Look at a small neighborhood of pixels
2. Multiply each pixel by the corresponding value in our pattern-matching window (kernel)
3. Sum up these products to get a single number
4. Move the window and repeat

For example, if we want to detect vertical edges, we might use this kernel:

$$K = \begin{bmatrix} -1 & 1 & 0 \\ -1 & 1 & 0 \\ -1 & 1 & 0 \end{bmatrix}$$

Let's see how it works on a small image region (3x3 pixels):

$$\begin{bmatrix} 10 & 80 & 10 \\ 10 & 80 & 10 \\ 10 & 80 & 10 \end{bmatrix}$$

We multiply each pixel by the corresponding kernel value and sum: $(10 \times -1 + 80 \times 1 + 10 \times 0) \times 3 = 210$. The high positive value (210) indicates a strong vertical edge was detected. Here are some common kernel patterns:

Vertical edge detection:

$$\begin{bmatrix} -1 & 1 & 0 \\ -1 & 1 & 0 \\ -1 & 1 & 0 \end{bmatrix} \quad \text{Looks for } \xrightarrow{\text{dark to bright}}$$

Horizontal edge detection:

$$\begin{bmatrix} -1 & -1 & -1 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad \text{Looks for } \downarrow \text{ dark to bright}$$

Blur/Smoothing:

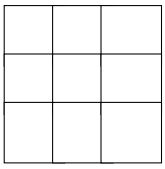
$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad \text{Averages all neighboring pixels}$$

1.2 Image Processing and Convolution

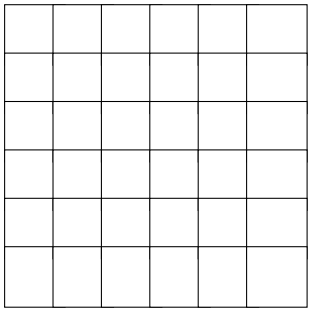
Consider a 6x6 grayscale image showing a diagonal line pattern:

80	10	10	10	10	10
10	80	10	10	10	10
10	10	80	10	10	10
10	10	10	80	10	10
10	10	10	10	80	10
10	10	10	10	10	80

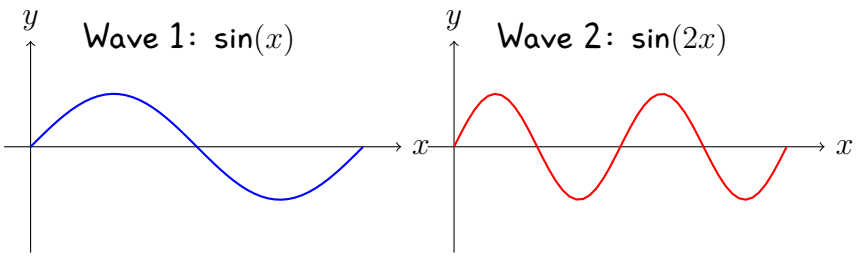
1. If we want to detect diagonal edges, which kernel of size 3x3 would be most appropriate? The kernel should have the values between -1 and 1.



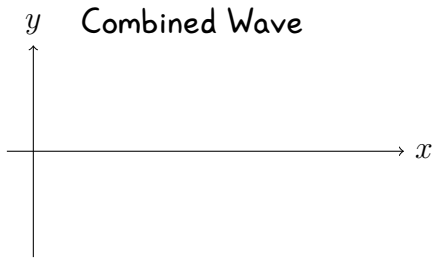
2. Apply your kernel to compute the convoluted image. No need to calculate the value of each pixel exactly but show your estimate by shading the pixels. For the boundary pixels, leave them blank since the kernel exceeds the boundary of the image.



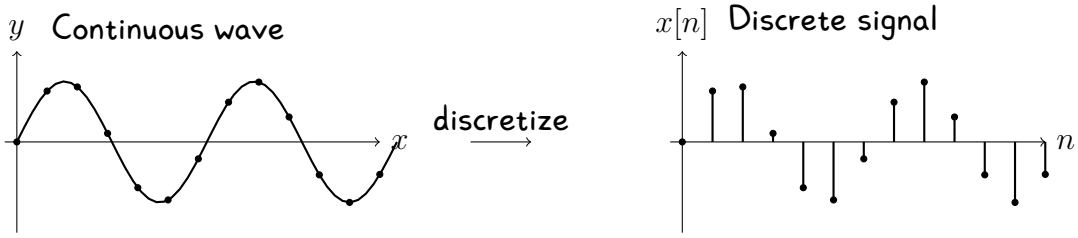
3. Now, let's learn how JPEG compression works. Consider this waves:



We combine these two waves by weighting the first wave by 1.5 and the second wave by 0.5. Combined Wave $= 1.5 \cdot \sin(x) + 0.5 \cdot \sin(2x)$. Draw the combined waves.



4. The Fourier transform is a reverse operation: it decomposes, not combines, waves into basic waves. The waves are continuous functions. But we can discretize them for computation as follows:



This results in a vector of values $[10, 80, 10, 80, 10, 80, 10, 80]$. Now, let's create a discretized mixed wave Z from X and Y as follows

$$Z = X + Y = [10, 90, 30, 90, 10, 70, -10, 70, 10] \tag{1}$$

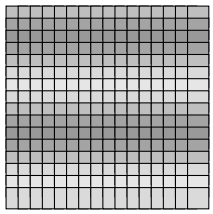
where

$$X = [10, 80, 10, 80, 10, 80, 10, 80, 10], \quad Y = [0, 10, 20, 10, 0, -10, -20, -10, 0] \tag{2}$$

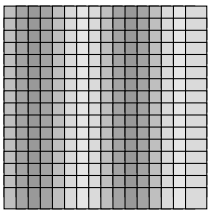
(a) If we apply a kernel $K = [-1, 1, -1]$ to this signal, what will be the resulting signal? What kind of frequencies will this kernel emphasize?

(c) If we apply a kernel $K = [1, 1, 1]$ to this signal, what will be the resulting signal? What kind of frequencies will this kernel emphasize?

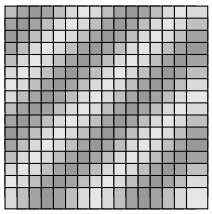
5. Just as 1D signals can be decomposed into sine waves, 2D images can be decomposed into 2D waves as follows. The Fourier transform can be applied to 2D images to decompose them into a sum of 2D waves.



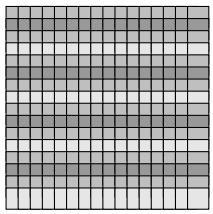
Horizontal (Low freq.)



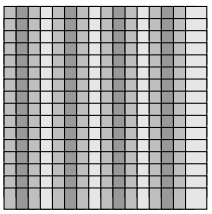
Vertical (Low freq.)



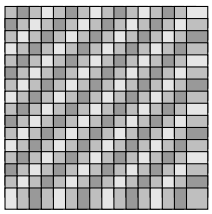
Diagonal (Low freq.)



Horizontal (High freq.)



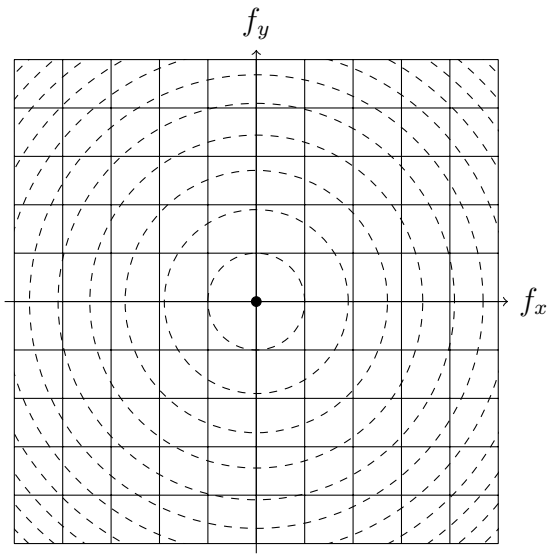
Vertical (High freq.)



Diagonal (High freq.)

Now, consider this checkerboard pattern on the left. Mark where you expect the highest magnitudes in the Fourier transform grid. The dashed circles represent the basis 2D waves in the Fourier domain.

80	10	80	10	80	10
10	80	10	80	10	80
80	10	80	10	80	10
10	80	10	80	10	80
80	10	80	10	80	10
10	80	10	80	10	80

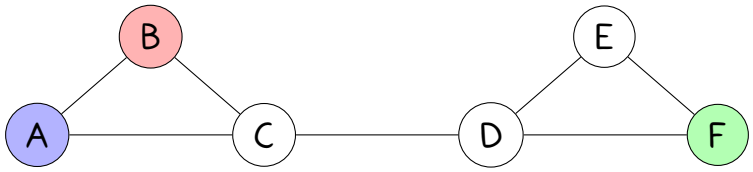


Fourier Transform Grid

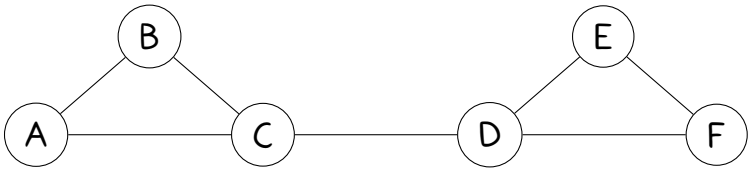
6. The image can be mapped to the Fourier transform grid (called frequency domain). We can also map it back to the original image domain (called spatial domain). Thus, we can manipulate the image in the frequency domain to remove some waves from the original image. If we want to keep only the low-frequency components of the checkerboard pattern, what regions of the Fourier transform grid should we set to zero?

2 Color Spreading in Social Networks

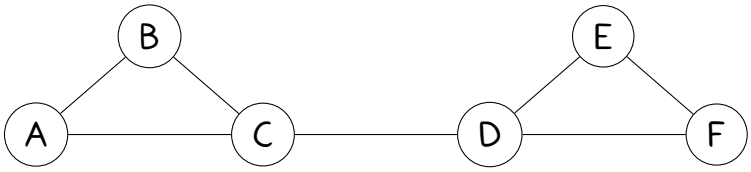
In a network of students forming study groups, each student starts with a preferred subject, shown by colors - Math (Blue), Physics (Green), or Arts (Red). Through their study group interactions, students influence each other's interests. The network diagram below shows students as circles connected by lines representing their study partnerships.



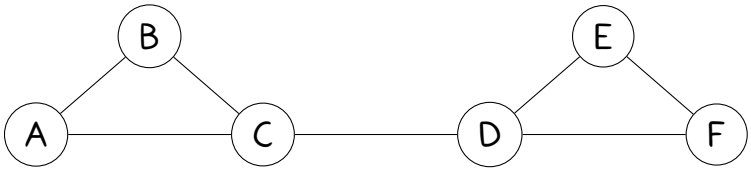
7. For each student, we'll follow this rule: "Adopt the most common subject among your study partners." If tied, choose Math (Blue) > Arts (Red) > Physics (Green). Now, let's start by looking at student D, who is currently undecided. What subject would student D choose?
8. Now look at student E. Using the same rule, what subject would they choose?
9. Starting with nodes D, E, F, C, B, and A in that order, what color pattern emerges after many iterations?



10. If we repeat this process in order of A->B->C->D->E->F infinitely, what would be the final pattern of colors?



11. Let's introduce a self-reinforcing rule: When counting subjects among neighbors, a student's own chosen subject counts as an extra vote with weight three times as much as a vote from a neighbor. For example, if a student has chosen red and has two neighbors with green, red gets counted three times while green gets counted twice. Starting with nodes A, B, C, D, E, F in that order, what color pattern emerges after many iterations?



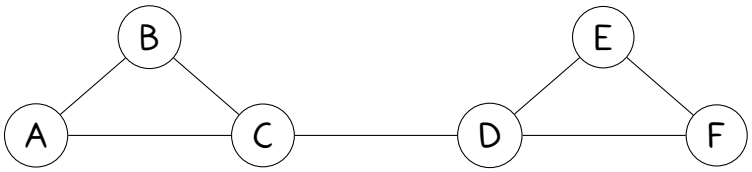
12. Let's make it more operational. Each color can be represented as a one-hot vector as follows:

    - Blue (Math) = [1,0,0]
    - Green (Science) = [0,1,0]
    - Red (Arts) = [0,0,1]
    - White (Undecided) = [0,0,0]

    Then, how can we represent the rule "Adopt the most common subject among your study partners" as a computational operation? Ignore self-reinforcing rule. Hint: it is a max-pooling operation.

13. Design a computational operation that follows the self-reinforcing rule without altering the operation in the previous question?

14. What if we use "mean" instead of "max" and repeat the operation infinitely? Draw the final pattern of colors. There is no tie in this case.

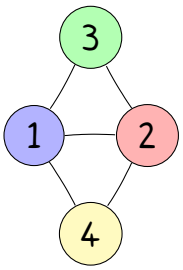


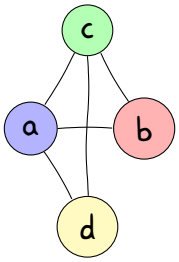
15. What if we use "sum" operation and repeat the operation infinitely?

3 The Weisfeiler-Lehman Test

The color spreading process we just studied is closely related to a fundamental concept in graph theory called the Weisfeiler-Lehman (WL) test. This test helps us determine if two graphs might be structurally different by looking at how colors spread through them.

16. Consider these two graphs. Are they the same (isomorphic)?

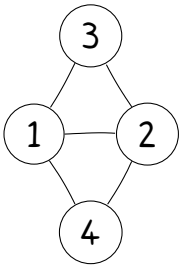


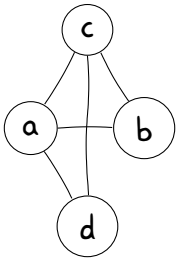


17. Now let's look at how the WL test works:

    - For each node,
      - Collect the colors of its neighbors (including itself)
      - Count the frequency of each color, and create a list of color frequencies in order of Red, Green, Blue. For example, if there are two reds, one green, and one blue, the sorted frequency list is [2, 1, 1].
    - Once all nodes have been processed, reassign the same color to the nodes with the same sorted frequency list.
    - Count the number of nodes for each color, and create a sorted list of color frequencies for a graph.

    For the graphs above, perform one iteration of the WL test, and provide the colors of the nodes after the iteration, and the sorted color frequency list for each graph.

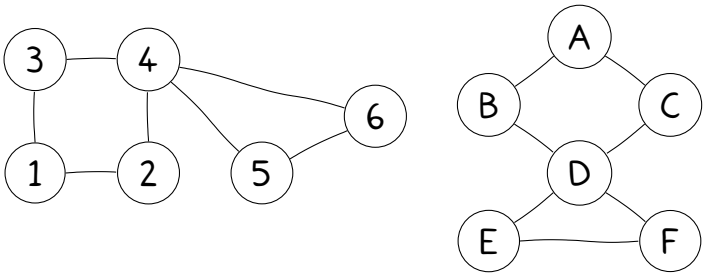




Sorted color frequency list (the first graph): _____

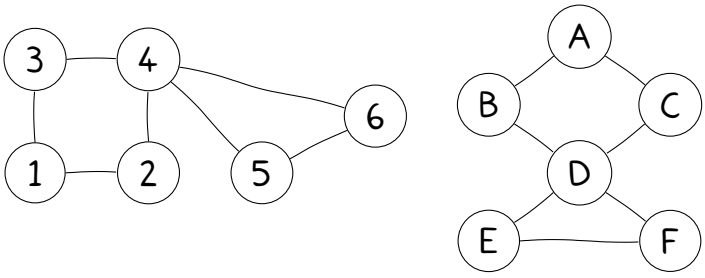
Sorted color frequency list (the second graph): _____

18. Consider these two graphs. At first glance, they look similar. Apply one iteration of the WL test to see if they're actually the same:

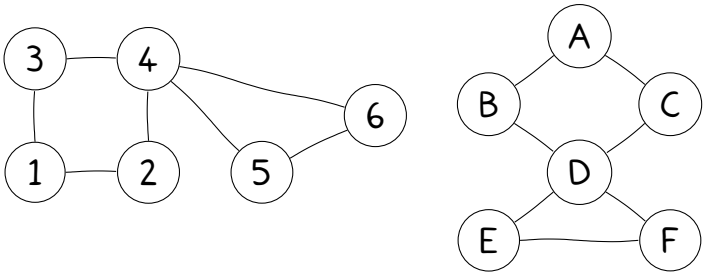


19. Apply one iteration of the WL test to the two graphs above. And then apply the second iteration. Provide the colors of the nodes after the iteration, and the sorted color frequency list for each graph.

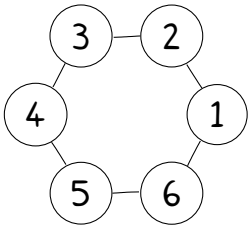
    First iteration:



    Second iteration:



20. The WL test is heuristic and not always correct. Consider a graph of six nodes forming a ring. Create another non-isomorphic graph with the same number of nodes, and edges that the WL test results in the same color frequency list.



21. What is the difference between the WL test and the color spreading process we studied in the previous section?