

Homework #1

Deep Learning for Computer Vision

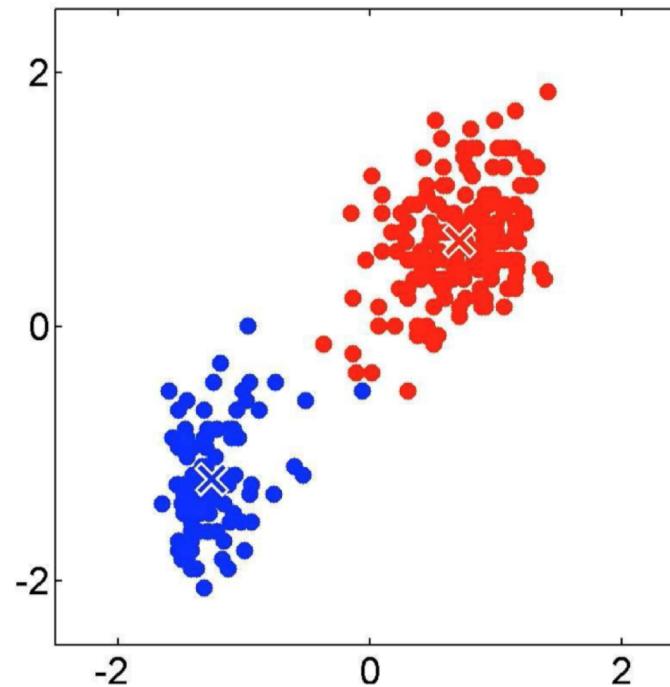
NTU, Fall 2020

109/9/29

109/10/13 (Tue.) 02:00 AM due

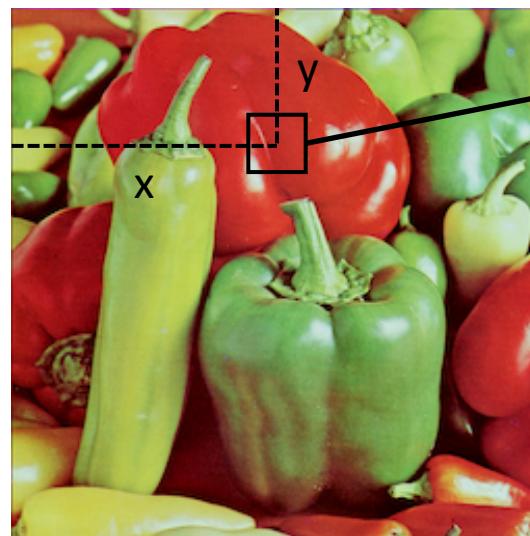
Problem 1: K-Means Clustering

- Perform K-means clustering as taught in the lectures
- Example: $K = 2$



Problem 1: K-Means Clustering

- Perform K-means clustering on each pixel's color
- Perform K-means clustering on each pixel's color and location



(R, G, B)

Problem 1: K-Means Clustering

- Example: K-means clustering on each pixel's color with different number of clusters
 - For each clustering group, replace all pixels' RGB values with that of the cluster center

K = 2



K = 4



K = 8



K = 16

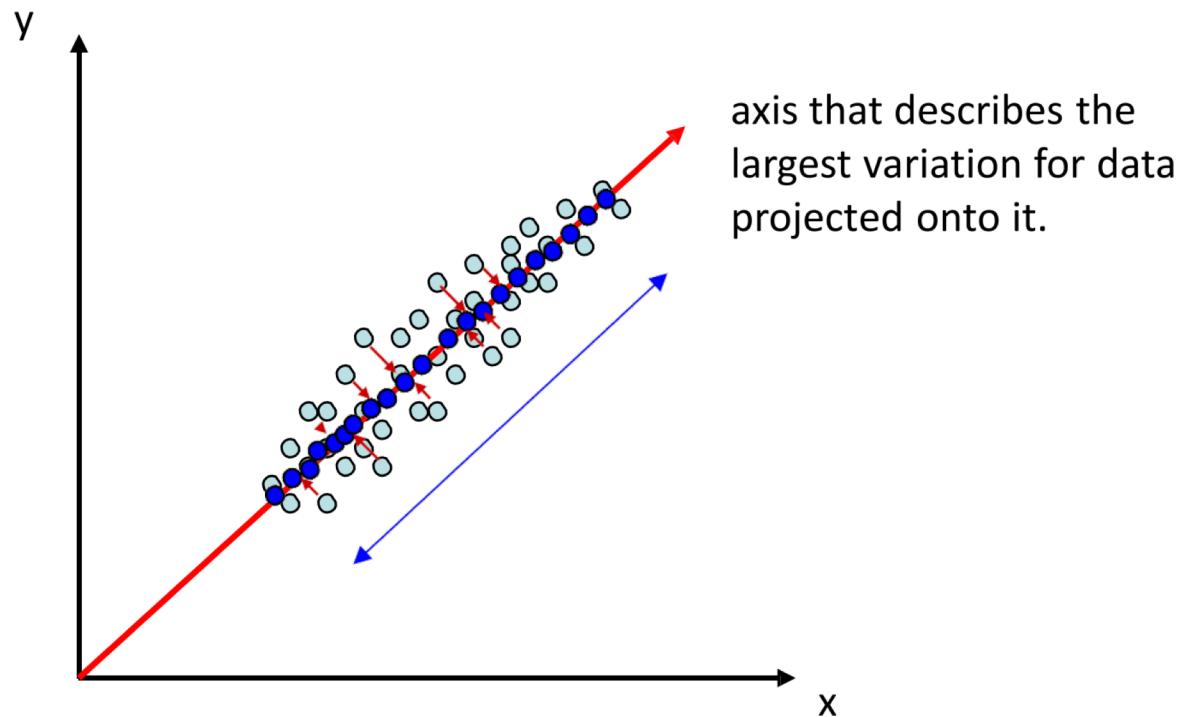


K = 32



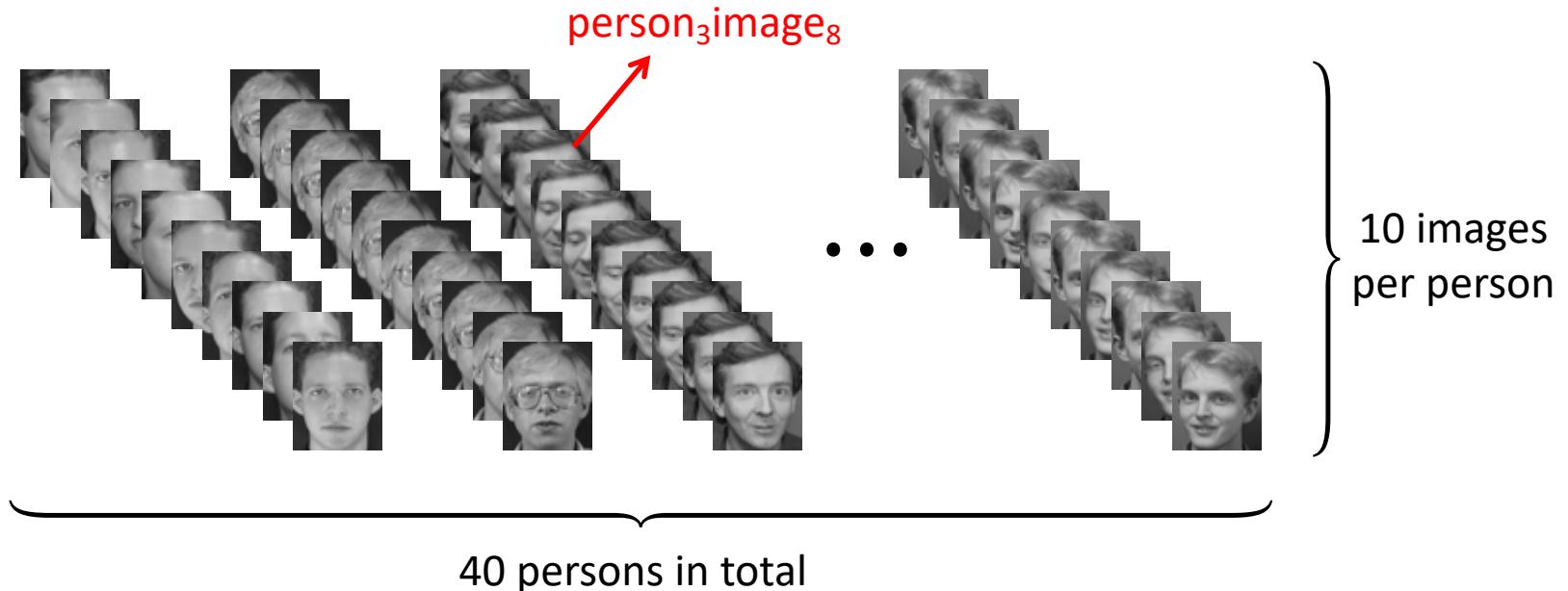
Problem 2: Principle Component Analysis

- Perform PCA as taught in the lectures



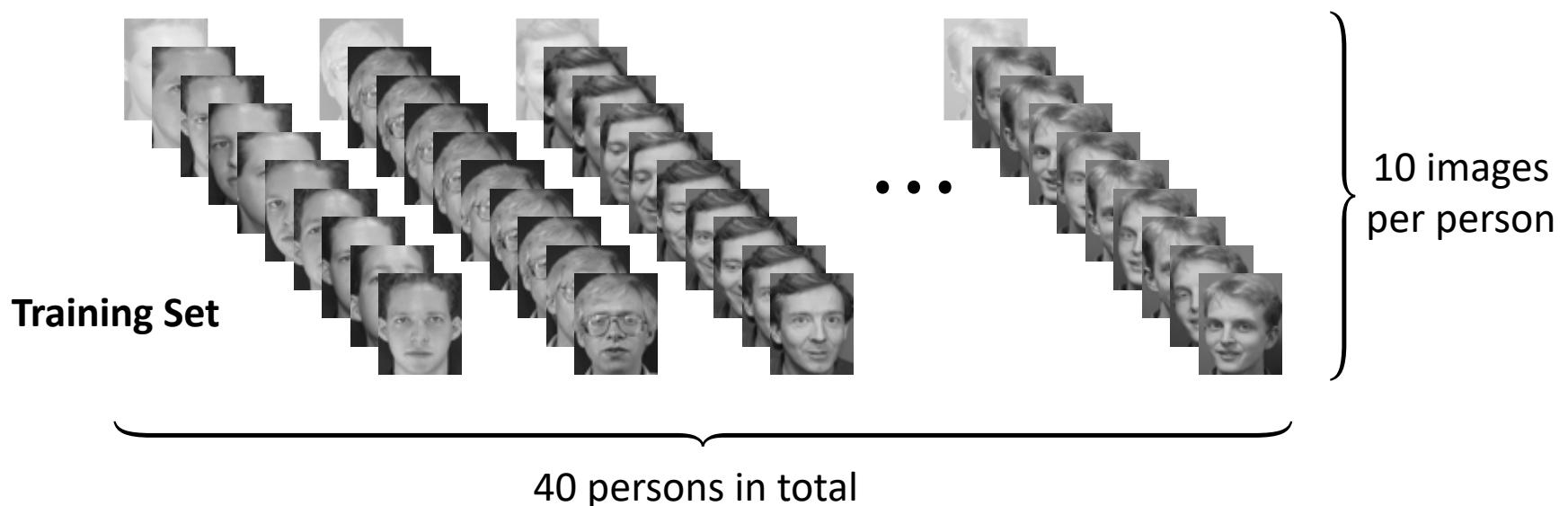
Problem 2: Principle Component Analysis

- Perform PCA as taught in the lectures
- Dataset



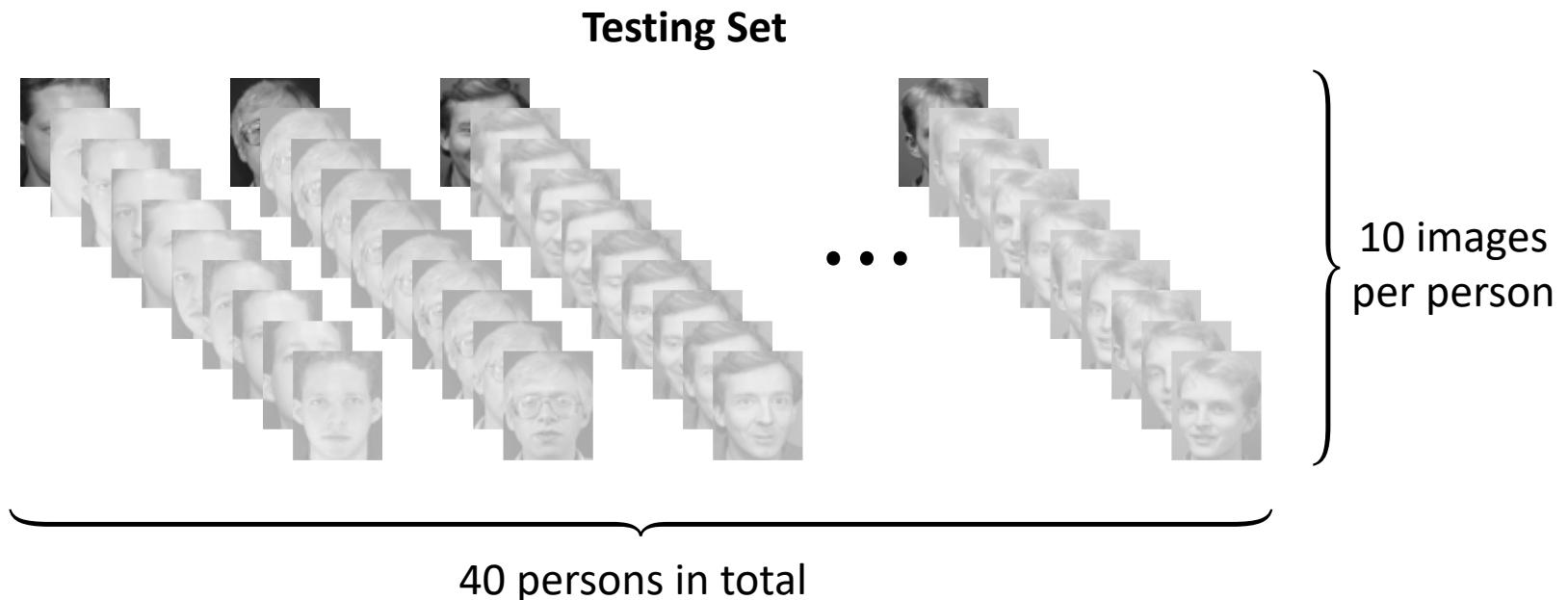
Problem 2: Principle Component Analysis

- Perform PCA as taught in the lectures
- Dataset



Problem 2: Principle Component Analysis

- Perform PCA as taught in the lectures
- Dataset

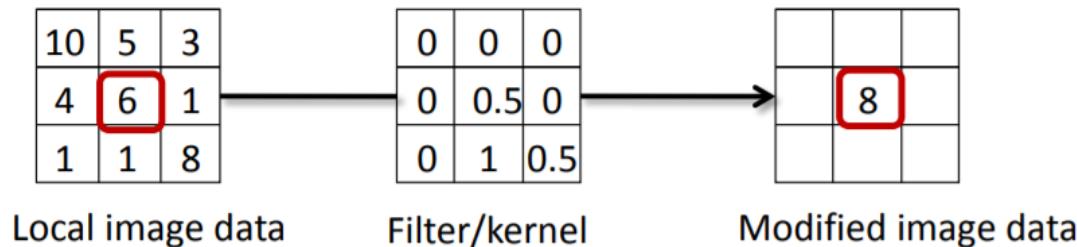


Problem 2: Principle Component Analysis

- Plot the mean face and eigenfaces
- Project face images onto the eigenspace
- Plot the reconstructed image
- Compute mean squared error
- Apply k -nearest neighbors for classification

Problem 3: Image Filtering

- What's Image Filtering?
 - Image Filtering
 - For each pixel of interest, compute the function of local neighborhood and output the new value.
 - Generally (while not always true), same function applied to each position
 - Linear filtering
 - function is a weighted sum/difference of pixel values (all the filtering methods we use in this homework are linear filtering)



Problem 3: Image Filtering

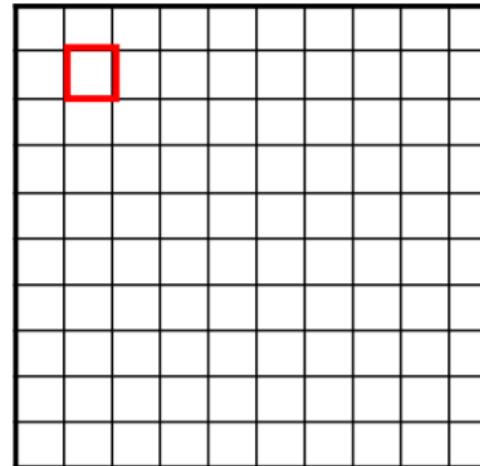
- Example: Image Filtering with a Box Filter

$$g[\cdot, \cdot] \frac{1}{9} \begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix}$$

$$f[.,.]$$

0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	0	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

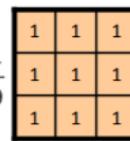
$$h[.,.]$$



$$h[m,n] = \sum_{k,l} g[k,l] f[m+k, n+l]$$

Problem 3: Image Filtering

- Example: Image Filtering with a Box Filter

$$g[\cdot, \cdot] \frac{1}{9}$$


A 3x3 matrix where every element is 1, representing a uniform box filter with a size of 3x3.

$$f[.,.]$$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0

$$h[.,.]$$

0	10								

$$h[m,n] = \sum_{k,l} g[k,l] f[m+k, n+l]$$

Slide credit: S. Seitz

Problem 3: Image Filtering

- Example: Image Filtering with a Box Filter

$$g[\cdot, \cdot] \frac{1}{9} \begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix}$$

$$f[.,.]$$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$$h[.,.]$$

		0	10	20					

$$h[m, n] = \sum_{k,l} g[k, l] f[m + k, n + l]$$

Problem 3: Image Filtering

- Example: Image Filtering with a Box Filter

$$g[\cdot, \cdot] \begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix}$$

$$f[.,.]$$

0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	90	0	90	90	90	0	0	0
0	0	0	90	90	90	90	90	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

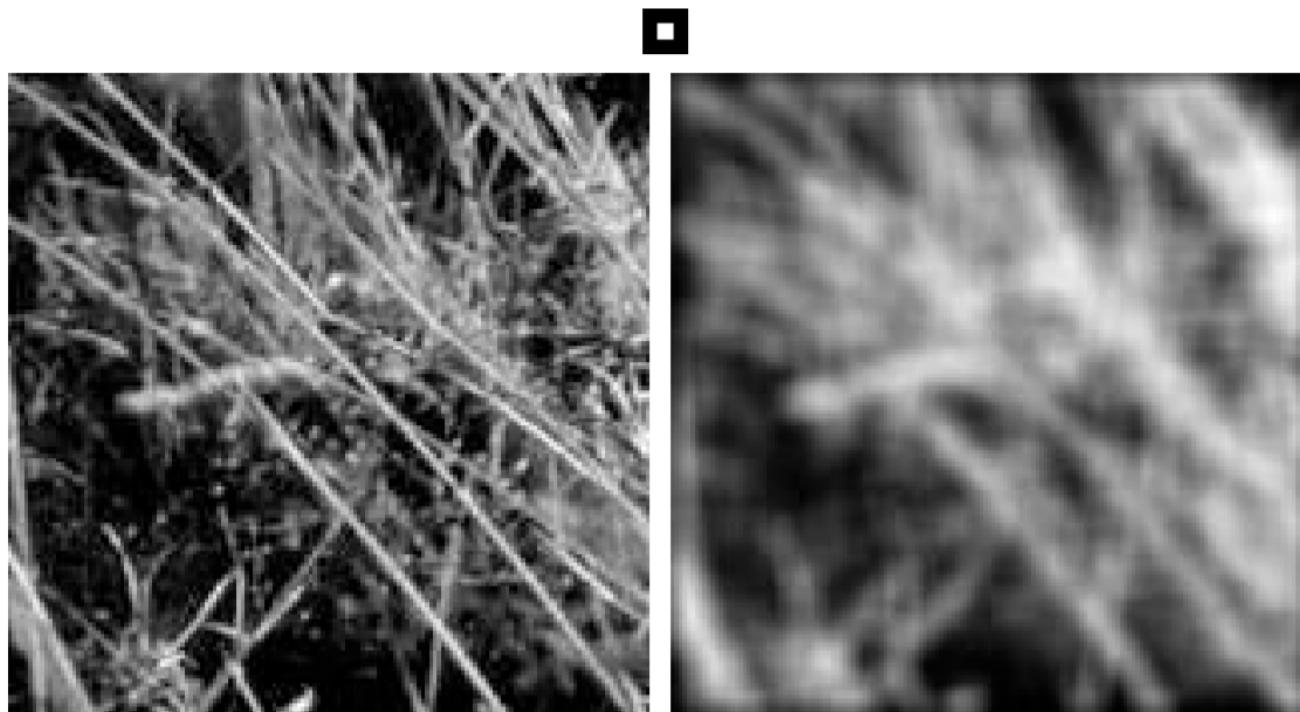
$$h[.,.]$$

	0	10	20	30	30	30	20	10		
	0	20	40	60	60	60	40	20		
	0	30	60	90	90	90	60	30		
	0	30	50	80	80	90	60	30		
	0	30	50	80	80	90	60	30		
	0	20	30	50	50	60	40	20		
10	20	30	30	30	30	20	10			
10	10	10	0	0	0	0	0	0		

$$h[m, n] = \sum_{k,l} g[k, l] f[m + k, n + l]$$

Problem 3: Image Filtering

- Smoothing with a Box Filter

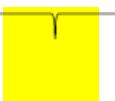


Problem 3: Image Filtering

- Rewrite the formulation of box filtering.

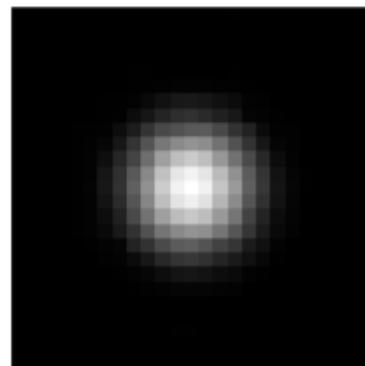
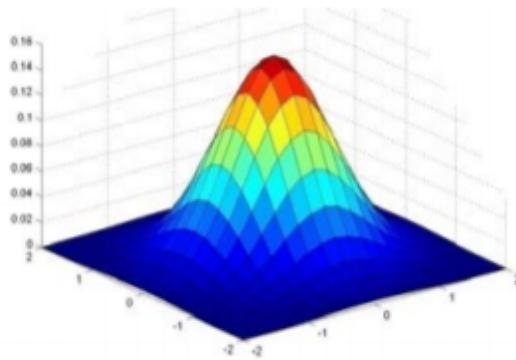
$$G[i, j] = \frac{1}{(2k+1)^2} \underbrace{\sum_{u=-k}^k}_{\text{Attribute uniform weight}} \underbrace{\sum_{v=-k}^k}_{\text{to each pixel}} F[i+u, j+v]$$

- Now generalize to allow **different weights** depending on neighboring pixel's relative position:

$$G[i, j] = \sum_{u=-k}^k \sum_{v=-k}^k H[u, v] F[i+u, j+v]$$


Problem 3: Image Filtering

- Gaussian Filter



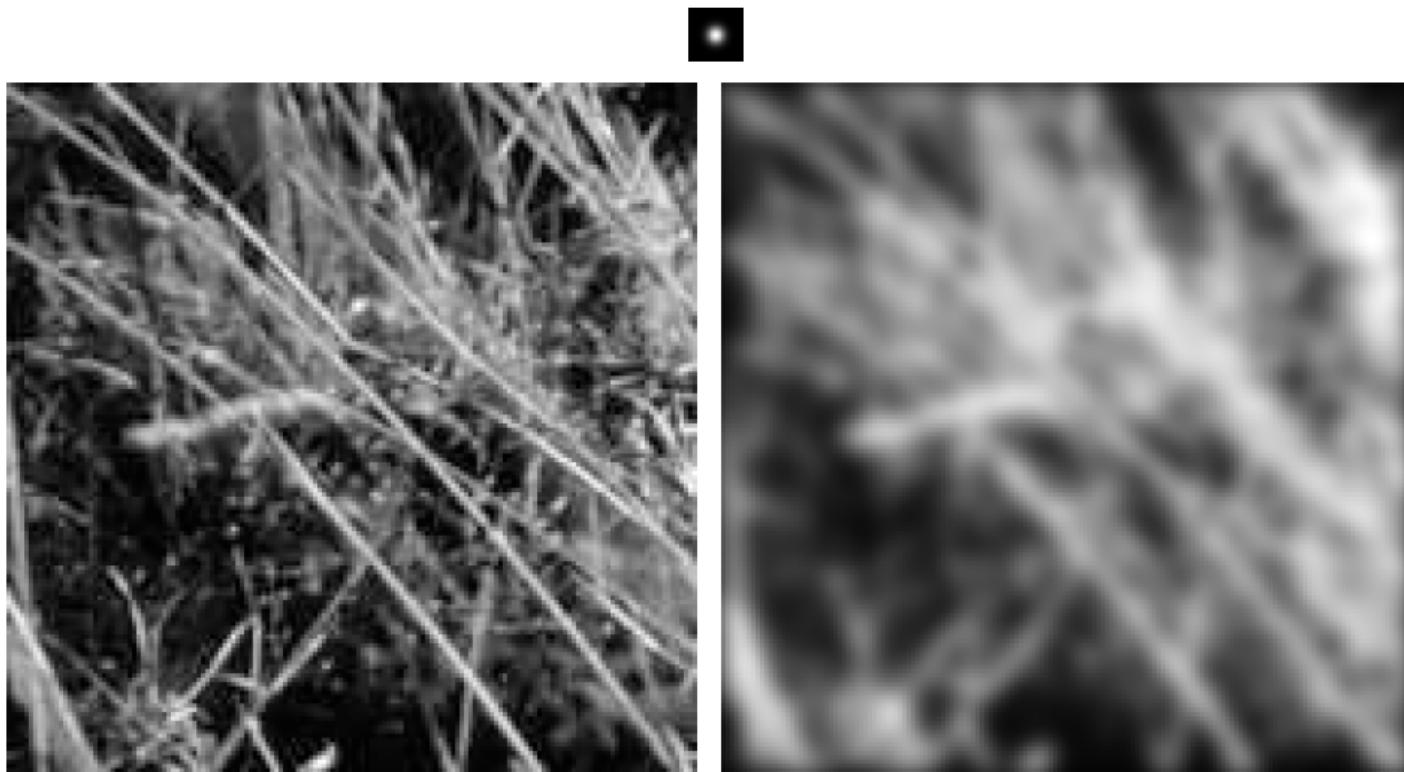
0.003	0.013	0.022	0.013	0.003
0.013	0.059	0.097	0.059	0.013
0.022	0.097	0.159	0.097	0.022
0.013	0.059	0.097	0.059	0.013
0.003	0.013	0.022	0.013	0.003

5 x 5, $\sigma = 1$

$$G_{\sigma} = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

Problem 3: Image Filtering

- Smoothing with a Gaussian Filter



Problem 3: Image Filtering

- Convolution

$$G[i, j] = \sum_{u=-k}^k \sum_{v=-k}^k H[u, v] F[i - u, j - v]$$

$$G = H \star F$$



*Notation for
convolution
operator*

Problem 3: Image Filtering

- Implement a simple 2D Gaussian filter
- Apply edge detection using derivatives of pixel values
- Plot the detected vertical and horizontal edges
- Plot the gradient magnitude image

Remarks

- You are allowed to use **any** programming language you desire, including all related packages/functions.
- In your report, provide detailed explanations or discussions about your answer.
- If unsure about your answer, write down how you obtained it as detailed as possible.