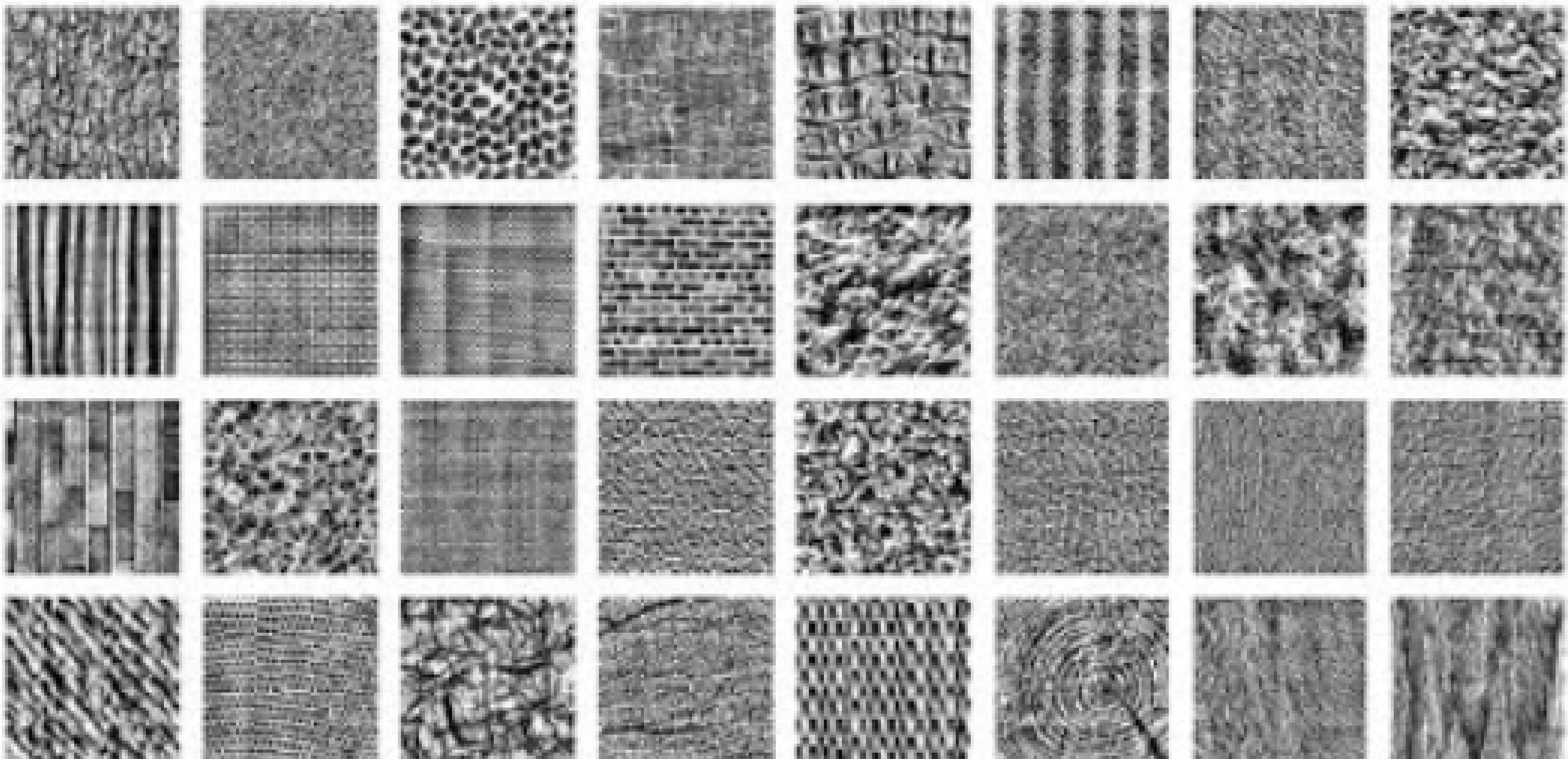
Texture Analysis

C.-C. Jay Kuo
University of Southern California

Introduction

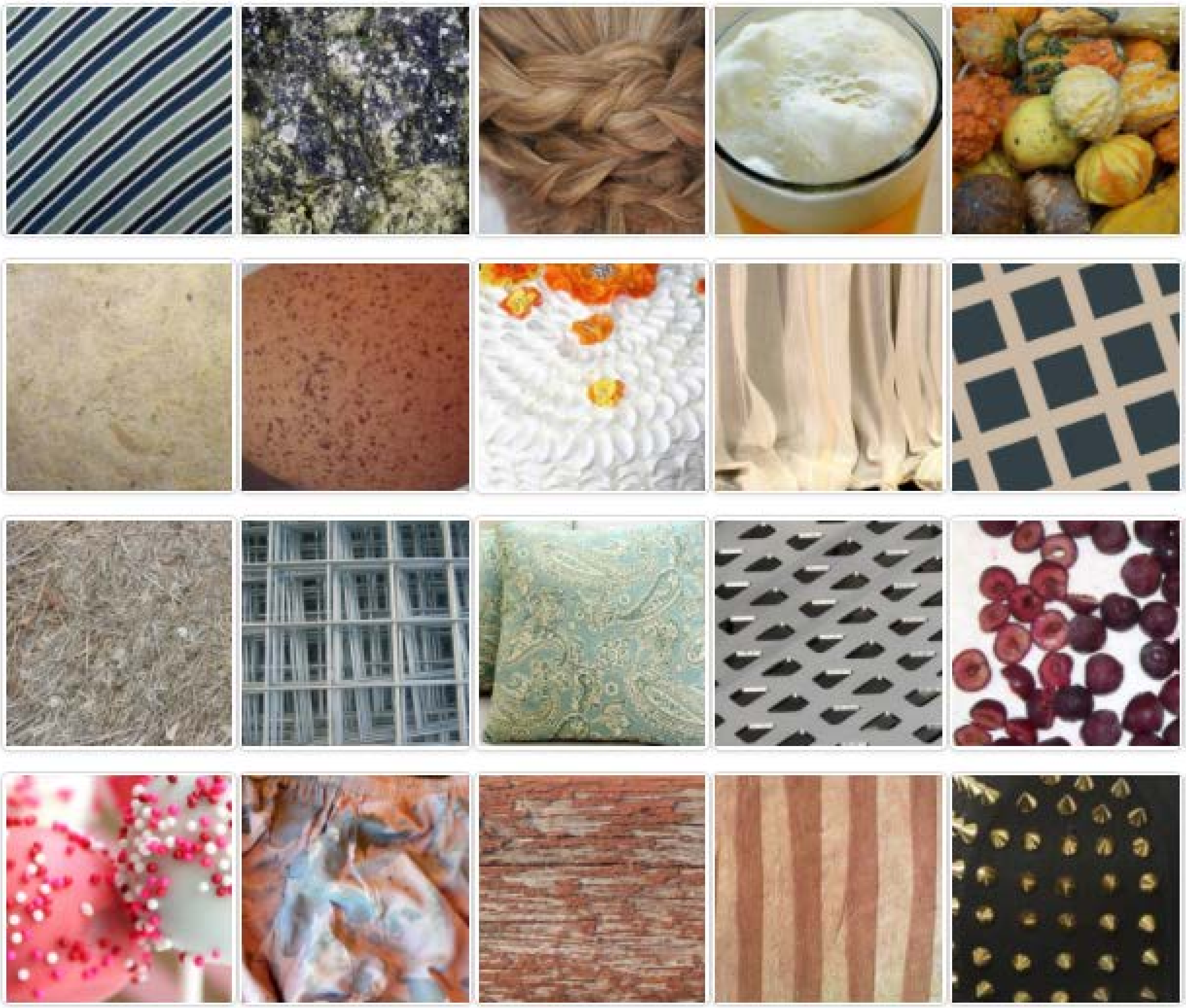
USC Brodatz Texture dataset



MIT Vision Texture Dataset

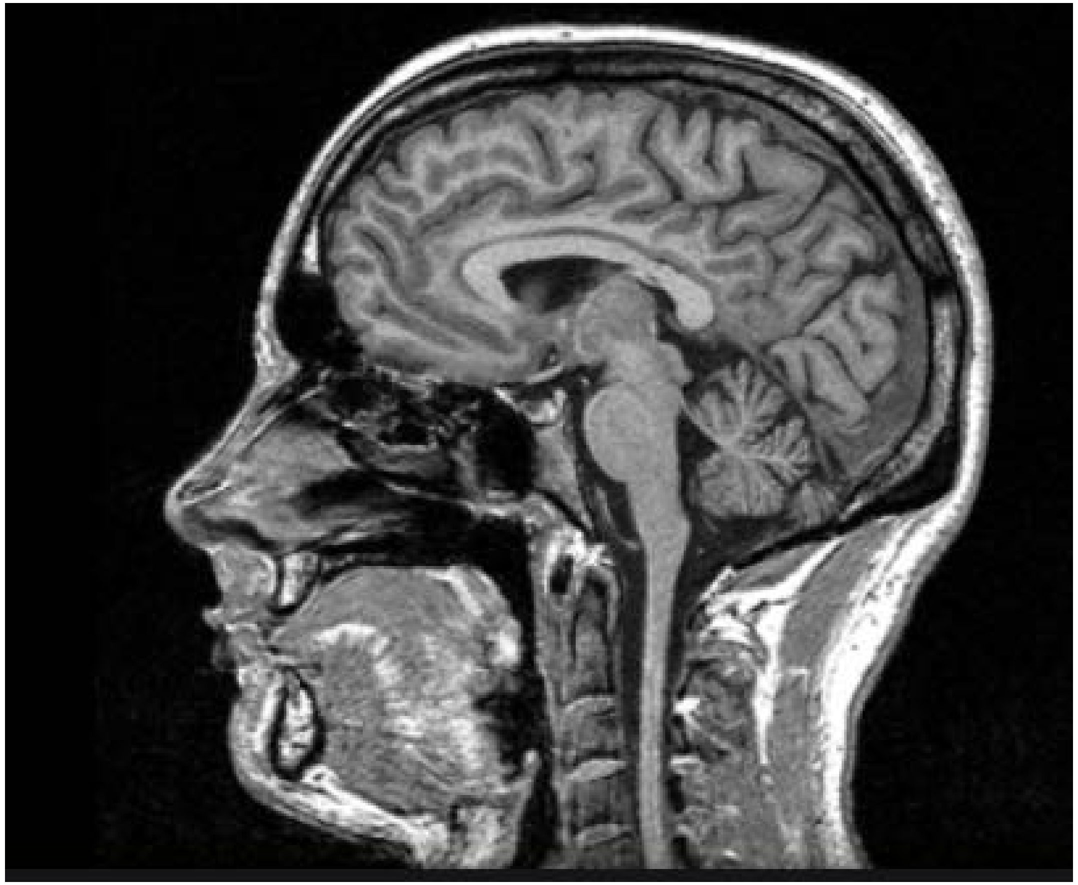
(a) (b) (c) (d) (e)

Describable Textures Dataset (DTD)



Remote Sensing Images

MRI Scanned Images



Real World Images (1)

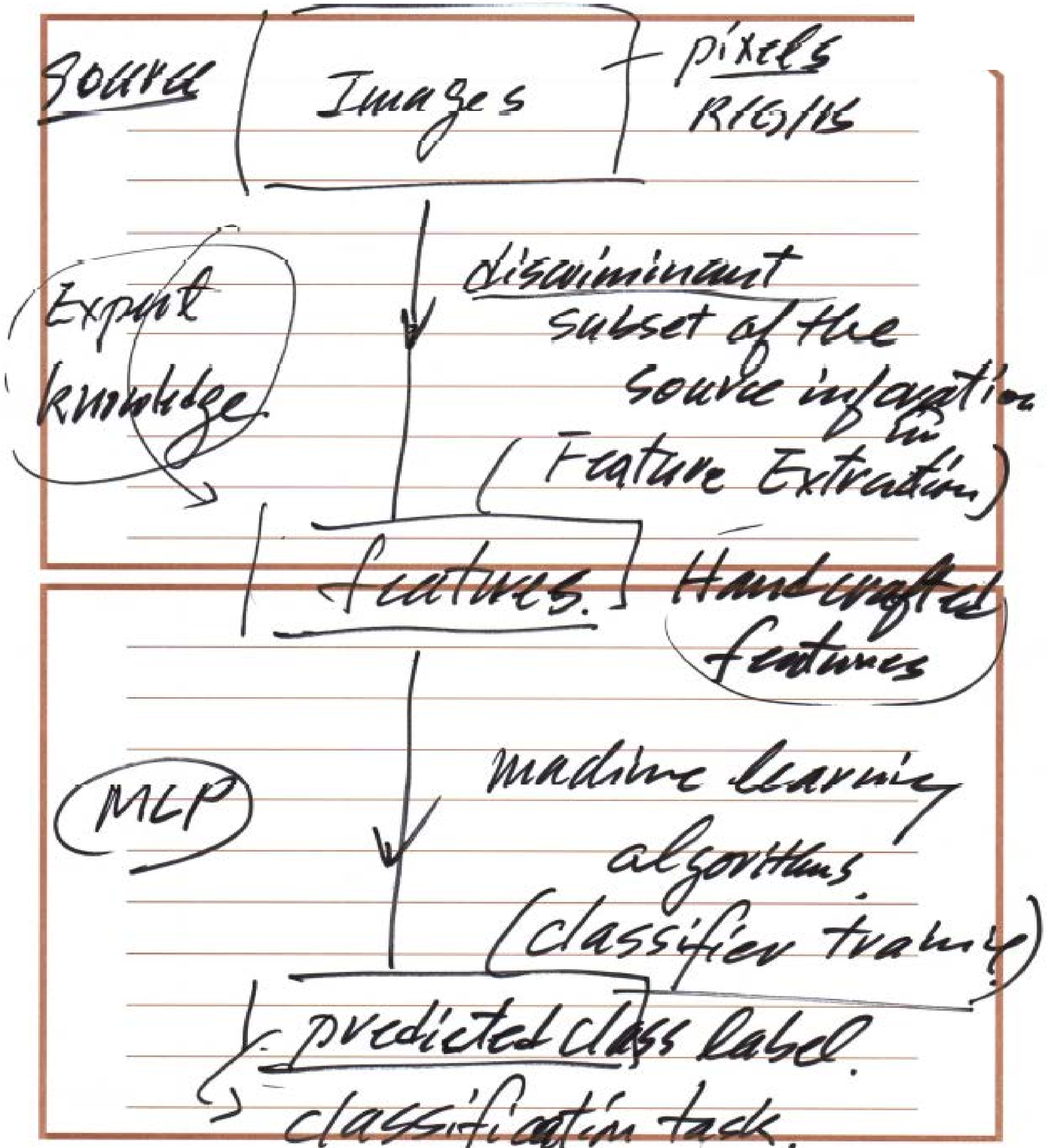
Real World Images (2)

Texture Analysis

Texture Definition and Applications

- Texture
  - No formal mathematical definition
  - Defined by examples
    - Regions or surfaces exhibit certain patterns, e.g., water, grass, wood, cloud, etc.
    - Most natural images consist of smooth regions, textured regions and edge regions
- Applications of Texture Analysis
  - Remote sensing image analysis (segmentation, classification, etc.)
  - Medical image analysis

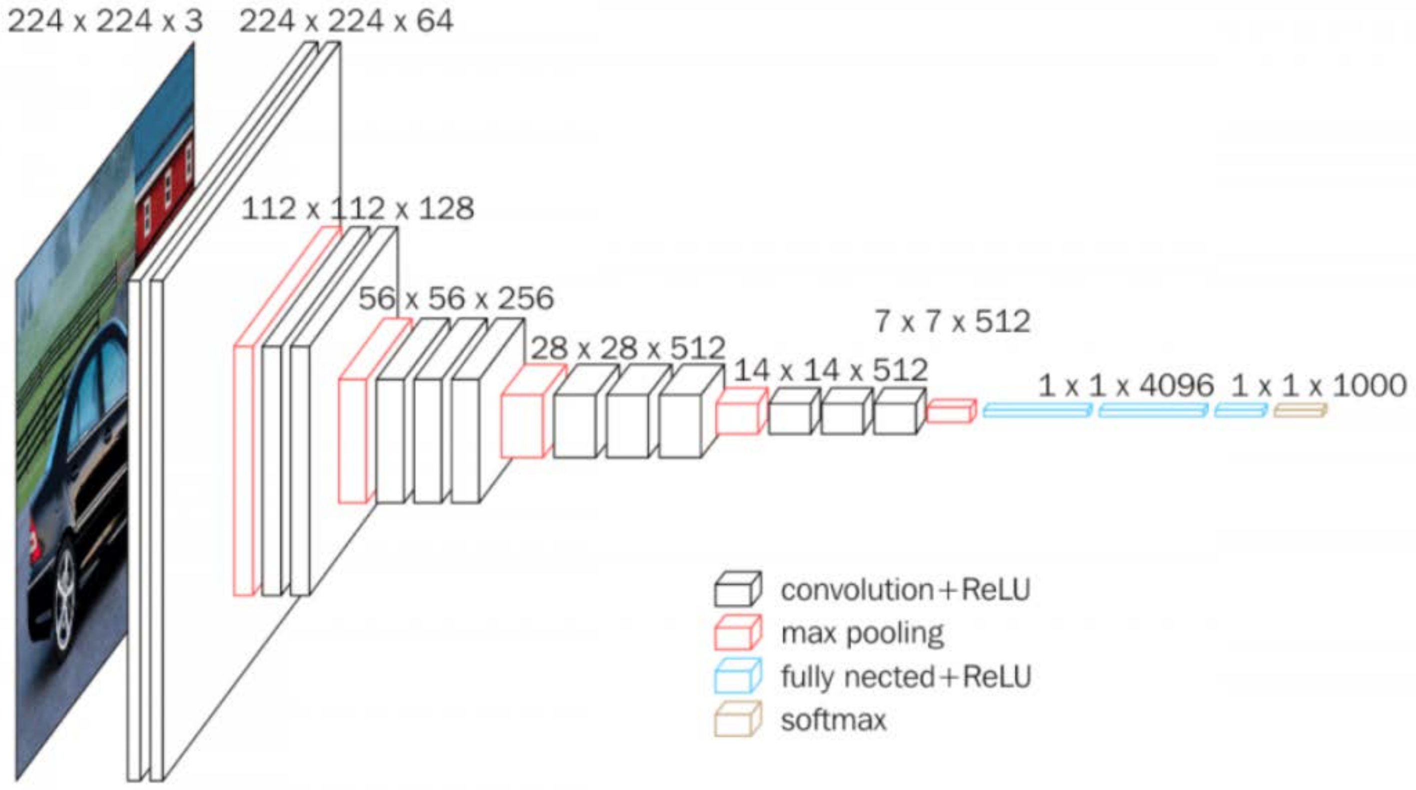
Image Analysis (a.k.a. Low-Level Computer Vision)



“End-to-End” or “Modularized” Design

- Recent Trend
  - The end-to-end optimized network design becomes popular since 2012
  - Deep learning

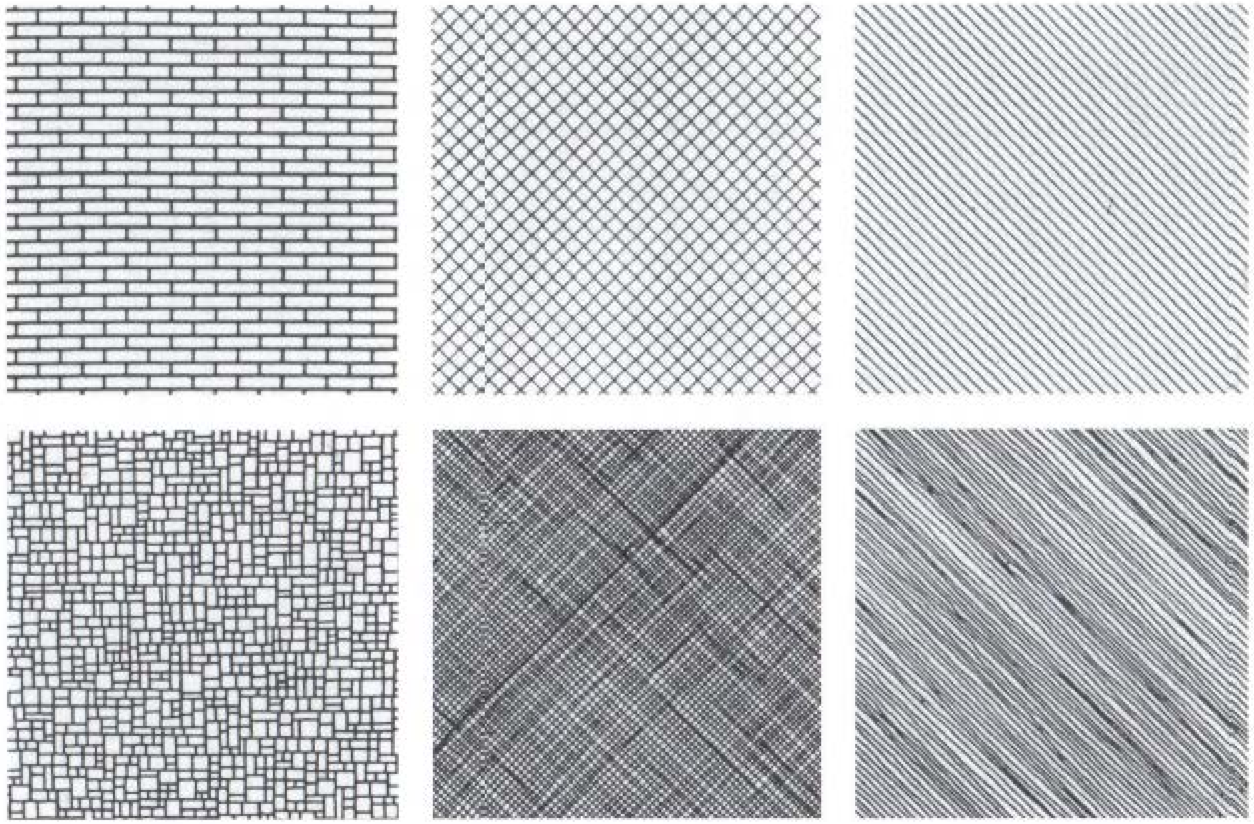
VGG-16 Network



Challenges of Texture Analysis

- Quasi-periodic, 2D random field, dominated by high frequency components

Pratt's Book
Page 546



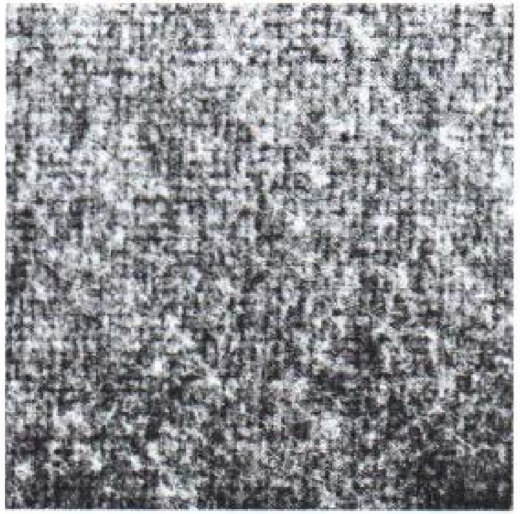
Artificial Textures

Brodatz Texture Examples

Pratt's Book
Page 547

(*a*) Sand

(*b*) Grass



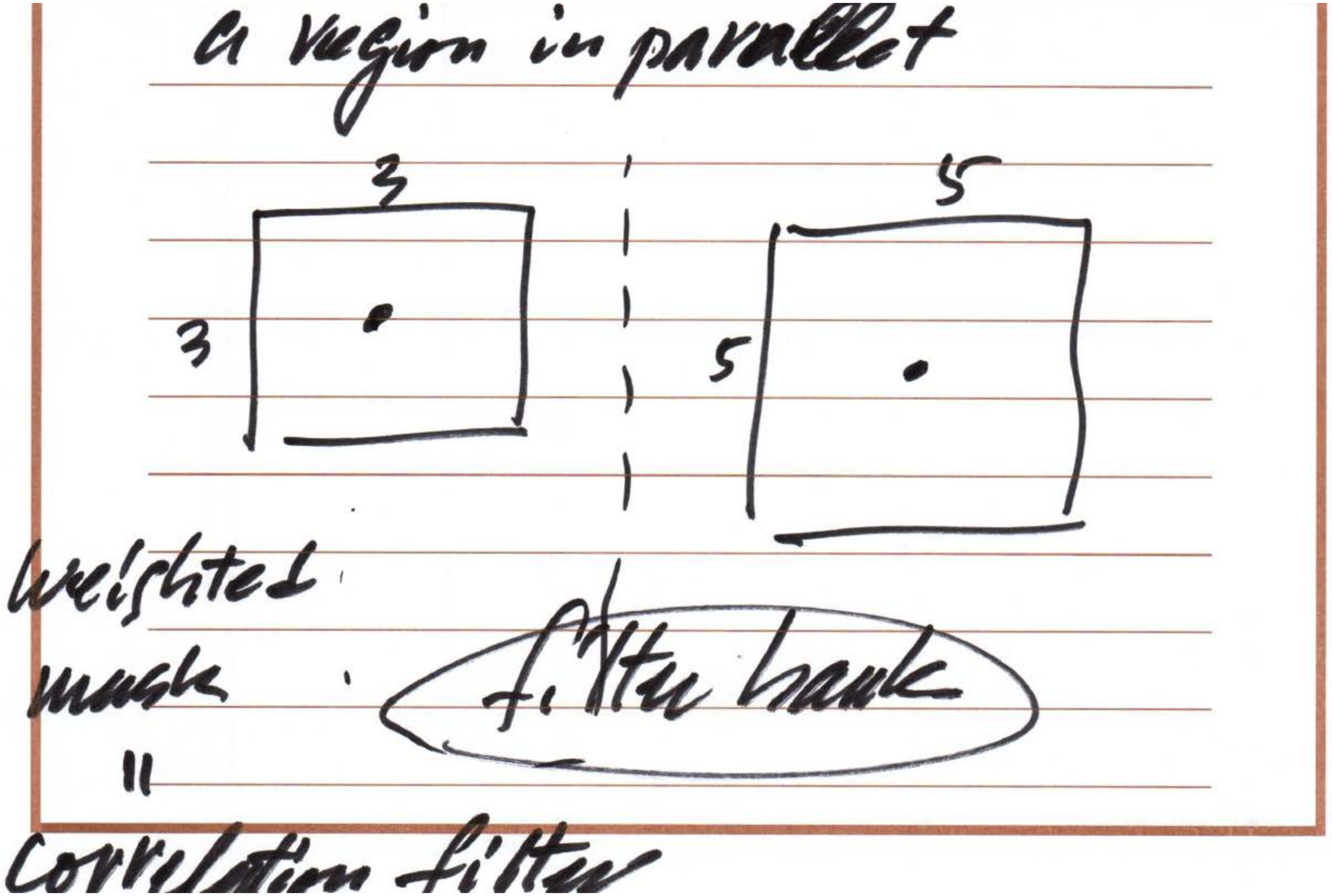
(*c*) Wool

(*d*) Raffia

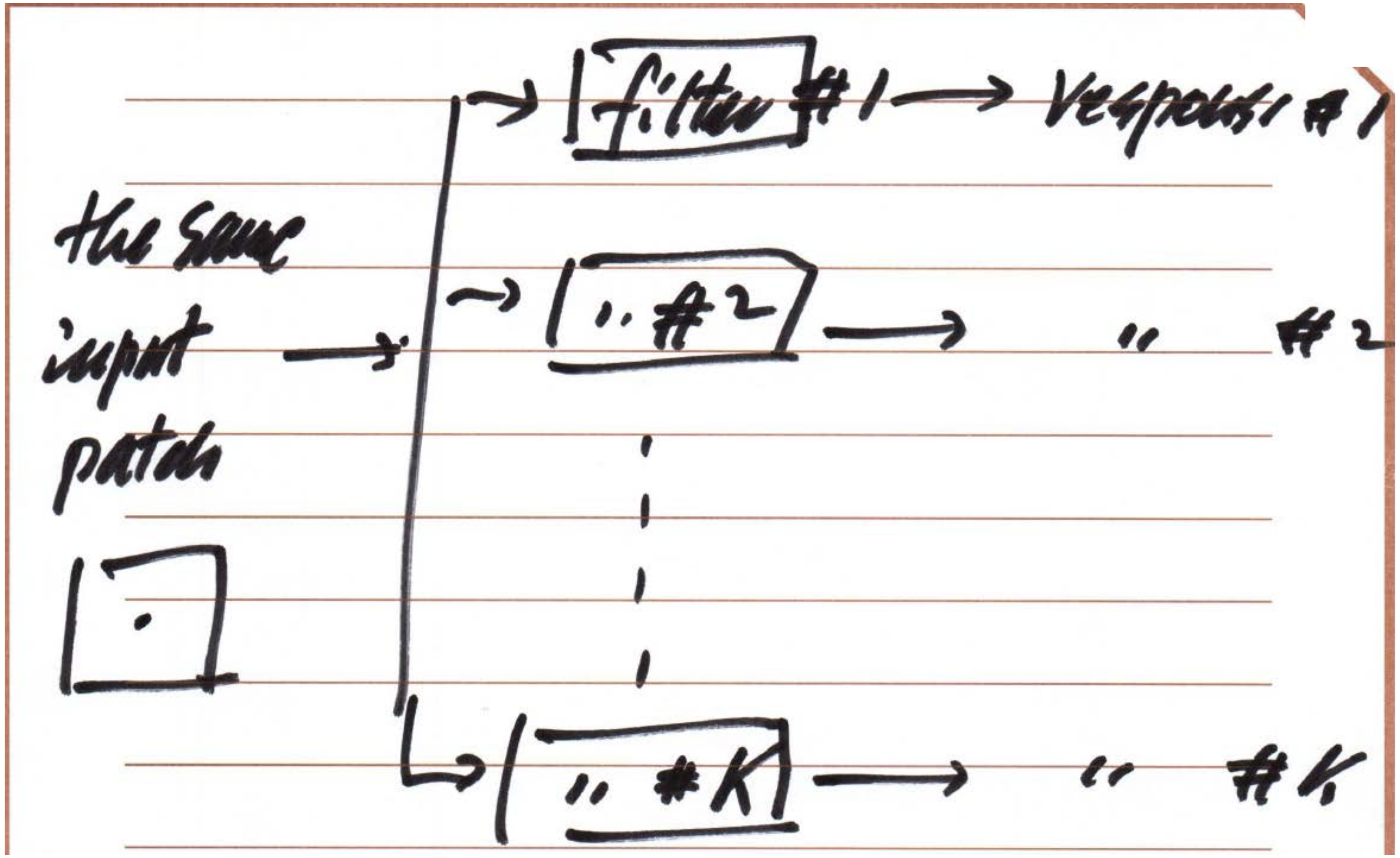
Texture Feature Extraction

Long history of texture feature
extraction. — ad hoc.
(in the textbook)
1980 Kenneth Laws.
a set of filters applied to
(3×3 or 5×5)

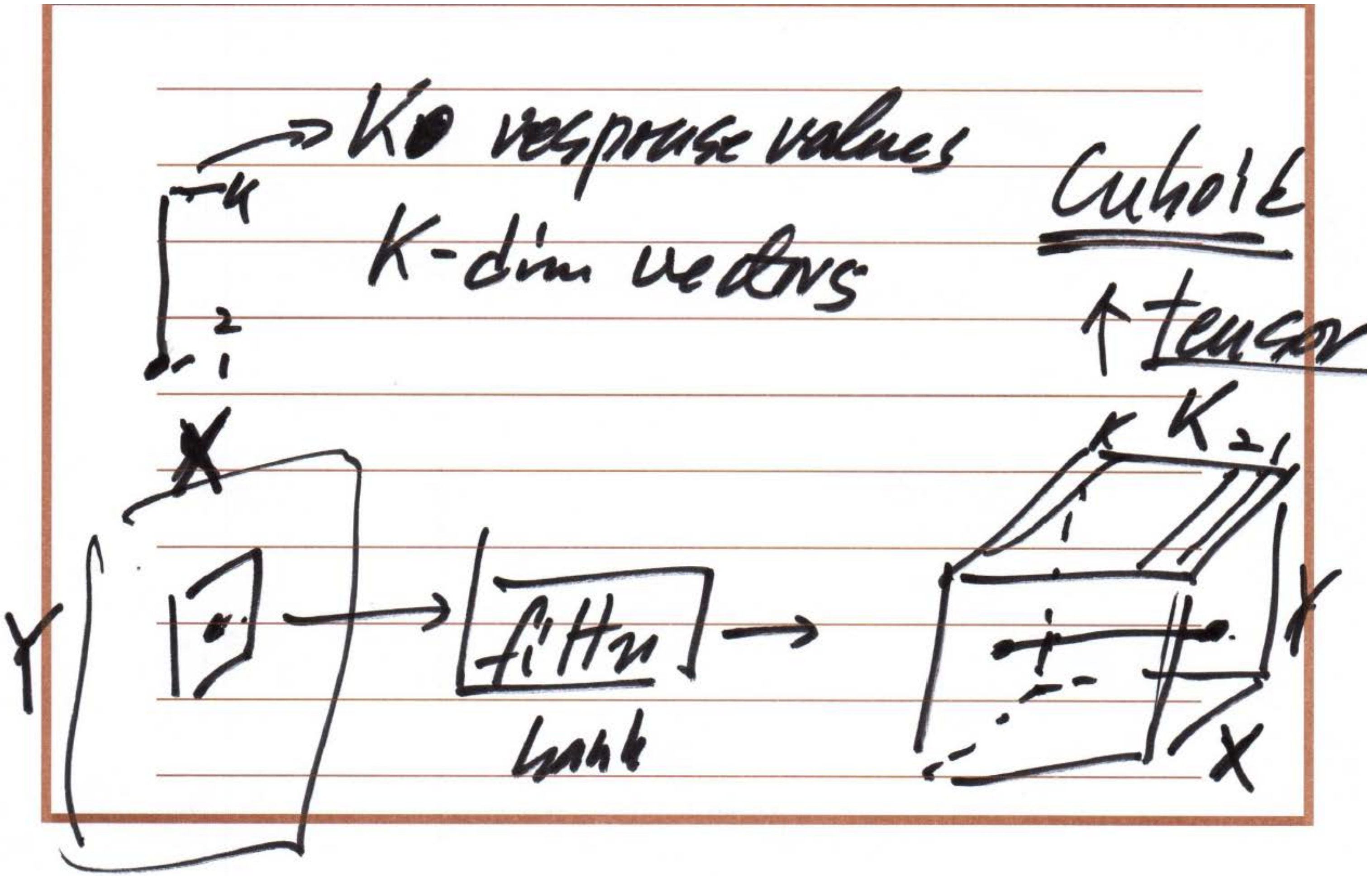
3x3 and 5x5 Laws' Filters



Filter Banks



Filter Response Vector



Laws' 3x3 Filters along x-Axis

Horizontal Filters

Laws' filters. 1980.

3x3

Three 1D filters

L3: $\frac{1}{6}(1, 2, 1)$

local averages

E3: $\frac{1}{2}(-1, 0, 1)$

edge

S3: $\frac{1}{2}(-1, 2, -1)$

spot

1D tensor

Laws’ Filters along y-Axis

Vertical Filters

$$(L3)^T: \quad \frac{1}{6}\begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix}$$

$$(E3)^T: \quad \frac{1}{2}\begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix}$$

$$(S3)^T: \quad \frac{1}{2}\begin{pmatrix} -1 \\ 2 \\ -1 \end{pmatrix}$$

1D tensor

2D Laws' Filters (3x3 Filter Masks)

2D tensor

tensor product

$$(E3) \otimes (L3)^T$$

$$\frac{1}{2}(-1, 0, 1) \otimes \frac{1}{6}\begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix}$$

$$= \frac{1}{12}\begin{pmatrix} -1, & 0, & 1 \\ -2, & 0, & 2 \\ -1, & 0, & 1 \end{pmatrix}$$

Understanding Laws' Filters (A DSP Approach)

L3: $\frac{1}{6}(1, 2, 1)$

local averages

E3: $\frac{1}{2}(-1, 0, 1)$

edge

S3: $\frac{1}{2}(-1, 2, -1)$

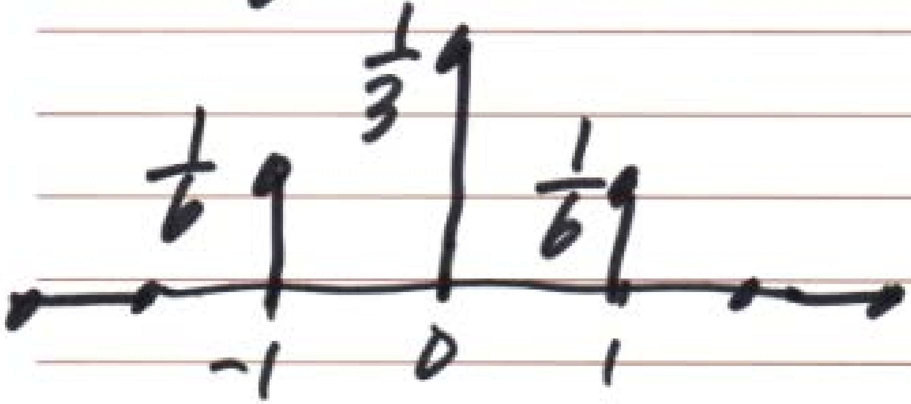
spot

1D tensor

Analysis of Filter L3

- Impulse response

$L3: \frac{1}{6}(1,2,1)$



$h_{L3}[n] = \frac{1}{6}\delta[n-1] + \frac{1}{3}\delta[n] + \frac{1}{6}\delta[n+1]$

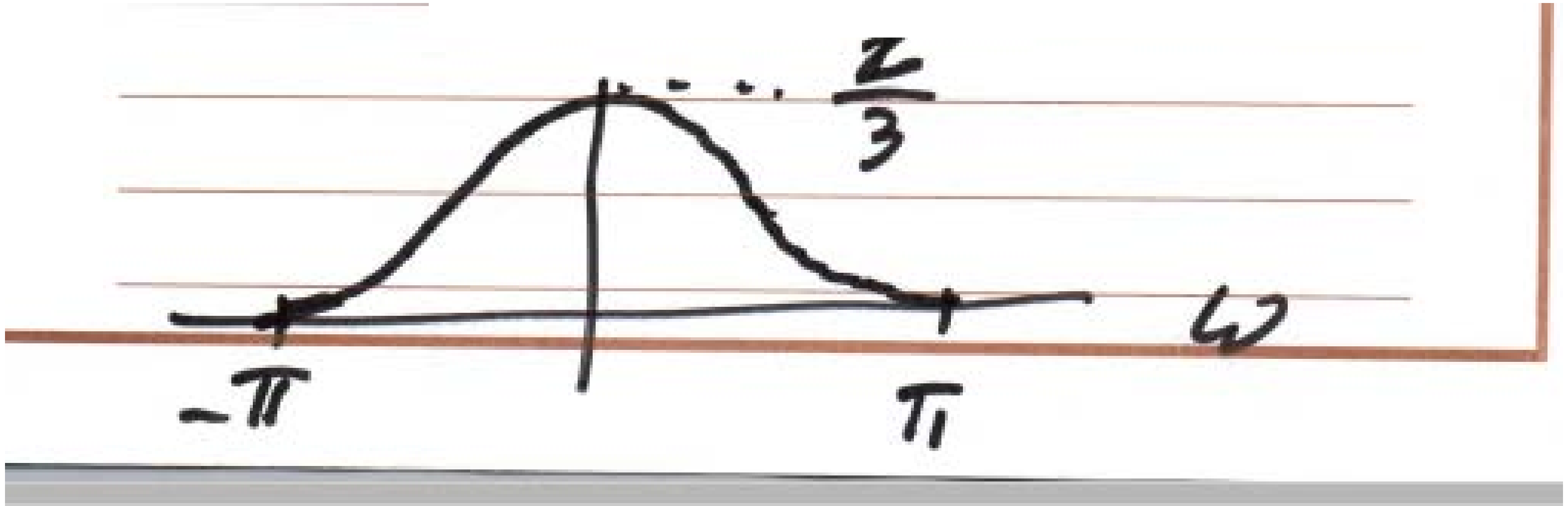
Frequency Response of Filter L3 (1)

input $e^{i\omega n}$

$$H_{L3}(e^{i\omega n}) = \underbrace{\frac{1}{6}\left(e^{-j\omega} + 2 + e^{j\omega}\right)}_{\text{eigenvalue (response.)}} \cancel{e^{i\omega n}}$$

$$= \frac{1}{3}\left(1 + \cos\omega\right).$$

Frequency Response of Filter L3 (2)



L3 is a lowpass filter

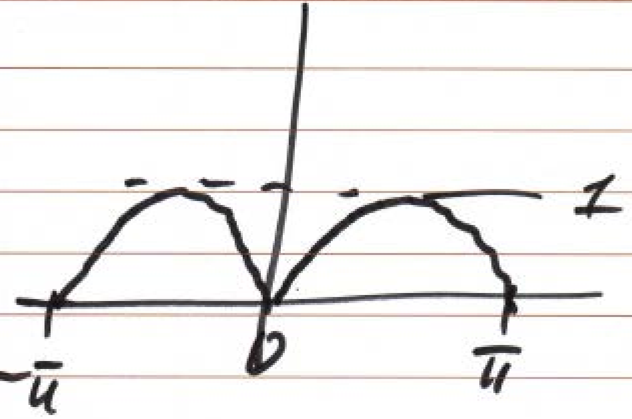
Analysis of Filter E3

- E3: 0.5 x (-1, 0, 1)
- Frequency Response
    - E3 is a bandpass filter

$$H_{E3}(e^{i\omega}) = \frac{1}{2}(-e^{-i\omega} + e^{i\omega})$$

$$= j\sin\omega$$

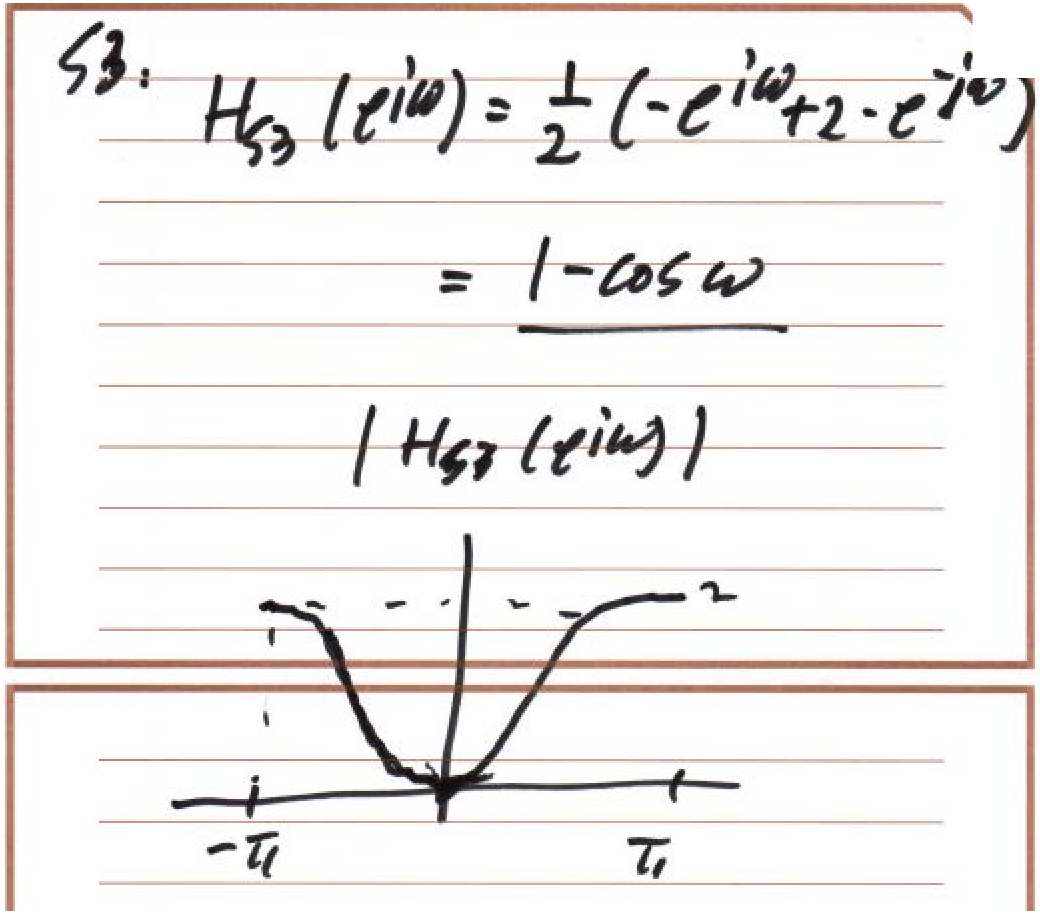
$$|H_{E3}(e^{i\omega})|$$



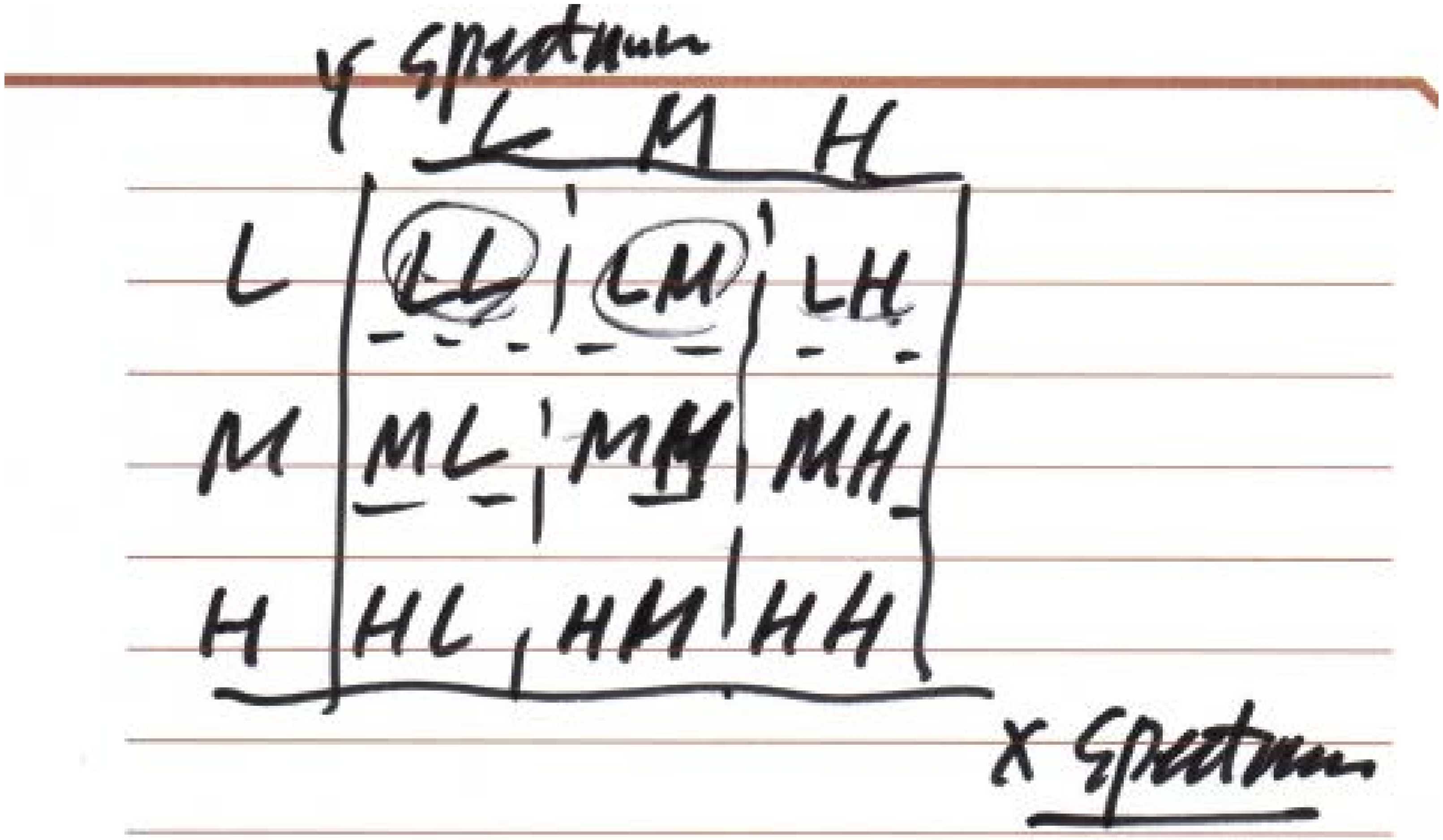
E3 is a bandpass filter.

Analysis of Filter S3

- S3: 0.5 x (-1, 2, -1)
- Frequency Response
  - S3 is a highpass filter



Responses of 2D Filters in A 3x3 Patch

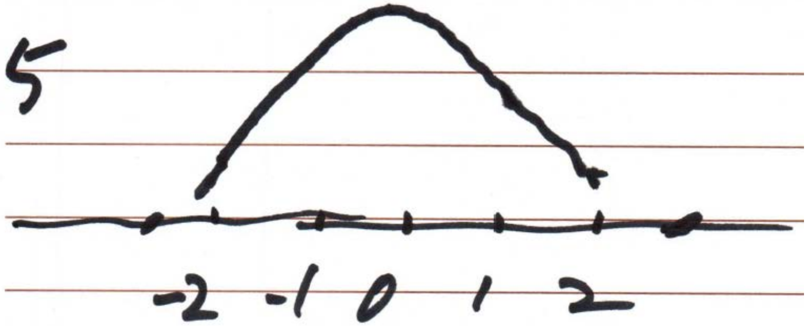


Laws' 5x5 Filters

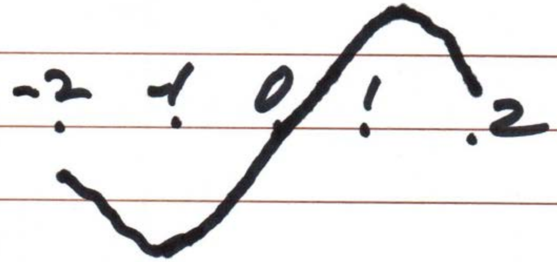
Table 1: 1D Kernel for 5x5 Laws Filters

Name	Kernel
L5 (Level)	[1 4 6 4 1]
E5 (Edge)	[-1 -2 0 2 1]
S5 (Spot)	[-1 0 2 0 -1]
W5 (Wave)	[-1 2 0 -2 1]
R5 (Ripple)	[1 -4 6 -4 1]

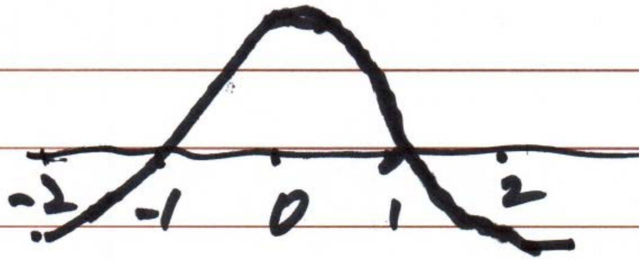
L5



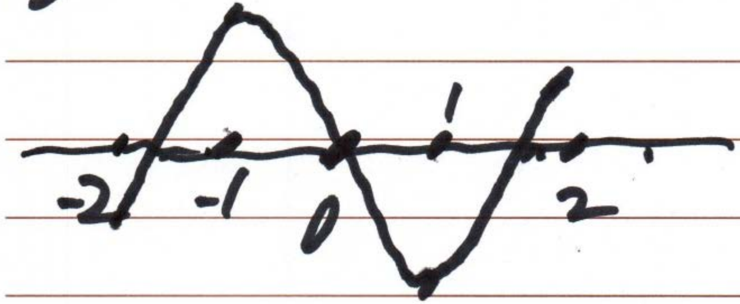
E5



S5



W5

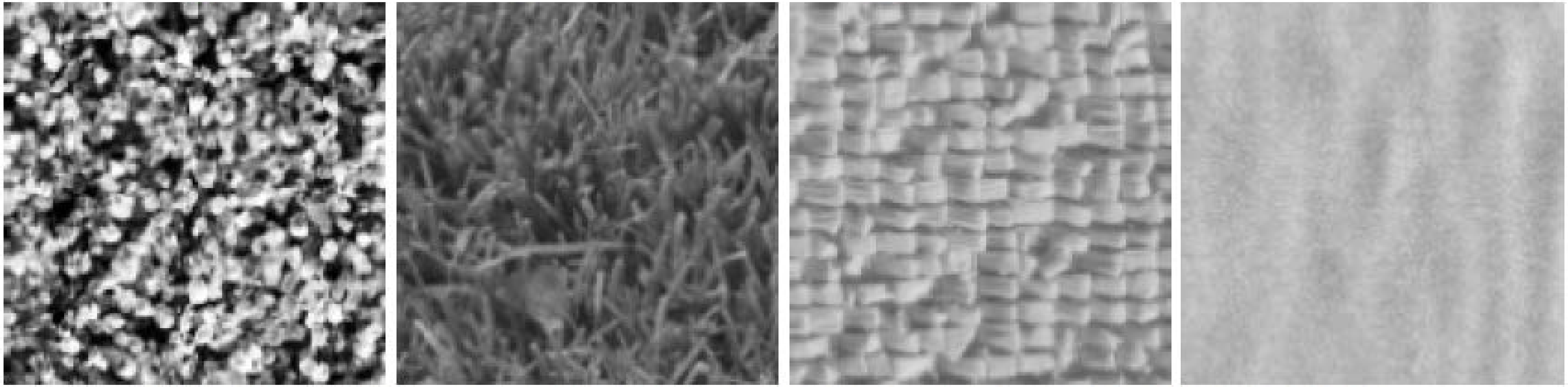


R5

Exercise:

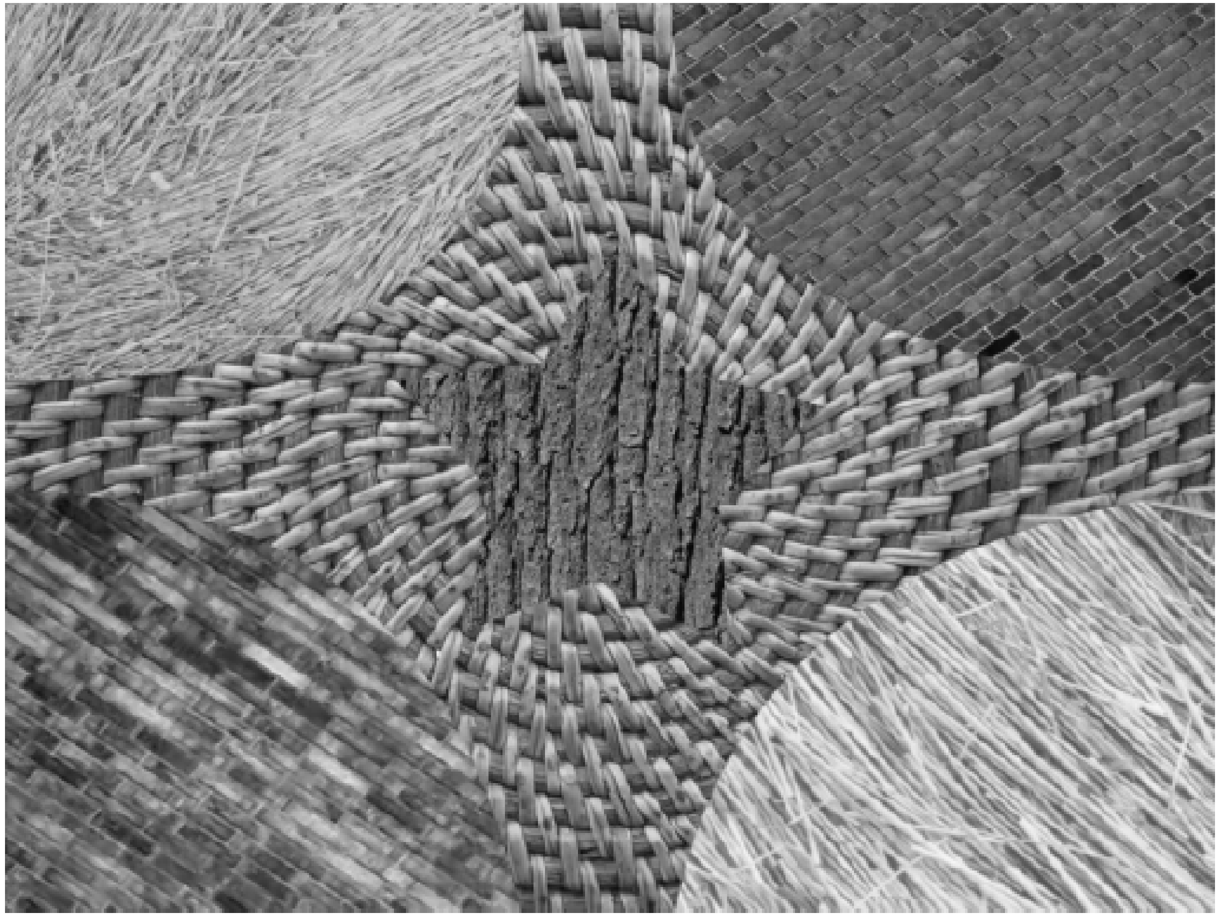
Conduct Frequency Responses on These Five Filters

Texture Classification Problem



Four types of textures: rock, grass, weave, and sand

Texture Segmentation Problem



Texture Mosaic

Pixelwise Response and Its 2nd Order Statistics

- Scan the entire texture image pixel by pixel (stride = 1) using a bank of filters (e.g. 3x3 Laws' filters)
  - 9 filters -> 9 responses -> a random vector of 9 dimensions -> response vector
- Find the second-order statistics of the response vector
  - Mean vector
    - The LL response has a non-zero mean
    - The remaining 8 responses have the same mean (zero mean)
  - Covariance matrix
    - Symmetric matrix
    - Diagonally dominant matrix
      - Weak correlation between different elements of the feature vector

Ordering of Laws' Filters

$$\frac{1}{36}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Laws 1

$$\frac{1}{12}\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

Laws 2

$$\frac{1}{12}\begin{bmatrix} -1 & 2 & -1 \\ -2 & 4 & -2 \\ -1 & 2 & -1 \end{bmatrix}$$

Laws 3

$$\frac{1}{12}\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Laws 4

$$\frac{1}{4}\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$

Laws 5

$$\frac{1}{4}\begin{bmatrix} -1 & 2 & -1 \\ 0 & 0 & 0 \\ 1 & -2 & 1 \end{bmatrix}$$

Laws 6

$$\frac{1}{12}\begin{bmatrix} -1 & -2 & -1 \\ 2 & 4 & 2 \\ -1 & -2 & -1 \end{bmatrix}$$

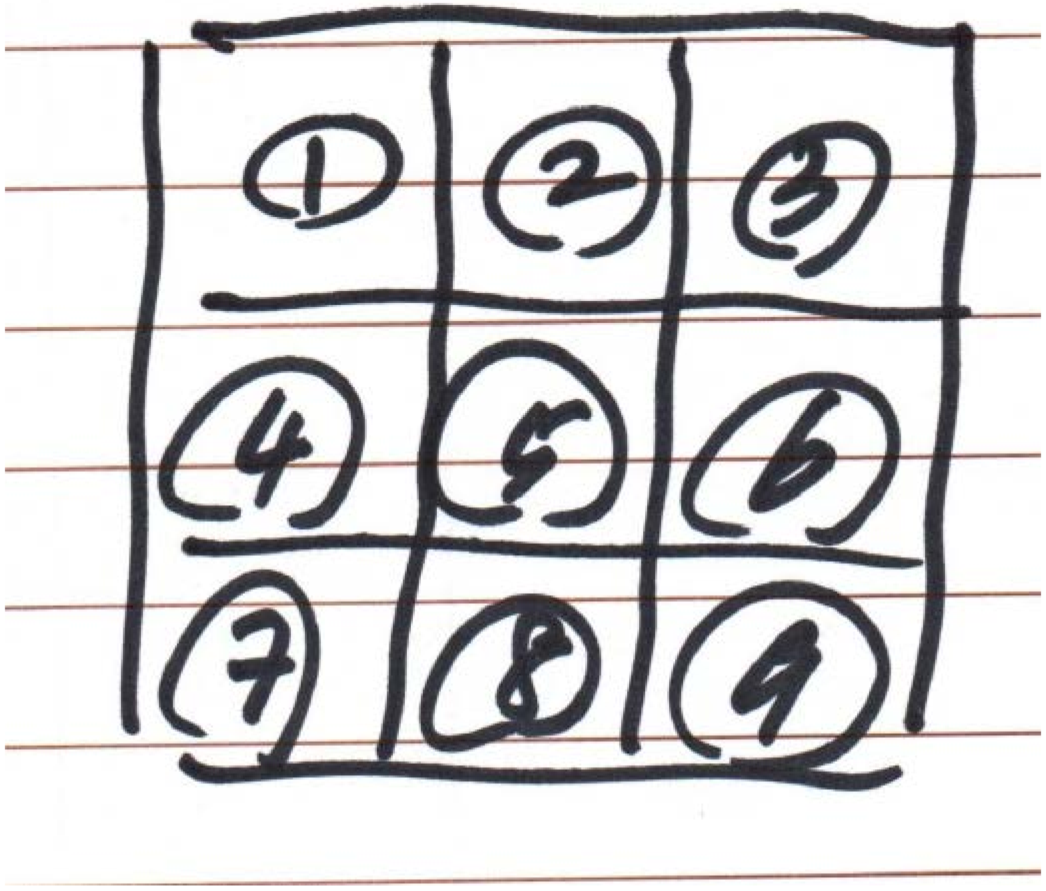
Laws 7

$$\frac{1}{4}\begin{bmatrix} -1 & 0 & 1 \\ 2 & 0 & -2 \\ -1 & 0 & 1 \end{bmatrix}$$

Laws 8

$$\frac{1}{4}\begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

Laws 9



2^{nd} Order Statistics of Response Vector

$$\underline{r} = \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_q \end{pmatrix} \qquad \underline{m}_{\underline{r}} = \begin{pmatrix} m_{\underline{r}_1} \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

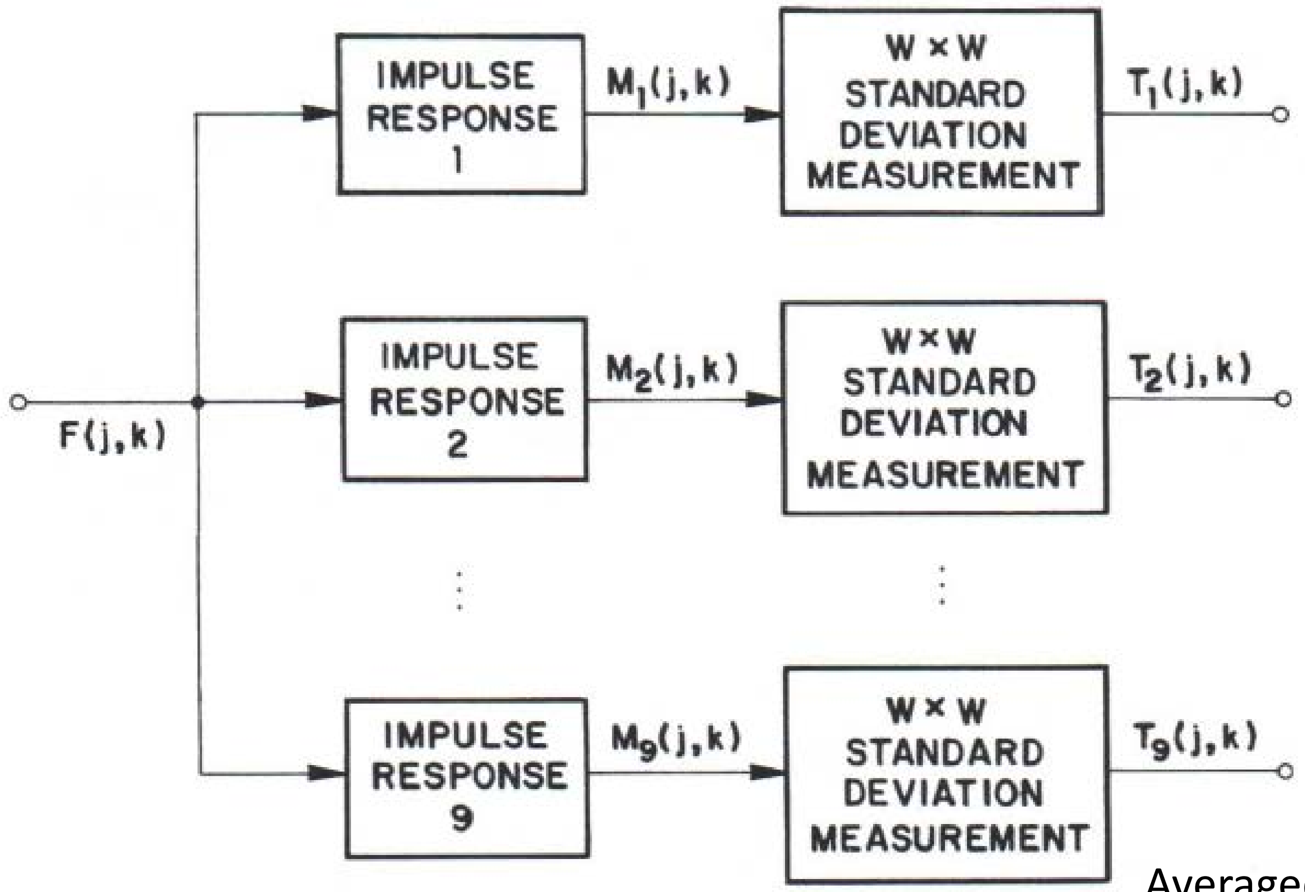
$$K_{\underline{r}} = E\left[(\underline{r} - \underline{m}_{\underline{r}})(\underline{r} - \underline{m}_{\underline{r}})^T\right]$$

$$= \begin{pmatrix} \sigma_{11} & \sigma_{21} & & \\ \sigma_{12} & \sigma_{22} & & \\ & & \ddots & \\ & & & \sigma_{qq} \end{pmatrix}.$$

Diagonal elements:
- Intra-channel (self) correlation
Off-diagonal elements
- Inter-channel correlation

Laws' Filter Energy Feature Extraction Block-diagram

Pratt's Book
Page 566



Response Vector

Averaged Energy Vector over a WxW window

Energy Feature Vector

① $\sigma_{ii} >> \sigma_{ij} \quad j \neq i$

② focus on σ_{ii}

$\sigma_{ii} = |f_i|^2$ energy of the ith channel

Why Averaging over WxW Window ?

9-dimensional feature vector at pixel location (i,j)

$$\underline{f}(i,j) = \begin{pmatrix} |r_1(i,j)|^2 \\ \vdots \\ \vdots \\ |r_9(i,j)|^2 \end{pmatrix}$$

Statistical Fluctuation

Digression: Basic Machine Learning

Supervised versus Un-supervised Machine Learning

- Supervised ML
  - Training data with labels provided by humans
  - Heavily supervised ML
    - The # of training samples is much larger than the # of test samples
  - Weakly supervised ML
    - The # of training samples is much smaller than the # of test samples
- Un-supervised ML
  - No training data with human labels
- Examples
  - Sobel and Canny edge detectors are un-supervised methods
  - Structured edge detector is a supervised method

Commonly Used Supervised Classifiers in Machine Learning

- Nearest Neighbor (NN) classifier
- k Nearest Neighbor (kNN) classifier
- Support vector machine (SVM)
- Random forest (RF)
- Multi-layer Perceptron (MLP)
- Adaptive Boosting (Adaboost)
- Gradient Boosting and Extremely Gradient Boosting (XGBoost)

Distance-Based Classifiers

- General principle
  - Compute the distance of the feature vector of a test image from each class' centroid and select the class that gives the minimum distance
- What kind of distance
  - Euclidean distance?
    - This is fine if the variance of each dimension is normalized to unity
    - Otherwise, we should consider Mahalanobis distance

Mahalanobis distance (or "generalized squared interpoint distance" for its squared value[3]) can also be defined as a dissimilarity measure between two random vectors $\vec{x}$ and $\vec{y}$ of the same distribution with the covariance matrix S:

$$d(\vec{x},\vec{y}) = \sqrt{(\vec{x}-\vec{y})^T S^{-1} (\vec{x}-\vec{y})}.$$

Mahalanobis Distance

If the covariance matrix is the identity matrix, the Mahalanobis distance reduces to the Euclidean distance. If the covariance matrix is diagonal, then the resulting distance measure is called a *standardized Euclidean distance*:

$$d(\vec{x},\vec{y}) = \sqrt{\sum_{i=1}^{N} \frac{(x_i - y_i)^2}{s_i^2}},$$

where s_i is the standard deviation of the x_i and y_i over the sample set.

Unsupervised Classifier – K-means Clustering (1)

- Objective:
  - Cluster N feature vectors of dimension D, denoted by R^D, into K clusters to minimize the total distortion between each feature vector and its associated cluster centroid
- Initialization (m=0)
  - Select K feature vectors as the initial set of cluster centroids, called a codebook
- Generalized Lloyd Iteration (m=0,1,…)
  - Given codebook $C_m = \{ y_i, i = 1, \ldots, K \}$ obtained from the m^{th} iteration, find a new optimal partitioning of space R^D using the nearest-neighbor condition to form the nearest-neighbor cells $R_i = \{ x: d(x,y_i) < d(x,y_j), j \neq i \}$

Unsupervised Classifier – K-means Clustering (1)

- Centroid update
  - Compute new centroids from new partition cells

$$C_{m+1} = \{ \text{centroid}(R_i),\ i = 1, \ldots, K \}$$

$$centroid(R_i) = \frac{1}{n_i} \sum_{j=1}^{n_i} x_j$$

n_i: No. of vectors in R_i

- Continue this iterative optimization procedure until obtain an optimum codebook

Complexity Analysis of GLA

- For one step of GLA, computational complexity is O(K*D*L)

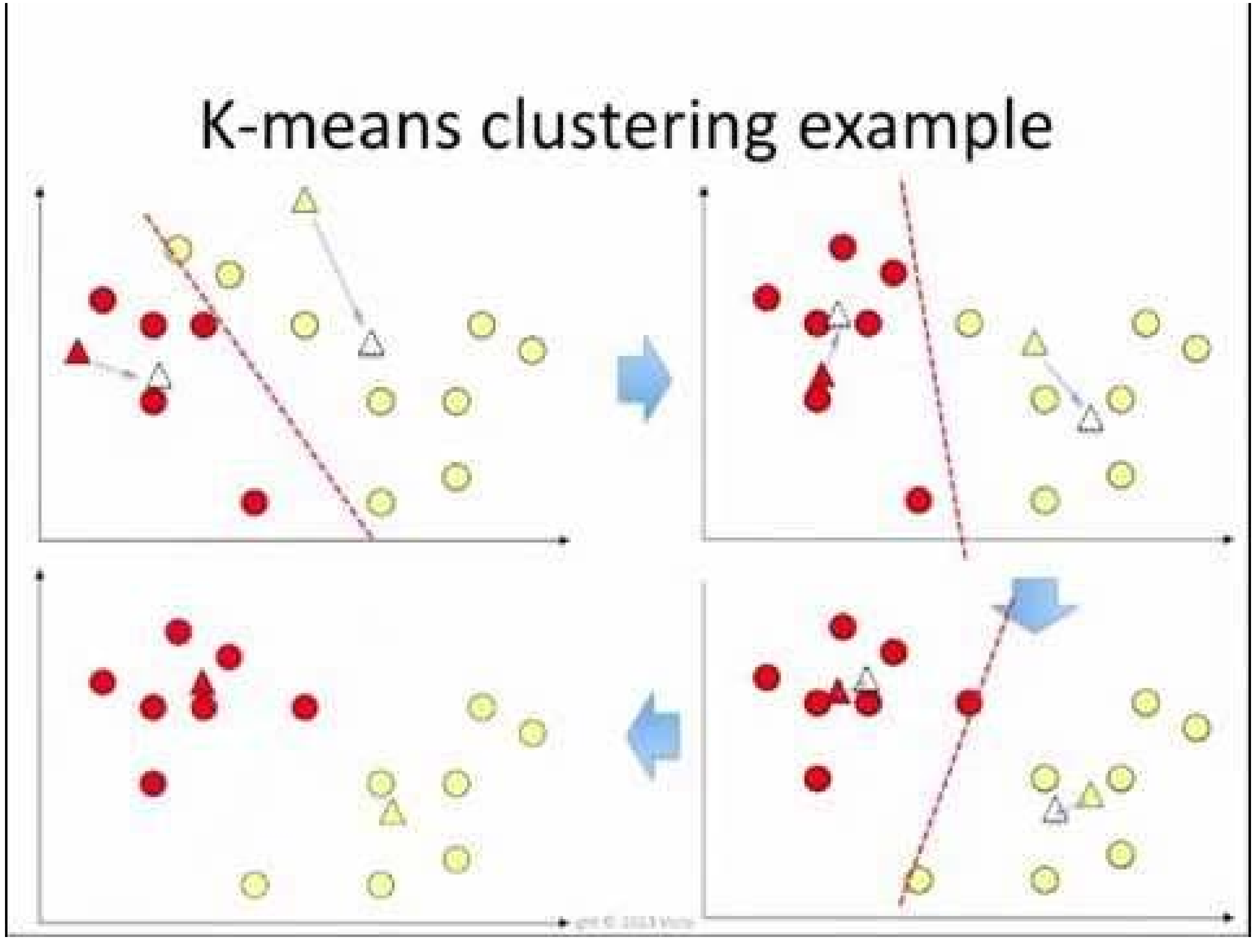
  K: codebook size

  D: vector dimension

  L: size of training vectors

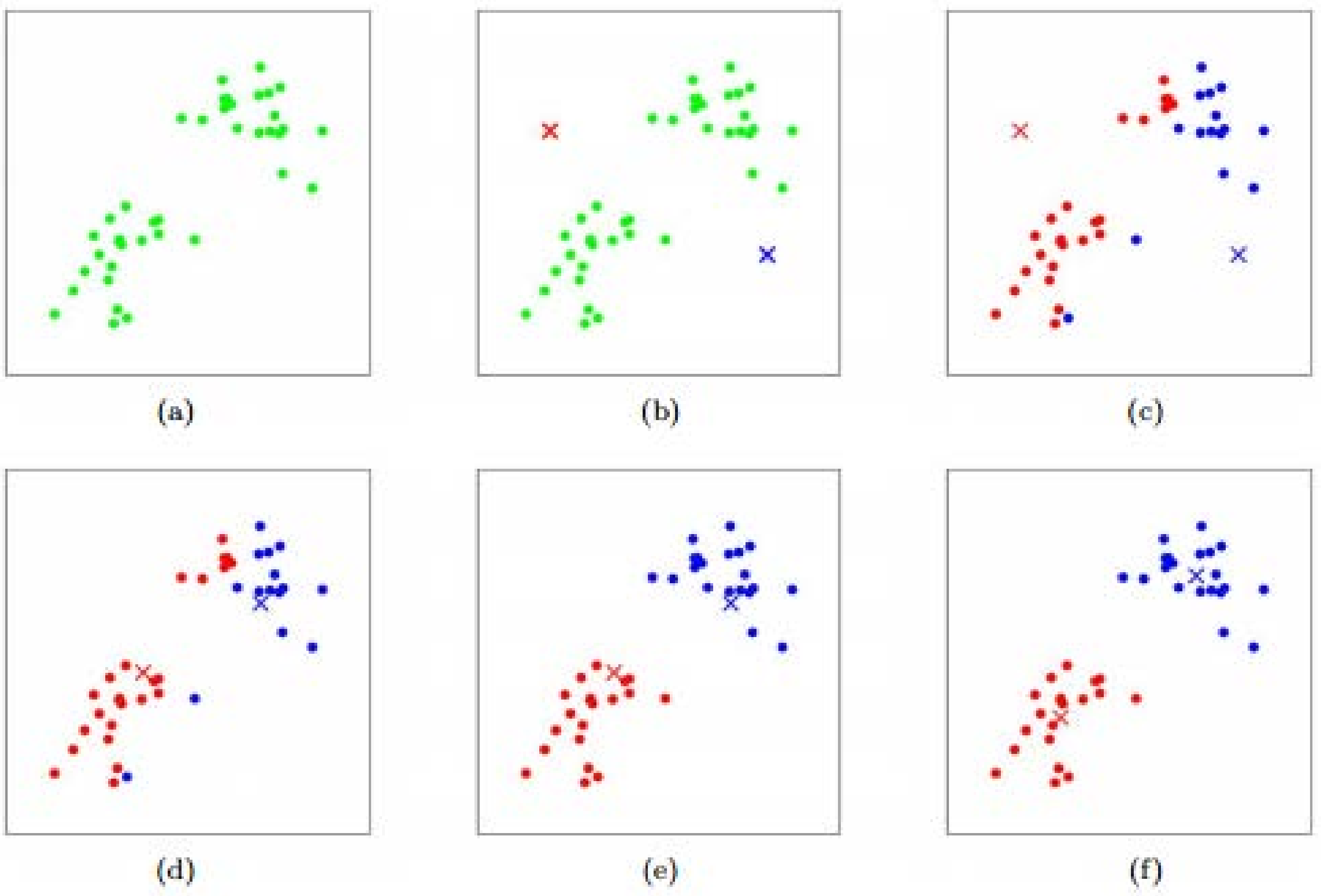
Graphic Illustration of K-means Clustering

Example 1



Graphic Illustration of K-means Clustering

Example 2



Comments on K-means Clustering

- Non-convex optimization
  - There are multiple local minima
  - The converged solution is dependent on the initial centroid choice
- The choice of the optimal K value is a problem

What is the optimal K value?

Application of K-means Clustering

General image segmentation is challenging!

Application of K-means Clustering

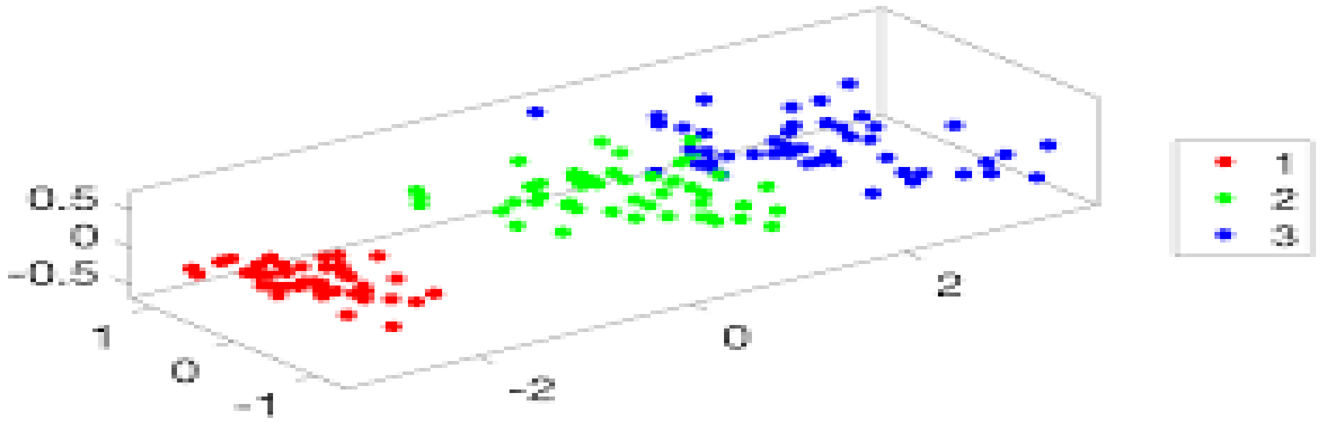
Application-specific image segmentation is more manageable

Semantic Segmentation

Instance Segmentation

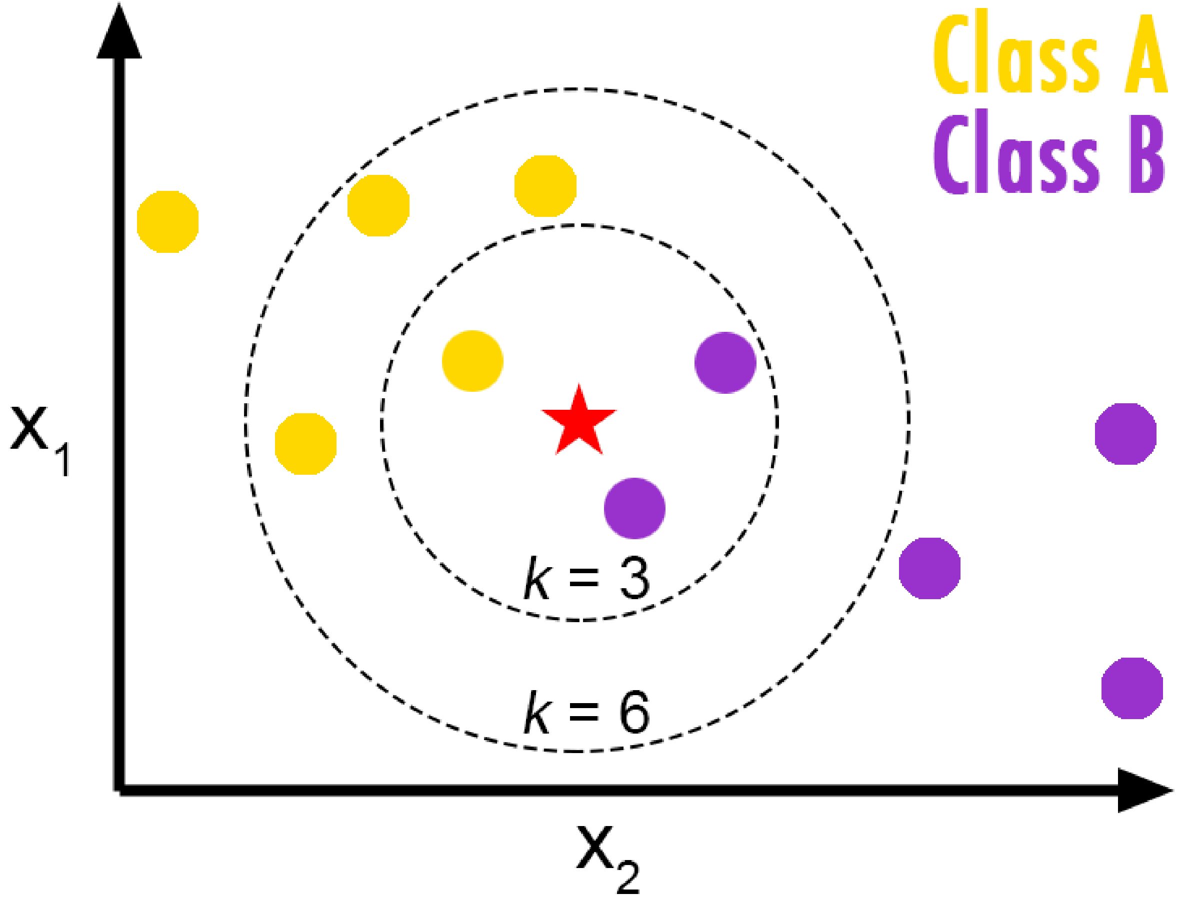
Supervised Distance-based Classifier (1)

- Nearest Neighbor Classifier
  - Intra-cluster variation is smaller than inter-cluster variation
  - Compute the centroid of each cluster
  - Choose the class that has the smallest test sample-to-centroid distance



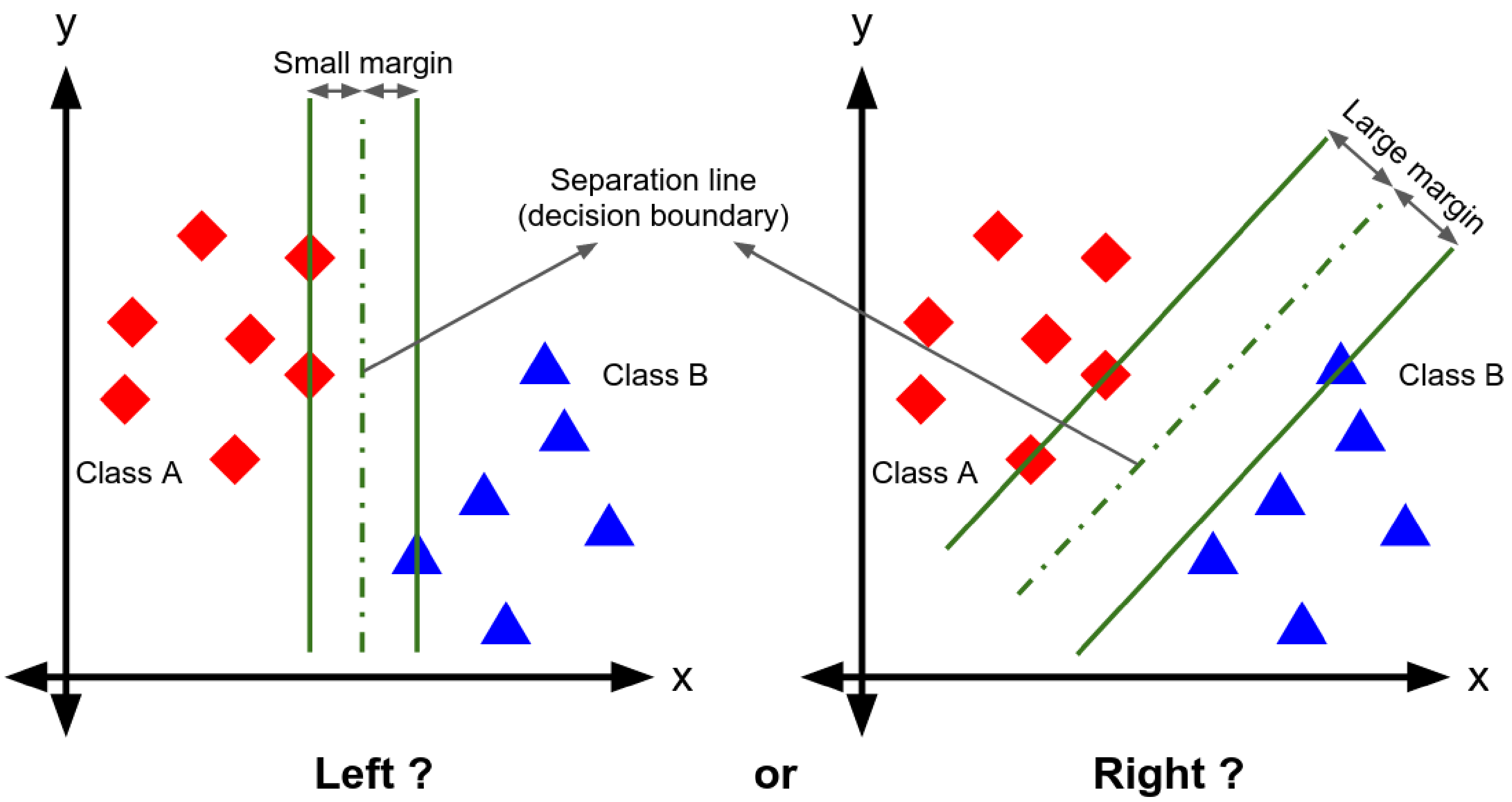
Supervised Distance-based Classifier (2)

- KNN Classifier

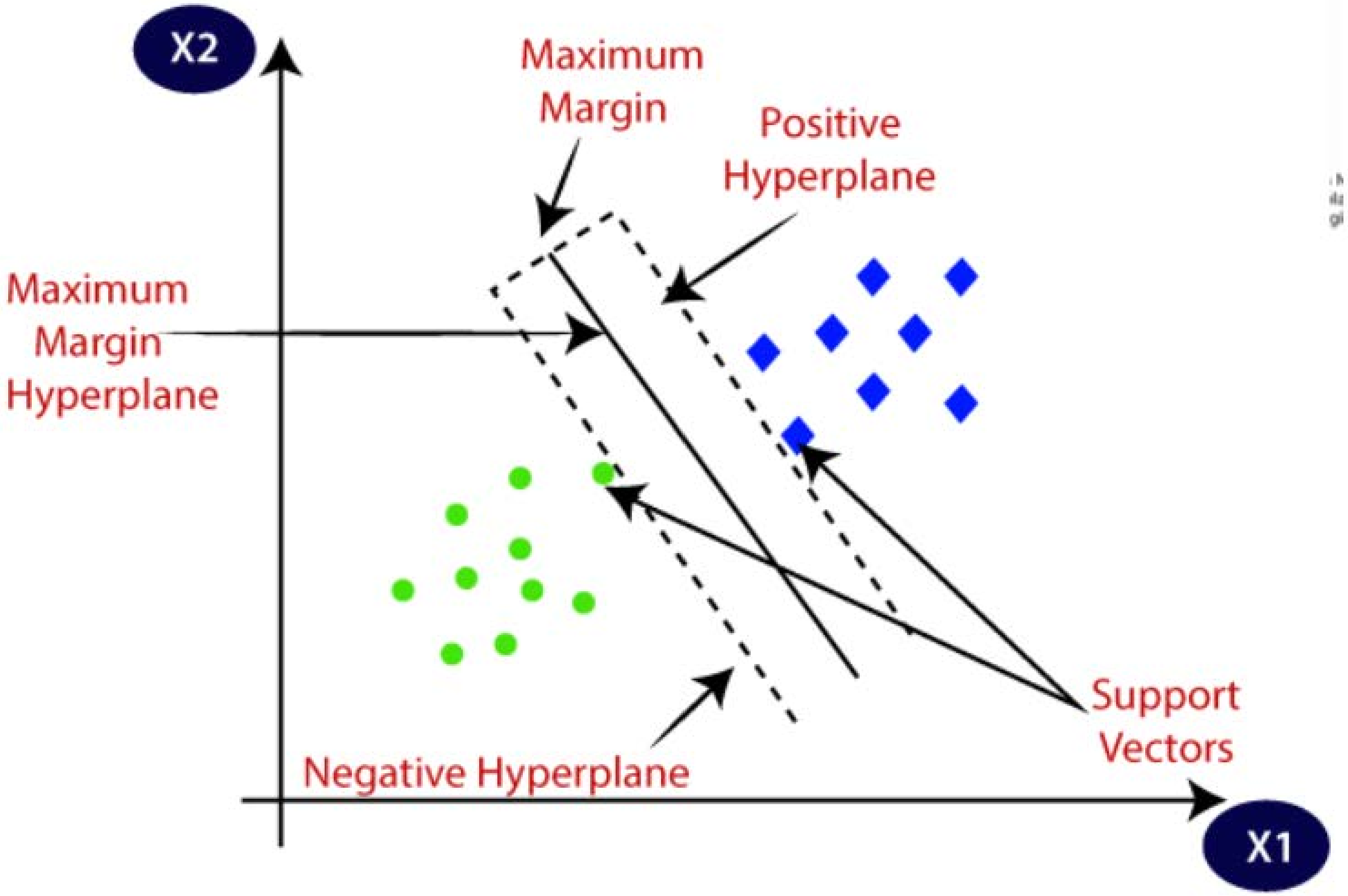


Other Supervised Classifiers: SVM (1)

- Support Vector Machine (SVM) classifier

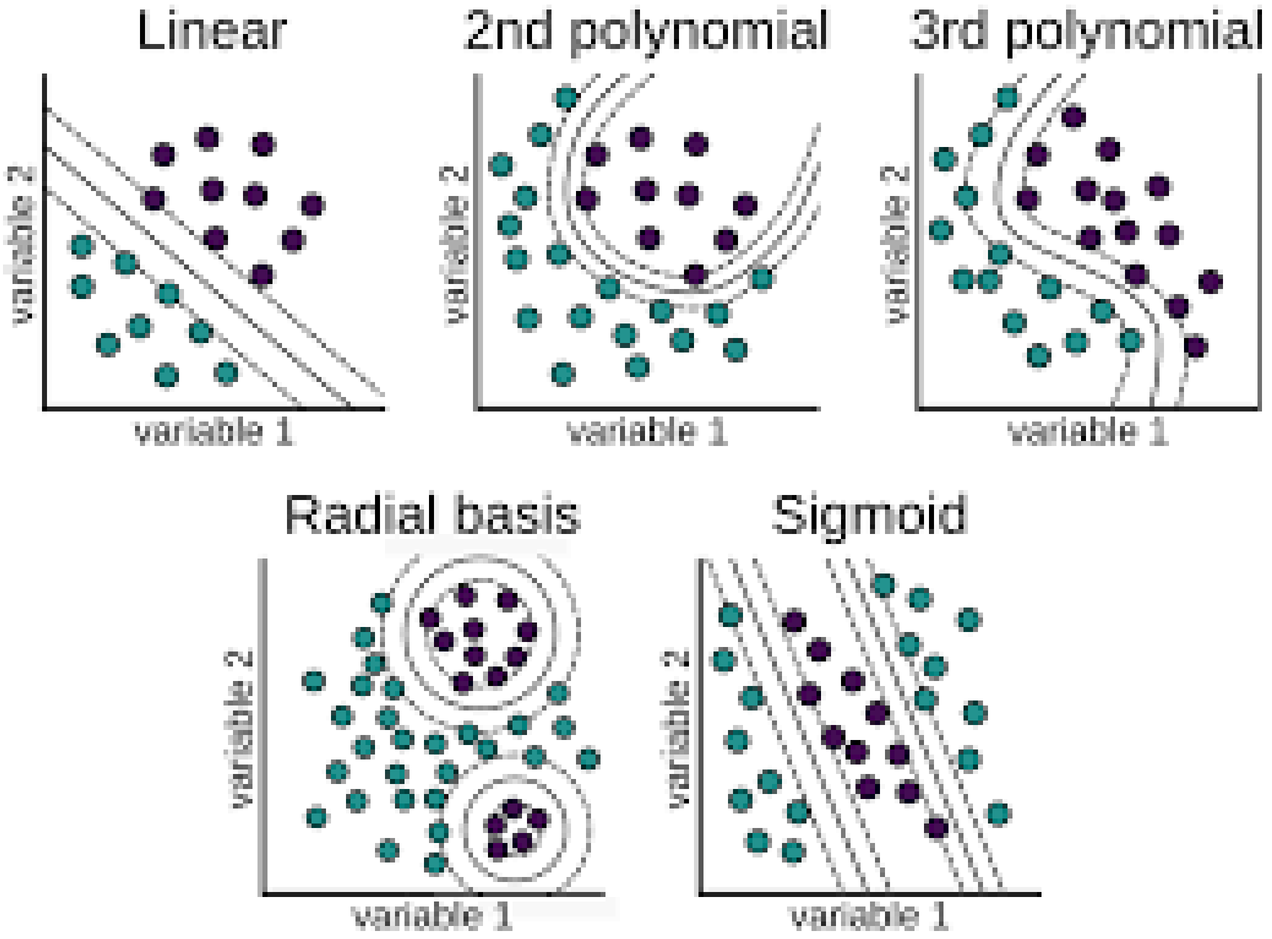


Other Supervised Classifiers: SVM (2)



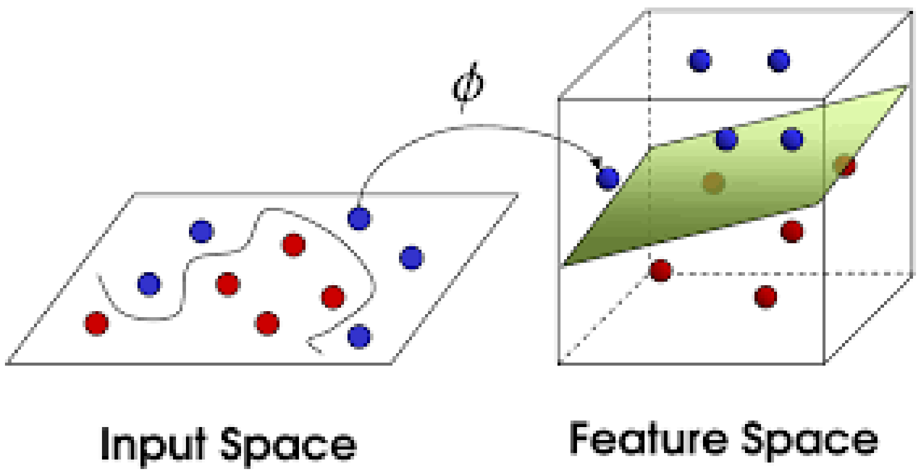
Other Supervised Classifiers: SVM (3)

Nonlinear SVM

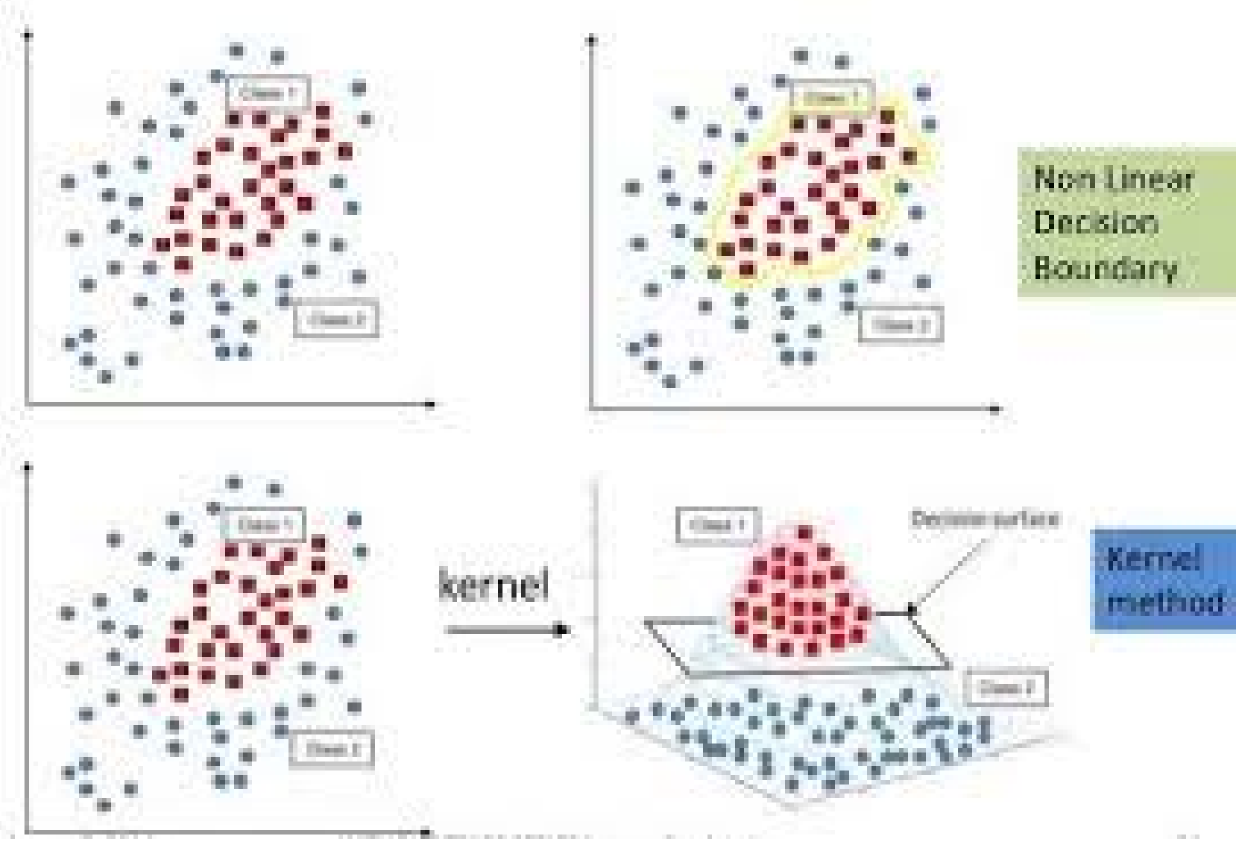


Other Supervised Classifiers: SVM (4)

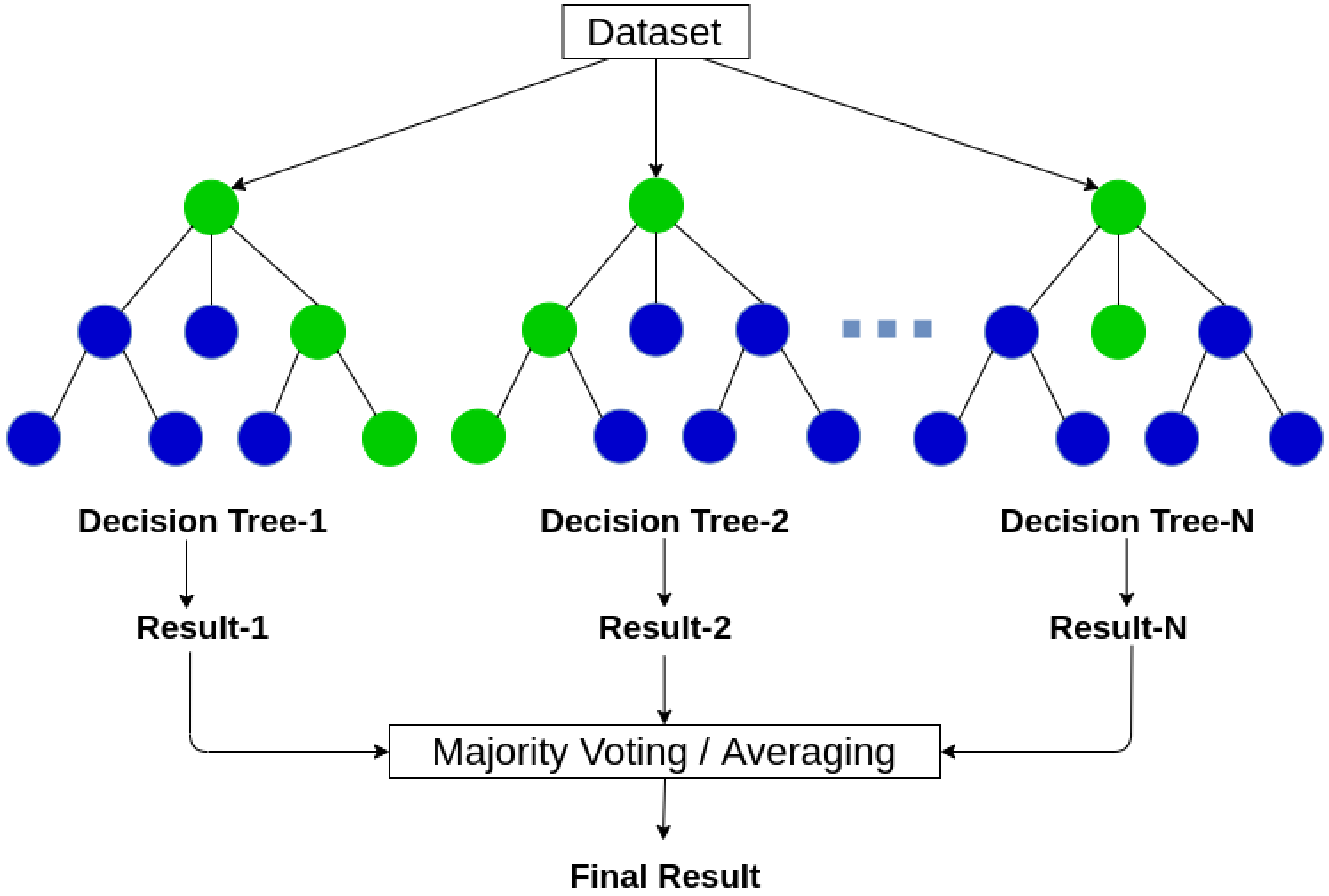
Example 1



Example 2



Other Supervised Classifier: Random Forest



Back to
Texture Classification and Segmentation

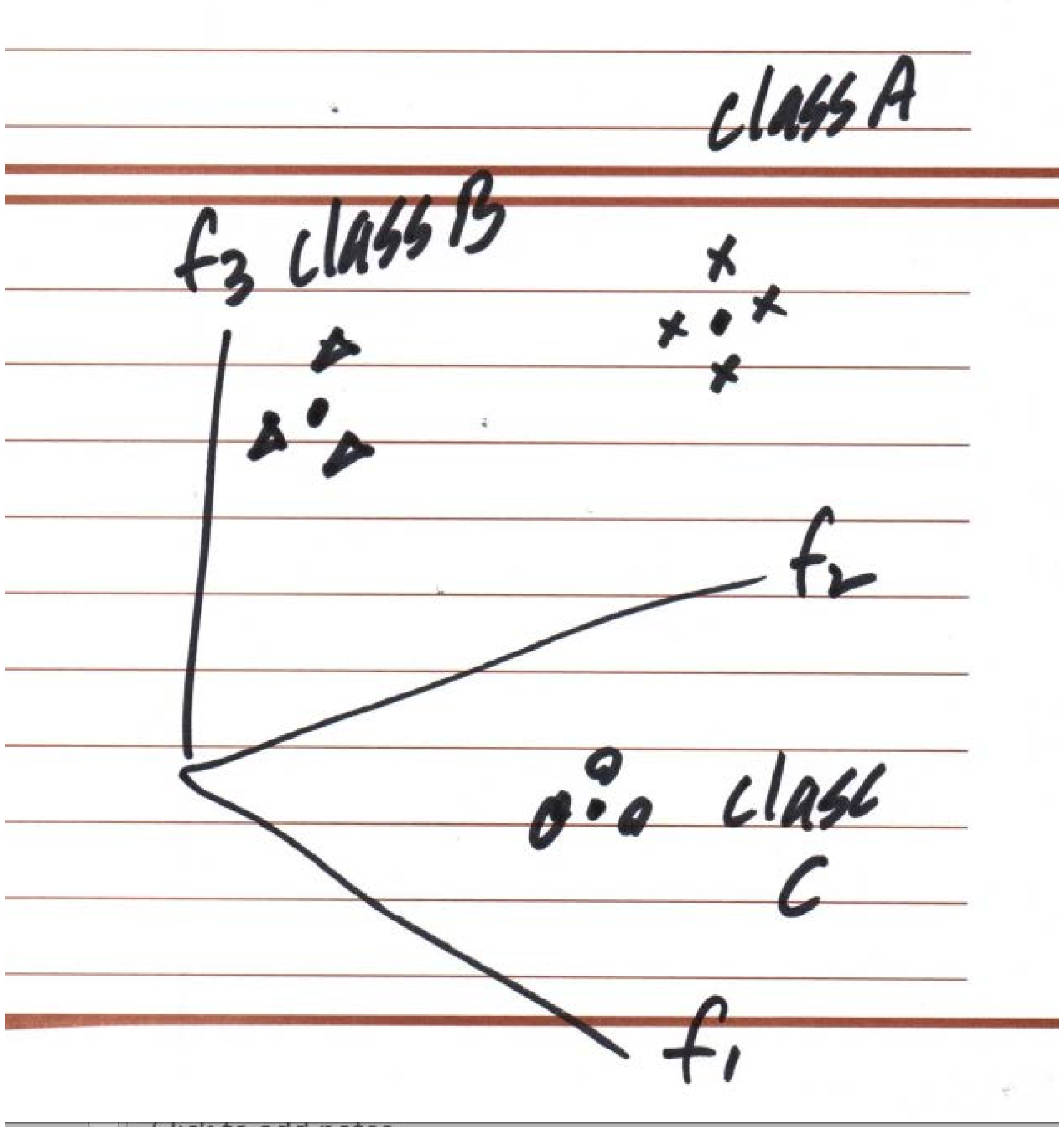
Differences

- Texture classification is usually treated as a supervised learning problem
  - Offer exemplar texture images from several texture types (e.g. texture type A, texture type B, texture type C, etc.)
  - Given test image X, please find its texture type
- Texture segmentation is typically treated as an unsupervised learning problem
  - One texture mosaic image that contains multiple texture types

Texture Classification

- Choose the window size to be the same as the image size
  - Namely, take the average of feature vectors at all pixel locations
- Suppose that there are C texture classes, where each class has Nc training images
  - Find the feature vector of each training image
  - Average the feature vectors of training images
    - Centroid of each class

Feature Space

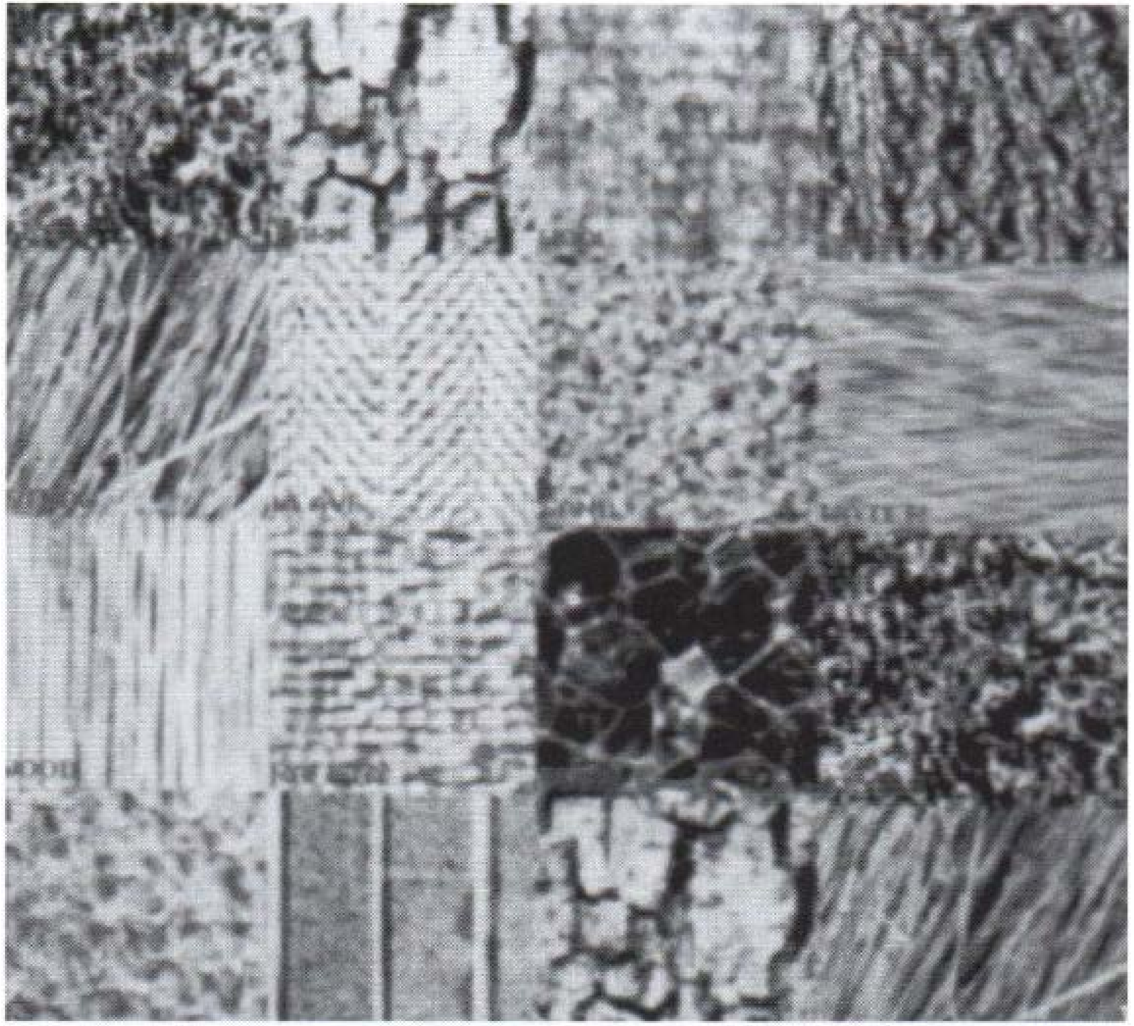


Texture Segmentation

- Set window size to W=13, 15 or 17
- Tradeoff:
  - For a larger window size
    - The averaged feature vector is more stable for pixels in the interior region
    - The averaged feature vector can cover multiple texture types more easily for pixels close to boundaries
  - For a smaller window size
    - The averaged feature vector could fluctuate more for pixels in the interior region
    - The average feature vector tends to cover fewer texture types for pixels close to boundaries
  - Consider a hybrid solution
    - Two-pass algorithm
    - 1st pass – larger window size
    - 2nd – smaller window size

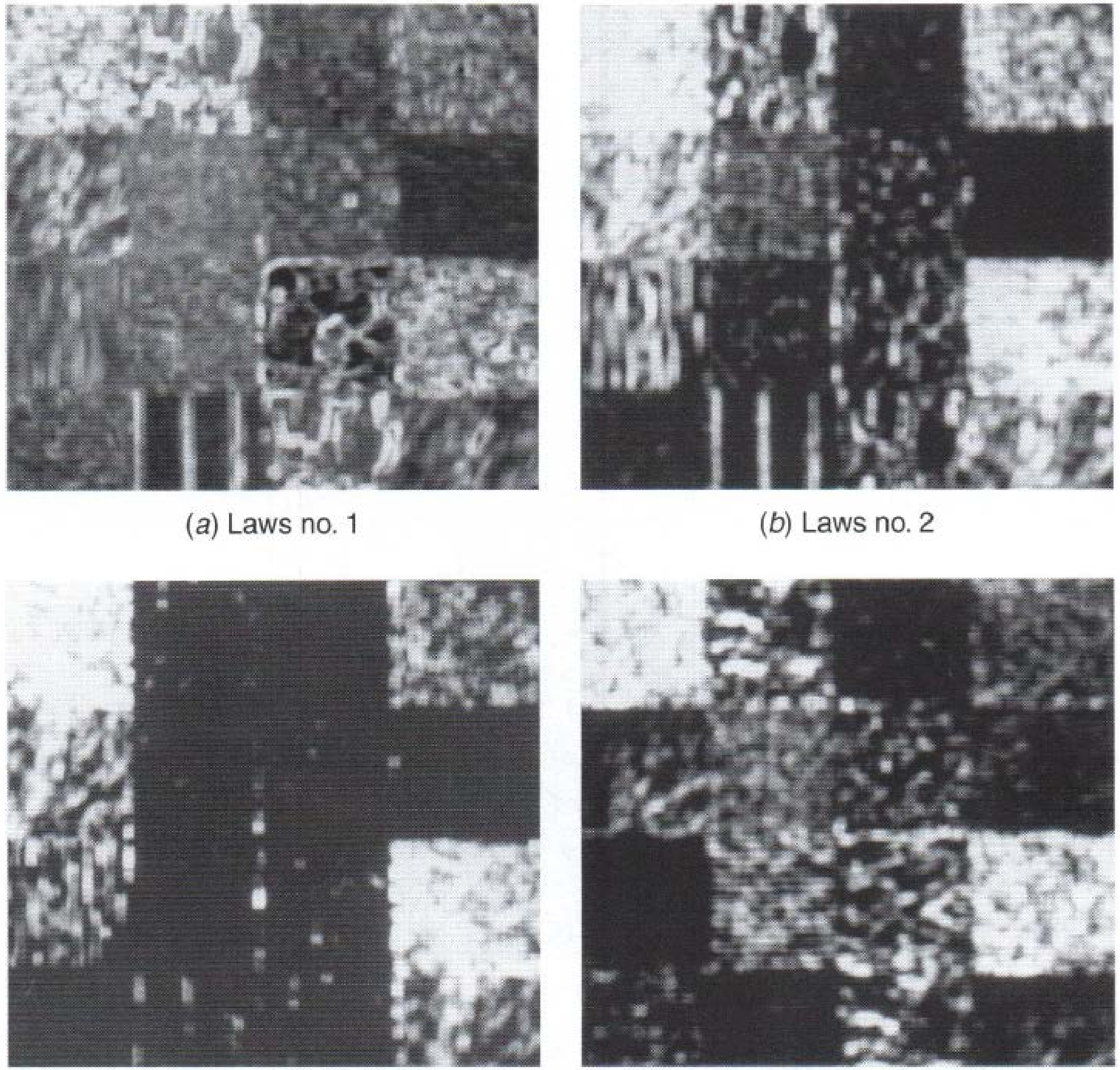
Example of Texture Mosaic

Pratt's Book
Page 568



Energy Feature Maps of Different Channels (1)

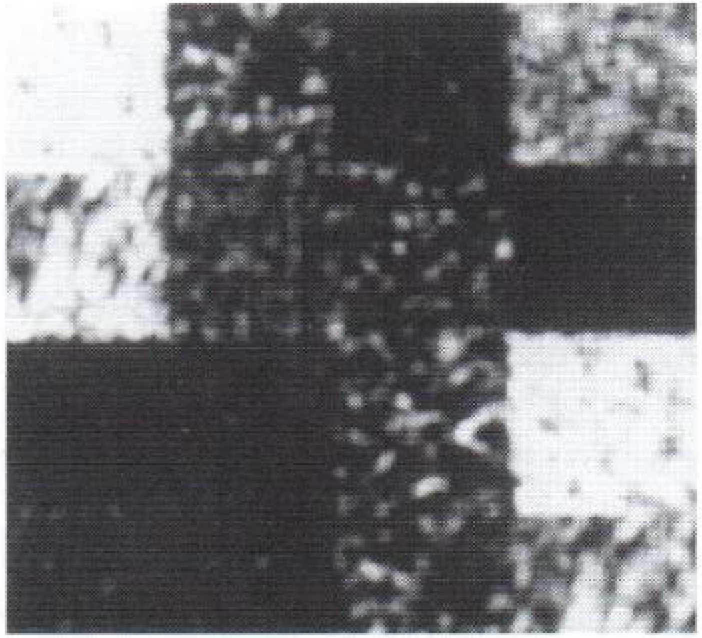
Pratt's Book
Page 569



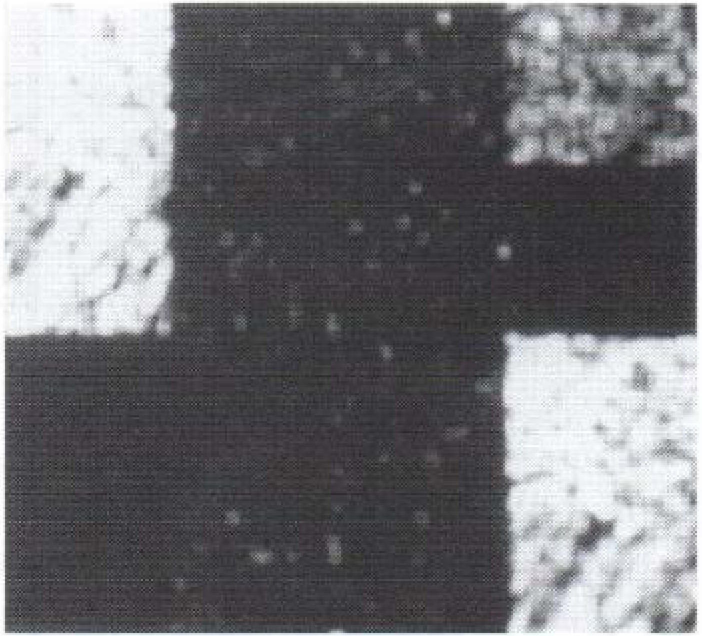
(*a*) Laws no. 1
(*b*) Laws no. 2
(*c*) Laws no. 3
(*d*) Laws no. 4

Energy Feature Maps of Different Channels (2)

Pratt's Book
Page 569



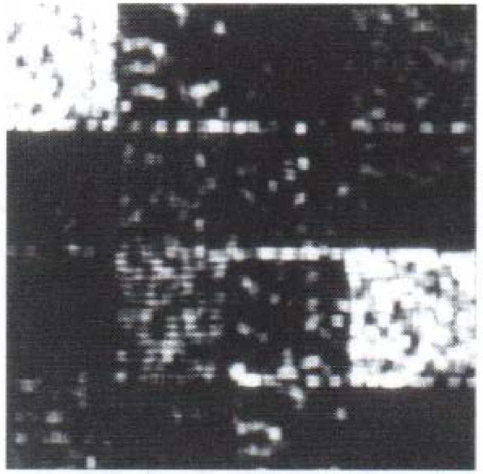
(e) Laws no. 5



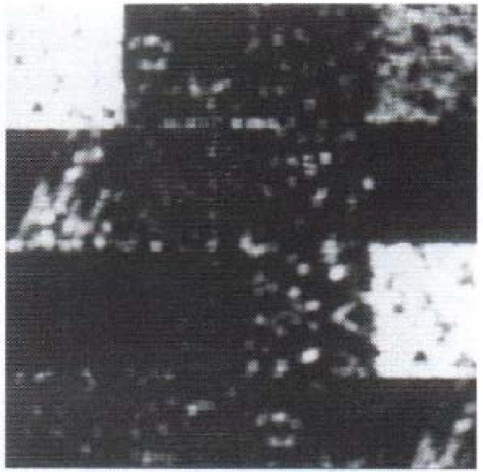
(f) Laws no. 6

Energy Feature Maps of Different Channels (3)

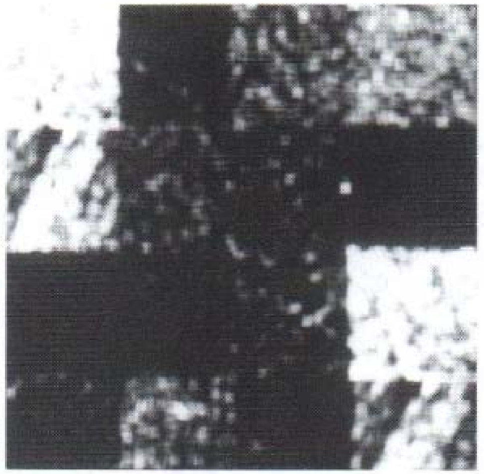
Pratt's Book
Page 570



(*g*) Laws no. 7



(*h*) Laws no. 8



(*i*) Laws no. 9

Window Size Selection for Energy Feature Averaging

- Set window size to W=13, 15 or 17
- Tradeoff:
  - For a larger window size
    - The averaged feature vector is more stable for pixels in the interior region
    - The averaged feature vector can cover multiple texture types more easily for pixels close to boundaries
  - For a smaller window size
    - The averaged feature vector could fluctuate more for pixels in the interior region
    - The average feature vector tends to cover fewer texture types for pixels close to boundaries
  - Consider a hybrid solution
    - Two-pass algorithm
    - 1^{st} pass – larger window size
    - 2^{nd} – smaller window size

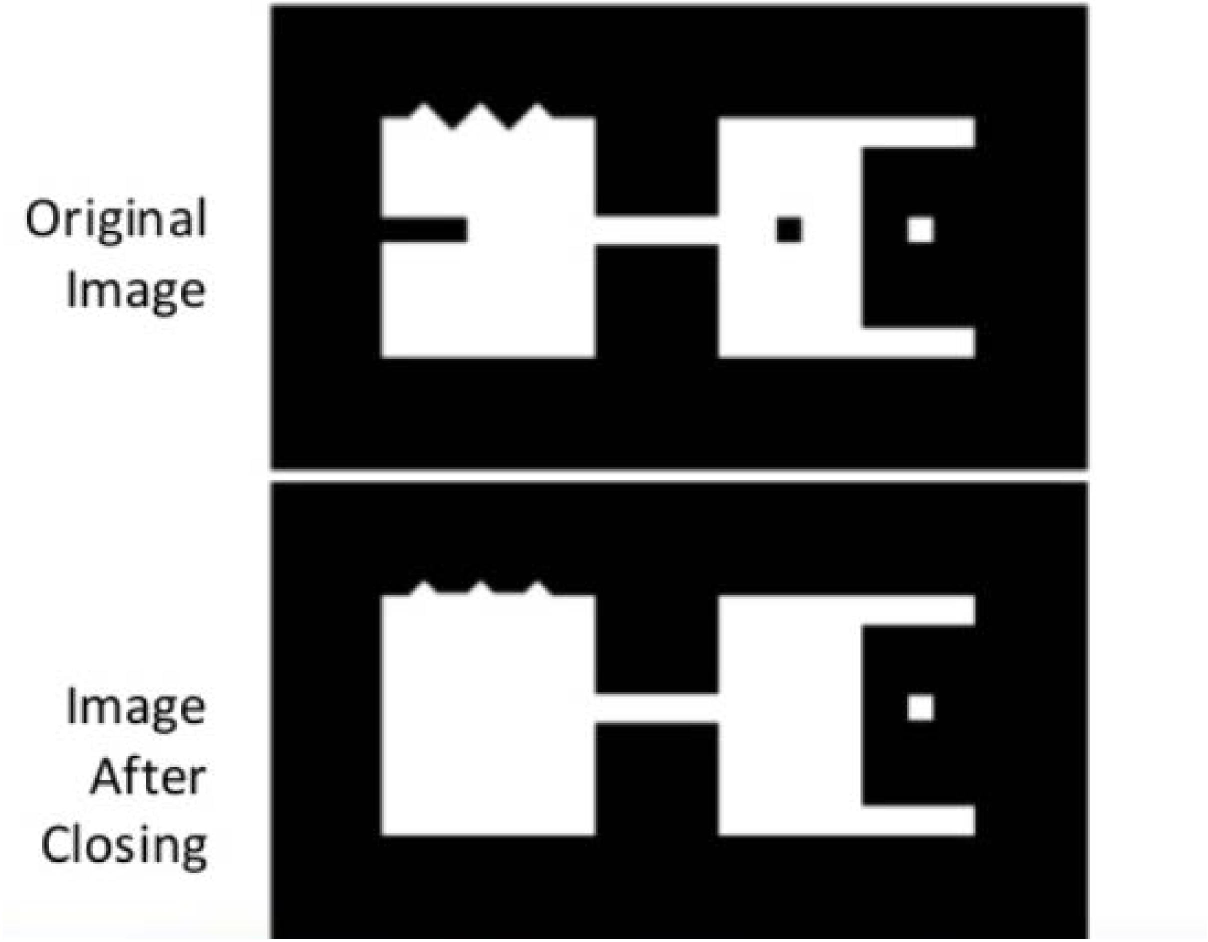
Criteria for Good Segmentation Results

- Qualitative Measure (rather than Quantitative)
  - Regions of a segmented image should be uniform and homogeneous w.r.t. some characteristics such as gray levels or texture
  - Region interiors should be simple and without small holes
  - Boundaries of each segment should be smooth, not ragged

Post-Processing of Segmentation Results

- Morphological processing to remove small holes
  - Identify small holes
  - Eliminate small holes with the close operation (structuring elements)
- Morphological processing to smooth boundaries

Recall: Closing Operation



You need to define the object and background properly
- Object (majority)
- Holes/background (minority)

Credit: https://www.slideshare.net/shkulathilake/morphological-image-processing-43465879