HA NOI UNIVERSITY OF SCIENCE AND TECHNOLOGY
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Biometric authentication systems

Chapter 4: Segmentation and object detection

1

Content

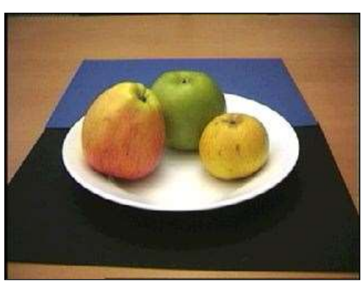
- Segmentation
- Object detection

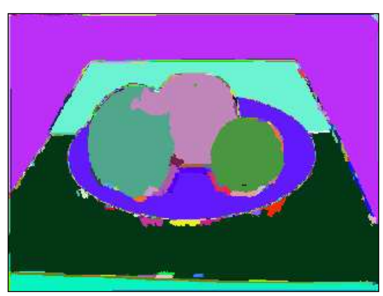
2

Introduction

- Purpose:
  - to partition an image into meaningful regions with respect to a particular application
- Goal:
  - to cluster pixels into salient image regions, i.e., regions corresponding to individual surfaces, objects, or natural parts of objects.
- The segmentation is based on the feature measurements taken from the image:
  - grey level, color, texture, depth or motion…

Introduction





Source : Jean-Christophe Baillie, ENSTA, uei.ensta.fr/baillie/assets/ES322%20-%20Segmentation.ppt

Introduction

- Entity can be extracted from images using mask

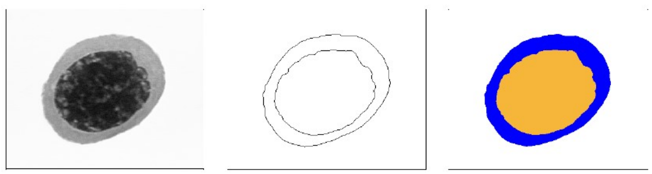
Source : Pascal Bertolino, Cours de Traitement d'images. LIS, INPG (France)

Approaches for image segmentation

- Segmentation is usually based on:
  - discontinuities: edges
    - sudden changes, borders (frontiers) between regions…
  - homogeneous zones: regions
    - same color, texture, intensity, ...

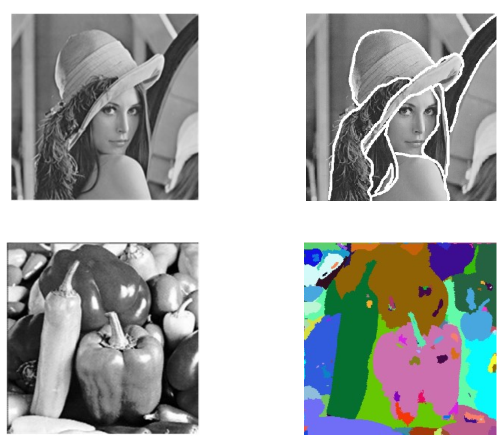
Approaches for image segmentation

- Pixel-based approach
- Region-based approach:
  - look for homogeneous areas in the image
- Edge-based approach :
  - look for discontinuities in the image
  - A closed edge is equivalent to a region
- Hybrid (Dual) approach (region + edge)



Examples

Original images Segmented images



Pixel-based approach

- Pixel-based approach
  - Thresholding
  - Clustering
- It is not a region segmentation technique
  - But we often in segmentation looking for regions
  - Need some post-processing

9

Thresholding

- Thesholding is a *simple and popular* method for object segmentation in digital images
- Thresholding can be
  - *Global:* one threshold for the whole image
  - *Local*: one threshold for a part of the image
  - *Adaptive:* one threshold ajusted according to each image or each image part

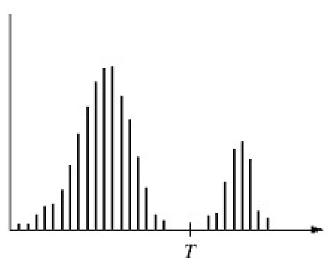
10

Basic global thresholding

- Basic thresholding (2 classes) – main idea :
  - IF value(pixel) >= threshold THEN value(pixel) = 1 (or 255)
  - IF value(pixel) < threshold THEN value(pixel) = 0
- The result is a binary image
- It is also possible to use *n thresholds* to split the image in n+1 classes
- Problem: choosing the threshold(s)!

Basic global thresholding

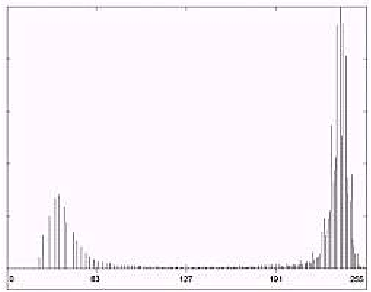
- Find the threshold on histogram of gray level intensity (histogram thresholding)

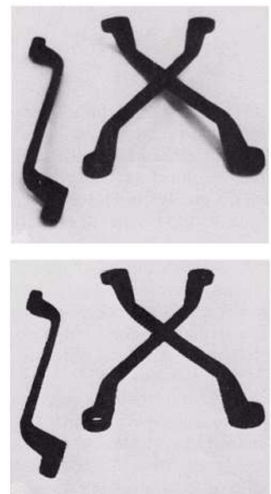


$$g(x,y) = \begin{cases} 0, & f(x,y) < T \\ 1, & f(x,y) \geq T \end{cases}$$

Basic global thresholding

- Threshold value: not difficile if
  - Controled environment
  - Industrial applications





13

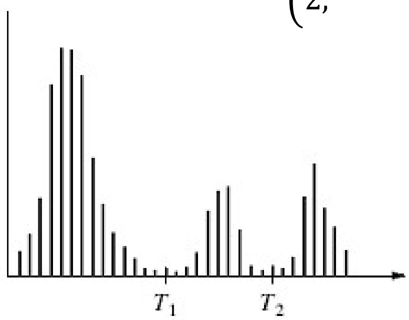
Multi-thresholds

- n thresholds to split the image in n+1 classes:
  - IF value(pixel) < threshold_1
    - THEN value(pixel) = 0
  - IF value(pixel) >= threshold_1 && value(pixel) < threshold_2
    - THEN value(pixel) = 1
  - …
  - IF value(pixel) >= threshold_n
    - THEN value(pixel) = n
- Problems: How many thresholds?

14

Multi-thresholds

$$g(x,y) = \begin{cases} 0, & f(x,y) < T1 \\ 1, & T2 > f(x,y) \geq T1 \\ 2, & f(x,y) \geq T2 \end{cases}$$

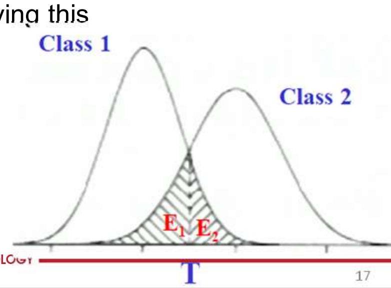


Threshold value

- Global thresholding: How to find the value of the threshold **T** ?
  - Value obtained by tests
  - The mean value of gray values
  - The median value between the min gray level and the max one
  - One value balancing both sections of the histogram
    - automatic thresholding

Choice of thresholds (optimal)

- 2 surfaces (background and object) in an image
  - We suppose mathematical models for distributions (gaussians, etc.)
  - We determine the probability of error in the classes 1 and 2 (surfaces 1 et 2)
  - We search for a threshold T resulting in a minimum error
    - Several methods for achieving this



Source : www.iro.umontreal.ca/~dift2730/

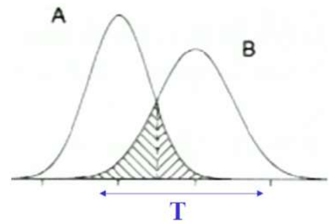
Example: Global automatic thresholding

- One possible algorithm:
  - Choose an initial value for the threshold T (mean, median, …)
  - We obtain 2 groups of pixels
    - G1 where f(x,y) >= T and G2 where f(x,y) < T
  - Compute the gray level means for G1 and G2 -> μ1 and μ2
  - Compute a new value for T
    - T = 1/2 (μ1 + μ2)
  - Repeat until T is ~ constant

- There exist many other global automatic methods
  - Otsu, Kittler, K-means, …
  - No solution on which one to use
  - Must be tested for each new application

Example: Otsu algorithm

- Sweep all possible threshold value for **T**
- For **each value of T**:
  - Compute the mean and variance for each class
  - We look for the intraclass variance
    - Means: μ_1, μ_2
    - Variances: σ_1^2, σ_2^2
    - Intra-class variance:
      $\sigma_w^2 = P_1 * \sigma_1^2 + P_2 * \sigma_2^2$
- The optimal threshold is the one with the minimum value for σ_w^2
- It is based on the idea that classes are well defined and well grouped



$$\sigma_1^2 = \frac{1}{T}\sum_{i=0}^{T-1}\left(h(i) - \mu_1\right)^2$$

$$\sigma_2^2 = \frac{1}{256 - T}\sum_{i=T}^{255}\left(h(i) - \mu_2\right)^2$$

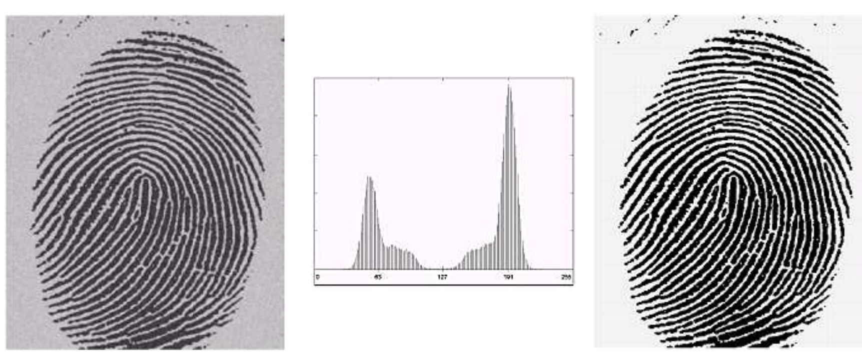
$$\mu_1 = \frac{1}{T}\sum_{i=0}^{T-1} h(i) \quad P_1 = \frac{1}{N \cdot M}\sum_{i=0}^{T-1} h(i)$$

$$\mu_2 = \frac{1}{256 - T}\sum_{i=T}^{255} h(i) \quad P_2 = \frac{1}{N \cdot M}\sum_{i=T}^{255} h(i)$$

Source : www.iro.umontreal.ca/~dift2730/

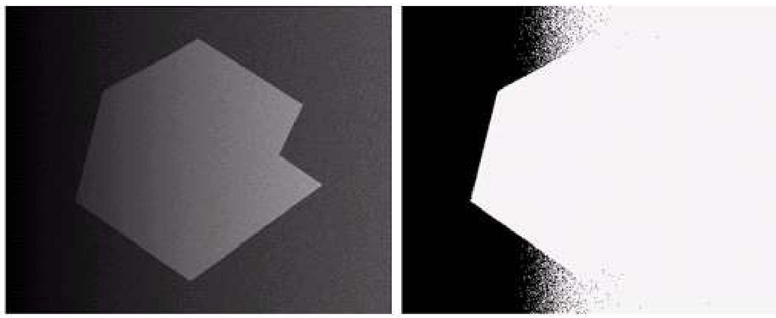
Example: Otsu algorithm

- Threshold found by the algorithm:
  - T = 125



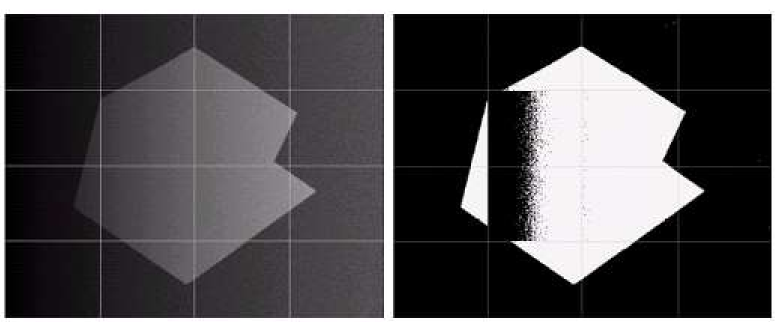
Global threshold: problem

- Problem:
  - Global thresholding cannot be used in that case
  - Solution: adaptive local thresholding

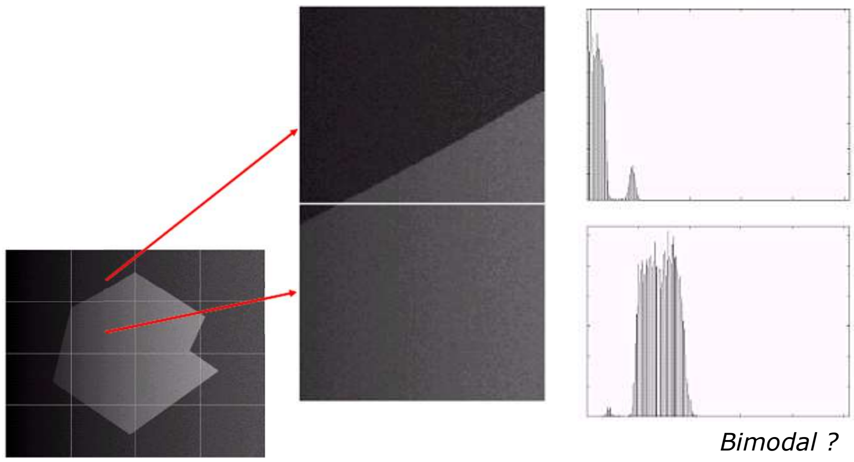


Example of adaptive thresholding

- Split the image in sub-images and process each sub-image with its own threshold
- The main decision is to choose the size of the sub-images
- Before processing each sub-image, check the variance to make sure that the sub-image contains two regions, and not only one.
  - Example: no thresholding for a sub-image if variance<100



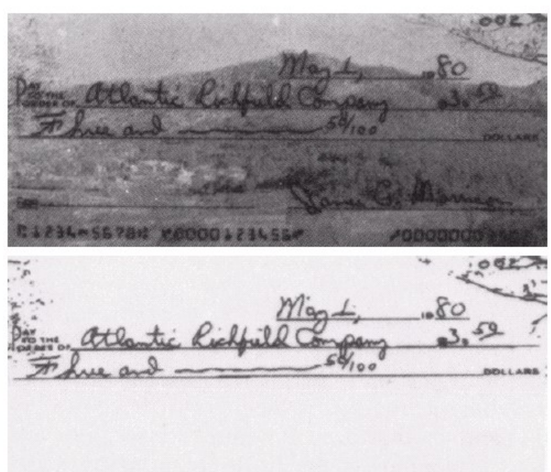
Example of adaptive thresholding



Example of adaptive thresholding

a
b

FIGURE 10.37
(a) Original image. (b) Image segmented by local thresholding. (Courtesy of IBM Corporation.)



Clustering-based segmentation

- Image is considered as a set of N image pixels.
- Attributes (property) of the pixels
  - gray level of single-band gray tone images,
  - color values of three-band color images: (r, g, b)
  - values of multi-band images, …
- Based on the similar attribute, pixels classification operators partition an image into homogeneous regions.
  - Clustering provides a grouping of the pixels that is dependent on their values in the image but not necessarily on their locations in the image unless location is a specified property
  - Classifier provide the pixel classes which should be homogeneous regions.

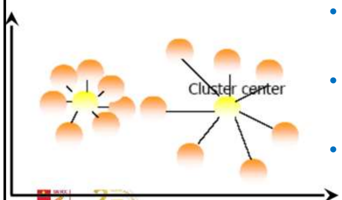
Clustering algorithms

- **Image segmentation approaches including**:
  - Feature space clustering approaches
  - Graph-based approaches

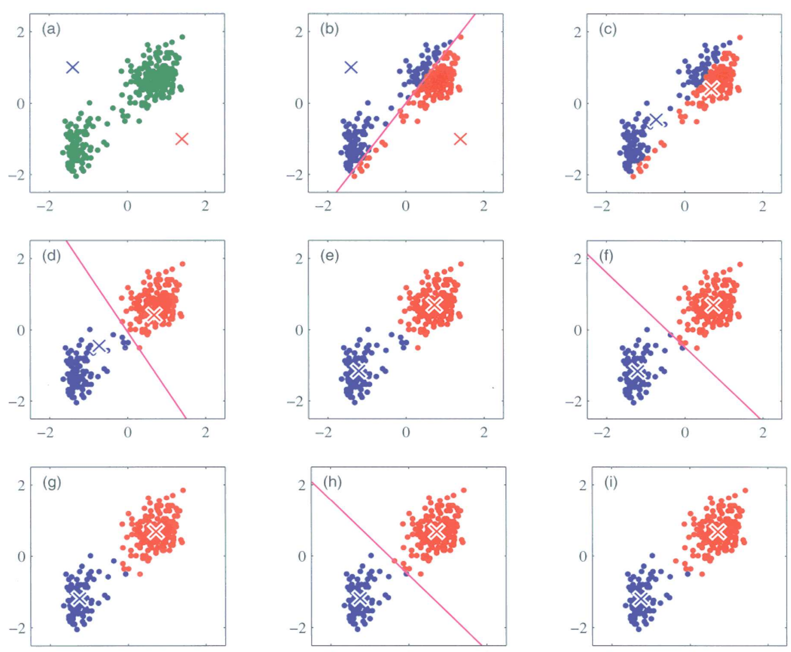
- **Clustering algorithms**:
  - K-Means clustering
  - Mean-Shift Clustering
  - Expectation-Maximization Clustering
  - Watershed Segmentation
  - Graph Cuts (Spectral clustering)
  - Normalized cuts
  - …

K-means Clustering

- Let $X = \{p_1, \ldots, p_N\}$ be a set of N image pixels:
  - $V(p_i)$: the property vector associated with pixel p_i
  - The clustering algorithm is to partition the image into K clusters (K regions)
- The **K-means** algorithm:
  - Initialization step: An initial property vector of each class C_k is chosen randomly from the set of all property vector, note $\mu_k(0)$
  - Interactive step: Assignment of image pixels to K clusters



- Pixel p_i is assigned to the closest cluster, using a distance between 2 property vectors.
- Update the property vector of each class: $\mu_k(t)$ is computed as the mean of $\{V(p_i) \mid x_i \in C_k\}$.
- 2 Steps above are repeated until algorithm convergence. each class C_k should be a region R_k



K-means Clustering

Input image

K-means on gray level

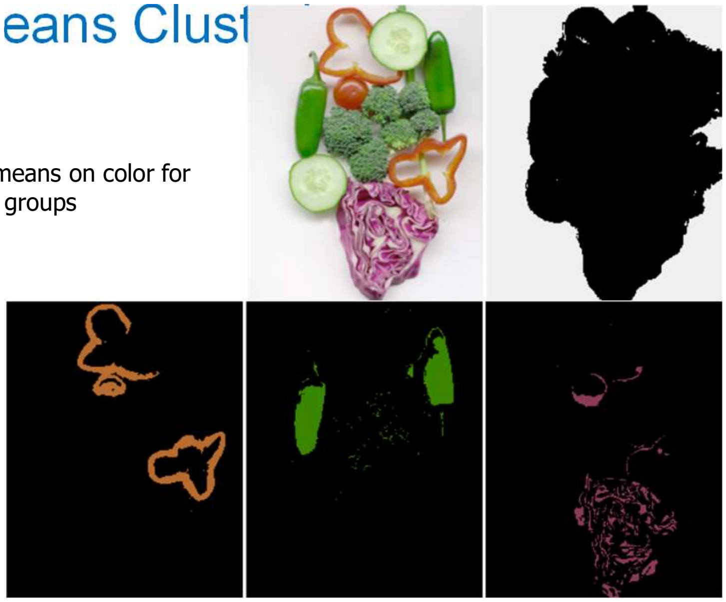
K-means on color

Source : D.A. Forsyth and J. Ponce. Computer Vision : A Modern Approach. Prentice-Hall, 2002.

29

K-means Clust

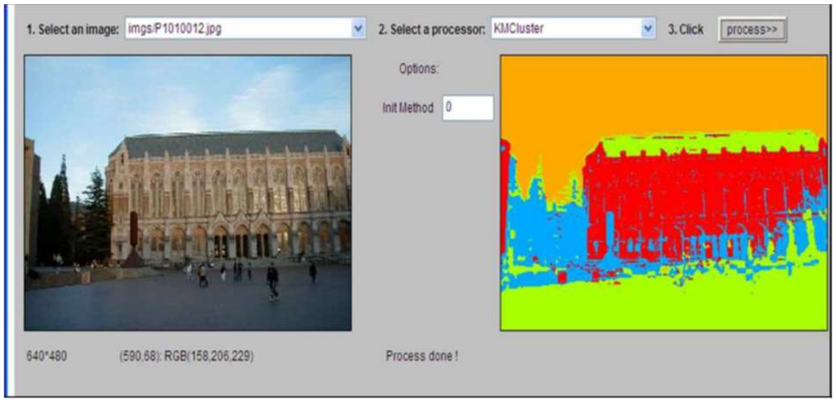
K-means on color for
11 groups



Source : D.A. Forsyth and J. Ponce. Computer Vision : A Modern Approach. Prentice-Hall, 2002.

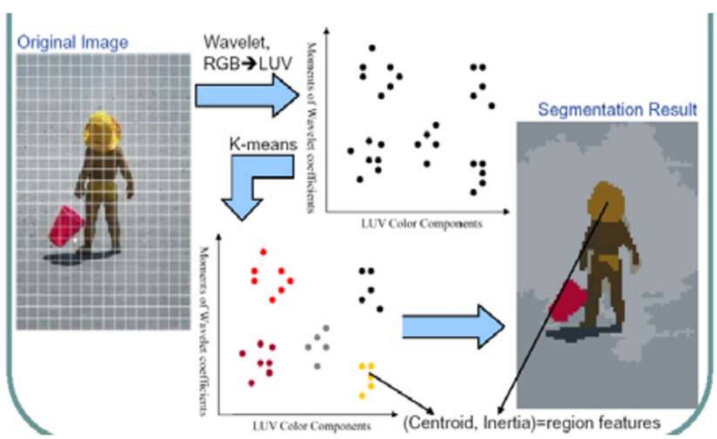
30

Example



31

Example



32

Pixel-based approach: Pros & cons

- Pros
  - Simple, fast
- Cons: thresholding is mainly an operation on pixels
  - It does not give connected regions ➔ can add more features
  - we need to « clean » the results
    - erase lonely pixels, keep regions
- Other segmentation methods exist
    - that can keep the integrity of regions (connected pixels)

Features for segmentation

- Intensity, Color?
- Position
- Texture
- …

Segmentation as clustering

Depending on what we choose as the *feature space*, we can group pixels in different ways.

Grouping pixels based on **intensity** similarity

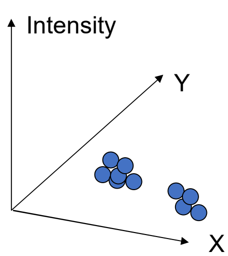
Feature space: intensity value (1-d)

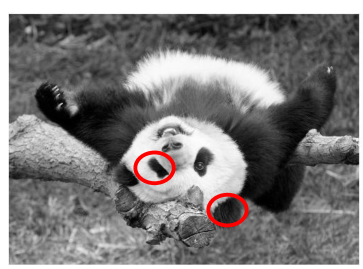
Slide credit: Kristen Grauman

Segmentation as clustering

Depending on what we choose as the *feature space*, we can group pixels in different ways.

Grouping pixels based on **intensity+position** similarity





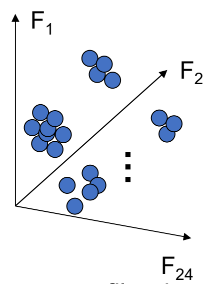
Both regions are black, but if we also include **position (x,y),** then we could group the two into distinct segments; way to encode both ***similarity & proximity.***

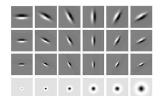
Slide credit: Kristen Grauman

Segmentation as clustering

Depending on what we choose as the *feature space*, we can group pixels in different ways.

Grouping pixels based on **texture** similarity





Filter bank of 24 filters

Feature space: filter bank responses (e.g., 24-d)

Slide credit: Kristen Grauman

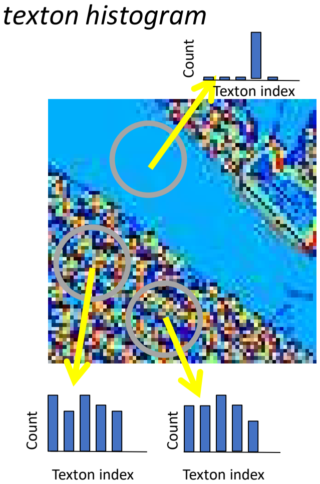
37

Segmentation with texture features

- Find “textons” by **clustering** vectors of filter bank outputs
- Describe texture in a window based on *texton histogram*

Image

Texton map

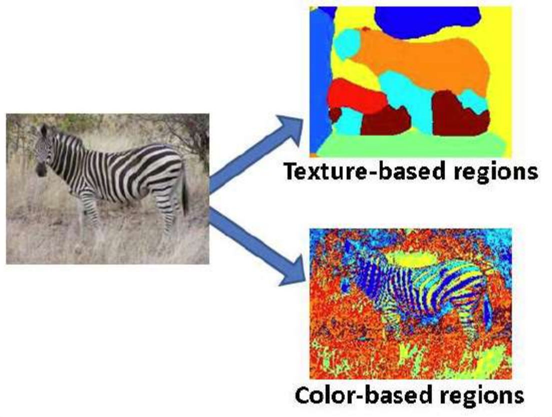


Malik, Belongie, Leung and Shi. IJCV 2001.

Adapted from Lana Lazebnik

38

Image segmentation example



Slide credit: Kristen Grauman

39

Region-based segmentation

- Finding region based on the criterion of homogeneity and connectivity of pixels (region)
  - Each region is homogeneous (i.e., uniform in terms of the pixel attributes such as intensity, color, range, or texture, etc.)
  - and connected
- **Algorithms**:
  - Region growing
  - Split and merge algorithm
  - Hierarchical clustering
  - …

40

Region-based segmentation

- Region-based approaches provide :
  - All pixels must be assigned to regions
  - Each pixel must belong to a single region only
  - Each region must be uniform
  - Any merged pair of adjacent regions must be non-uniform
  - Each region must be a connected set of pixels
- Region-based approaches:
  - Different methods
  - Common point: **homogeneity criteria**

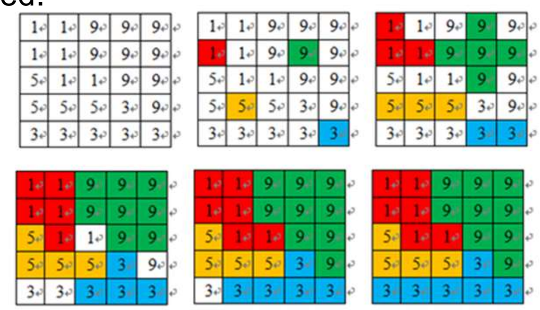
Region growing

- Start from a point (seed) and add neighbor pixels following a given criteria
- The seeds can be manually or automatically chosen
  - automatic seeds in very homogeneous zones for example

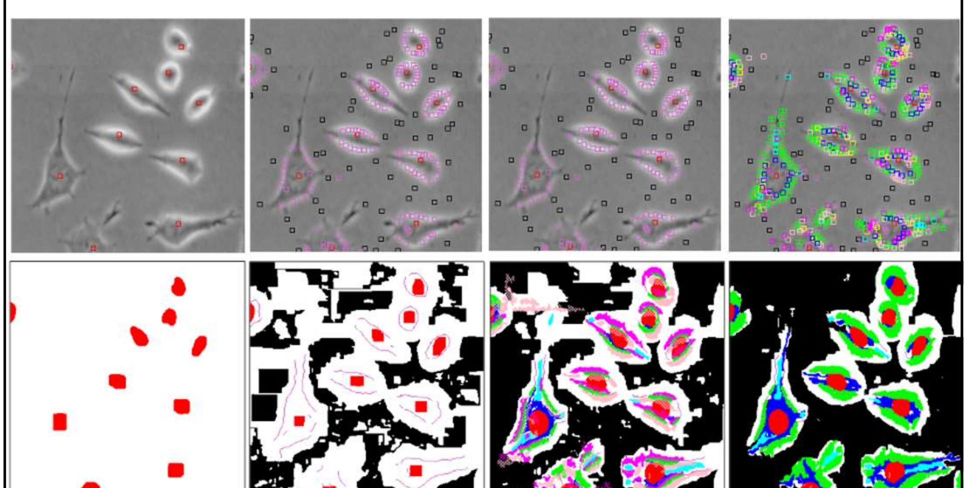
image *Seed* *Region growing* *Final result*

Region growing algorithm

- Algorithm:
  - Choose K random pixels in K regions
  - Use 8-connected and threshold to determine
  - Repeat a and b until almost points are K classified.
- Example illustrated:



Region growing with multi-seeds

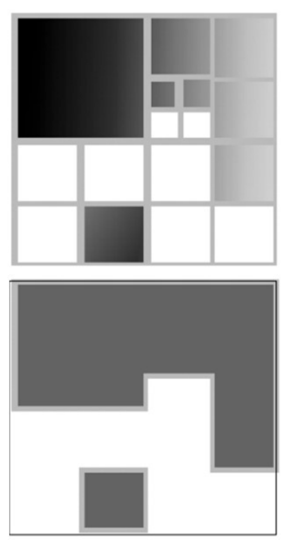


Example

Simulation of region growing (90% pixels)

Split-and-merge

- Split (step 1)
  - Recursively split all non-homogeneous regions following a given criteria
    - variance, max-min, …
  - Dividing one region gives 4 subregions
  - Subregion attributes are re-computed
- Merge (step 2)
  - Group all homogeneous adjacent regions following a given criteria



Split-and-merge: split

Image initiale

0	1	0	0	7	7	7	7
1	0	2	2	7	7	7	7
0	2	2	2	7	7	7	7
4	4	2	2	7	7	7	7
0	0	1	1	3	3	7	7
1	1	2	2	3	7	7	7
2	4	3	0	5	7	7	7
2	3	3	5	5	0	7	7

Split1

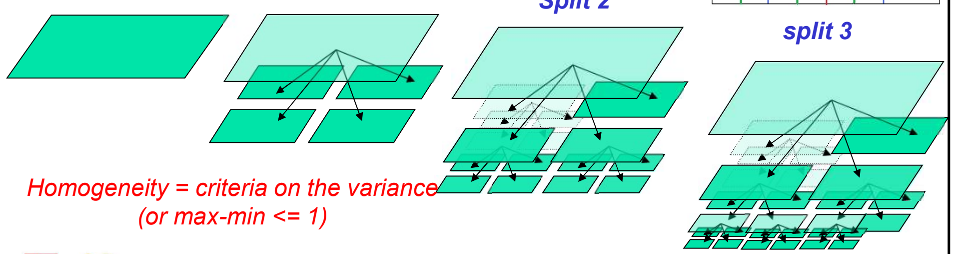
0	1	0	0	7	7	7	7
1	0	2	2	7	7	7	7
0	2	2	2	7	7	7	7
4	4	2	2	7	7	7	7
0	0	1	1	3	3	7	7
1	1	2	2	3	7	7	7
2	4	3	0	5	7	7	7
2	3	3	5	5	0	7	7

Split 2

0	1	0	0	7	7	7	7
1	0	2	2	7	7	7	7
0	2	2	2	7	7	7	7
4	4	2	2	7	7	7	7
0	0	1	1	3	3	7	7
1	1	2	2	3	7	7	7
2	4	3	0	5	7	7	7
2	3	3	5	5	0	7	7

split 3

0	1	0	0	7	7	7	7
1	0	2	2	7	7	7	7
0	2	2	2	7	7	7	7
4	4	2	2	7	7	7	7
0	0	1	1	3	3	7	7
1	1	2	2	3	7	7	7
2	4	3	0	5	7	7	7
2	3	3	5	5	0	7	7



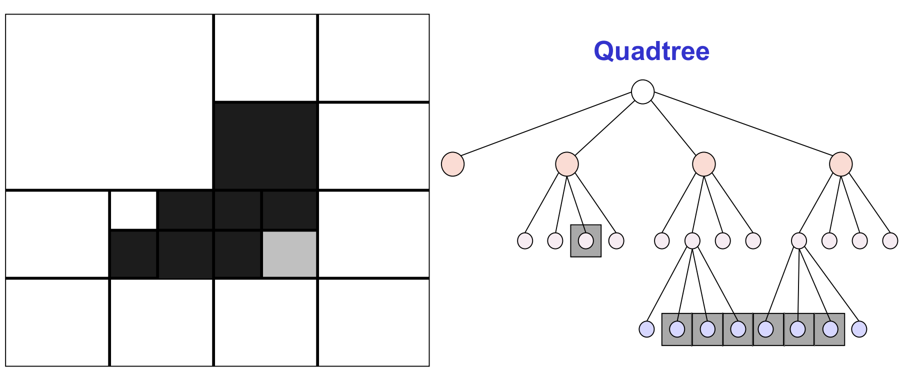
Homogeneity = criteria on the variance (or max-min <= 1)

Source : Jean-Christophe Baillie. Cours de segmentation. ENSTA ParisTech (France).

47

Split-and-merge: merge

Phase 1: Create homogeneous zones (split)
Phase 2: Group homogeneous zone (merge)

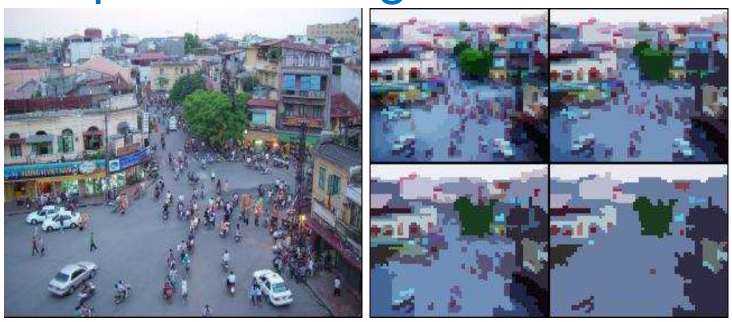


Connect homogeneous adjacent regions

Source : Jean-Christophe Baillie. Cours de segmentation. ENSTA ParisTech (France)

48

Split-and-merge



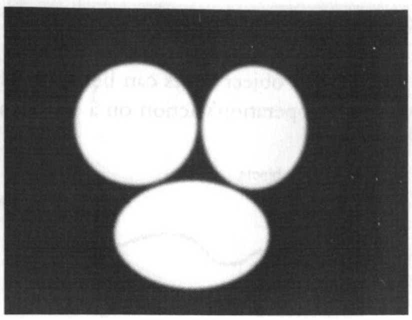
Edge-based segmentation

- Finding region based on edges

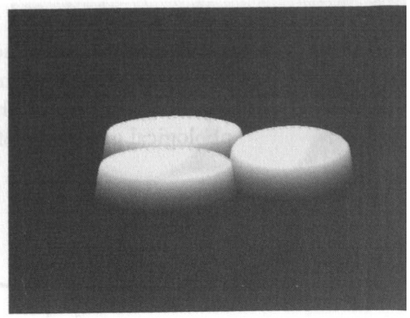
- **Algorithms**:
  – Basic Edge Detection
  – The Marr-Hildreth edge detector (LoG)
  – Short response Hilbert transform (SRHLT)
  – Watersheds

Watershed segmentation

- We consider the image as a 3D shape using the gray level as the third dimension



2D image

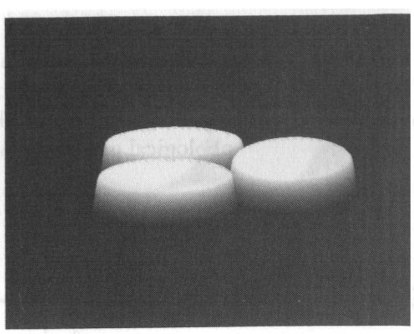


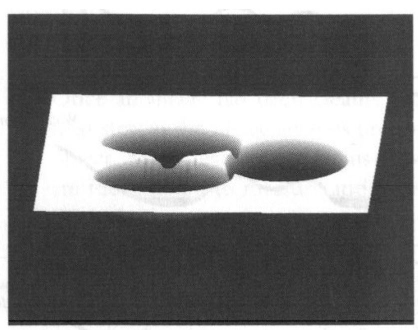
Visualization in 3D

Source : http://www.gpa.etsmtl.ca/cours/gpa669/

Watershed segmentation

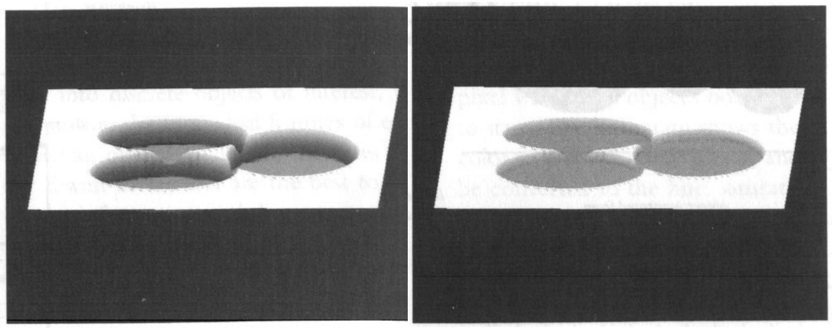
- After we reverse (upside down) the values to create « holes » in the shape





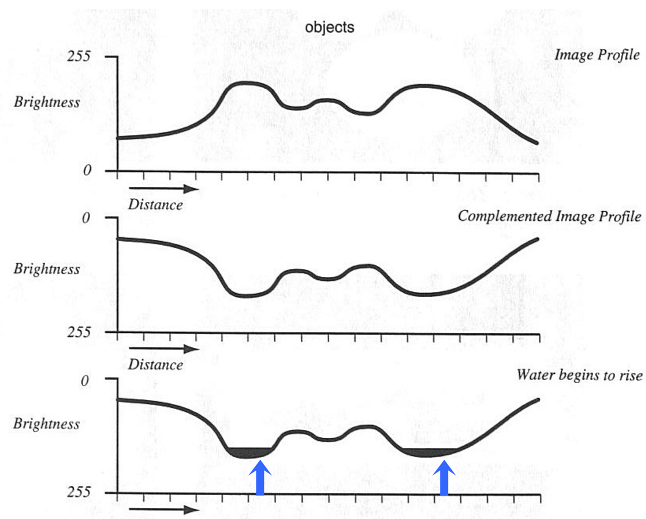
Watershed segmentation

- Next we fill in the holes with water



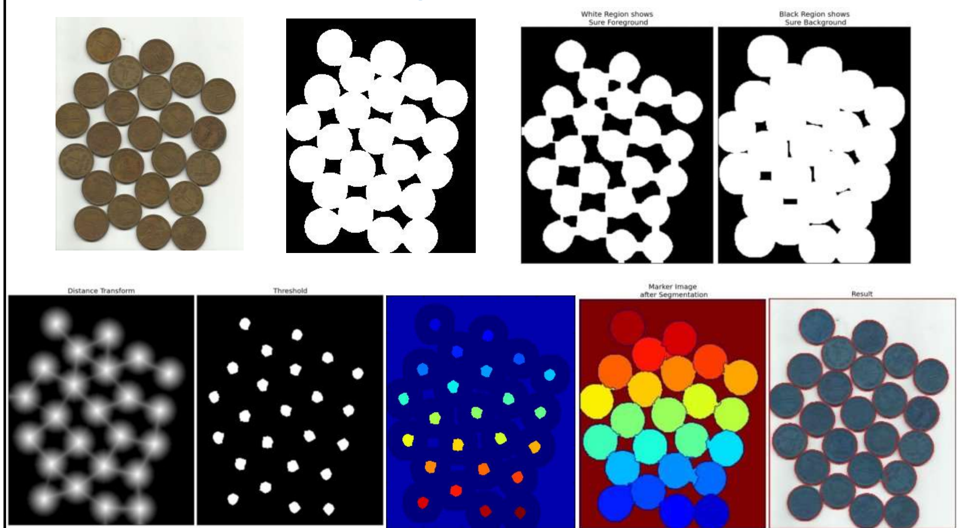
53

Watershed segmentation



54

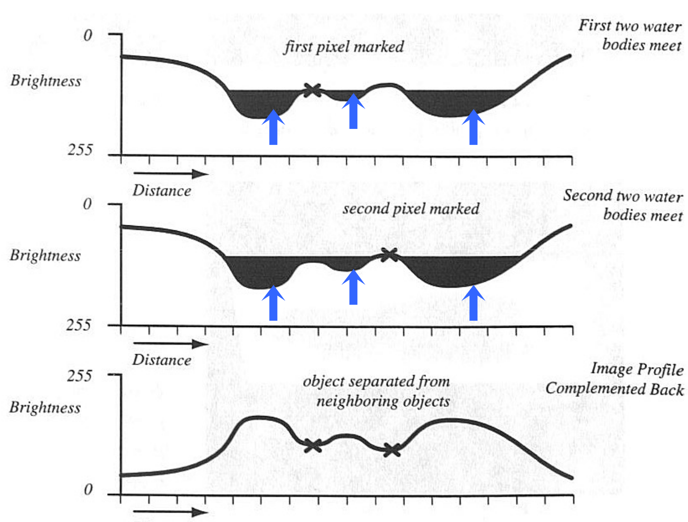
Watershed segmentation



https://docs.opencv.org/master/d3/db4/tutorial_py_watershed.html

55

Watershed segmentation



56

Segmentation – advices

- Image segmentation
  - No method works for all images
  - No miracle, no warranty!
- One of the main problem is to define the goal of segmentation:
  - What exactly are we looking for in the image?
    - Global regions or small details?
    - Presence or not of persons details in the face?
- It is good to think in advance what we will do with this segmentation results
  - This helps to define the level of precision needed

57

Segmentation – advices

- Image Pre-processing:
  - good selection of sensors and energy source, and controled image acquisition conditions help to make segmentation easier and more efficient
- For some applications, we realize today that we can avoid to segment the image. It is often better like this.

58

Limits of segmentation

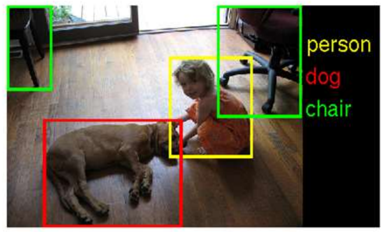
Image segmentation alone cannot find all image objects as we can interpret them

Object detection

Object Detection

- **Problem**: Detecting and localizing generic objects from various categories, such as cars, people, etc.

- Challenges:
  - Illumination,
  - viewpoint,
  - deformations,
  - Intra-class variability



Window-based generic object detection

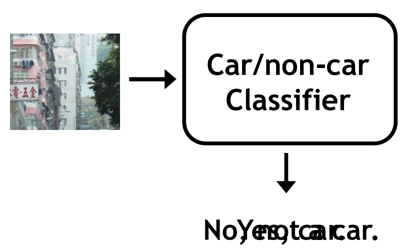
Basic pipeline

Generic category recognition: basic framework

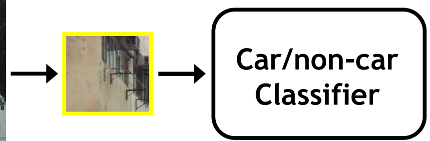
- Build/train object model
  - Choose a representation
  - Learn or fit parameters of model / classifier
- Generate candidates in new image
- Score the candidates

Window-based models
Building an object model

Given the representation, train a binary classifier



Window-based models
Generating and scoring candidates



Slide: Kristen Grauman

Window-based models
Generating and scoring candidates

- Slide through the image and check if there is an object at every location

YES!! Person match found

Window-based models
Generating and scoring candidates

- But what if we were looking for buses?

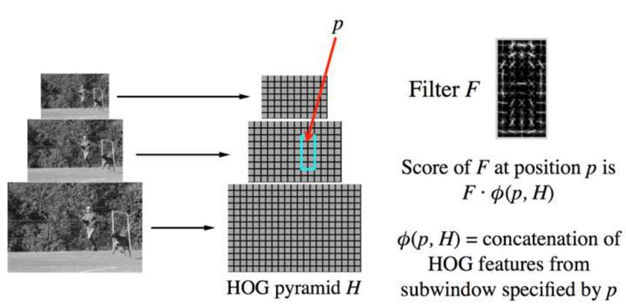
- We will never find the object if we don't choose our window size wisely!

67

Multi-scale sliding window

- Work with multiple size windows

- Create a feature pyramid

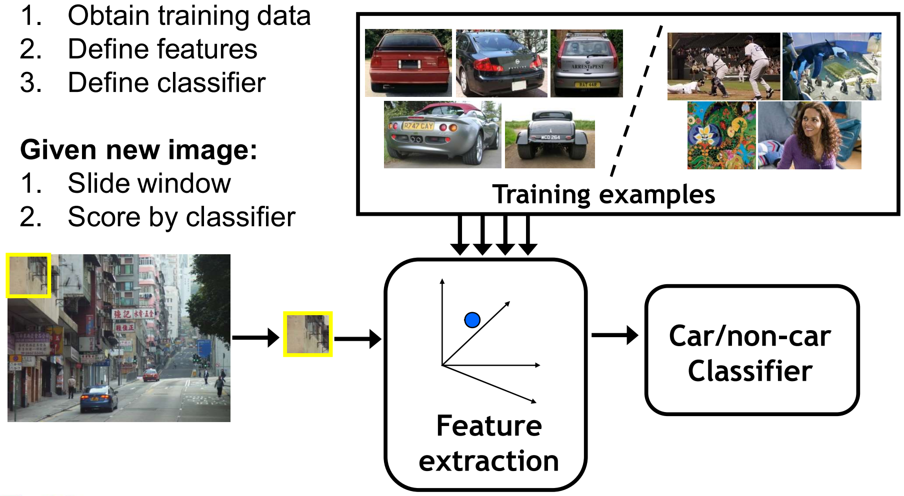


68

Window-based object detection: recap

Training:
1. Obtain training data
2. Define features
3. Define classifier

Given new image:
1. Slide window
2. Score by classifier



SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Slide: Kristen Grauman

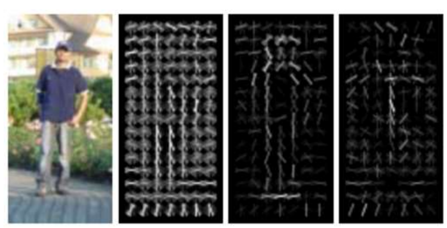
69

69

Features

70

- HOG



- Bags of visual words

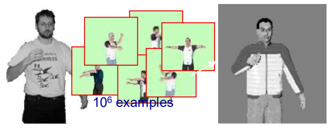
- Haar features, …

SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

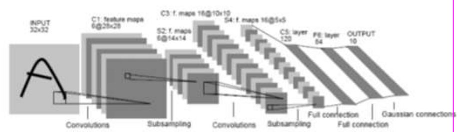
70

Discriminative classifier construction

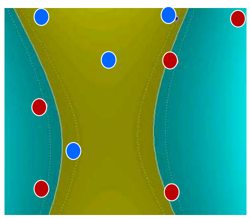
Nearest neighbor



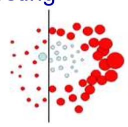
Neural networks



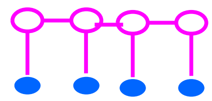
Support Vector Machines



Boosting



Conditional Random Fields

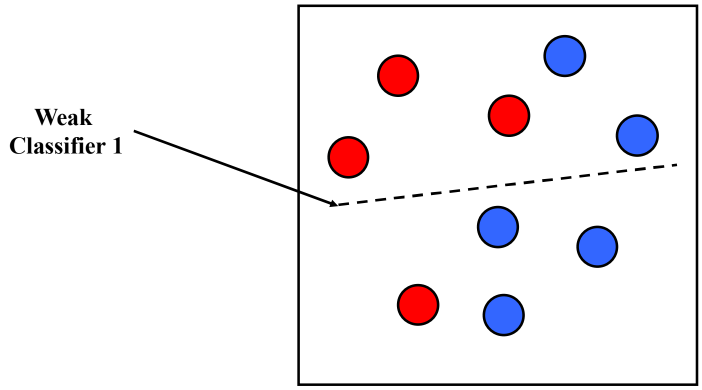


71

Boosting classifiers

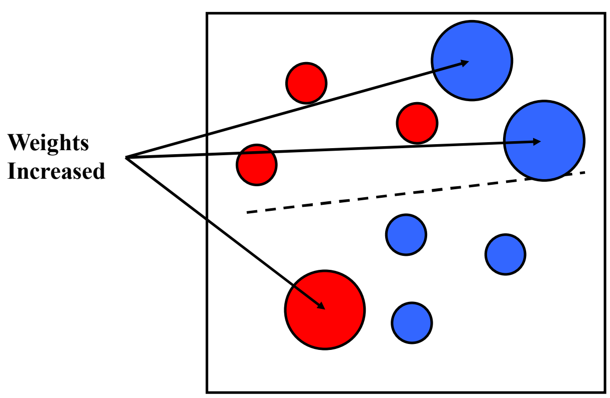
72

Boosting intuition



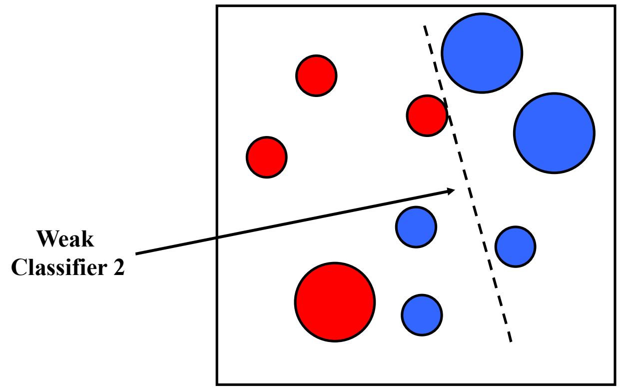
73

Boosting illustration

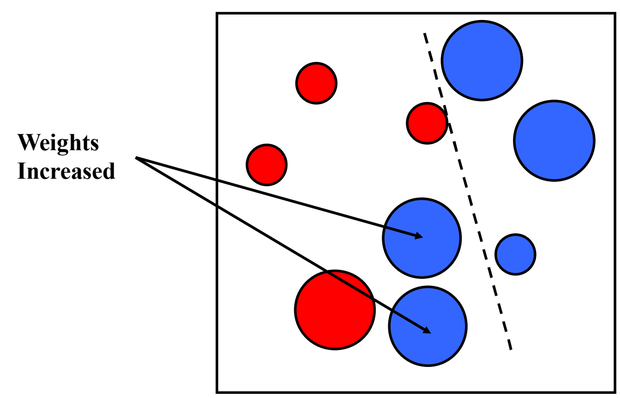


74

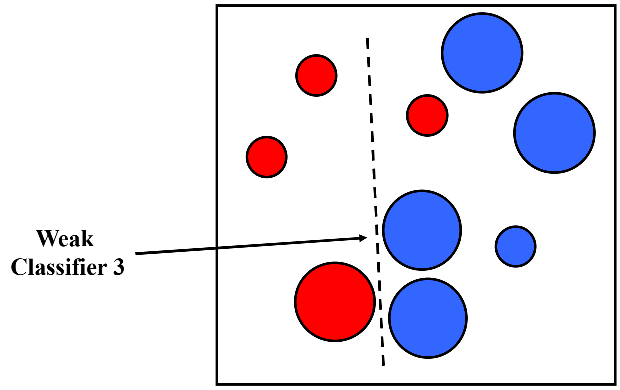
Boosting illustration



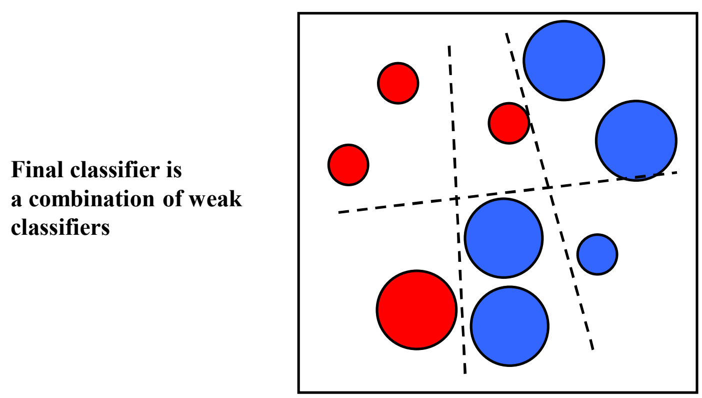
Boosting illustration



Boosting illustration



Boosting illustration



Boosting: training

- Initially, weight each training example equally
- In each boosting round:
  - Find the **weak learner** that achieves the lowest *weighted* training error
  - **Raise weights of training examples misclassified** by current weak learner
- Compute final classifier as linear combination of all weak learners
  - (weight of each learner is directly proportional to its accuracy)
- Exact formulas for re-weighting and combining weak learners **depend on the particular boosting scheme** (e.g., *AdaBoost*)

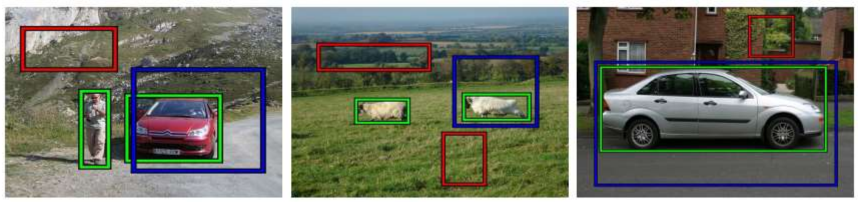
Slide credit: Lana Lazebnik

Object proposals

Object proposals

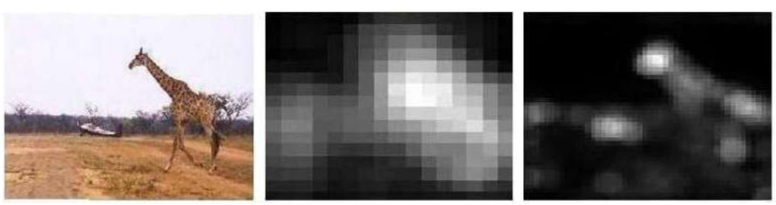
Main idea:

- Learn to generate category-independent regions/boxes that have **object-like** properties.
- Let object detector **search over "proposals**", not exhaustive sliding windows



Alexe et al. Measuring the objectness of image windows, PAMI 2012

Object proposals



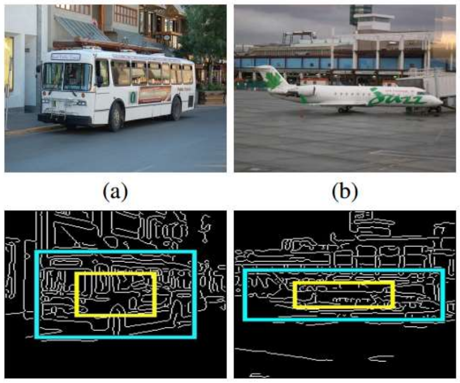
Multi-scale saliency

Color contrast

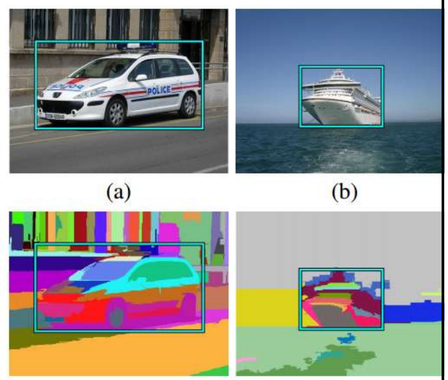
Alexe et al. Measuring the objectness of image windows, PAMI 2012

Object proposals

Edge density



Superpipxel straddling



Alexe et al. Measuring the objectness of image windows, PAMI 2012

83

Object proposals

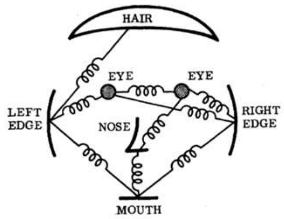
Yellow box: object detected
Cyan box: groundtruth

Alexe et al. Measuring the objectness of image windows, PAMI 2012

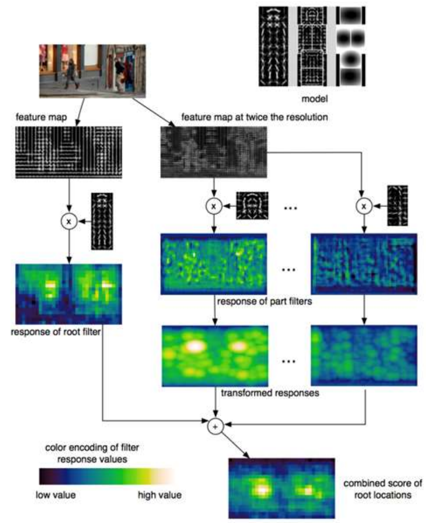
84

Deformable Part Model (DPM)

- Represents an object as a **collection of parts** arranged in a deformable configuration
- Each part represents **local appearances**
- Spring-like connections between certain pairs of parts



Fischler and Elschlager, Pictoral Structures, 1973

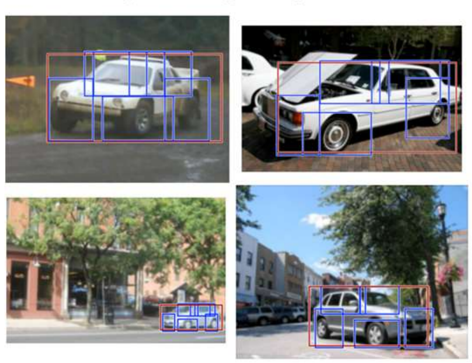


Felzenszwalb et al. , PAMI 2010

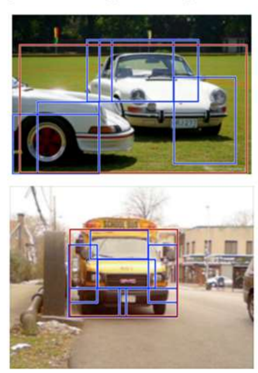
86

Deformable Part Model (DPM)

high scoring true positives



high scoring false positives



87

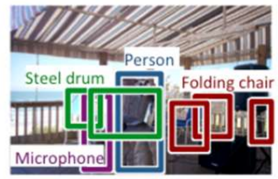
Deformable Part Model (DPM)

- References
  - Pedro F. Felzenszwalb & Daniel P. Huttenlocher, Pictorial Structures for Object Recognition, IJCV 2005
    - https://www.cs.cornell.edu/~dph/papers/pict-struct-ijcv.pdf
  - P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(9):1627–1645, 2010

Object detection: Evaluation

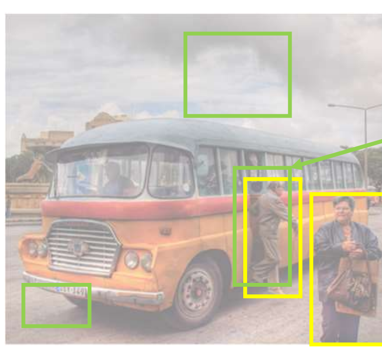
Object Detection Benchmarks

- PASCAL VOC Challenge
- ImageNet Large Scale Visual Recognition Challenge (ILSVR)
  - 200 Categories for detection



- Common Objects in Context (COCO)
  - 80 Object categories

How do we evaluate object detection?



predictions
ground truth

True positive:
- The overlap of the prediction with the ground truth is MORE than a threshold value (0.5)

How do we evaluate object detection?

predictions

ground truth

True positive:
False positive:
- The overlap of the prediction with the ground truth is LESS than a threshold value (0.5)

92

How do we evaluate object detection?

predictions

ground truth

True positive:
False positive:
False negative:
- The objects that our model doesn't find

93

How do we evaluate object detection?

predictions

ground truth

True positive:
False positive:
False negative:
- The objects that our model doesn't find

What is a **True Negative**?

	Predicted 1	Predicted 0
True 1	true positive	false negative
True 0	false positive	true negative

	Predicted 1	Predicted 0
True 1	TP	FN
True 0	FP	TN

	Predicted 1	Predicted 0
True 1	hits	misses
True 0	false alarms	correct rejections

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

How do we evaluate object detection?

predictions

ground truth

True positive: 1
False positive: 2
False negative: 1

So what is the
- precision?
- recall?

Precision versus recall

- Precision:
  - how many of the object detections are correct?

$$precision = \frac{TP}{TP + FP}$$

- Recall:
  - how many of the ground truth objects can the model detect?
  - True Positive Rate (TPR)

$$recall = \frac{TP}{TP + FN}$$

- In reality, our model makes a lot of predictions with varying scores between 0 and 1

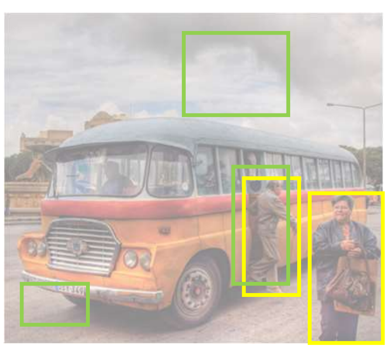
predictions
ground truth

Here are all the boxes that are predicted with score > 0.

This means that our
- **Recall** is perfect!
- But our precision is BAD!

98

How do we evaluate object detection?



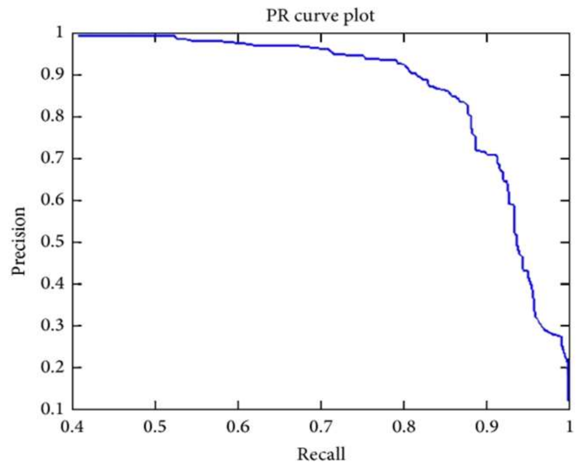
predictions
ground truth

Here are all the boxes that are predicted with score > 0.5

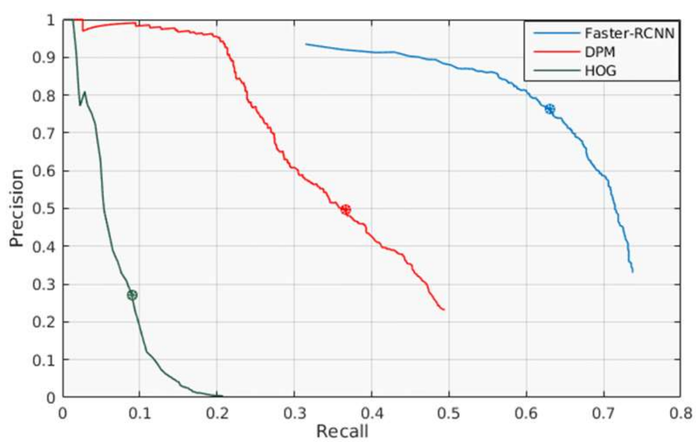
We are setting a threshold of 0.5

99

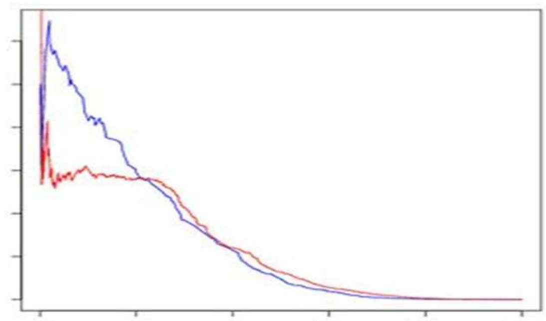
Precision – recall curve (PR curve)



Which model is the best?



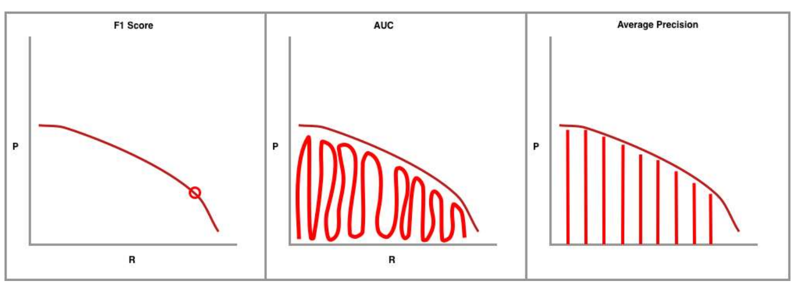
Which model is the best?



- **Area under curve (AUC), average precision (AP)**
- **F1-score** (highest value at optimal confidential score)

102

Which model is the best?



AP: The metric calculates the average precision (AP) for each class individually across all of the IoU thresholds

$$AP = \frac{1}{11} \sum_{r \in \{0, 0.1, 0.., 0.9, 1\}} p_{inter\_p}(r)$$

mAP: the average of AP

$$= \frac{1}{11}(1+1+1+1+0.67+0.67+0.67+0.5+0.5+0.5+0.5)$$

$$\approx 0.728$$

103

Summary

- Object recognition as classification task
  - Boosting (face detection ex)
  - Support vector machines and HOG (human detection ex)
  - Sliding window search paradigm
    - Pros and cons
    - Speed up with attentional cascade
    - Object proposals, proposal regions as alternative

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Thank you for your attention!

soict.hust.edu.vn/ fb.com/groups/soict