Day 2
Reconocimiento de objetos I
2.1 Categorías
• What is an object?
• What is a category?
The object
The texture

The object
The texture

The object

The scene
An example of categorical perception

- Continuous perception: graded response

- Categorical perception: “sharp” boundaries

Many perceptual phenomena are a mixture of the two: categorical at an everyday level of magnification, but continuous at a more microscopic level. It can also depend on cultural aspects, expertise, task, attention, …
Another example

• Continuous perception: graded response

20-24
25-29
30-34
35-39
40-44
45-49
50-54

• Categorical perception: “sharp” boundaries

Emotions have categorical boundaries
Why do we care about categories?

Perception of function: We can perceive the 3D shape, texture, material properties, without knowing about objects. But, the concept of category encapsulates also information about what can we do with those objects.

“We therefore include the perception of function as a proper –indeed, crucial- subject for vision science”, from Vision Science, chapter 9, Palmer.
Why do we care about categories?

When we recognize an object we can make predictions about its behavior in the future, beyond of what is immediately perceived.
The perception of function

- Direct perception (affordances): Gibson

- Mediated perception (Categorization)

One caveat of this comparison: deciding that something is a chair might require access to more features than the ones needed to decide that we can sit on something… (it is a different level of categorization)
Direct perception

Some aspects of an object function can be perceived directly

- Functional form: Some forms clearly indicate to a function (“sittable-upon”, container, cutting device, …)

It does not seem easy to sit-upon this…
Limitations of Direct Perception

Some aspects of an object function can be perceived directly

• Observer relativity: Function is observer dependent
Limitations of Direct Perception

Objects of similar structure might have very different functions.

Not all functions seem to be available from direct visual information only.

The functions are the same at some level of description: we can put things inside in both and somebody will come later to empty them. However, we are not expected to put inside the same kinds of things...
So, what do we use direct or indirect?

“It seems exceedingly unlikely (though logically possible) that we categorize everything in our visual fields”, Palmer.

**Hypothesis**: we categorize the objects that are relevant for a specific task that we have at hand, but we only extract affordances from the others.
How many categories?
“Muchas”
How many object categories are there?

~10,000 to 30,000

Biederman 1987
How many categories?

• Probably this question is not even specific enough to have an answer
Which level of categorization is the right one?

Car is an object composed of:
  a few doors, four wheels (not all visible at all times), a roof, front lights, windshield

If you are thinking in buying a car, you might want to be a bit more specific about your categorization.
Categorical hierarchies

Categories can be organized in hierarchies (tree structures are commonly used)

From Wordnet

This is a mapping of the Wordnet tree into the 2D plane
Organizing "things" into categories

(1) Feature-based

- Definition: disassembling a concept into a set of featural components

- Each feature is an essential element of the category: "for a thing to be an X, it must have that feature. Otherwise it is not an X"

(2) prototype

- Categories are formed on the basis of characteristics features, which describe the typical model of the category

- Characteristics features are commonly present in exemplars of concepts, but they are not always present."
Prototypical Theory

• According to the prototype view, an object will be classified as an instance of a category if it is sufficiently similar to the prototype.

• Similarity ~ the number of features shared between an object and the prototype (however, some features should be weighted more heavily as being more central to the prototype than are other features).
Prototype or Sum of Exemplars?

- **Prototype Model**
  - Category judgments are made by comparing a new exemplar to the prototype.

- **Exemplars Model**
  - Category judgments are made by comparing a new exemplar to all the old exemplars of a category or to the exemplar that is the most appropriate.

*Figure 7.3.* Schematic of the prototype model. Although many exemplars are seen, only the prototype is stored. The prototype is updated continually to incorporate more experience with new exemplars.

*Figure 7.4.* Schematic of the exemplar model. As each exemplar is seen, it is encoded into memory. A prototype is abstracted only when it is needed, for example, when a new exemplar must be categorized.
Levels of Categorization

The idea of prototypes and typicality led to the study of levels of categorization.

- Rosch et al: “albeit concepts exist at many different levels of a hierarchy, one level is fundamental: basic level.”

- Basic level: the best compromise between grouping together similar objects, and distinguish among objects from the same category.

- Willingham (247)
Levels of Categorization

SUPERORDINATE LEVEL CATEGORIES

BASIC-LEVEL CATEGORIES

SUBORDINATE LEVEL CATEGORIES

<table>
<thead>
<tr>
<th>Table 7.3. Examples of Nested Category Structures</th>
</tr>
</thead>
<tbody>
<tr>
<td>Superordinate Level</td>
</tr>
<tr>
<td>----------------------</td>
</tr>
<tr>
<td>Musical instrument</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Fruit</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
Rosch’s Levels of Categorization

Definition of Basic Level:

• **Similar shape**: Basic level categories are the highest-level category for which their members have similar shapes.

• **Similar motor interactions**: … for which people interact with its members using similar motor sequences.

• **Common attributes**: … there are a significant number of attributes in common between pairs of members.

![Graph showing similarity declines from subordinate to basic level, then drops dramatically.](graph.png)
Levels of Categorization

• Rosch et al (1976) found that when people are shown pictures of objects, they identify objects at a **basic level more quickly** than they identified objects at higher or lower levels.

• Objects appear to be **recognized first at their basic level**, and only afterwards they are classified in terms of higher or lower level categories.
Entry-level categories
(Jolicoeur, Gluck, Kosslyn 1984)

• Typical member of a basic-level category are categorized at the expected level
• Atypical members tend to be classified at a subordinate level.

A bird
An ostrich
We do not need to recognize the exact category

A new class can borrow information from similar categories
2.2 Un poco de historia
Object recognition
Is it really so hard?

Find the chair in this image

Output of normalized correlation

This is a chair
Object recognition
Is it really so hard?

Find the chair in this image

Pretty much garbage
Simple template matching is not going to make it
Object recognition
Is it really so hard?

Find the chair in this image

A “popular method is that of template matching, by point to point correlation of a model pattern with the image pattern. These techniques are inadequate for three-dimensional scene analysis for many reasons, such as occlusion, changes in viewing angle, and articulation of parts.” Nivatia & Binford, 1977.
So, let’s make the problem simpler: Block world

Fig. 1. A system for recognizing 3-d polyhedral scenes. a) L.G. Roberts. b) A blocks world scene. c) Detected edges using a 2x2 gradient operator. d) A 3-d polyhedral description of the scene, formed automatically from the single image. e) The 3-d scene displayed with a viewpoint different from the original image to demonstrate its accuracy and completeness. (b) - e) are taken from [64] with permission MIT Press.

Nice framework to develop fancy math, but too far from reality…

Fig. 3. The representation of objects by assemblies of generalized cylinders. a) Thomas Binford. b) A range image of a doll. c) The resulting set of generalized cylinders. (b) and (c) are taken from Agin [1] with permission.)
Binford and generalized cylinders

(a) Cross section.

(b) Sweeping rule.

(c) True cylinder

(d) Generalized cylinder
Recognition by components

Irving Biederman
Recognition by components

The fundamental assumption of the proposed theory, recognition-by-components (RBC), is that a modest set of generalized-cone components, called geons \((N = 36)\), can be derived from contrasts of five readily detectable properties of edges in a two-dimensional image: curvature, collinearity, symmetry, parallelism, and cotermination.

The “contribution lies in its proposal for a particular vocabulary of components derived from perceptual mechanisms and its account of how an arrangement of these components can access a representation of an object in memory.”
1) We know that this object is nothing we know
2) We can split this objects into parts that everybody will agree
3) We can see how it resembles something familiar: “a hot dog cart”

“The naive realism that emerges in descriptions of nonsense objects may be reflecting the workings of a representational system by which objects are identified.”
Hypothesis

• Hypothesis: there is a small number of geometric components that constitute the primitive elements of the object recognition system (like letters to form words).

• “The particular properties of edges that are postulated to be relevant to the generation of the volumetric primitives have the desirable properties that they are invariant over changes in orientation and can be determined from just a few points on each edge.”

• Limitation: “The modeling has been limited to concrete entities with specified boundaries.” (count nouns) – this limitation is shared by many modern object detection algorithms.
Constraints on possible models of recognition

1) Access to the mental representation of an object should not be dependent on absolute judgments of quantitative detail.

2) The information that is the basis of recognition should be relatively invariant with respect to orientation and modest degradation.

3) Partial matches should be computable. A theory of object interpretation should have some principled means for computing a match for occluded, partial, or new exemplars of a given category.
"Parsing is performed, primarily at concave regions, simultaneously with a detection of nonaccidental properties."
Non accidental properties

Certain properties of edges in a two-dimensional image are taken by the visual system as strong evidence that the edges in the three-dimensional world contain those same properties.

Non accidental properties, (Witkin & Tenenbaum, 1983): Rarely be produced by accidental alignments of viewpoint and object features and consequently are generally unaffected by slight variations in viewpoint.

e.g., Freeman, “the generic viewpoint assumption”, 1994
Examples:

- Colinearity
- Smoothness
- Symmetry
- Parallelism
- Cotermination

**Figure 4.** Five nonaccidental relations. (From Figure 5.2, *Perceptual organization and visual recognition* [p. 77] by David Lowe. Unpublished doctoral dissertation, Stanford University. Adapted by permission.)
The high speed and accuracy of determining a given nonaccidental relation (e.g., whether some pattern is symmetrical) should be contrasted with performance in making absolute quantitative judgments of variations in a single physical attribute, such as length of a segment or degree of tilt or curvature.

Object recognition is performed by humans in around 100ms.
"If contours are deleted at a vertex they can be restored, as long as there is no accidental filling-in. The greater disruption from vertex deletion is expected on the basis of their importance as diagnostic image features for the components."
From varied variation over only two or three levels in the nonaccidental relations of four attributes of generalized cylinders, a set of 36 GEONS can be generated.

Geons represent a restricted form of generalized cylinders.
More GEONS

<table>
<thead>
<tr>
<th>Geon</th>
<th>Edge</th>
<th>Symmetry</th>
<th>Size</th>
<th>Axis</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><img src="image1.png" alt="Image" /></td>
<td>S</td>
<td>++</td>
<td>++</td>
<td>+</td>
</tr>
<tr>
<td><img src="image2.png" alt="Image" /></td>
<td>C</td>
<td>++</td>
<td>++</td>
<td>+</td>
</tr>
<tr>
<td><img src="image3.png" alt="Image" /></td>
<td>S</td>
<td>+</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td><img src="image4.png" alt="Image" /></td>
<td>S</td>
<td>++</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td><img src="image5.png" alt="Image" /></td>
<td>C</td>
<td>++</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td><img src="image6.png" alt="Image" /></td>
<td>S</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>

Figure 7. Proposed partial set of volumetric primitives (geons) derived from differences in nonaccidental properties.

<table>
<thead>
<tr>
<th>Geon</th>
<th>Edge</th>
<th>Symmetry</th>
<th>Size</th>
<th>Axis</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><img src="image1.png" alt="Image" /></td>
<td>S</td>
<td>+</td>
<td>++</td>
<td>-</td>
</tr>
<tr>
<td><img src="image2.png" alt="Image" /></td>
<td>C</td>
<td>+</td>
<td>++</td>
<td>-</td>
</tr>
<tr>
<td><img src="image3.png" alt="Image" /></td>
<td>S</td>
<td>++</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td><img src="image4.png" alt="Image" /></td>
<td>C</td>
<td>++</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td><img src="image5.png" alt="Image" /></td>
<td>S</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td><img src="image6.png" alt="Image" /></td>
<td>C</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Figure 9. Geons with curved axis and straight or curved cross sections. (Determining the shape of the cross section, particularly if straight, might require attention.)
Objects and their geons

Geons

Objects
Scenes and geons

Mezzanotte & Biederman
The importance of spatial arrangement

*Figure 3.* Different arrangements of the same components can produce different objects.
Supercuadrics

1. Block
2. Tapered block
3. Pyramid
4. Bent Block
5. Cylinder
6. Tapered Cylinder
7. Cone
8. Barrel
9. Ellipsoid
10. Bent Cylinder

Introduced in computer vision by A. Pentland, 1986.
What is missing?

The notion of geometric structure.

Although they were aware of it, the previous works put more emphasis on defining the primitive elements than modeling their geometric relationships.
Parts and Structure approaches

With a different perspective, these models focused more on the geometry than on defining the constituent elements:

- Fischler & Elschlager 1973
- Yuille ‘91
- Brunelli & Poggio ‘93
- Lades, v.d. Malsburg et al. ‘93
- Cootes, Lanitis, Taylor et al. ‘95
- Amit & Geman ‘95, ‘99
- Felzenszwalb & Huttenlocher ’00, ’04
- Crandall & Huttenlocher ’05, ’06
- Leibe & Schiele ’03, ’04
- Many papers since 2000
Representation

• Object as set of parts
  – Generative representation

• Model:
  – Relative locations between parts
  – Appearance of part

• Issues:
  – How to model location
  – How to represent appearance
  – Sparse or dense (pixels or regions)
  – How to handle occlusion/clutter

We will discuss these models more in depth later
But, despite promising initial results…things did not work out so well (lack of data, processing power, lack of reliable methods for low-level and mid-level vision)

Instead, a different way of thinking about object detection started making some progress: learning based approaches and classifiers, which ignored low and mid-level vision.

Maybe the time is here to come back to some of the earlier models, more grounded in intuitions about visual perception.
Face detection and the success of learning based approaches

- Graded Learning for Object Detection - Fleuret, Geman (1999)
- Robust Real-time Object Detection - Viola, Jones (2001)
- …
2.3 Un detector simple
A simple object detector

- Simple but contains some of same basic elements of many state of the art detectors.
- Based on boosting which makes all the stages of the training and testing easy to understand.
Discriminative vs. generative

- Generative model
  *(The artist)*

- Discriminative model
  *(The lousy painter)*

- Classification function

\[ p(Data, Zebra) \]
\[ p(Data, No Zebra) \]
\[ p(Zebra|Data) \]
\[ p(No Zebra|Data) \]
\[ label = F_{Zebra}(Data) \]
Discriminative methods

Object detection and recognition is formulated as a classification problem. The image is partitioned into a set of overlapping windows … and a decision is taken at each window about if it contains a target object or not.

Where are the screens?

Bag of image patches

In some feature space
Discriminative methods

Nearest neighbor
- 10^6 examples
- Shakhnarovich, Viola, Darrell 2003
- Berg, Berg, Malik 2005
- ...

Neural networks
- LeCun, Bottou, Bengio, Haffner 1998
- Rowley, Baluja, Kanade 1998
- ...

Support Vector Machines and Kernels
- Guyon, Vapnik
- Heisele, Serre, Poggio, 2001
- ...

Conditional Random Fields
- McCallum, Freitag, Pereira 2000
- Kumar, Hebert 2003
- ...
Formulation

• Formulation: binary classification

\[ x = x_1 \ x_2 \ x_3 \ \ldots \ x_N \ \ldots \ x_{N+1} \ x_{N+2} \ \ldots \ x_{N+M} \]

\[ y = -1 \ \ +1 \ -1 \ -1 \ ? \ ? \ ? \]

Training data: each image patch is labeled as containing the object or background

Test data

• Classification function

\[ \hat{y} = F(x) \quad \text{Where } F(x) \text{ belongs to some family of functions} \]

• Minimize misclassification error

(Not that simple: we need some guarantees that there will be generalization)
Overview of section

• Object detection with classifiers

• **Boosting**
  – Gentle boosting
  – Weak detectors
  – Object model
  – Object detection
A simple object detector with Boosting

Download
• Toolbox for manipulating dataset
• Code and dataset

Matlab code
• Gentle boosting
• Object detector using a part based model

Dataset with cars and computer monitors

http://people.csail.mit.edu/torralba/iccv2005/
Why boosting?

• A simple algorithm for learning robust classifiers
  – Freund & Shapire, 1995
  – Friedman, Hastie, Tibshhirani, 1998

• Provides efficient algorithm for sparse visual feature selection
  – Tieu & Viola, 2000
  – Viola & Jones, 2003

• Easy to implement, not requires external optimization tools.
Boosting

• Defines a classifier using an additive model:

\[ F(x) = \alpha_1 f_1(x) + \alpha_2 f_2(x) + \alpha_3 f_3(x) + \ldots \]
Boosting

• Defines a classifier using an additive model:

\[ F(x) = \alpha_1 f_1(x) + \alpha_2 f_2(x) + \alpha_3 f_3(x) + \ldots \]

• We need to define a family of weak classifiers

\[ f_k(x) \] from a family of weak classifiers
Each data point has a class label: $y_t = \begin{cases} +1 & (\bullet) \\ -1 & (\circ) \end{cases}$ and a weight: $w_t = 1$
Toy example

Weak learners from the family of lines

Each data point has a class label:
\[ y_t = \begin{cases} +1 & \text{(○)} \\ -1 & \text{(○)} \end{cases} \]
and a weight:
\[ w_t = 1 \]

\[ h \Rightarrow p(\text{error}) = 0.5 \quad \text{it is at chance} \]
This one seems to be the best

Each data point has a class label:

\[ y_t = \begin{cases} +1 & (\bullet) \\ -1 & (\bigcirc) \end{cases} \]

and a weight:

\[ w_t = 1 \]

This is a ‘\textbf{weak classifier}’: It performs slightly better than chance.
Toy example

We set a new problem for which the previous weak classifier performs at chance again.

Each data point has a class label:

\[ y_t = \begin{cases} 
+1 (\bigcirc) \\
-1 (\bigotimes) 
\end{cases} \]

We update the weights:

\[ w_t \leftarrow w_t \exp\{-y_t H_t\} \]
Toy example

We set a new problem for which the previous weak classifier performs at chance again.

Each data point has a class label:

\[ y_t = \begin{cases} +1 & (\circ) \\ -1 & (\bigcirc) \end{cases} \]

We update the weights:

\[ w_t \leftarrow w_t \exp\{-y_t H_t\} \]
We set a new problem for which the previous weak classifier performs at chance again.

Each data point has a class label:

\[ y_t = \begin{cases} 
+1 & (\bullet) \\
-1 & (\bigcirc)
\end{cases} \]

We update the weights:

\[ w_t \leftarrow w_t \exp\{-y_t H_t\} \]
We set a new problem for which the previous weak classifier performs at chance again.

Each data point has a class label:

\[
y_t = \begin{cases} 
+1 & (\bullet) \\
-1 & (\circ) 
\end{cases}
\]

We update the weights:

\[
w_t \leftarrow w_t \exp\{-y_t H_t\}
\]
Toy example

The strong (non-linear) classifier is built as the combination of all the weak (linear) classifiers.
Boosting

- Different cost functions and minimization algorithms result in various flavors of Boosting
- In this demo, I will use gentleBoosting: it is simple to implement and numerically stable.
Overview of section

• Object detection with classifiers

• Boosting
  – **Gentle boosting**
  – Weak detectors
  – Object model
  – Object detection
Boosting fits the additive model

\[ F(x) = f_1(x) + f_2(x) + f_3(x) + \ldots \]

by minimizing the exponential loss

\[ J(F) = \sum_{t=1}^{N} e^{-y_t F(x_t)} \]

The exponential loss is a differentiable upper bound to the misclassification error.
Exponential loss

Misclassification error
Squared error
Exponential loss

Squared error
\[ J = \sum_{t=1}^{N} [y_t - F(x_t)]^2 \]

Exponential loss
\[ J = \sum_{t=1}^{N} e^{-y_tF(x_t)} \]

yF(x) = margin
Boosting

Sequential procedure. At each step we add

\[ F(x) \leftarrow F(x) + f_m(x) \]

to minimize the residual loss

\[
(\phi_m) = \arg \min_{\phi} \sum_{t=1}^{N} J(y_i, F(x_t) + f(x_t; \phi))
\]

gentleBoosting

• At each iteration:
  We chose $f_m(x)$ that minimizes the cost:

$$J(F + f_m) = \sum_{t=1}^{N} e^{-y_t(F(x_t)+f_m(x_t))}$$

Instead of doing exact optimization, gentle Boosting minimizes a Taylor approximation of the error:

$$J(F) \propto \sum_{t=1}^{N} e^{-y_tF(x_t)}(y_t - f_m(x_t))^2$$

Weights at this iteration

At each iterations we just need to solve a weighted least squares problem

Weak classifiers

- The input is a set of weighted training samples \((x, y, w)\)

- Regression stumps: simple but commonly used in object detection.

\[ f_m(x) = a[x_k < \theta] + b[x_k \geq \theta] \]

Four parameters: \([a, b, \theta, k]\)

\[ a = \mathbb{E}_w(y[x < \theta]) \]
\[ b = \mathbb{E}_w(y[x > \theta]) \]

fitRegressionStump.m
function classifier = gentleBoost(x, y, Nrounds)
    ...
    for m = 1:Nrounds
        fm = selectBestWeakClassifier(x, y, w);
        w = w .* exp(- y .* fm);
        % store parameters of fm in classifier
    end
    ...

Initialize weights $w = 1$
Solve weighted least-squares
Re-weight training samples
Demo gentleBoosting

Demo using Gentle boost and stumps with hand selected 2D data:

> demoGentleBoost.m
Flavors of boosting

• AdaBoost (Freund and Shapire, 1995)
• Real AdaBoost (Friedman et al, 1998)
• LogitBoost (Friedman et al, 1998)
• Gentle AdaBoost (Friedman et al, 1998)
• BrownBoosting (Freund, 2000)
• FloatBoost (Li et al, 2002)
• …
Overview of section

• Object detection with classifiers

• Boosting
  – Gentle boosting
  – Weak detectors
  – Object model
  – Object detection
We will now define a family of visual features that can be used as weak classifiers ("weak detectors")

\[ h_i(I, x, y) \]

Takes image as input and the output is binary response. The output is a weak detector.
Object recognition
Is it really so hard?

Find the chair in this image

But what if we use smaller patches? Just a part of the chair?
Parts
But what if we use smaller patches? Just a part of the chair?

Find a chair in this image

Seems to fire on legs… not so bad
Weak detectors

Textures of textures
Tieu and Viola, CVPR 2000. One of the first papers to use boosting for vision.

Every combination of three filters generates a different feature

This gives thousands of features. Boosting selects a sparse subset, so computations on test time are very efficient. Boosting also avoids overfitting to some extend.
Weak detectors

Haar filters and integral image
Viola and Jones, ICCV 2001

The average intensity in the block is computed with four sums independently of the block size.
Haar wavelets

Papageorgiou & Poggio (2000)

Polynomial SVM
Edges and chamfer distance

Gavrila, Philomin, ICCV 1999
Weak detector = k edge fragments and threshold. Chamfer distance uses 8 orientation planes.
Histograms of oriented gradients

- SIFT, D. Lowe, ICCV 1999
- Shape context
  Belongie, Malik, Puzicha, NIPS 2000

- Dalal & Trigs, 2006
Weak detectors

Other weak detectors:
• Carmichael, Hebert 2004
• Yuille, Snow, Nitzbert, 1998
• Amit, Geman 1998
• Papageorgiou, Poggio, 2000
• Heisele, Serre, Poggio, 2001
• Agarwal, Awan, Roth, 2004
• Schneiderman, Kanade 2004
• ...
Weak detectors

**Part based**: similar to part-based generative models. We create weak detectors by using parts and voting for the object center location.

These features are used for the detector on the course web site.
Weak detectors

First we collect a set of part templates from a set of training objects.

Vidal-Naquet, Ullman (2003)
Weak detectors

We now define a family of “weak detectors” as:

\[ h_i(I, x, y) = [I \otimes P_i] * g_i \]

Better than chance
Weak detectors

We can do a better job using filtered images

\[ h_i(I, x, y) = [I * f_i \otimes P_i] * g_i \]

Still a weak detector but better than before
Training

First we evaluate all the N features on all the training images.

Feature 1

\[
\begin{bmatrix}
\text{(image)} & \ast & \begin{array}{c}
\text{feature output}
\end{array}
\end{bmatrix} \ast \begin{array}{c}
\text{background}
\end{array} = \begin{array}{c}
\text{output image}
\end{array}
\]

\vdots

Feature N

\[
\begin{bmatrix}
\text{(image)} & \ast & \begin{array}{c}
\text{feature output}
\end{array}
\end{bmatrix} \ast \begin{array}{c}
\text{background}
\end{array} = \begin{array}{c}
\text{output image}
\end{array}
\]

Then, we sample the feature outputs on the object center and at random locations in the background:

\[
\begin{bmatrix}
1 & 2 & 3 & \cdots & N
\end{bmatrix}
\]

Positive Training Vectors

\[
\begin{bmatrix}
1 & 2 & 3 & \cdots & N
\end{bmatrix}
\]

Negative Training Vectors

\[
\begin{bmatrix}
1 & 2 & 3 & \cdots & N
\end{bmatrix}
\]
Representation and object model

Selected features for the screen detector

Lousy painter
Representation and object model

Selected features for the car detector

1 2 3 4 ... 10 ... 100
Overview of section

• Object detection with classifiers

• Boosting
  – Gentle boosting
  – Weak detectors
  – Object model
  – Object detection
Example: screen detection
Example: screen detection

Feature output

Thresholded output

Weak ‘detector’
Produces many false alarms.
Example: screen detection

Feature output → Thresholded output → Strong classifier at iteration 1
Example: screen detection

Feature output → Thresholded output → Strong classifier

Second weak ‘detector’
Produces a different set of false alarms.
Example: screen detection

Feature output → Thresholded output → Strong classifier

Strong classifier at iteration 2
Example: screen detection

Feature output → Thresholded output → Strong classifier

Strong classifier at iteration 10
Example: screen detection

Feature output

Thresholded output

Strong classifier

Adding features

Final classification

Strong classifier at iteration 200
Maximal suppression

Detect local maximum of the response. We are only allowed detecting each object once. The rest will be considered false alarms.

This post-processing stage can have a very strong impact in the final performance.
Evaluation

When do we have a correct detection?

Is this correct?

\[
\frac{\text{Area intersection}}{\text{Area union}} > 0.5
\]

- ROC
- Precision-recall
Evaluation

Precision-recall    ROC
Demo

Demo of screen and car detectors using parts, Gentle boost, and stumps:

> runDetector.m
Laboratorio


1)  Instalacion: seguid las instrucciones en “setup”

2)  Boosting
    Ejecutad demoGentleBoost.m
    Encontrad casos con ejemplos positivos y negativos que sean separables y otros que no lo sean
    Cuales son las limitaciones? Se podrian evitar?

3)  Deteccion de objetos
    Ejecutad runDetector.m
    Modificar el codigo para que pueda detectar objetos a diferentes escalas