Object detection and recognition
Example problems

- Detecting rigid objects
- PASCAL challenge
- Detecting non-rigid objects
- Medical image analysis
- Segmenting cells
Template matching

- “Appearance based method”.
- Object of interest defined by a template.
- Compare template to image data under each possible shift.
  - Scale template or image to find objects of different sizes.
Issues

• How to compare template to image?
  - Sum of squared differences, dot product, etc.

• What should we use as the template?
  - Cropped picture of the object.
  - Average of multiple pictures.

• How do we handle variation in appearance?
  - Variations due to pose, lighting, non-rigid objects, etc.
  - Don’t compare image intensities directly.
  - Works well for certain classes of objects (face detection).
Image intensities
Histogram of Gradient (HOG) features

- Discretize gradient orientation at each pixel into 9 possible values.
- Image is partitioned into 8x8 pixel blocks.
- In each block we compute a histogram of gradient orientations.
  - 9 dimensional feature vector.
  - Invariant to changes in lighting, small deformations, etc.
Template matching with HOG features

- Templates define weights for features in rectangular window.
  - $N \times M \times 9$ weights for a window of size $N \times M$.
- Take dot product of template and subwindow of HOG pyramid.

![Diagram showing template matching with HOG features](image)

Score at this location is $W \cdot X$
Face detection

template

element result
Learning the template

• Suppose we have positive and negative examples:
  - \((x_1, y_1), \ldots, (x_n, y_n)\)
  - \(x_i\) is a subwindow of a HOG pyramid.
  - \(y_i = 1\) if subwindow contains a face
  - \(y_i = -1\) otherwise

• Look for template \(W\) such that every positive example scores higher than every negative example.
  - \(\langle W \cdot x_i \rangle > b\) if \(y_i = 1\)
  - \(\langle W \cdot x_i \rangle < b\) if \(y_i = -1\)
Linear classifiers

- Linear classifier:
  - Defined by a weight vector $W$ and bias $b$.
  - $(W \cdot x_i) > b$ if $y_i = 1$
  - $(W \cdot x_i) < b$ if $y_i = -1$

- $W$ and $b$ define a hyperplane separating the positive and negative examples. $W$ is the orientation, $b$ is the distance from the origin.
Learning linear classifiers

- Given \((x_1, y_1), \ldots, (x_n, y_n)\)
- Find hyperplane separating positive and negative examples
- Classical problem in pattern recognition/machine learning
  - Linear programming
  - Perceptron algorithm
  - Support vector machines
Training a face model

• Positive examples:

• Negative examples: random patches from images without faces

• Model learned with SVM:
Face detection

template
Deformable Models

• Template matching works well for faces but...
• How do we handle other types of objects
  – People, cars, etc.
  – Non-rigid objects, object categories, etc.
• Deformable models approach:
  – Consider each object as a **deformed** version of a **template**.
Pictorial Structures

- Part-based models:
  - Each part represents local visual properties.
  - “Springs” capture spatial relationships.

Matching model to image involves joint optimization of part locations “stretch and fit”
Local evidence + global decision

- Parts have a match quality at each image location.
- Local evidence is noisy.
  - Parts are detected in the context of the whole model.
Matching problem

- Model is represented by a graph $G = (V, E)$.
  - $V = \{v_1, \ldots, v_n\}$ are the parts.
  - $(v_i, v_j) \in E$ indicates a connection between parts.
- $m_i(l_i)$ is a cost for placing part $i$ at location $l_i$.
- $d_{ij}(l_i, l_j)$ is a deformation cost.
- Optimal configuration for the object is $L = (l_1, \ldots, l_n)$ minimizing

$$E(L) = \sum_{i=1}^{n} m_i(l_i) + \sum_{(v_i, v_j) \in E} d_{ij}(l_i, l_j)$$
Matching problem

\[ E(L) = \sum_{i=1}^{n} m_i(l_i) + \sum_{(v_i,v_j) \in E} d_{ij}(l_i,l_j) \]

- Assume \( n \) parts, \( k \) possible locations for each part.
  - There are \( k^n \) configurations \( L \).
- If graph is a tree we can use dynamic programming.
  - \( O(nk^2) \) algorithm.
- If \( d_{ij}(l_i,l_j) = g(l_i-l_j) \) we can use min-convolutions.
  - \( O(nk) \) algorithm.
  - As fast as matching each part separately!
Dynamic Programming on Trees

\[ E(L) = \sum_{i=1}^{n} m_i(l_i) + \sum_{(v_i,v_j) \in E} d_{ij}(l_i,l_j) \]

- For each \( l_1 \) find best \( l_2 \):
  - \( \text{Best}_2(l_1) = \min \left[ m_2(l_2) + d_{12}(l_1,l_2) \right] \)

- “Delete” \( v_2 \) and solve problem with smaller model.
  - Let \( m_1(l_1) = m_1(l_1) + \text{Best}_2(l_1) \)

- Keep removing leafs until there is a single part left.
Min-convolution speedup

\[
\text{Best}_2(l_1) = \min \left[ m_2(l_2) + d_{12}(l_1,l_2) \right]
\]

- Brute force: \(O(k^2)\)  ---  \(k\) is number of locations.
- Suppose \(d_{12}(l_1,l_2) = g(l_1-l_2)\):
  - \(\text{Best}_2(l_1) = \min \left[ m_2(l_2) + g(l_1-l_2) \right]\)
- Min-convolution: \(O(k)\) if \(g\) is convex.
Finding motorbikes

Model with 6 parts:
2 wheels
2 headlights
front & back of seat
Human pose estimation
Processing steps

Match costs for each part

DT of non-reference match costs - min-convolution with parabola

Sum shifted DTs and reference match cost

Find best location for reference
Multiscale models with HOG features

Model has a root filter plus deformable parts
Object configuration

Image pyramid
HOG feature pyramid

Score is sum of filter scores minus deformation costs

Multiscale model captures features at two-resolutions
Score of hypothesis

- Let $z$ be an object configuration:
  - Specifies location for root and parts in HOG pyramid.
  - Score is sum of filter scores minus deformation costs.

- Optimize using distance transforms:
  - Compute best score for every possible placement of root.
  - Report placements that lead to good score.

- Connection to linear classifier:
  - Score of $z$ on image $x$ is $W \cdot \Phi(x, z)$
Training

- Training data consists of images with labeled bounding boxes.
- Need to learn the model structure, filters and deformation costs.
Connection to linear classifier

- Positive example specifies that score should be high for some configuration in a range.
- Negative example specifies that score should be low for all configurations in a range.

\[ f_w(x) = \max_z w \cdot \Phi(x, z) \]

- \( w \) is a model
- \( x \) is a detection window (range)
- \( z \) are filter placements

concatenation of features and part displacements
concatenation of filters and deformation parameters
Latent SVMs

\[ f_w(x) = \max_z w \cdot \Phi(x, z) \]

Linear in \( w \) if \( z \) is fixed

Training data: \((x_1, y_1), \ldots, (x_n, y_n)\) with \( y_i \in \{ -1, 1 \} \)

Learning: find \( w \) such that \( y_i f_w(x_i) > 0 \)

\[ w^* = \arg\min_w \lambda \|w\|^2 + \sum_{i=1}^{n} \max(0, 1 - y_i f_w(x_i)) \]

Regularization  Hinge loss
Learned models

Bottle

Car

Sofa

Bicycle
More results