Digital Image Processing (CSE/ECE 478)

Lecture 19 : Representation and Description (3)

Ravi Kiran

Rajvi Shah

Modern Features / Descriptors

- Point Descriptors : SIFT, SURF, DAISY, LBP
- Region Descriptors : HOG, MSER
- Global Descriptors : Bag of Words, GIST
- Introduction to Learned Representation

Modern Features / Descriptors

- **Point Descriptors : SIFT,** SURF, DAISY, LBP
- Region Descriptors : HOG, MSER
- **Global Descriptors : Bag of Visual Words**, GIST
- Introduction to Learned Representation

Modern Features / Descriptors

- **Point Descriptors : SIFT**, SURF, DAISY, LBP
- **Region Descriptors : HOG**, MSER
- **Global Descriptors : Bag of Visual Words**, GIST
- Introduction to Learned Representation

Histogram of Oriented Gradients

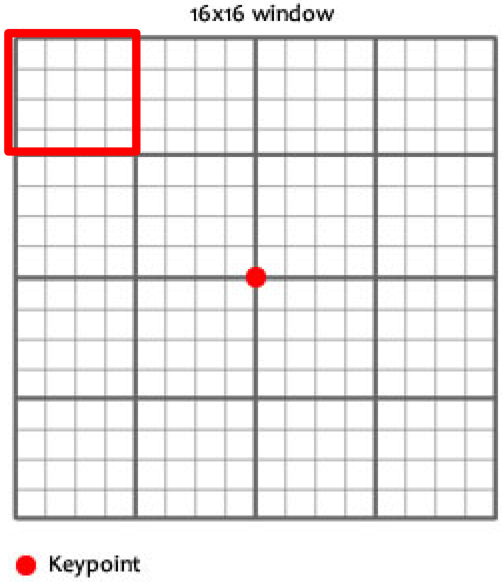
Histogram of Oriented Gradients for Human Detection

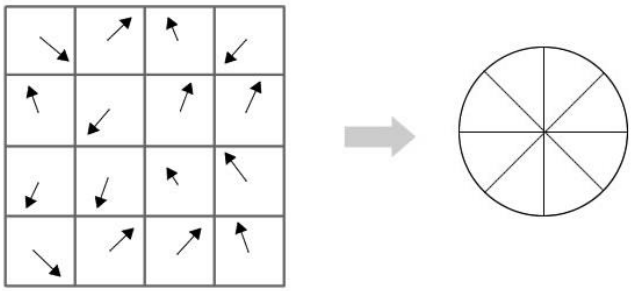
Navneet Dalal & Bill Triggs, CVPR 2005

~24000 citations

Histogram of Oriented Gradients

- Recall SIFT Descriptor





$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2}$$

$$\theta(x,y) = \tan^{-1}((L(x,y+1) - L(x,y-1))/(L(x+1,y) - L(x-1,y)))$$

Histograms of Oriented Gradients for Human Detection

Navneet Dalal and Bill Triggs
INRIA Rhône-Alps, 655 avenue de l'Europe, Montbonnot 38334, France
{Navneet.Dalal,Bill.Triggs}@inrialpes.fr, http://lear.inrialpes.fr

Abstract

We study the question of feature sets for robust visual object recognition, adopting linear SVM based human detection as a test case. After reviewing existing edge and gradient based descriptors, we show experimentally that grids of Histograms of Oriented Gradient (HOG) descriptors significantly outperform existing feature sets for human detection. We study the influence of each stage of the computation on performance, concluding that fine-scale gradients, fine orientation binning, relatively coarse spatial binning, and high-quality local contrast normalization in overlapping descriptor blocks are all important for good results. The new approach gives near-perfect separation on the original MIT pedestrian database, so we introduce a more challenging dataset containing over 1800 annotated human images with a large range of pose variations and backgrounds.

What are the claims of the paper?

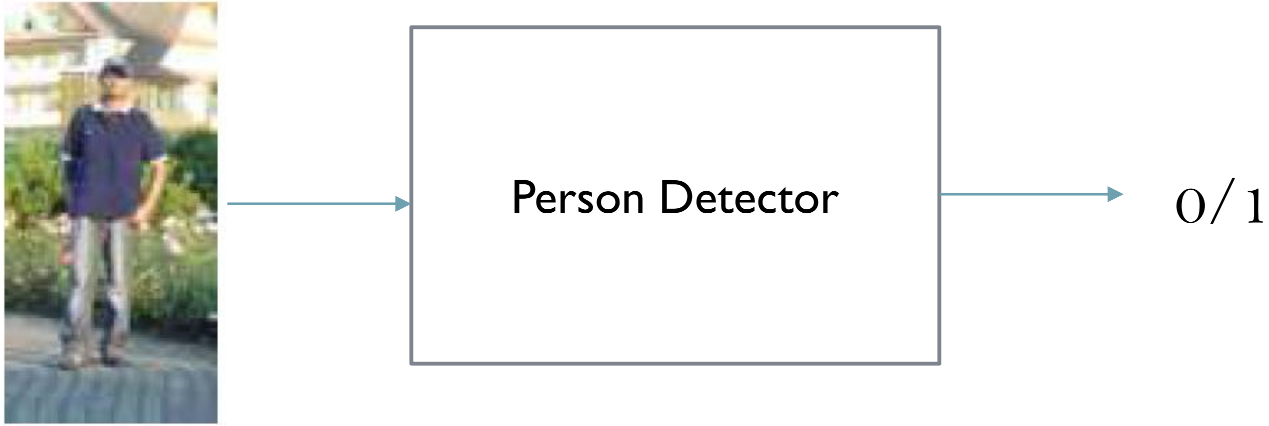
- Grids of HOG outperform other features
- Study effect of each stage of computation (choice of parameters) on performance
- Introduce a harder dataset

§2, give an overview of our method §3, describe our data sets in §4 and give a detailed description and experimental evaluation of each stage of the process in §5–6. The main conclusions are summarized in §7.

2 Previous Work

There is an extensive literature on object detection, but here we mention just a few relevant papers on human detection [18, 17, 22, 16, 20]. See [6] for a survey. Papageorgiou *et al* [18] describe a pedestrian detector based on a polynomial SVM using rectified Haar wavelets as input descriptors, with a parts (subwindow) based variant in [17]. Depoortere *et al* give an optimized version of this [2]. Gavrila & Philomen [8] take a more direct approach, extracting edge images and matching them to a set of learned exemplars using chamfer distance. This has been used in a practical real-time pedes-

Detection / Binary Classification



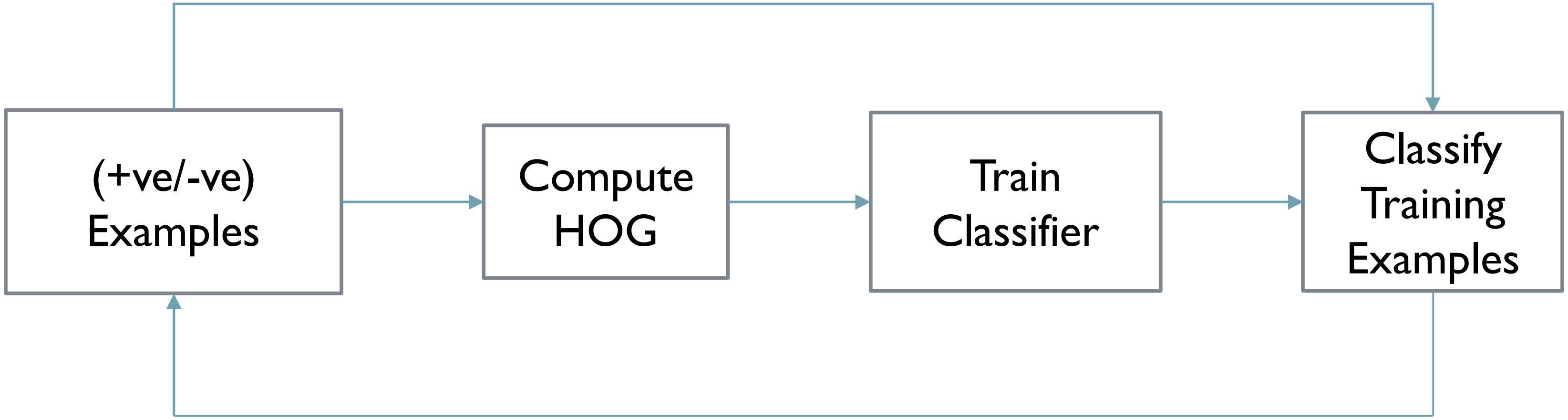
Detection / Binary Classification

		Predicted Class	
		No	Yes
Observed Class	No	TN	FP
	Yes	FN	TP

TN	True Negative
FP	False Positive
FN	False Negative
TP	True Positive

Pedestrian Detection Training Pipeline

- 1239 Positive Examples (+H-Reflections = 2478)
- 12180 Negative Examples (Person-free windows)



Hard Negatives added to 1 more training iteration

Pedestrian Detection Performance Evaluation

- False Positives Per window Tested
- Performance reported in 10^{-4} FPPW

- Detection Error Tradeoff (DET)
  - Miss Rate / FPPW
  - Lower the better

HOG Computation

Input Image (I)

Normalize Gamma & Color

Compute Gradients

Weighted Vote into Spatial & Orientation Cells

Contrast Norm. Over Overlapping Spatial Blocks

Collect HOGs over detection window

Linear SVM

Person (Y/N)

HOG Computation

Input Image (I)

Normalize Gamma & Color

Compute Gradients

Weighted Vote into Spatial & Orientation Cells

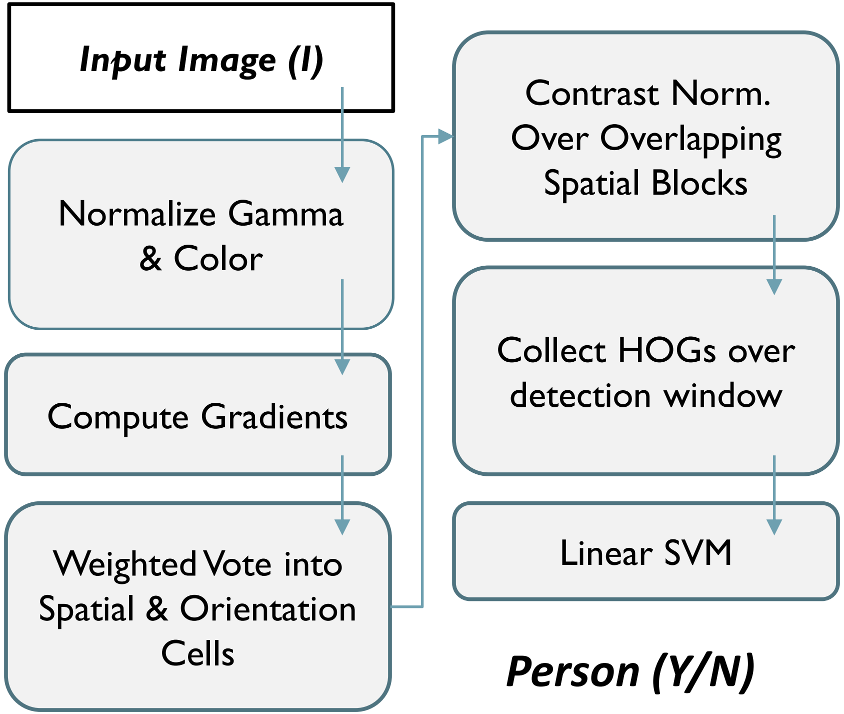
Contrast Norm. Over Overlapping Spatial Blocks

Collect HOGs over detection window

Linear SVM

Person (Y/N)

HOG Computation



HOG Computation

Input Image (I)

Normalize Gamma & Color

Compute Gradients

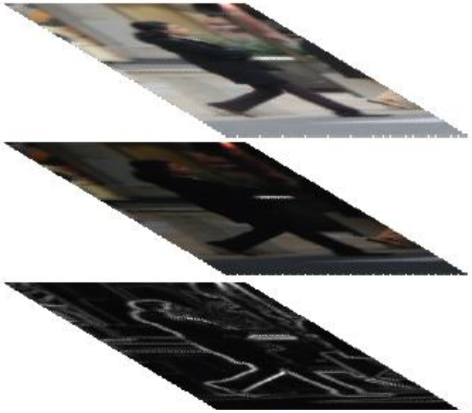
Weighted Vote into Spatial & Orientation Cells

Contrast Norm. Over Overlapping Spatial Blocks

Collect HOGs over detection window

Linear SVM

Person (Y/N)



HOG Computation

Input Image (I)

Normalize Gamma & Color

Compute Gradients

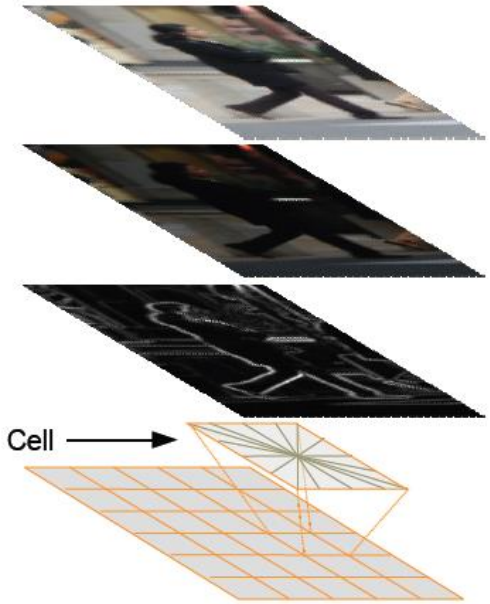
Weighted Vote into Spatial & Orientation Cells

Contrast Norm. Over Overlapping Spatial Blocks

Collect HOGs over detection window

Linear SVM

Person (Y/N)



HOG Computation

Input Image (I)

Normalize Gamma & Color

Compute Gradients

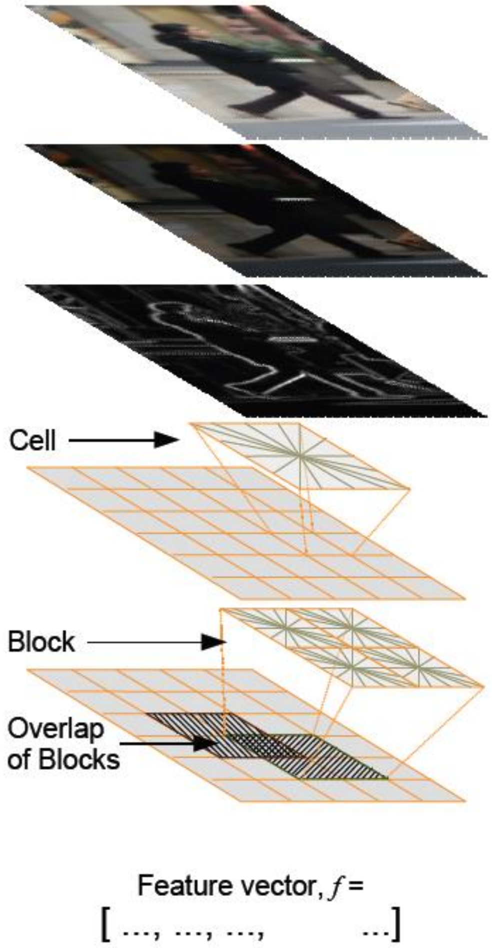
Weighted Vote into Spatial & Orientation Cells

Contrast Norm. Over Overlapping Spatial Blocks

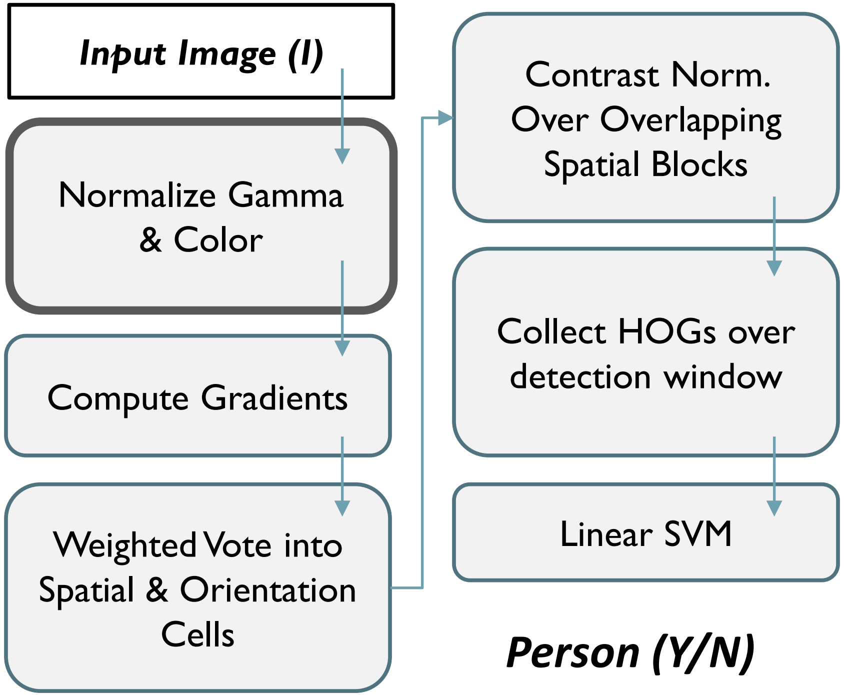
Collect HOGs over detection window

Linear SVM

Person (Y/N)



HOG Computation



- Tested with
  - RGB
  - LAB
  - Grayscale
- Gamma Normalization and Compression
  - Square root
  - Log

HOG Computation

Input Image (I)

Normalize Gamma & Color

Compute Gradients

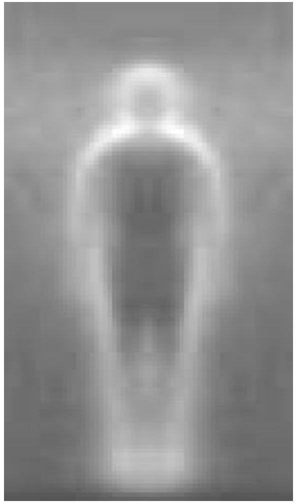
Weighted Vote into Spatial & Orientation Cells

Contrast Norm. Over Overlapping Spatial Blocks

Collect HOGs over detection window

Linear SVM

Person (Y/N)



Outperforms

-1	0	1

centered

0	1
-1	0

diagonal

-1	1

uncentered

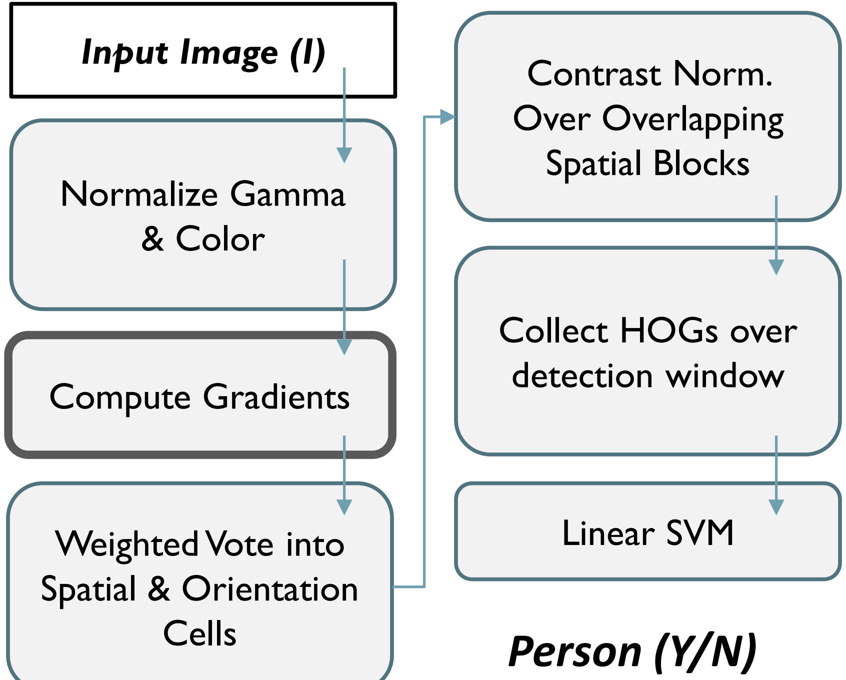
-1	0	1
-2	0	2
-1	0	1

Sobel

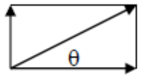
1	-8	0	8	-1

cubic-corrected

HOG Computation



- Centered: $f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x-h)}{2h}$
- Filter masks in x and y directions
  - Centered: $\begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$ $\begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$
- Gradient
  - Magnitude: $s = \sqrt{s_x^2 + s_y^2}$



  - Orientation: $\theta = \arctan(\frac{s_y}{s_x})$

HOG Computation

Input Image (I)

Normalize Gamma & Color

Compute Gradients

Weighted Vote into Spatial & Orientation Cells

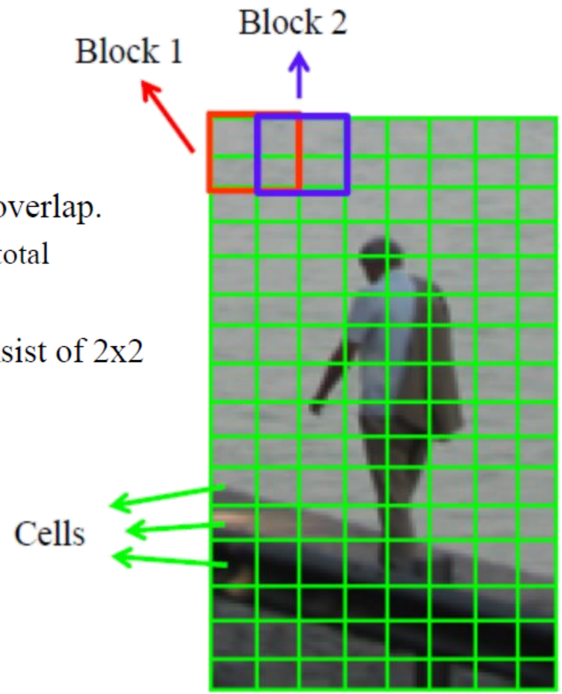
Contrast Norm. Over Overlapping Spatial Blocks

Collect HOGs over detection window

Linear SVM

Person (Y/N)

- 16x16 blocks of 50% overlap.
  - 7x15=105 blocks in total
- Each block should consist of 2x2 cells with size 8x8.



HOG Computation

Input Image (I)

Normalize Gamma & Color

Compute Gradients

Weighted Vote into Spatial & Orientation Cells

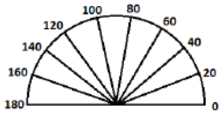
Contrast Norm. Over Overlapping Spatial Blocks

Collect HOGs over detection window

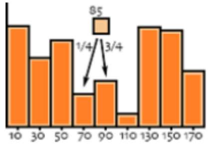
Linear SVM

Person (Y/N)

- Each block consists of 2x2 cells with size 8x8
- Quantize the gradient orientation into 9 bins (0-180)



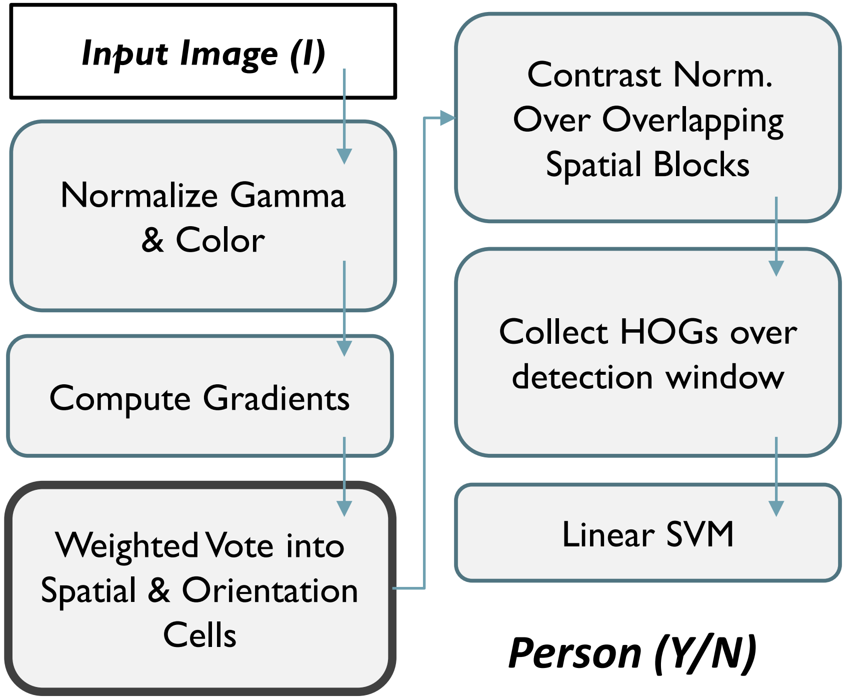
9 Bins



Bin centers

  - The vote is the gradient magnitude
  - Interpolate votes linearly between neighboring bin centers.
    - Example: if θ=85 degrees.
    - Distance to the bin center Bin 70 and Bin 90 are 15 and 5 degrees, respectively.
    - Hence, ratios are 5/20=1/4, 15/20=3/4.
  - The vote can also be weighted with Gaussian to down weight the pixels near the edges of the block.

HOG Computation



5 Contrast Normalization Schemes

L2-norm, $\mathbf{v} \rightarrow \mathbf{v}/\sqrt{\|\mathbf{v}\|_2^2 + \epsilon^2}$

L2-Hys, L2-norm followed by clipping

$\boxed{\textit{L1-norm}, \mathbf{v} \rightarrow \mathbf{v}/(\|\mathbf{v}\|_1 + \epsilon)}$ Red. ~5%

L1-sqrt

$\boxed{\textit{No Normalization}}$ Reduces ~27%

HOG Computation

Input Image (I)

Normalize Gamma & Color

Compute Gradients

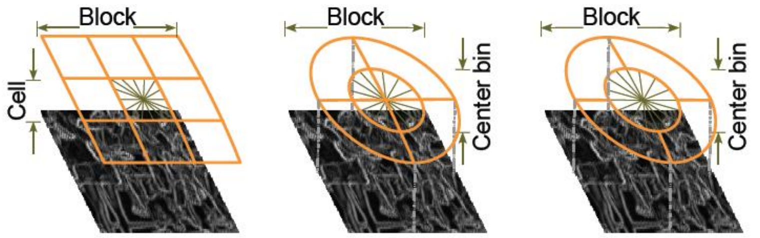
Weighted Vote into Spatial & Orientation Cells

Contrast Norm. Over Overlapping Spatial Blocks

Collect HOGs over detection window

Linear SVM

Person (Y/N)



(a) R-HOG/SIFT (b) C-HOG (c) Single centre C-HOG

Variants of HOG descriptors. (a) A rectangular HOG (R-HOG) descriptor with 3 × 3 blocks of cells. (b) Circular HOG (C-HOG) descriptor with the central cell divided into angular sectors as in shape contexts. (c) A C-HOG descriptor with a single central cell.

HOG Computation

Input Image (I)

→ Normalize Gamma & Color

→ Compute Gradients

→ Weighted Vote into Spatial & Orientation Cells

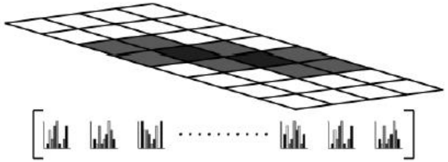
→ Contrast Norm. Over Overlapping Spatial Blocks

→ Collect HOGs over detection window

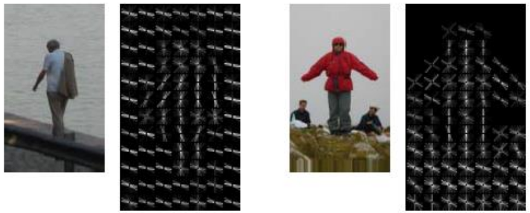
→ Linear SVM

Person (Y/N)

- Concatenate histograms
  - Make it a 1D vector of length 3780.

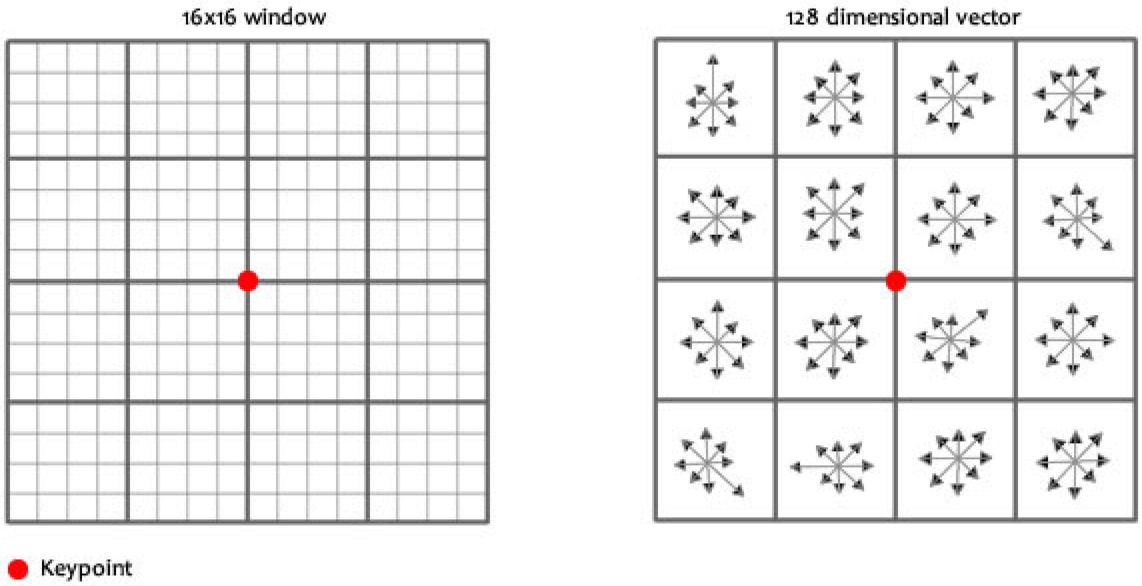


- Visualization



Histogram of Oriented Gradients

- Recall SIFT Descriptor



SIFT Vs HOG

SIFT

- 128 dimensional vector
- 16 by 16 window
- 4x4 sub-window (16 total)
- 8 bin histogram

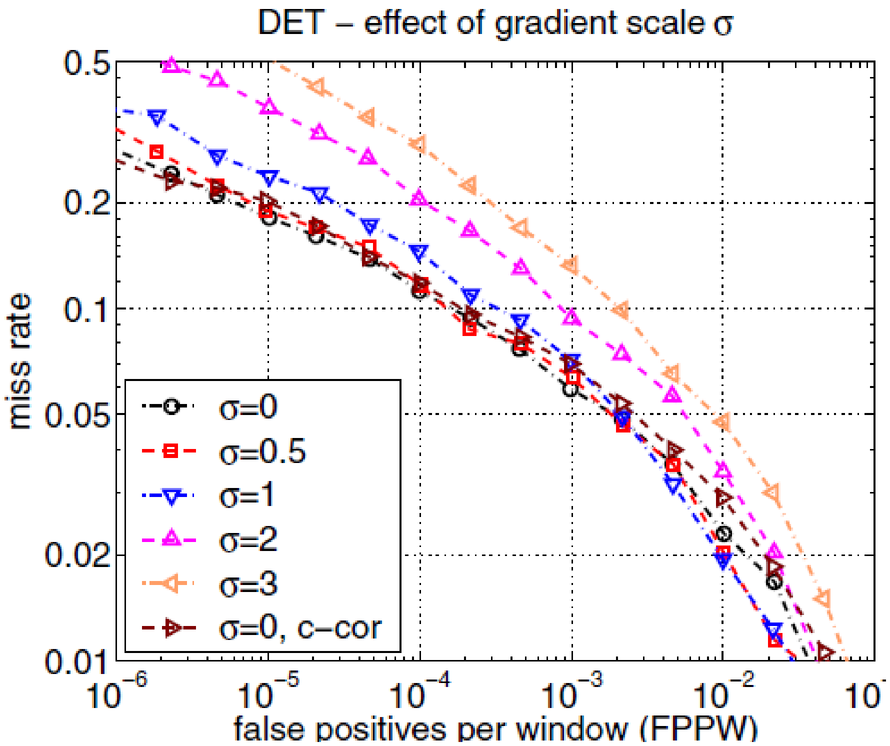
HOG

- 3,780 dimensional vector
- 64 by 128 window
- 16 by 16 blocks with overlap
- Each block consists of 2 by 2 cells each of 8 by 8
- Overlapping
- 9 bin histogram

HOG Parameters and Schemes

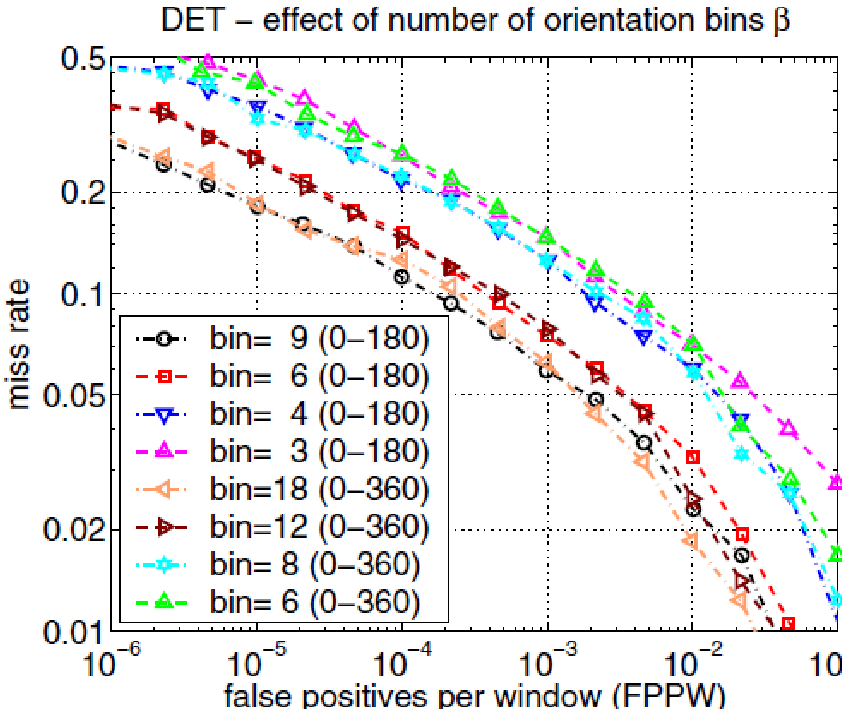
- Schemes
  - Color Space
  - Gradient Operator
  - Signed vs. Unsigned Grad
  - Block-type
    - Rectangular/Circular
  - Norm-type
- Parameters
  - Gradient Scale
  - Number of Gradient Bins
  - Block Overlap

Parameter Sweeping

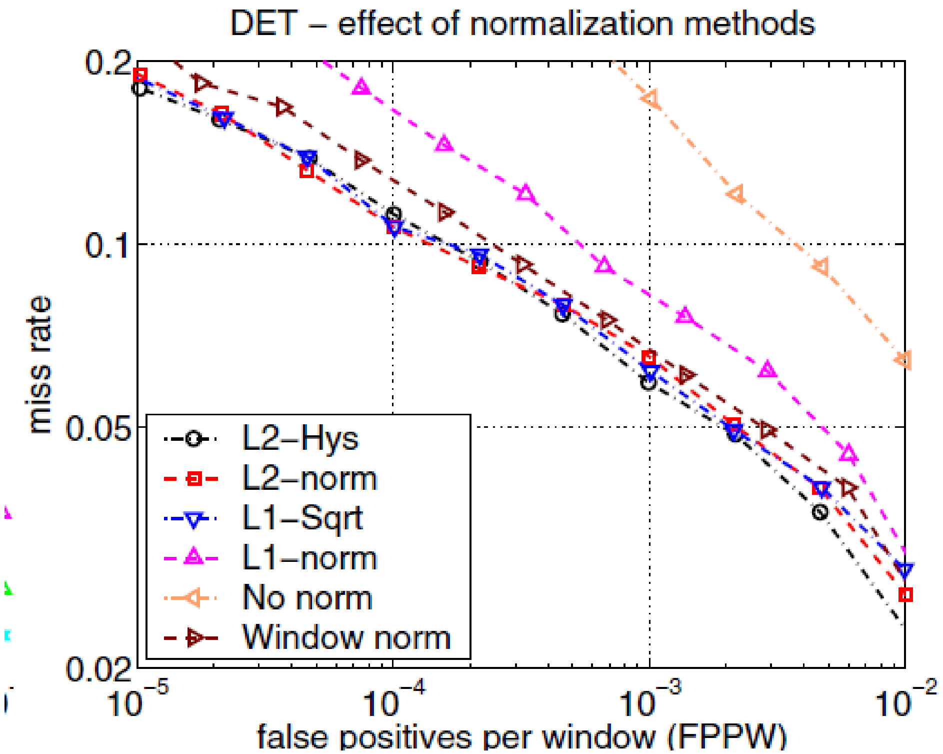


(a)

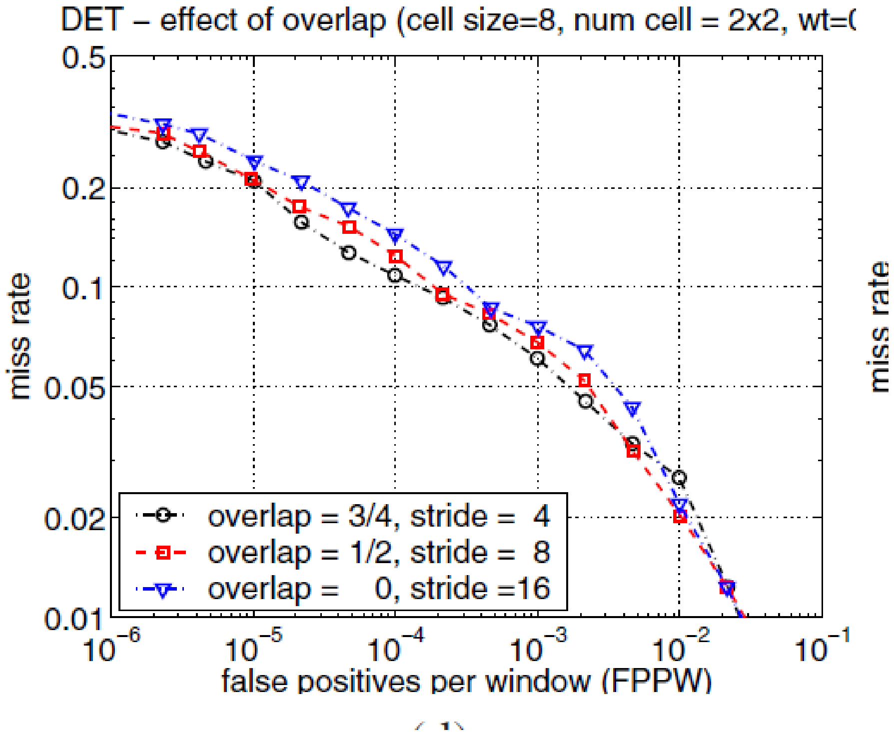
Parameter Sweeping



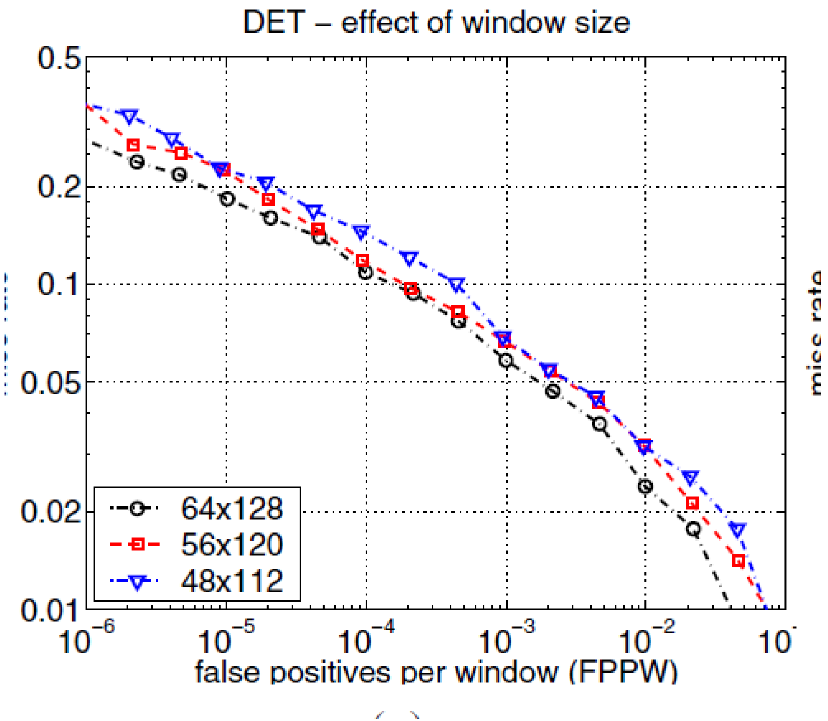
Parameter Sweeping



Parameter Sweeping

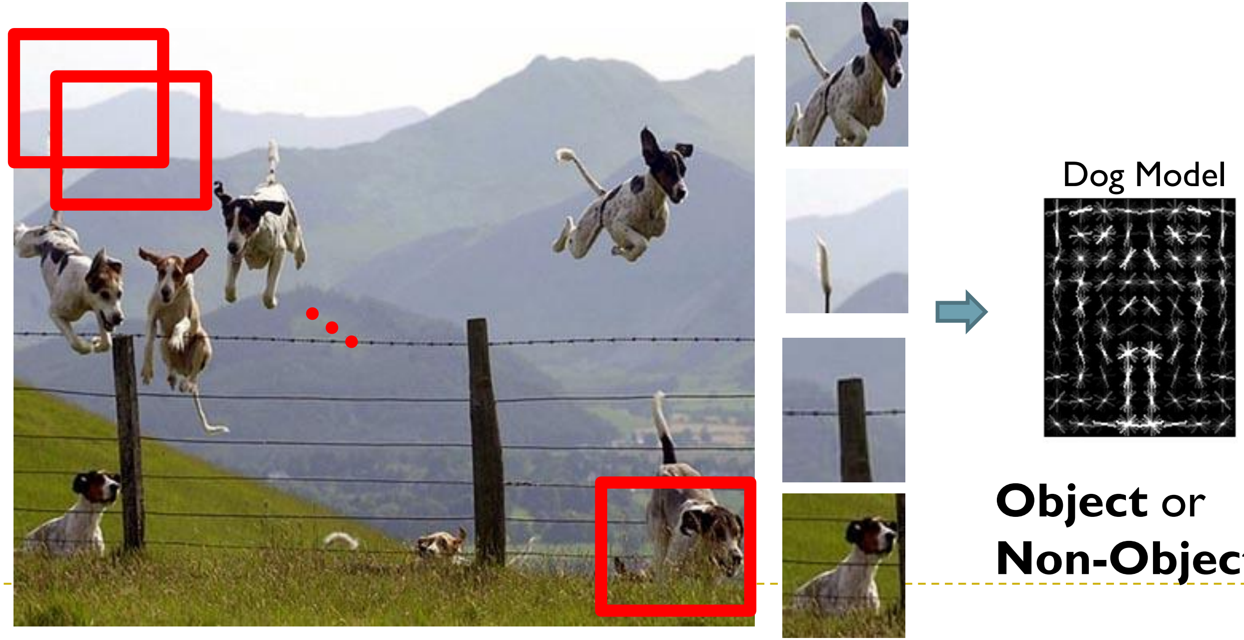


Parameter Sweeping



Object Category Detection

- Focus on object search: "Where is it?"
- Build templates that quickly differentiate object patch from background patch



Object or
Non-Object?

Challenges in modeling the object class

Illumination

Object pose

Clutter

Occlusions

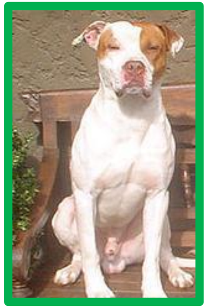
Intra-class appearance

Viewpoint

Slide from K. Grauman, B. Leibe

Challenges in modeling the non-object class

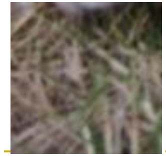
True
Detections



Bad
Localization

Confused with
Similar Object

Misc. Background



Confused with
Dissimilar Objects

General Process of Object Recognition

Specify Object Model

↓

Generate Hypotheses

↓

Score Hypotheses

↓

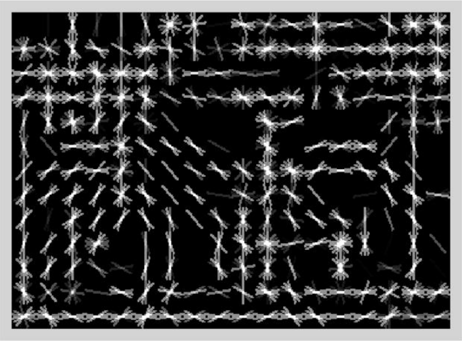
Resolve Detections

What are the object parameters?

Specifying an object model

1. Statistical Template in Bounding Box
   - Object is some (x,y,w,h) in image
   - Features defined wrt bounding box coordinates

Image



Template Visualization

Images from Felzenszwalb

General Process of Object Recognition

Specify Object Model

↓

Generate Hypotheses

↓

Score Hypotheses

↓

Resolve Detections

Propose an alignment of the model to the image

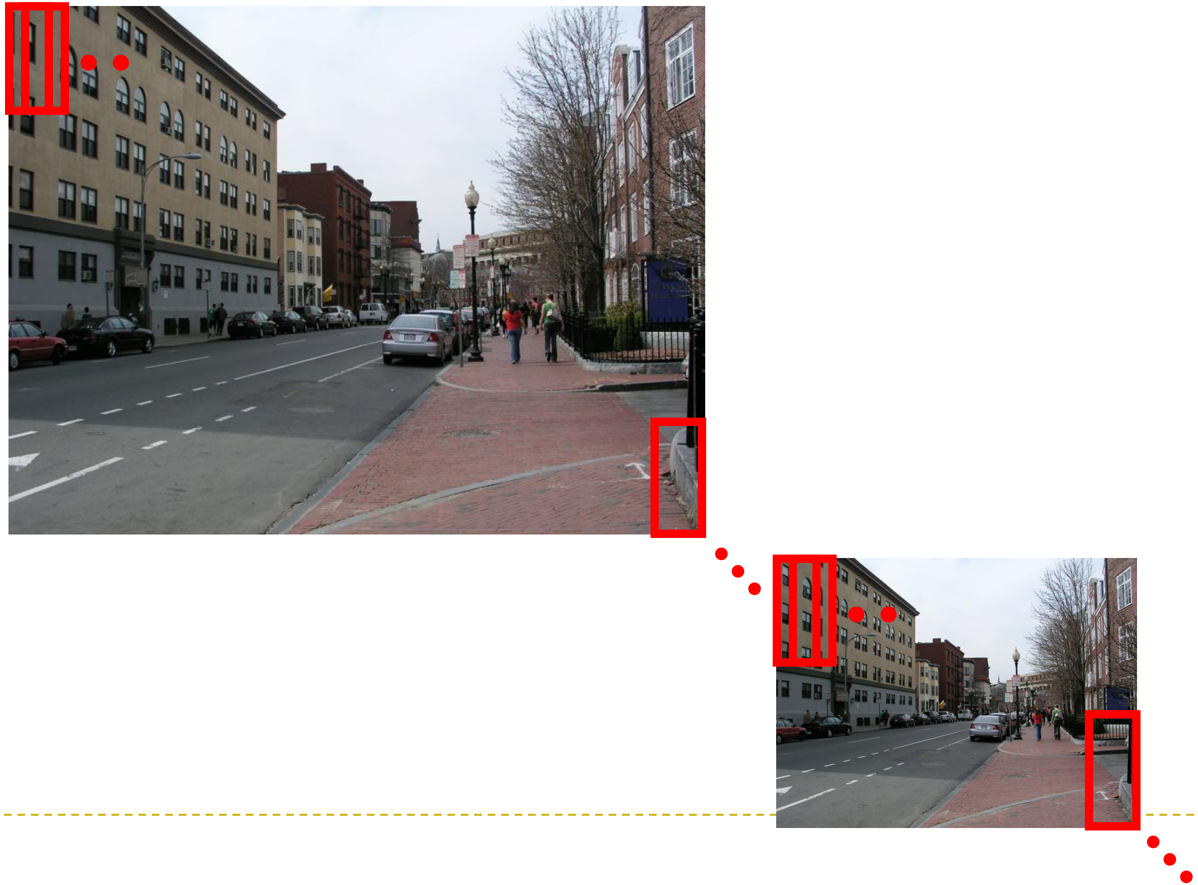
Generating hypotheses

1. Sliding window
   - Test patch at each location and scale

Generating hypotheses

1. Sliding window
   - Test patch at each location and scale

Sliding window: a simple alignment solution



Each window is separately classified

General Process of Object Recognition

Specify Object Model

↓

Generate Hypotheses

↓

Score Hypotheses

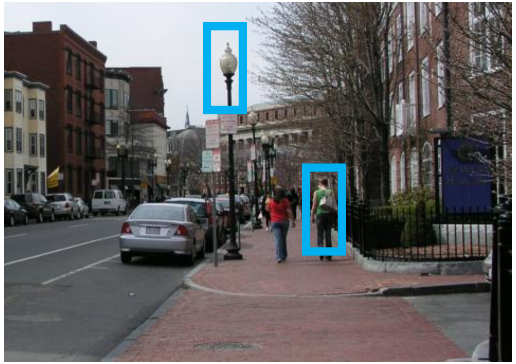
↓

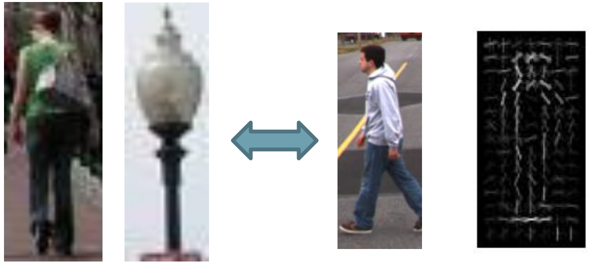
Resolve Detections

Mainly-gradient based or CNN features, usually based on summary representation, many classifiers

Score Hypothesis

1. Classifiers
   - Compute similarity to an example object or to a summary representation
   - Which differences in appearance are important?





Aligned Possible Objects

Exemplar

Summary

Statistical Template

- Object model = sum of scores of features at fixed positions

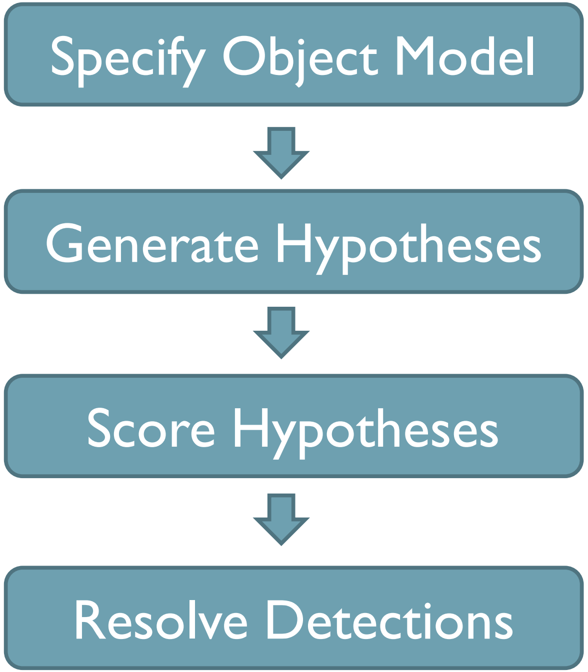
$+3 +2 -2 -1 -2.5 = -0.5 \overset{?}{>} 7.5$

Non-object

$+4 +1 +0.5 +3 +0.5 = 10.5 \overset{?}{>} 7.5$

Object

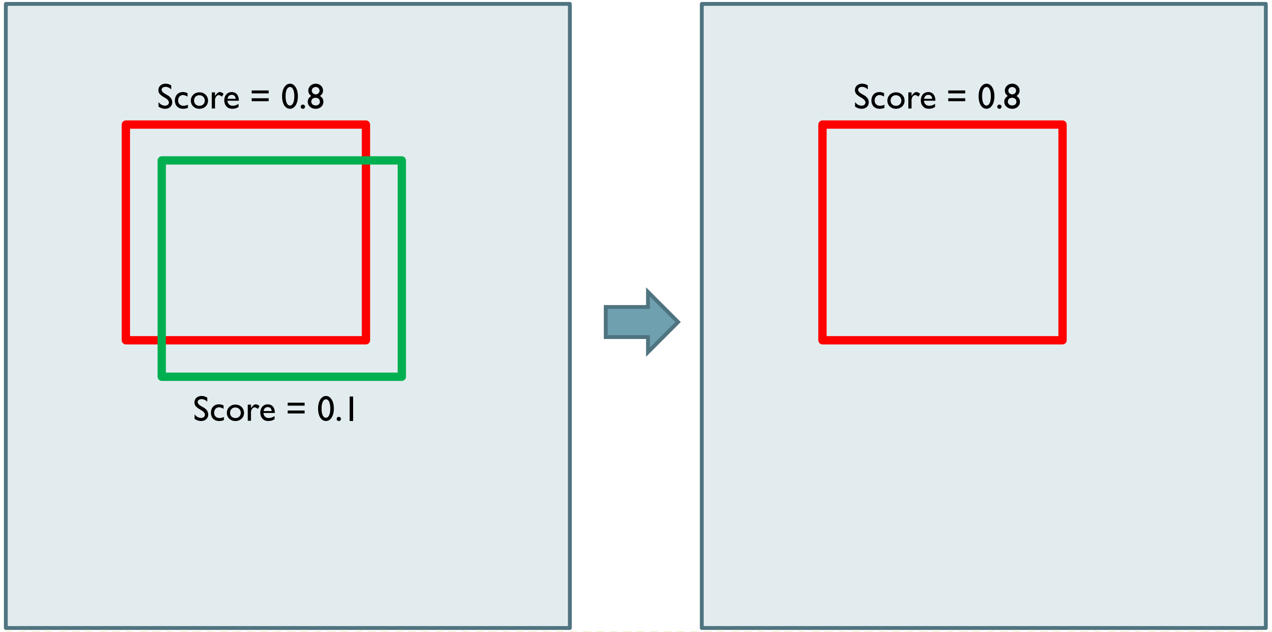
General Process of Object Recognition



Rescore each proposed object based on whole set

Resolving detection scores

1. Non-max suppression



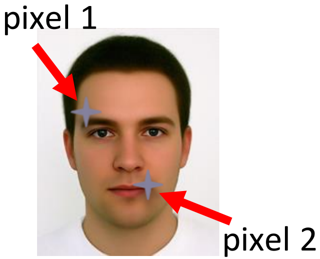
Design challenges

- How to efficiently search for likely objects
  - Even simple models require searching hundreds of thousands of positions and scales
- Feature design and scoring
  - How should appearance be modeled? What features correspond to the object?
- How to deal with different viewpoints?
  - Often train different models for a few different viewpoints
- Implementation details
  - Window size
  - Aspect ratio
  - Translation/scale step size
  - Non-maxima suppression

Modern Features / Descriptors

- **Point Descriptors : SIFT**, SURF, DAISY, LBP
- **Region Descriptors : HOG**, MSER
- **Global Descriptors : Bag of Visual Words**, GIST
- **Introduction to Learned Representation**

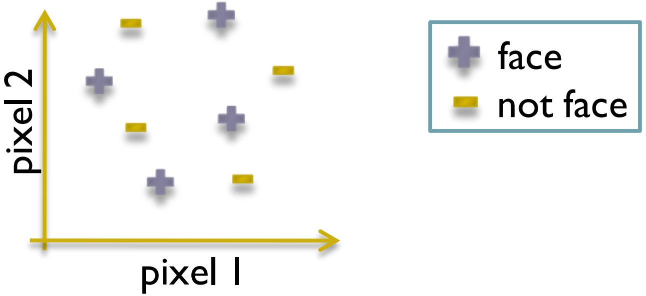
A simple learning based pipeline for Computer Vision



Input

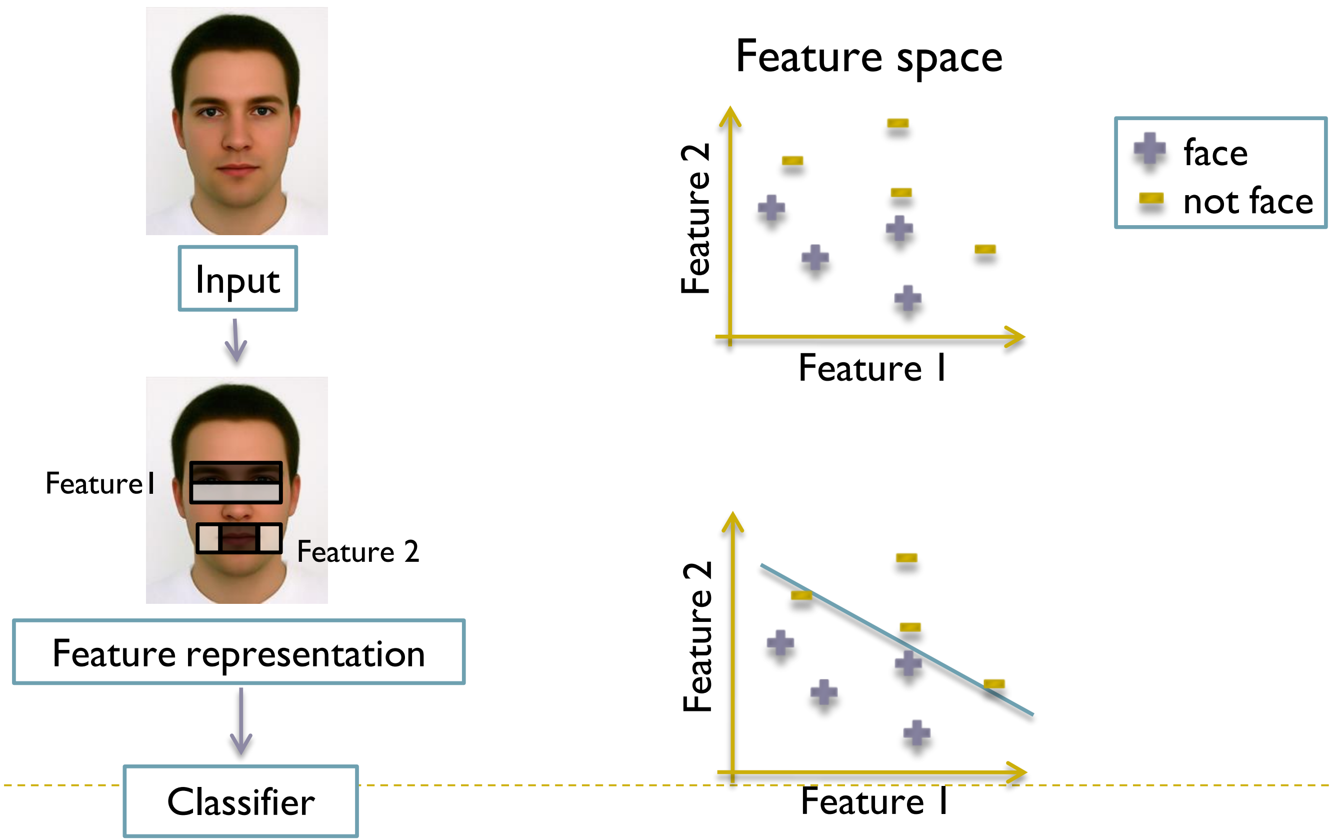
Classifier

Pixel space

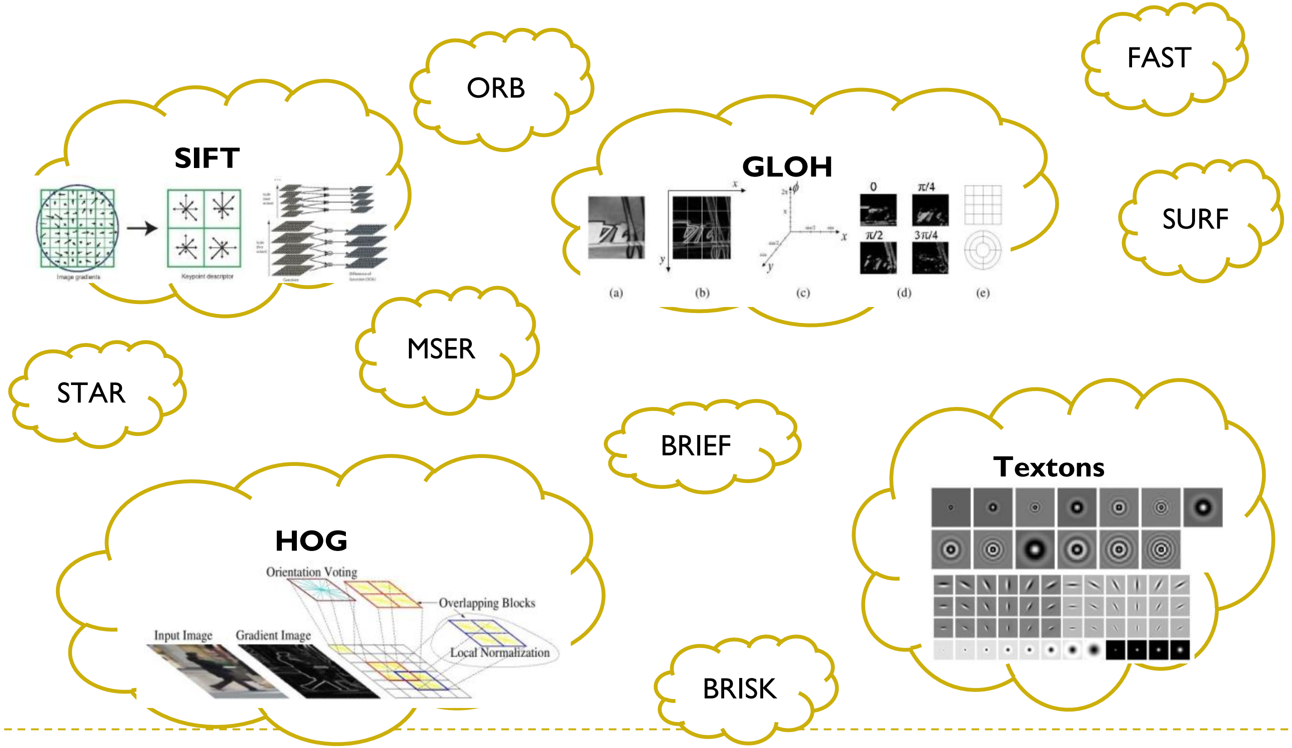


Classification is difficult

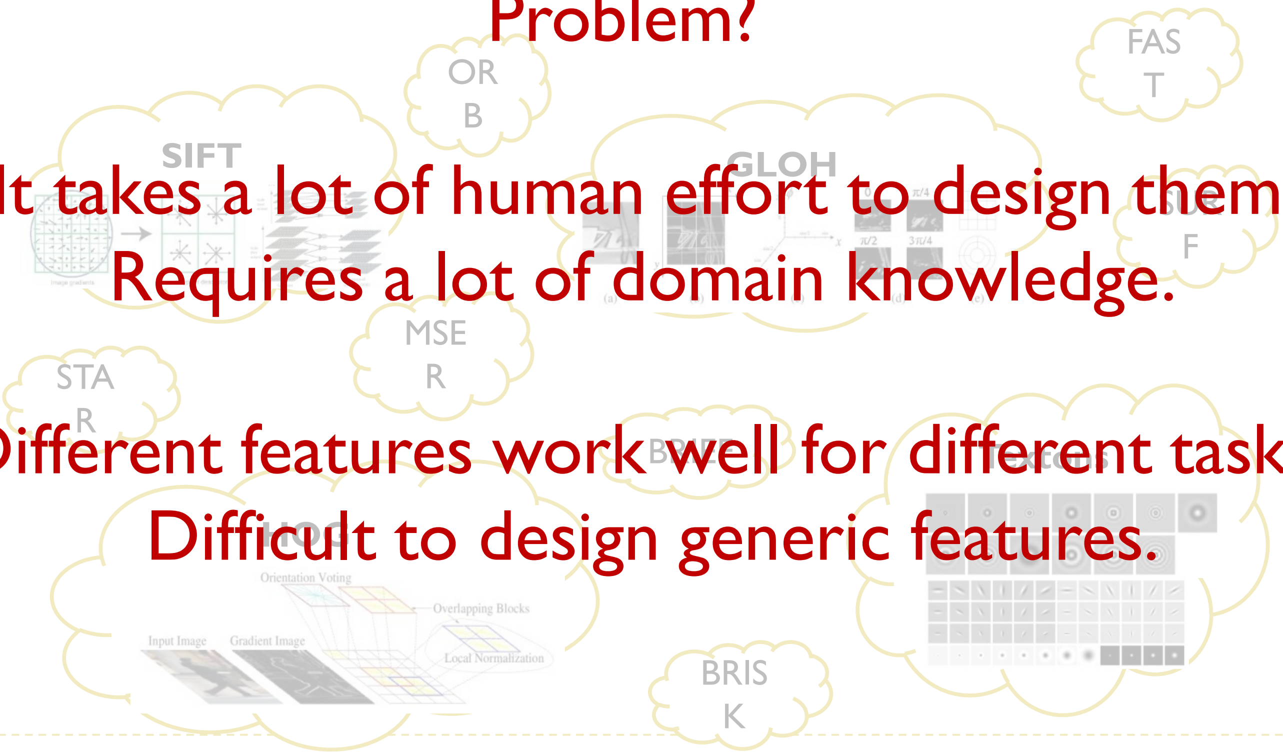
Feature representation in Computer Vision



Hand engineered features in Computer Vision



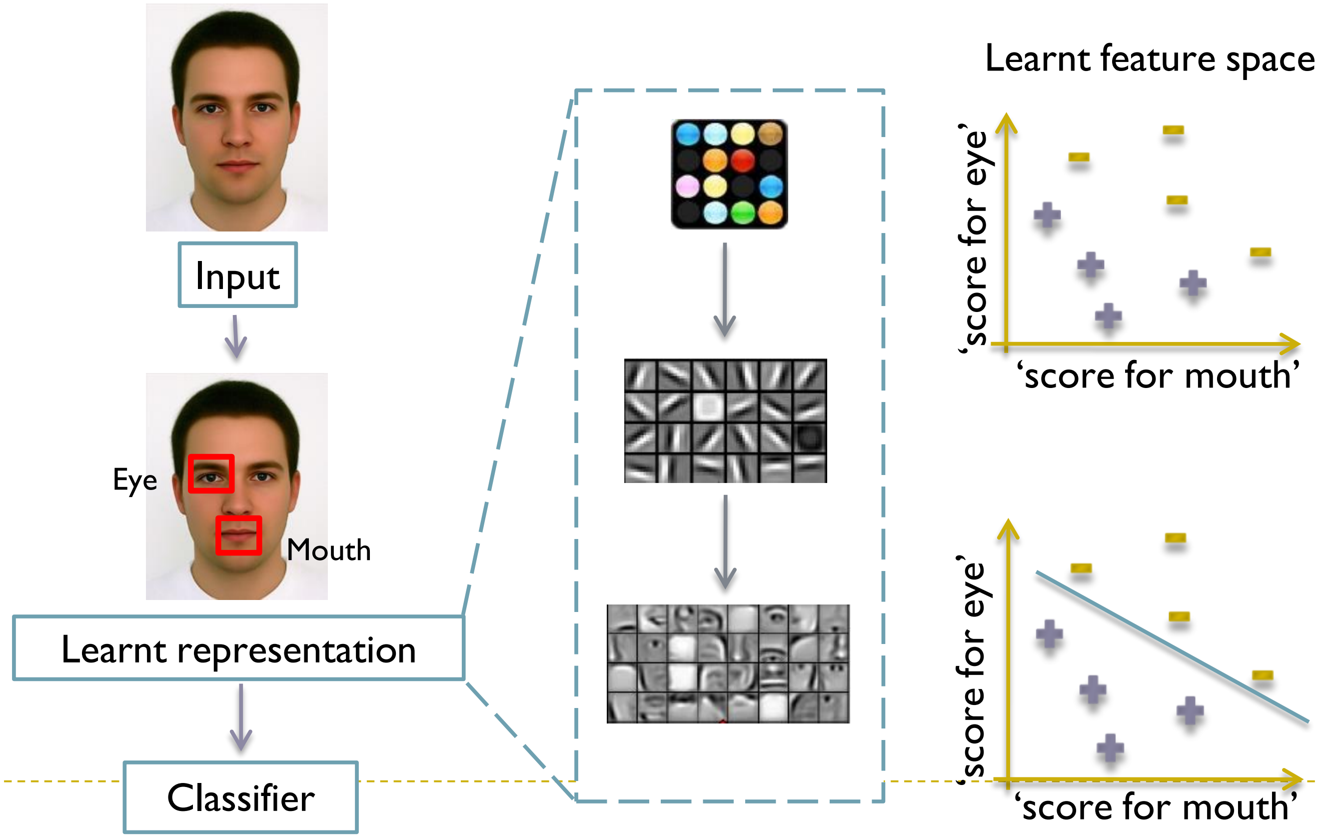
Computer vision features



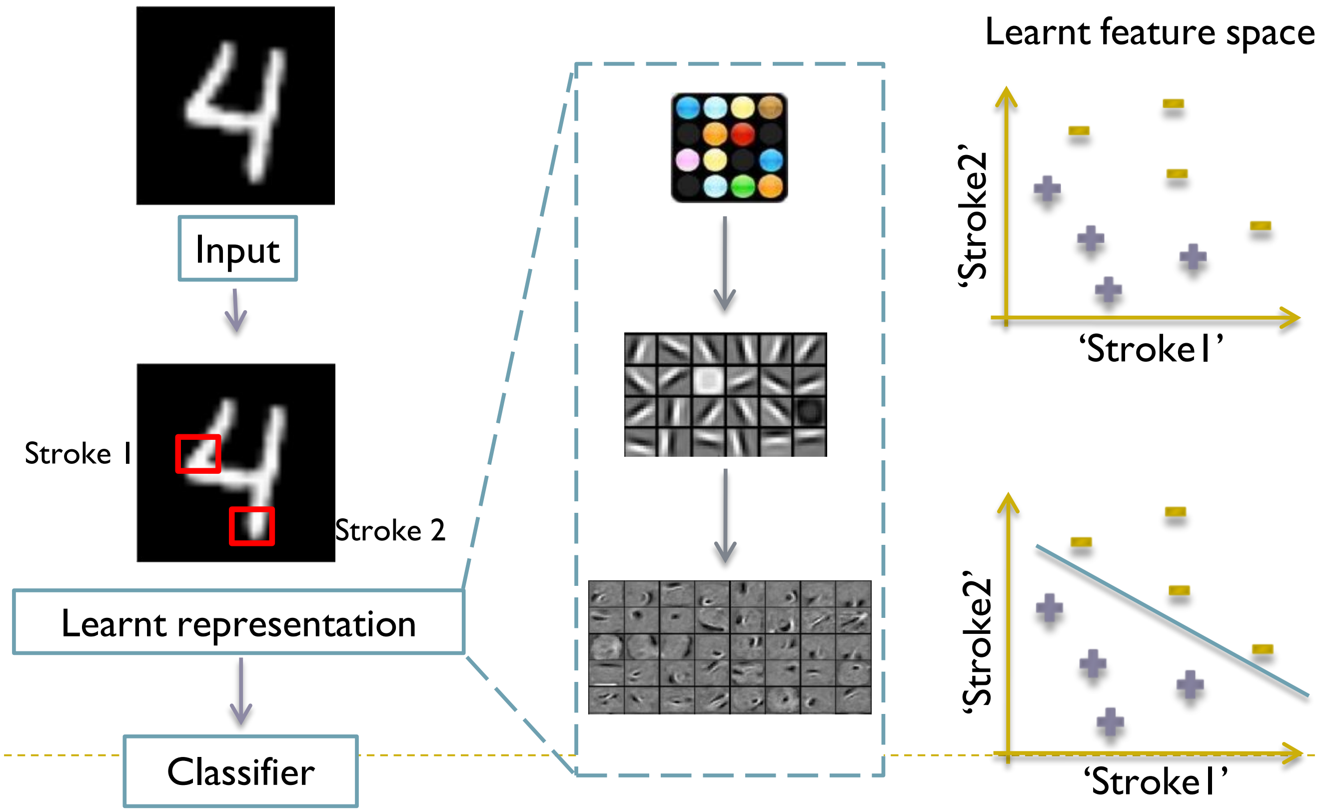
Problem?

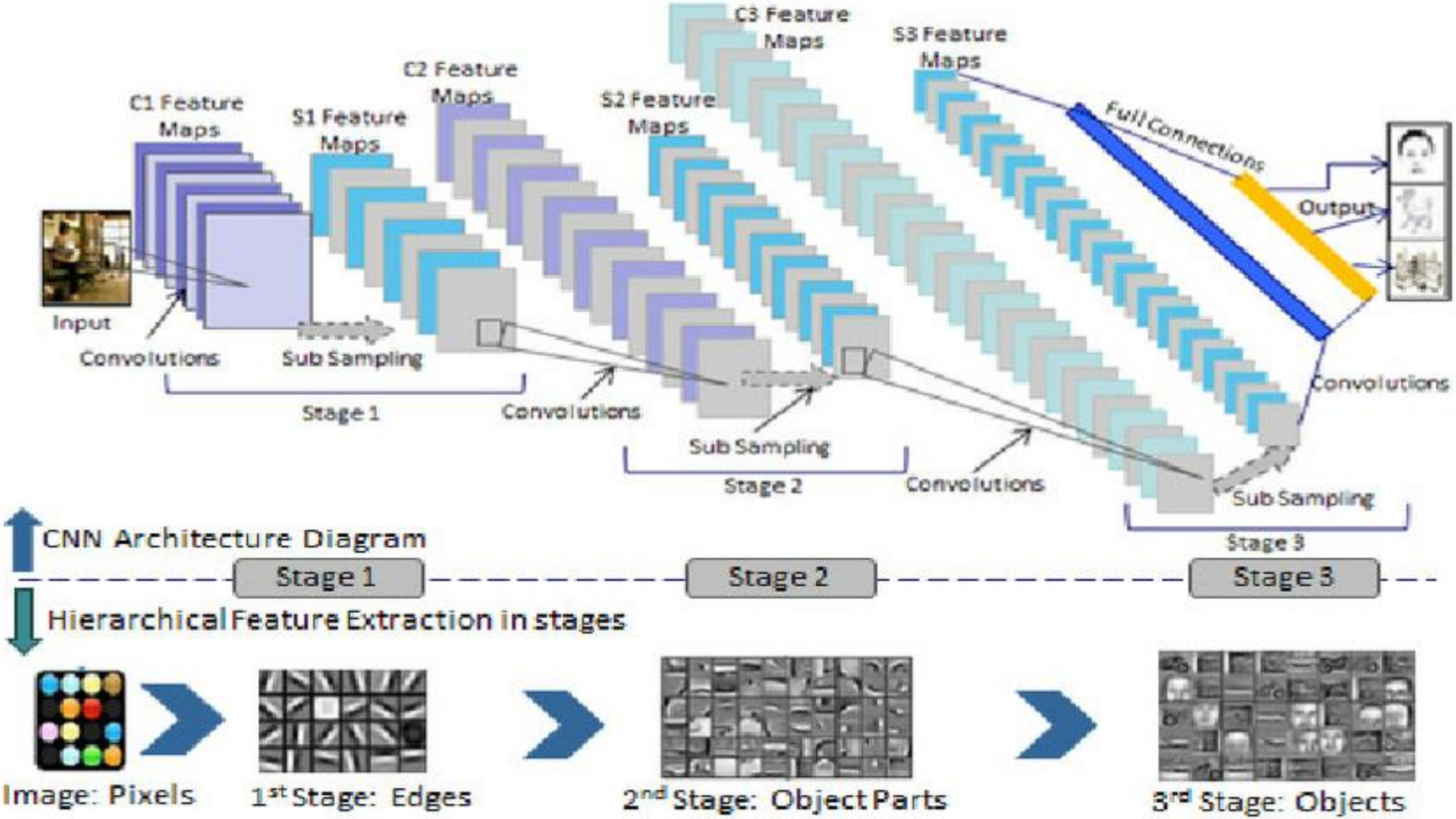
- It takes a lot of human effort to design them. Requires a lot of domain knowledge.
- Different features work well for different tasks. Difficult to design generic features.

Representation Learning



Same framework across different domains




C1 Feature Maps
S1 Feature Maps
C2 Feature Maps
S2 Feature Maps
C3 Feature Maps
S3 Feature Maps
Full Connections
Output
Input
Convolutions
Sub Sampling
Stage 1
Convolutions
Sub Sampling
Stage 2
Convolutions
Sub Sampling
Stage 3
Convolutions
CNN Architecture Diagram
Stage 1
Stage 2
Stage 3
Hierarchical Feature Extraction in stages
Image: Pixels
1st Stage: Edges
2nd Stage: Object Parts
3rd Stage: Objects