12/05/2011

Computer Vision: Summary and Discussion

Computer Vision

CS 143, Brown

James Hays

Many slides from Derek Hoiem

Announcements

- Today is last day of regular class
- Second quiz on Wednesday (Dec 7th)
- Final projects due next Monday (Dec 12th)
- Final presentations next Tuesday (Dec 13th)
  - If you proposed your own final project, you need to prepare a **5 minute** presentation highlighting what you've done.

Today's class

- Review of important concepts

- Some important open problems
  - Especially attribute-based representations

Computer Vision Builds On…

- Image Processing
  - to extract low-level information from images.
- Machine Learning
  - to make decisions based on data.

Fundamentals of Computer Vision

- Geometry
  - How to relate world coordinates and image coordinates
- Matching
  - How to measure the similarity of two regions
- Alignment
  - How to align points/patches
  - How to recover transformation parameters based on matched points
- Grouping
  - What points/regions/lines belong together?
- Categorization / Recognition
  - What similarities are important?

Geometry

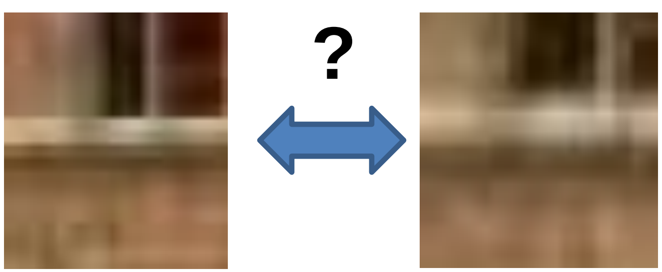
- $\mathbf{x} = \mathbf{K}\,[\mathbf{R}\ \mathbf{t}]\,\mathbf{X}$
  - Maps 3d point $\mathbf{X}$ to 2d point $\mathbf{x}$
  - Rotation $\mathbf{R}$ and translation $\mathbf{t}$ map into 3D camera coordinates
  - Intrinsic matrix $\mathbf{K}$ projects from 3D to 2D

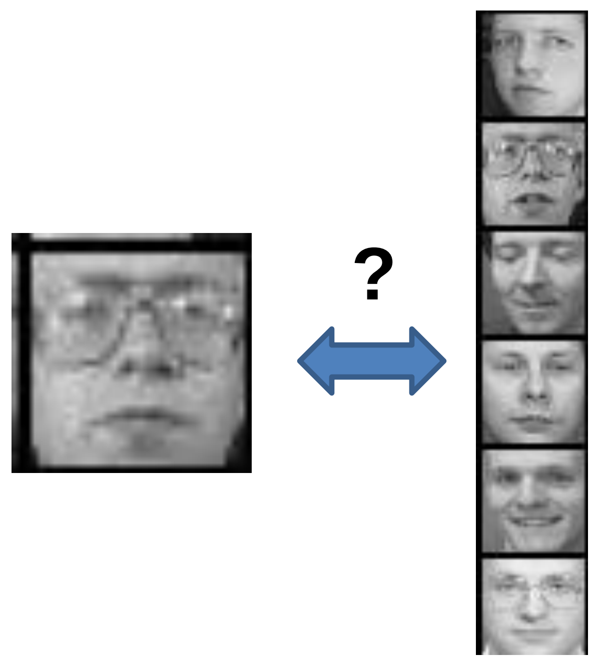
- Parallel lines in 3D converge at the **vanishing point** in the image
  - A 3D plane has a vanishing line in the image

- $\mathbf{x}'^{\mathrm{T}}\,\mathbf{F}\,\mathbf{x} = 0$
  - Points in two views that correspond to the same 3D point are related by the fundamental matrix $\mathbf{F}$

Matching

- Does this patch match that patch?
  - In two simultaneous views? (stereo)
  - In two successive frames? (tracking, flow, SFM)
  - In two pictures of the same object? (recognition)

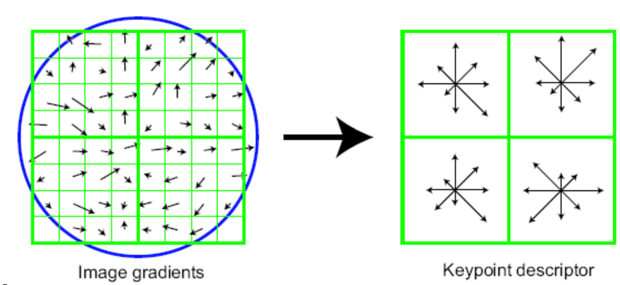




Matching

Representation: be invariant/robust to expected deformations but nothing else

- Often assume that shape is constant
  - Key cue: local differences in shading (e.g., gradients)
- Change in viewpoint
  - Rotation invariance: rotate and/or affine warp patch according to dominant orientations
- Change in lighting or camera gain
  - Average intensity invariance: oriented gradient-based matching
  - Contrast invariance: normalize gradients by magnitude
- Small translations
  - Translation robustness: histograms over small regions



But can one representation do all of this?

- SIFT: local normalized histograms of oriented gradients provides robustness to in-plane orientation, lighting, contrast, translation
- HOG: like SIFT but does not rotate to dominant orientation

Alignment of points

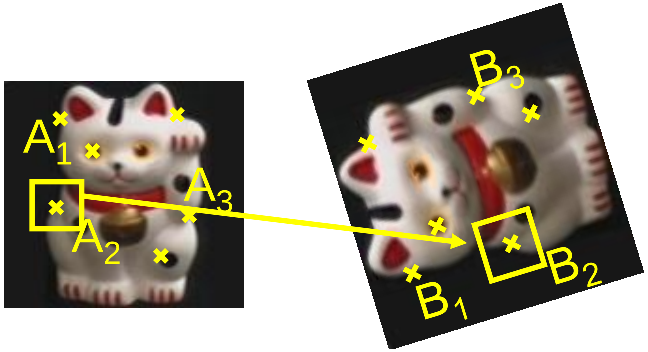
Search: efficiently align matching patches

- Interest points: find repeatable, distinctive points
  - Long-range matching: e.g., wide baseline stereo, panoramas, object instance recognition
  - Harris: points with strong gradients in orthogonal directions (e.g., corners) are precisely repeatable in x-y
  - Difference of Gaussian: points with peak response in Laplacian image pyramid are somewhat repeatable in x-y-scale
- Local search
  - Short range matching: e.g., tracking, optical flow
  - Gradient descent on patch SSD, often with image pyramid
- Windowed search
  - Long-range matching: e.g., recognition, stereo w/ scanline

Alignment of sets

Find transformation to align matching sets of points

- Geometric transformation (e.g., affine)
  - Least squares fit (SVD), if all matches can be trusted
  - Hough transform: each potential match votes for a range of parameters
    - Works well if there are very few parameters (3-4)
  - RANSAC: repeatedly sample potential matches, compute parameters, and check for inliers
    - Works well if fraction of inliers is high and few parameters (4-8)
- Other cases
  - Thin plate spline for more general distortions
  - One-to-one correspondence (Bipartite matching, Hungarian algorithm)



Grouping

- Clustering: group items (patches, pixels, lines, etc.) that have similar appearance
  - Discretize continuous values; typically, represent points within cluster by center
  - Improve efficiency: e.g., cluster interest points before recognition
  - Summarize data

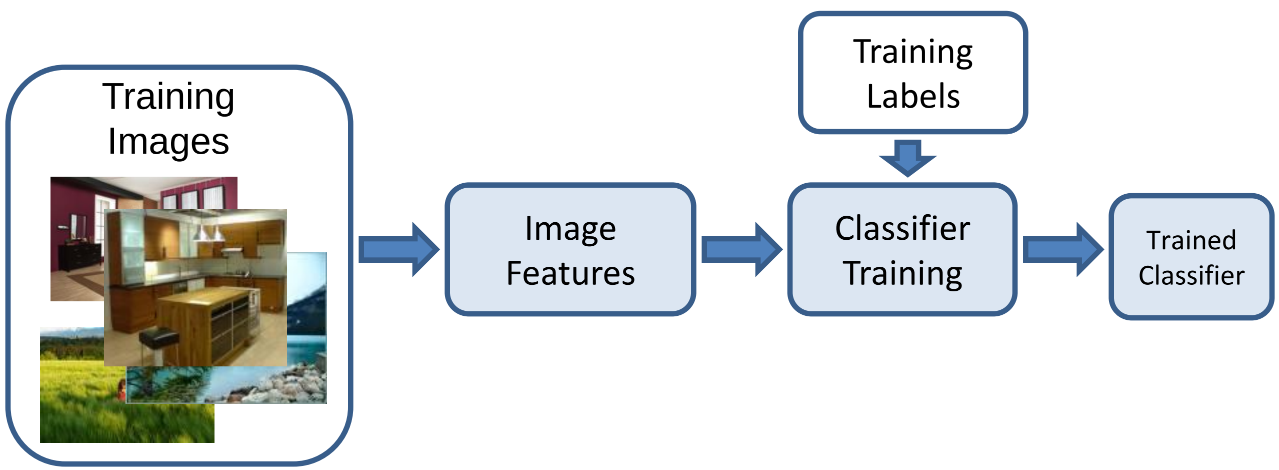
- Segmentation: group pixels into regions of coherent color, texture, motion, and/or label
  - Mean-shift clustering
  - Watershed
  - Graph-based segmentation: e.g., MRF and graph cuts

- EM, mixture models: probabilistically group items that are likely to be drawn from the same distribution, while estimating the distributions' parameters

Categorization

Match objects, parts, or scenes that may vary in appearance

- Categories are typically defined by human and may be related by function, location, or other non-visual attributes
- Key problem: what are important similarities?
  - Can be learned from training examples



Categorization

Representation: ideally should be compact, comprehensive, direct

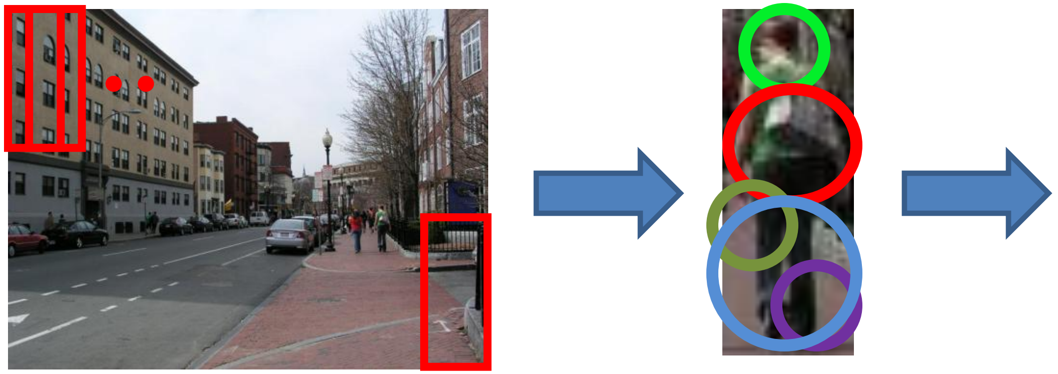
- Histograms of quantized local descriptors (SIFT, HOG), color, texture
  - Typical for image or region categorization
  - Degree of spatial encoding is controllable by using spatial pyramids
- HOG features at specified position
  - Often used for finding parts or objects

Object Categorization

Search by Sliding Window Detector

- May work well for rigid objects

- Key idea: simple alignment for simple deformations



Object or Background?

Object Categorization

Search by Parts-based model

- Key idea: more flexible alignment for articulated objects
- Defined by models of **part appearance**, **geometry** or spatial layout, and **search** algorithm

Vision as part of an intelligent system

3D Scene

Feature Extraction	Texture	Color	Optical Flow	Stereo Disparity	
Grouping	Surfaces	Bits of objects	Sense of depth	Motion patterns	
Interpretation	Objects	Agents and goals	Shapes and properties	Open paths	Words
Action	Walk, touch, contemplate, smile, evade, read on, pick up, …				

Important open problems

Computer vision is potentially worth major $$$, but there are major challenges to overcome first.

- Driver assistance
  - MobileEye received >$100M in funding from Goldman Sachs
- Entertainment (Kinect, movies, etc.)
  - Intel is spending $100M for visual computing over next five years
- Security
  - Potential for billions of deployed cameras
- Robot workers
- Many more

Important open problems

Object category recognition: where is the cat?

Important open problems

Object category recognition: where is the cat?

Important questions:

- How can we better align two object instances?
- How do we identify the important similarities of objects within a category?
- How do we tell if two patches depict similar shapes?

Important open problems

- Spatial understanding: what is it doing? Or how do I do it?

Important open problems

- Spatial understanding: what is it doing? Or how do I do it?

Important questions:

- What are good representations of space for navigation and interaction? What kind of details are important?
- How can we combine single-image cues with multi-view cues?

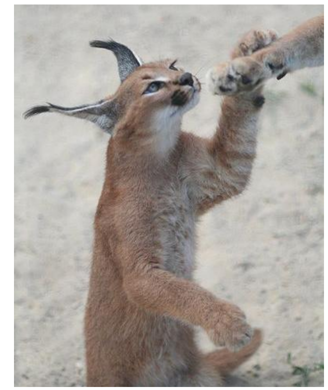
Important open problems

Object representation: what is it?

$\overset{?}{=}$

Important open problems

Object representation: what is it?



Important questions:

- How can we pose recognition so that it lets us deal with new objects?
- What do we want to predict or infer, and to what extent does that rely on categorization?
- How do we transfer knowledge of one type of object to another?

Describing Objects by their Attributes

Ali Farhadi, Ian Endres,
Derek Hoiem, David Forsyth

CVPR 2009


U.S. AIR FORCE

What do we want to know about this object?

What do we want to know about this object?

Object recognition expert: "Dog"

What do we want to know about this object?

Object recognition expert:
“Dog”

Person in the Scene:
“Big pointy teeth”, “Can move fast”, “Looks angry”

Our Goal: Infer Object Properties

Can I **poke with it**?

Can I **put stuff in it**?

Is it **alive**?

What **shape** is it?

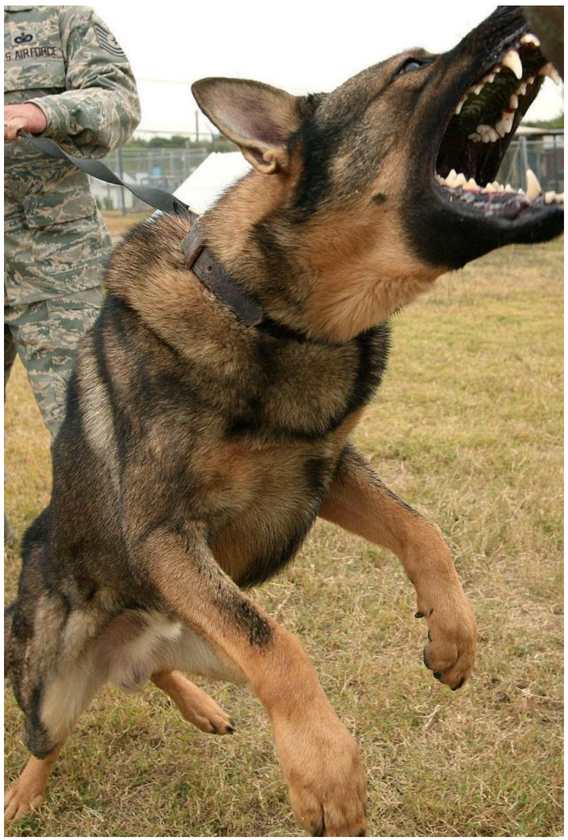
Is it **soft**?

Does it have a **tail**?

Will it **blend**?

Why Infer Properties

1. We want detailed information about objects



"Dog"
vs.
"Large, angry animal with pointy teeth"

Why Infer Properties

2. We want to be able to infer something about unfamiliar objects

Familiar Objects

New Object

Why Infer Properties

2. We want to be able to infer something about unfamiliar objects

If we can infer category names…

Familiar Objects

New Object

Cat

Horse

Dog

???

Why Infer Properties

2. We want to be able to infer something about unfamiliar objects

If we can infer properties…

Familiar Objects

Has Stripes
Has Ears
Has Eyes
….

Has Four Legs
Has Mane
Has Tail
Has Snout
….

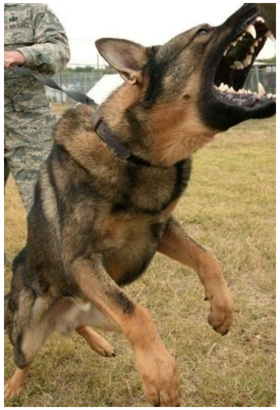
Brown
Muscular
Has Snout
….

New Object

Has Stripes (like cat)
Has Mane and Tail (like horse)
Has Snout (like horse and dog)

Why Infer Properties

3. We want to make comparisons between objects or categories



What is unusual about this dog?


What is the difference between horses and zebras?

Strategy 1: Category Recognition

Object Image

classifier →

Category

"Car"

associated properties →

Has Wheels
Used for Transport
Made of Metal
Has Windows
…

Category Recognition: PASCAL 2008
Category → Attributes: ??

Strategy 2: Exemplar Matching

Object Image

similarity function

Similar Image

associated properties

Has Wheels
Used for Transport
Made of Metal
Old
…

Malisiewicz Efros 2008
Hays Efros 2008
Efros et al. 2003

Strategy 3: Infer Properties Directly

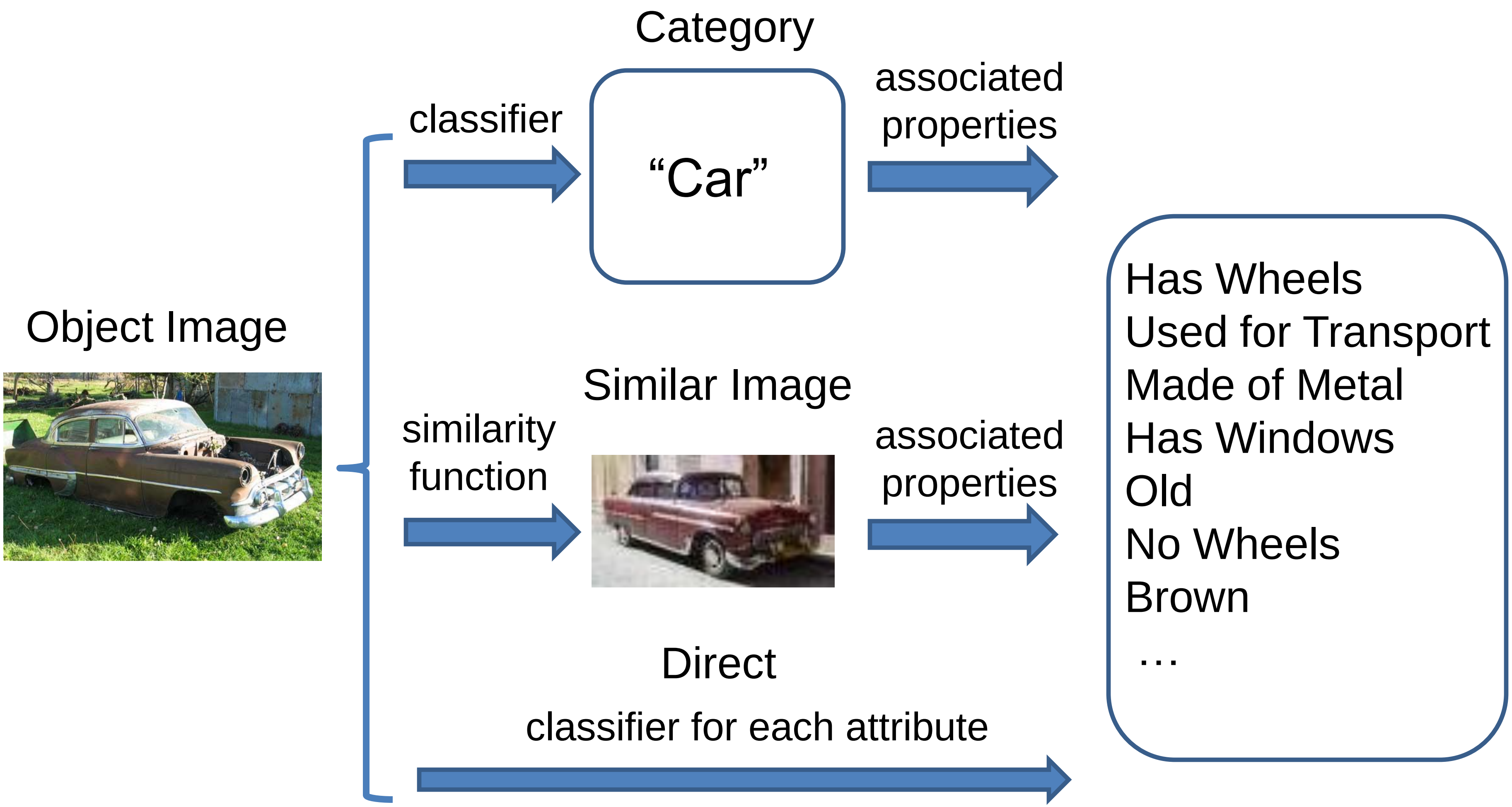
Object Image

classifier for each attribute

No Wheels
Old
Brown
Made of Metal
…

See also Lampert et al. 2009
Gibson's affordances

The Three Strategies



Our attributes

- Visible parts: “has wheels”, “has snout”, “has eyes”
- Visible materials or material properties: “made of metal”, “shiny”, “clear”, “made of plastic”
- Shape: “3D boxy”, “round”

Attribute Examples

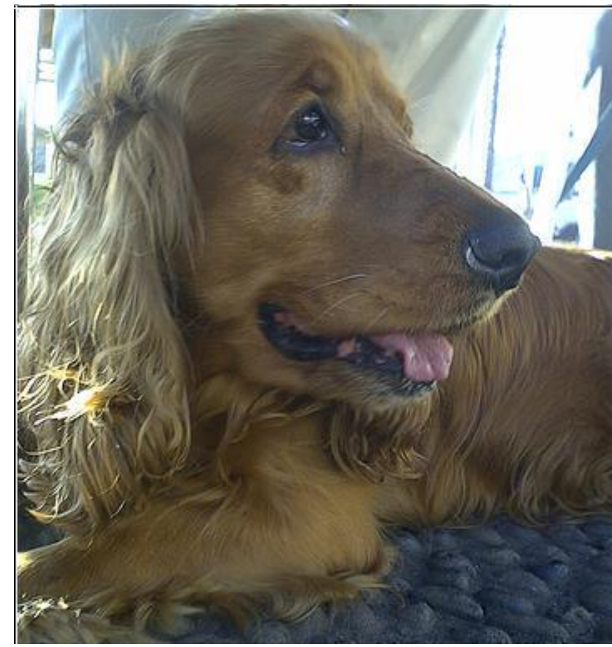
Shape: Horizontal Cylinder
Part: Wing, Propeller, Window, *Wheel*
Material: *Metal*, Glass

Shape:
Part: Window, *Wheel*, Door, Headlight, Side Mirror
Material: *Metal*, Shiny

Attribute Examples

Shape:
Part: Head, Ear, Nose, Mouth, Hair, Face, Torso, Hand, Arm
Material: Skin, Cloth

Shape:
Part: Head, Ear, Snout, Eye
Material: Furry



Shape:
Part: Head, Ear, Snout, Eye, Torso, Leg
Material: Furry

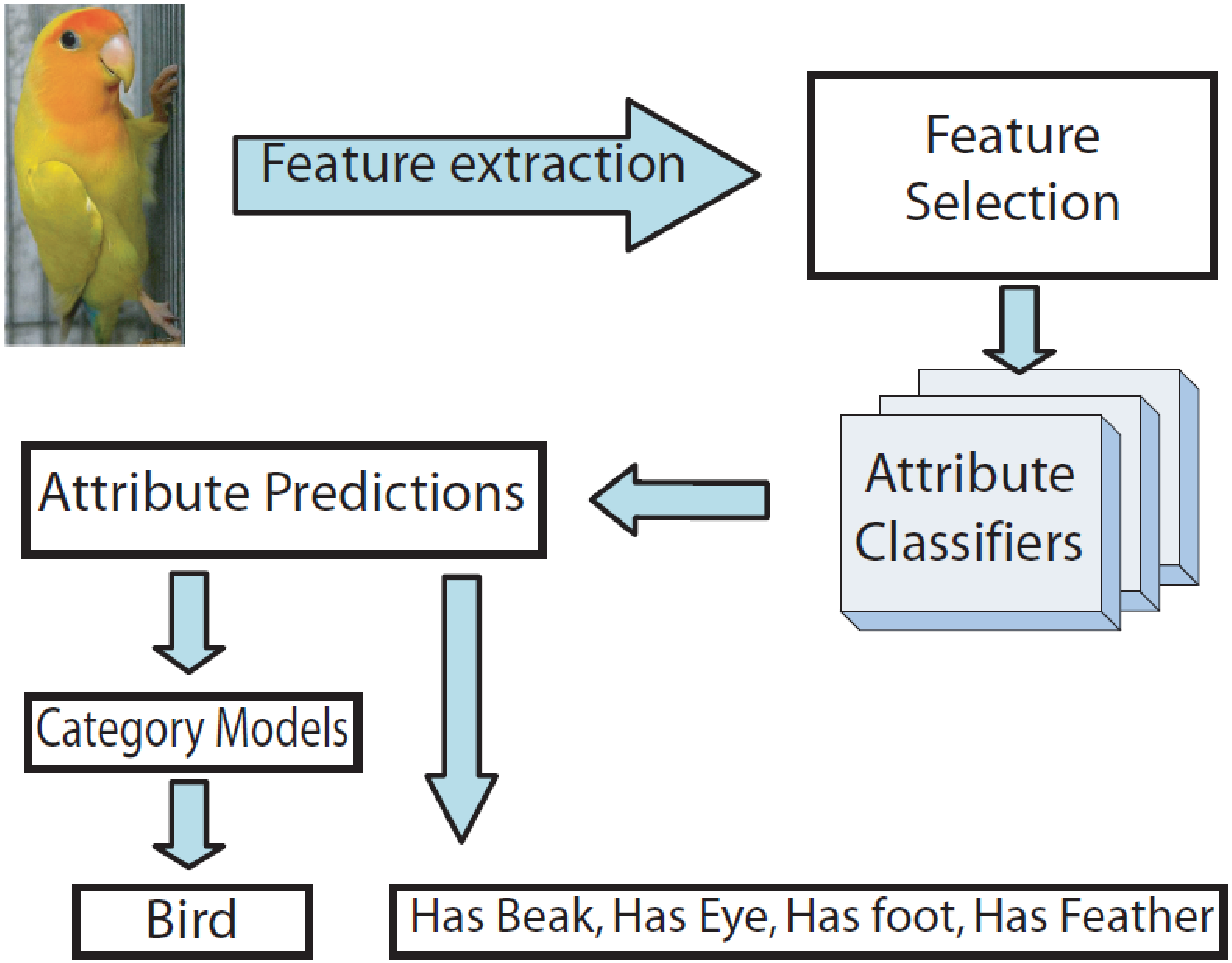
Datasets

- a-Pascal
  - 20 categories from PASCAL 2008 trainval dataset (10K object images)
    - airplane, bicycle, bird, boat, bottle, bus, car, cat, chair, cow, dining table, dog, horse, motorbike, person, potted plant, sheep, sofa, train, tv monitor
  - Ground truth for 64 attributes
  - Annotation via Amazon's Mechanical Turk

- a-Yahoo
  - 12 new categories from Yahoo image search
    - bag, building, carriage, centaur, donkey, goat, jet ski, mug, monkey, statue of person, wolf, zebra
  - Categories chosen to share attributes with those in Pascal

- Attribute labels are somewhat ambiguous
  - Agreement among "experts" 84.3
  - Between experts and Turk labelers 81.4
  - Among Turk labelers 84.1

Our approach



Features

Strategy: cover our bases

- Spatial pyramid histograms of quantized
  - Color and texture for **materials**
  - Histograms of gradients (HOG) for **parts**
  - Canny edges for **shape**

Learning Attributes

- Learn to distinguish between things that have an attribute and things that do not
- Train one classifier (linear SVM) per attribute

Learning Attributes

Simplest approach: Train classifier using all features for each attribute independently

"Has Wheels"

"No Wheels Visible"

Dealing with Correlated Attributes

Big Problem: Many attributes are strongly correlated through the object category

Most things that “have wheels” are “made of metal”

When we try to learn “has wheels”, we may accidentally learn “made of metal”

Has Wheels, Made of Metal?

Experiments

- Predict attributes for unfamiliar objects

- Learn new categories
  - From limited examples
  - Learn from verbal description alone

- Identify what is unusual about an object

- Provide evidence that we really learn intended attributes, not just correlated features

Describing Objects by their Attributes

No examples from these object categories were seen during training

Describing Objects by their Attributes

No examples from these object categories were seen during training

Identifying Unusual Attributes

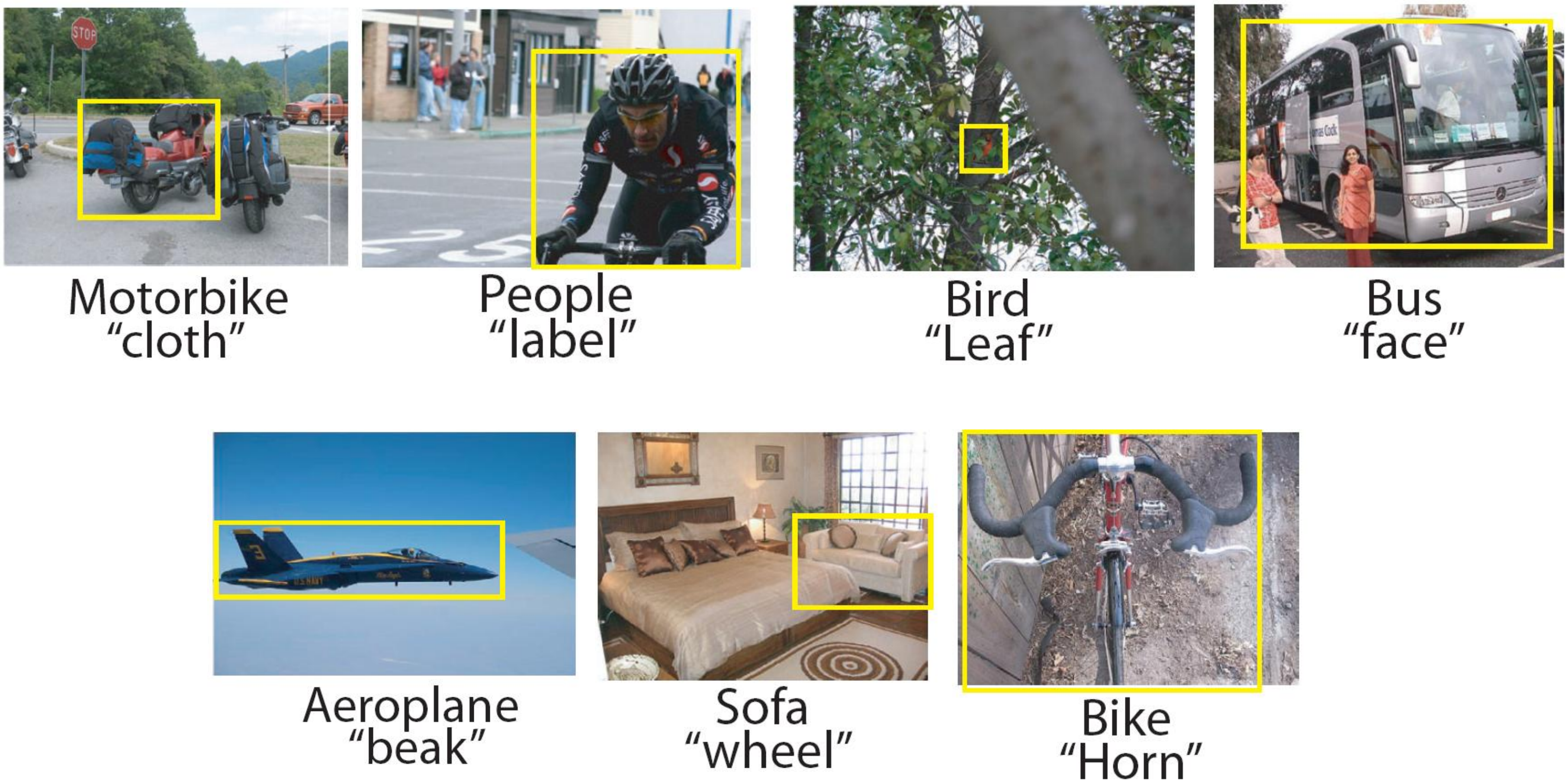
- Look at predicted attributes that are not expected given class label

Absence of typical attributes

752 reports

68% are correct

Presence of atypical attributes



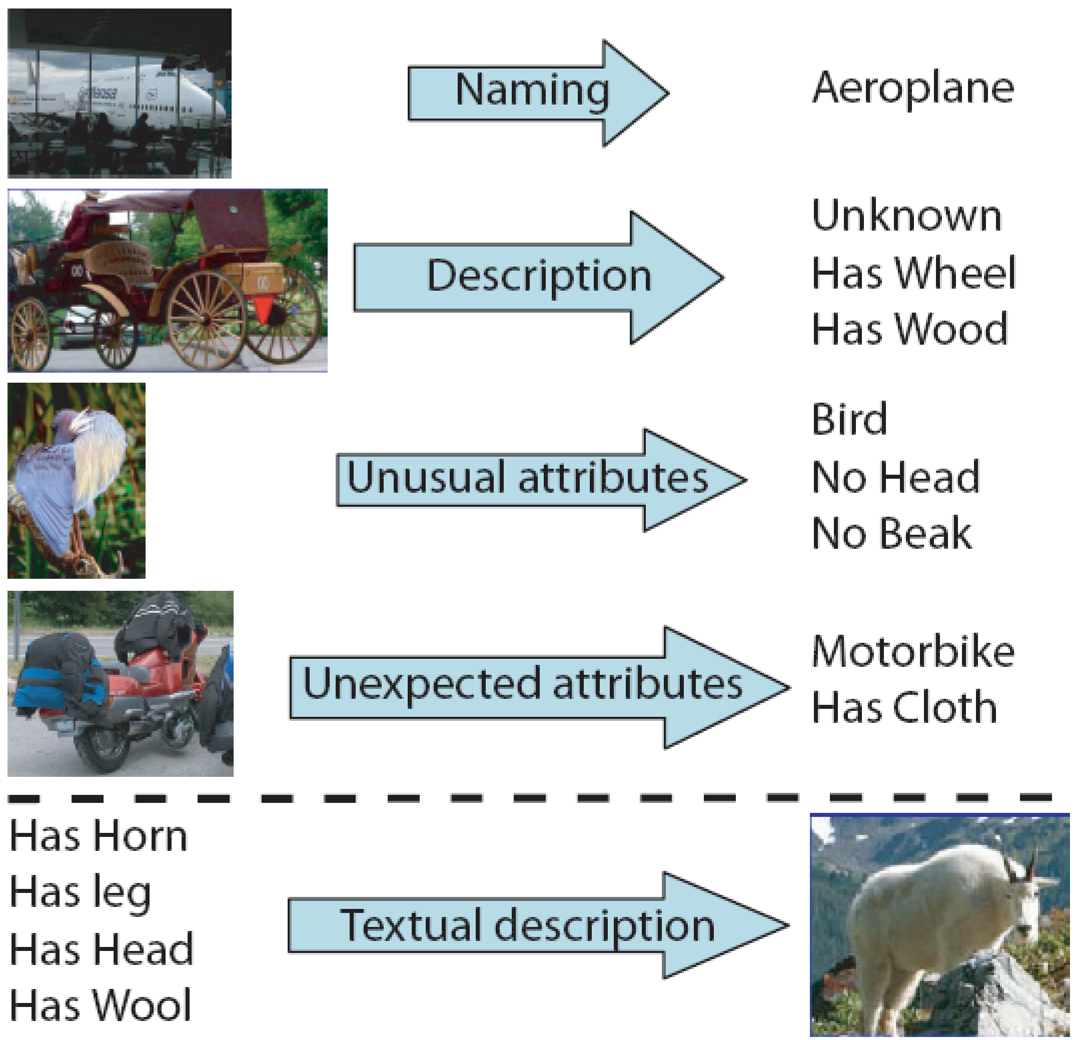
951 reports

47% are correct

Conclusion

- Inferring object *properties* is the central goal of object recognition
  - Categorization is a means, not an end
- We have shown that a special form of feature selection allows better learning of intended attributes
- We have shown that learning properties directly enables several new abilities
  - Predict properties of new types of objects
  - Specify what is unusual about a familiar object
  - Learn from verbal description
- Much more to be done

Thank you



A. Farhadi, I. Endres, D. Hoiem, D.A. Forsyth, "Describing Objects by their Attributes", CVPR 2009

Back to the big picture…

If you want to learn more…

- Read lots of papers: IJCV, PAMI, CVPR, ICCV, ECCV, NIPS, ICCP
- Related Classes
  - CS 2951B – Data-driven Vision and Graphics (Spring ‘12)
  - CS 1950F – Intro. Machine Learning (Spring ‘12)
  - ENGN 2520 – Pattern Rec. and Machine Learning (Spring ’12)
  - ENGN 2502 – 3d Photography (Spring ‘12)
  - CS 123 – Computational Photography (Fall ‘12)
- Just implement stuff, try demos, see what works