DESCRIBING OBJECTS WITH ATTRIBUTES

Visual recognition course

By Yarin Didi & Guy Ezer

DESCRIBING OBJECTS BY THEIR ATTRIBUTES

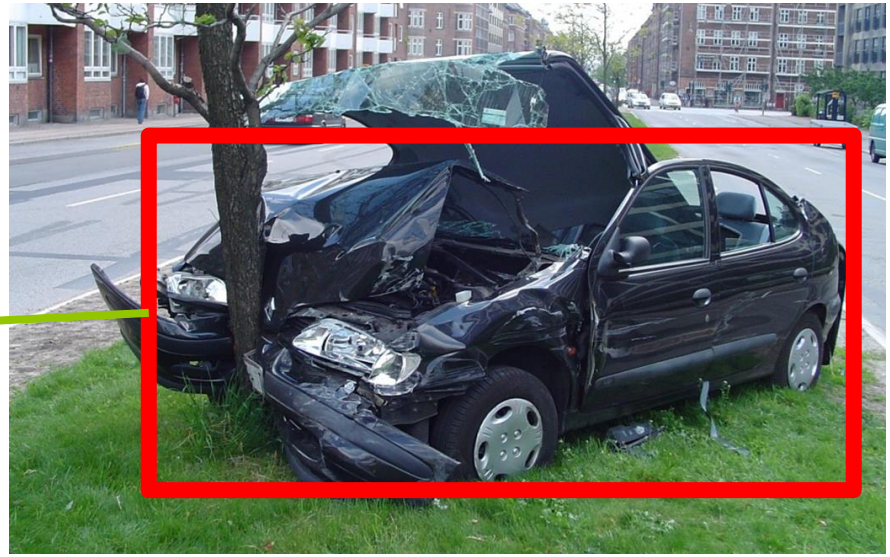
Concept – simplify recognition

- Reduction the problem of recognition to smaller and more simple tasks.

Problem – recognition lack with details

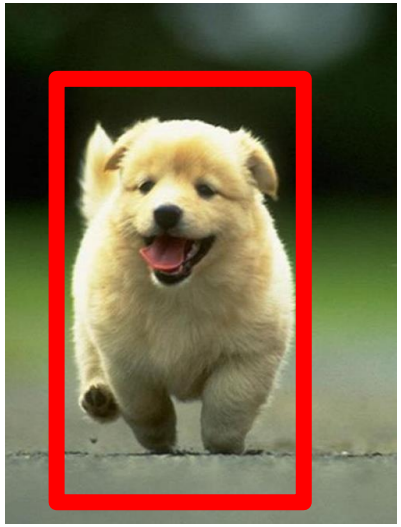
- Naming objects alone is not very informative.

car



What do we want to know?

- recognition program : is a dog .
- Human in the scene:
  - “puppy”
  - “running”
  - “move forward”
  - “looks happy and calm”
  - “bright”



Attributes – infer properties

Can I **poke with it**?

Is it **alive**?

What **shape** is it?

Does it have a **tail**?

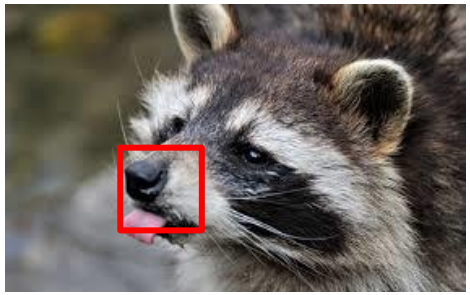
Can I **put stuff in it**?

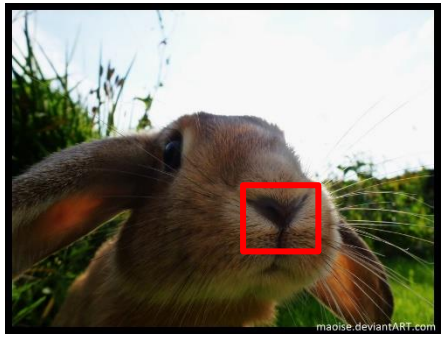
Is it **soft**?

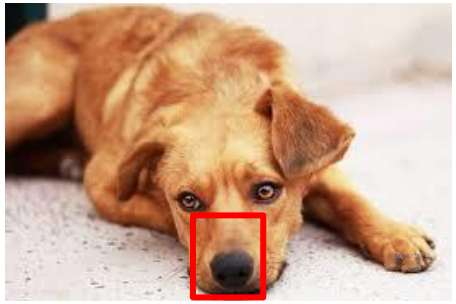
Will it **blend**?

Principle - General attribute

- Attributes might and should be shared with relevant categories.



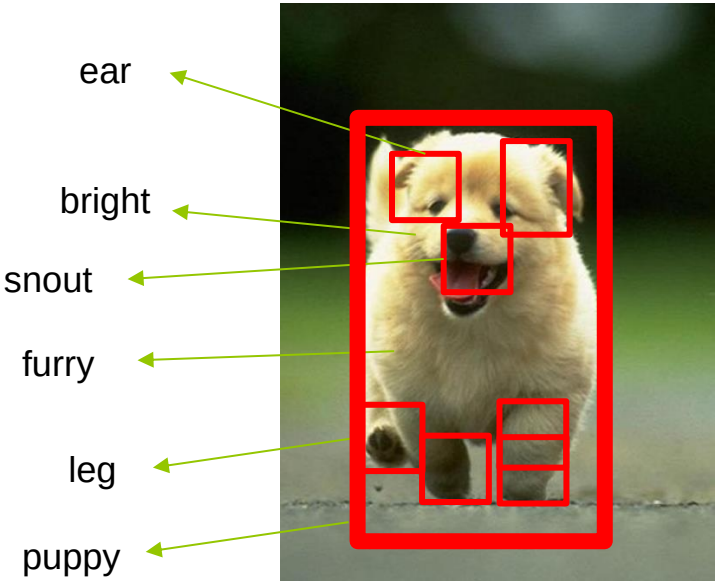




Snout should be shared by all relevant classes

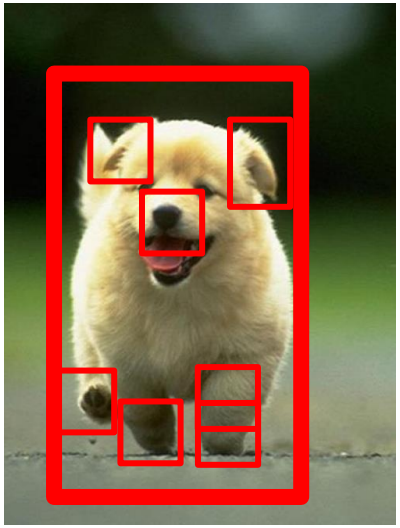
What do we want to know?

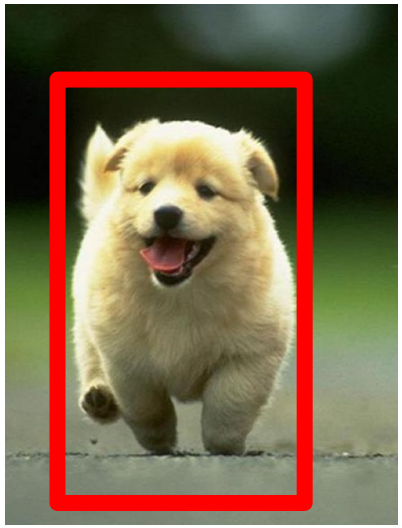
- Attributes formation:
  - is alive
  - Has four legs
  - Has ears
  - Texture: Is furry
  - Color : #yellow
  - Has snout



Motivation - detailed information

Lets us to perform some more complex queries.

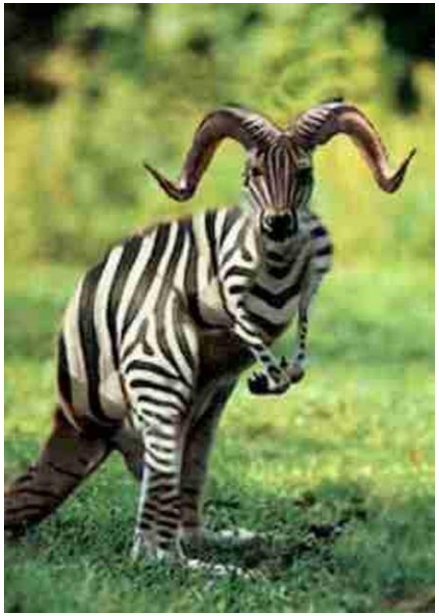




Detailed information	Recognition
is alive Has four legs Has ears Texture: Is furry Color : #yellow Has snout Bright puppy running move forward looks happy and calm	Dog/puppy

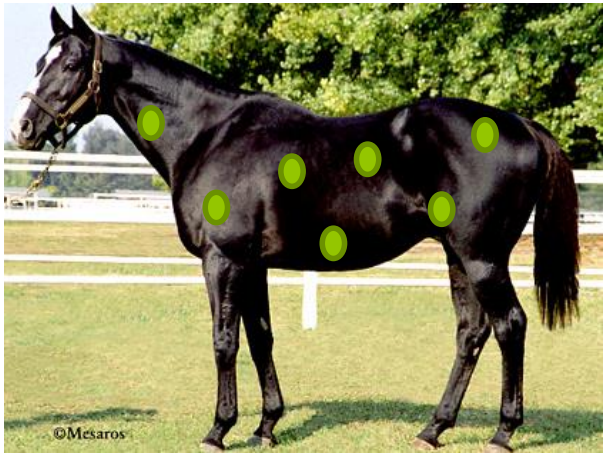
Motivation – describe unknown object

- The ability to infer some properties of unknown abject is better than conclude nothing.

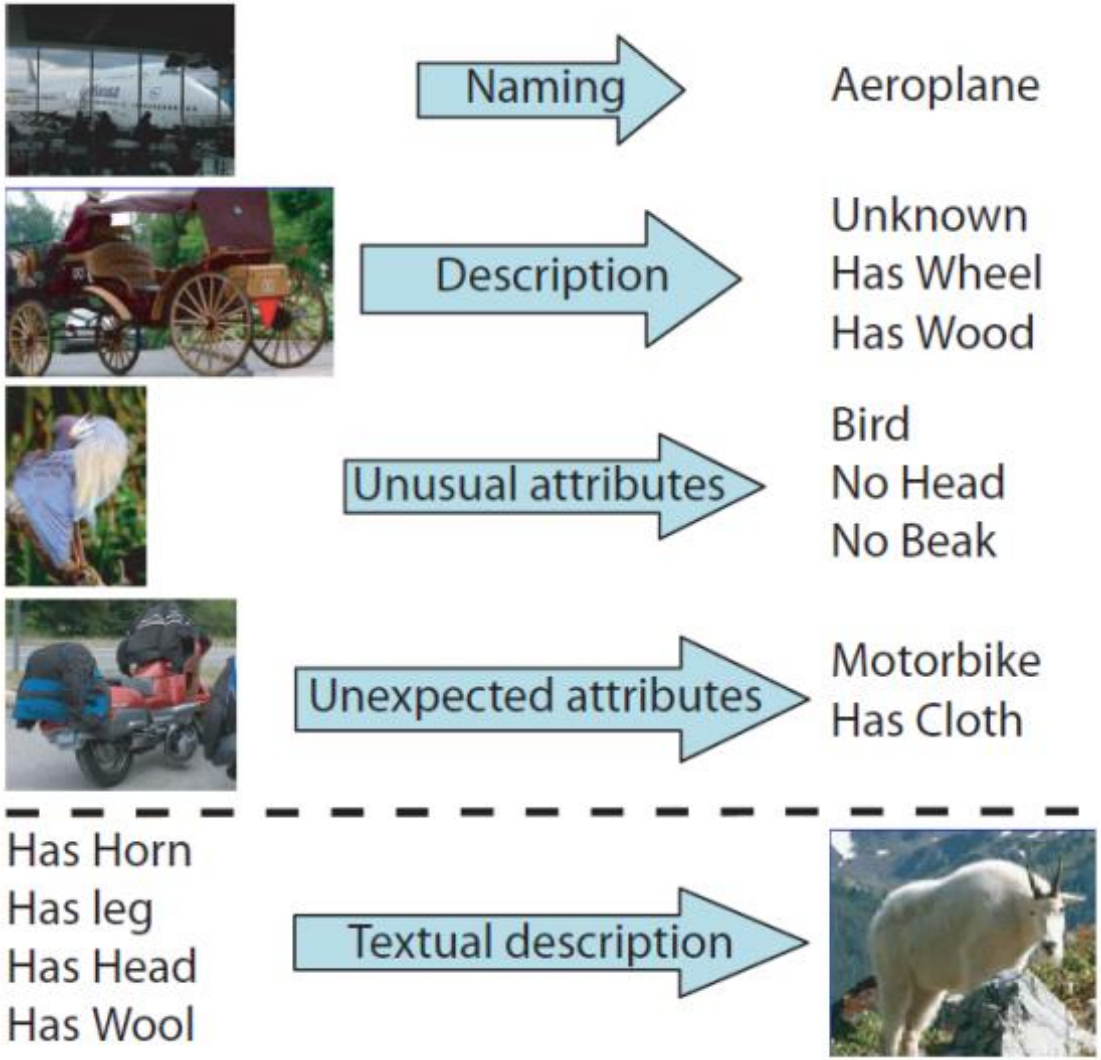


Information	Zero information
Has horn Has snout Has strips Has tail Has two legs Has hands Has eyes standing	unknown

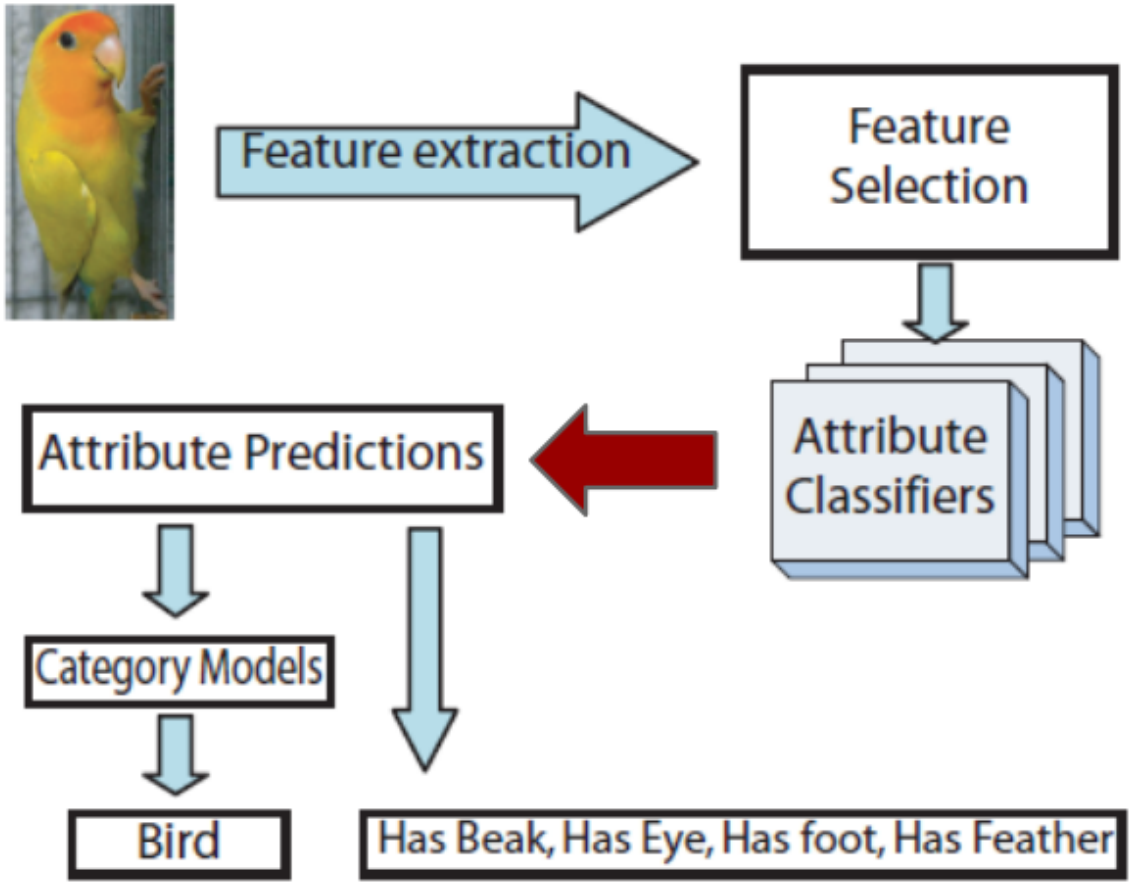
Motivation - report about unusual things



Abilities


Naming
Aeroplane
Description
Unknown
Has Wheel
Has Wood
Unusual attributes
Bird
No Head
No Beak
Unexpected attributes
Motorbike
Has Cloth
Has Horn
Has leg
Has Head
Has Wool
Textual description

Scheme

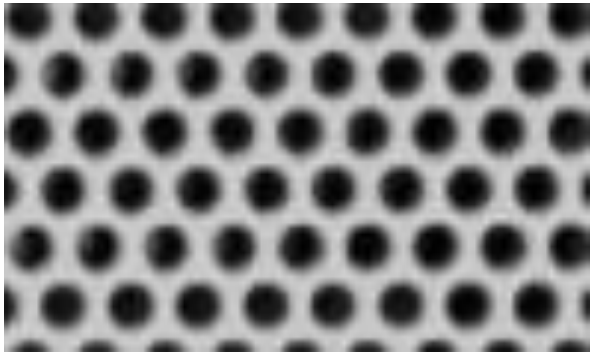

Feature extraction
Feature Selection
Attribute Classifiers
Attribute Predictions
Category Models
Bird
Has Beak, Has Eye, Has foot, Has Feather

Semantic attribute

Colors

Parts

Textures



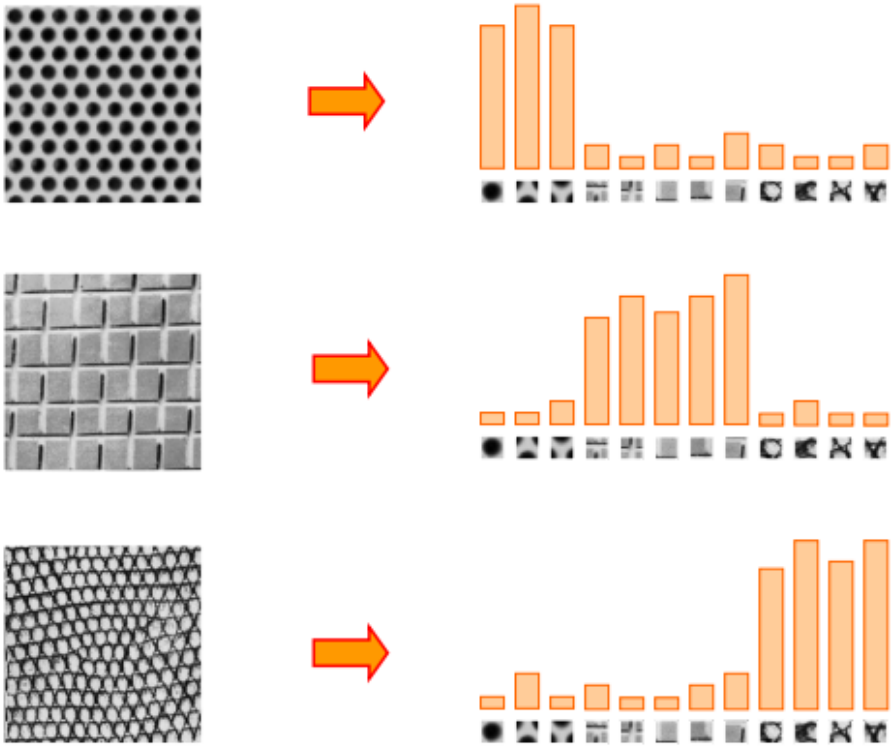
Shape

Semantic attributes

- **Bag of words for each:**
  - Texture descriptors – (for materials)
    - Computed for each pixel and quantized to NN 256 kmeans, extracted with a texton filterbank.
  - Visual words – (parts)
    - HOG spatial pyramid quantized to NN 1000 kmeans.
  - Edges –
    - standard canny edge detector.
  - Color descriptors – (materials)
    - Sampled for each pixel, and quantized to the NN 128 kmeans .

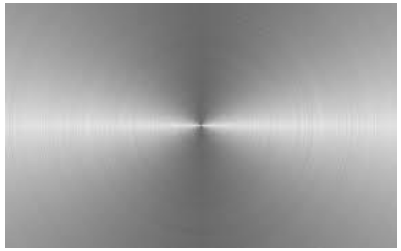
Bag of words – reminder

- 1. sampling “images”
- 2. learning a dictionary - kmeans
- 3. test image

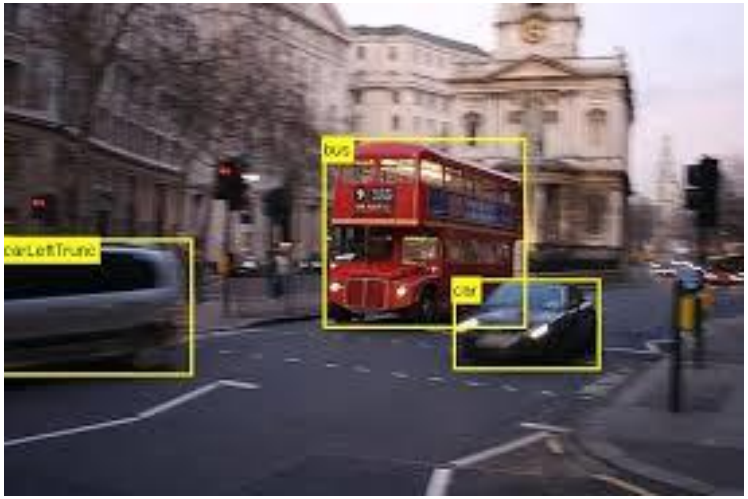


Correlated attribute problem

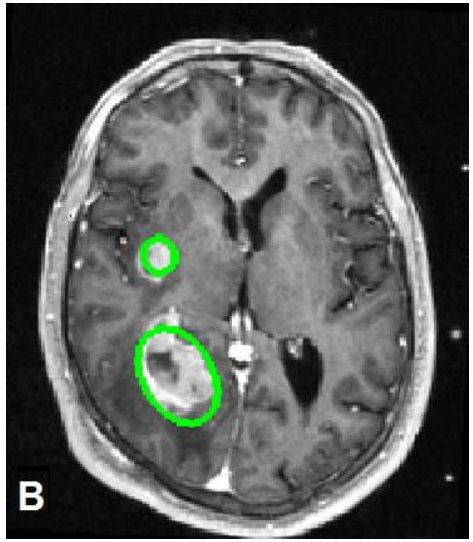
- The classifier might be accidentally learn correlated feature instead the feature.



Bounding box problem



Bounding-box might influence with irrelevant issues



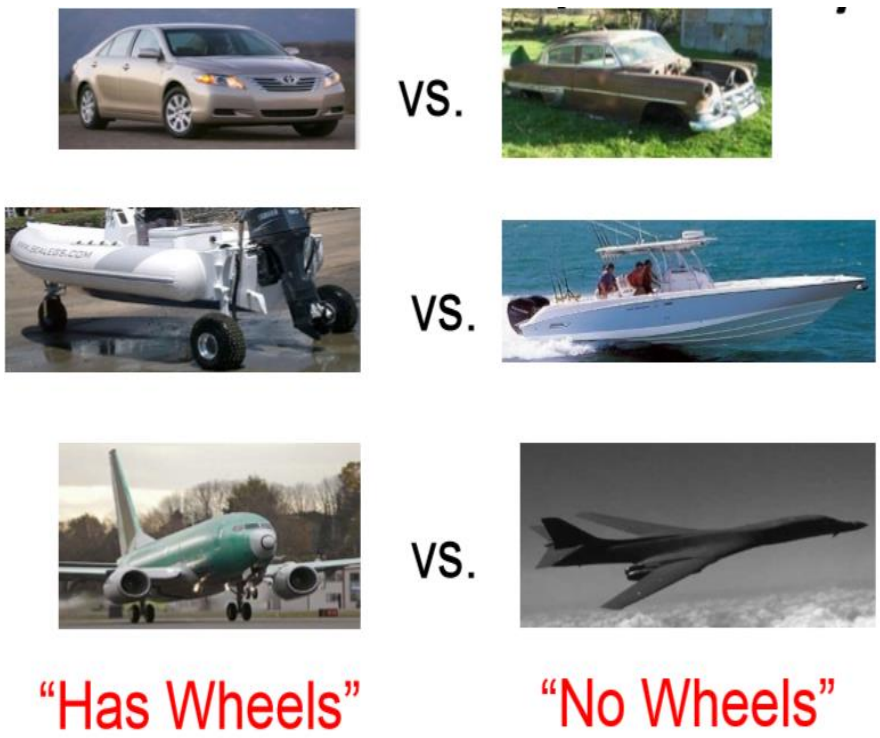
One possible solution is appropriate segmentation

Category Generalization - Feature selection

- Conventional feature selection will not apply to the problem because confused with semantically irrelevant correlation.
- Instead of choosing all features we will choose a subset of it that perform well distinguishing.

Deal with correlation

- select features that perform well at distinguishing examples of cars with and without wheel.
- Run same procedure to for bikes, motorbikes, boats…
- Learn the “wheel” classifier with the selected features.



Correlation between ground - truth “wheel” and metallic”

correlation between ground-trues and “wheel” and metallic”

PASCAL	Yahoo
0.71	0.17

Training with on **PASCAL**

Testing on Yahoo

Whole features	Selection features
0.56	0.28

We might notice that the generalizing worked because it

Influence of feature selection

	With feature selection	Without feature selection
learning from textual description	25.2%	32.5%
reporting the absence of typical attributes	54.8%	68.2%
presence of atypical attributes	24.5%	47.3%

Localization using a grid

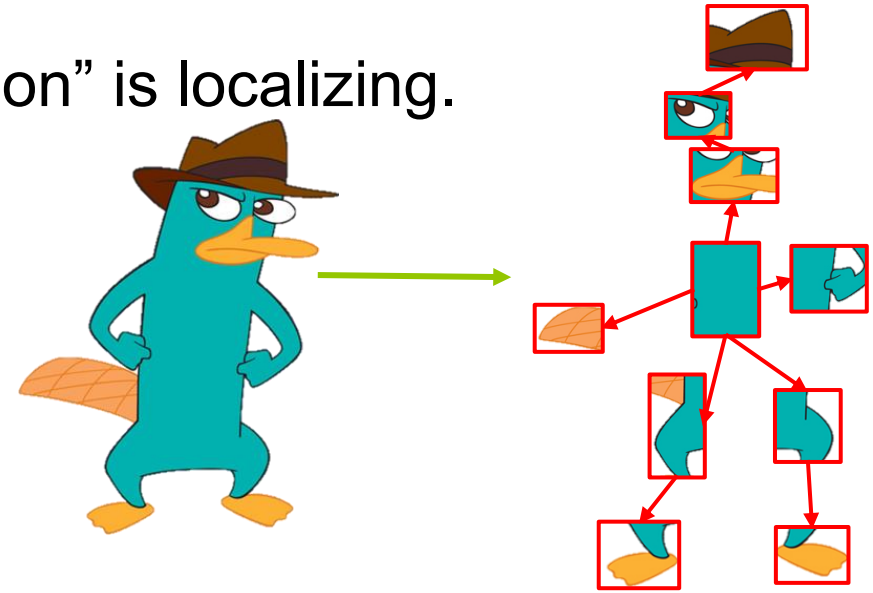
Furniture-leg

Furniture-leg tend to appear In the lower part of the image.

Differences from parts based model

- The parts based model is not flexible to other classes because it used the latent parts on an object.
- This paper define “parts” which can be shared with other classes.
- Side effect of “feature selection” is localizing.

$$L^* = \arg\min_L \left(\sum_{i=1}^{n} m_i(l_i) + \sum_{(v_i,v_j)\in E} d_{ij}(l_i, l_j) \right)$$



Datasets

- a-Pascal
  - – 20 categories from PASCAL 2008 train 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
- 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

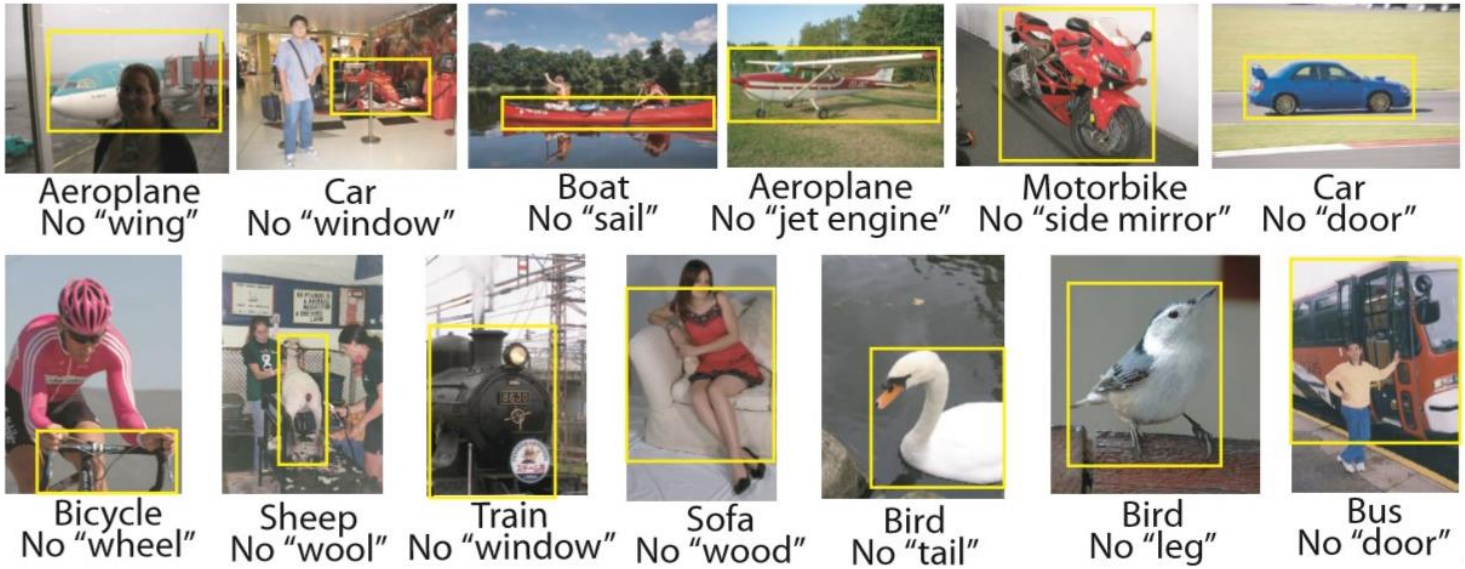
Predicting attributes

- Train on 20 object classes from a-Pascal train set
  - Feature selection for each attribute
  - Train a linear SVM classifier
- Test on 12 object classes from Yahoo image search (cross-category) or on a-Pascal test set (within-category)
  - Apply learned classifiers to predict each attribute

Experiments : description

Experiments : naming - absence of typical attributes

752 reports 68% are correct



Experiments : presence of atypical attributes

951 reports 47% are correct

Experiments : localization of attributes

Feature selection is necessary to localize attributes.

Experiments : localization of atypical attributes

Experiments : localization using selected features

Row of windows

Flower

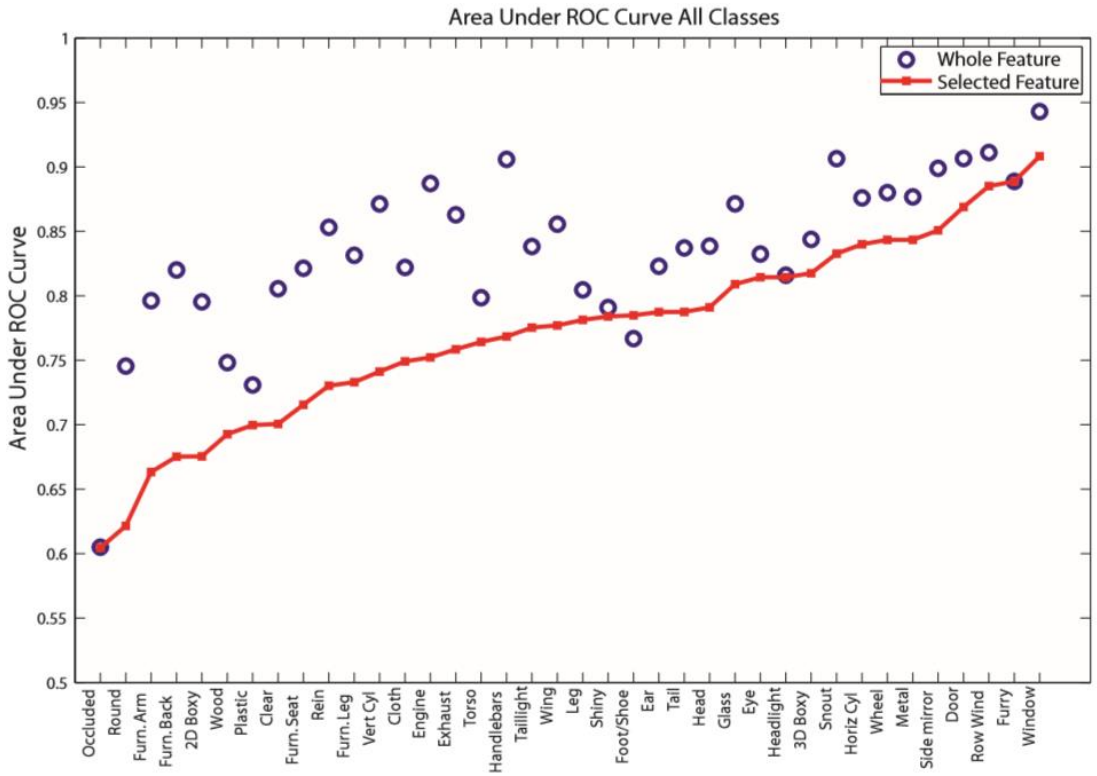
Snout

Furniture Back

Experiments – naming – directly

PASCAL 08	Base Features	Whole Features		Selected Features	
		Sem. Attr.	All Attr.	Sem. Attr.	All Attr.
SVM	58.5 (35.5)	56.1 (34.3)	58.3 (38.1)	54.6 (28.4)	**59.4** (37.7)
Logistic Regression	54.6 (36.9)	51.2 (31.4)	53.4 (33.5)	51.8 (32.3)	53.5 (35.1)

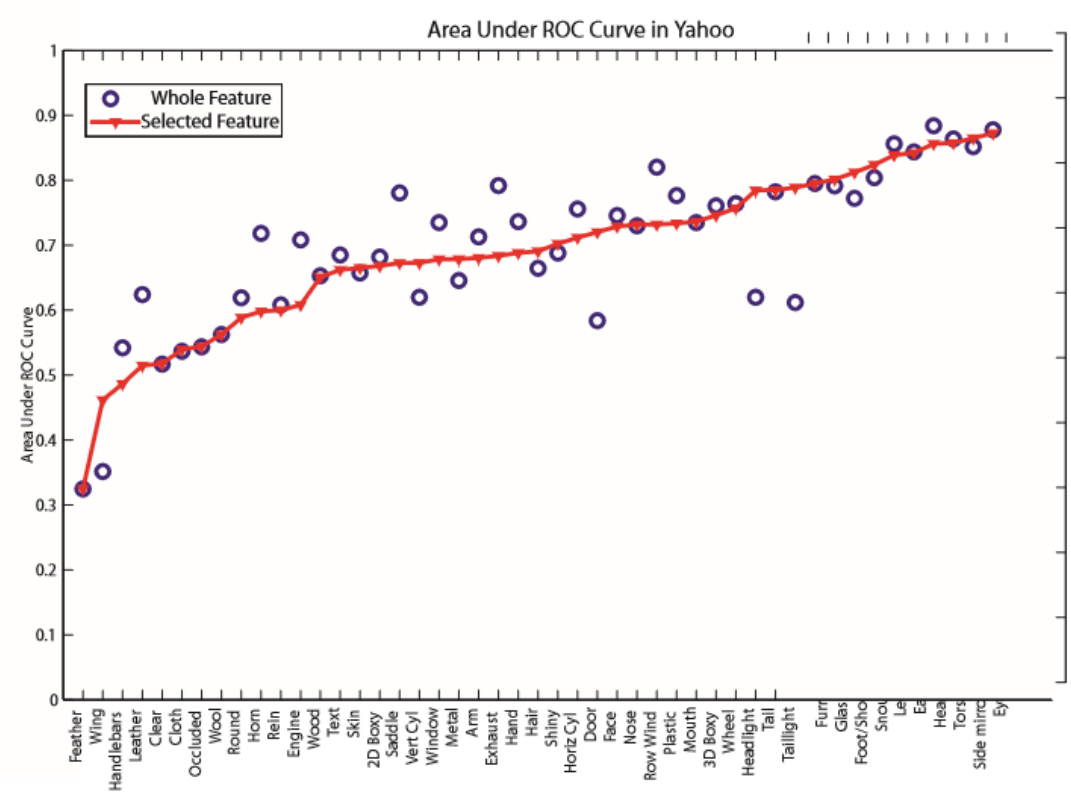
Attribute prediction



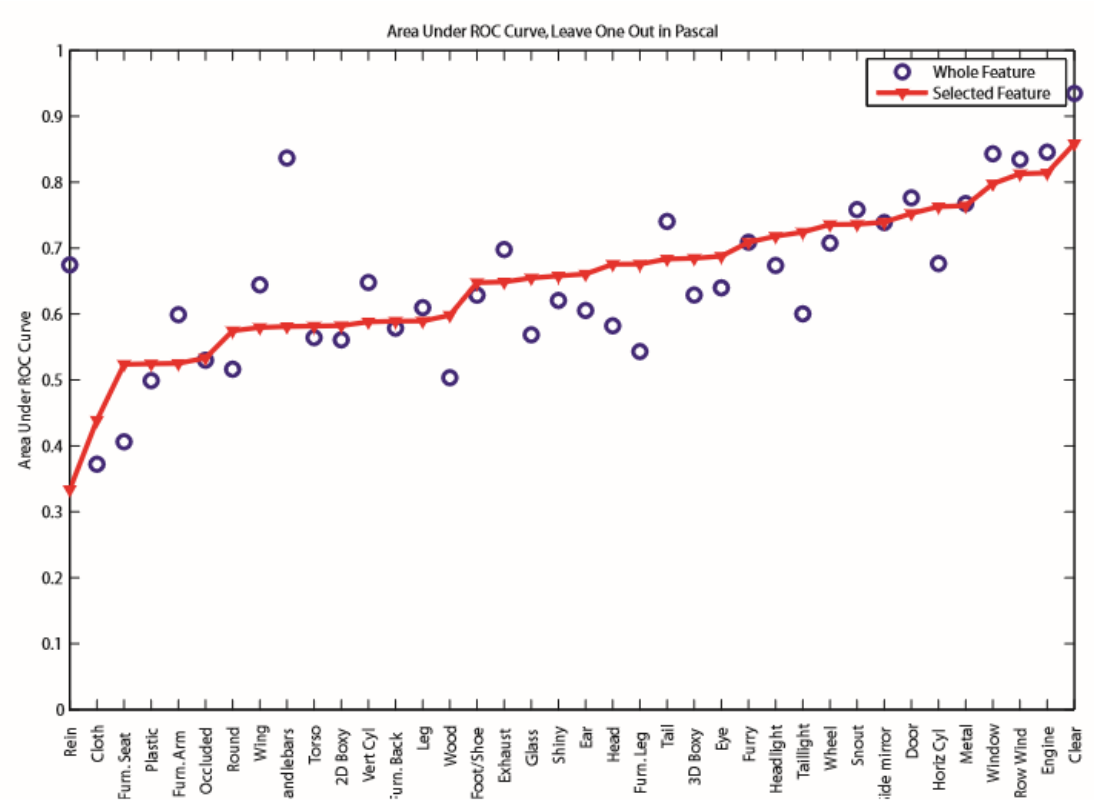
- Attribute prediction for attribute classifiers trained on a-Pascal and tested on a-Pascal
- This means that the attribute statistics are not changing during training and testing. So generalize don't need.

- Good
  - Eye, Side, head, ear, snout
- Mid
  - Plastic, 3/2 D boxy, Metal, Shiny
- Bad
  - Feather, Wing, Horn, close

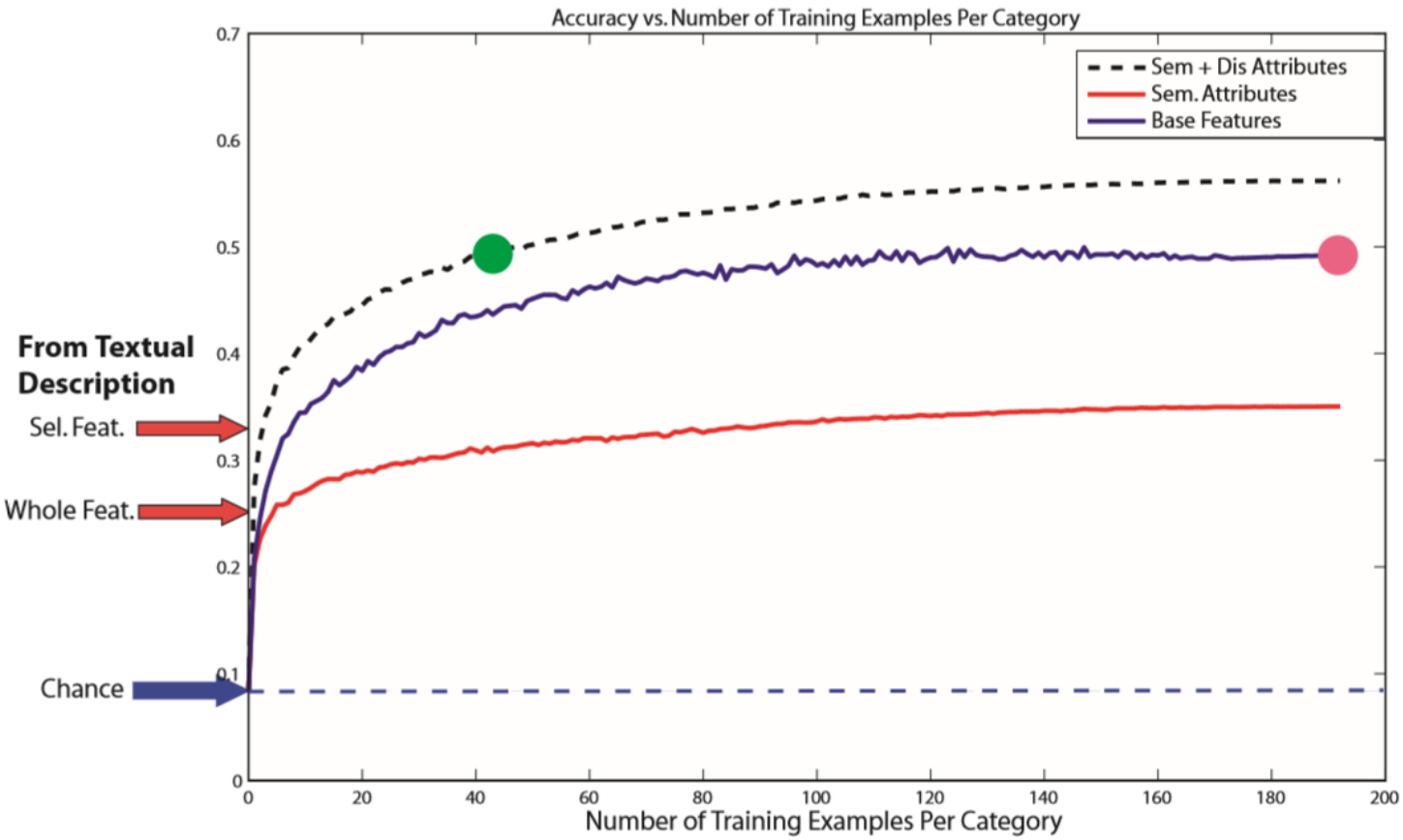
Attribute prediction - across category protocols

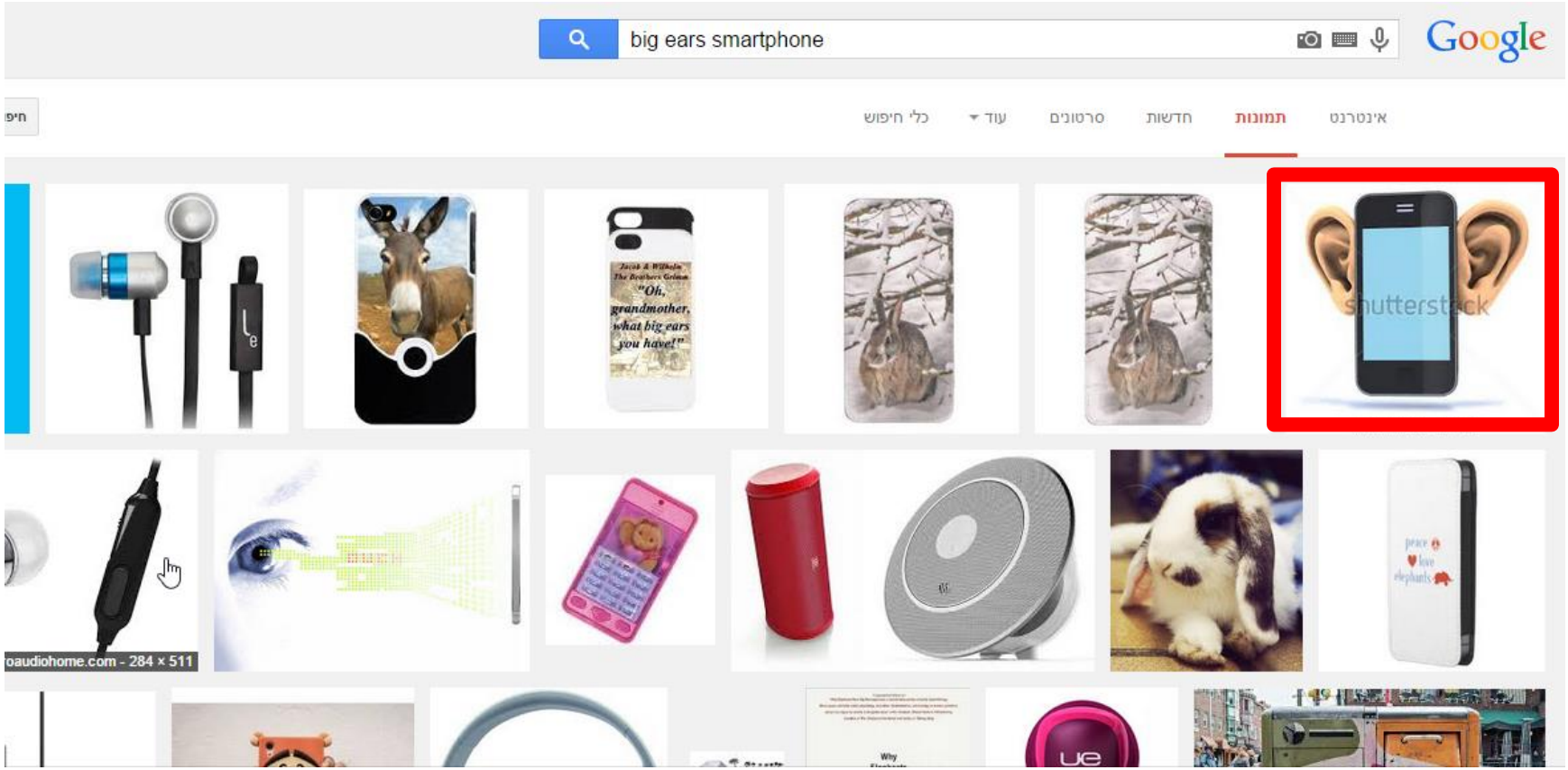


a


Area Under ROC Curve, Leave One Out in Pascal
Whole Feature
Selected Feature
Area Under ROC Curve
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Rein
Cloth
Furn. Seat
Plastic
Furn. Arm
Occluded
Round
Wing
andlebars
Torso
2D Boxy
Vert Cyl
urn. Back
Leg
Wood
Foot/Shoe
Exhaust
Glass
Shiny
Ear
Head
Furn. Leg
Tail
3D Boxy
Eye
Furry
Headlight
Taillight
Wheel
Snout
ide mirror
Door
Horiz Cyl
Metal
Window
Row Wind
Engine
Clear

Experiment – Naming




big ears smartphone
Google
אינטרנט
תמונות
חדשות
סרטונים
עוד
כלי חיפוש
"Oh, grandmother, what big ears you have!"
shutterstock
oaudiohome.com - 284 × 511

Conclusions

- Inferring object properties should be an important goal of object recognition
- Learning attributes enables several new abilities :
  - Predicting properties of new types of objects
  - Identifying unusual about a familiar object
  - Learning from verbal description
- Raises an important issue concerning dataset biases while learning.

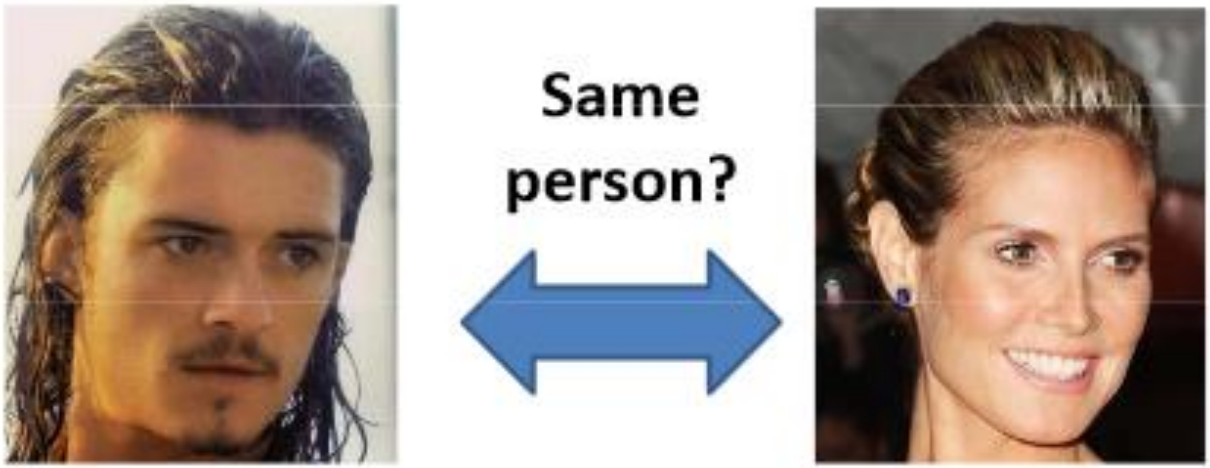
Attributes and Simile Classifier for face verification

Agenda:

- Face verification using attributes
- Face verification using simile

Face Verification

Given two face images – determine if they show the same individual.



Introduction

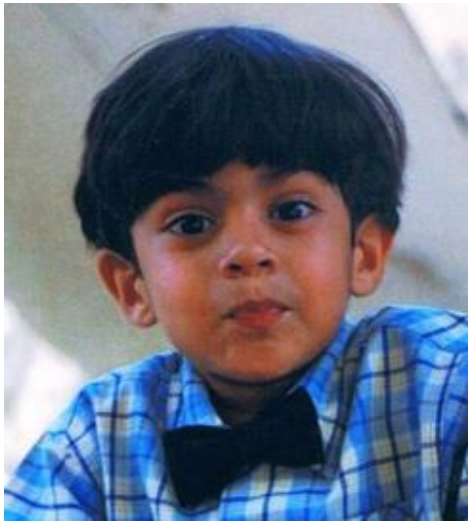
There is large variability in ways which the same face presents itself to the camera, for example:

- Pose
- Hairstyle & Skin
- Expression
- Illumination

Recognition using visual attributes

- Different objects can be described by attributes:

Stripes
Four-legged
Furry



Male
Child
Black hair

Attributes can define categories

Female Eyeglasses Middle-aged Dark hair

The Attributes Approach

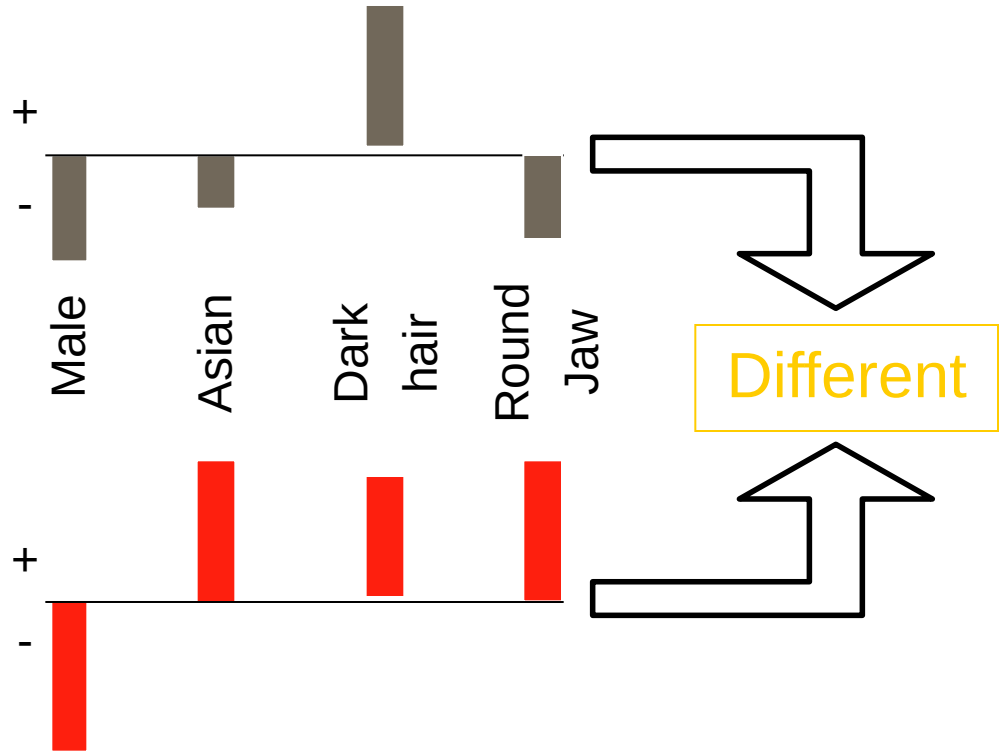
- The first step in our approach is to extract **“low-level”** features – such as SIFT , HOG , the pixel values..
- The second step is to use these low-level features to compute “high-level” visual features, or **traits**, which are insensitive to changes in pose, illumination and expression.
- The third step is preforming the verification, based on the traits values we get.

The Attributes Approach

Images ⇨ Low-level features ⇨ Attributes ⇨ Verification

RGB
HOG
SIFT
…

RGB
HOG
SIFT
…



Low-level features

- We first detect faces and fiducial point locations using a commercial face detector.
- The faces are then recertified to a common coordinate system using an affine wrap.
- The low-level features are constructed by choosing a face region, a feature type to extract from this region, and a normalization and aggression of these values.

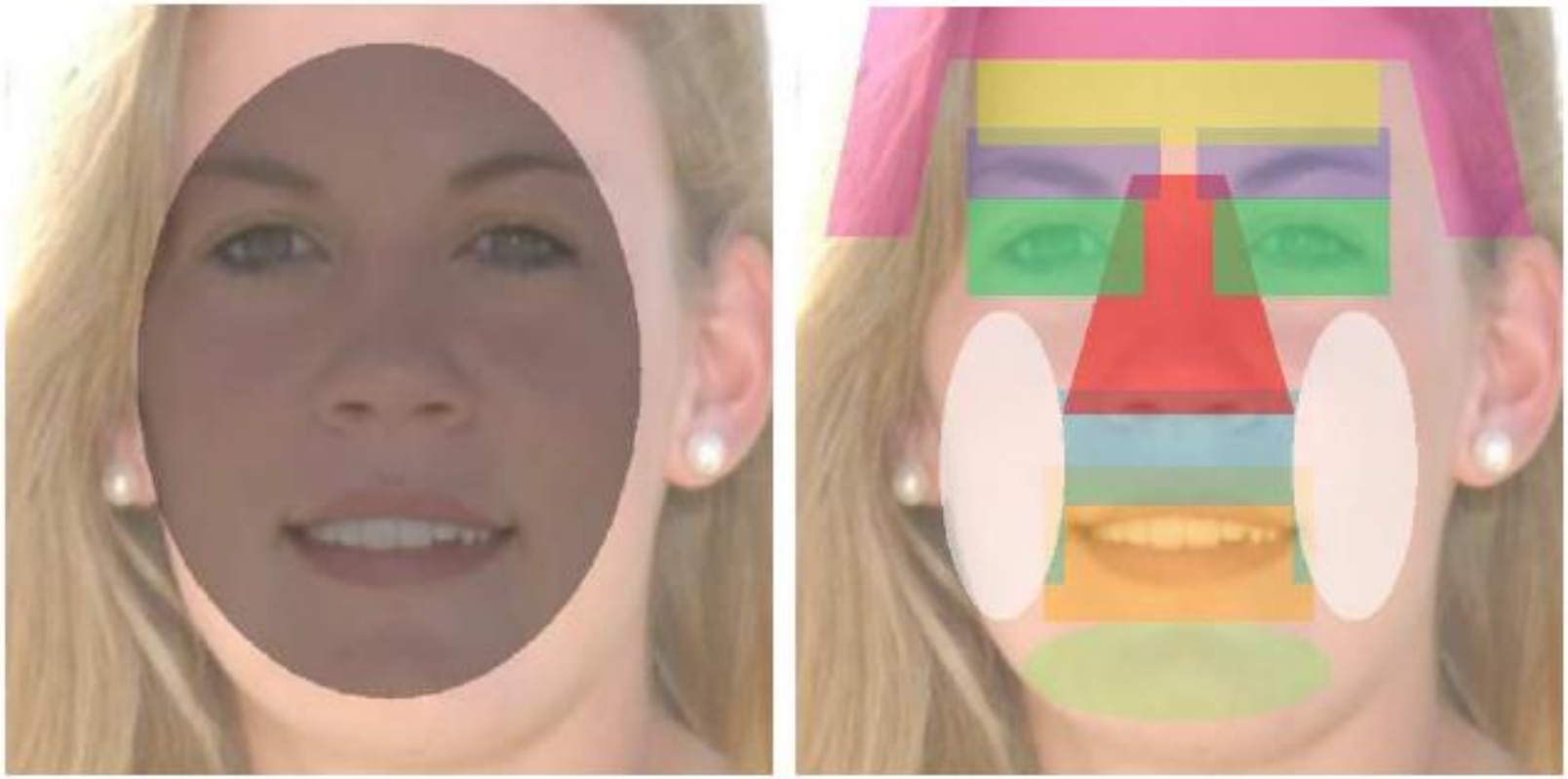
Low-level features – cont.

The regions are constructed by hand-labeling different parts of the rectified face images – nose, eyes , mouth , etc.

- Regions are defined manually, but only once.
- This division takes advantage of the common geometry of human faces, while still allowing differences.
- Extracted values are normalized to generalize the image.

Low-level features – cont.

A sample face split into regions of interest:



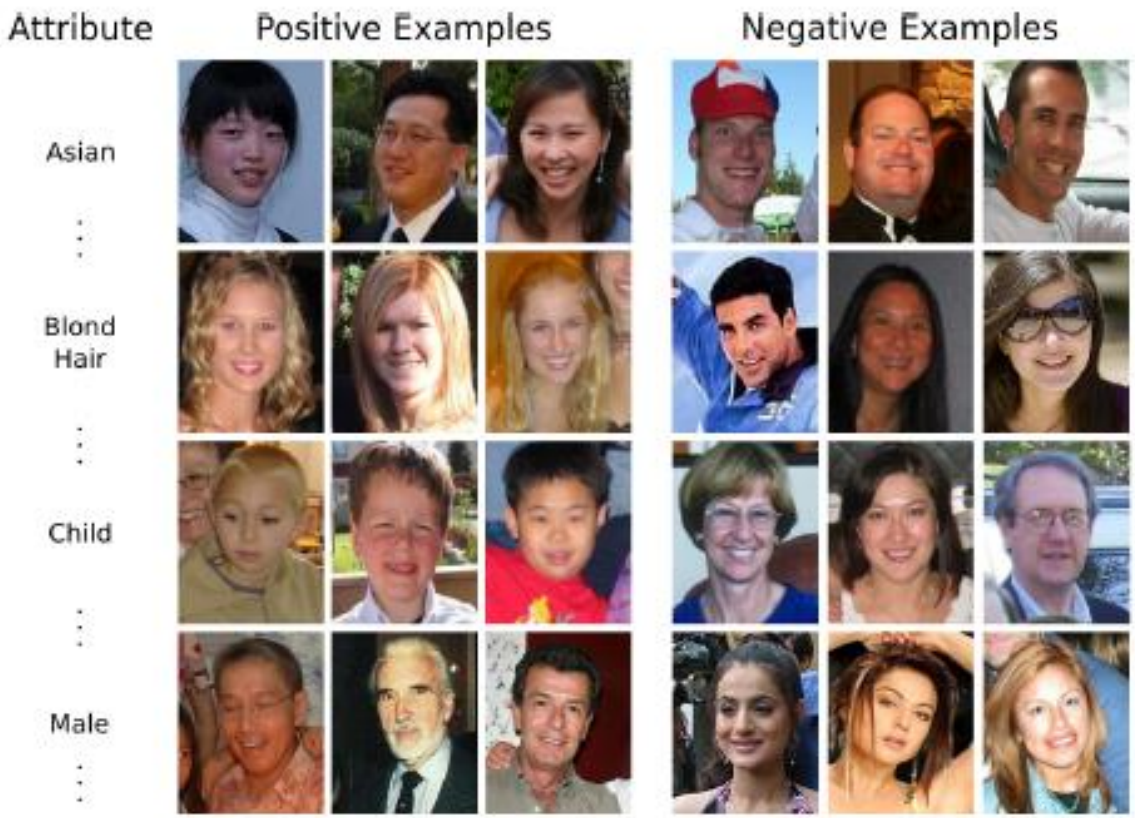
Low-level features – cont.

- There are many features types – as RGB and HSV color spaces, edge magnitudes and gradient directions.

- We get a large number of low-level features $\{f_i\}$ - those features are used for each trait classifier C_i as described next.

Traits Classifiers

- For each attribute, we build a classifier C_i to detect the describable attribute.

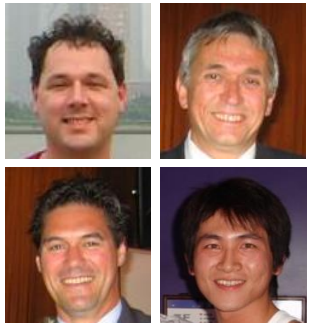


Traits Classifiers – cont.

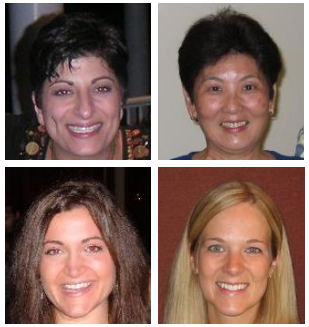
- We train several attribute classifiers, each requires a set of positive and negative examples for each attribute.
- Uses a simplified version of adaboost.
- Best features for classification chosen automatically from our low-level features, these are used to train the final classifiers.

Learning an attribute classifier

Training images → Low-level features → Feature selection → Train classifier



Males



Females

RGB
HoG
HSV
…

RGB
HoG
HSV
…

RGB, Nose

HoG, Eyes

HSV, Hair

Edges, Mouth

…

Gender classifier

Male

0.87

Traits Classifiers

65 attributes classifiers were trained using the system.

Attribute	Accuracy
Asian	92.32%
Attractive Woman	81.13%
Baby	90.45%
Bags Under Eyes	86.23%
Bald	83.22%
Bangs	88.70%
Black	88.65%
Black Hair	80.32%
Blond Hair	78.05%
Blurry	92.12%
Brown Hair	72.42%
Child	83.58%
Chubby	77.24%
Color Photo	95.50%
Curly Hair	68.88%
Double Chin	77.68%
Environment	84.80%
Eye Width	90.02%
Eyebrow Shape	80.90%
Eyebrow Thickness	93.40%
Eyeglasses	91.56%
Eyes Open	92.52%
Flash Lighting	72.33%
Frowning	95.47%
Goatee	80.35%
Gray Hair	87.18%
Gray Hair	87.18%
Harsh Lighting	78.74%
High Cheekbones	84.70%
Indian	86.47%
Male	81.22%
Middle-Aged	78.39%
Mouth Closed	89.27%
Mouth Partially Open	85.13%
Mouth Wide Open	89.63%
Mustache	91.88%
No Beard	89.53%
No Eyewear	93.55%
Nose Shape	86.87%
Nose Size	87.50%
Nose-Mouth Lines	93.10%
Obstructed Forehead	79.11%
Oval Face	70.26%
Pale Skin	89.44%
Posed Photo	69.72%
Receding Hairline	84.15%
Rosy Cheeks	85.82%
Round Face	74.33%
Round Jaw	66.99%
Semi-Obscured Forehead	77.02%
Senior	88.74%
Shiny Skin	84.73%
Sideburns	71.07%
Smiling	95.33%
Soft Lighting	67.81%
Square Face	81.19%
Straight Hair	76.81%
Sunglasses	94.91%
Teeth Not Visible	91.64%
Teeth Visible	91.64%
Visible Forehead	89.43%
Wavy Hair	64.49%
Wearing Hat	85.97%
Wearing Lipstick	86.78%
White	91.48%
Youth	85.79%

Traits Classifiers – collecting examples

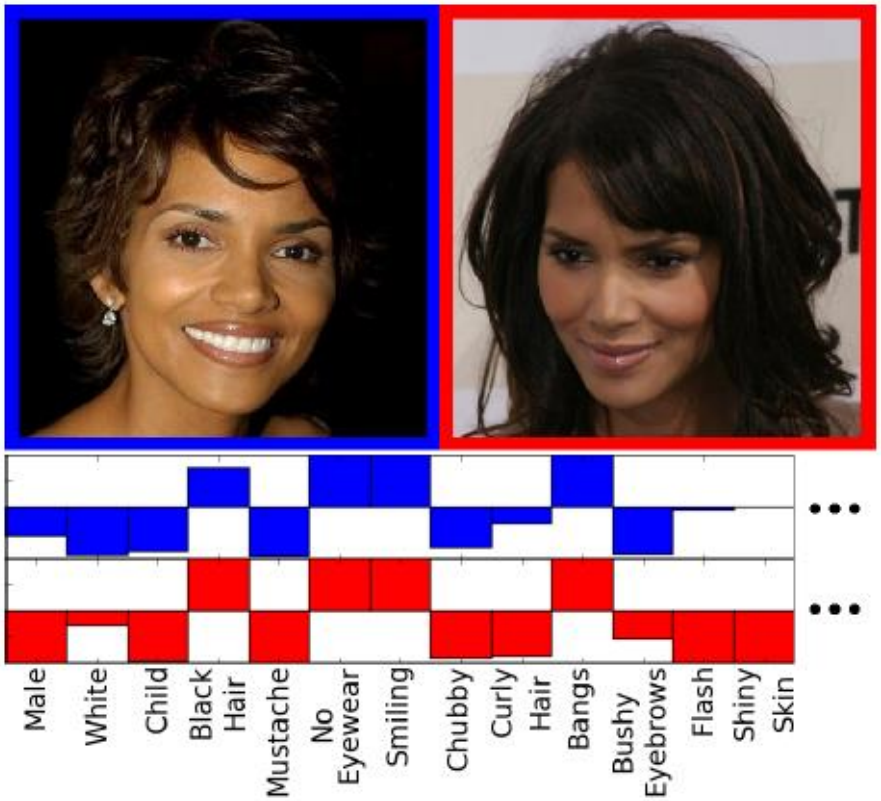
- Each attribute had at least 1,000 examples – at least 500 positive and 500 negative.

- That means, that for 65 attributes, at least 65,000 images needed to be obtained and labeled!

Trait Classifiers – cont.

We can see a correlation between the trait values of photos of the same person.

- The “Flash” and “Shiny skin” attributes produce differently.
- The rest of the attributes are still agree despite the change in pose.



Simile Classifiers

- One can imagine that there are many visual cues to people’s identities that cannot be described – at least not concisely.

- In order to use this information, we use the concept of a “simile” classifier.

- The basic idea is that we can describe a person's appearance in terms of the similarity of different parts of their face to a limited set of “reference” people.

Simile Classifier

Female

Black hair

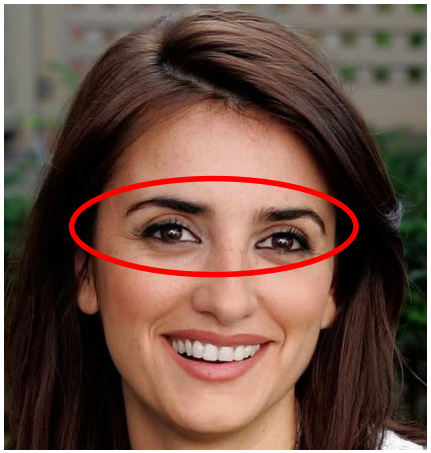
Female

Mouth closed

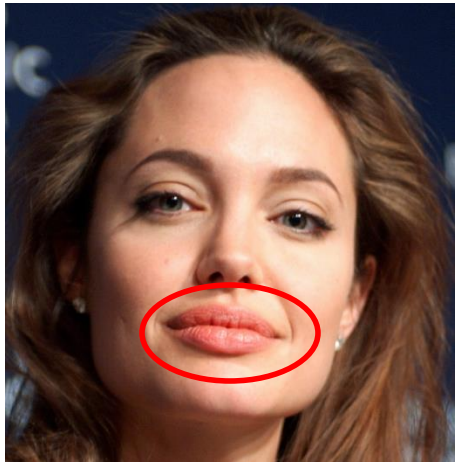
White

Young

Simile Classifier



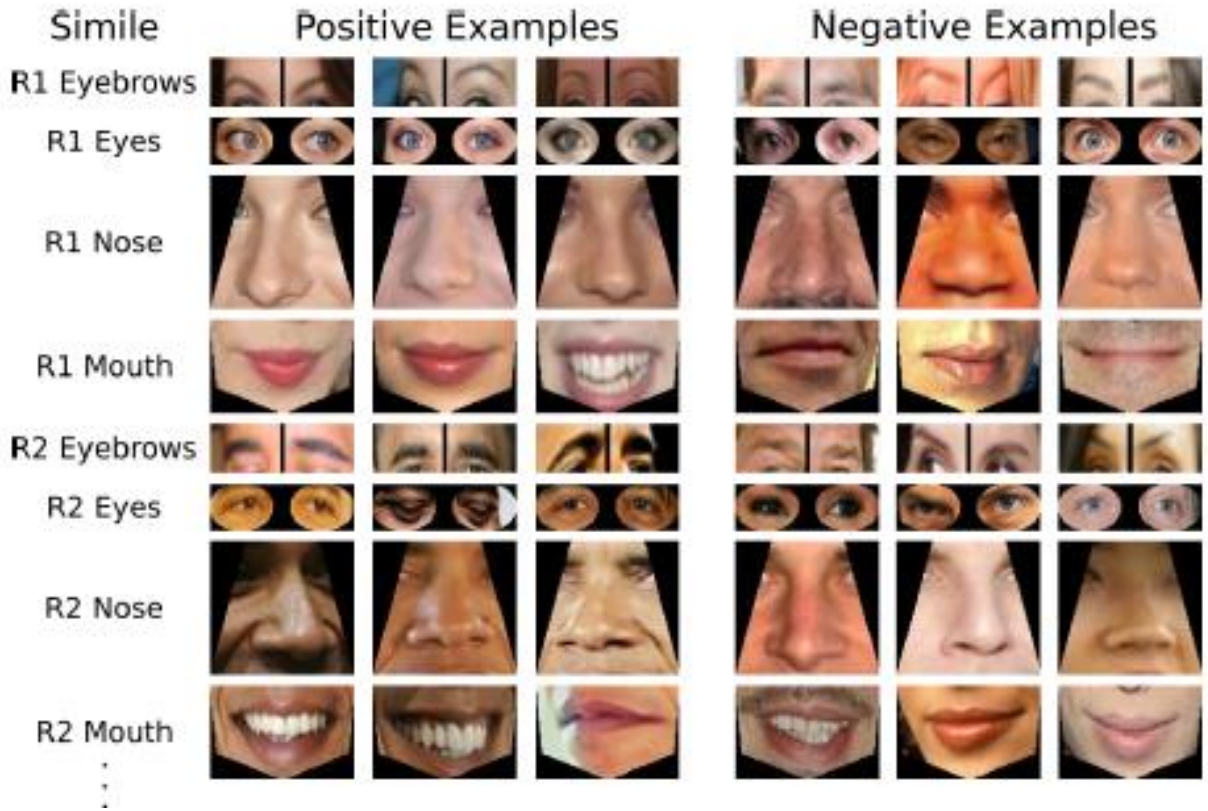
Penelope Cruz



Angelina Jolie

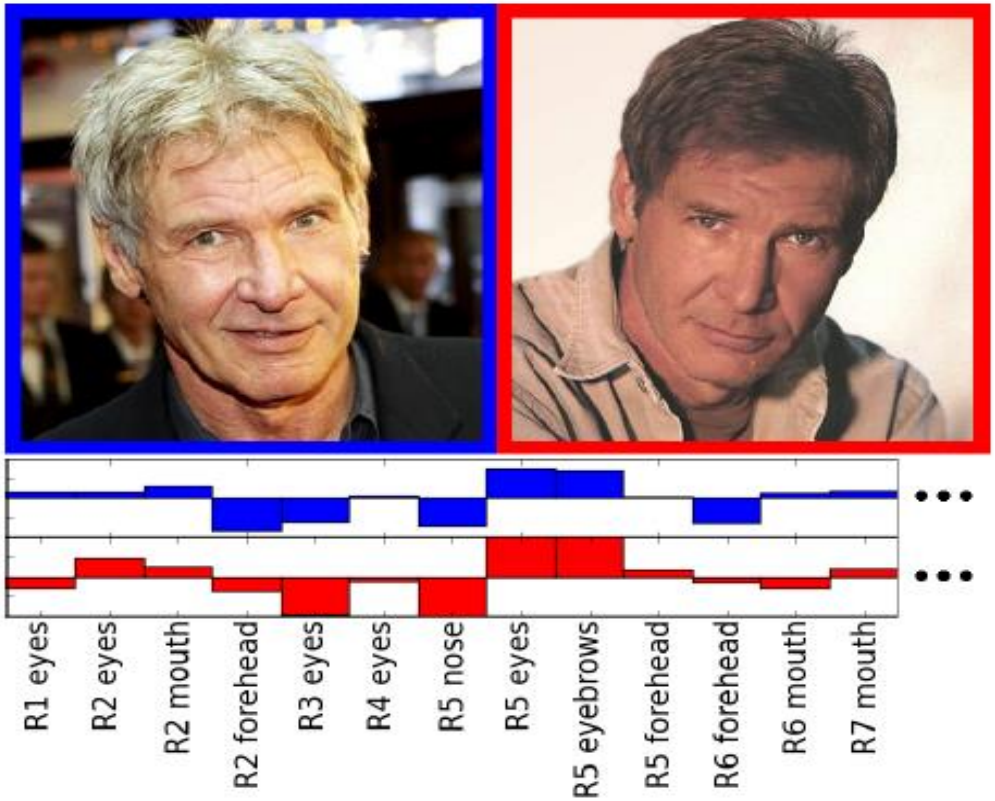
Simile Classifier – cont.

- For each reference person, several simile classifiers are trained for each face region – one per feature type, yielding a large set of total classifiers.



Simile Classifier – cont.

We can see correlation between the similies values of photos of the same person:



Performing verification

To decide if two face images, $I_1 \; and \; I_2$ are the same person, we compare their trait vectors using a final classifier F which defines our verification function:

$$v(I_1, I_2) = D(C(I_1), C(I_2))$$

This function is positive when the two images show the same person and negative otherwise.

D is a **verification classifier.**

Verification Classifier

- The trait vectors are obtained by the attributes method, or by the similies method (or both).

- As said, corresponding values of $C_i(I_1) and\ C_i(I_2)$ from the *i*-th iteration should be similar if the images are of the same person.

- In addition, the SVM classifier optimize for separating data at the separation boundary – so differences in values close to 0 are more important than differences between those with greater absolute values.

Verification Classifier – cont.

- Let a_i and b_i be the outputs of the *i*-th trait classifier for each image (I_1, I_2) - $a_i = C_i(I_1), b_i = C_i(I_2)$.

- The absolute value of $a_i - b_i$ nets us the similarity result, and the product $a_i * b_i$ gives us the signal.

- We save the pair $p_i = \left\langle |a_i - b_i|, (a_i * b_i) * g(\frac{1}{2}(a_i + b_i)) \right\rangle$

  where g is a gaussian with mean 0 and variance 1.

Verification Classifier – cont.

- These pairs are concatenated to form the 2n dimension vector that we actually classify:

$$v(I_1, I_2) = D(\langle p_1, p_2, \dots, p_n \rangle)$$

- Training D requires pairs of positive examples and negative examples.

- SVM with an RBF kernel was used for D.

Experiments and Results

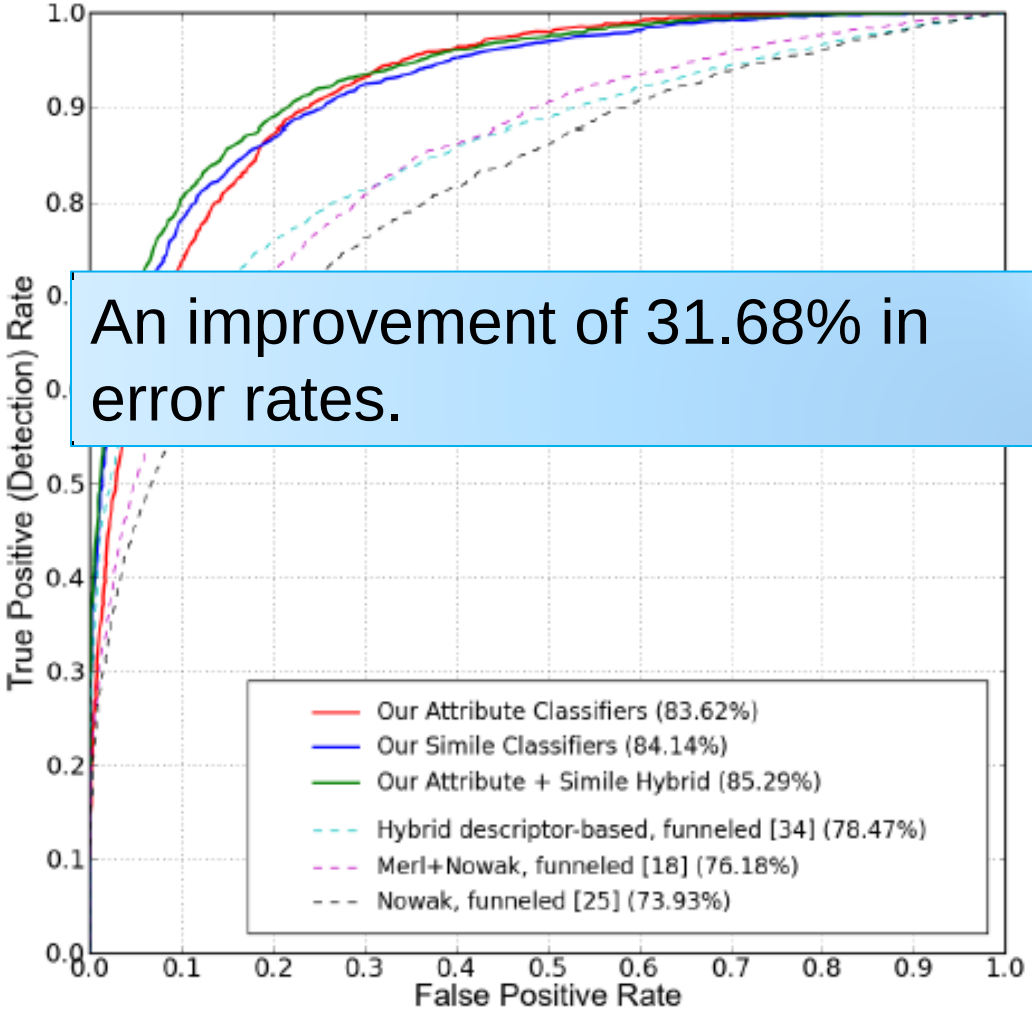
Tested on two datasets:

- LFW – Labeled Faces in the Wild – 13,233 images of 5,749 people.
- PubFig – a more challenging dataset. Build by the writers of the article.

LFW Dataset

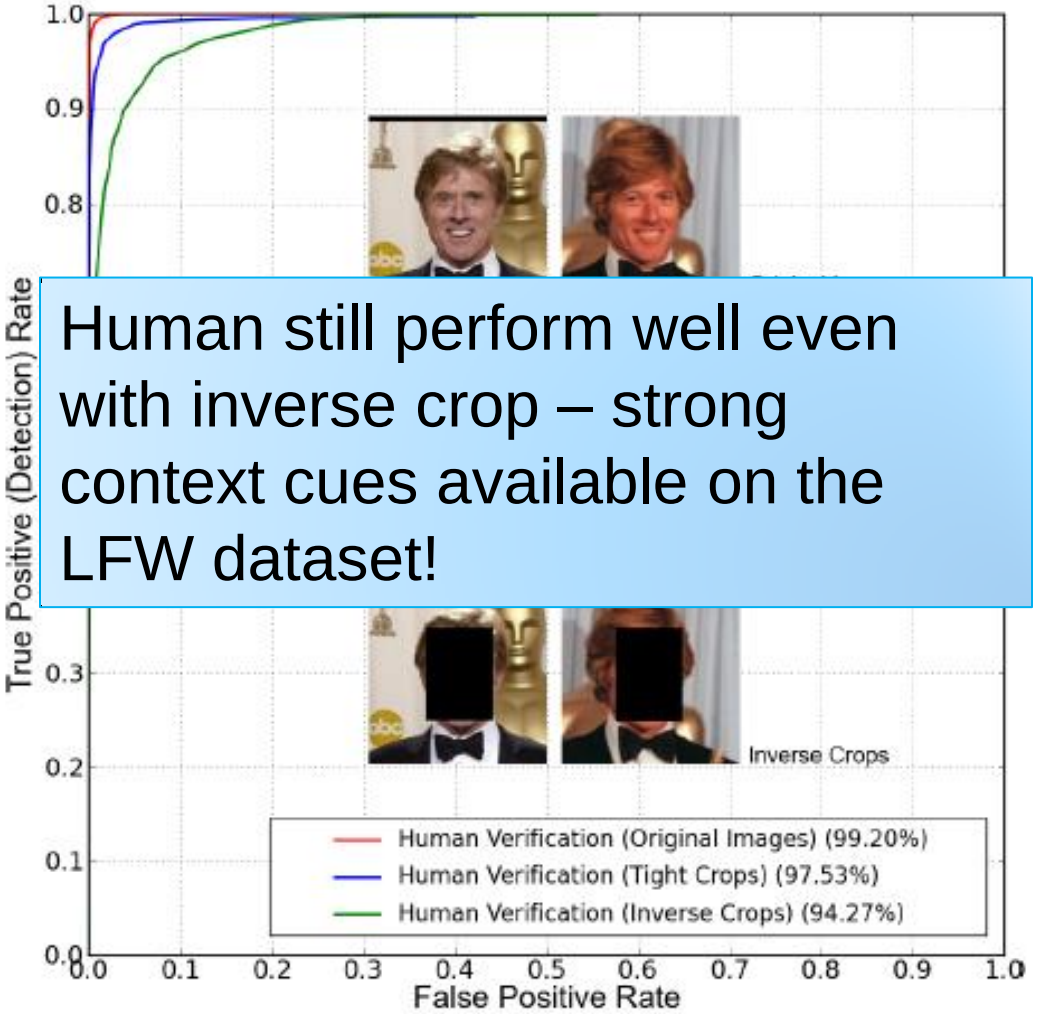
vis-www.cs.umass.edu/lfw

Performance on LFW



An improvement of 31.68% in error rates.

Human face verification performance



Human still perform well even with inverse crop – strong context cues available on the LFW dataset!

PubFig dataset & benchmark

Public figures:

- Politicians
- Celebrities

Larger & deeper:

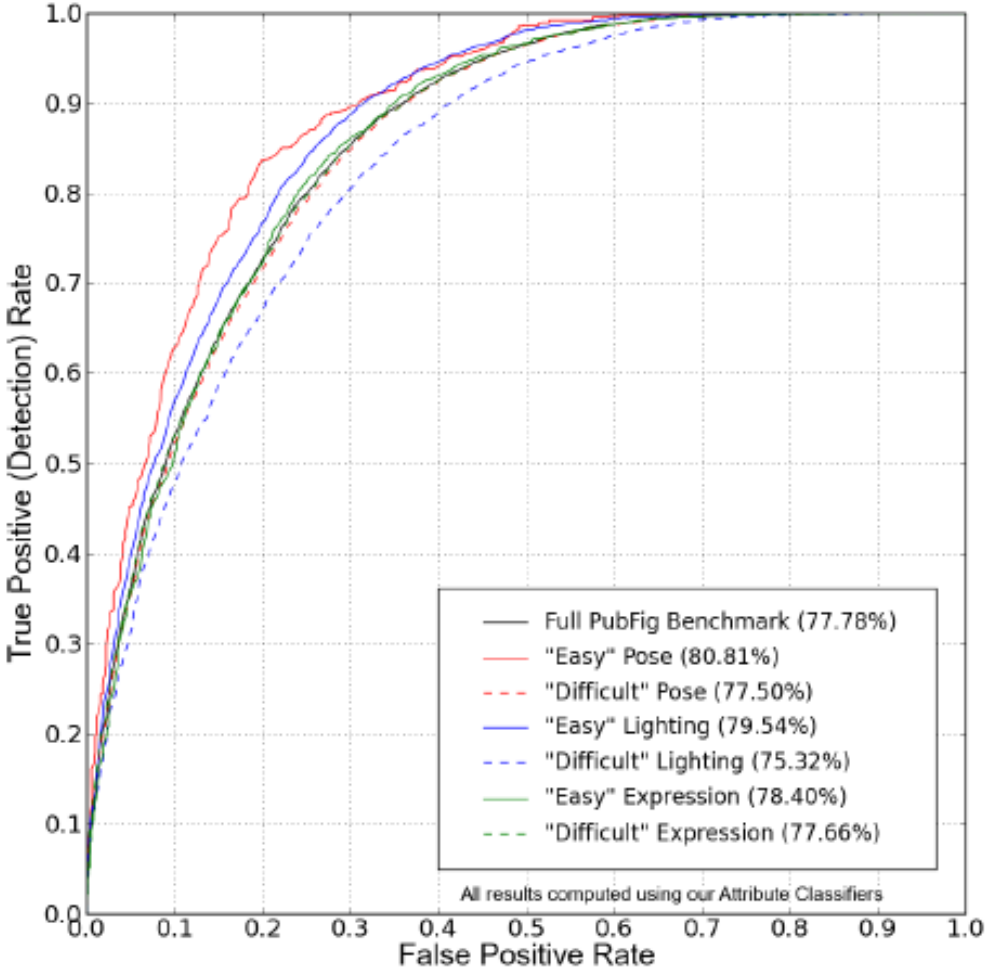
- 60,000 Images
- 200 People
- 300 Images per person

Subsets:

- Pose
- Illumination
- Expression

http://www.cs.columbia.edu/CAVE/databases/pubfig/

Performance on PubFig



Future work

- Further improvements can be made by using:
  - More attributes
  - Improving the training process
  - Combining attributes and simile classifiers with low level image cues.

- Applying attributes to other objects?

Observations

- Taken also background as a feature, will error-rate on the LFW dataset decrease even more?

- Can similar results be achieved if only low-level data were used?

Summary

- Two approaches for face verification using traits computed in face images
  - Attribues
  - Similie

- Improvement on the LFW dataset – without the background or the context, but still there is room for improvement.

- PubFig dataset – larger and deeper than previous datasets.

Resources

- **Attribute and Simile Classifiers for Face Verification, N. Kumar, A. Berg, P. Belhumeur, S. Nayar. ICCV 2009**
- **Reading group presentation by Esa Rahtu** http://www.robots.ox.ac.uk/~vgg/rg/slides/faceverification.pdf
- **Attribute and Simile Classifiers for Face Verification**

http://www.cs.columbia.edu/CAVE/projects/faceverification/

- **Labeled Faces in the Wild**

  http://vis-www.cs.umass.edu/lfw/
- **PubFig**

  http://www.cs.columbia.edu/CAVE/databases/pubfig/