Towards a detailed understanding of images

CLSP 2012 Summer Workshop

Matthew Blaschko, *École Centrale Paris*

Ross B. Girshick, *University of Chicago*

Juho Kannala, *University of Oulu*

Iasonas Kokkinos, *École Centrale Paris*

Siddharth Mahendran, *Johns Hopkins University*

Subhransu Maji, *Toyota Technological Institute*

Sammy Mohamed, *Stony Brook University*

Esa Rahtu, *University of Oulu*

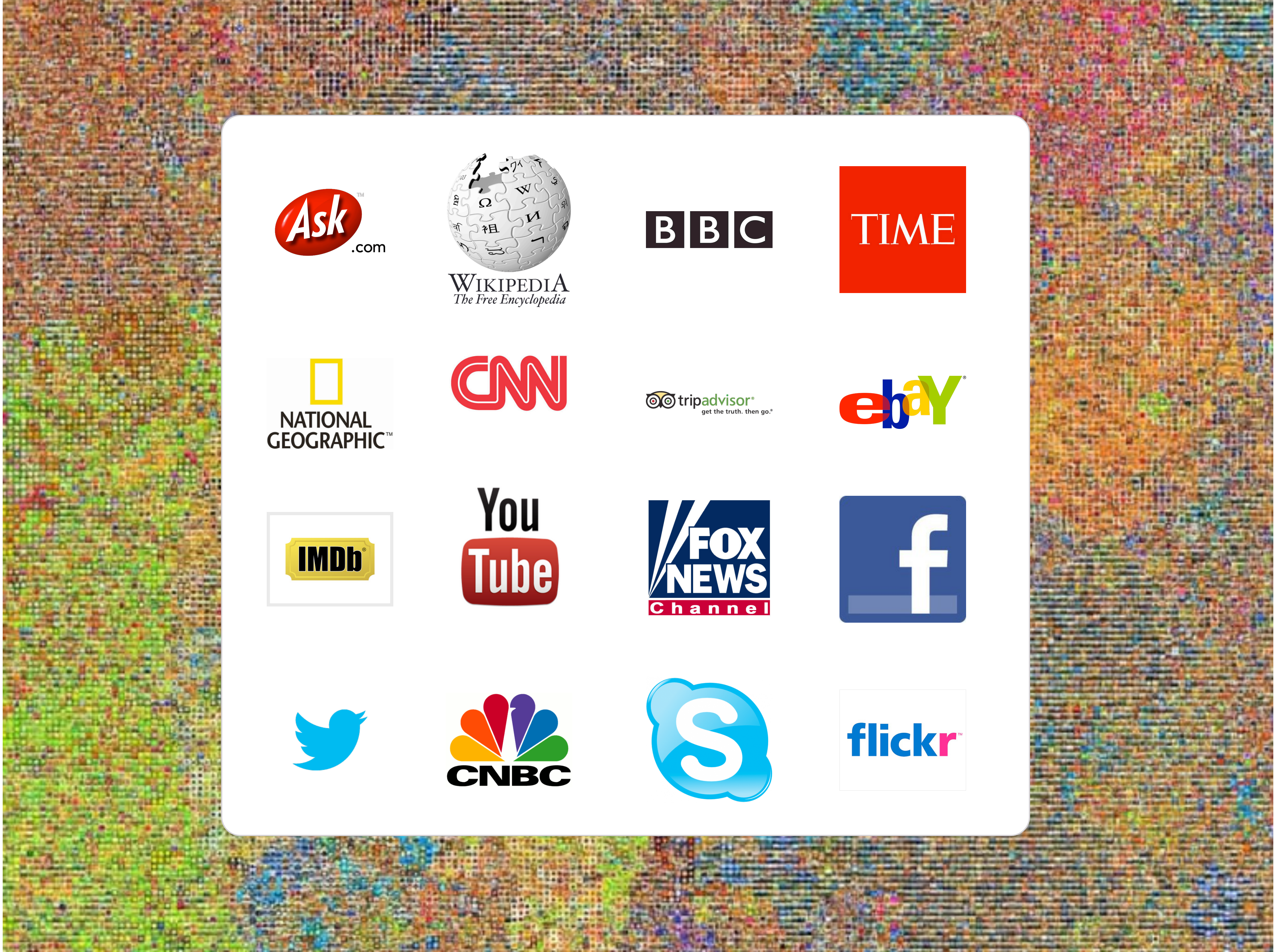
Naomi Saphra, *Carnegie Mellon University*

Karen Simonyan, *University of Oxford*

Ben Taskar, *University of Pennsylvania*

Andrea Vedaldi, *University of Oxford*

David Weiss, *University of Pennsylvania*


Ask.com
WIKIPEDIA
The Free Encyclopedia
BBC
TIME
NATIONAL GEOGRAPHIC
CNN
tripadvisor
get the truth. then go.
ebay
IMDb
You Tube
FOX NEWS Channel
f
CNBC
S
flickr

Semantic search

description $\xrightarrow{\textbf{text2im}}$ image

Searching personal photos

Emiliy
gift

cherry blossom
cityscape

Searching press archives

Obama
kneeling
beach

Searching the BBC collection

John Cleese
jacket and tie
phone

Captioning and summarisation

im2text

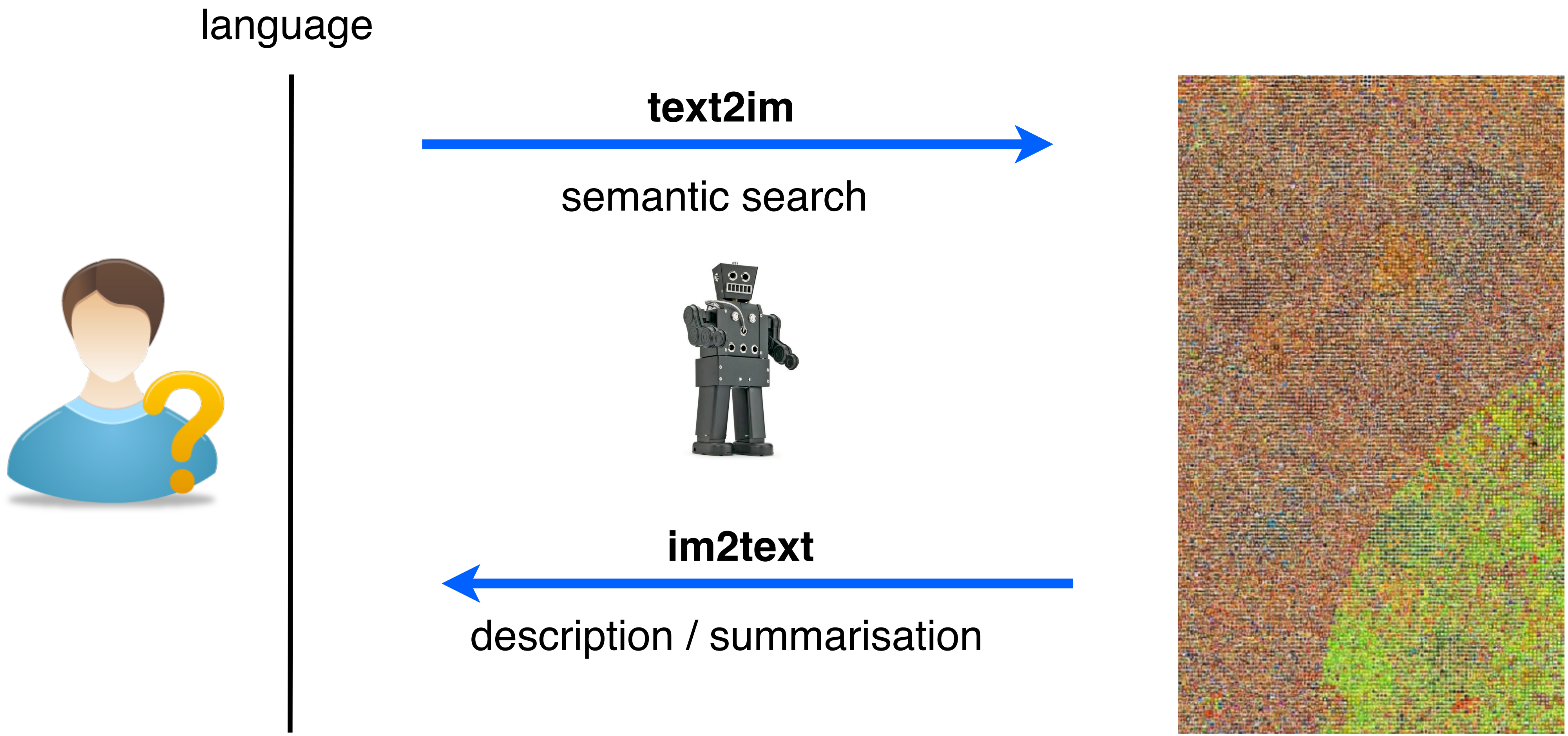
description ← image

This picture shows one person, one grass, one chair, and one potted plant. The person is near the green grass, and in the chair. The green grass is by the chair, and near the potted plant

a cow with green grass | sheep with a gray sky by the road | people with boats | a brown cow | people at a wooden table

[Kulrani *et al.* 11, Mitchell *et al.* 12]

Human-centric machine vision



Semantic tasks in computer vision

image classification

object detection

Coarse semantics.

Beyond categories: objects in detail

Most human-centric tasks require
understanding the details of objects.

object class

bicycle

viewing conditions

right-facing

parts, materials, colours, ...

chrome-blue gear

white frame

handle bar

seat

part relations

seat

handle bar

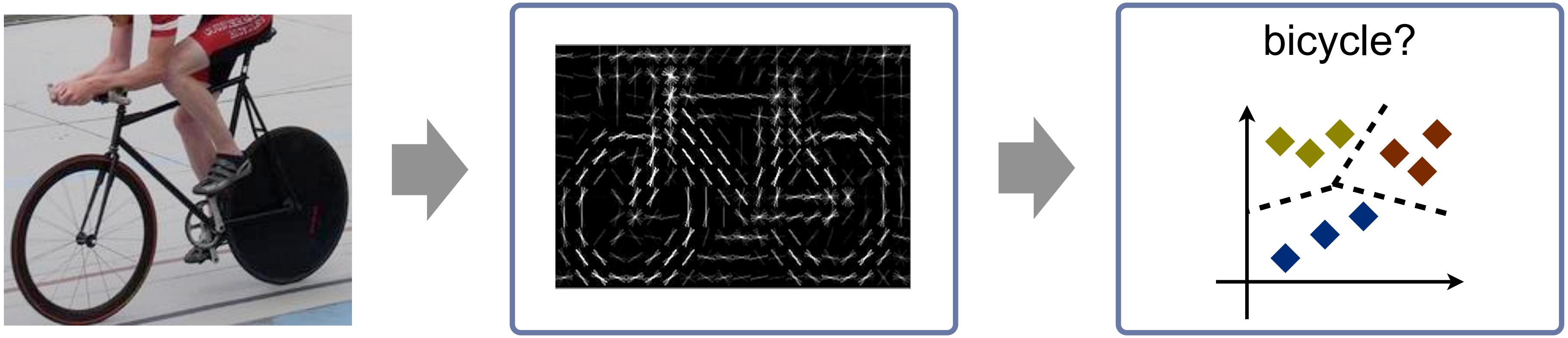
higher

Advantages of detailed understanding

Better support for human-centric tasks.

Current models are opaque, semantically shallow:

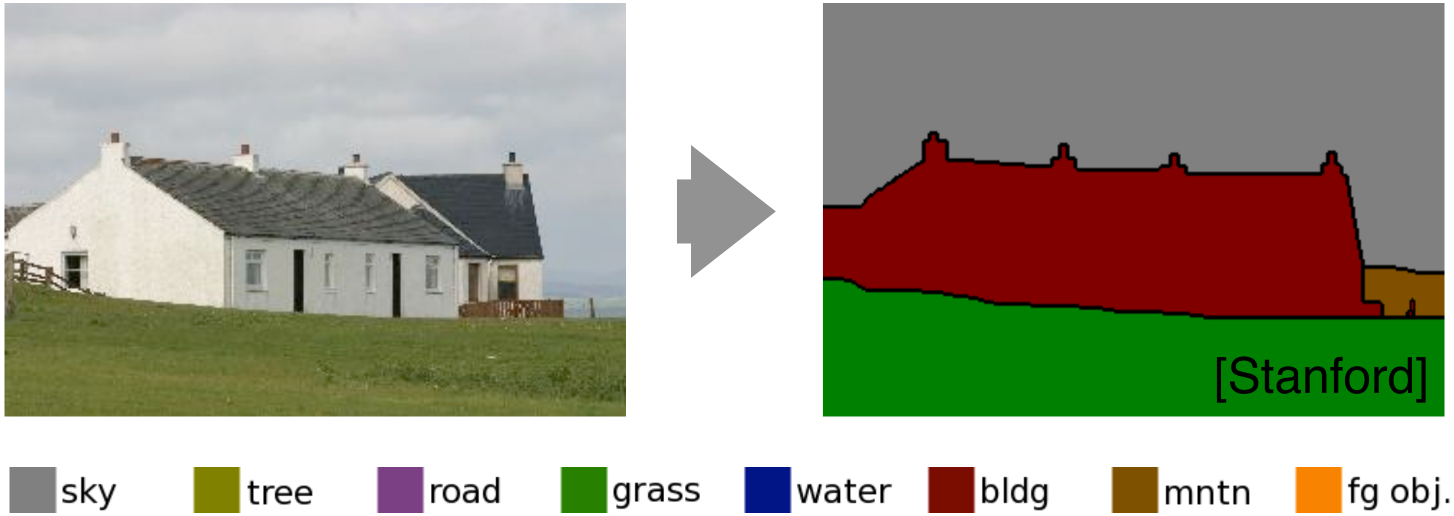
input **representation** **interpretation**



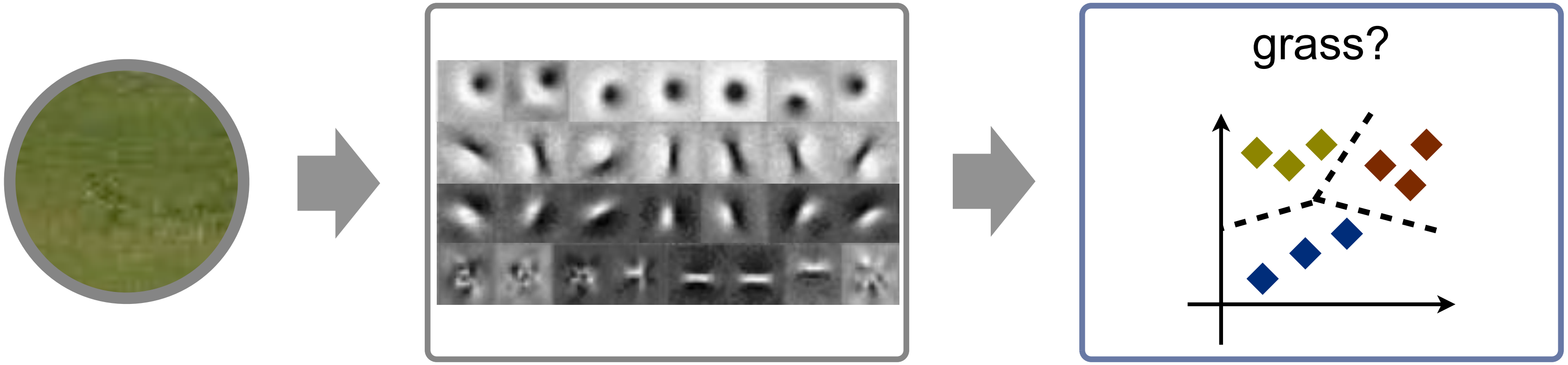
A semantically decomposed model is easier to
understand, diagnose, and improve.

Not just objects: texture semantic

segmenting stuff



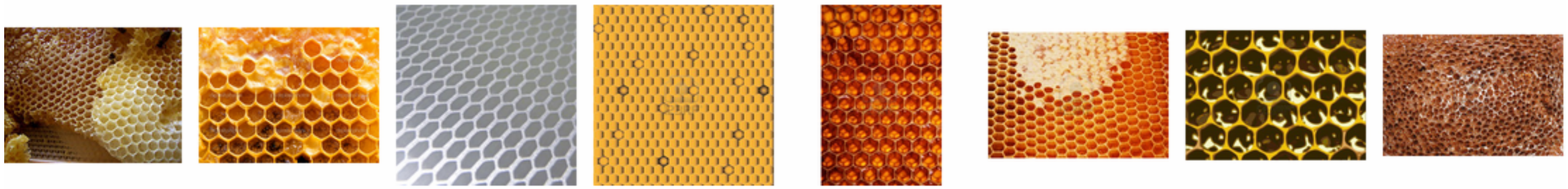
input **representation** **interpretation**



Stuff in detail

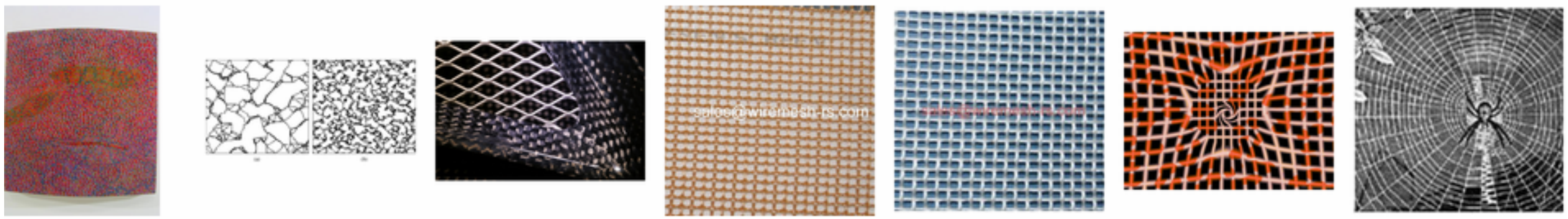
Texture models for human-centric tasks.

honeycombed

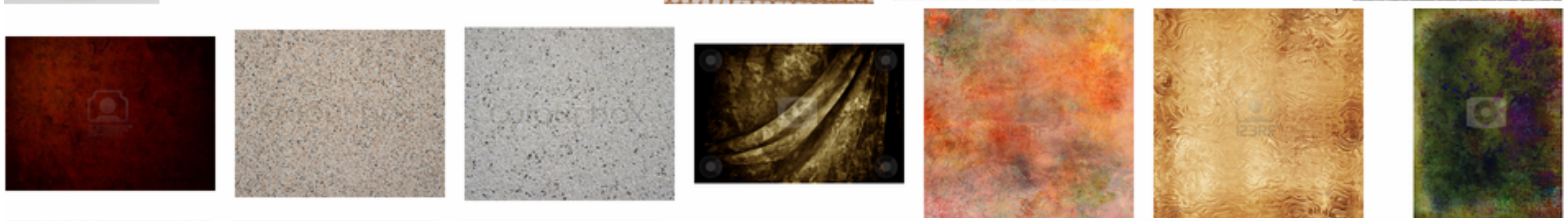


latticed

netlike



mottled



meshed

Opening a path to detailed semantic analysis

Problem	Data	Time frame	Progress
Image Classification	Caltech-101	2003-06	star models, BoW
Object Detection	PASCAL VOC	2006-12	DPMs, large scale learning
Parts & Attributes	?	2012-?	?

what can you do
in six weeks?

Overview

Objects in Detail

Parts & attributes

- A new dataset
- An object lexicon
- Localising parts
- Layouts
- Recognising attributes
- The cost of data collection

Stuff in Detail

Texture

- A texture lexicon
- A new dataset
- Transformation invariant semantic

Parsing

Bottom-up inference

- Learning to merge
- Cascading
- Scoring regions by attributes

Overview

Objects in Detail

Parts & attributes

- A new dataset
- An object lexicon
- Localising parts
- Layouts
- Recognising attributes
- The cost of data collection

Stuff in Detail

Texture

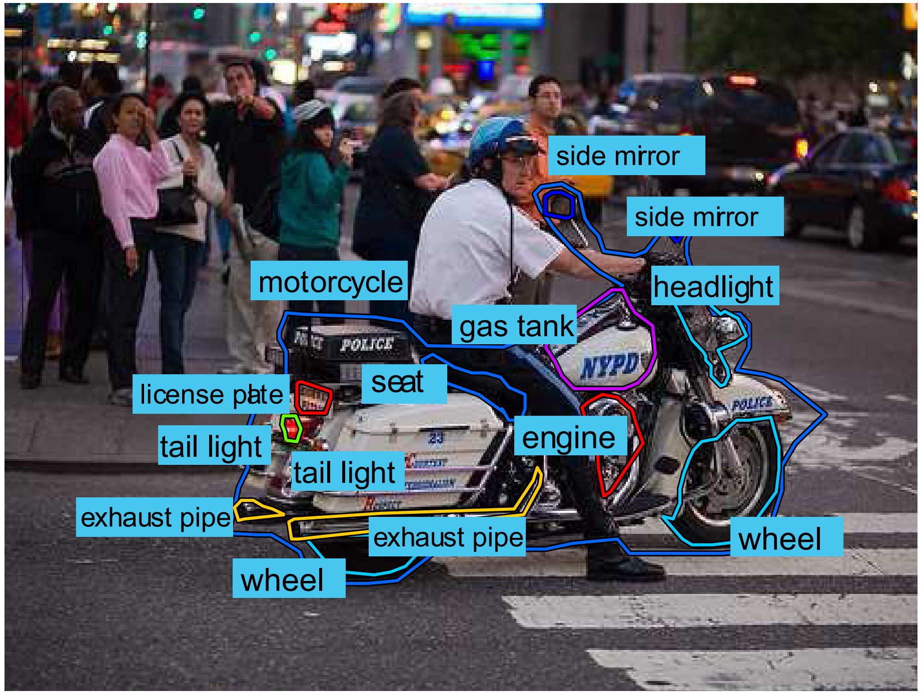
- A texture lexicon
- A new dataset
- Transformation invariant semantic

Parsing

Bottom-up inference

- Learning to merge
- Cascading
- Scoring regions by attributes

The need for a dataset

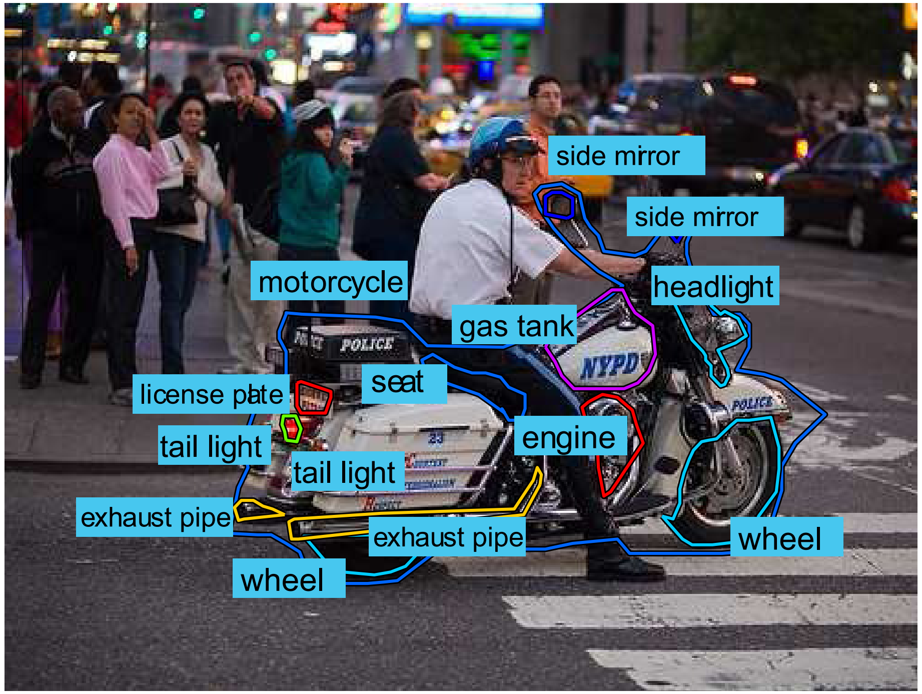


Detailed semantic tasks:

- which type of motorcycle is this?
- where is the right exhaust pipe?
- what is the tail-light shape?
- what is the colour of the panniers?
- is the head light visible?
- is there a rider?

- **Why annotated data:**
  1. Evaluation
  2. Training

The need for a **new** dataset

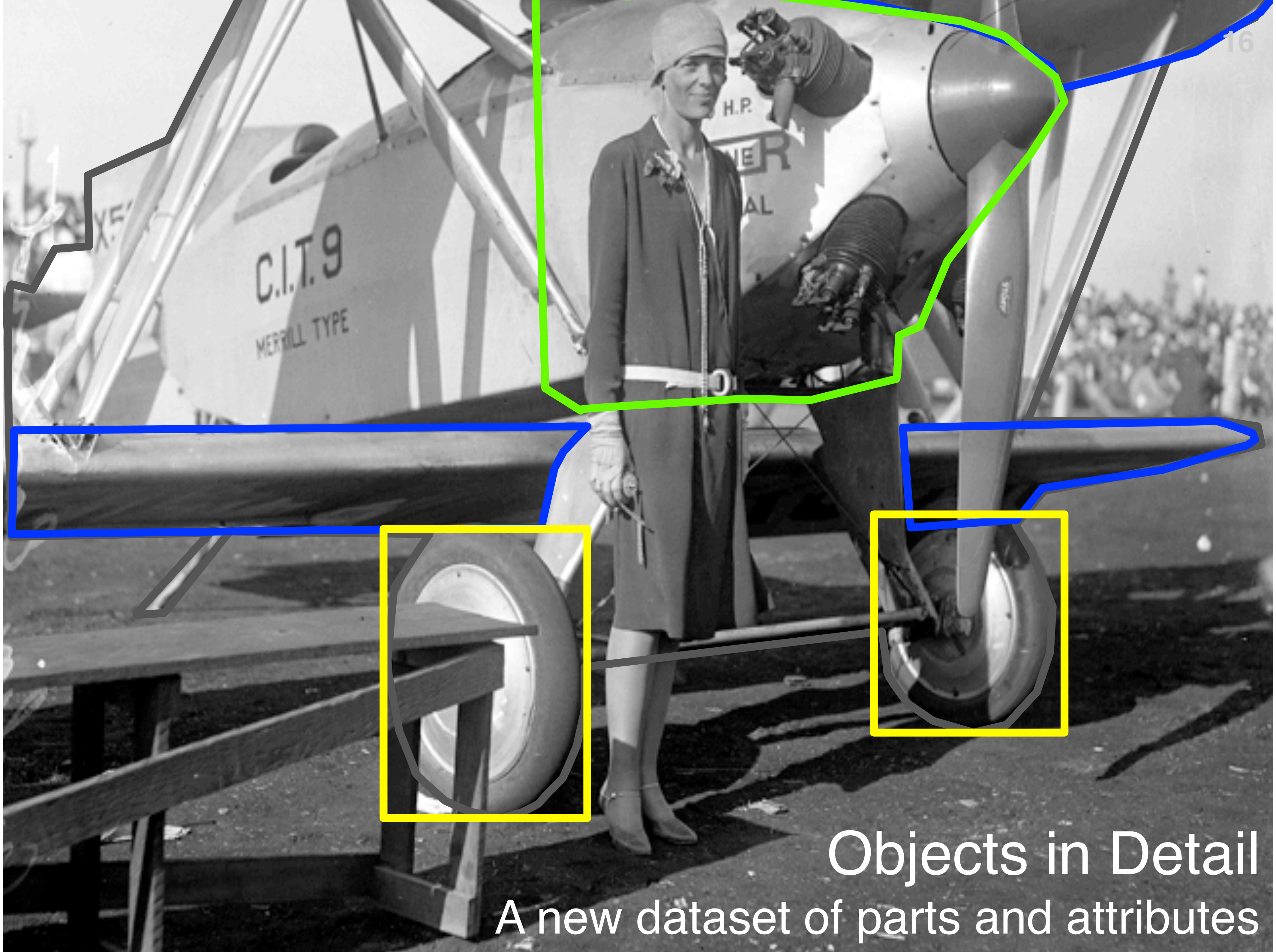


CORE Dataset
[Farhadi Endres Hoiem 2010]

✔ Sharing of parts

✗ Accurate recognition of parts and their attributes

category	**# parts / object**
airplane	9.49
alligator	8.90
bat	8.55
bicycle	6.62
blimp	5.29
boat	3.76
bus	11.32
camel	12.15
car	8.15
carriage	5.43
cat	11.50
cow	10.93
crow	8.08
dog	13.17
dolphin	6.17
eagle	8.01
elephant	11.97
elk	11.47
hovercraft	4.81
jetski	3.64
lizard	9.27
monkey	11.90
motorcycle	9.03
penguin	6.95
semi	11.51
ship	4.29
snowmobile	5.99
whale	4.82



Objects in Detail

A new dataset of parts and attributes

Designing a dataset of parts & attributes

- **Motivation**
  - we know that parts & attributes are useful for sharing, etc.
  - but how well can we recognise parts & attributes?
- **Aims of the dataset**
  - ~~object recognition~~ $\longrightarrow$ parts & attributes recognition
  - benchmarking: measure and encourage progress
  - inspire new technical challenges
- **How**
  - high-quality annotations (*e.g.* PASCAL VOC)
  - sufficiently large to be representative of data variability
  - the object class and location is *given*
  - define new tasks and metrics
    - part localisation
    - attribute recognition
    - joint tasks

An rich object category

With parts and attributes

Spotters: an effective data source

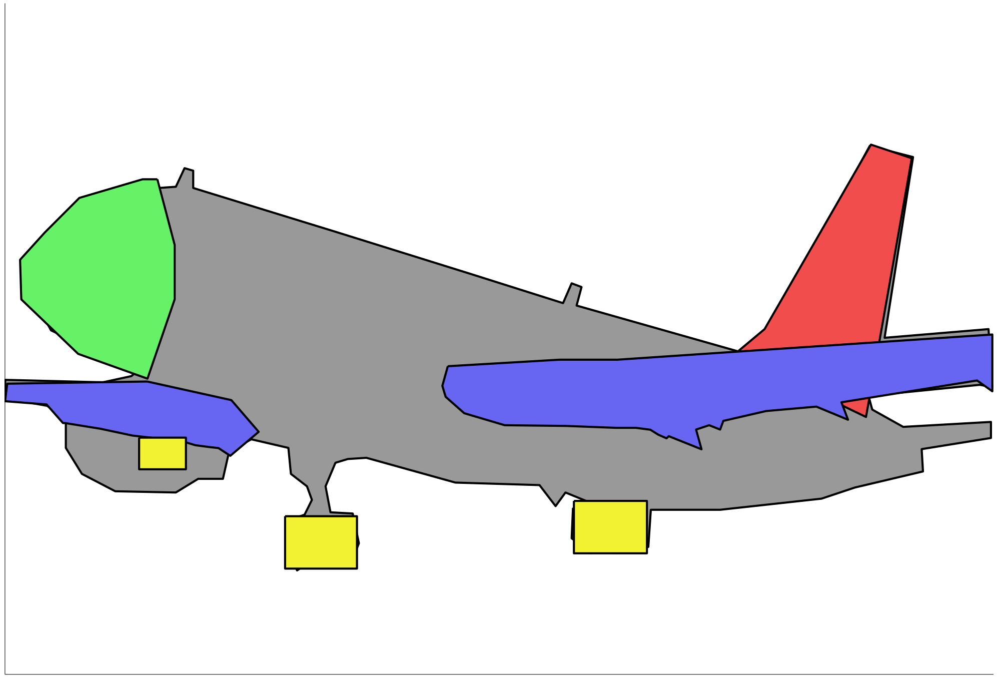
Aircraft Spotters http://www.airliners.net/

Selected about 7,500 for annotation.

Trivial extension to **other classes**

railways: http://railpictures.net/

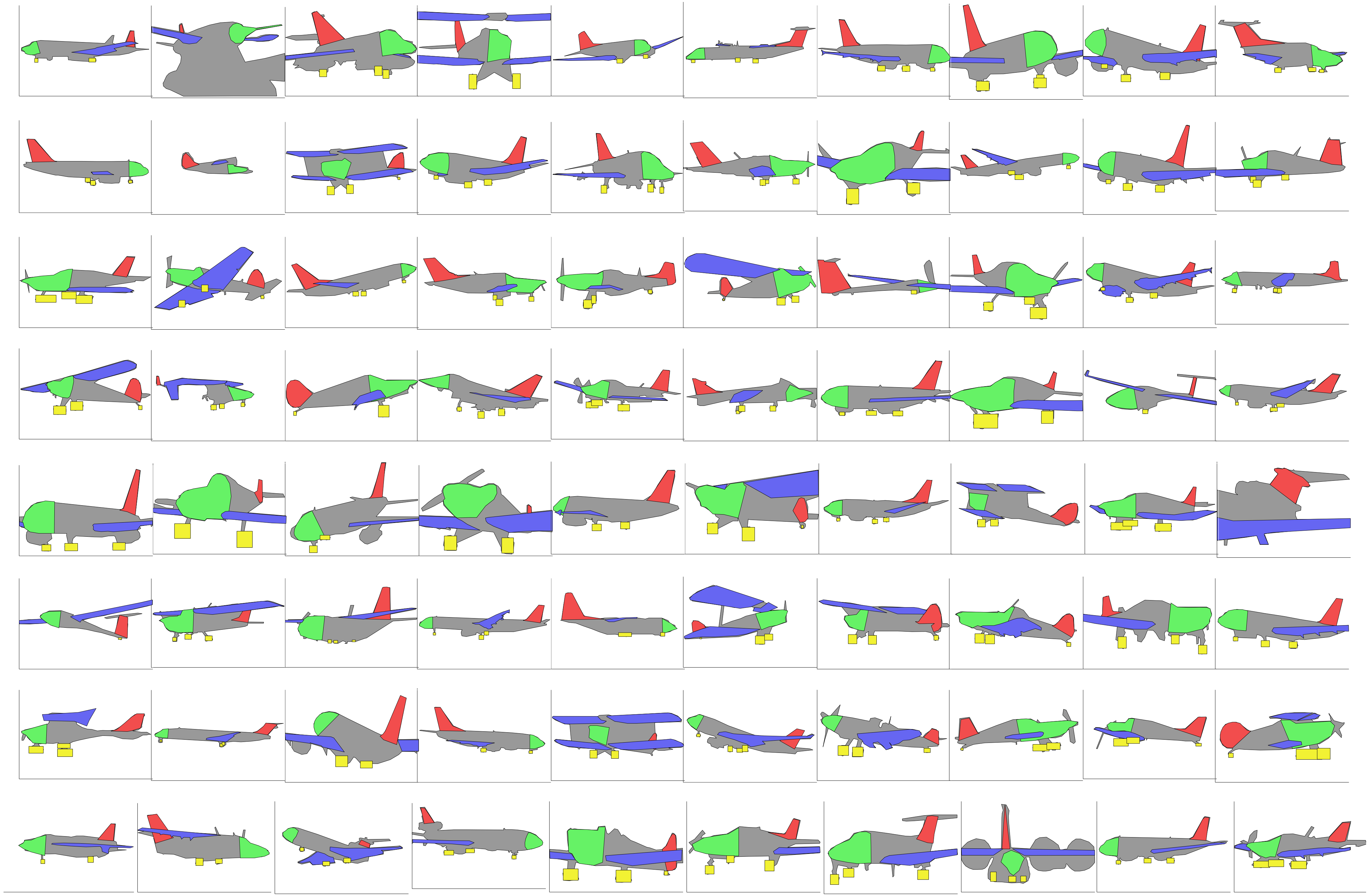
Part annotations



aeroplane | vertical stabiliser | nose | wheel | wing

Part	# train	# val	# test	# total
aeroplane	1,859	1,854	3,713	7,426
vertical stabiliser	1,885	1,866	3,742	7,493
nose	1,848	1,845	3,700	7,393
wing	3,007	3,047	5,958	12,012
wheel	4,919	4,958	9,917	19,794

Examples



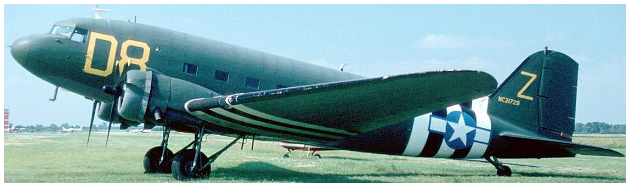
Attribute annotations

Part	Attribute	Values
aeroplane	airline	2Excel Aviation,ACE Transvalair,ATA Airlines,ATE Avia
aeroplane	model	AESL Airtourer T2,AESL Airtourer T5 Super 150,AESL Gl
aeroplane	isAirliner	yes,no
aeroplane	isCargoPlane	yes,no
aeroplane	isMilitaryPlane	yes,no
aeroplane	isPropellorPlane	yes,no
aeroplane	isSeaPlane	yes,no
aeroplane	facingDirection	E,SE,S,SW,W,NW,N,NE
aeroplane	planeLocation	on ground/water,landing/taking off,in air
aeroplane	planeSize	small plane,medium plane,large plane
wing	wingType	single wing plane,bi-plane,tri-plane
wing	wingHasEngine	1-on-bottom,1-on-top,2-on-bottom,2-on-top,3-on-bottom
vertical stabilizer	tailHasEngine	1-middle-top,2-on-sides,3-on-top-and-sides,no-engine
nose	noseHasEngineOrAntenna	has-antenna,has-engine,none
wheel	undercarriageArrangement	not visible,one-front-two-back,other,two-front-one-ba
wheel	coverType	fixed-inside,fixed-outside,fixed-outside-with-cover,r
wheel	groupType	1-wheel-1-axle,14-wheels-7-axles,2-wheels-1-axle,4-wh
wheel	location	back-left,back-middle,back-right,front-left,front-mic

is airliner: yes

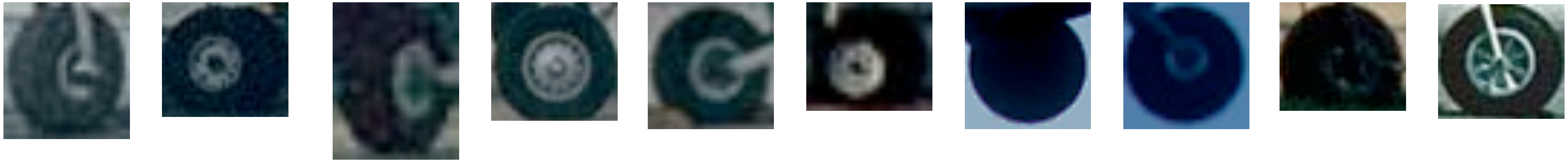
is airliner: no

is military plane: yes



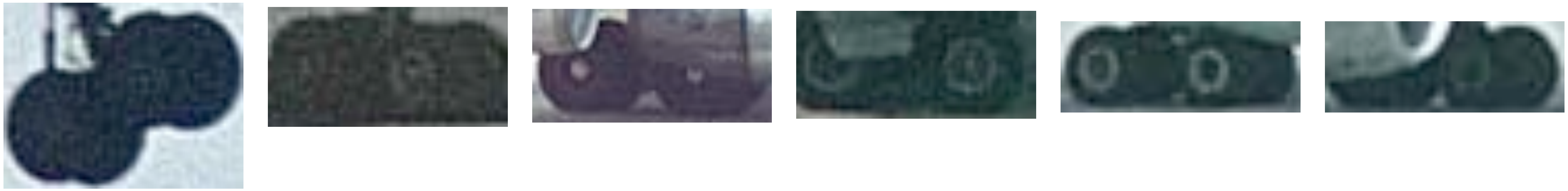
is sea plane: yes

wheel - group type: 1-wheel-1-axle

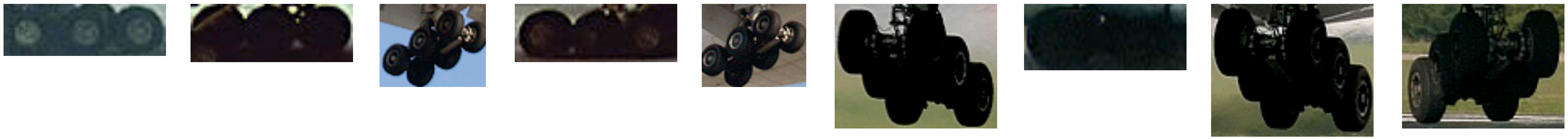


wheel - group type: 2-wheels-1-axle

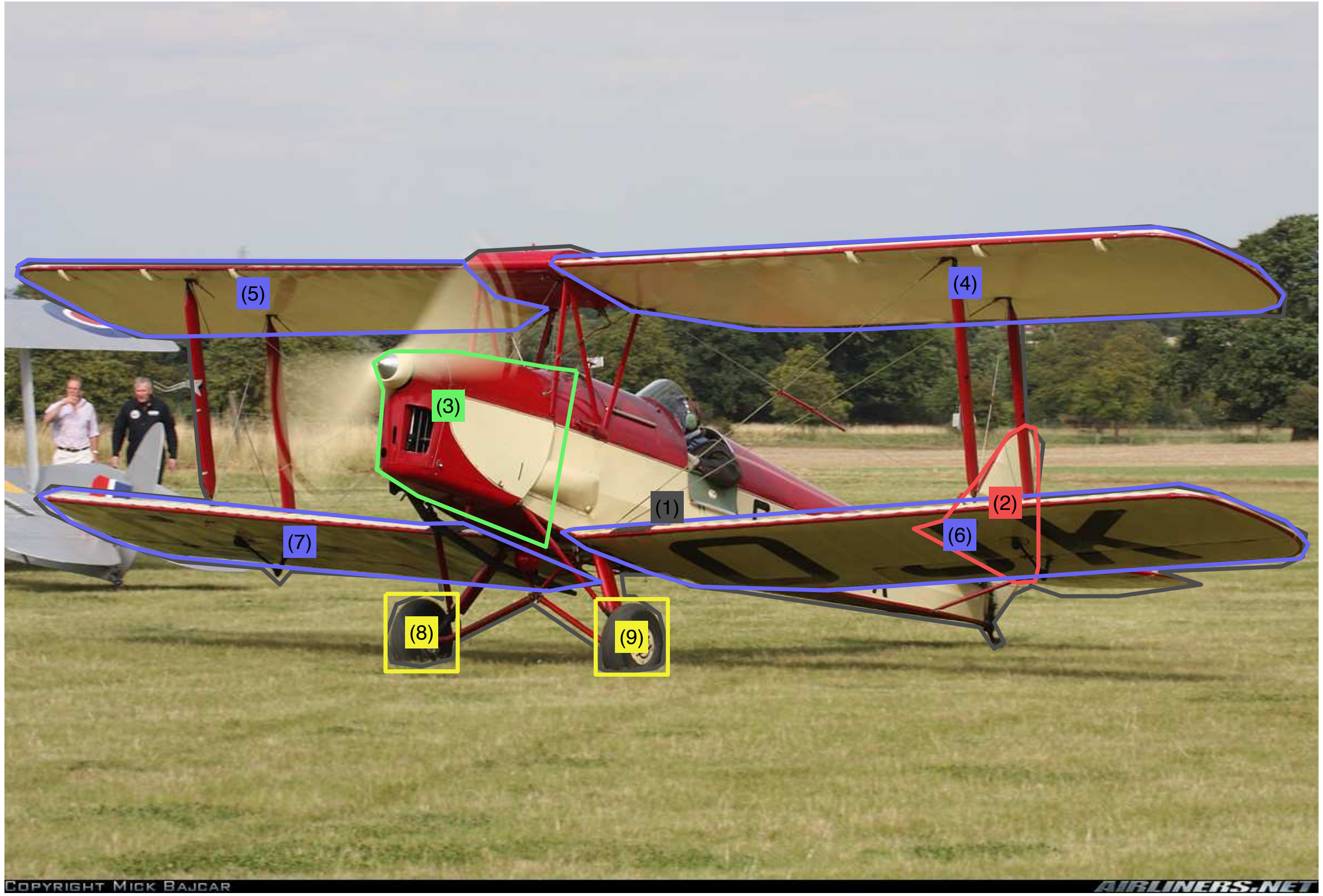
wheel - group type: 4-wheels-2-axles



wheel - group type: 6-wheels-3-axles



Complete annotation examples



(1) aeroplane
- isAirliner:no (0.8)
- isCargoPlane:no (1.0)
- isMilitaryPlane:no (0.8)
- isPropellorPlane:yes (1.0)
- isSeaPlane:no (1.0)
- facingDirection:SW (0.8)
- planeLocation:on ground/water (1.0)
- planeSize:small plane (0.6)
- wingType:bi–plane (1.0)
- undercarriageArrangement:two–front–one–back (1.0)
- noseHasEngineOrAntenna:has–engine (1.0)
- tailHasEngine:no–engine (1.0)
- wingHasEngine:no–engine (1.0)
- airline:none
- model:De Havilland DH–82A Tiger Moth II

(2) verticalStabilizer

(3) nose

(4) wing

(5) wing

(6) wing

(7) wing

(8) wheel
- coverType:fixed–outside (1.0)
- groupType:1–wheel–1–axle (1.0)
- location:front–right (0.8)

(9) wheel
- coverType:fixed–outside (1.0)
- groupType:1–wheel–1–axle (1.0)
- location:front–left (0.8)

(1) aeroplane
isAirliner:yes (1.0)
isCargoPlane:no (0.8)
isMilitaryPlane:no (1.0)
isPropellorPlane:no (1.0)
isSeaPlane:no (1.0)
facingDirection:NW (0.4)
planeLocation:landing/taking off (0.6)
planeSize:large plane (1.0)
wingType:single wing plane (1.0)
undercarriageArrangement:one–front–two–back (0.6)
noseHasEngineOrAntenna:none (0.8)
tailHasEngine:no–engine (0.6)
wingHasEngine:1–on–bottom (1.0)
airline:Monarch Airlines
model:Airbus A300B4–605R

(2) verticalStabilizer

(3) nose

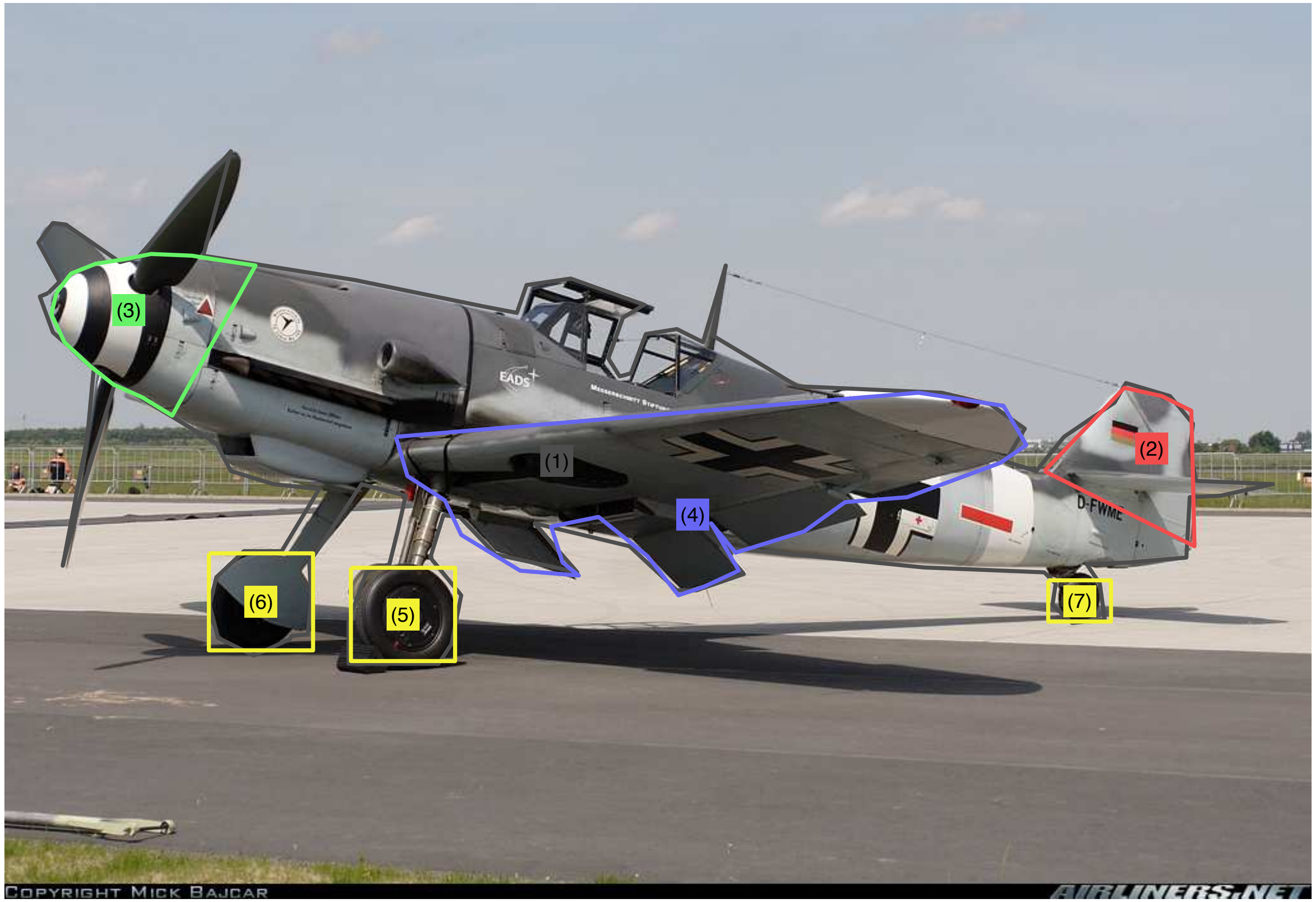
(4) wing

(5) wing

(6) wheel
coverType:retractable (0.8)
groupType:2–wheels–1–axle (0.8)
location:back–left (0.6)

(7) wheel
coverType:retractable (0.6)
groupType:2–wheels–1–axle (0.8)
location:back–right (0.4)

(1) aeroplane
isAirliner:no (1.0)
isCargoPlane:no (1.0)
isMilitaryPlane:no (0.8)
isPropellorPlane:yes (1.0)
isSeaPlane:no (0.6)
facingDirection:E (0.8)
planeLocation:on ground/water (0.8)
planeSize:small plane (0.8)
wingType:single wing plane (1.0)
undercarriageArrangement:two–front–one–back (1.0)
noseHasEngineOrAntenna:has–engine (1.0)
tailHasEngine:no–engine (1.0)
wingHasEngine:no–engine (1.0)
airline:2Excel Aviation
model:Extra EA–300L
(2) verticalStabilizer
(3) nose
(4) wing
(5) wheel
coverType:fixed–outside–with–cover (1.0)
groupType:1–wheel–1–axle (1.0)
location:front–right (1.0)
(6) wheel
coverType:fixed–outside–with–cover (1.0)
groupType:1–wheel–1–axle (1.0)
location:front–left (1.0)
(7) wheel
coverType:fixed–outside (1.0)
groupType:1–wheel–1–axle (1.0)
location:back–middle (1.0)



(1) aeroplane
isAirliner:no (0.6)
isCargoPlane:no (1.0)
isMilitaryPlane:yes (0.8)
isPropellorPlane:yes (1.0)
isSeaPlane:no (0.8)
facingDirection:W (0.6)
planeLocation:on ground/water (1.0)
planeSize:small plane (0.6)
wingType:single wing plane (1.0)
undercarriageArrangement:two–front–one–back (1.0)
noseHasEngineOrAntenna:has–engine (1.0)
tailHasEngine:no–engine (0.8)
wingHasEngine:no–engine (0.8)
airline:none
model:Messerschmitt Bf–109G–4

(2) verticalStabilizer

(3) nose

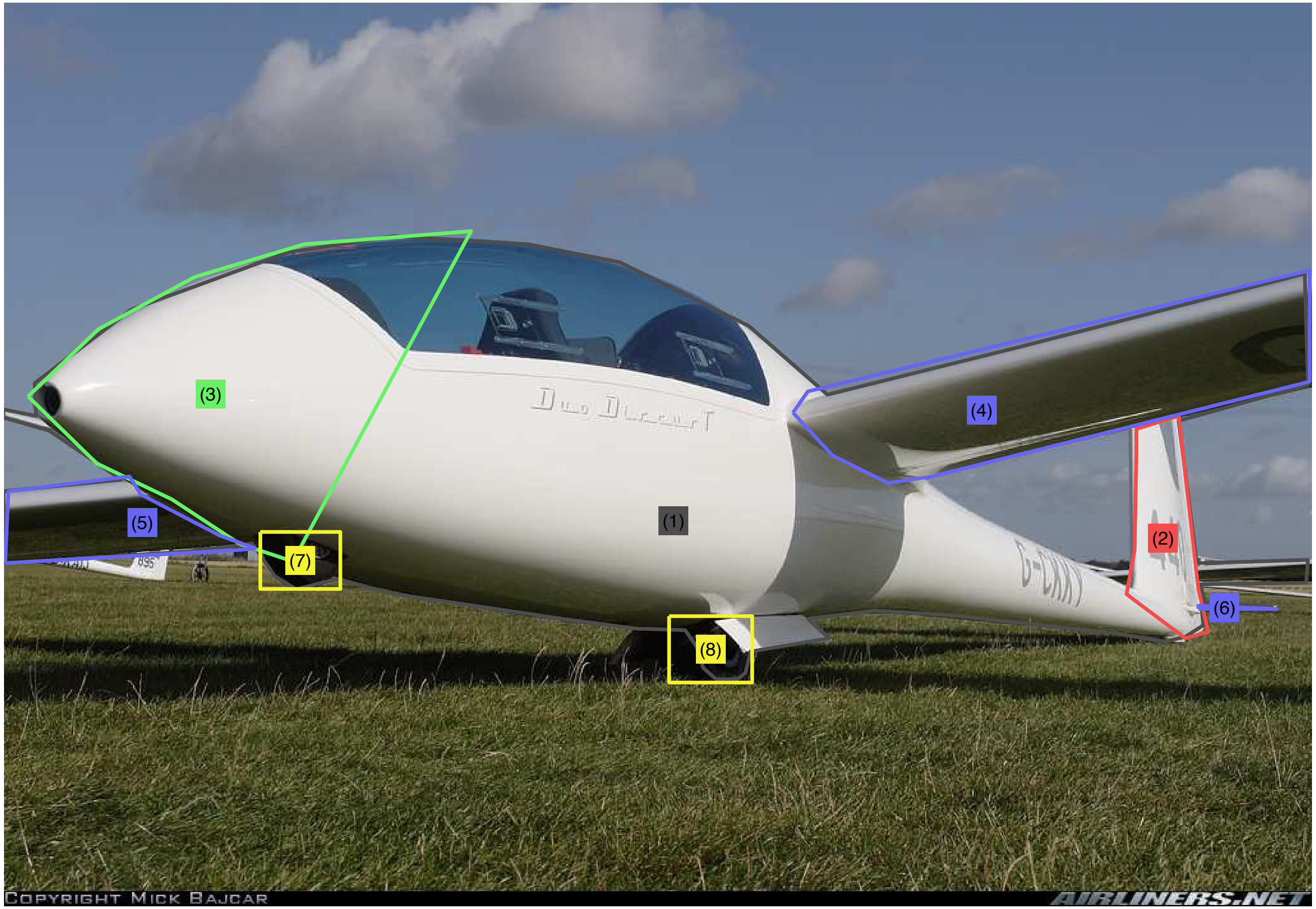
(4) wing

(5) wheel
coverType:retractable (0.6)
groupType:1–wheel–1–axle (1.0)
location:front–left (1.0)

(6) wheel
coverType:retractable (0.6)
groupType:1–wheel–1–axle (1.0)
location:front–right (1.0)

(7) wheel
coverType:retractable (0.8)
groupType:1–wheel–1–axle (1.0)
location:back–middle (1.0)

(1) aeroplane
isAirliner:yes (0.8)
isCargoPlane:no (1.0)
isMilitaryPlane:no (0.8)
isPropellorPlane:no (1.0)
isSeaPlane:no (1.0)
facingDirection:SW (1.0)
planeLocation:on ground/water (1.0)
planeSize:large plane (1.0)
wingType:single wing plane (1.0)
undercarriageArrangement:not visible (0.6)
noseHasEngineOrAntenna:none (1.0)
tailHasEngine:3–on–top–and–sides (0.6)
wingHasEngine:no–engine (0.8)
airline:British Airways
model:Hawker Siddeley HS–121 Trident 3B
(2) verticalStabilizer
(3) wing



(1) aeroplane
isAirliner:no (0.8)
isCargoPlane:no (1.0)
isMilitaryPlane:no (1.0)
isPropellorPlane:no (1.0)
isSeaPlane:no (1.0)
facingDirection:SW (0.8)
planeLocation:on ground/water (0.8)
planeSize:small plane (0.8)
wingType:single wing plane (0.8)
undercarriageArrangement:other (0.6)
noseHasEngineOrAntenna:none (1.0)
tailHasEngine:no–engine (1.0)
wingHasEngine:no–engine (1.0)
airline:none
model:Schempp–Hirth Duo Discus T
(2) verticalStabilizer
(3) nose
(4) wing
(5) wing
(6) wing
(7) wheel
coverType:fixed–inside (0.8)
groupType:1–wheel–1–axle (1.0)
location:front–middle (0.6)
(8) wheel
coverType:fixed–inside (0.6)
groupType:1–wheel–1–axle (1.0)
location:front–left (0.6)

Overview

Objects in Detail

Parts & attributes

- A new dataset
- An object lexicon
- Localising parts
- Layouts
- Recognising attributes

Stuff in Detail

Texture

- A texture lexicon
- A new dataset
- Transformation invariant semantic

Parsing

Bottom-up inference

- Learning to merge
- Cascading
- Scoring regions by attributes

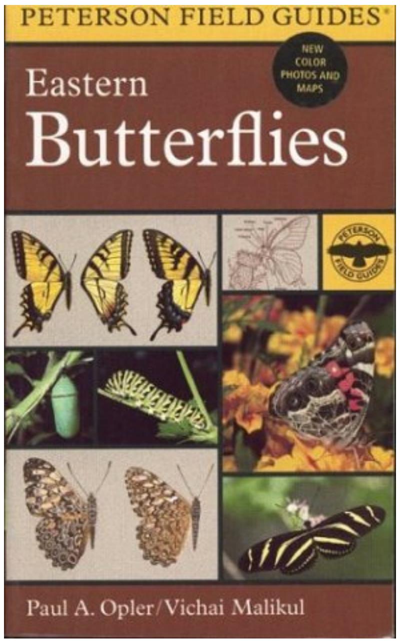
Lexicon of Parts and Attributes

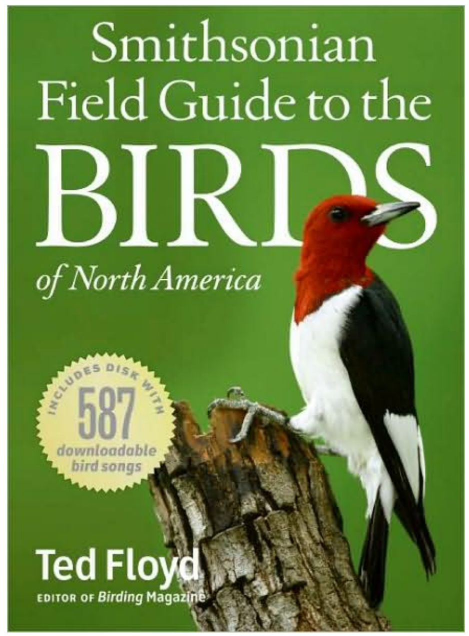
How do people describe objects?

Subhransu Maji
TTI Chicago

Source of Parts and Attribute Lexicons

- Field guides:
  - Provides exhaustive lists when available





Experts vs. Layman

Source of Parts and Attribute Lexicons

- Captioned images

Dazzle after dark with Judith Leiber's decadent oversized crystal-embellished silver-tone clutch. Carry this fabulous extra to add high-octane glamour to an LBD and teetering heels. Shown here with an Emilio Pucci dress and Givenchy shoes.

The 12K pink and green gold leaves gently cascade down on these delicate beaded 10K gold earrings.

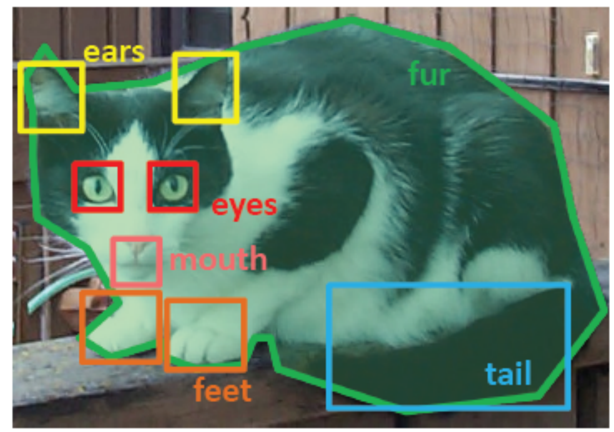
Rock and roll in these sexy, strappy heels from Report Signature. The smoldering Rockwell features a grey patent leather upper with pleated satin crossing at the open-toe atop a 1 inch platform, patent straps closing around the ankle with a gold buckled, and finally a 5 inch patent cone heel. Sizzle in these fierce mile-high shoes.

Limited by sources of such text
(not always visual attributes)

Berg et al., ECCV 2010

Parts and Attributes: Why?

High-heel Blue Shoe



Object Categories: animal, land animal, domestic, mammal, carnivore, cat

Viewpoint/pose: lying down, left side, facing camera

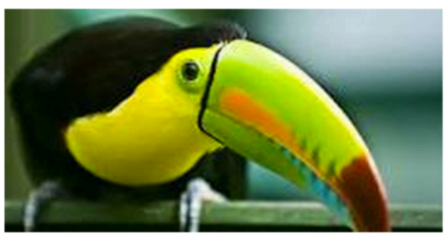
Other likely parts: four legs

Helps differentiate instances of an object

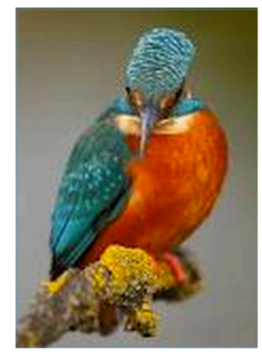
Communication requires a lexicon

What are good attribute lexicons?







Goals: Differentiation + Communication

Discriminative Description

Describe the (visual) differences between the two

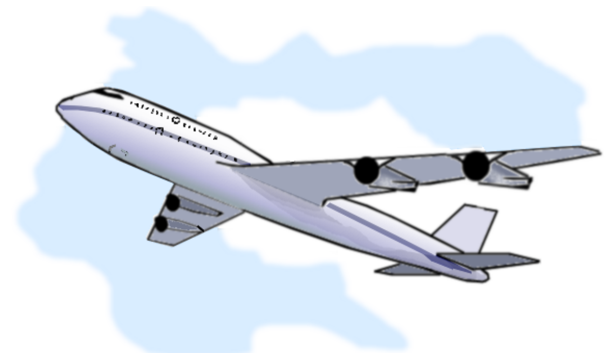
Description

list properties

- plane
- has engine
- red color
- has rudder

Discriminative Description

vs.



list differences

propeller plane vs. passenger plane
one engine vs. four engines
red color vs. white color
round rudder vs. pointy rudder

Helps elicit a lexicon that enables fine grained discrimination
Is task specific by design

The Annotation Task Interface

Find differences between the two aeroplanes

Click here to see example answers.

List 5 differences between the two *aeroplanes*

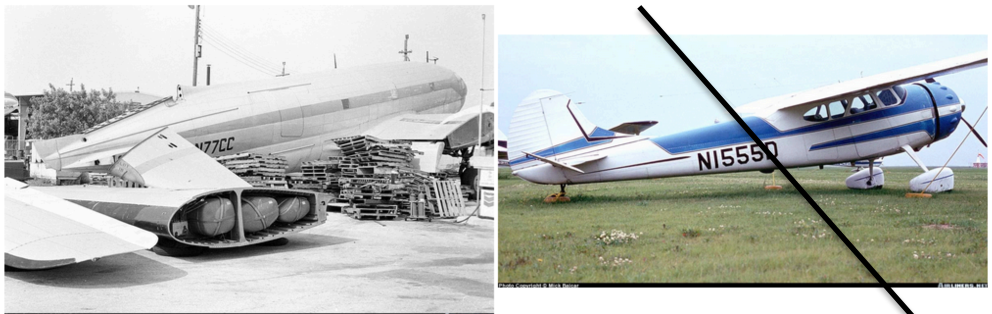
1.
2.
3.
4.
5.

Submit Answer

The Annotation Task Interface

Find differences between the two aeroplanes

Click here to see example answers.



List 5 differences between the two *aeroplanes*

1.
2.
3.
4.
5.

Submit Answer

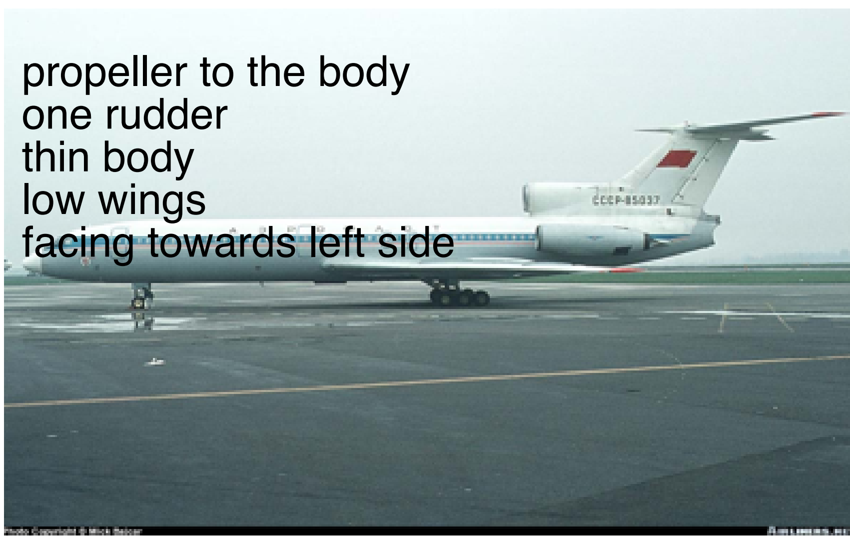
- yellow and red rudder vs blue rudder
- two wings vs one wing
- yellow body vs blue body
- facing right vs facing left

- left facing vs right facing
- red rudder vs white rudder
- passenger plane vs propeller plane
- wings near bottom vs wings on top
- two side engines vs one side engine

Outputs in free form English separated by ``vs”

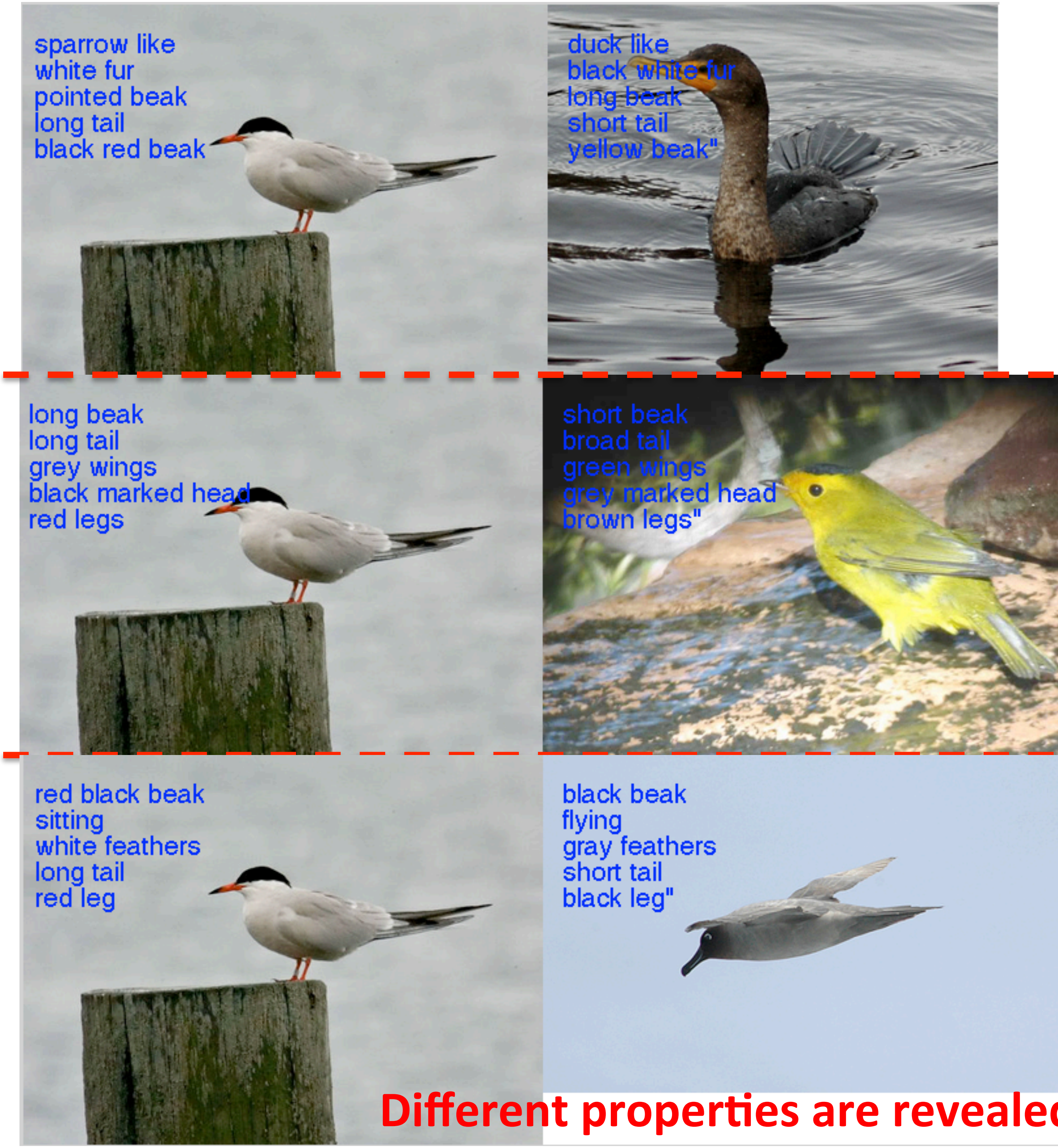
amazon mechanical turk
beta
Artificial Artificial Intelligence

Example Annotations



Example Annotations

Images are from CUB 200 dataset



Different properties are revealed in each pair

Frequencies of Properties

5: blue wings
4: black beak
4: long tail
3: blue head
3: short beak
2: blue white body
2: blue white fur
2: sharp beak
2: short tail
1: big body
1: black leggs
1: black legs
1: black wings
1: blue body
1: blue color head
1: gray body
1: long bird
1: orange white body
1: short wings
1: slim body
1: white and blue fur

Frequencies of Properties

4: yellow head
3: short beak
2: fat body
2: long tail
2: short tail
2: small tail
1: big body
1: black sharp beaks
1: black wings
1: broad body
1: brown wing
1: grey yellow body
1: having tail
1: mixed yellow body
1: multi colour
1: orange white body
1: point beak
1: small beak
1: small size
1: small wings
1: thick fur
1: very fat body
1: wings
1: yellow and brown color
1: yellow black wings
1: yellow color head
1: yellow color wing
1: yellow feather
1: yellow fur
1: yellow neck

Discovering parts & attribute lexicons

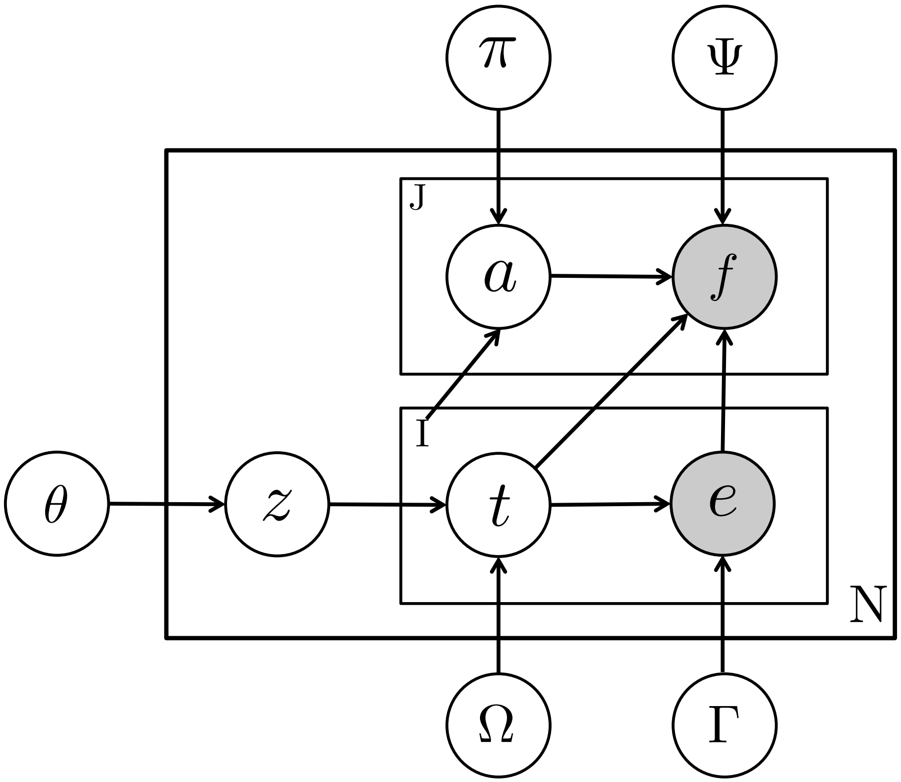
list differences

Red **rudder** vs. *White* **rudder**
Pointy **nose** vs. *Round* **nose**

{*Red, White*} → Color
{Pointy, Round} → Shape

- Analyzing sentence pairs
  - Words that *repeat* across a sentence pair are parts (nouns)
  - Words that are *different* across a sentence pair are from the same semantic *modifier* category
  - Each sentence has only one noun and modifier topic

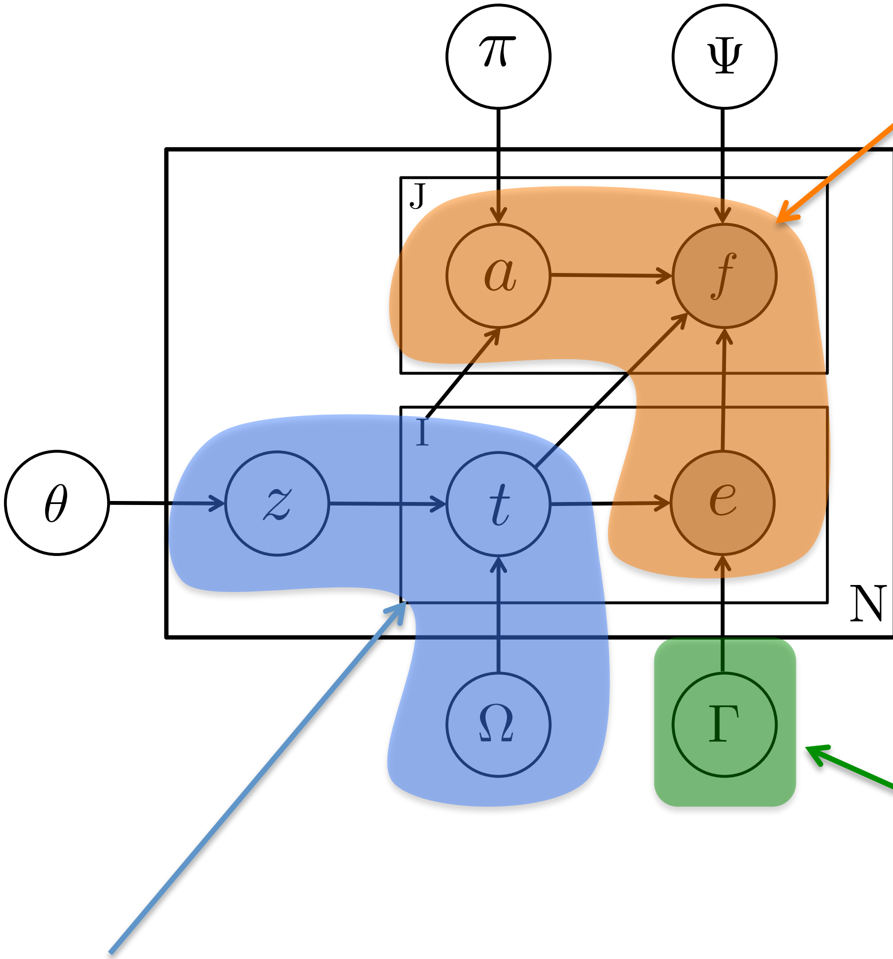
Bipartite Topic Translation Model for Sentence Pairs



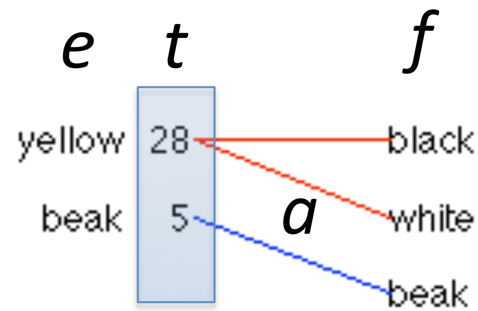
For each sentence pair $\mathbf{e}_s, \mathbf{f}_s$, $s \in \{1, \ldots, N\}$

- Sample relation $z_s \sim Multinomial(\theta)$
- For each word position $i \in 1, \ldots, I_s$ in $\mathbf{e_s}$
  - Sample topic $t_{s,i} \sim Multinomial\left(\Omega_{z_s}\right)$
  - Sample word $e_{s,i} \sim Multinomial\left(\Gamma_{t_{s,i}}\right)$
- For each word position $j = \{1, \ldots, J_s\}$ in $\mathbf{f_s}$
  - Sample $a_j \in \{1, \ldots, I\} \propto \pi(|a_j - j|)$,
  - Sample word $f_{s,j} \sim Multinomial\left(\Psi_{e_{a_j}, t_{a_j}}\right)$

Bipartite Topic Translation Model for Sentence Pairs



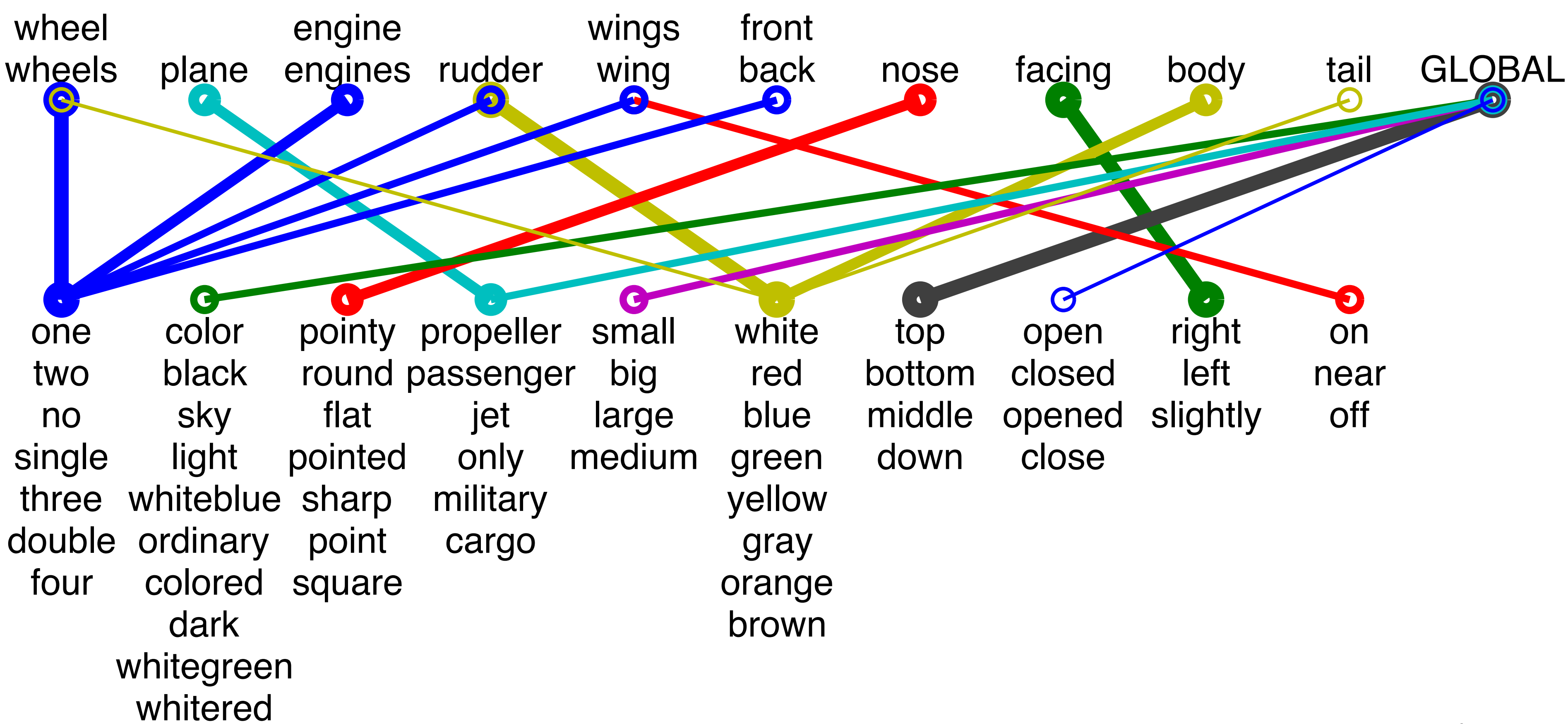
Topic specific word-word translation



Topics encode parts and modifiers

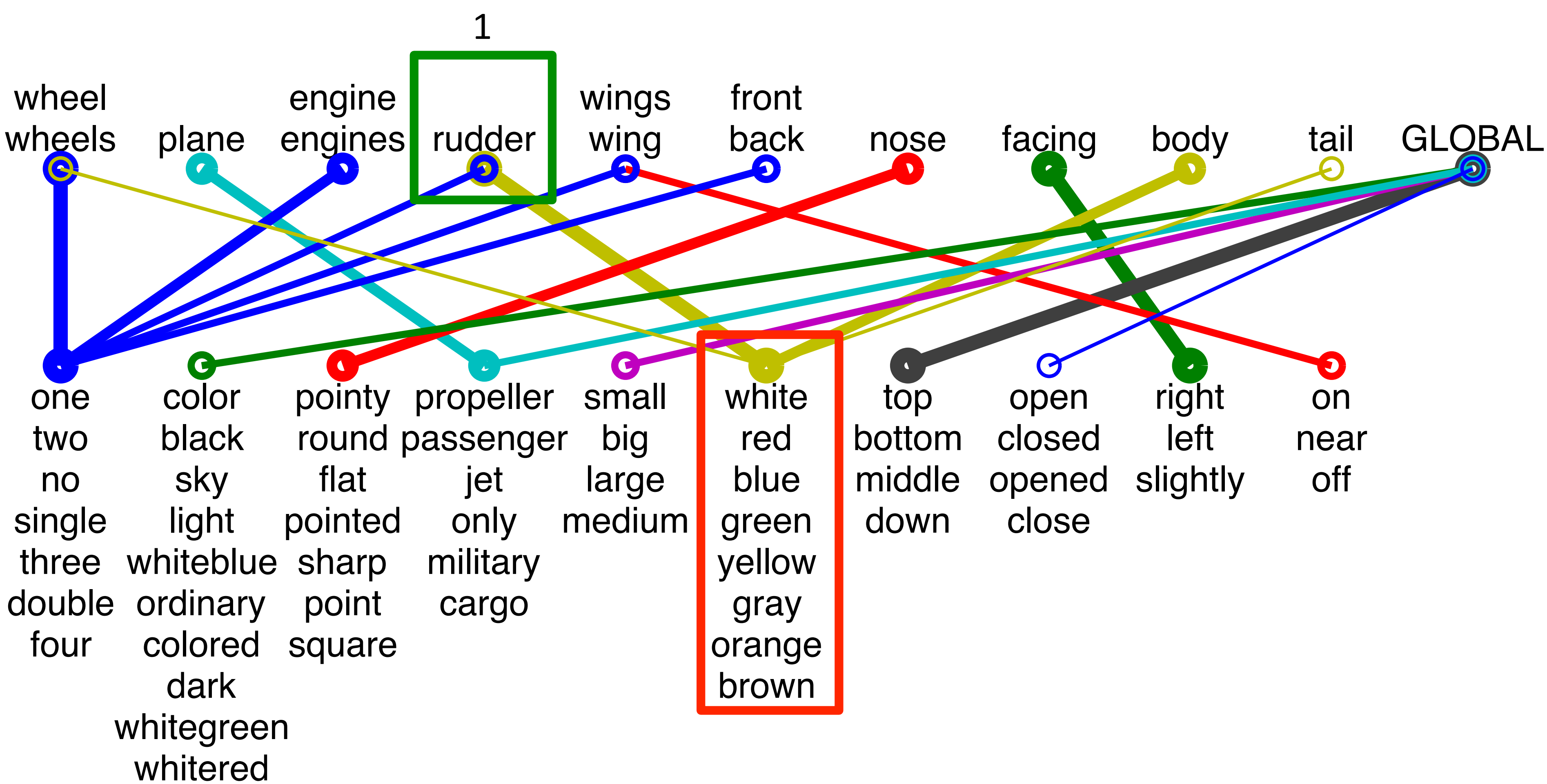
Bipartite topics: Each sentence has one noun and one modifer topic

Parts & Attributes of Planes

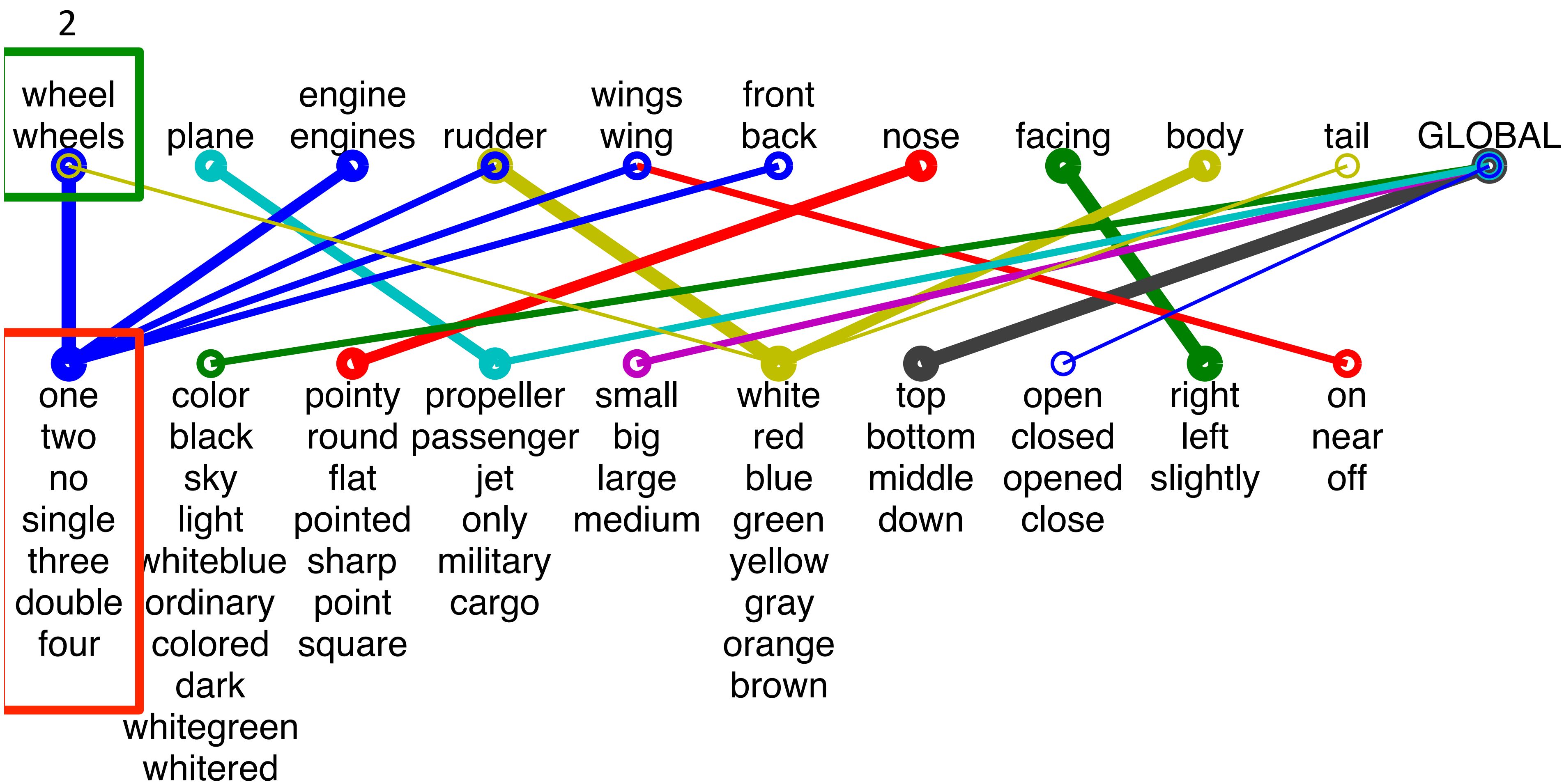


200 images, 1000 pairs, 1c/pair

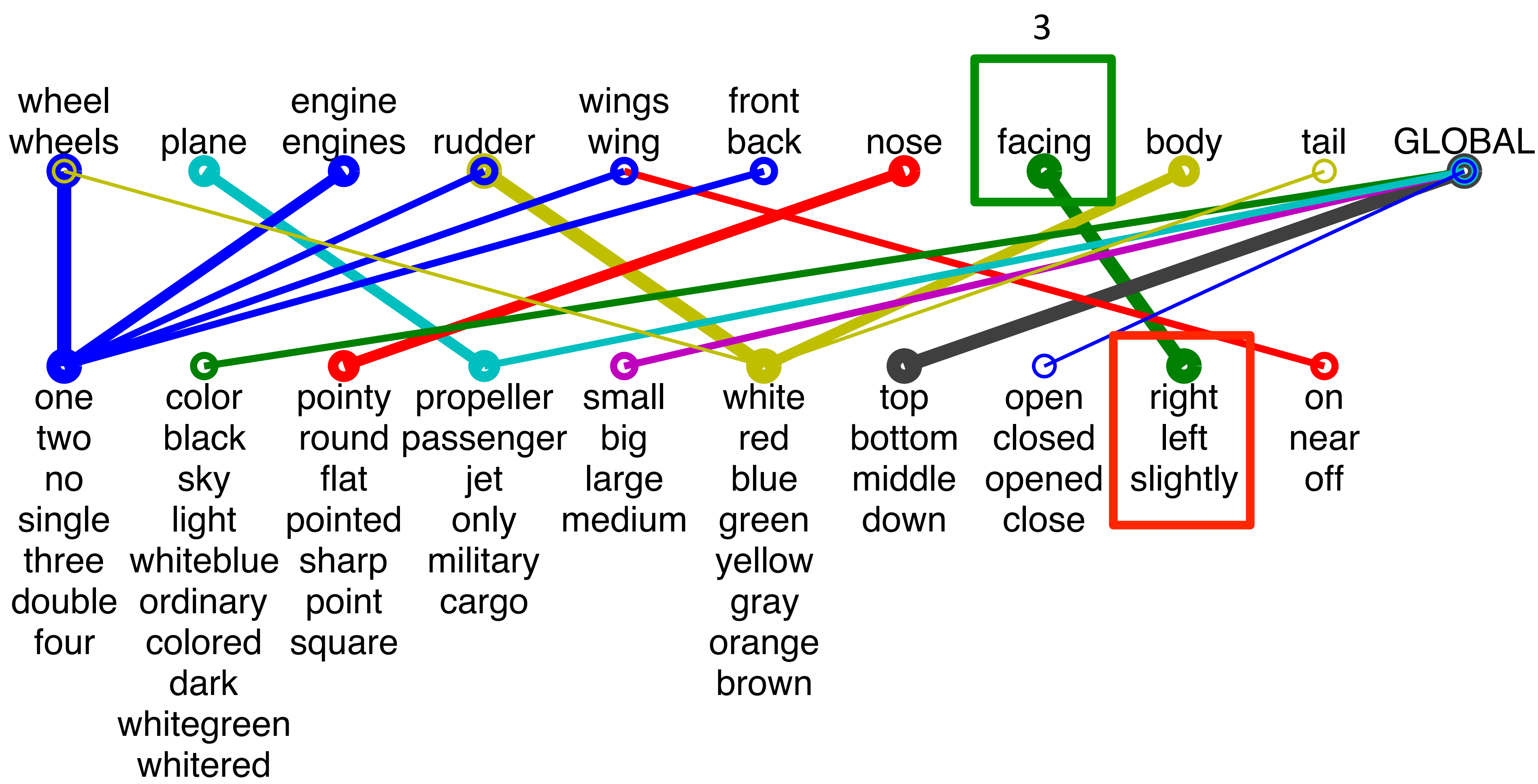
Parts & Attributes of Planes



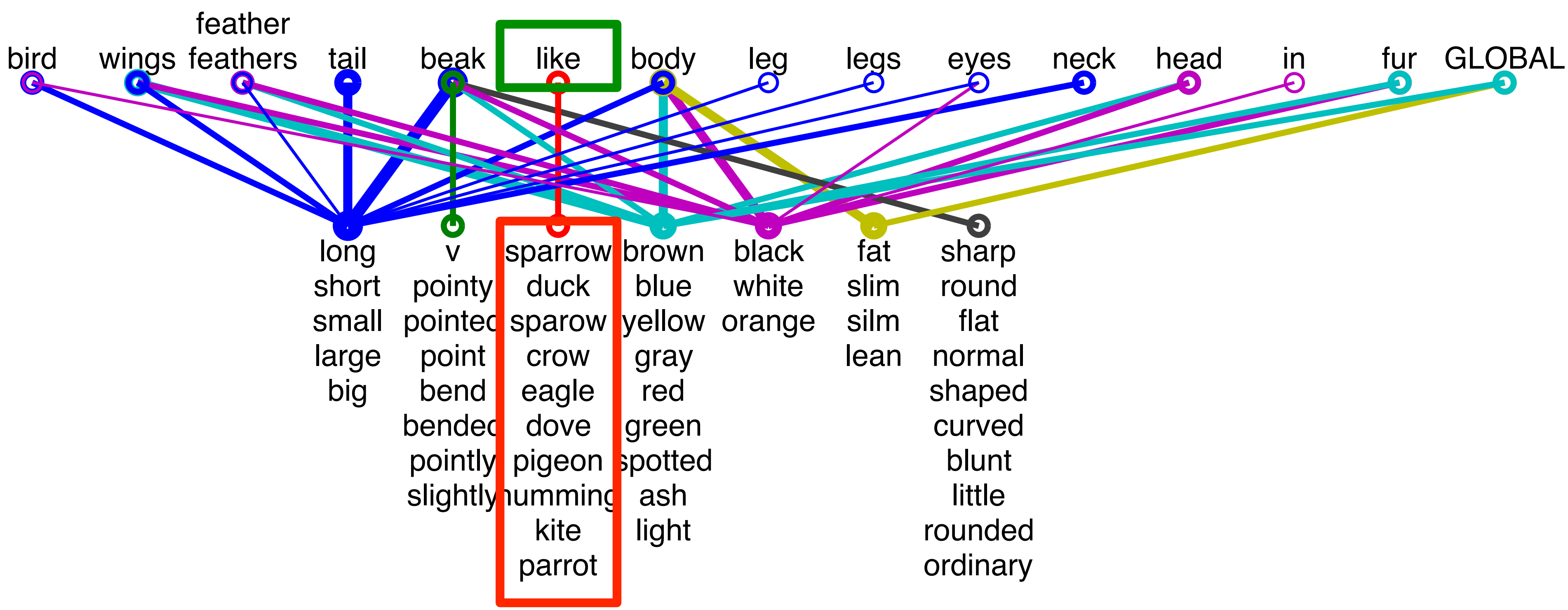
Parts & Attributes of Planes



Parts & Attributes of Planes



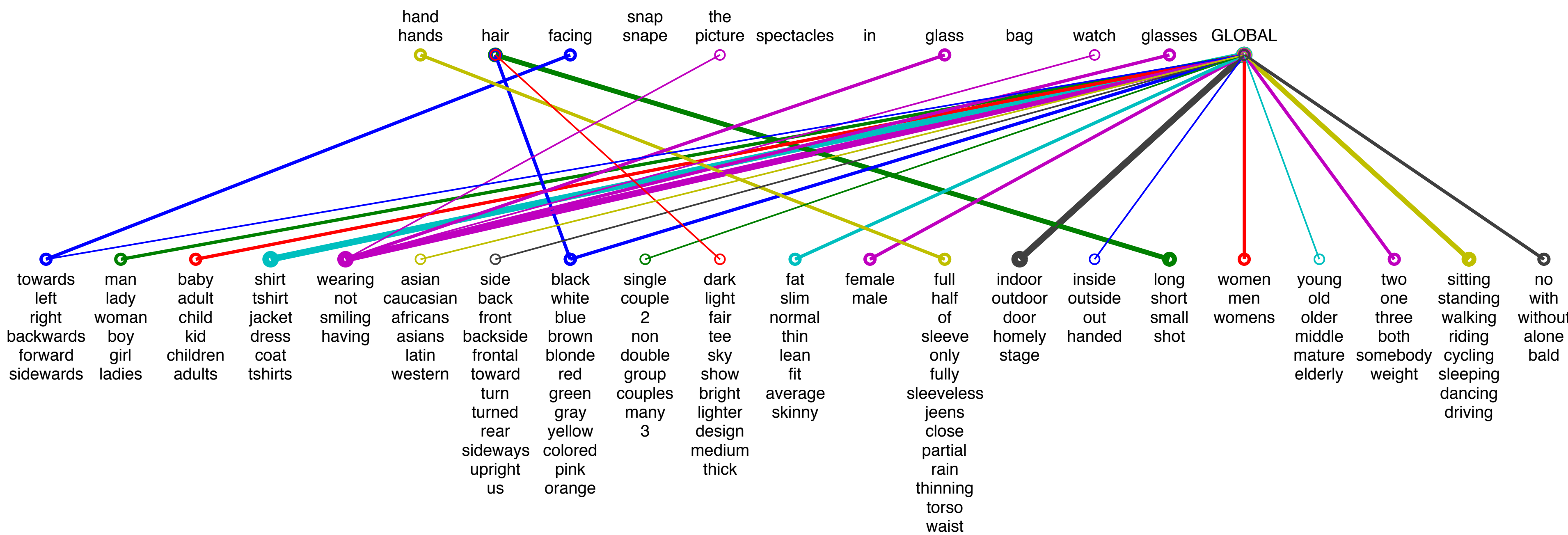
Attributes of Birds



200 images, 1600 pairs, 1c/pair

One image per category CUB200

Parts & Attributes of People



400 images, 1600 pairs, 1c/pair

random images from PASCAL VOC 10

Summary

- Discriminative description is an effective way to obtain a lexicon of parts and attributes that are useful for fine-grained discrimination
- Simple analysis of such text can help discover topics that encode parts, modifiers and their relations.

Overview

Objects in Detail

Parts & attributes

- A new dataset
- An object lexicon
- Localising parts
- Layouts
- Recognising attributes

Stuff in Detail

Texture

- A texture lexicon
- A new dataset
- Transformation invariant semantic

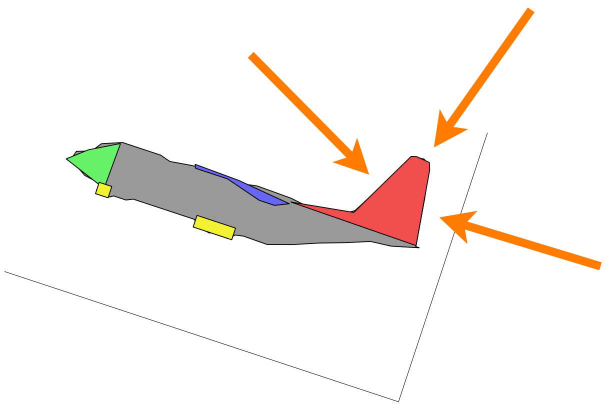
Parsing

Bottom-up inference

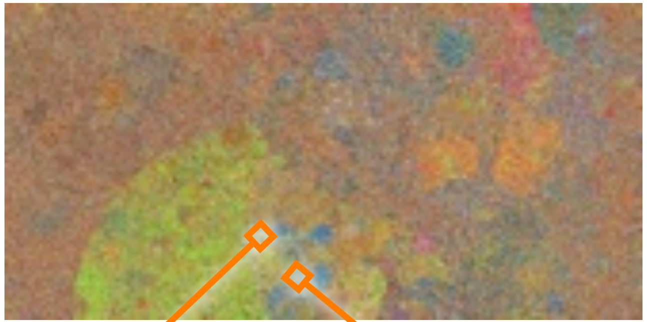
- Learning to merge
- Cascading
- Scoring regions by attributes

Localizing parts

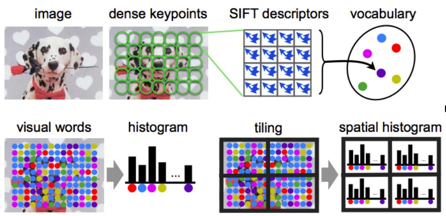
Ross Girshick
University of Chicago



Find airplanes with propellors on their noses



Not on nose!
Confusing occlusion
Context?



Coarse model
(e.g., BoW black box)

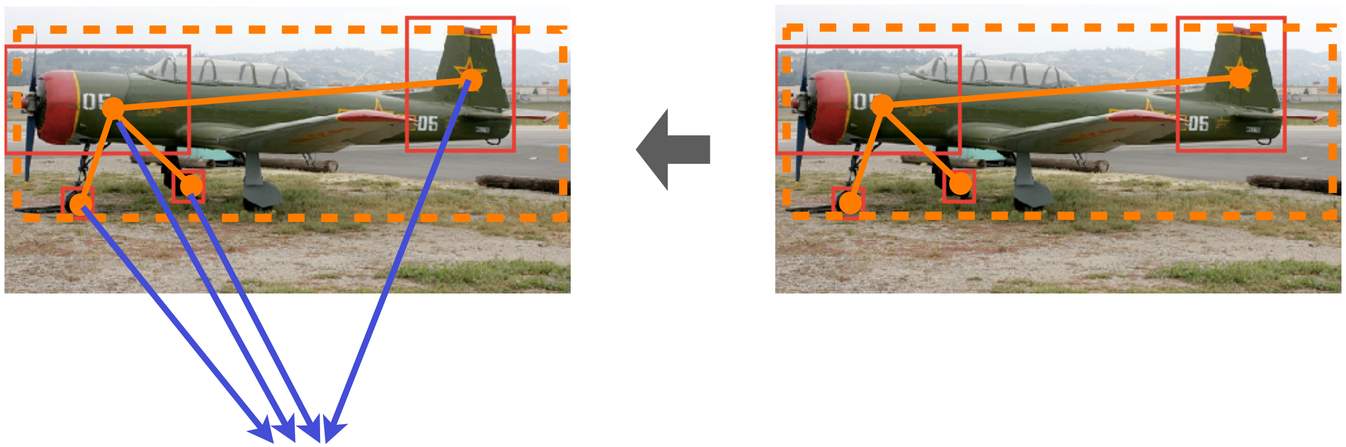
Use parts to align vision models with language

Overview of approach

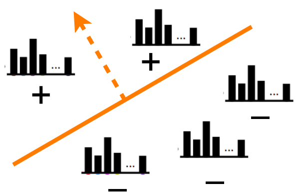
Q: Propellor on nose?

1. Candidate part detections (this talk)

2. Consistent layout generation (next talk)



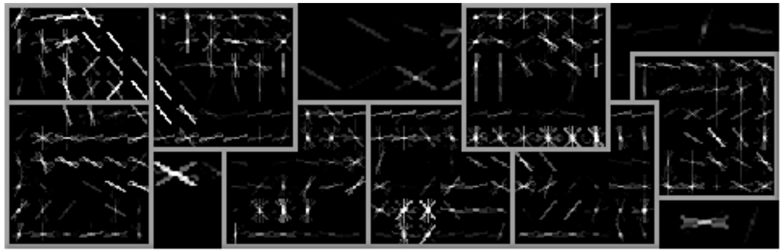
3. Extract *semantically aligned* features ...



... A: Yes

Why semantic parts?

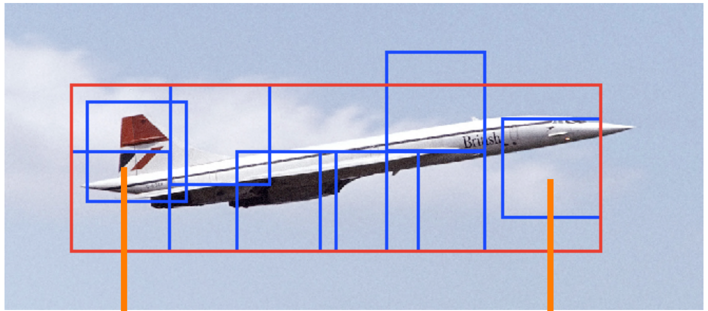
Deformable parts model



Structured, but not aligned
(parts learned without supervision)

Object detection

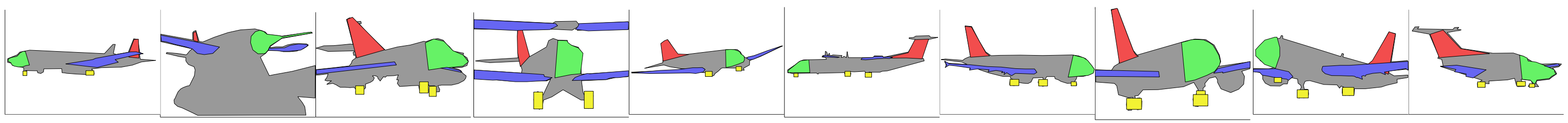
Detection



Vertical stabilizer
(but we don’t know that!)

Nose?

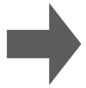
- Without semantic parts
  - the semantic alignment is unknown or nonexistent
    - *show me the vertical stabilizer*
  - no ground-truth for debugging performance bottlenecks
    - *are the part detectors failing?* *is the spatial model too rigid?*

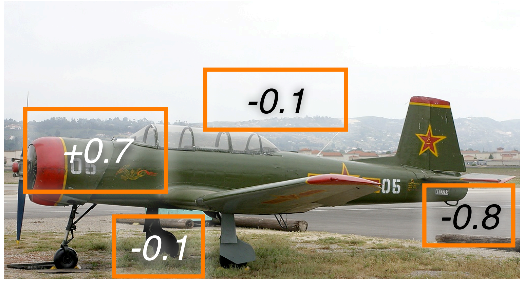


Part detection: evaluation metric

- Task: predict part bounding boxes

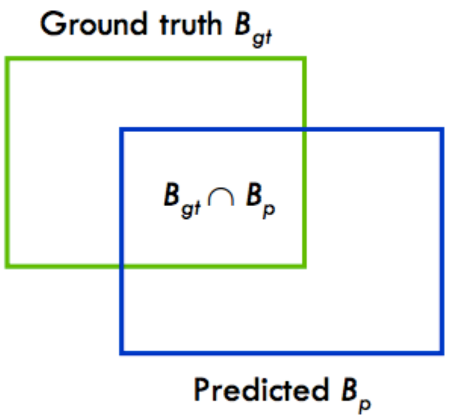
Test image



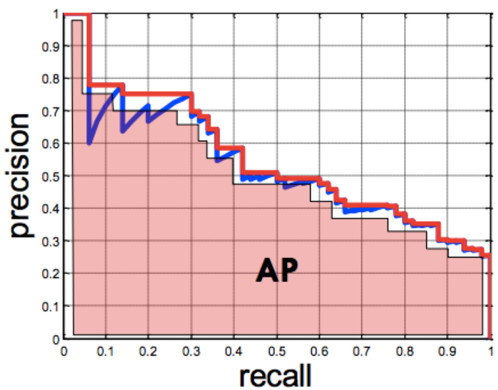


Scored candidate detections

- PASCAL VOC Challenge evaluation
  - Sort candidate detections by confidence score
  - Grade each as true positive or false positive (overlap ≥ 0.5)
  - Precision-recall curve & average precision (AP)

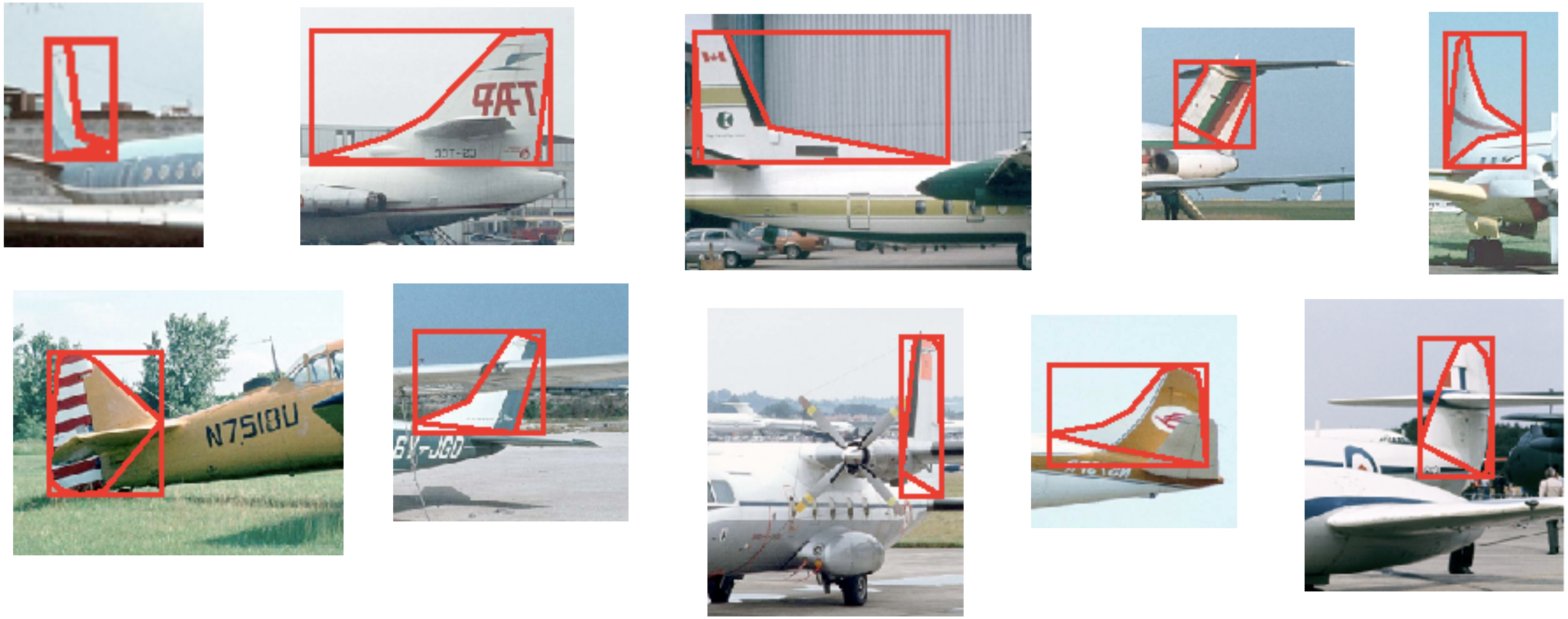


$$\mathrm{overlap}(B_{\mathrm{gt}}, B_{\mathrm{p}}) = \frac{|B_{\mathrm{gt}} \cap B_{\mathrm{p}}|}{|B_{\mathrm{gt}} \cup B_{\mathrm{p}}|}$$



Training data

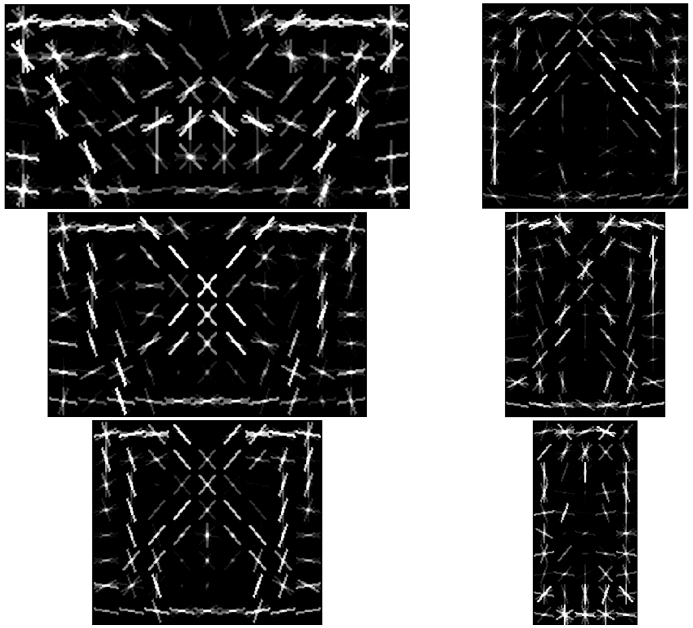
Vertical stabilizers



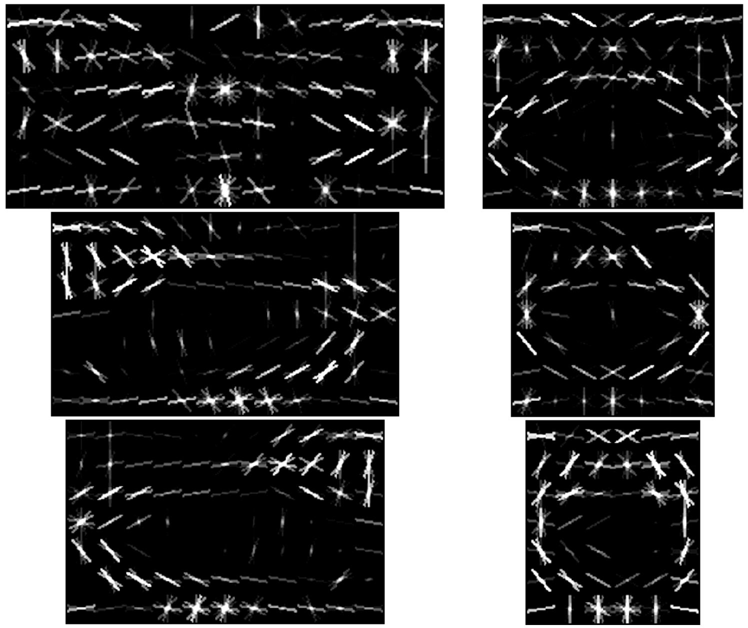
Noses

Baseline part detector

- Model: mixture of filters on gradient orientation (HOG) features



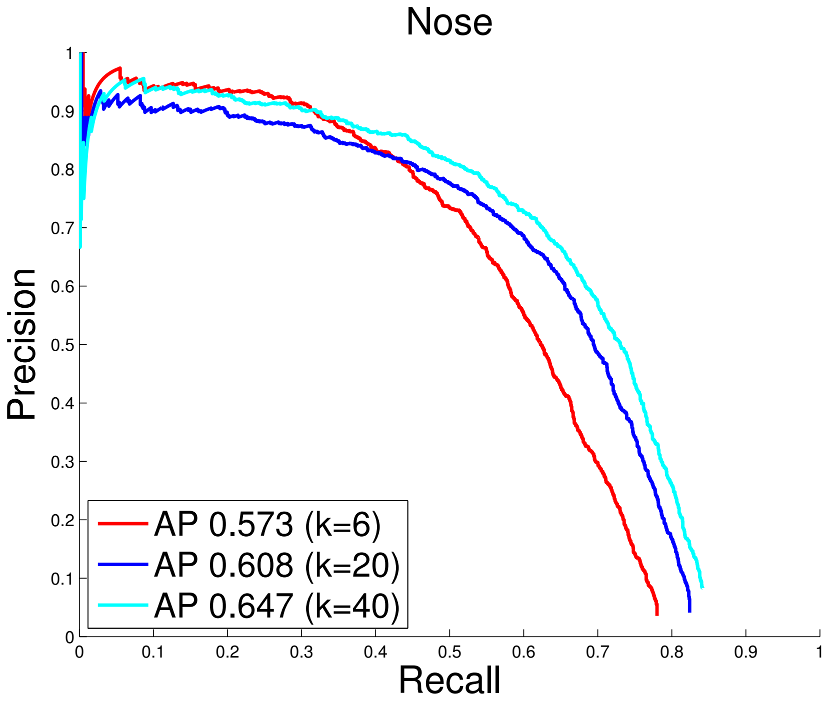
Vertical stabilizer (k=6)

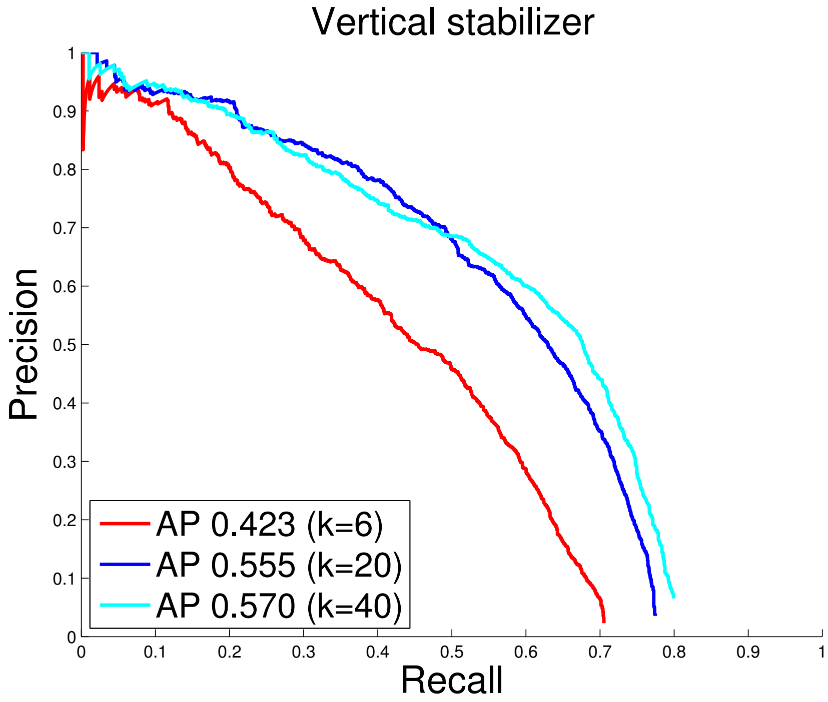


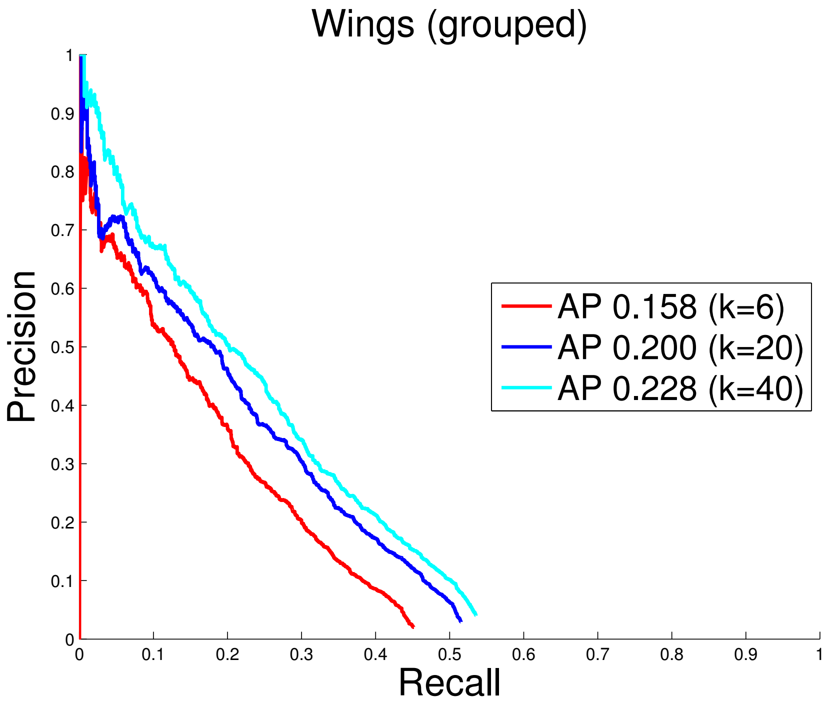
Nose (k=6)

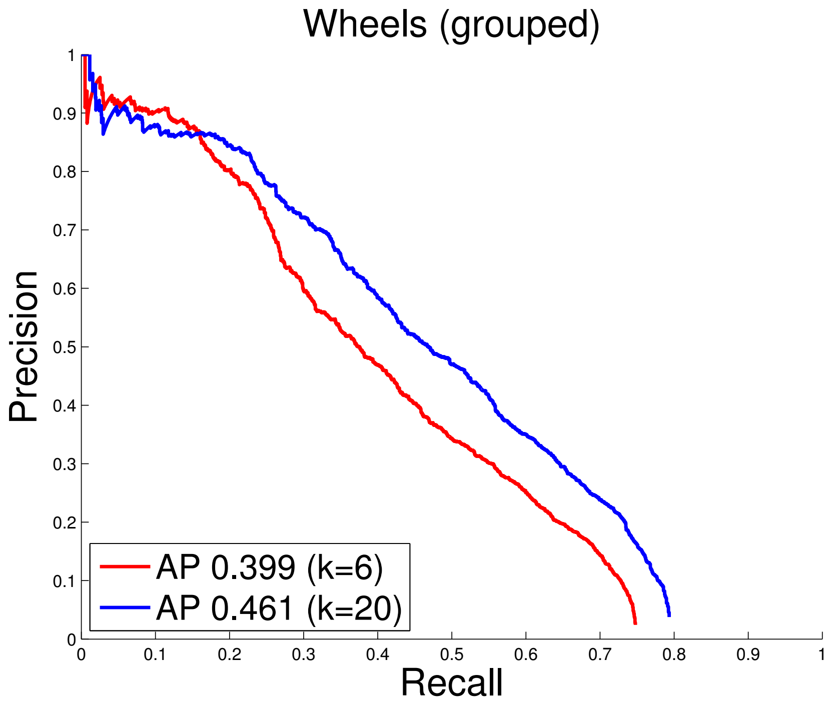
- Weak supervision (bounding box only; position, scale, mixture all *latent*)
- Trained with latent SVM
  - mixtures initialized by aspect ratio clustering

Baseline part detector results



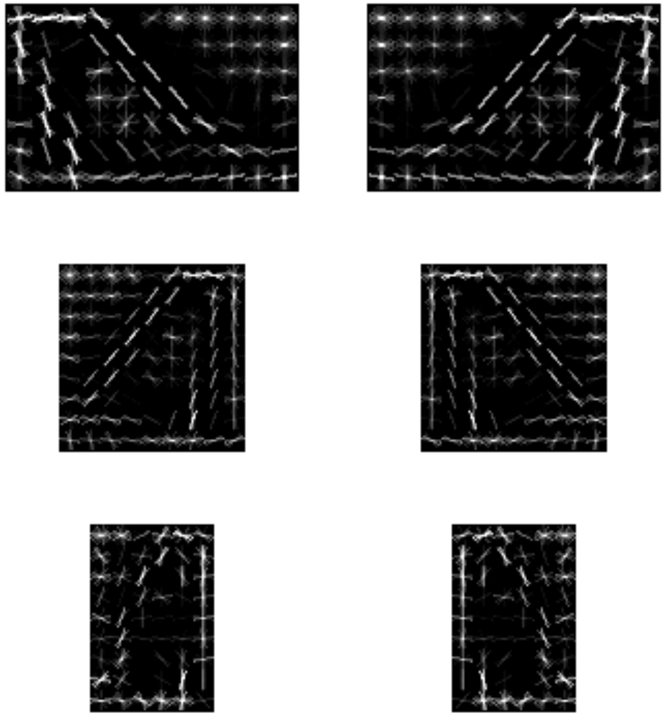




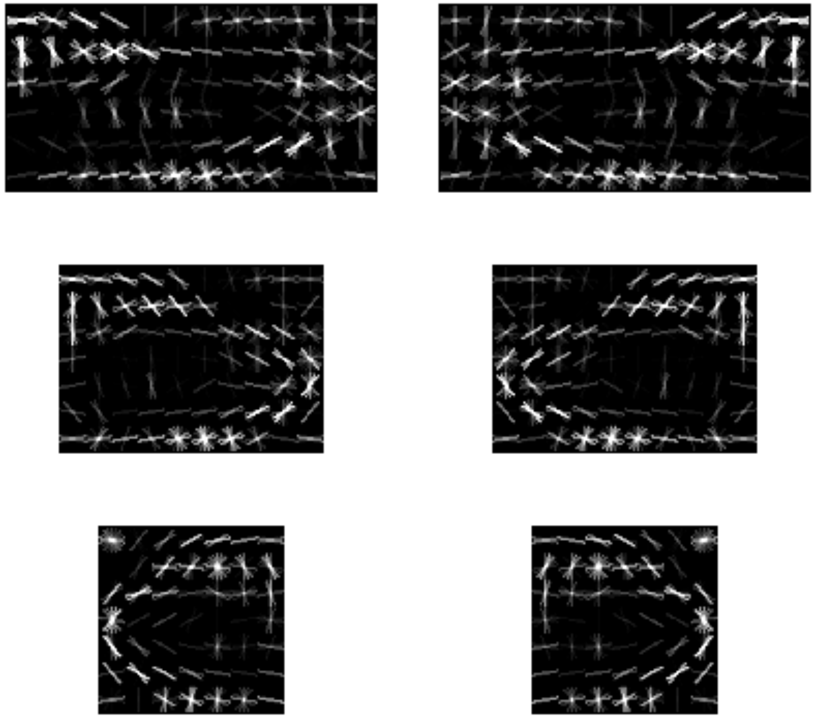


Improving part detectors

- **Method I:** unsupervised left *vs* right orientation clustering



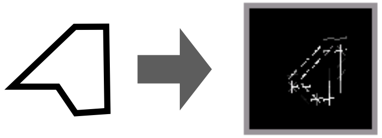
Vertical stabilizer (k=6)

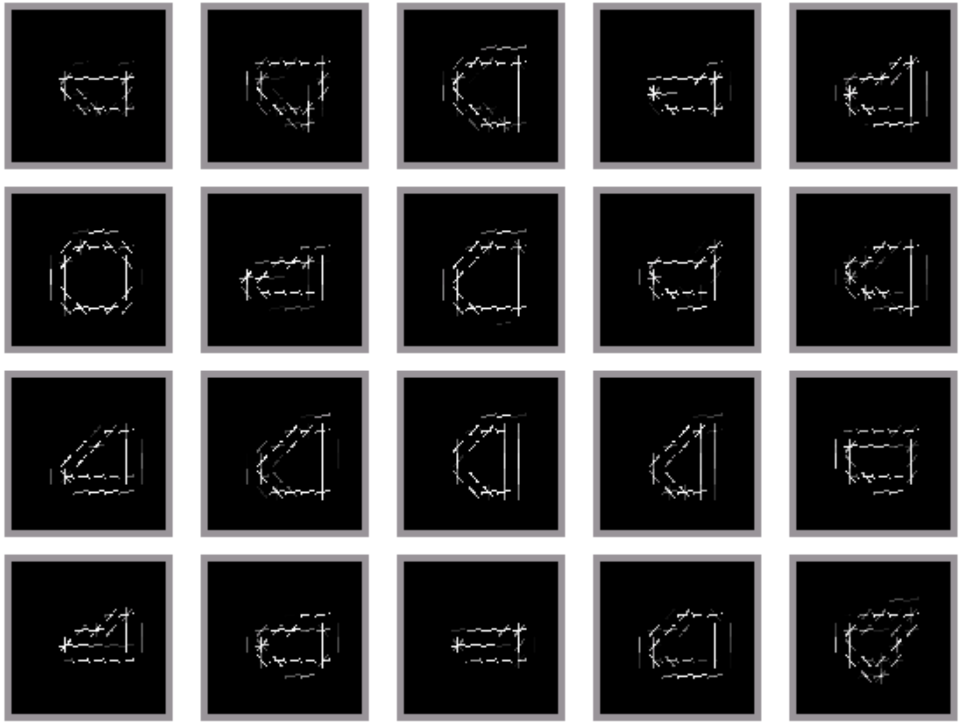


Nose (k=6)

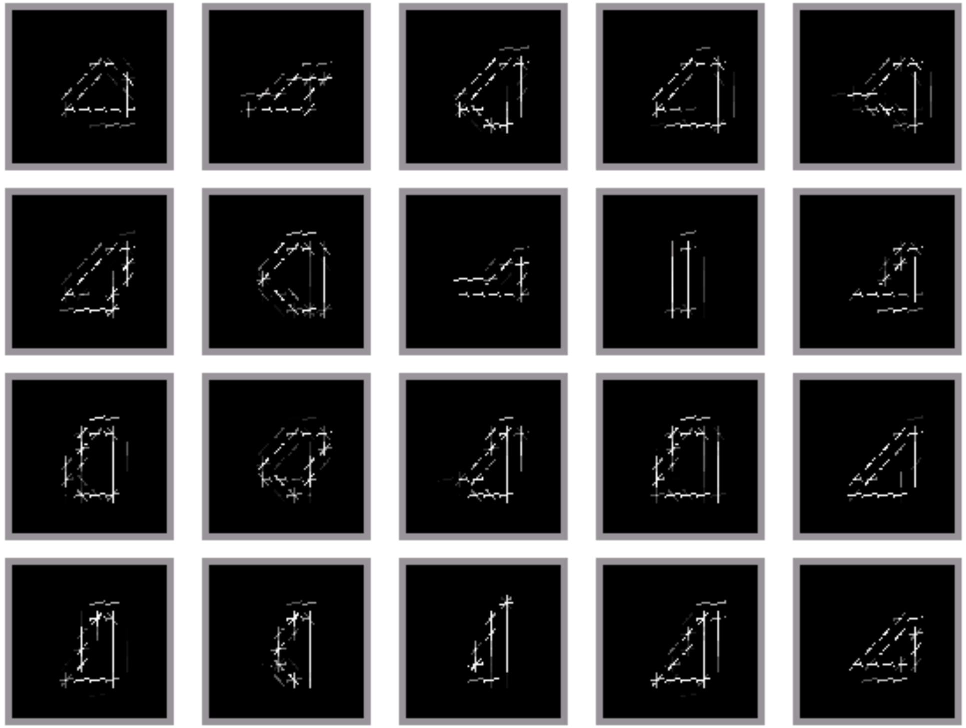
- **Method 2:** use segmentation masks for shape clustering
  - ✓ does not rely on aspect ratio
  - ✗ requires additional annotations (ok, we have them)

Leveraging shape annotations

- Binary oriented edge features from shape masks 
- EM (latent translation, scale & cluster) with mixtures of Bernoulli templates

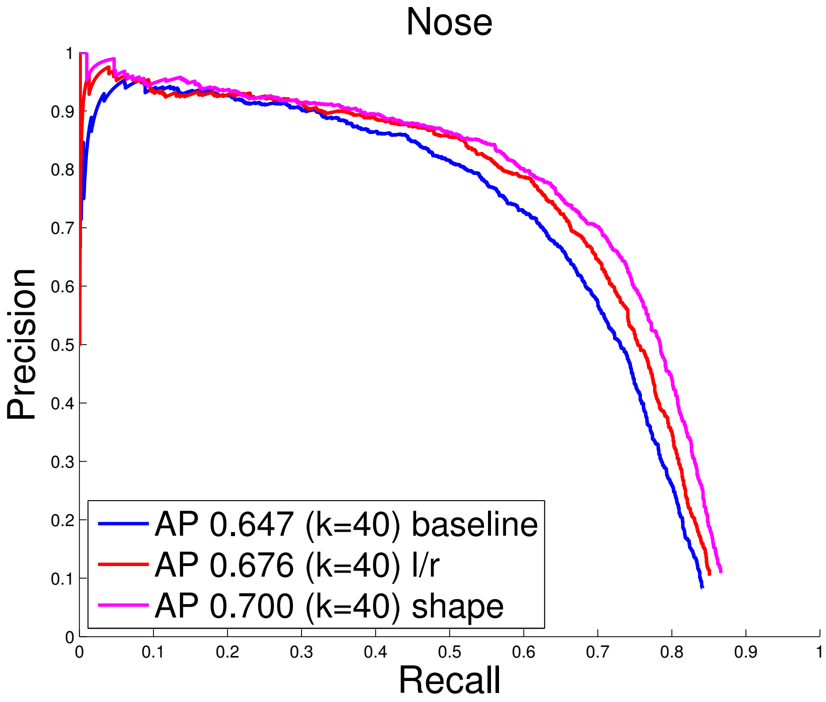


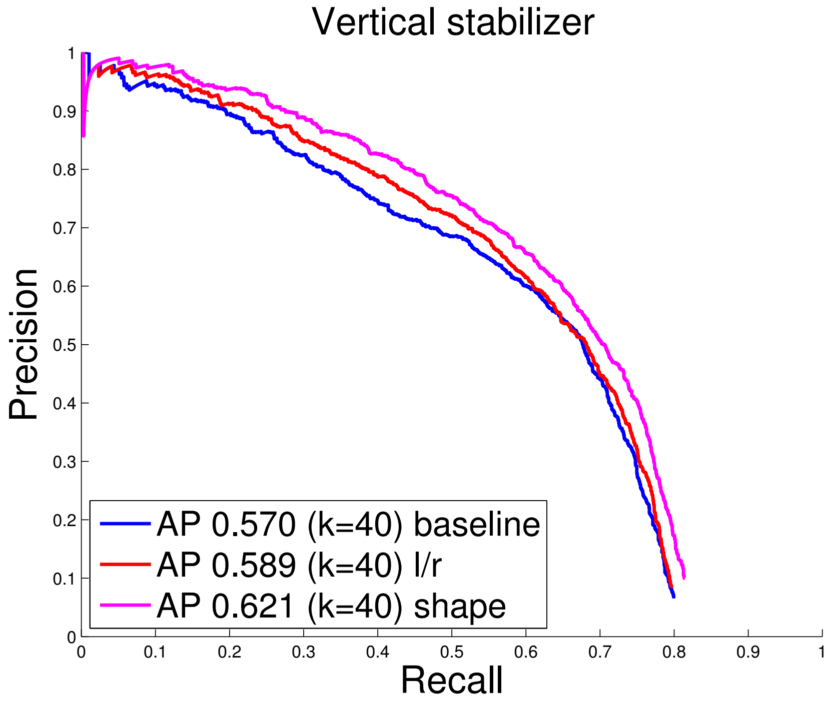
Nose

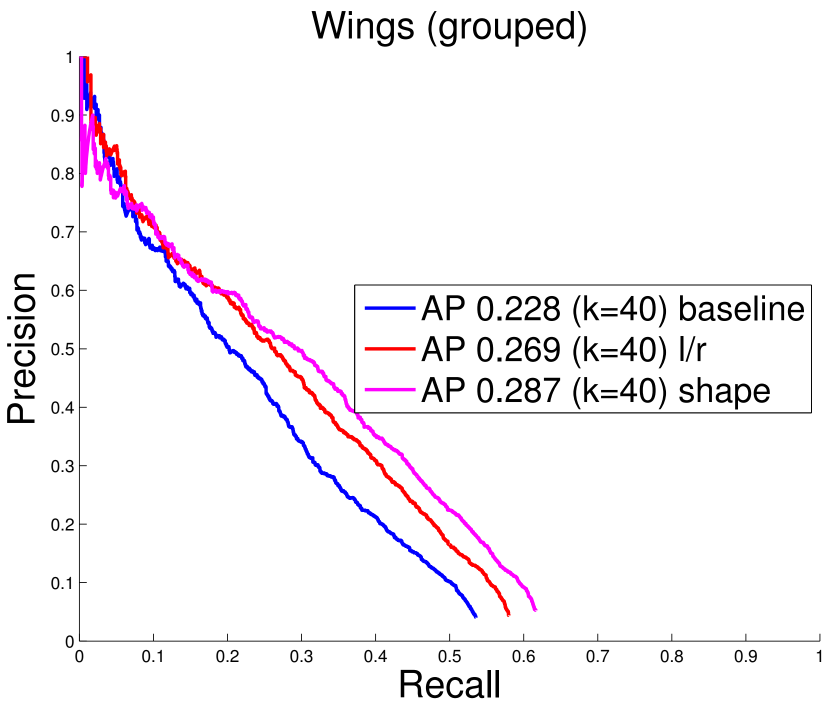


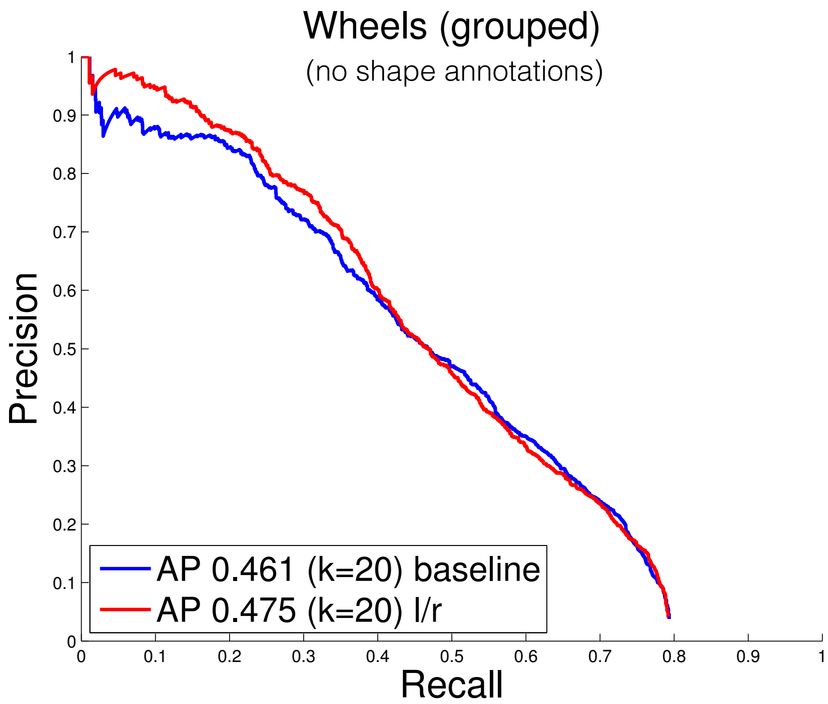
Vertical stabilizer

Left-right and shape clustering results



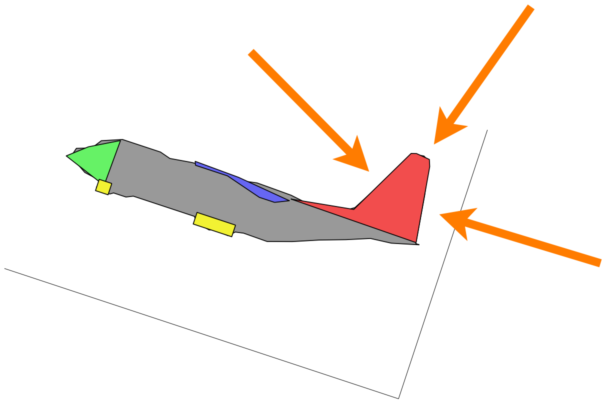






Localizing parts: summary

- Semantically aligned parts: good for applications and debugging
- Unsupervised left *vs* right helps tease out shape information
- Shape masks initialization works even better (good for square parts!)



Overview

Objects in Detail

Parts & attributes

- A new dataset
- An object lexicon
- Localising parts
- Layouts
- Recognising attributes

Stuff in Detail

Texture

- A texture lexicon
- A new dataset
- Transformation invariant semantic

Parsing

Bottom-up inference

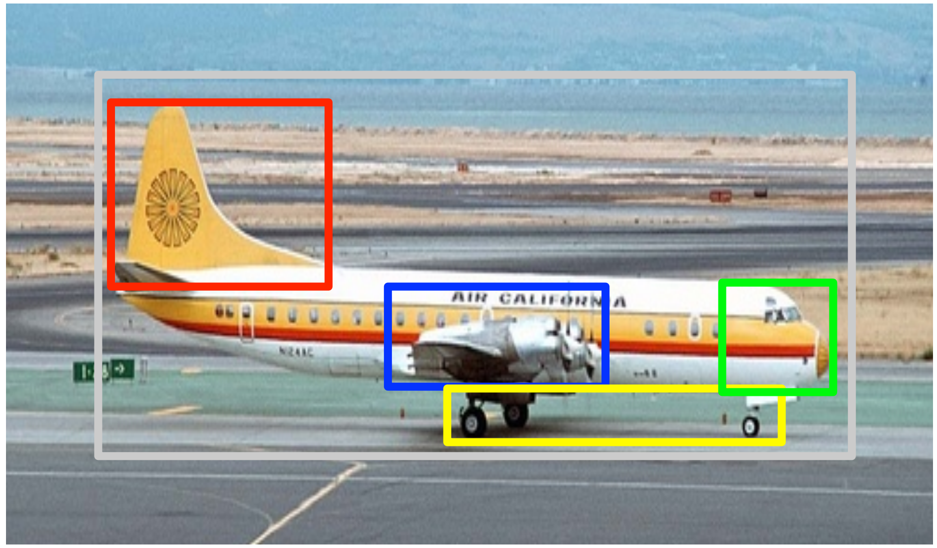
- Learning to merge
- Cascading
- Scoring regions by attributes

Estimating Layouts

Putting parts in context

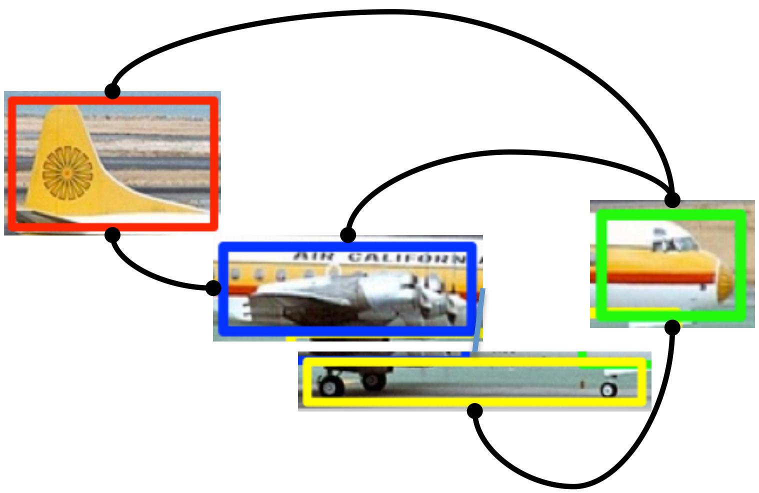
Subhransu Maji
TTI Chicago

Putting Parts in Context

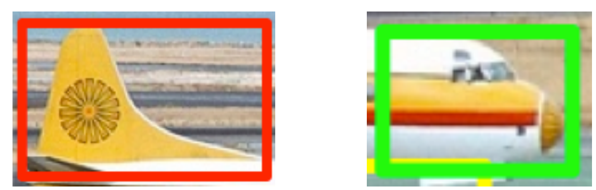


This talk

Spatial Layout



Appearance Layout



Similarity of Color & Texture
Shape compatibility
Contour continuity

Spatial Layout Variability

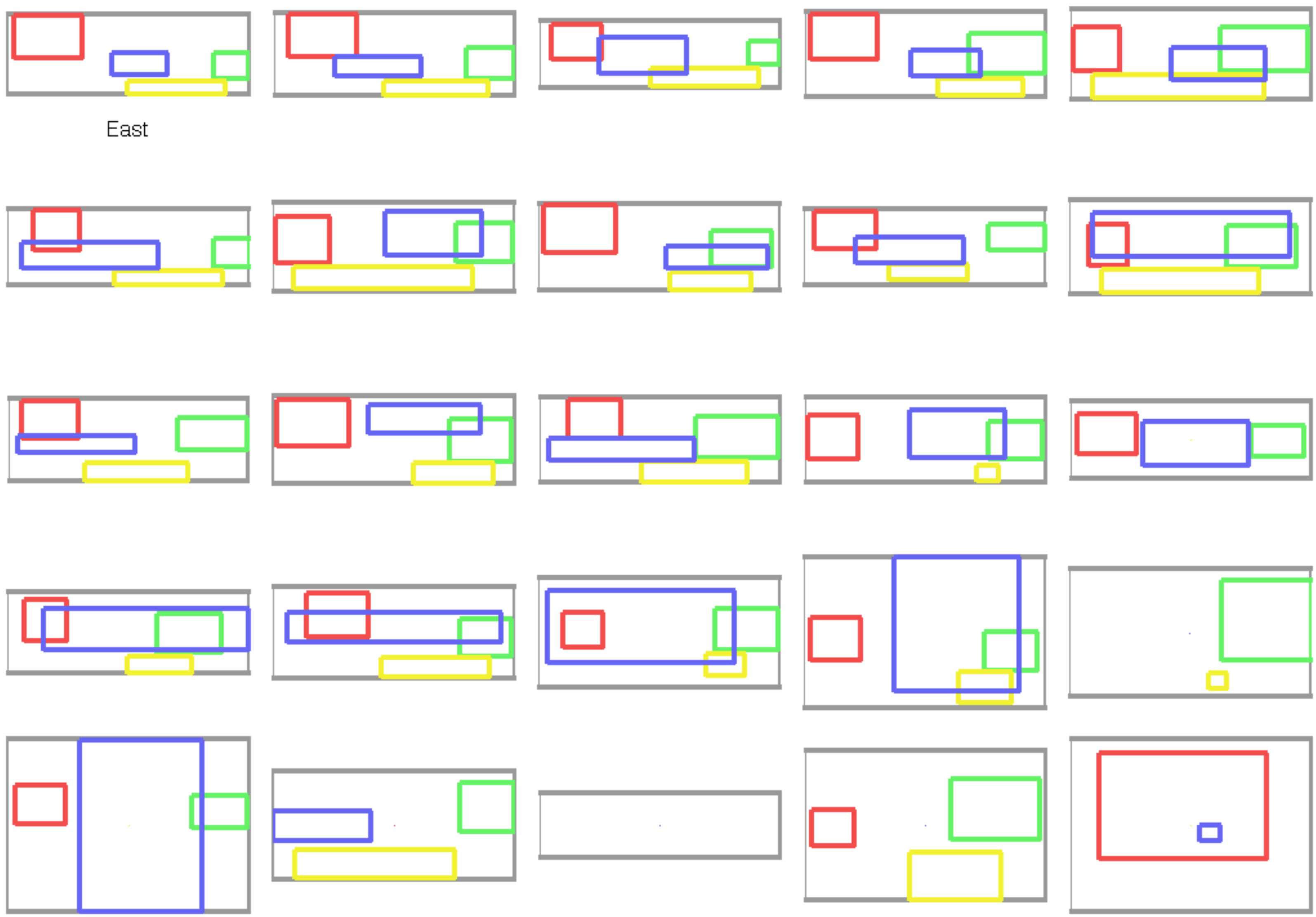
Viewpoint

Structural Variability

The need for mixture models

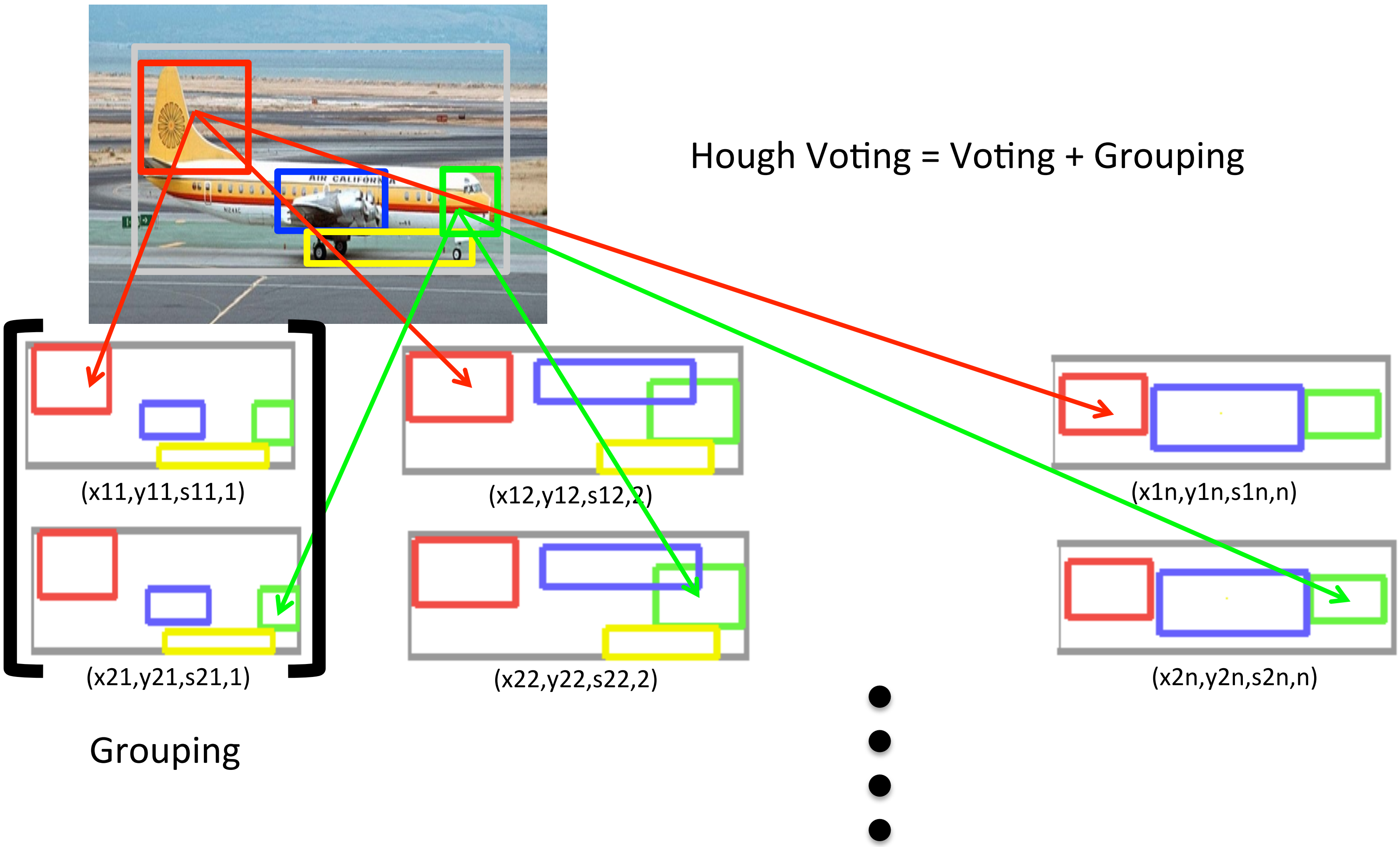
Aeroplane | Vert. Stab. | Wings | Nose | Wheel (Grp.)

Layouts of planes facing east

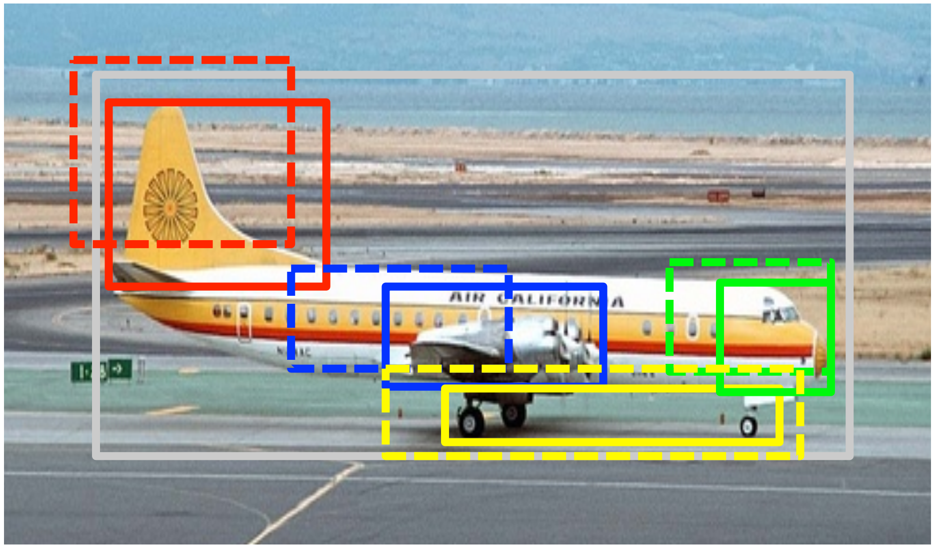


Efficiently Sampling Layouts

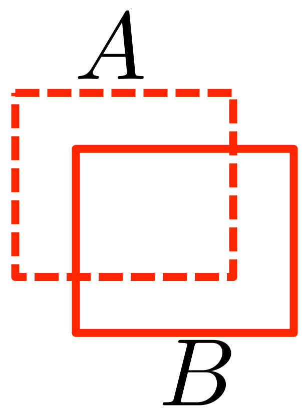
- Start from top *k* detections for each of the *n* parts
- Naïve solution : O(k^n) – all combinations
- Faster solution : Hough voting – O(n k #layouts)



Scoring and Evaluating a Layout



Measuring overlap



$$\frac{A \cap B}{A \cup B} > \tau$$

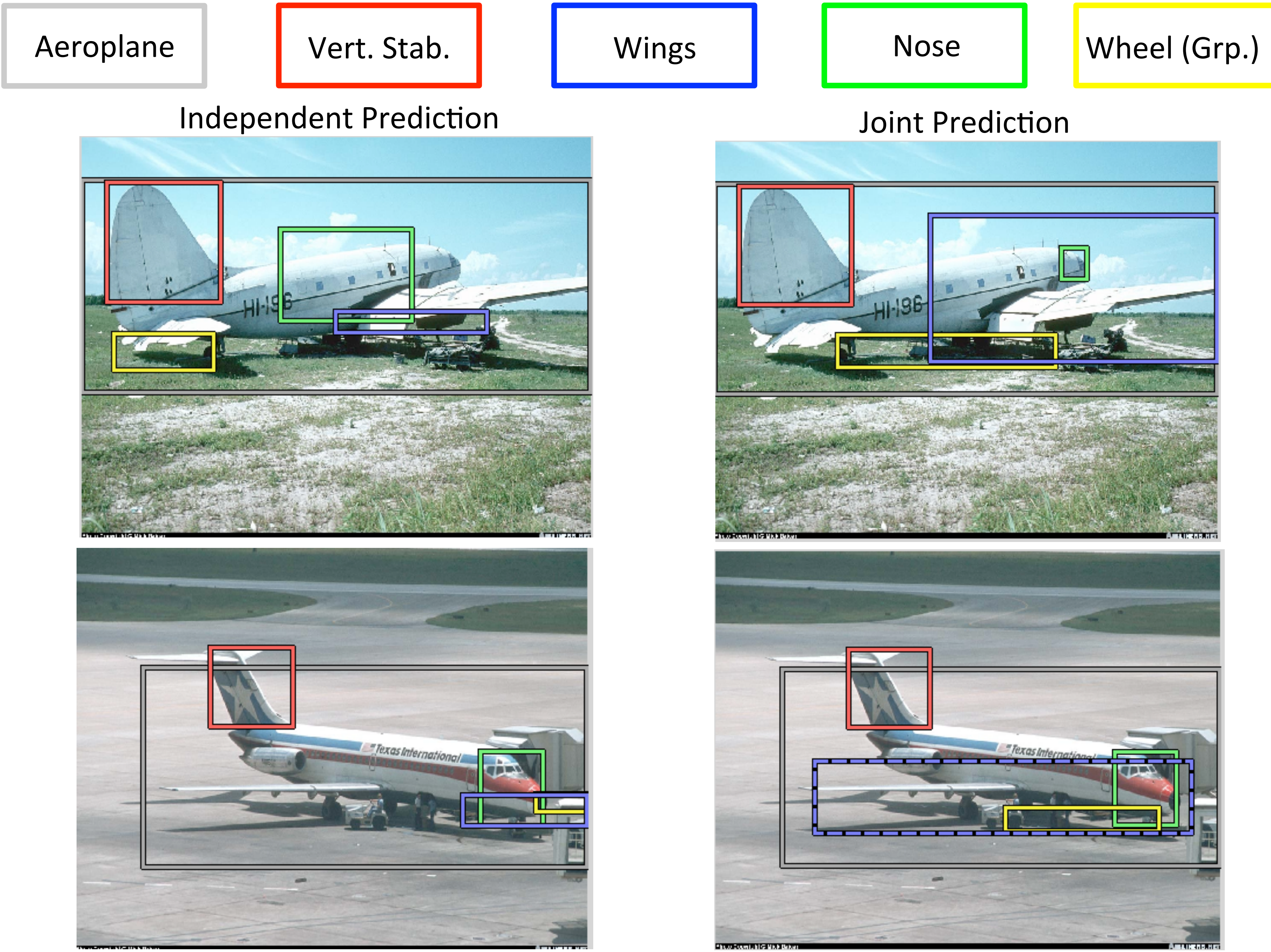
$$\texttt{Loss}(\mathbf{x}) = \sum_i \texttt{Loss}(x_i) + \max(0, \#true - \#predicted)$$

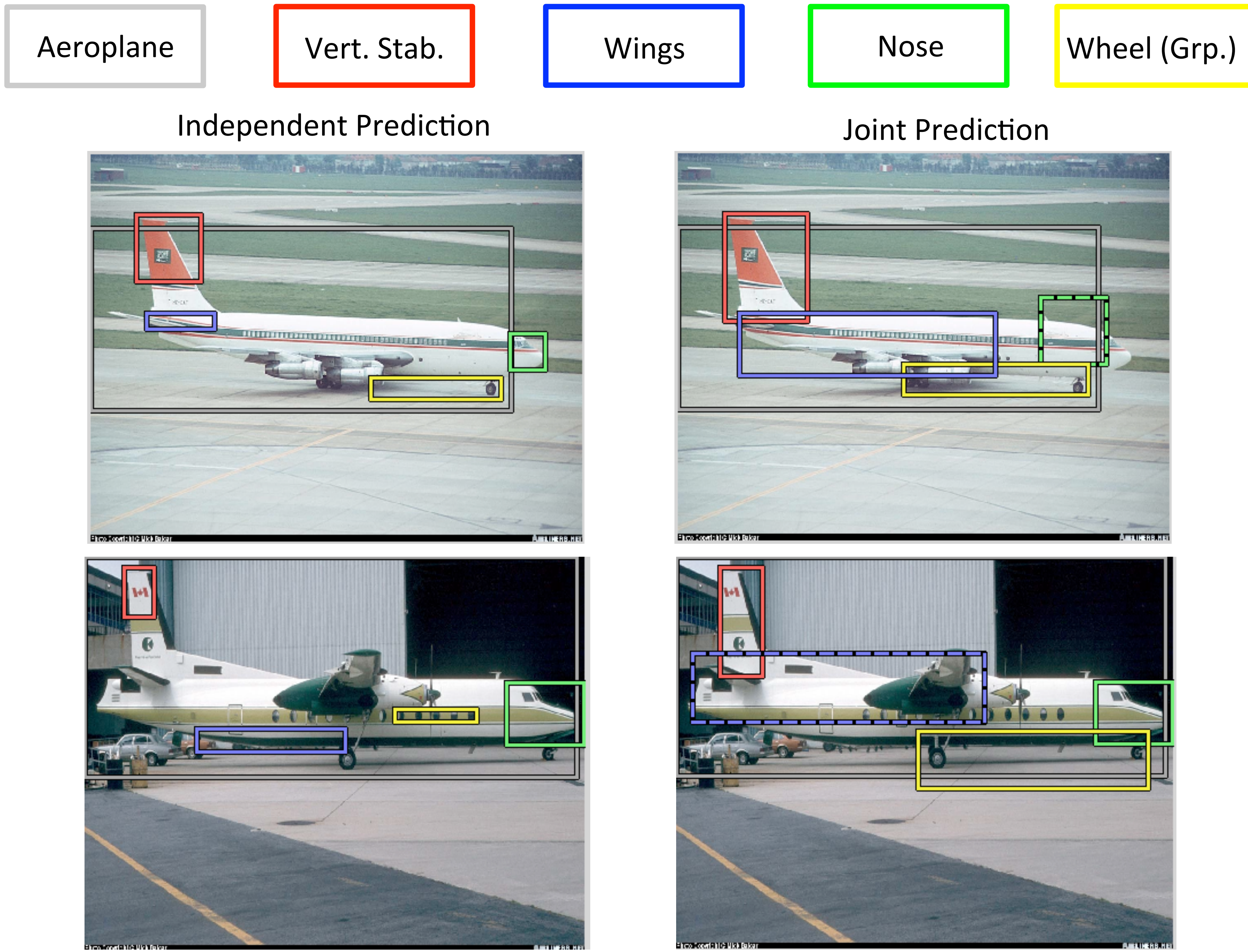
overlap — predicting too few

Score of a layout: $\mathbf{w}^T \Phi(\mathbf{x})$

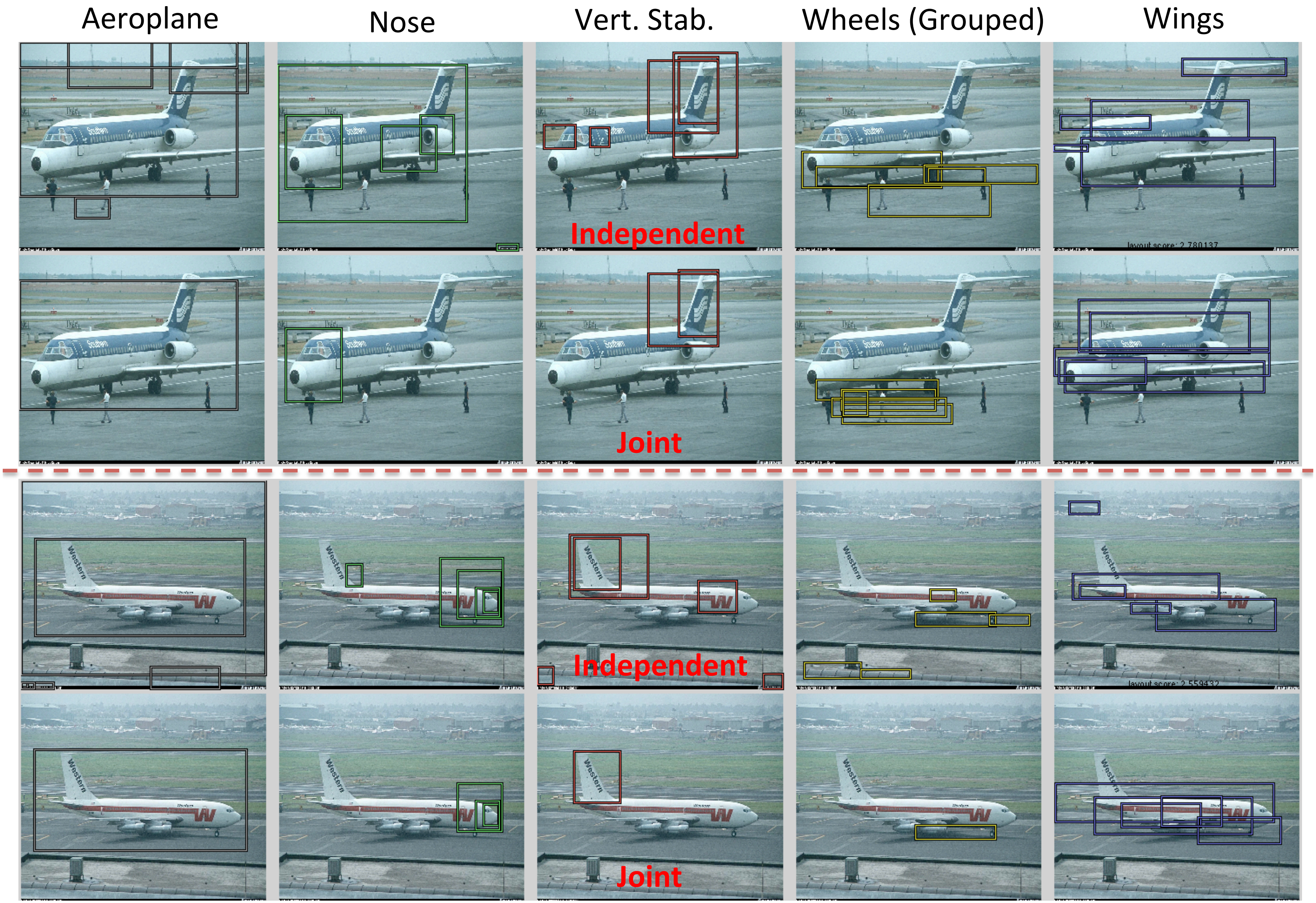
$\Phi(\mathbf{x})$ - features extracted from the part locations

Weights trained using MIL learning

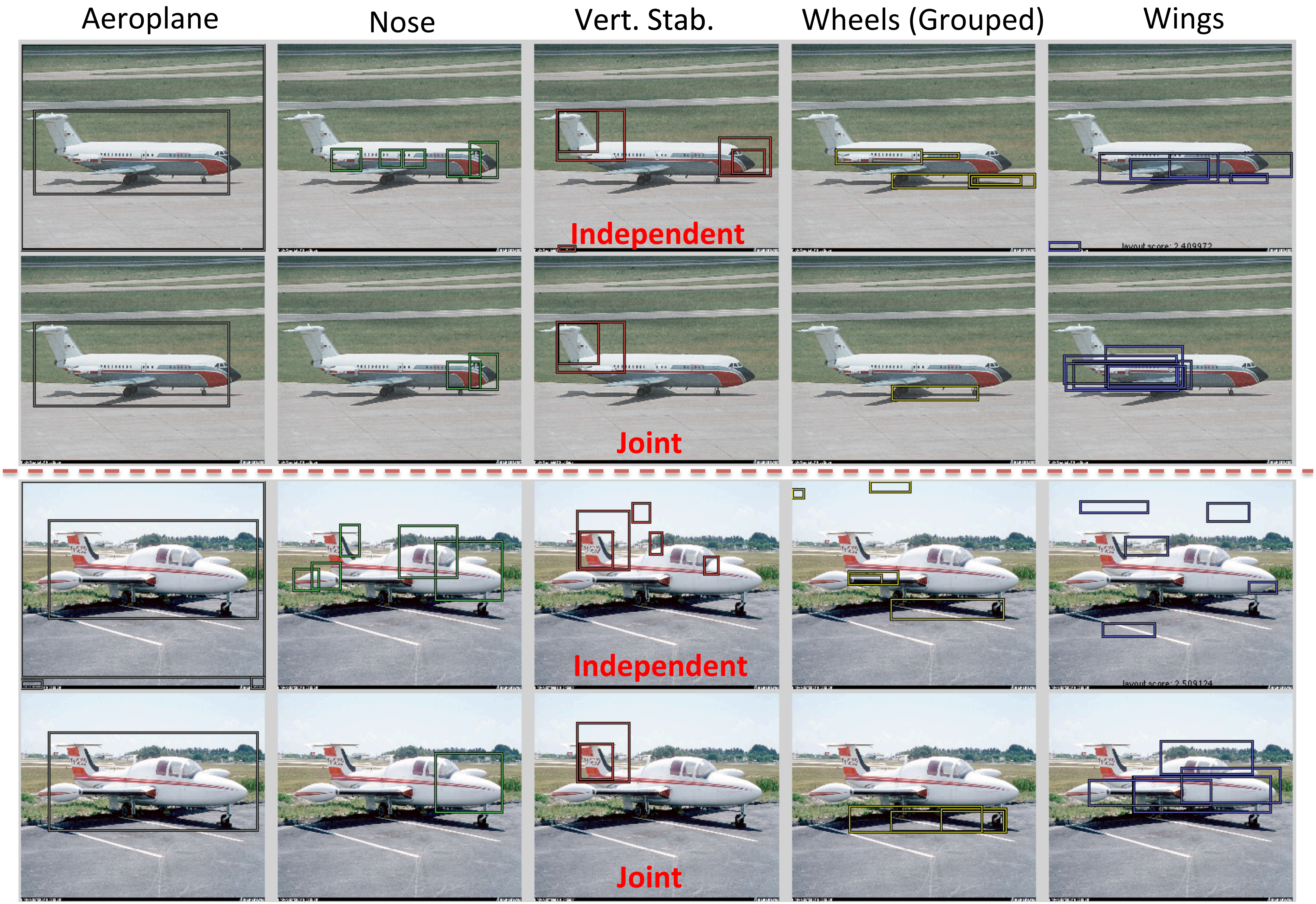

Aeroplane
Vert. Stab.
Wings
Nose
Wheel (Grp.)
Independent Prediction
Joint Prediction


Aeroplane
Vert. Stab.
Wings
Nose
Wheel (Grp.)
Independent Prediction
Joint Prediction

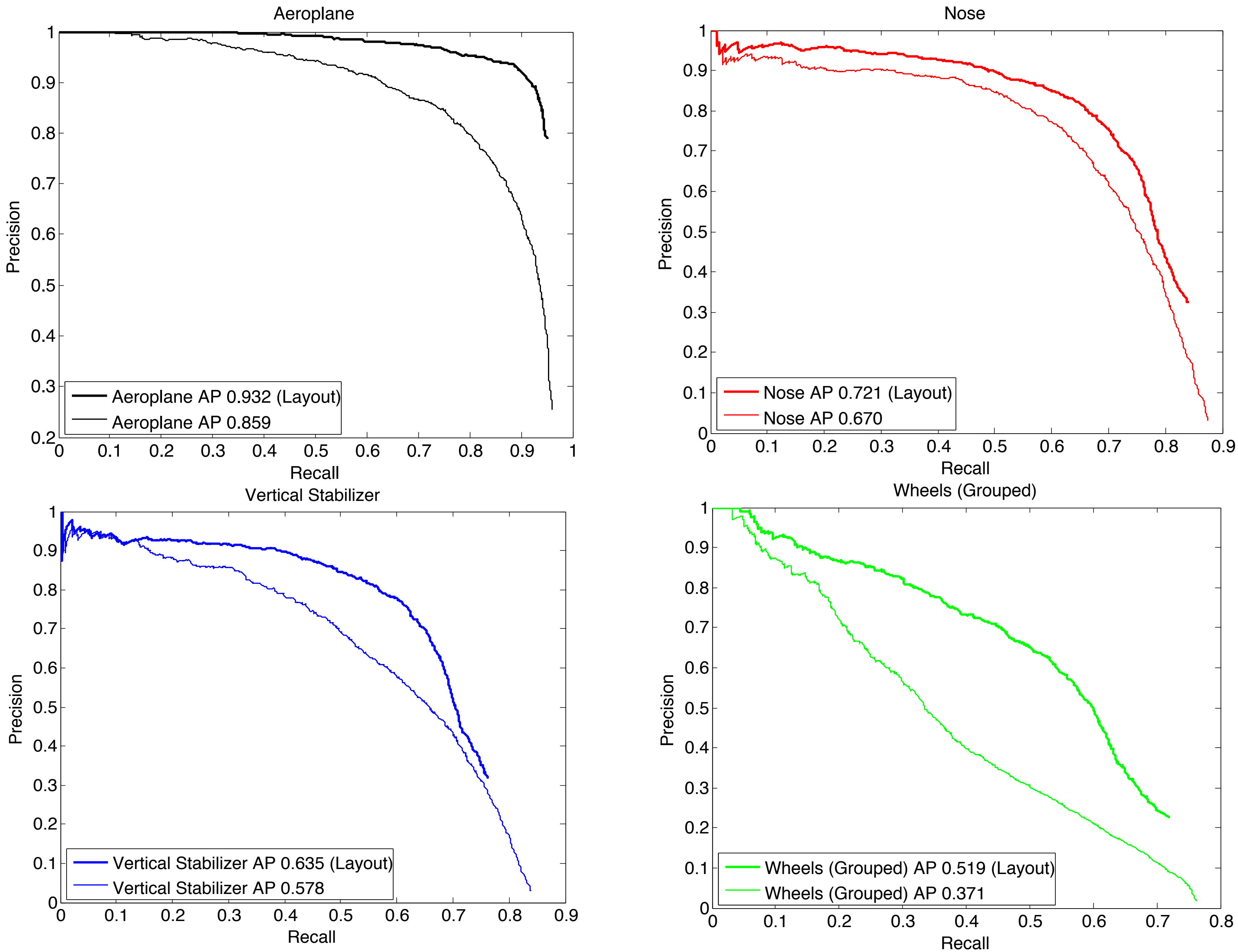
Part Detections



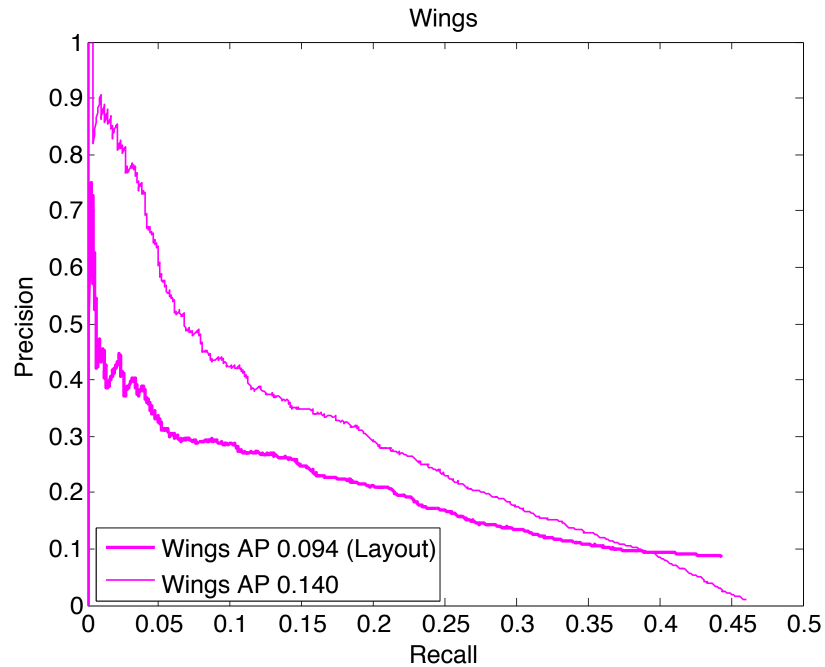
Part Detections



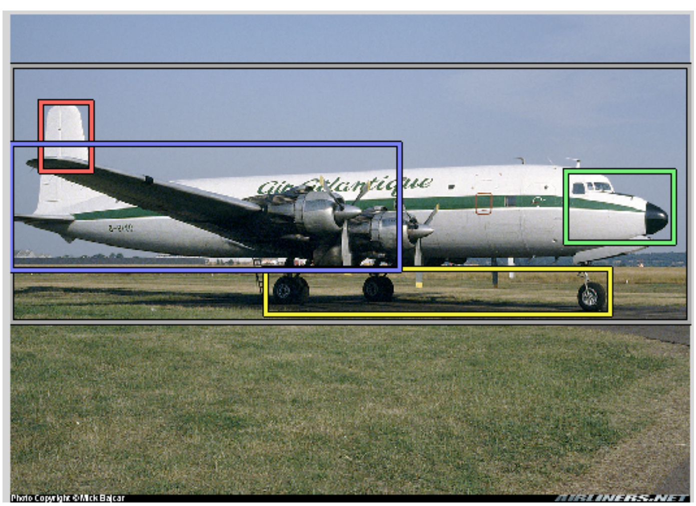
Part Detection Results

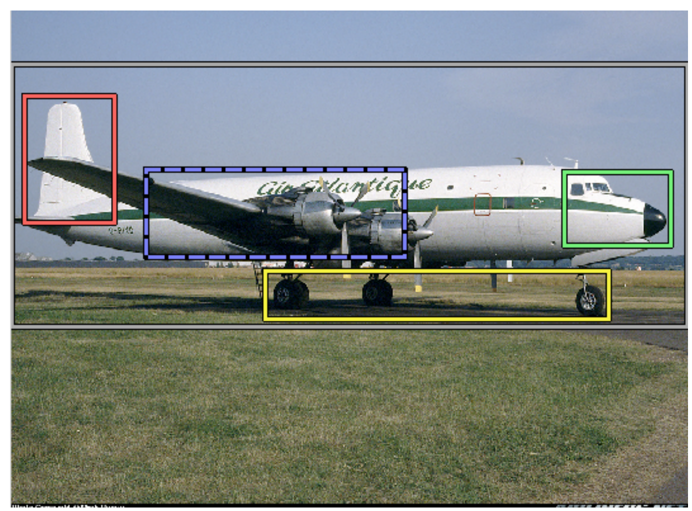


Part Detection Results



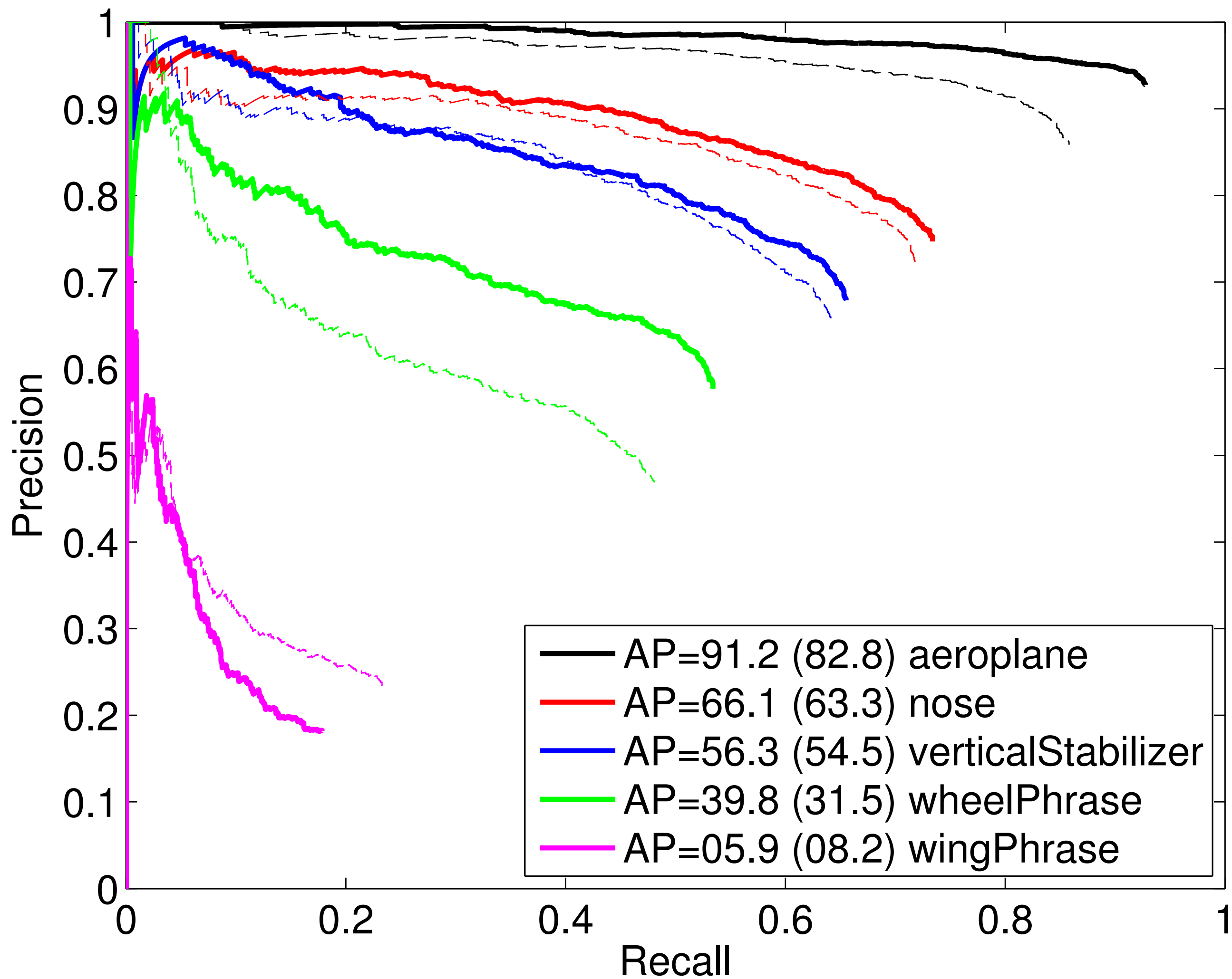
The model has learned to ignore the wing detections



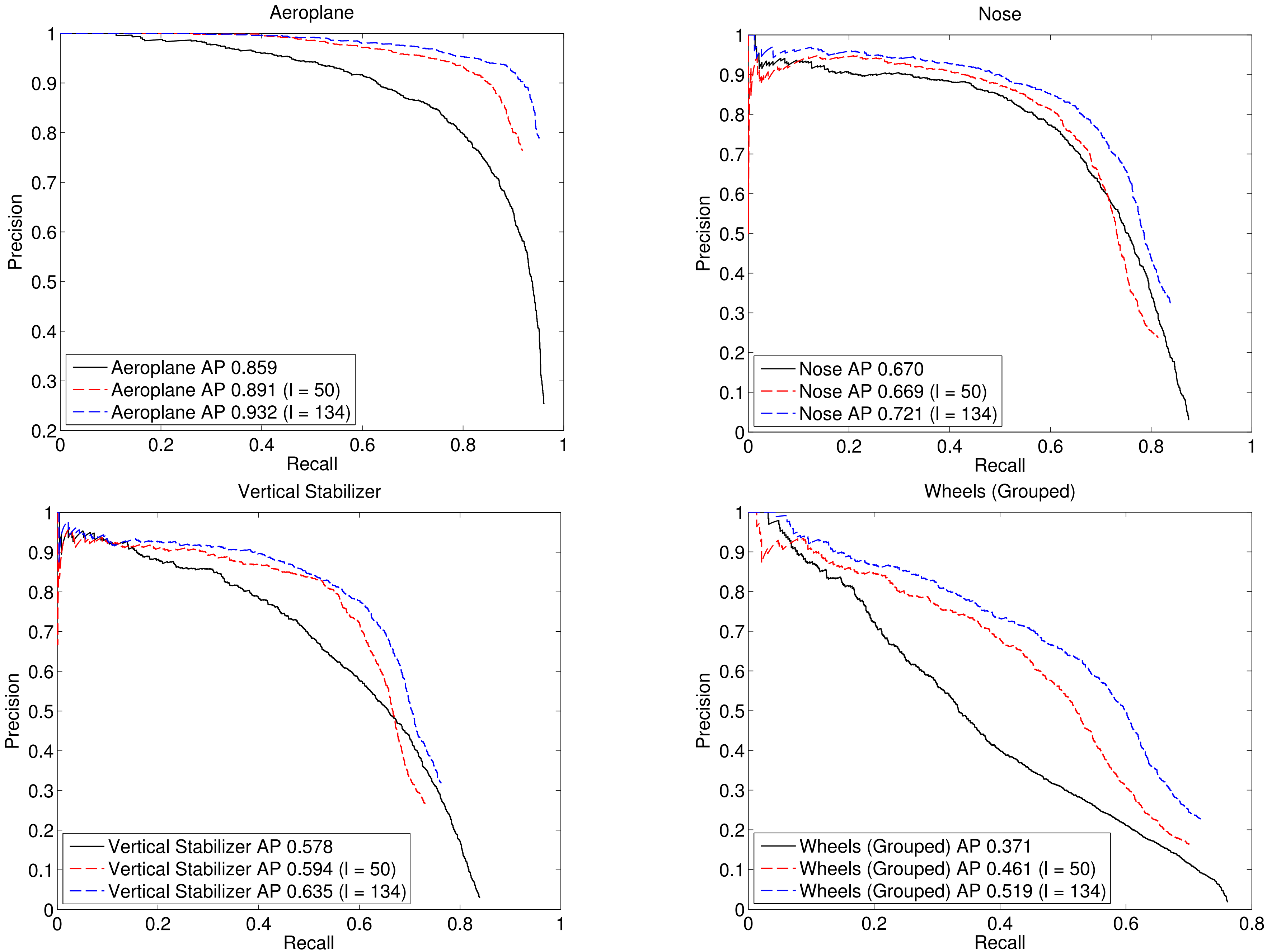


Layout Estimation Task

(allowed one layout per image)



How many layouts necessary?



Summary

- Planes have wide variety of layouts due to the view point and structural differences.
- This is a unique property of this dataset, which enables new directions in research about part detection (i.e. beyond a few mixture models)
- We explored a possible way of representing such spatial layouts and showed that it improves detection quite a bit
- Appearance layouts will be explored in the future.

Overview

Objects in Detail
Parts & attributes

- A new dataset
- An object lexicon
- Localising parts
- Layouts
- Recognising attributes

Stuff in Detail
Texture

- A texture lexicon
- A new dataset
- Transformation invariant semantic

Parsing
Bottom-up inference

- Learning to merge
- Cascading
- Scoring regions by attributes

Overview

Objects in Detail

Parts & attributes

- A new dataset
- An object lexicon
- Localising parts
- Layouts
- Recognising attributes

Stuff in Detail

Texture

- A texture lexicon
- A new dataset
- Transformation invariant semantic

Parsing

Bottom-up inference

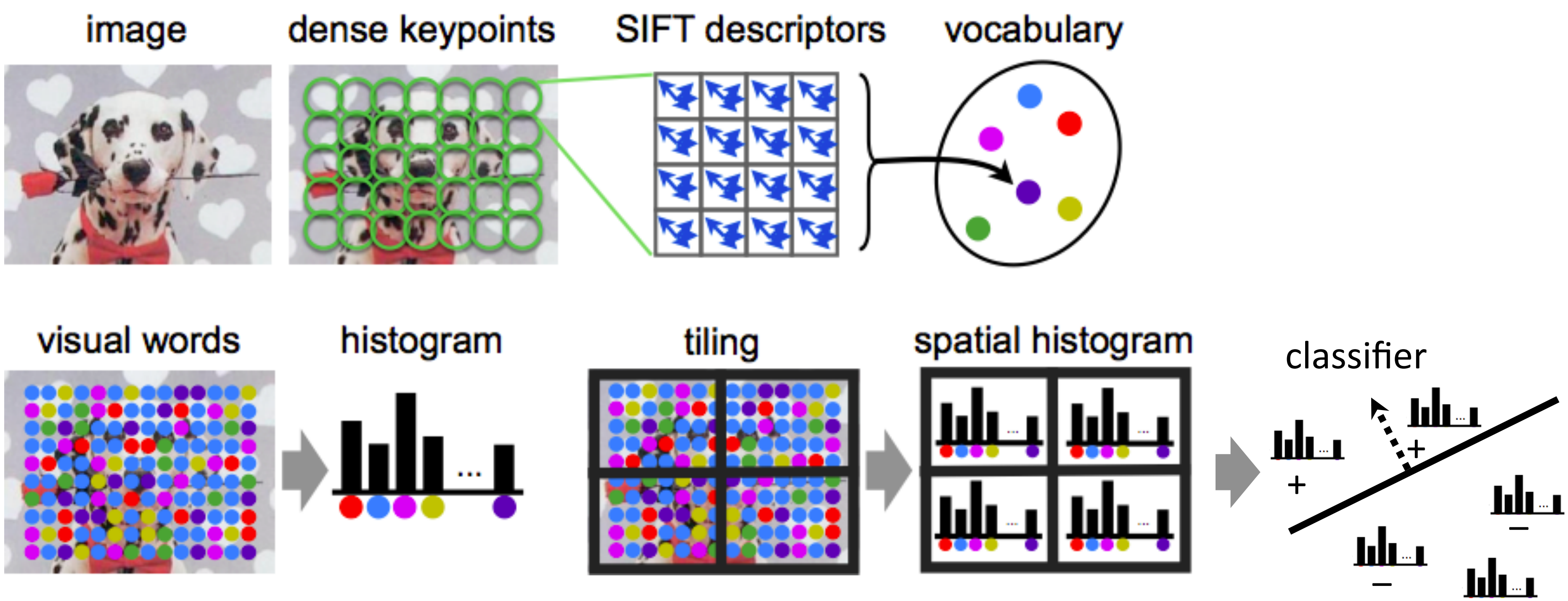
- Learning to merge
- Cascading
- Scoring regions by attributes

Attributes

isAirliner: ‘yes’
isMilitaryPlane: ‘no’
isSeaPlane: ‘no’
facingDirection: ‘W’
planeLocation: ‘on ground’

wingType: ‘single-wing plane
tailHasEngine: ‘no-engine’
wheel-coverType: ‘retractable’

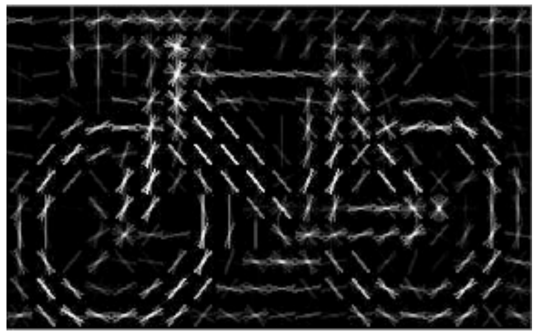
Bag of Visual Words



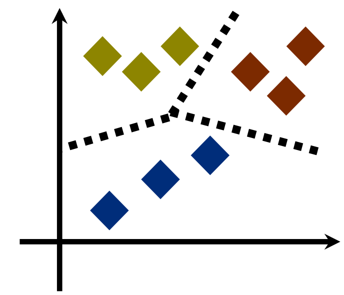
Current Methodology

input

representation



interpretation



edges, blobs, textures

bicycle?

has rider?

has wheel?

semantic gap

Context is Important

left signaling light

Predict the Attributes

Where is the plane located ? **What kind of aeroplane is it ?**

What type of wing does it have ? **What direction is it facing ?**


SCANDINAVIAN
SAS
Photo Copyright © Mick Bajcar
AIRLINERS.NET

Objects in Detail

- Image
- Aeroplane
- Parts
  - Background
  - Vertical Stabilizer
  - Nose
  - Wing
  - Wheel
  - Fuselage
- Undercarriage

isMilitaryPlane:‘yes’

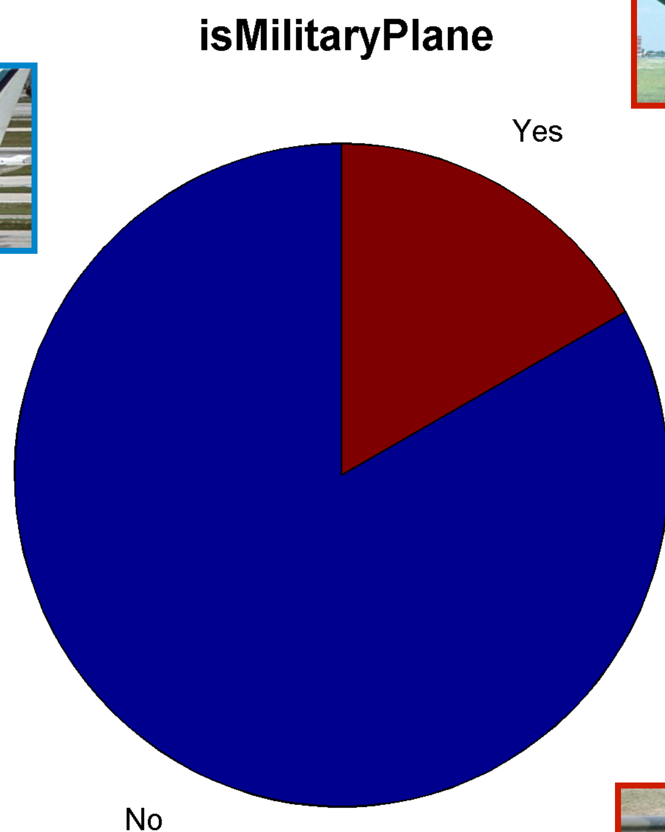


Image : isMilitaryPlane

isMilitaryPlane: ‘yes’

AP : 73.92

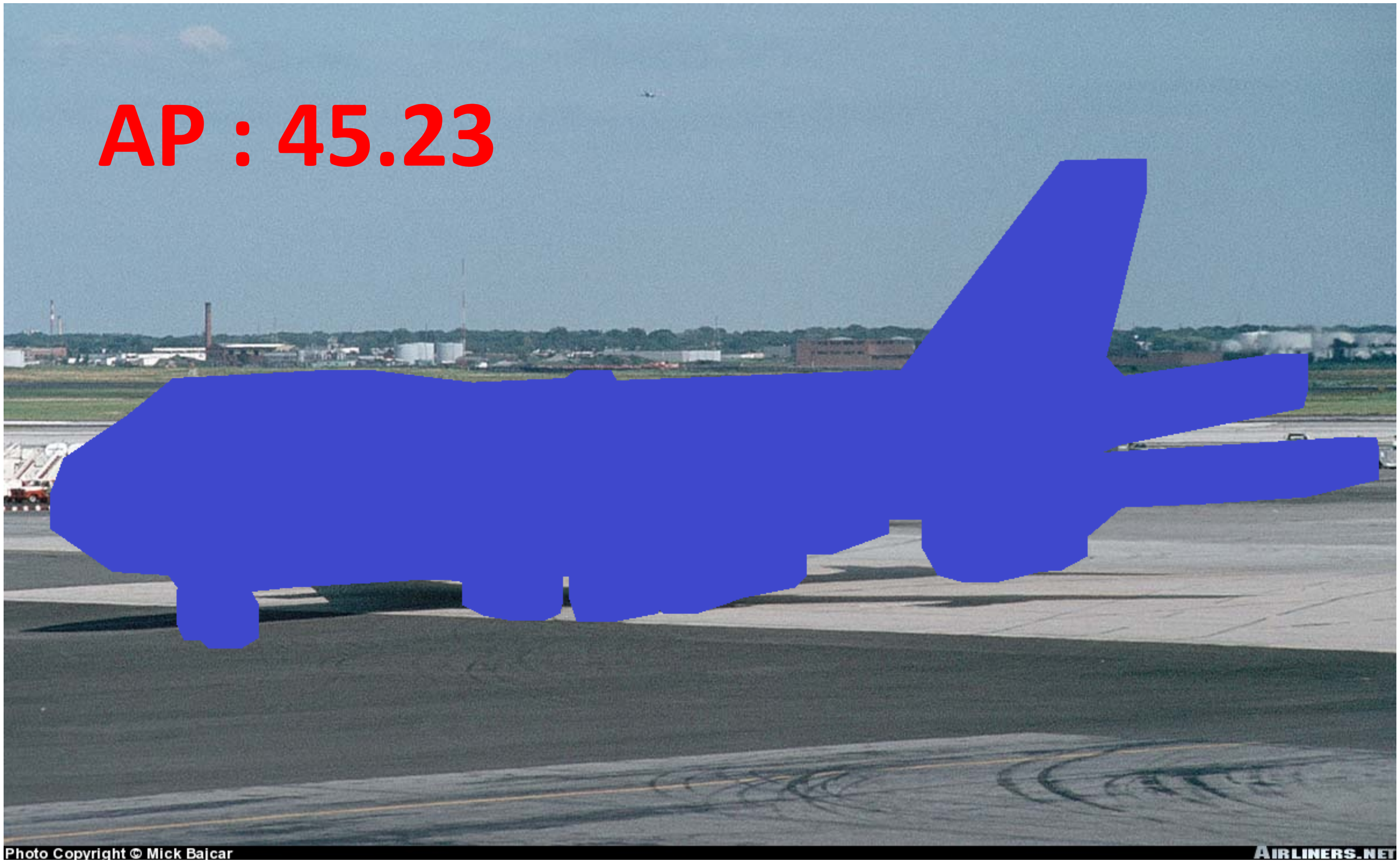
- Image
- Aeroplane
- Parts
  - Background
  - Vertical Stabilizer
  - Nose
  - Wing
  - Wheel
  - Fuselage
- Undercarriage

Aeroplane

isMilitaryPlane: ‘yes’

- Image **AP : 73.92**
- Aeroplane **AP : 83.88**
- Parts
  - Background
  - Vertical Stabilizer
  - Nose
  - Wing
  - Wheel
  - Fuselage
- Undercarriage

Background



isMilitaryPlane: 'yes'

- Image **AP : 73.92**
- Aeroplane **AP : 83.88**
- Parts
  - Background **AP : 45.23**
  - Vertical Stabilizer
  - Nose
  - Wing
  - Wheel
  - Fuselage
- Undercarriage

Vertical Stabilizer

isMilitaryPlane: 'yes'

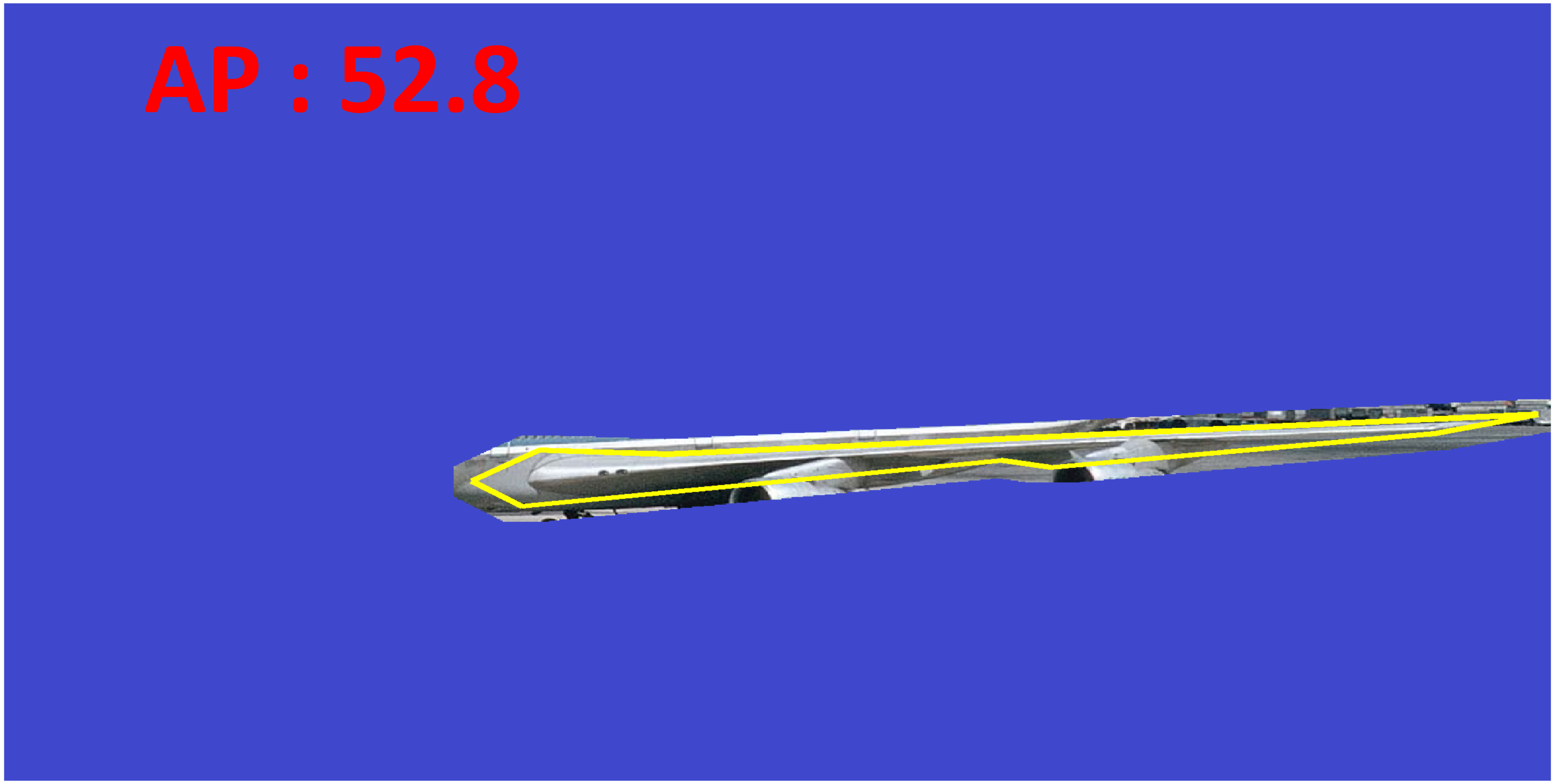
- Image **AP : 73.92**
- Aeroplane **AP : 83.88**
- Parts
  - Background **AP : 45.23**
  - Vertical Stabilizer **AP : 71.30**
  - Nose
  - Wing
  - Wheel
  - Fuselage
- Undercarriage

Nose

isMilitaryPlane: ‘yes’

- Image **AP : 73.92**
- Aeroplane **AP : 83.88**
- Parts
  - Background **AP : 45.23**
  - Vertical Stabilizer **AP : 71.30**
  - Nose **AP : 75.21**
  - Wing
  - Wheel
  - Fuselage
- Undercarriage

Wing



isMilitaryPlane: ‘yes’

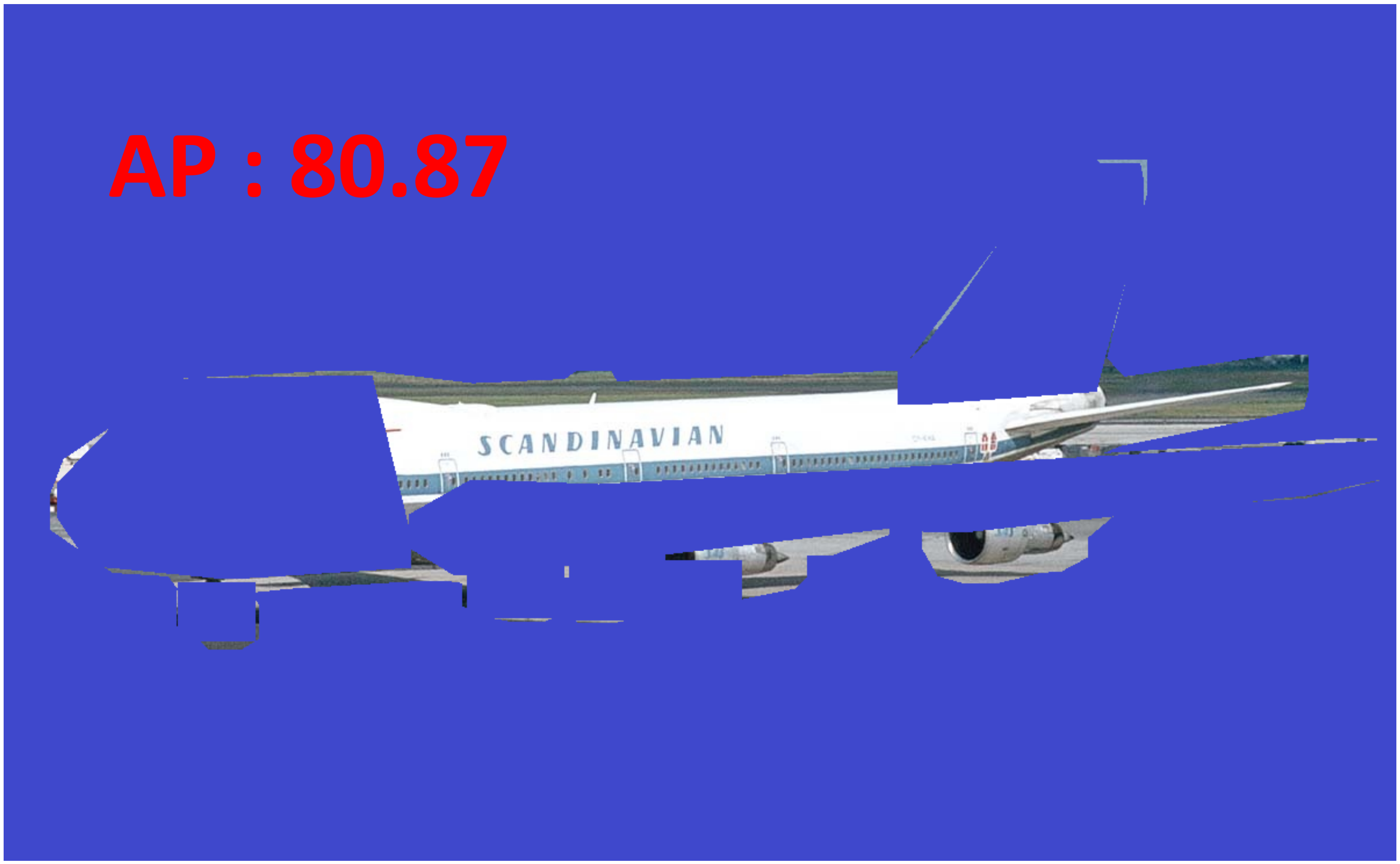
- Image **AP : 73.92**
- Aeroplane **AP : 83.88**
- Parts
  - Background **AP : 45.23**
  - Vertical Stabilizer **AP : 71.30**
  - Nose **AP : 75.21**
  - Wing **AP : 52.80**
  - Wheel
  - Fuselage
- Undercarriage

Wheel

isMilitaryPlane: ‘yes’

- Image **AP : 73.92**
- Aeroplane **AP : 83.88**
- Parts
  - Background **AP : 45.23**
  - Vertical Stabilizer **AP : 71.30**
  - Nose **AP : 75.21**
  - Wing **AP : 52.80**
  - Wheel **AP : 45.99**
  - Fuselage
- Undercarriage

``Fuselage''



isMilitaryPlane: ‘yes’

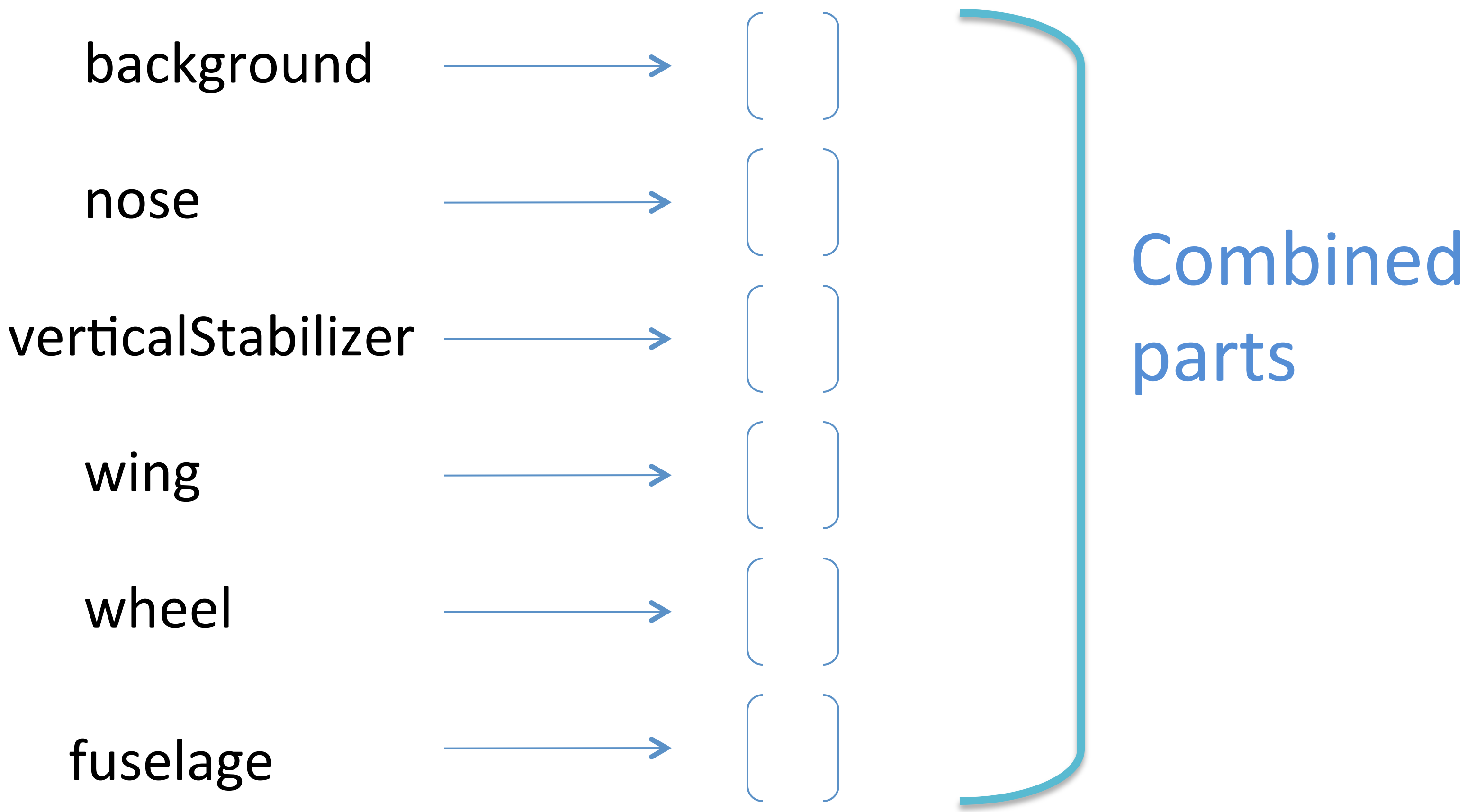
- Image **AP : 73.92**
- Aeroplane **AP : 83.88**
- Parts
  - Background **AP : 45.23**
  - Vertical Stabilizer **AP : 71.30**
  - Nose **AP : 75.21**
  - Wing **AP : 52.80**
  - Wheel **AP : 45.99**
  - Fuselage **AP : 80.87**
- Undercarriage

Undercarriage

isMilitaryPlane: ‘yes’

- Image **AP : 73.92**
- Aeroplane **AP : 83.88**
- Parts
  - Background **AP : 45.23**
  - Vertical Stabilizer **AP : 71.30**
  - Nose **AP : 75.21**
  - Wing **AP : 52.80**
  - Wheel **AP : 45.99**
  - Fuselage **AP : 80.87**
- Undercarriage **AP : 45.63**

Combined parts



isMilitaryPlane: 'yes'

- Image — **AP : 73.92**
- Aeroplane — **AP : 83.88**
- Parts
  - Background — **AP : 45.23**
  - Vertical Stabilizer — **AP : 71.30**
  - Nose — **AP : 75.21**
  - Wing — **AP : 52.80**
  - Wheel — **AP : 45.99**
  - Fuselage — **AP : 80.87**
- Undercarriage — **AP : 45.63**
- Combined parts — **AP : 87.92**

Possible Variations (seg. v/s box.)

Parts & Attributes - fuselage

- isAirliner (1.5;nose)
- isCargoPlane (18.19)
- isMilitaryPlane (5.66)
- isPropellorPlane (0.68;nose)
- isSeaPlane (42.51)
- isGlider (9.43)
- planeSize (7.52)
- noseHasEngineOrAntenna (0.53;nose)
- wingHasEngine (1.34;nose)
- wheel-coverType (6.8)

Parts & Attributes - wheel

- planeLocation (1.72;background)
- undercarriageArrangement (8.98)
- wheel-location (2.69)

Parts & Attributes - nose

- facingDirection (3.96)
- wheel-groupType (1.18;fuselage)

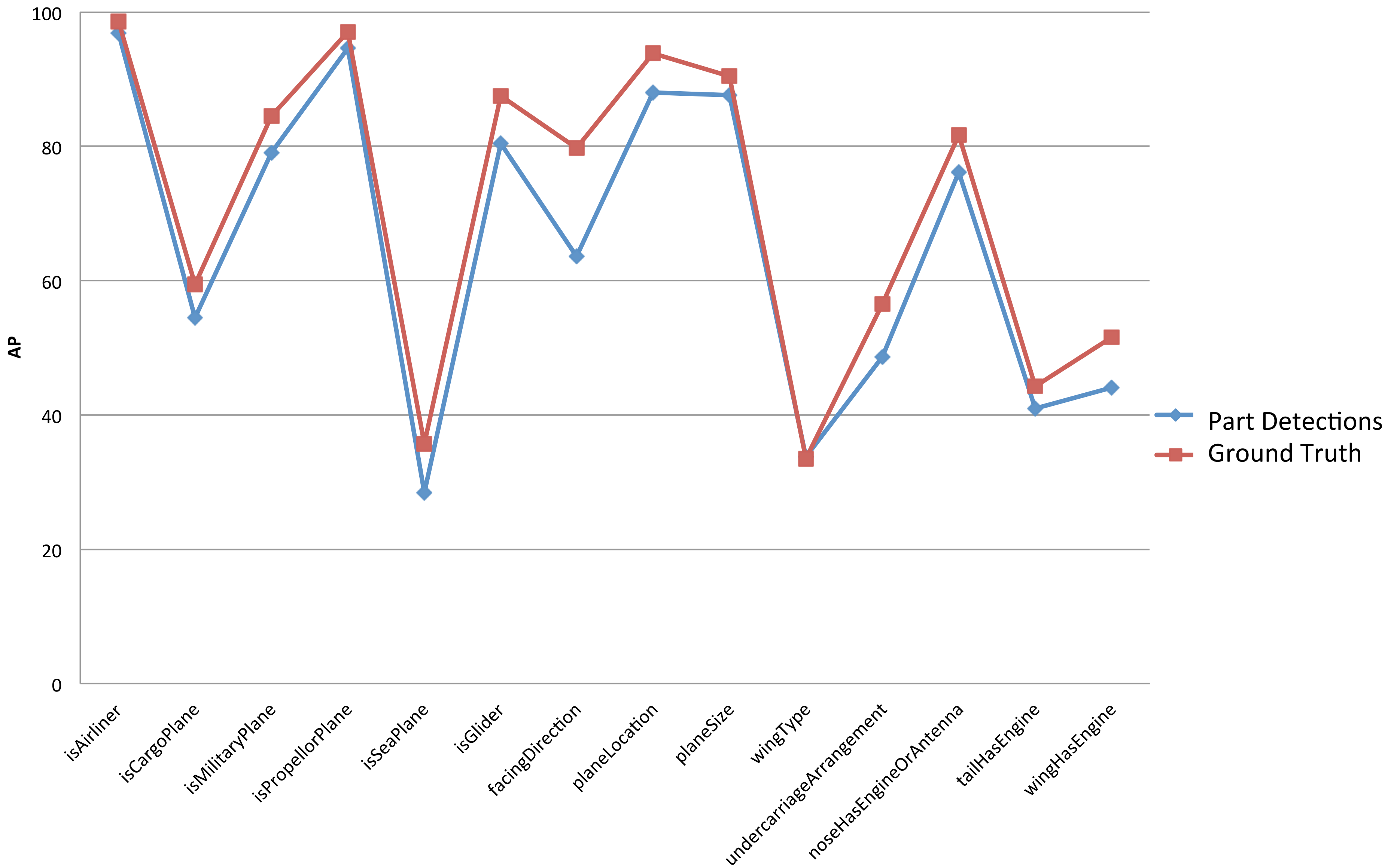
Parts & Attributes - wing

- wingType (1.94;fuselage)

Parts & Attributes - verticalStabilizer

- tailHasEngine (3.28)

Attribute Recognition : Using Part detections



Conclusions

- Some regions of an image are more informative than others for a given task
- Utilizing part segmentations to add structure to Bag of Words improves performance significantly
- Fuselage and Wheel are the two most important parts accounting for 13/17 attributes
- Understanding which parts are more important can help focus effort in part detection stage

Overview

Objects in Detail

Parts & attributes

- A new dataset
- An object lexicon
- Localising parts
- Layouts
- Recognising attributes

Stuff in Detail

Texture

- A texture lexicon
- A new dataset
- Transformation invariant semantic

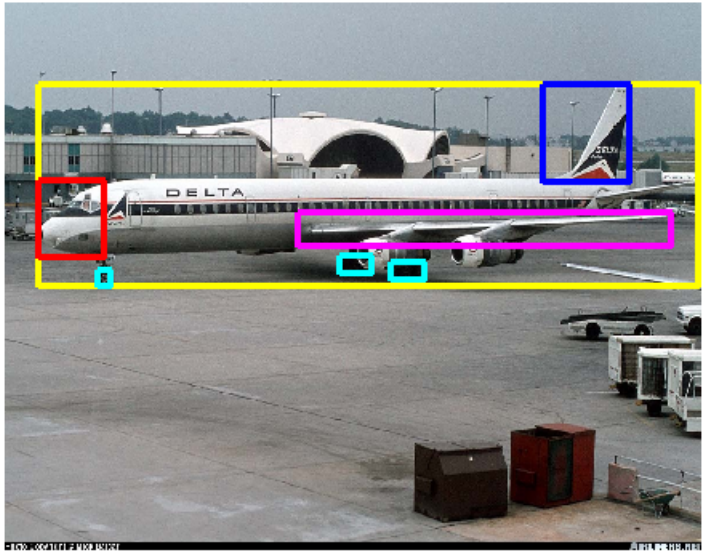
Parsing

Bottom-up inference

- Learning to merge
- Cascading
- Scoring regions by attributes

Attribute prediction using part-based models

- Task:



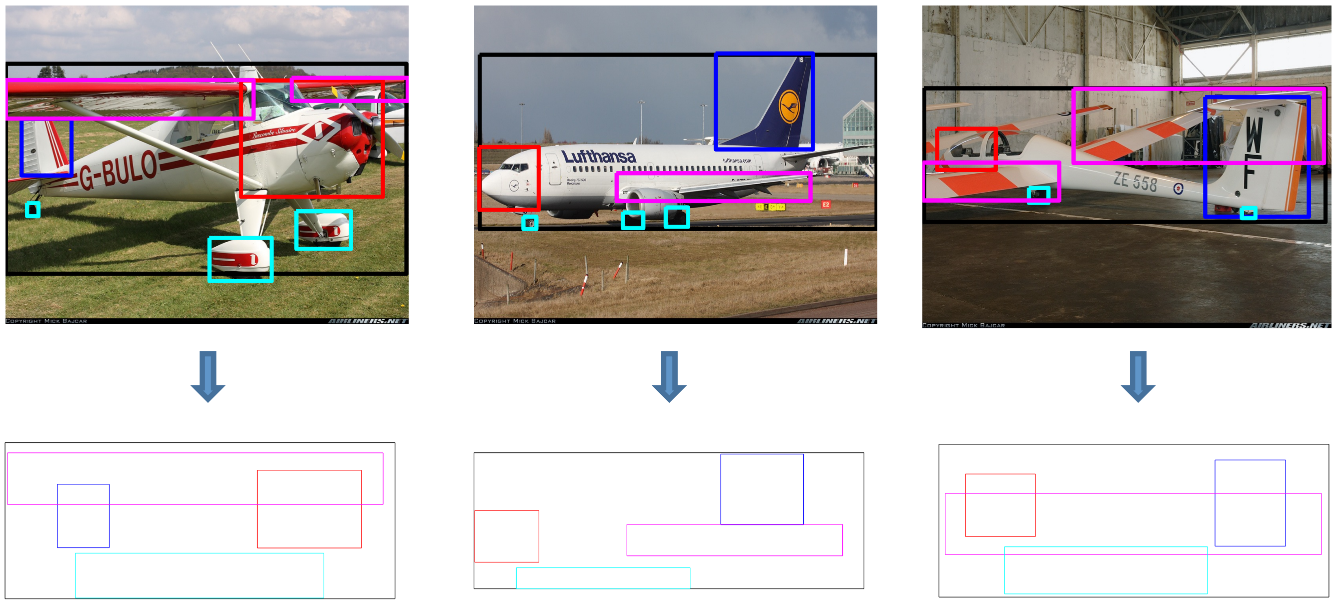
- Is airliner? (yes/no)
- Is military plane? (yes/no)
- Is facing East? (yes/no)
- Does nose have engine? (yes/no)
- Is Lufthansa plane? (yes/no)

Given an object detection, predict the attributes of the object.

Here we focus on geometry based features which encode spatial layout of object's parts

Layout features

- We cluster the geometric layouts of parts
- Given 5 airplane parts we concatenate their 5 bounding boxes into a 20-dimensional feature vector and perform kmeans clustering
- The closest cluster centers for a few ground truth detections:



- Each detection is assigned to the closest one of the k clusters

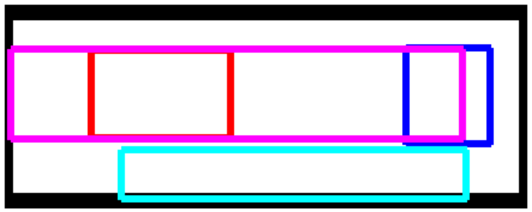
➡ k-dimensional binary feature vector to attribute classifiers

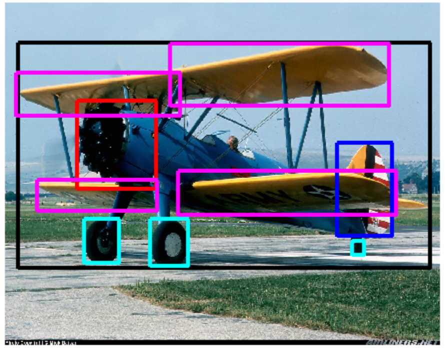
Layout features when the number of parts is varying

- Some detections may have all the parts but some may have less parts
- We cluster all possible detection configurations separately (16 in total)
- We get different layout vocabularies for different configurations
- We train attribute classifiers separately for each configuration
  (but training data is partly shared)
- In order to enhance robustness to hallucinated parts, the final feature vector is obtained by concatenating the layout features of all sub-configurations

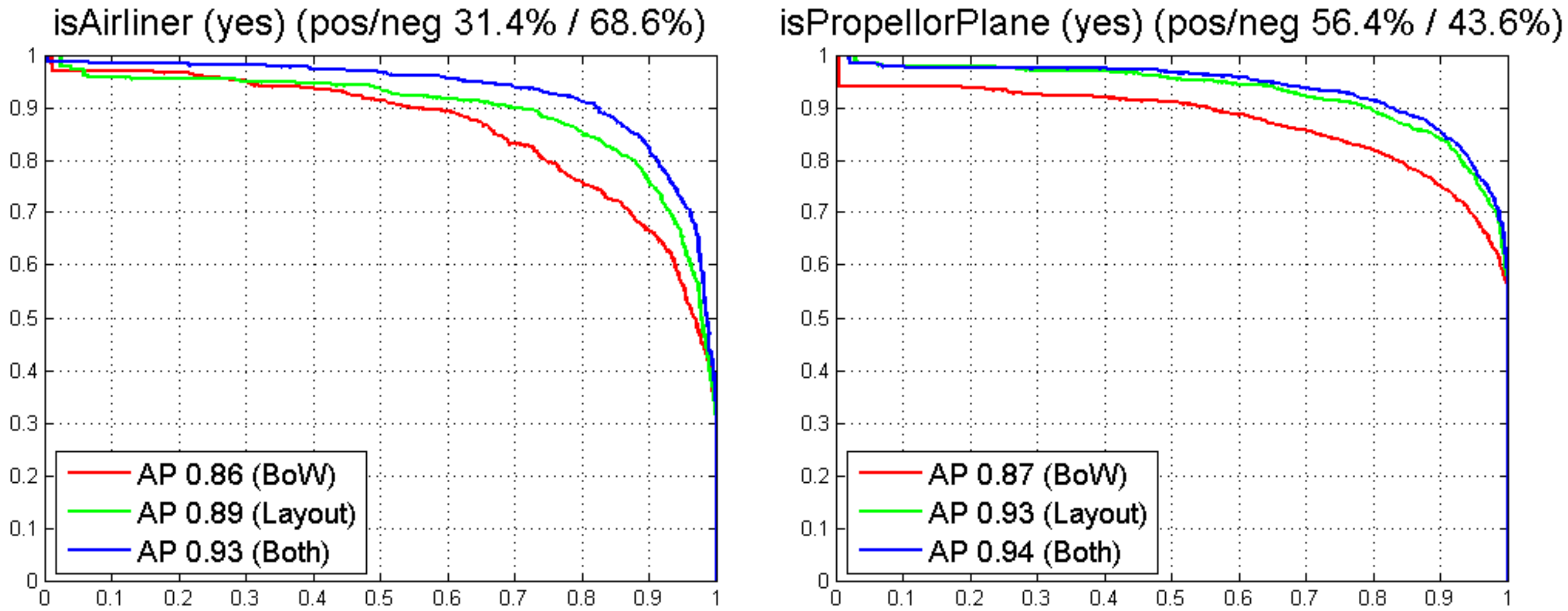
Example

- Can you say whether this layout refers to a jet airliner or a propellor plane?



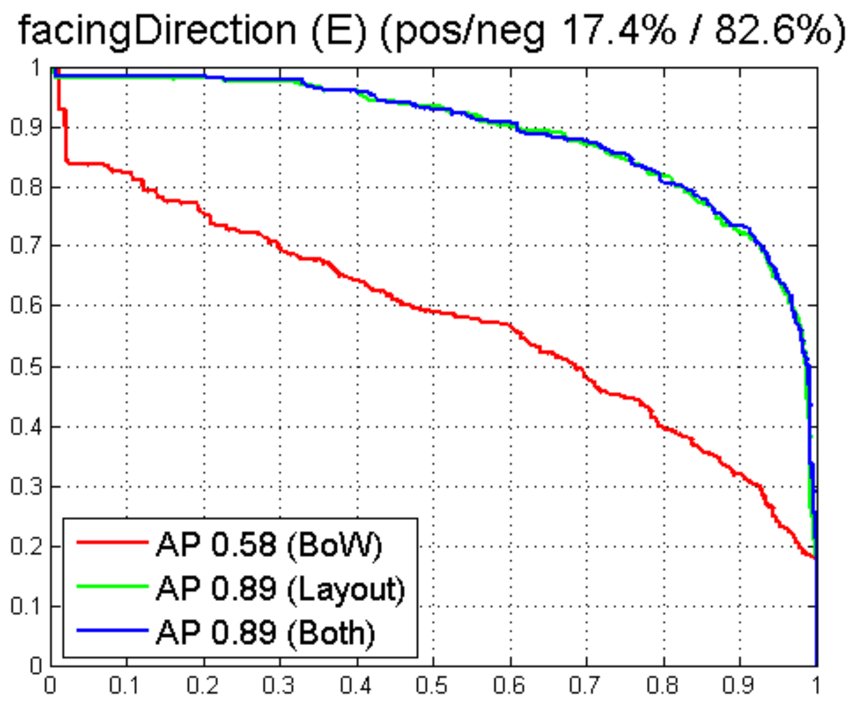


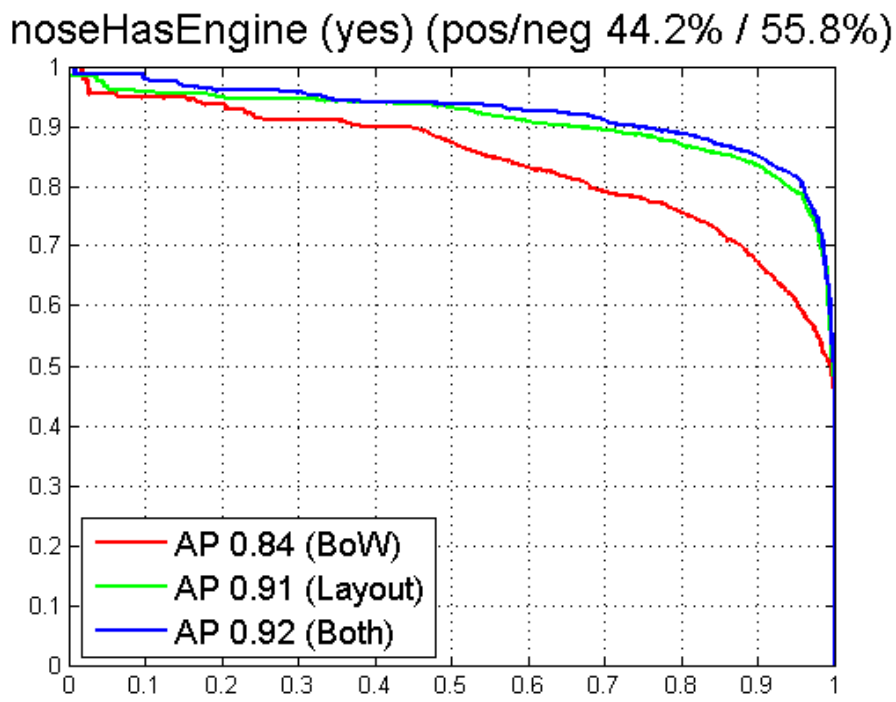
- Precision-recall curves for ground truth boxes in the test set:



Additional examples

- Precision-recall curves for ground truth boxes in the test set:





Using both layout and bags-of-words from all parts

- We extract the layout features (as explained on previous slides)

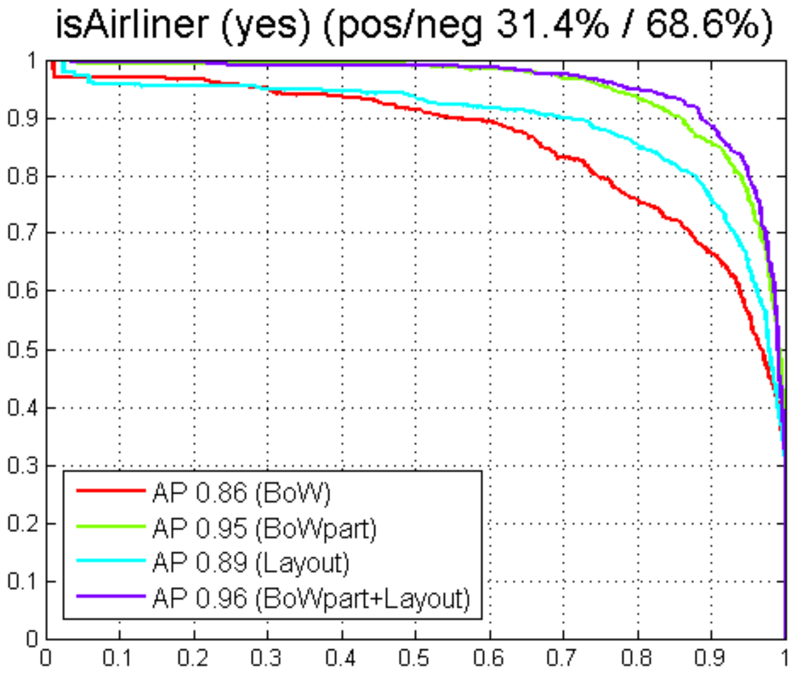
- We train first-layer attribute classifiers for each part+attribute pair using a single bag-of-words histogram as a feature

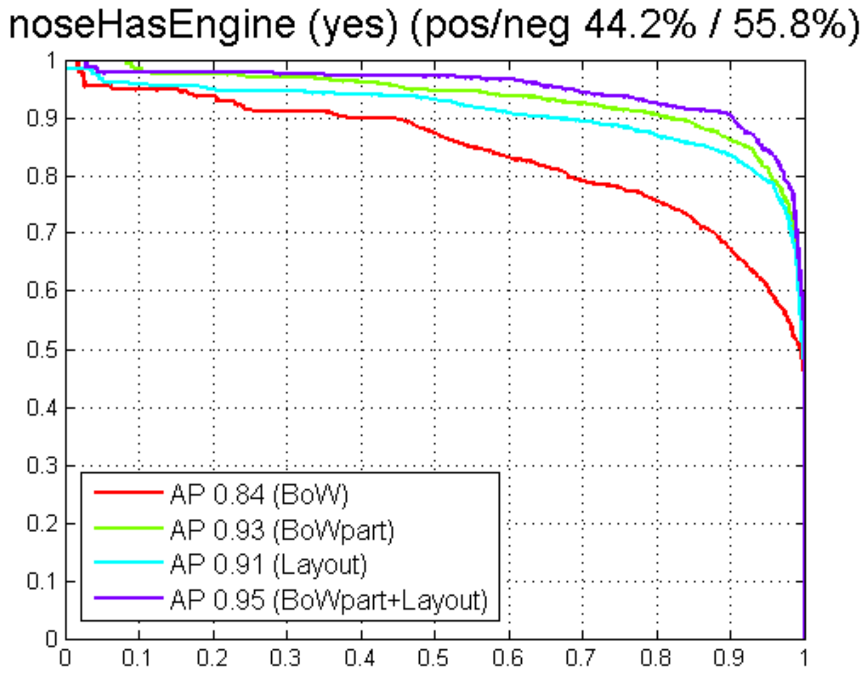
- We take the scores from the first-layer classifiers of detected parts and use them with the layout features to train the final second layer classifier for each attribute

- At test time, we apply the classifier that is designed for this particular detection configuration, i.e., different classifier for ”airplane+nose” detections than for ”airplane+nose+tail” detections

Results

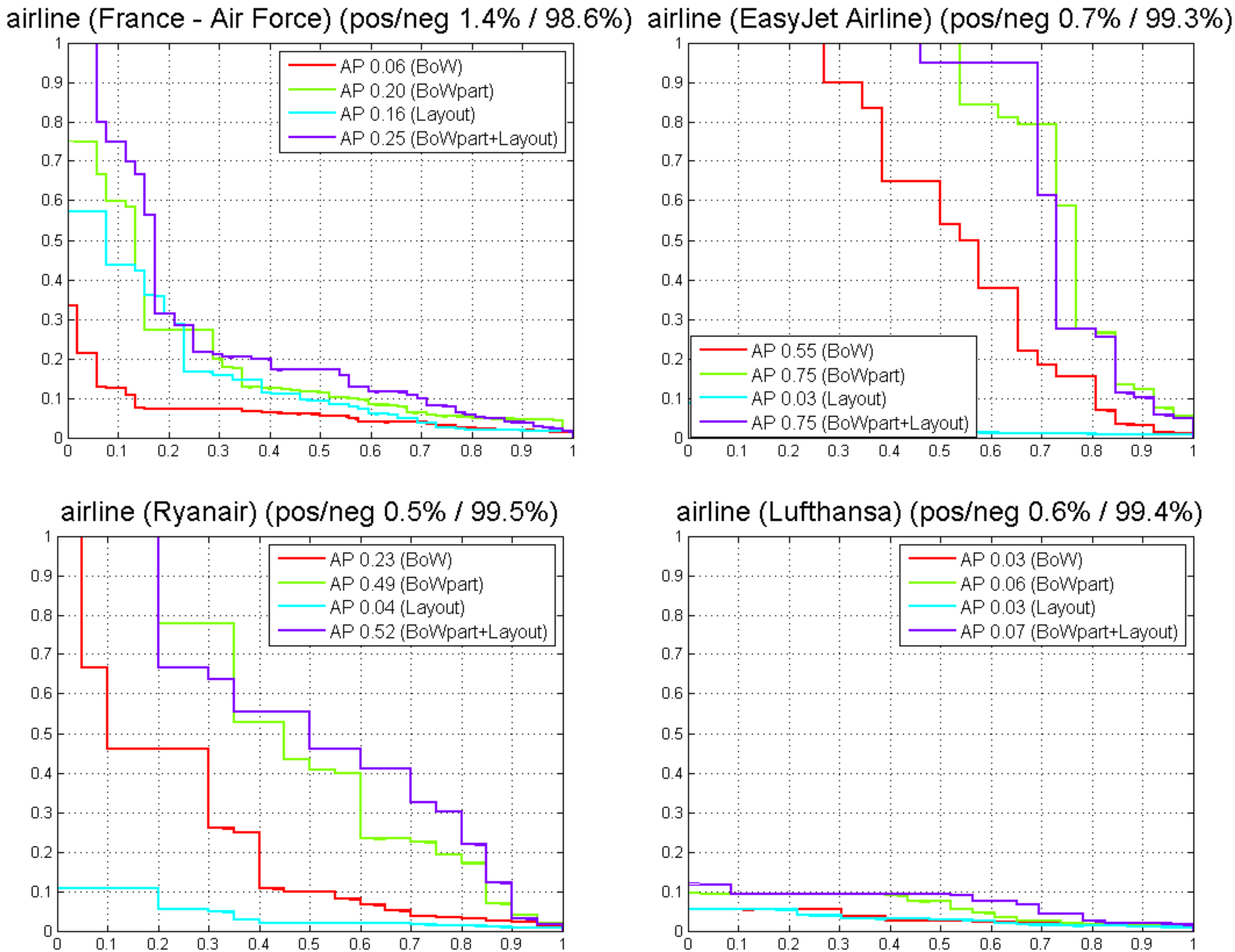
- Bag-of-words features from all parts + layout features give best results:





- Mean average precision over all 54 binary attributes:

BoW **0.40** Layout **0.43** BoWpart **0.53** BoWpart+Layout **0.56**


airline (France - Air Force) (pos/neg 1.4% / 98.6%)
AP 0.06 (BoW)
AP 0.20 (BoWpart)
AP 0.16 (Layout)
AP 0.25 (BoWpart+Layout)
airline (EasyJet Airline) (pos/neg 0.7% / 99.3%)
AP 0.55 (BoW)
AP 0.75 (BoWpart)
AP 0.03 (Layout)
AP 0.75 (BoWpart+Layout)
airline (Ryanair) (pos/neg 0.5% / 99.5%)
AP 0.23 (BoW)
AP 0.49 (BoWpart)
AP 0.04 (Layout)
AP 0.52 (BoWpart+Layout)
airline (Lufthansa) (pos/neg 0.6% / 99.4%)
AP 0.03 (BoW)
AP 0.06 (BoWpart)
AP 0.03 (Layout)
AP 0.07 (BoWpart+Layout)

Conclusion

- Part detections have potential to improve attribute predictions
- Part detections can be utilized in many ways
- Experiments show that bag-of-words features and layout features are complementary and best results are obtained by using both
- In future it would be necessary to combine object detection (object +parts) and attribute prediction into a single pipeline
- In addition, one could consider object detection and attribute prediction jointly (e.g. by using feedback from attribute classifiers to choose the best combination of part detections)

Overview

Objects in Detail

Parts & attributes

- A new dataset
- An object lexicon
- Localising parts
- Layouts
- Recognising attributes

Stuff in Detail

Texture

- A texture lexicon
- A new dataset
- Transformation invariant semantic

Parsing

Bottom-up inference

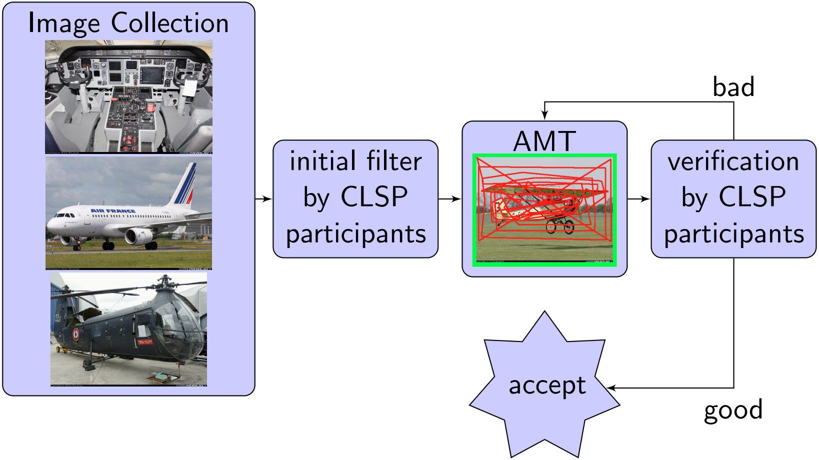
- Learning to merge
- Cascading
- Scoring regions by attributes

Annotations

Naomi P. Saphra
Carnegie Mellon University

August 3, 2012

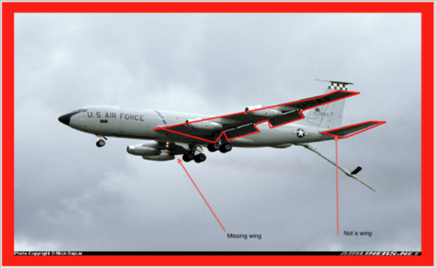
The Annotation Process



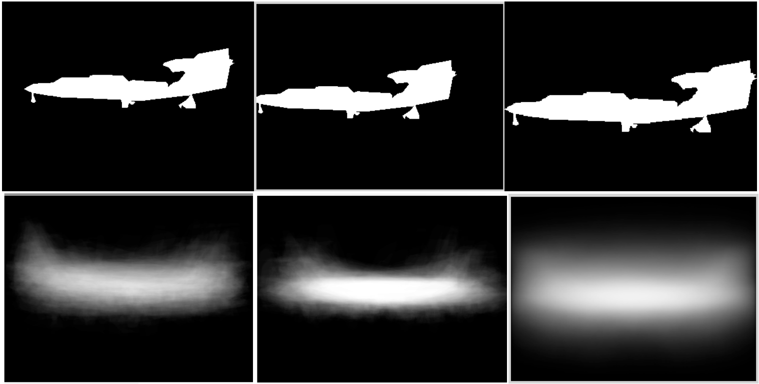
Collecting Data: Parts and Attributes

Check the examples below carefully.

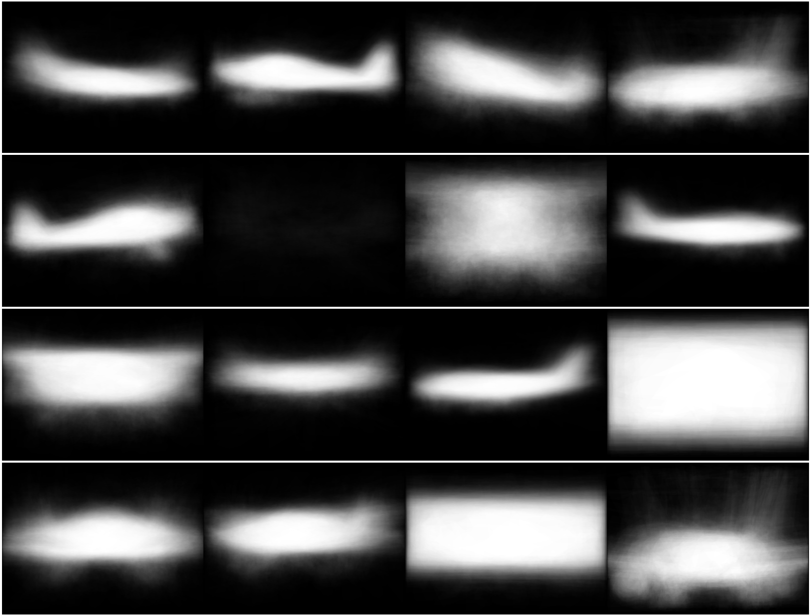
- **To add a polygon.** Click on the point where you want the new polygon to start, then on the second point, the third, and so on. The polygon is completed by going back to the first point, closing the figure.
- **To edit an existing polygon.** Click and drag any of the blue points on the polygon to adjust it.
- **To select a polygon.** Click on a control point or near a segment. The selected polygon appears in red.
- **To delete the selected polygon.** Press **d** or **D**.
- **To delete all polygons.** Press **R** (capital R).

Good Annotations	**Bad Annotations**
Good: marks the two wings separately; includes the whole extent until the junction with the fuselage; includes the wing flaps.	**Bad**: does not mark the right wing; marks an horizontal stabilizer as a wing.
	
	Bad: Does not extend to the junction-wing fuselage ; marks an horizontal stabilizer as a wing

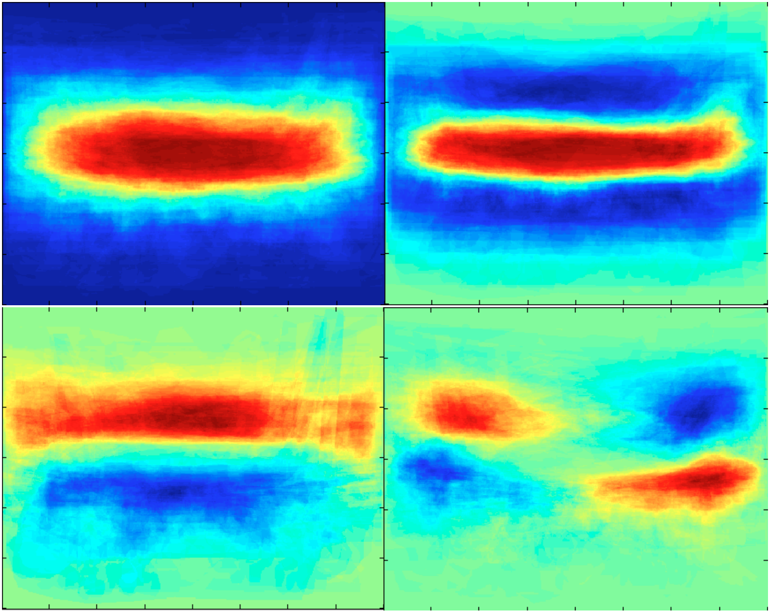
Getting To Know The Data



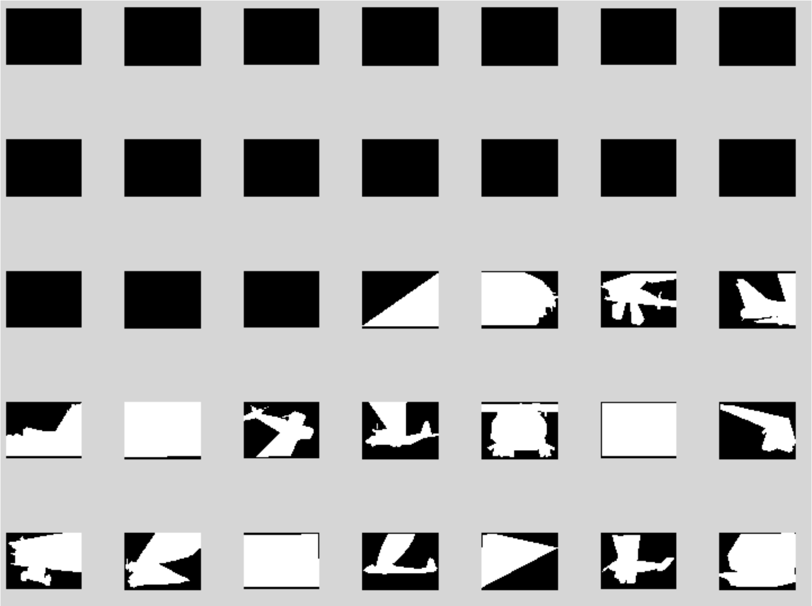
K-Means



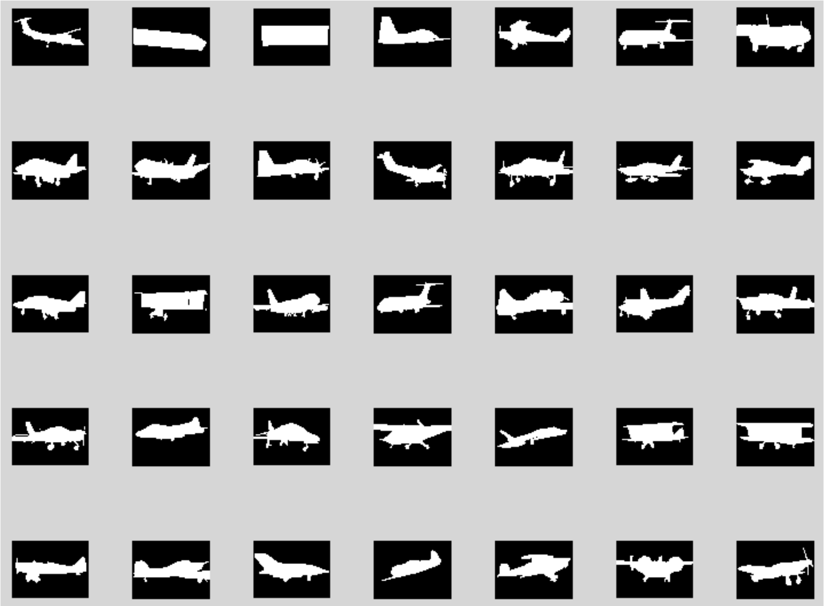
PCA: Eigenplanes



Gaussian: Unlikely

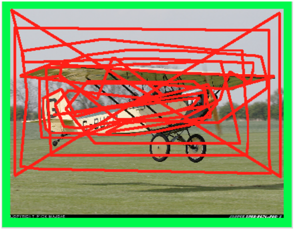


Gaussian: Likely



Annotation Problems

- Instructions had bounding boxes and polygons in same picture.
- Turkers didn't read instructions.
  - Thought they had to trace every outline.
  - Ended before desired end of nose or wing.
- Turkers were careless.
  - Miss parts.
  - Loose outlines.
- Didn't realize they were annotating a new part.
- Didn't bother annotating anything.
- Got frustrated.



Verifying Annotations: Manually

- Juho and Esa created tools for manually verifying annotations.
- 7700 planes, 10 parts, 3 annotations per part per plane per pass-through, some required several pass-throughs.
- Tool for correcting borderline polygons.

Verifying Annotations: Automatically

- PCA
- SVM
- Identify worst annotators, invite only best back to annotate other parts.

SVM: Metadata

- features
  - mask pixels
  - vertex count
  - annotator ID
  - time spent annotating
  - L1 normalized histogram of angles in polygon
  - PCA likelihood: Likelihood of annotation being an annotation of a *different* airplane part.
- combinations
  - baseline: Accept every annotation.
  - mask
  - vertex count, annotator ID, time
  - angle, vertex count, annotator ID, time
  - mask, vertex count, annotator ID, time
  - angle, vertex count, annotator ID, time, PCA likelihood

SVM: Results

	airplane	vert stabilizer	nose
baseline	76	92	94
mask	80	94	94
angle, CAT	80	92	95
CAT	79	92	95
mask, CAT	82	92	94
angle, mask, CAT, PCA	76	92	94

CAT = vertex *C*ount, *A*nnotator ID, *T*ime spent annotating

Future Work

- Polygon edge-feature edge similarity
- Use new part classifiers to bootstrap validation
- Incorporate these tools more into verification process

Overview

Objects in Detail

Parts & attributes

- A new dataset
- An object lexicon
- Localizing parts
- Layouts
- Recognizing attributes

Stuff in Detail

Texture

- A texture lexicon
- A new dataset
- Transformation invariant semantic

Parsing

Bottom-up inference

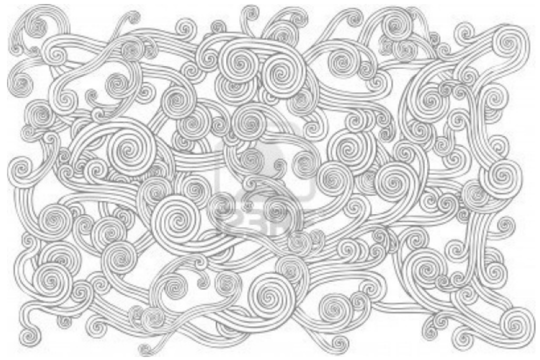
- Learning to merge
- Cascading
- Scoring regions by attributes

Nameable textures

Iasonas Kokkinos
Ecole Centrale Paris

Subhransu Maji
TTI-Chicago

Sammy Mohamed
Stony Brook

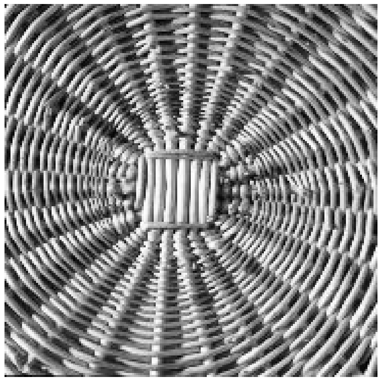
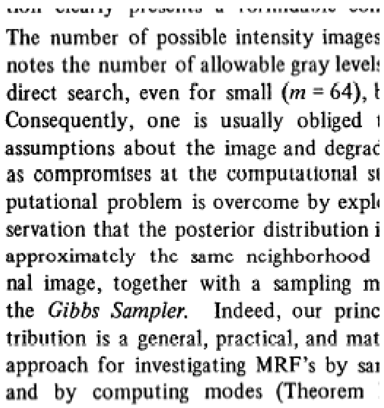


Visual texture

– Natural processes




– Man-made structures

The number of possible intensity images
notes the number of allowable gray level
direct search, even for small (m = 64), t
Consequently, one is usually obliged t
assumptions about the image and degrac
as compromises at the computational st
putational problem is overcome by expl
servation that the posterior distribution i
approximately the same neighborhood
nal image, together with a sampling m
the *Gibbs Sampler*. Indeed, our princ
tribution is a general, practical, and mat
approach for investigating MRF's by sa
and by computing modes (Theorem


What defines a texture?

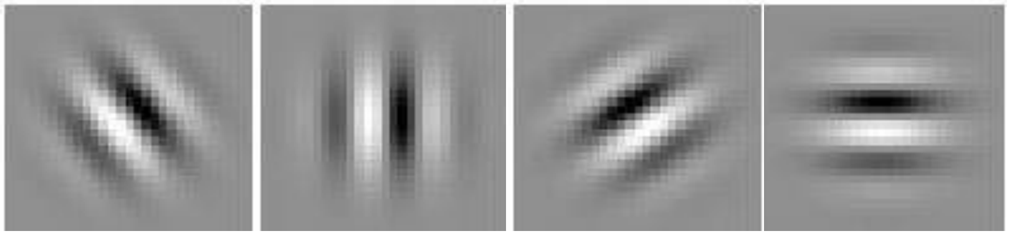
- What is common in these images?
  - No common deterministic model
  - Statistical properties..

“What features and statistics are characteristics of a texture pattern, so that texture pairs that share the same features and statistics cannot be told apart by pre-attentive human visual perception?” ---- Julesz 1960s-1980s

Texture analysis and image processing

2D Gabor-filters $G_{\omega_1,\omega_2,\sigma}(x,y) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)\exp(j\omega_1 x+\omega_2 y)$

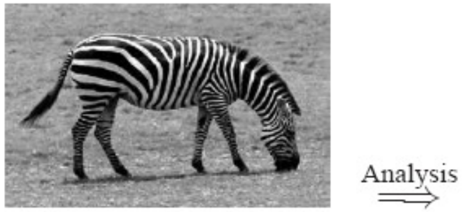
$\angle\omega$

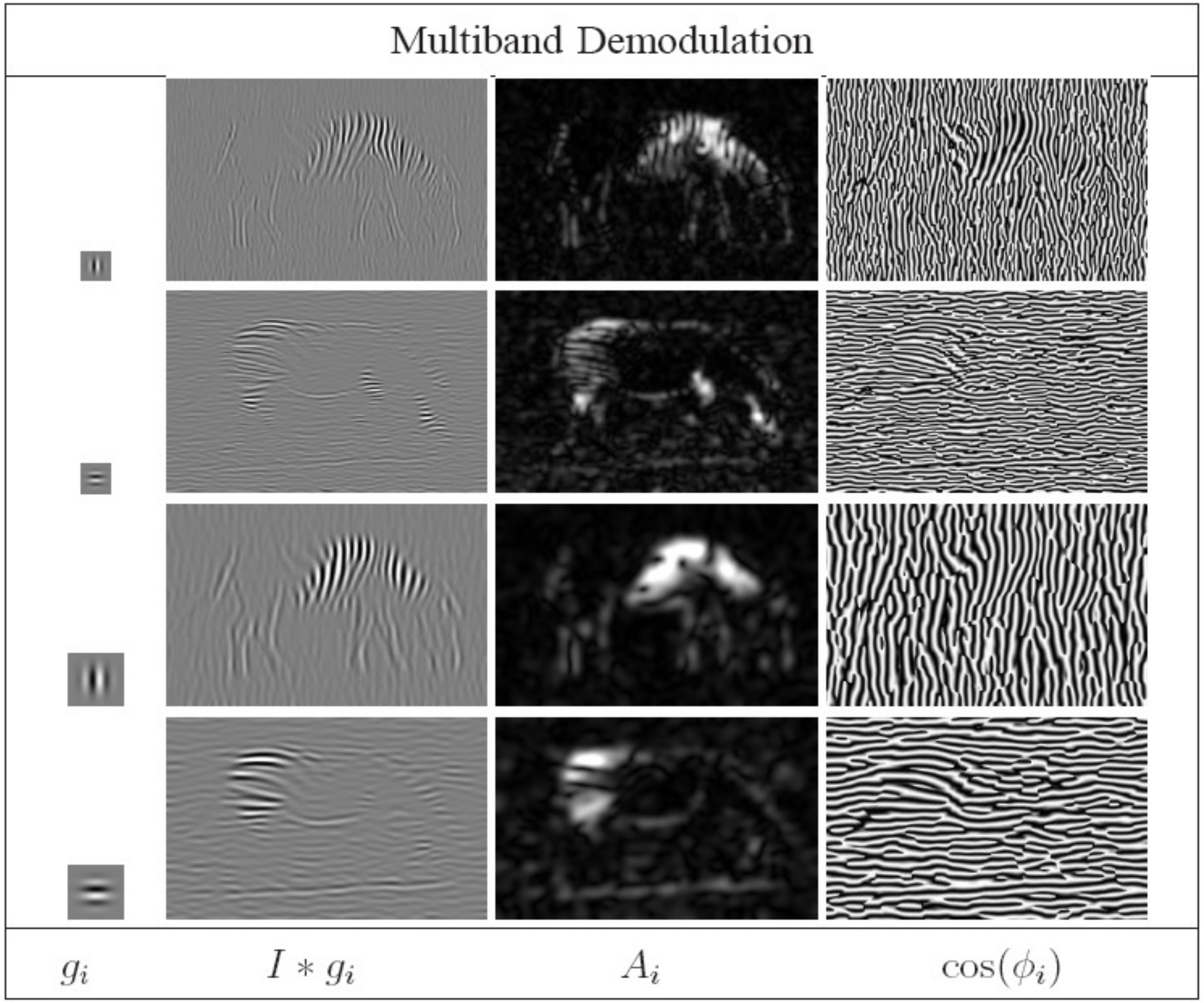


$|\omega|$

`periodicity detectors'

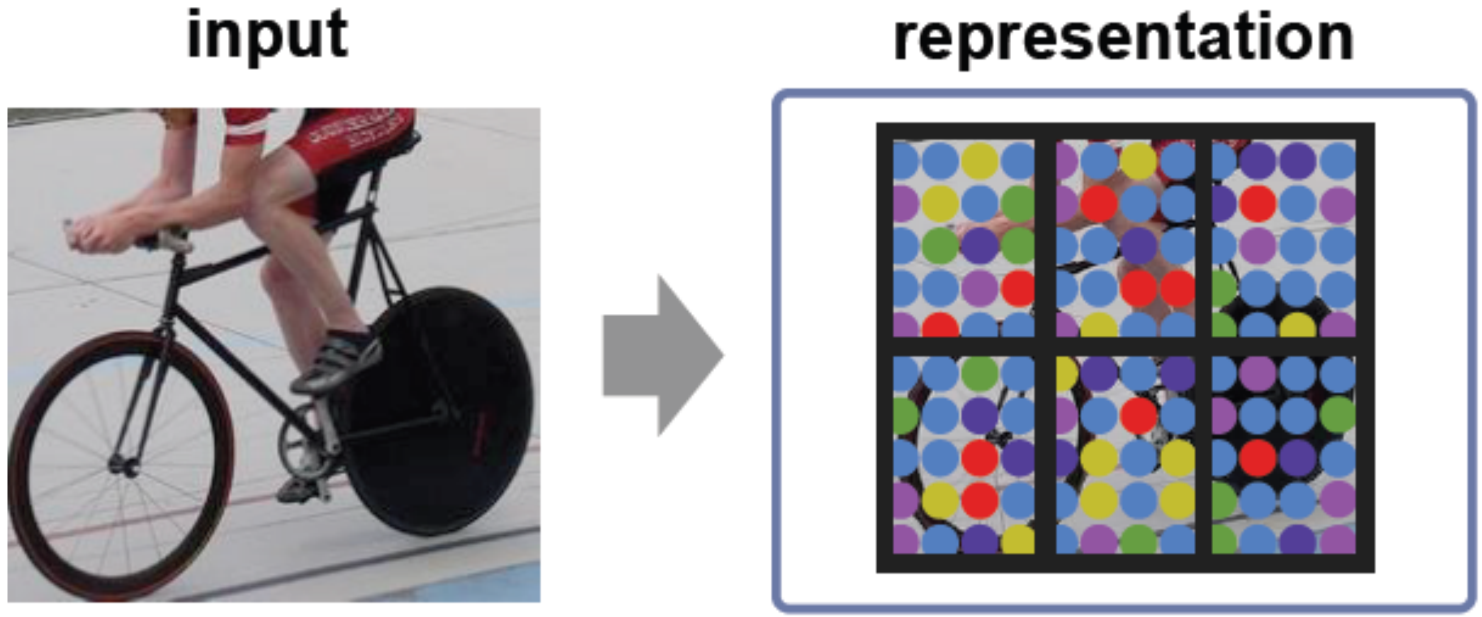
Multi-scale and multi-orientation texture analysis





Texture analysis and `visual words'

- K-means on SIFT descriptors ~ textons
- Bag-of-Words/Spatial Pyramid models



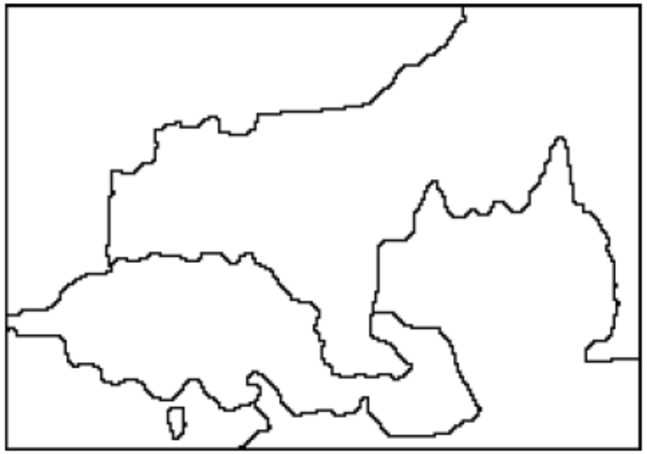
What can we do with texture?

High-dimensional description of an image patch

- Roughly translation invariant (stationarity assumption)
- Potentially scale & orientation invariant

Texture = features

Texture segmentation

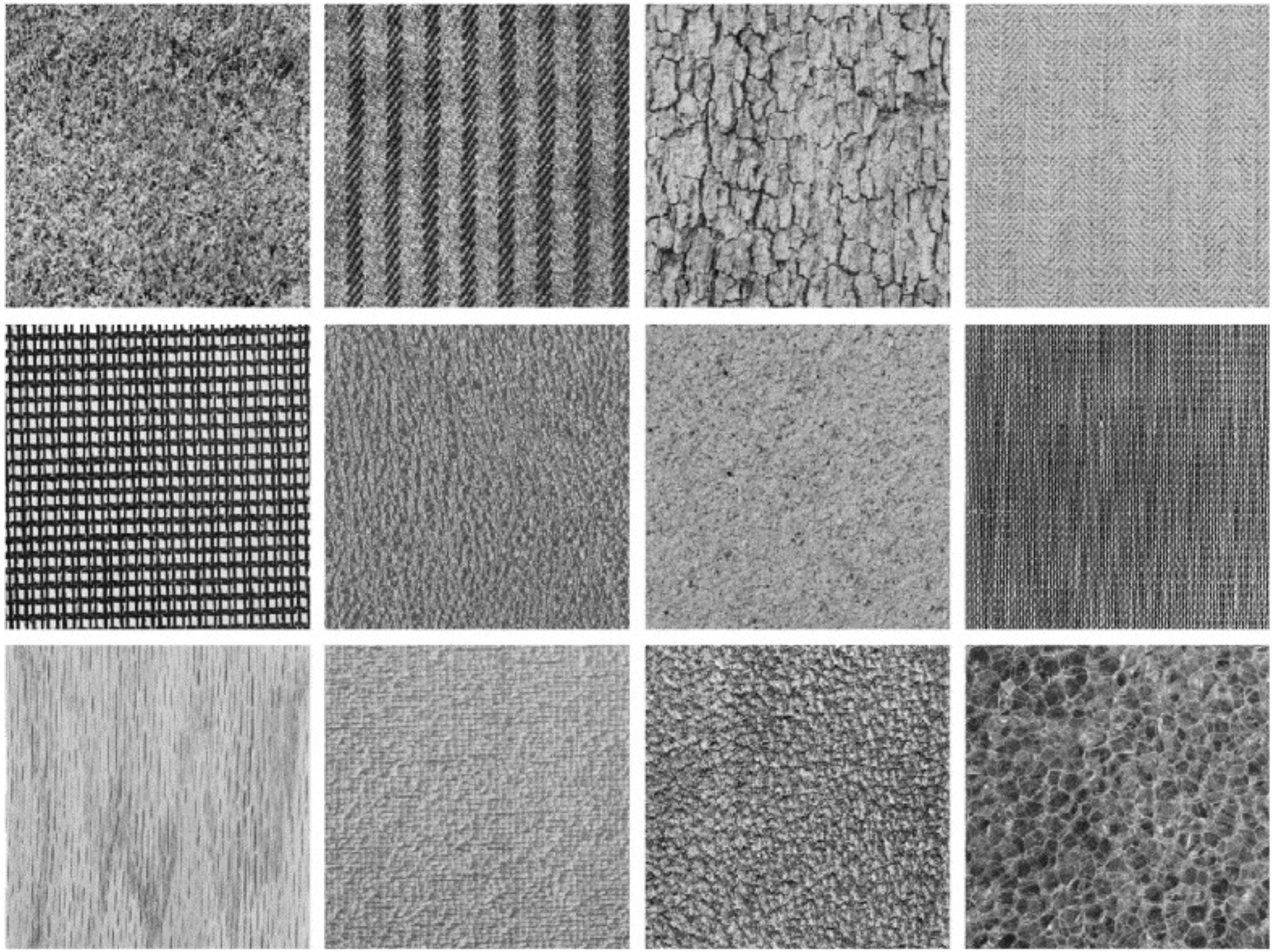


Zhu & Yuille, Region Competition, PAMI 1996

Delong et al, Fast Approximate Energy Minimization with Label Costs, IJCV 2012

Texture classification

Brodatz 98 textures (Caltech 101 of the 90's)



Texture-based labelling

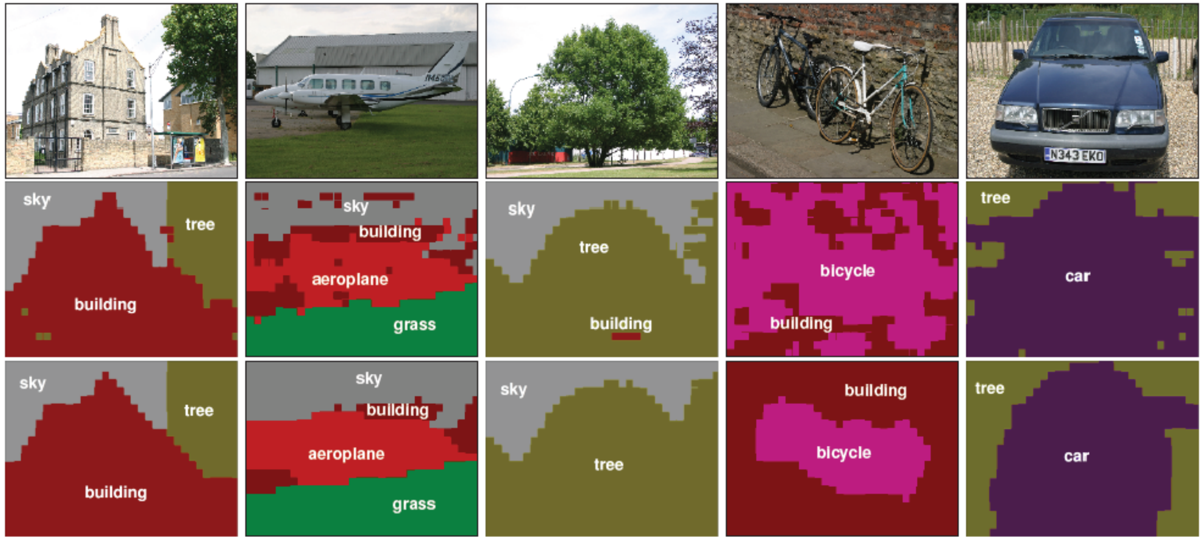


Figure 3. (Best viewed in color). Example images (top), and segmentations using PLSA (middle) and PLSA-MRF (bottom), with topics learned from image labels.

Region Classification with Markov Field Aspect Models, Verbeek and Triggs, CVPR 07

Textonboost for image understanding, Shotton et al, IJCV 07

What can we do with texture? (revisited)

Soaring heights and unfathomable lows of vision (recognition, segmentation)

We want something in between

Not too high: decoupled from object-specific aspects (color, pose, occlusion..)

- -stationary & `pure'
- -shareable across categories

Not too low: semantic (e.g. `striped', `dotted', `honeycombed', etc.)

- -interpretable by humans
- -categorical

Overview

Objects in Detail

Parts & attributes

- A new dataset
- An object lexicon
- Localizing parts
- Layouts
- Recognizing attributes

Stuff in Detail

Texture

- A texture lexicon
- A new dataset
- Transformation invariant semantic

Parsing

Bottom-up inference

- Learning to merge
- Cascading
- Scoring regions by attributes

Nameable textures

Human-centric merit: use texture in image queries

Vision-centric merit: stratification of `texture jungle’, `debuggable’ vision models

Is there a proper lexicon for textures?

COGNITIVE SCIENCE Vol 21 (2) 1997, pp. 219–246 ISSN 0364-0213

The Texture Lexicon: Understanding the Categorization of Visual Texture Terms and Their Relationship to Texture Images

NALINI BHUSHAN
Smith College

A. RAVISHANKAR RAO
IBM Watson Research Center

GERALD L. LOHSE
The Wharton School, University of Pennsylvania

In this paper we present the results of two experiments. The first is on the categorization of texture words in the English language. The goal was to determine whether there is a common basis for subjects’ groupings of words related to visual texture, and if so, to identify the underlying dimensions used to categorize those words.

Eleven major clusters were identified through hierarchical cluster analysis, ranging from ‘random’ to ‘repetitive’. These clusters remained intact in a

COGNITIVE SCIENCE Vol 21 (2) 1997, pp. 219–246 ISSN 0364-0213

The Texture Lexicon: Understanding the Categorization of Visual Texture Terms and Their Relationship to Texture Images

NALINI BHUSHAN
Smith College

A. RAVISHANKAR RAO
IBM Watson Research Center

GERALD L. LOHSE
The Wharton School, University of Pennsylvania

In this paper we present the results of two experiments. The first is on the categorization of texture words in the English language. The goal was to determine whether there is a common basis for subjects' groupings of words related to visual texture, and if so, to identify the underlying dimensions used to categorize those words.

Eleven major clusters were identified through hierarchical cluster analysis, ranging from 'random' to 'repetitive'. These clusters remained intact in a

Intended to be a thorough list of words used in describing surface texture.

Started with a list of 367 words, cut down to 98.

Examples: entwined facetted fibrous flecked flowing fractured freckled frilly furrowed gauzy gouged grooved holey interlaced intertwined knitted lacelike latticed lined matted meshed messy mottled netlike perforated periodic pitted pleated porous potholed random regular repetitive rhythmic ridged rumpled scaly scrambled spattered spiralled sprinkled stained stratified striated studded twisted veined webbed winding wizened woven …….

Challenges

Several words are not easy to pin down:

Scrambled, regular, messy, jumbled, random, disordered, indefinite, complex…

Based on a Google image query for each word, we assigned to each word a level of difficulty.

List of words with difficulty <7/10:

Uniform, Smooth, Dotted, Checkered, Grid, Spotted, Polka-Dotted, Waffled, Marbled, Zigzagged, Corrugated, Honeycombed, Speckled, Fibrous, Flecked, Facetted, Flowing, Fractured, Flecked, Frilly, Furrowed, Gauzy, Gouged, Grooved, Holey, Interlaced, Intertwined, Knitted, Lacelike, Latticed, Whirly, Swirly, Ribbed, Cracked, Banded, Wrinkled, Crosshatched

Overview

Objects in Detail

Parts & attributes

- A new dataset
- An object lexicon
- Localizing parts
- Layouts
- Recognizing attributes

Stuff in Detail

Texture

- A texture lexicon
- A new dataset
- Transformation invariant semantic

Parsing

Bottom-up inference

- Learning to merge
- Cascading
- Scoring regions by attributes

Google query results for `Ribbed’

`Good’

`Partially good’

`Wrong’

Additional challenges: duplicates, watermarks, resolution, blur, noise

Strategy: get good data for now, and leave partial data for later

Amazon Turk instructions

Annotation instructions for Honeycombed textures

Task: classify images as Good, Partially good, Bad.

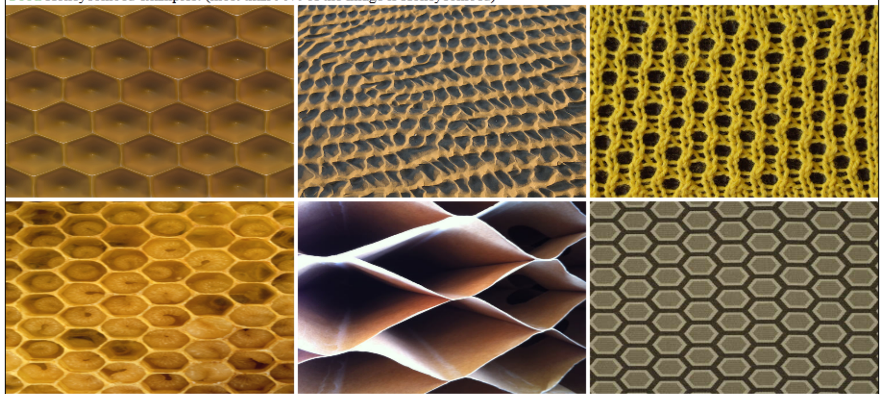
Good: most (more than 90%) of the image is Honeycombed.

Partially good: only part (less than 90%) of the image is Honeycombed.

Bad: the image has no Honeycombed region.

To decide between good and partially good, estimate the number of pixels that are Honeycombed.

Good Honeycombed examples: (most than 90% of the image is Honeycombed)



Partially good Honeycombed examples: (less than 90% of the image is Honeycombed)

Bad Honeycombed examples:(the image has no Honeycombed region)

Annotation instructions for Polka-dotted textures

Task: classify images as Good, Partially good, Bad.

Good: most (more than 90%) of the image is Polka-dotted.

Partially good: only part (less than 90%) of the image is Polka-dotted.

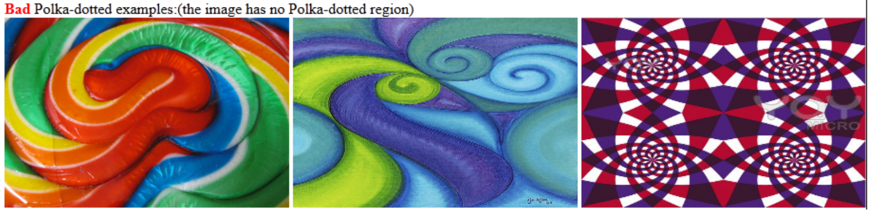
Bad: the image has no Polka-dotted region.

To decide between good and partially good, estimate the number of pixels that are Polka-dotted.

Good Polka-dotted examples: (most than 90% of the image is Polka-dotted)

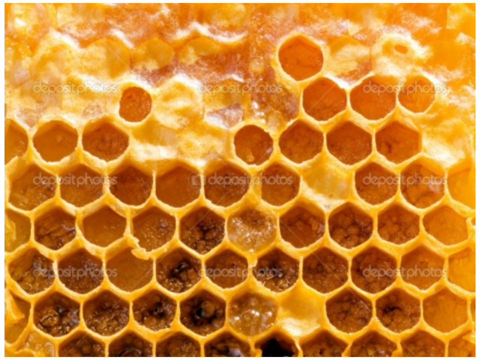
Partially good Polka-dotted examples: (less than 90% of the image is Polka-dotted)

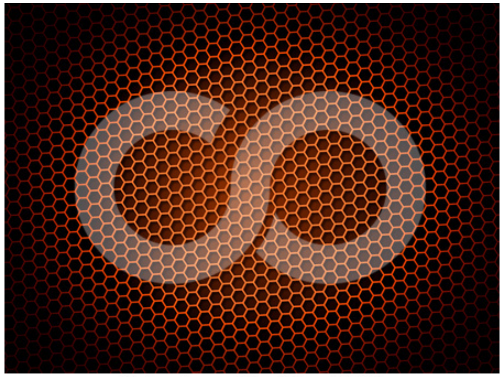
Bad Polka-dotted examples:(the image has no Polka-dotted region)



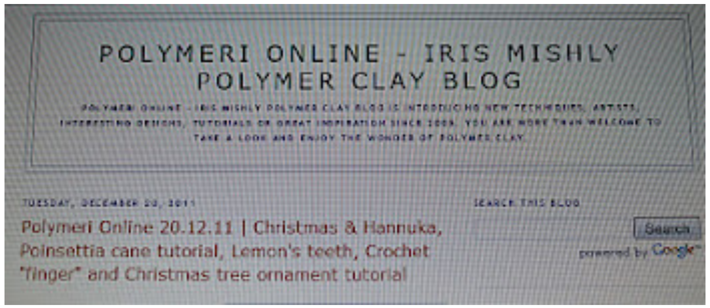
Validation results: honeycombed

3/3 good



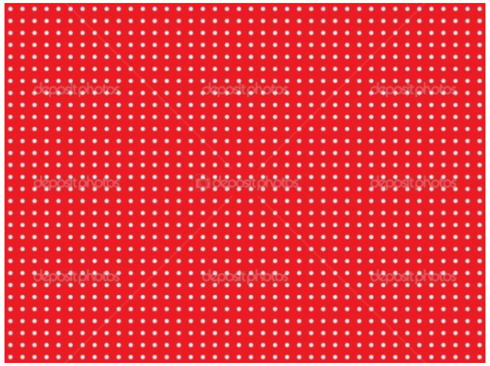


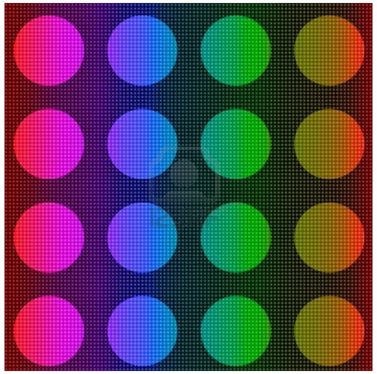
3/3 bad

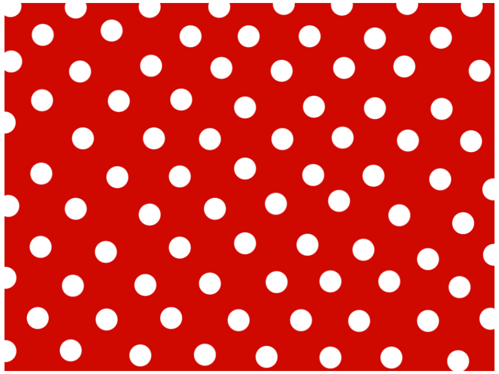


Validation results: polka-dotted

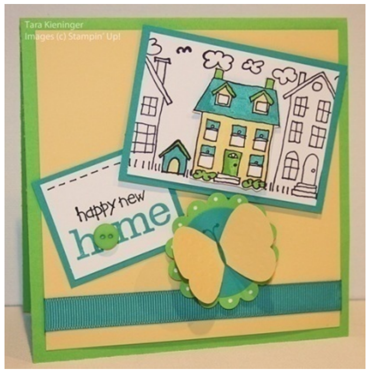
3/3 good

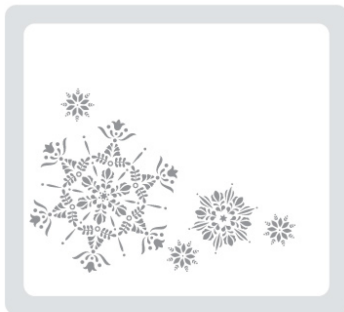






3/3 bad



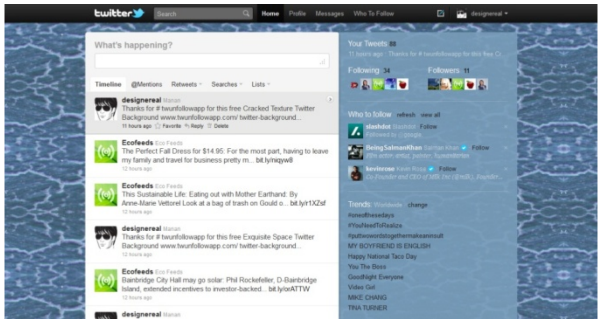


Validation results: cracked

3/3 good

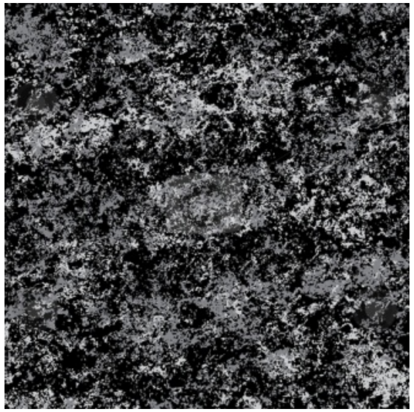


3/3 bad



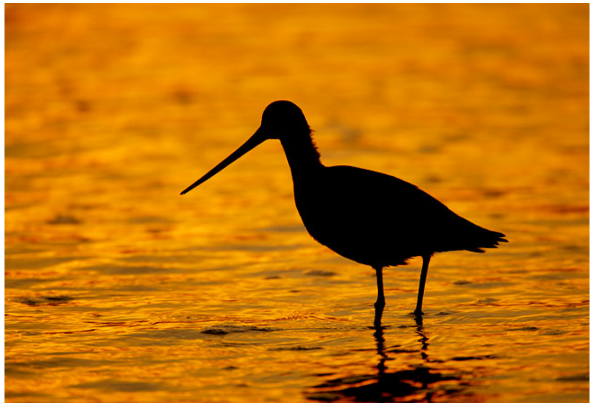
Validation results: marbled

3/3 good




© arena creative * www.ClipartOf.com/90675

3/3 bad

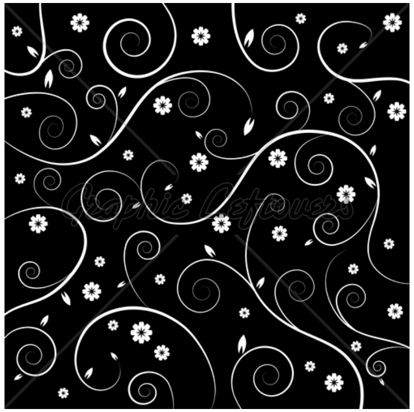


Validation results: swirly

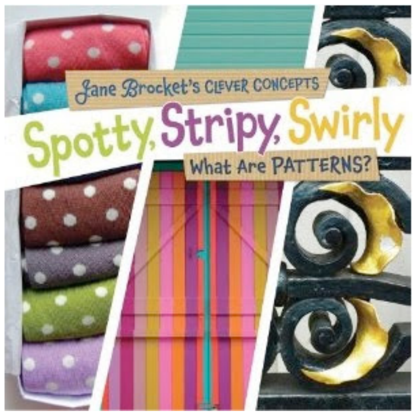
Brodatz:

3/3 good

© arena creative * www.ClipartOf.com/88211

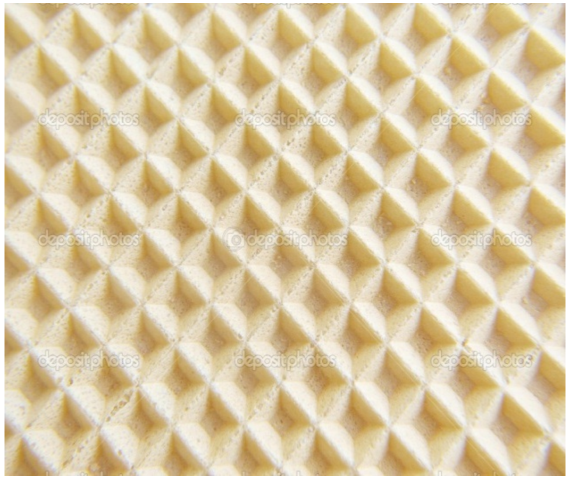


3/3 bad



Validation results: waffled

3/3 good



3/3 bad



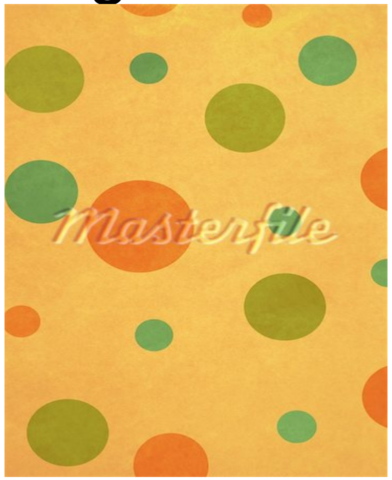
Validation results: wrinkled

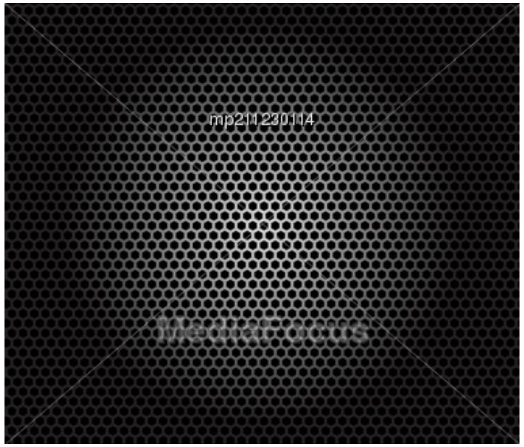
3/3 good

3/3 bad

Validation results: spotted

3/3 good





3/3 bad

Validation results: knitted

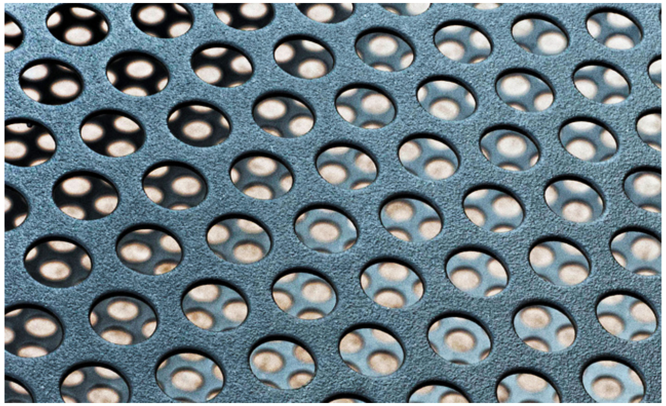
3/3 good

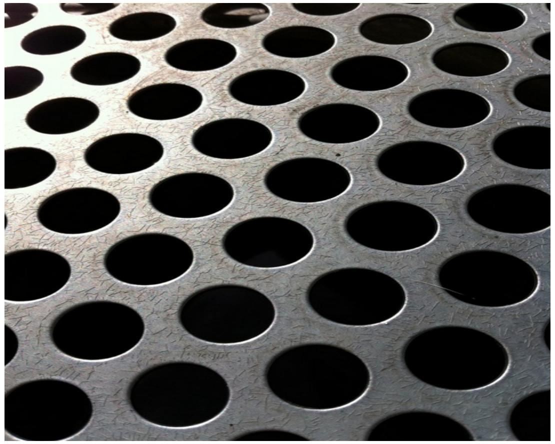


3/3 bad

Validation results: holey

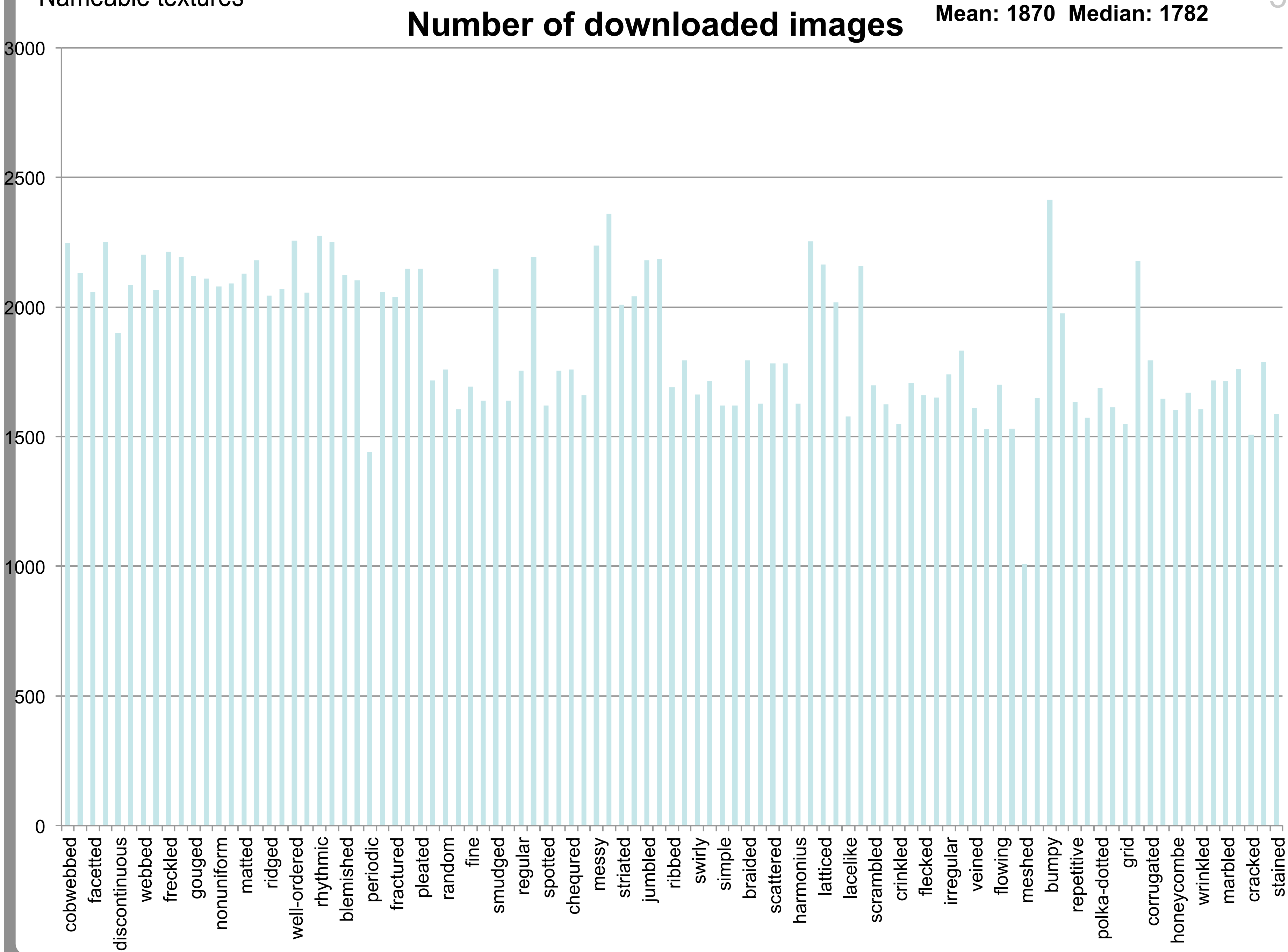
3/3 good



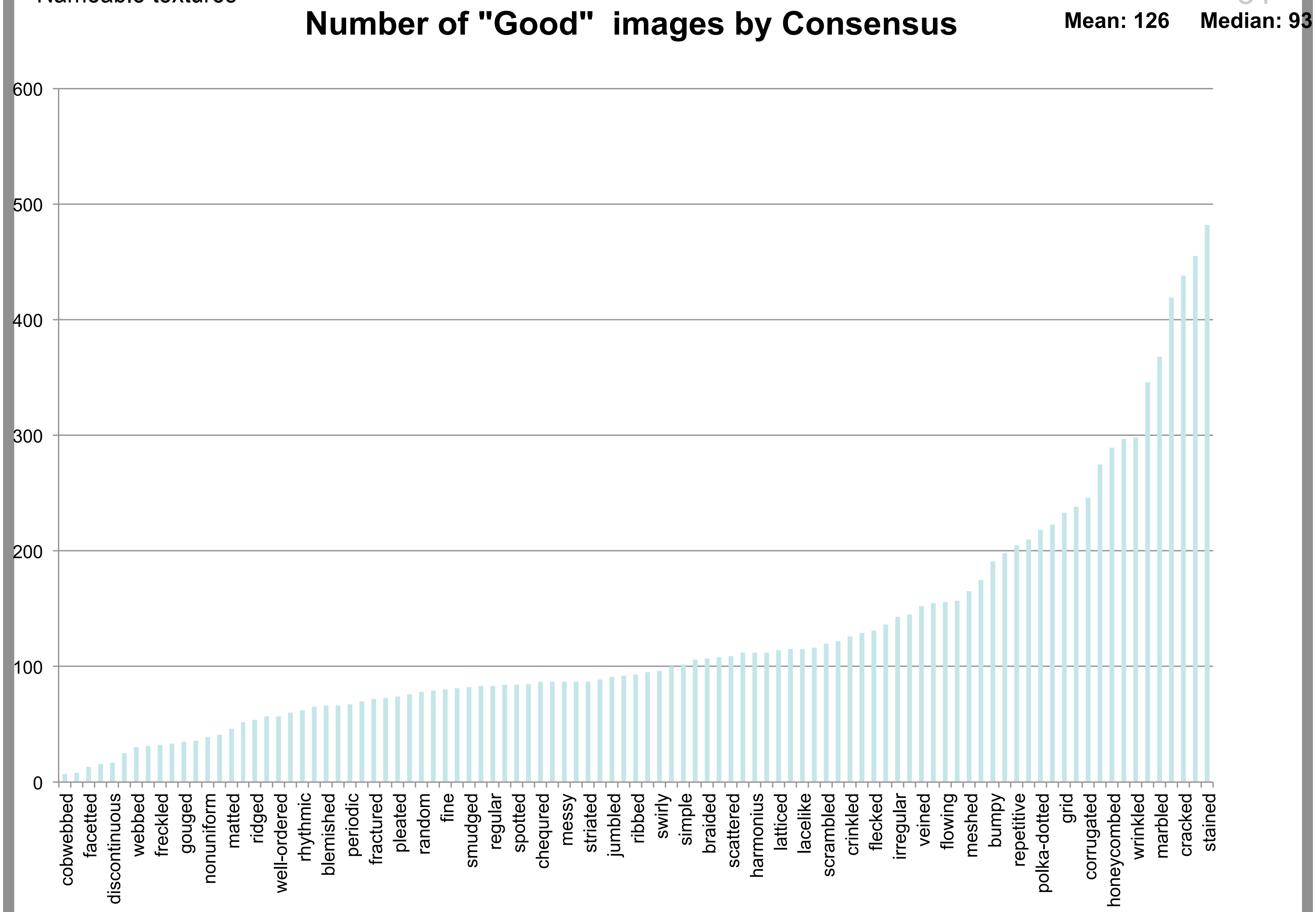


3/3 bad

Nameable textures



Nameable textures



Overview

Objects in Detail

Parts & attributes

- A new dataset
- An object lexicon
- Localizing parts
- Layouts
- Recognizing attributes

Stuff in Detail

Texture

- A texture lexicon
- A new dataset
- Transformation invariant semantic

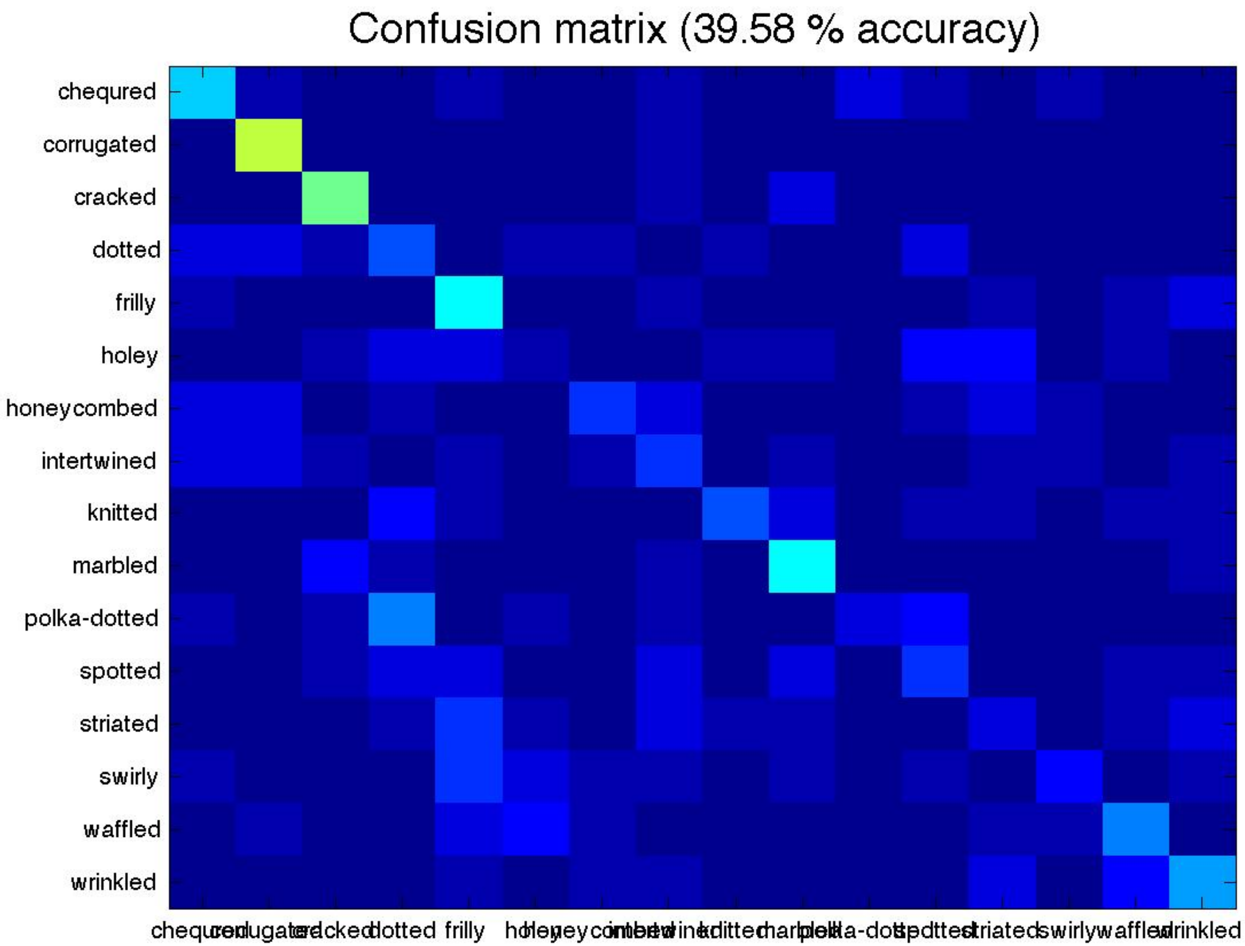
Parsing

Bottom-up inference

- Learning to merge
- Cascading
- Scoring regions by attributes

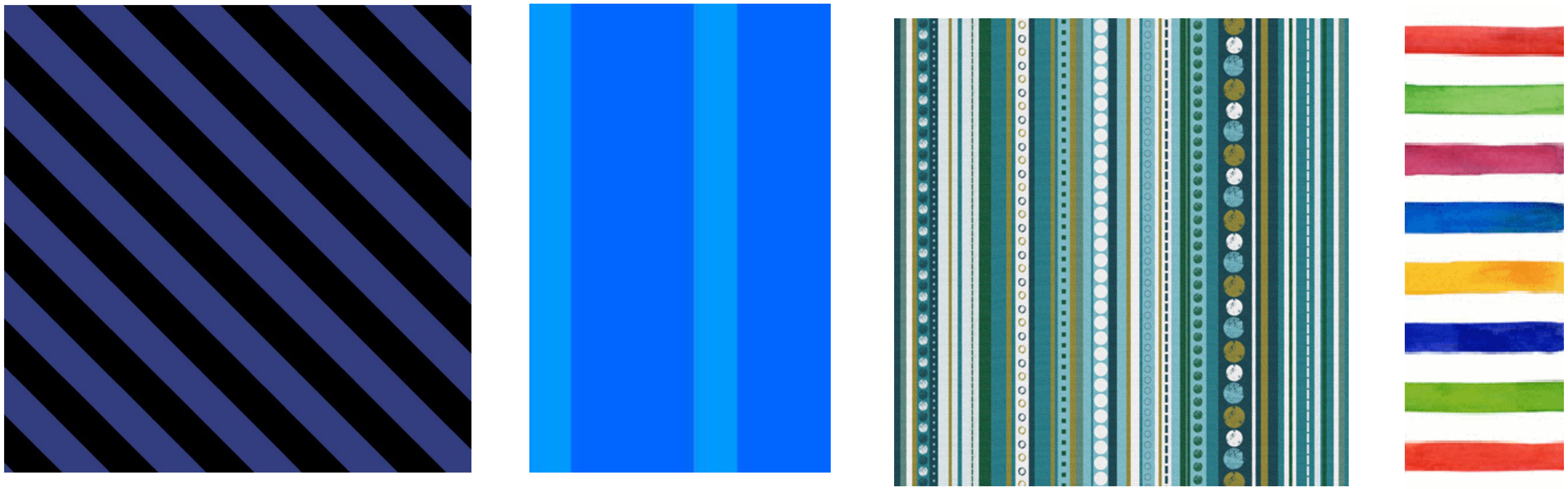
Baseline results

SVM classifier for bag-of-words with k-chi kernel $k(x,y) = \sum_i \frac{2x_i y_i}{x_i + y_i}$



Intra-category variability

Images for `banded’ category



Scale and orientation: nuisance parameters

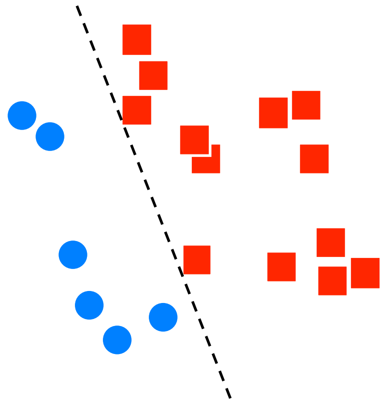
Sneaking in

mom’s keychain grandma’s keychain dad’s keychain

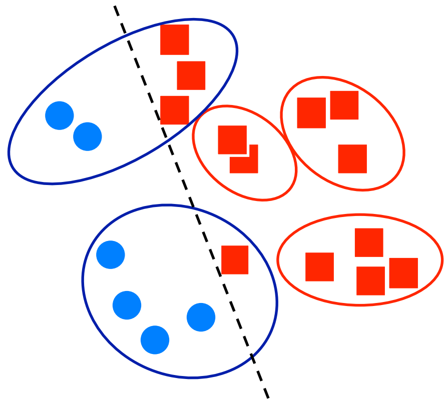
We know that dad cannot enter

Which key should we try?

Multiple Instance Learning



Typical Learning



Multiple Instance Learning

$$S = \{(x^i, y^i)\}$$

$$S = \{(\underbrace{\{x^{i,1}, \dots, x^{i,|B_i|}\}}_{B_i}, y^i)\}$$

$$y^i \in \{0,1\} \quad x^i \in \mathcal{X}$$

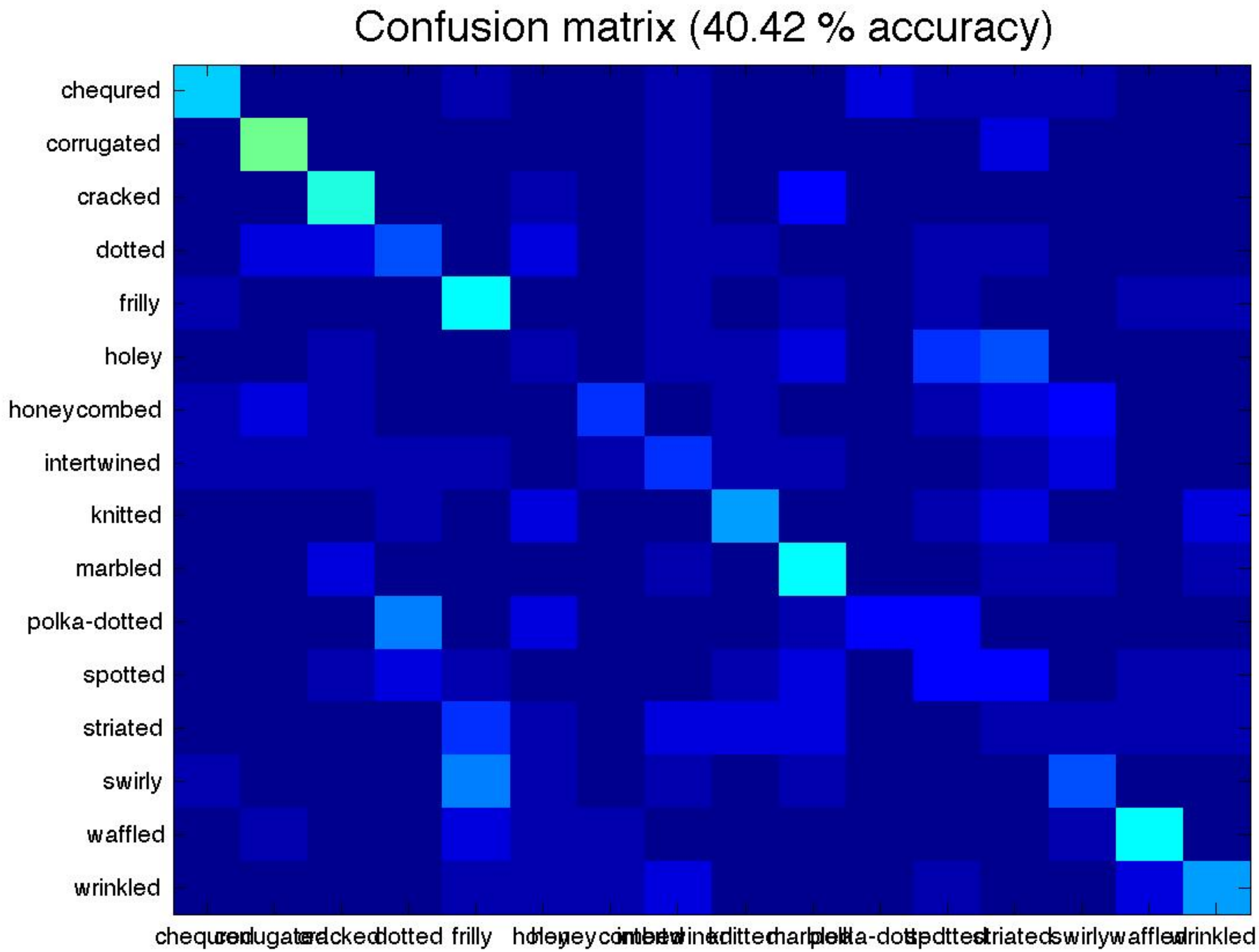
$$F : \mathcal{X} \to \{0,1\}$$

Positive bag: at least one instance should be positive

Negative bag: no instance should be positive

$$\max_{x \in B_i} F(x) = y^i$$

Multiple Instance Learning + BOW



Fisherfeatures

BOW problem: part of the signal is `lost in quantization’

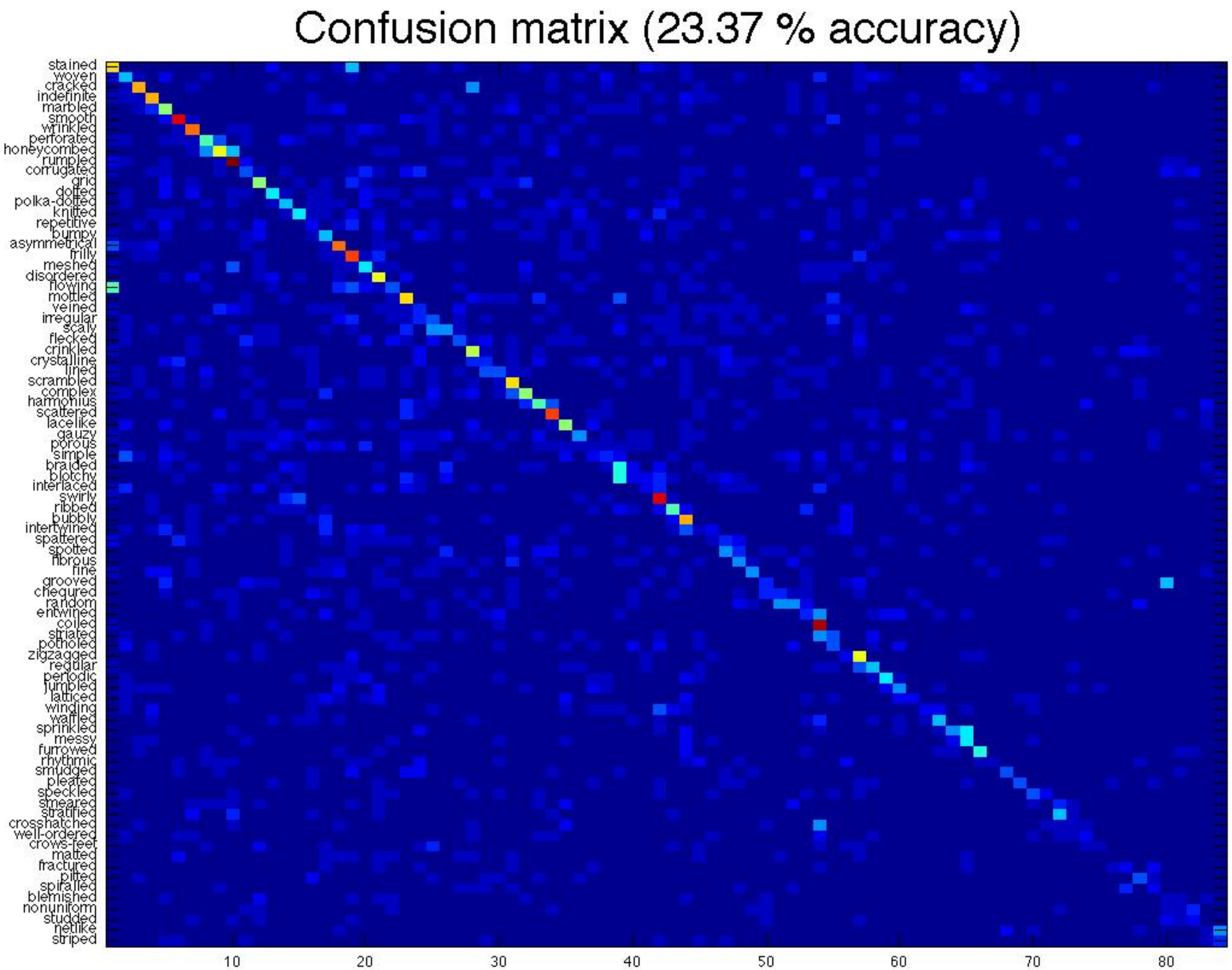
`Fisherfeatures’ : replace vector quantization through GMMs

T. Jaakkola and D. Haussler, Exploiting Generative Models in Discriminative Classifiers. NIPS 1998

F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for image classification. ECCV, 2010.

K. Chatfield, A. Vedaldi, L. Victor, and Z. Zisserman. The devil is in the details: an evaluation of recent feature encoding methods, BMVC 2011

The more, the merrier



Nameable textures: a roadmap for visual textures

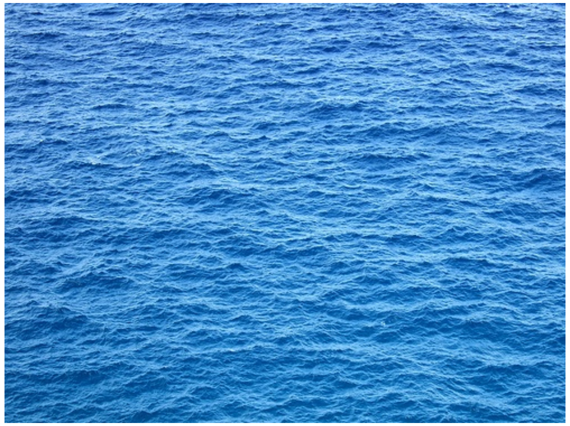
A new dataset for texture category classification

Multiple Instance Learning & Fishervectors for texture models

Future work:

- sliding window/superpixel-based scoring
- texture-based superpixel merging
- texture-based object detection
- semi-supervised learning





Texture lexicon: a stratification of visual textures

A new dataset for nameable texture classification

98 Categories, 30-100 words per category

Cast texture representation in multi-class classification terms

Multiple Instance Learning of texture models

Bottom-Up Image Parsing

Part 1

Karén Simonyan, David Weiss,
Andrea Vedaldi, Ben Taskar

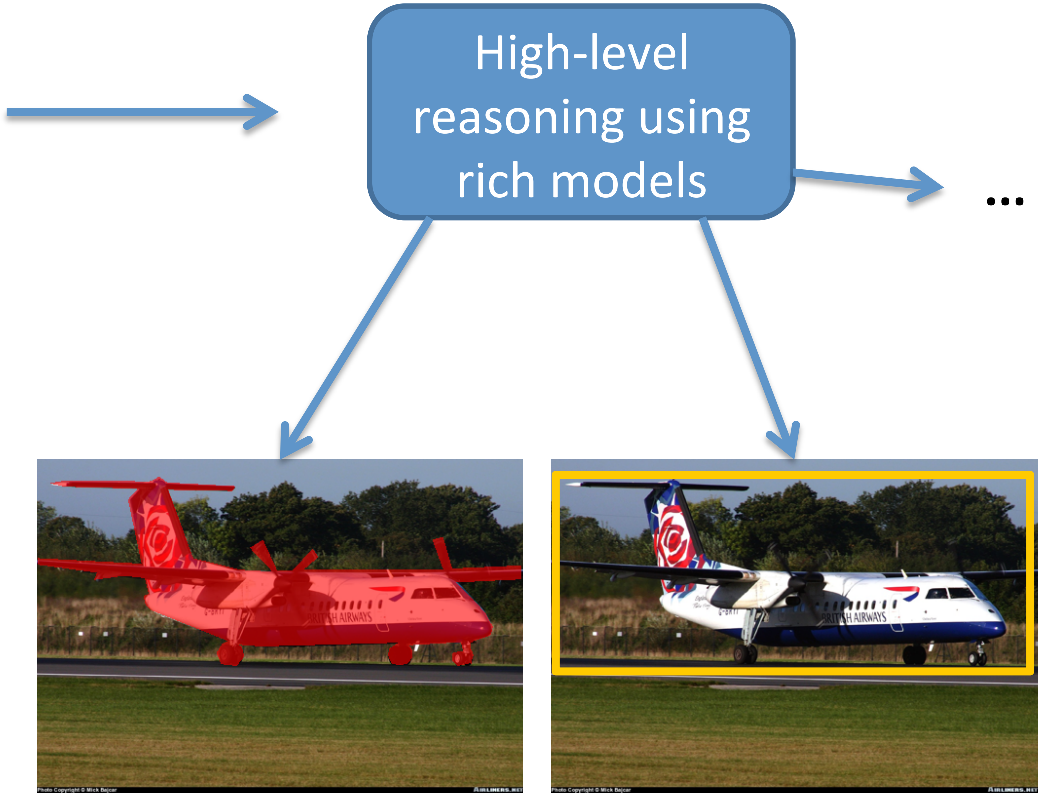
What Is Bottom-Up Image Parsing?

- **Image parsing**: decomposing an image into a set of meaningful structures (e.g. objects, parts, boundary-aligned segments)

- **Bottom-up parsing:** start with a set of primitives (e.g. super-pixels) and gradually merge them into larger structures

Motivation

fast image parsing into a multi-scale pool of segments



object segmentation

object detection

Our Approach

Greedy merging (agglomerative clustering):

- start with over-segmentation into super-pixels
- at each step, spatial neighbors with the highest score are merged

merging video

Related Work

Super-pixel grouping

- Classification Model for Segmentation [Ren, 2003]
- Optimal Contour Closure [Levinshtein, 2010]
- Efficient Region Search for Object Detection [Grauman, 2011]

Greedy merging

- gPb-owt-ucm [Arbelaez, 2010]
- Selective Search for Object Recognition [van de Sande, 2011]

Top-down merging

- Unifying Segmentation, Detection, and Recognition [Tu, 2003]

Scoring a Merge

Scoring model for segments (S_1, S_2) :

$$f\left(S_1, S_2\right) = g\left(S_1 \cup S_2\right) - \alpha d(S_1, S_2)$$

"objectness"
of segments union

distance
between segments

Complementary cues:

- distance is effective on uniform areas
- objectness captures appearance cues
  - how an object/part should look like
  - inter-segment variability can be high

Scoring Function Learning

$$f\left(S_1, S_2\right) = g\left(S_1 \cup S_2\right) - \alpha d(S_1, S_2)$$

Discriminative learning from ground-truth segmentation

Goal – learn a scoring model:

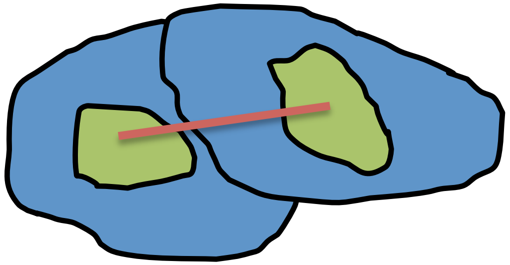
- pair inside an object – high score
- pair crossing the object – low score

Two research directions:

- Distance metric learning
- Objectness learning (next talk)

Distance Learning

Segment distance: $d_A(S_1, S_2) = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} d_A(S_1^{(i)}, S_2^{(j)})$



distance
between
super-pixels

Mahalanobis distance for super-pixels:

$$d_A(U, V) = (\phi_U - \phi_V)^T A(\phi_U - \phi_V)$$

Learn A from the constraints:

- $d_A(U_P, V_P) < \Delta_1$ if segments belong to the same class
- $d_A(U_N, V_N) > \Delta_2$ if segments belong to different classes

Distance Learning (2)

Convex max-margin objective:

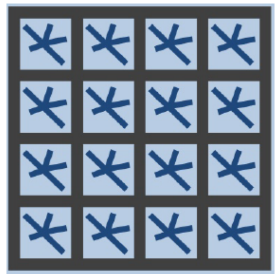
$$\min_{A \succeq 0} \sum_{(U_P, V_P)} \max(d_A(U_P, V_P) - \Delta_1, 0) + \sum_{(U_N, V_N)} \max(\Delta_2 - d_A(U_N, V_N), 0) + \frac{\lambda}{2} \|A\|_F^2$$

Solver: stochastic projected sub-gradient method

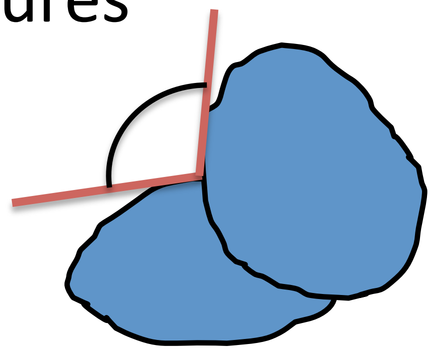
- projection on the cone of P.S.D. matrices by eigenvalue truncation
- step size $\gamma_t = 1/(\lambda t)$ due to strong convexity

Super-Pixels and Visual Features

- Super-pixels
  - Graph-based [Felzenszwalb, 2004]
  - SLIC [Achanta, 2012]

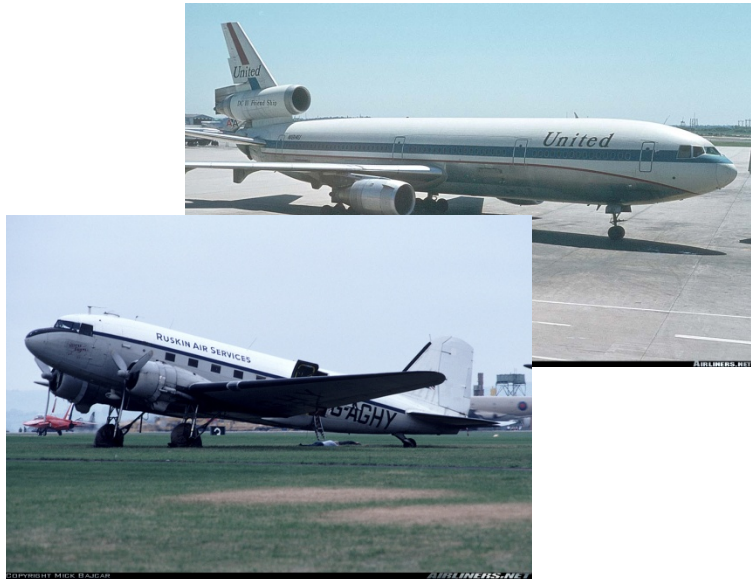


- Conventional features: bags of visual words
  - Dense multi-scale SIFT (500-D histogram)
  - Lab color (200-D histogram)
- Work in progress: boundary and shape features
  - Boundary strength, smoothness
  - Segment perimeter to area ratio



Datasets

- PASCAL VOC 2011
  - 20 classes, **single model**
  - training & validation - 1111 images
  - testing - 1112 images



- Airplanes
  - single class
  - training & validation - 2958 images
  - testing - 2979 images

Evaluation Measures

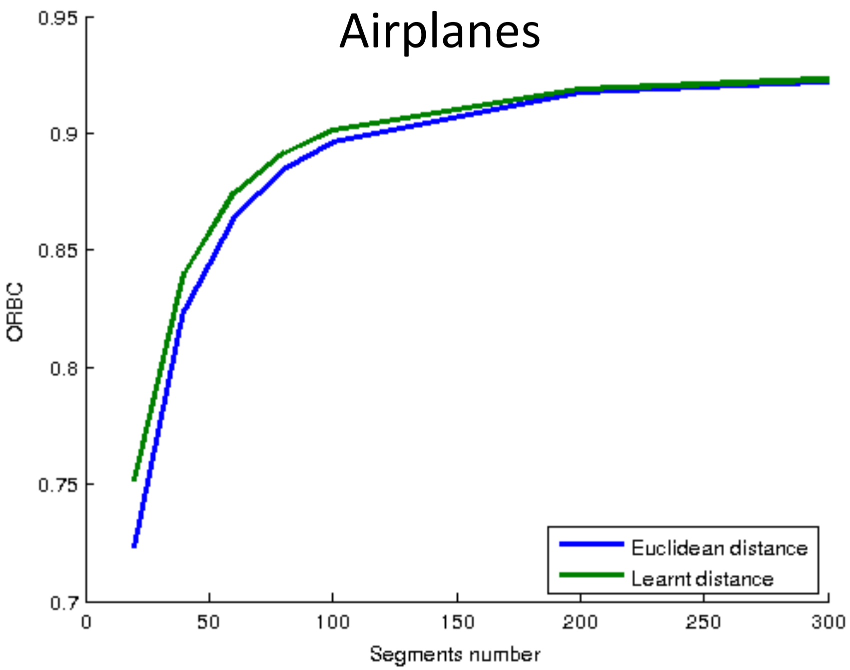
- Segmentation proposal recall
  - each segment is treated as a putative segmentation mask
  - ground-truth overlap ratio: $s = \frac{|GT \cap Prop|}{|GT \cup Prop|}$
  - recall – ratio of objects for which a good proposal $(s > 0.5)$ exists

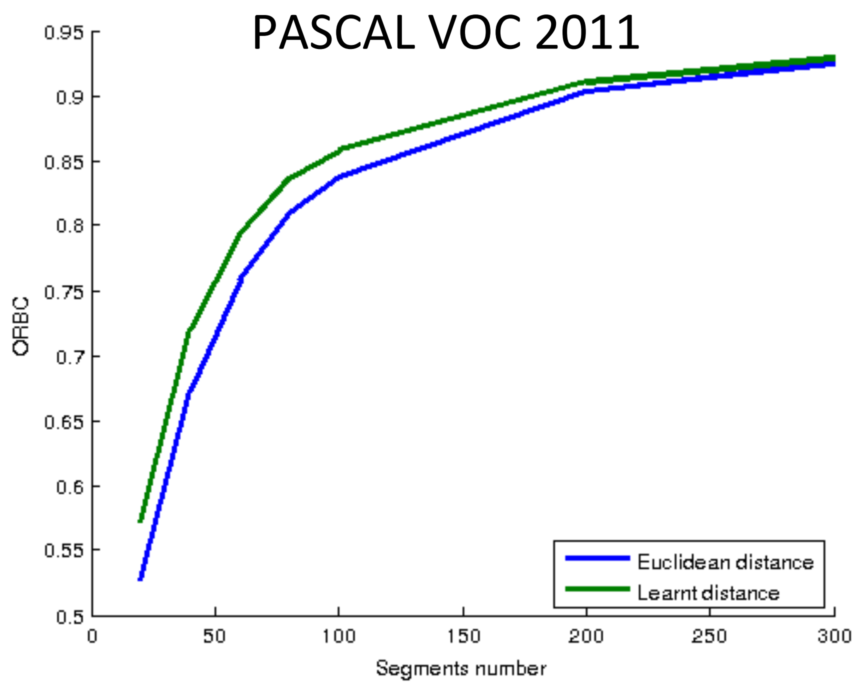
s=0.6

- Overlap Ratio Best Case (ORBC)
  - "best case" segmentation – union of segments with high ground-truth overlap
  - ORBC – overlap ratio of the "best case" segmentation
  - upper bound on segmentation accuracy

s=0.95

Results: Learnt vs Euclidean





ORBC

	Airplanes	PASCAL VOC 2011
Euclidean	0.638	**0.601**
Learnt	**0.673**	**0.601**

Proposal recall

Summary

- Fast bottom-up parsing – a pre-processing step for high-level vision algorithms (< 2 s/image)
- Two complementary merging cues
  - distance between segments
  - appearance of segment union
- Distance learning leads to slight improvement with off-the-shelf features
- Appearance learning – 2nd part of the talk...

Learning Appearance Models for Bottom-Up Parsing (LAMBUP)

David Weiss, Karen Simonyan,
Ben Taskar, Andrea Vedaldi

Re-cap: Greedy Merging

Re-cap: Greedy Merging

Objective:

s(i,j) = Objectness(Union(i,j)) - Distance(i,j)

Re-cap: Greedy Merging

Objective:

$$s(i,j) = -\,\text{Distance}(i,j)$$

Objectness Features

Objective:

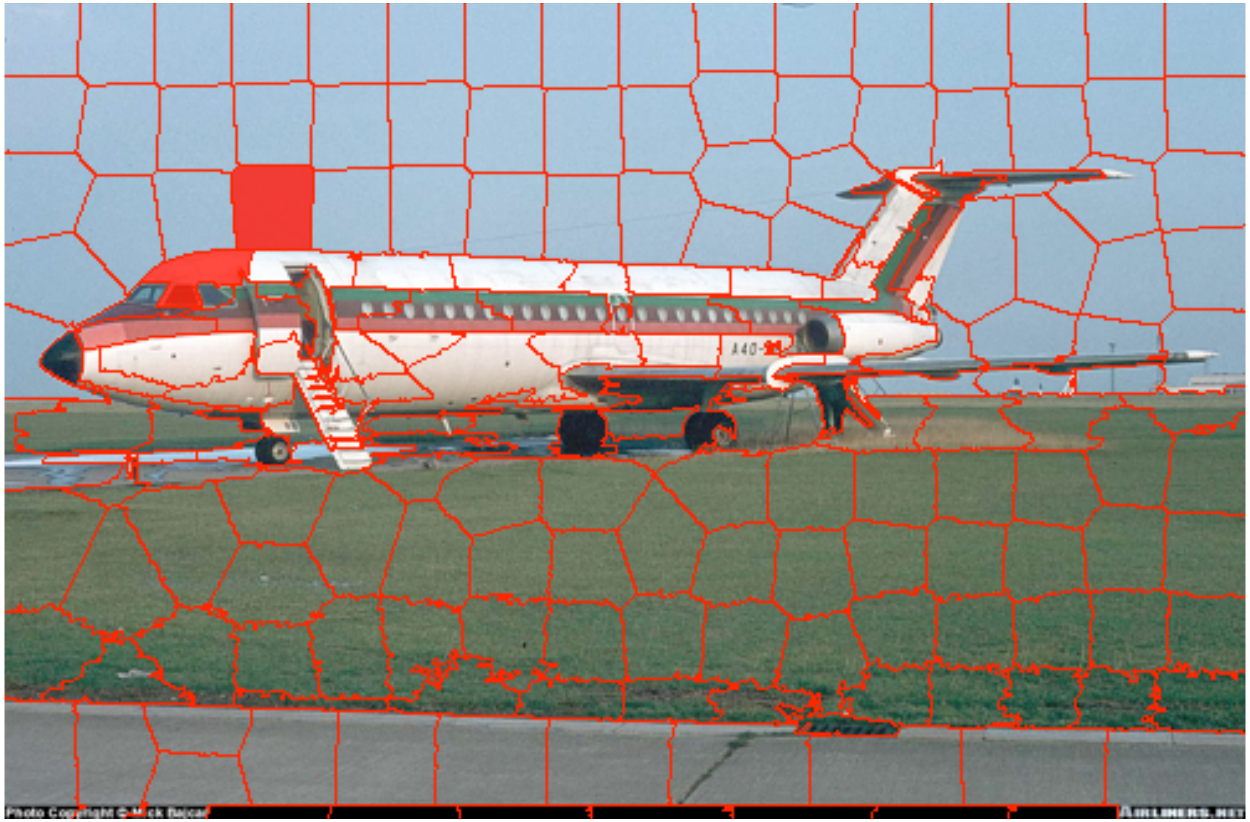
s(i,j) = Objectness(Union(i,j))

$$s(i,j) = \mathbf{w}^\top \mathbf{f}(x_i, x_j)$$

f = [color, texture , - Distance(i,j)]

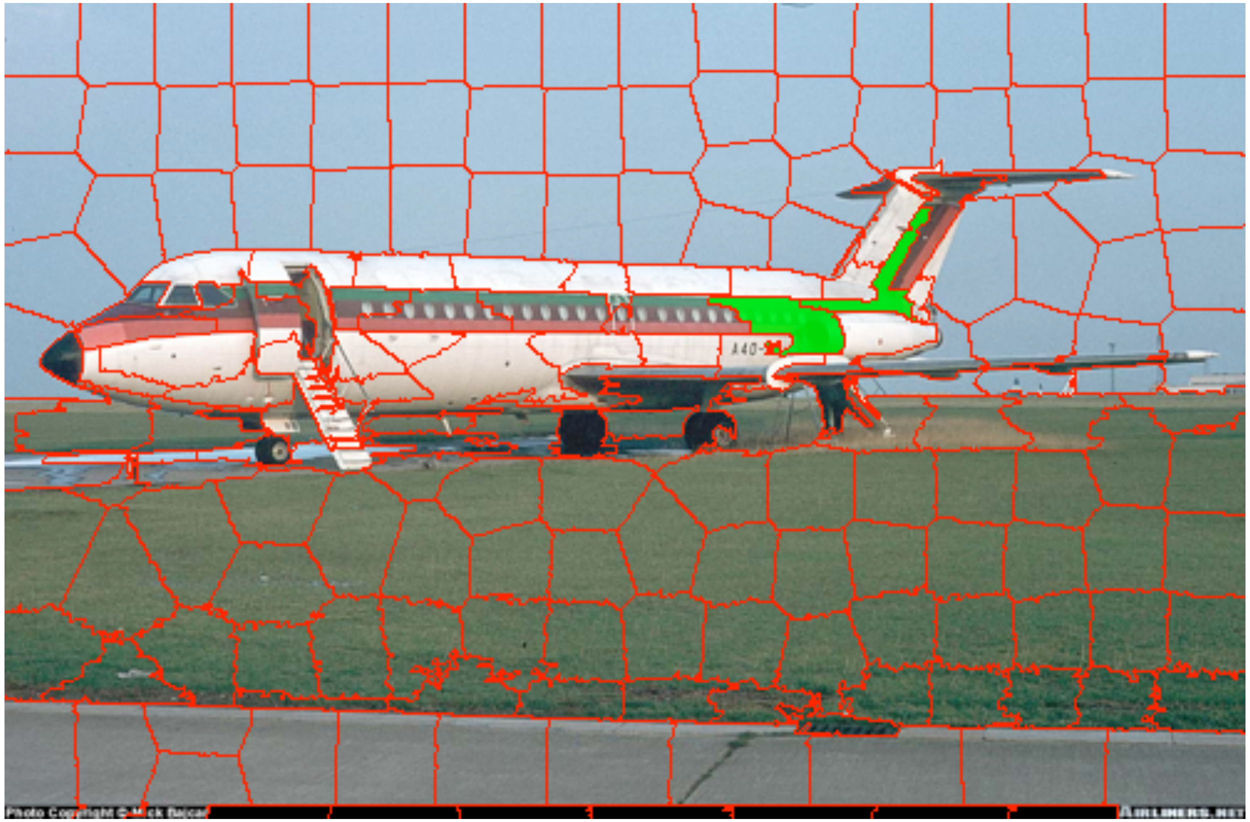
Learning the Weights

Bad merges



Learning the Weights

Good merges



Learning the Weights

N: Bad merges P: Good merges x: Image

$$\min \frac{1}{2}||\mathbf{w}||^2$$

$$\mathbf{w}^\top \mathbf{f}(x_u, x_v) \leq -1 + \xi^x_{uv}, \quad \forall (u,v) \in N^x$$

$$\mathbf{w}^\top \mathbf{f}(x_i, x_j) \geq 1 - \xi^x_{ij}, \quad \forall (i,j) \in P^x$$

<u>"Standard SVM" Formulation</u>

Learning the Weights

- In practice, difficult to score *all* positives above threshold
- Not all pairs need to be merged: Labels are **ambiguous**
- Can incorporate into learning for more robust procedure

Learning the Weights

N: Bad merges P: Good merges x: Image

$$\min \frac{1}{2}||\mathbf{w}||^2 + C\sum_{x} \sum_{(u,v)\in N^x} \xi_{uv}^x \quad + \quad \xi_P^x$$

$$\mathbf{w}^\top \mathbf{f}(x_u, x_v) \leq -1 + \xi_{uv}^x, \quad \forall (u,v) \in N^x$$

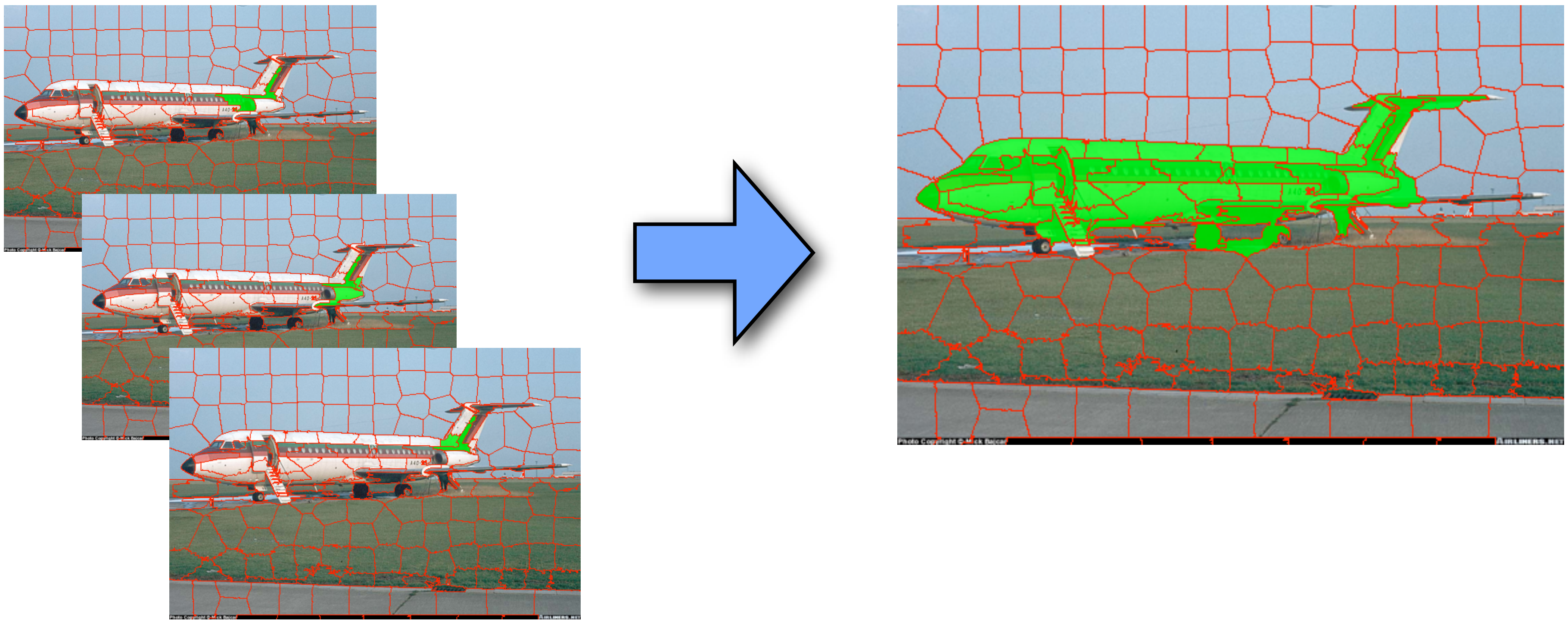
$$\mathbf{w}^\top \mathbf{f}(x_i, x_j) \geq 1 - \xi_{ij}^x, \quad \forall (i,j) \in P^x$$

$$\frac{1}{|P^x|} \sum_{(i,j)\in P^x} \mathbf{w}^\top \mathbf{f}(x_i, x_j) \geq 1 - \xi_P^x, \quad \forall x$$

<u>"Ambiguous Labels" Formulation</u>

Learning the Weights

"Ambiguous Labels" Formulation



Evaluation

- Output destined for object detector
- Propose object segmentations
- One merge = One Proposal

Evaluation Proposals

- One merge = One Proposal

Evaluation Proposals

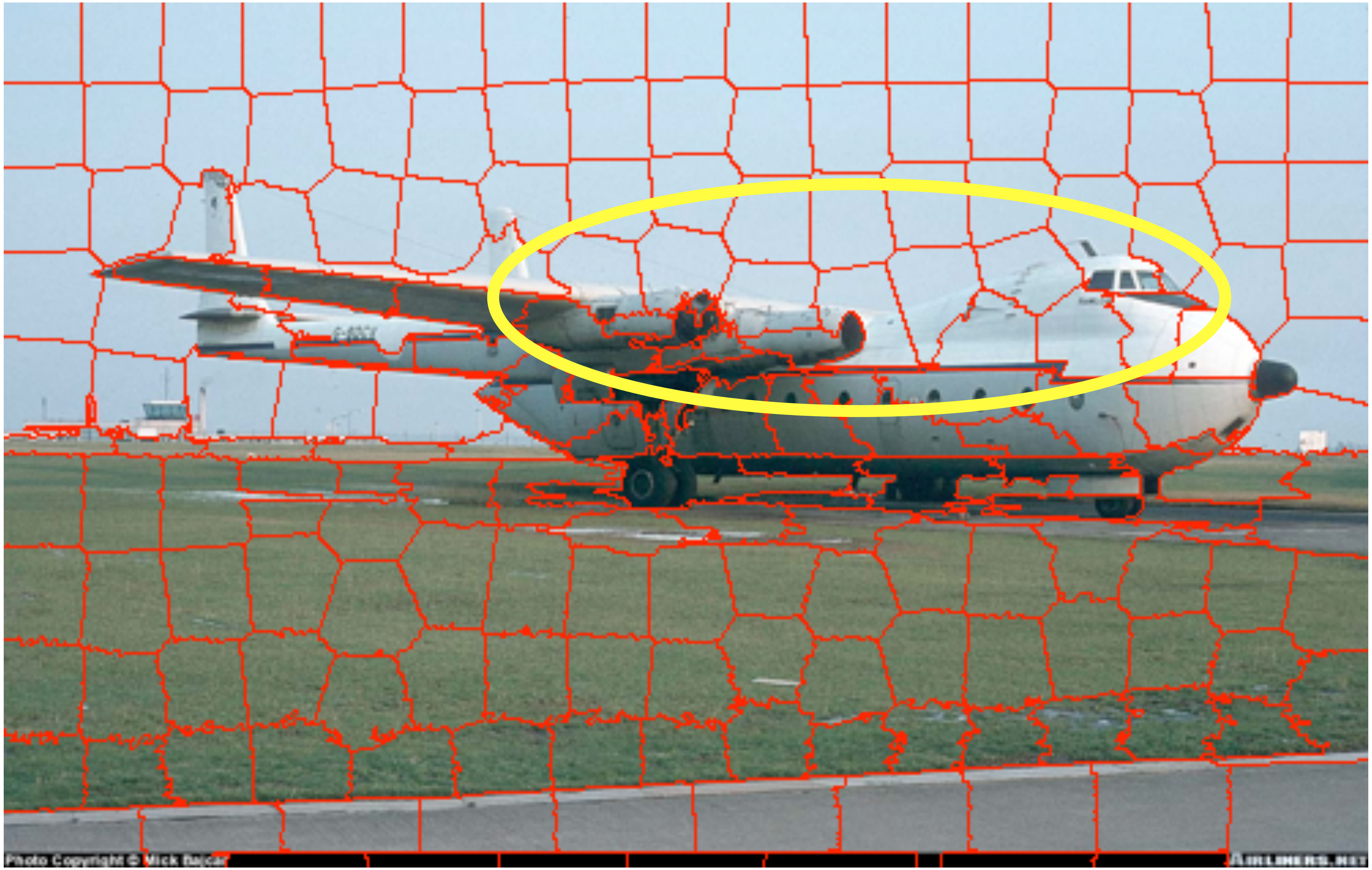
- Compute Intersection over Union (IoU)
- IoU >= 0.5 = “hit”
- Measure recall

0.7399

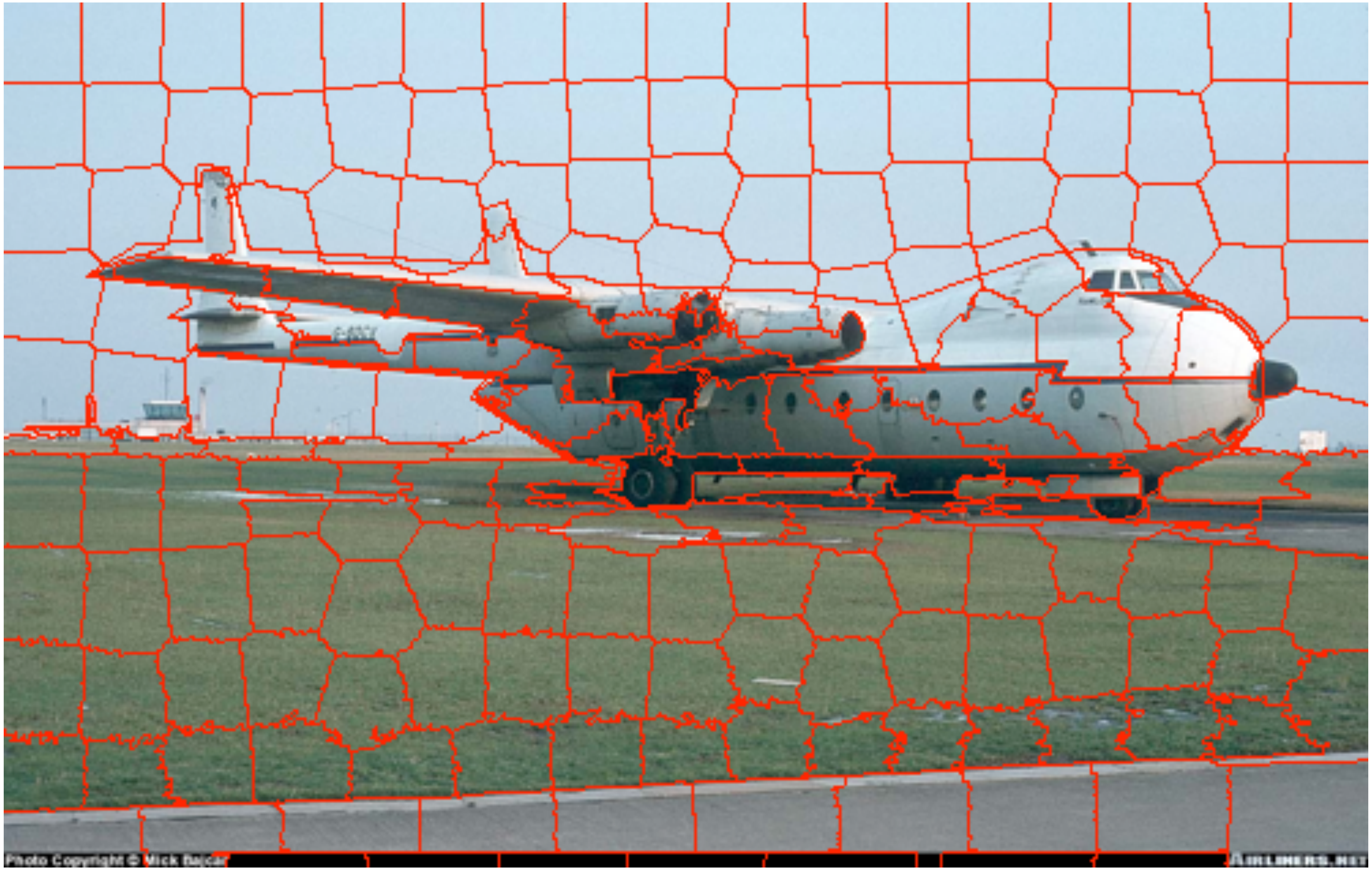
Evaluation Proposals

Method	Recall
Distance Only	67.0
Standard SVM	71.5
Ambiguous Labels	**72.9**

Improving Training Data



Improving Training Data



Improving Training Data

Method	Recall	Recall (Improved)
Distance Only	67.0	--
Standard SVM	71.5	**75.9**
Ambiguous Labels	**72.9**	75.7

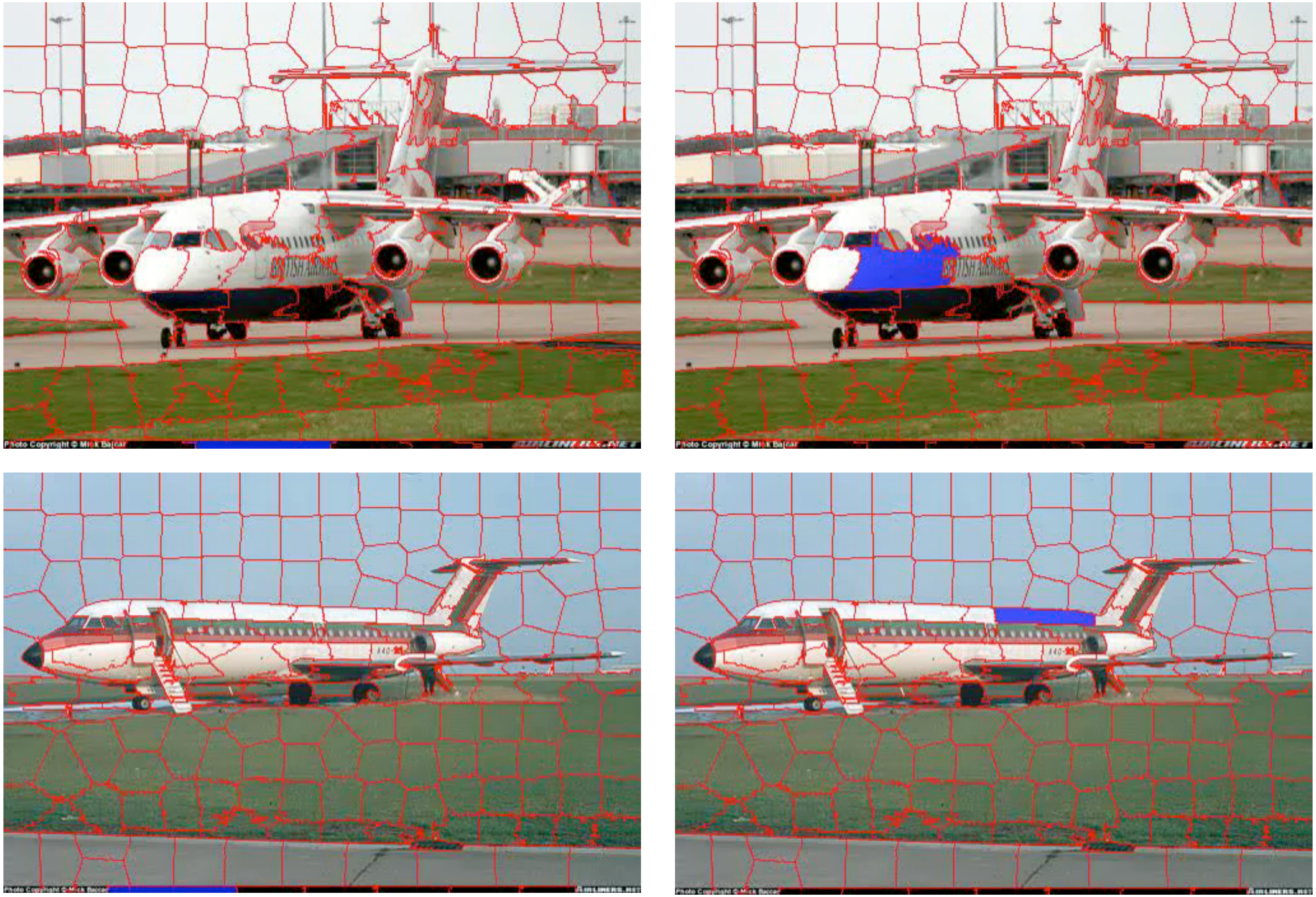
Fixing data --> easier to learn

Work-In-Progress

- Merging = **Changing Feature Distribution**
- Model should **adapt**
- Solution: **novel cascade architure**

<u>Implemented, but not enough features</u>

Objectness Helps!



	S	P	FS	FP	Base
aeroplane	0.42	0.44	**0.50**	0.40	**0.53**
bicycle	**0.07**	**0.07**	0.06	0.06	**0.08**
bird	0.67	0.63	**0.71**	0.63	**0.67**
boat	0.26	**0.27**	**0.27**	**0.27**	**0.40**
bottle	**0.39**	0.30	0.35	0.33	**0.39**
bus	**0.41**	0.25	**0.41**	0.25	**0.46**
car	0.34	**0.35**	**0.35**	0.33	**0.38**
cat	0.80	0.76	**0.84**	0.78	**0.85**
chair	**0.38**	0.36	0.36	0.37	**0.43**
cow	**0.66**	0.63	0.62	**0.66**	0.62
diningtable	0.44	0.46	0.49	0.49	**0.54**
dog	0.65	**0.66**	0.65	0.65	0.57
horse	**0.65**	**0.65**	0.62	**0.65**	0.58
motorbike	**0.49**	0.47	0.46	0.47	**0.61**
person	0.37	**0.40**	0.39	0.39	0.38
pottedplant	0.39	0.38	0.39	**0.40**	**0.40**
sheep	0.52	**0.53**	0.52	0.52	0.46
sofa	0.69	0.66	**0.75**	0.69	**0.75**
train	0.39	0.39	**0.42**	0.42	**0.63**
tvmonitor	0.59	0.59	0.59	**0.60**	**0.68**

Improving Training Data

Method	Recall	Recall (Improved)
Distance Only	52.0	--
Standard SVM	47.8	48.7
Ambiguous Labels	46.3	46.7

LabSIFTPairwise

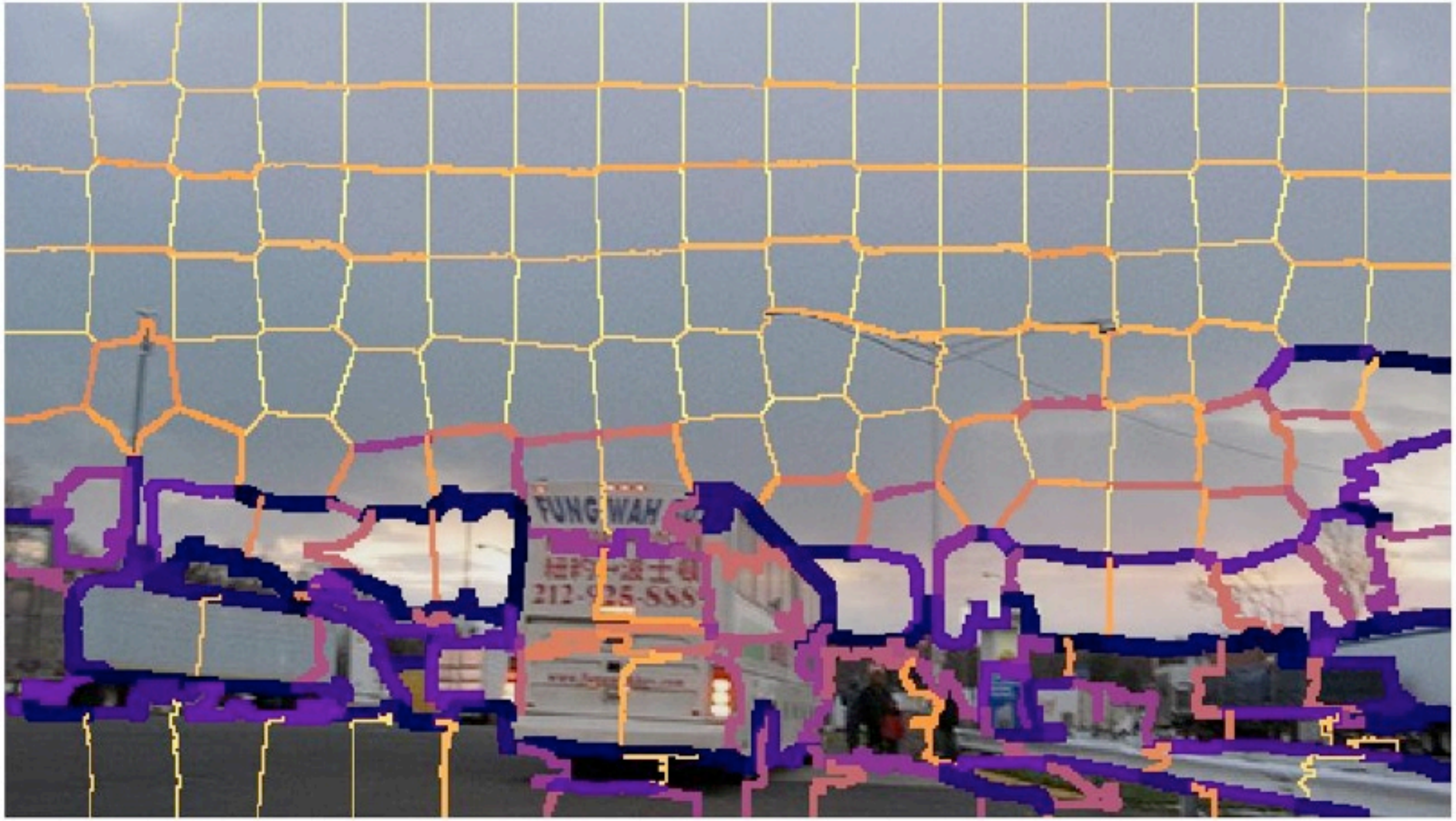
LabSIFTPairwise

LabSIFTPairwise+LabSIFTUnary

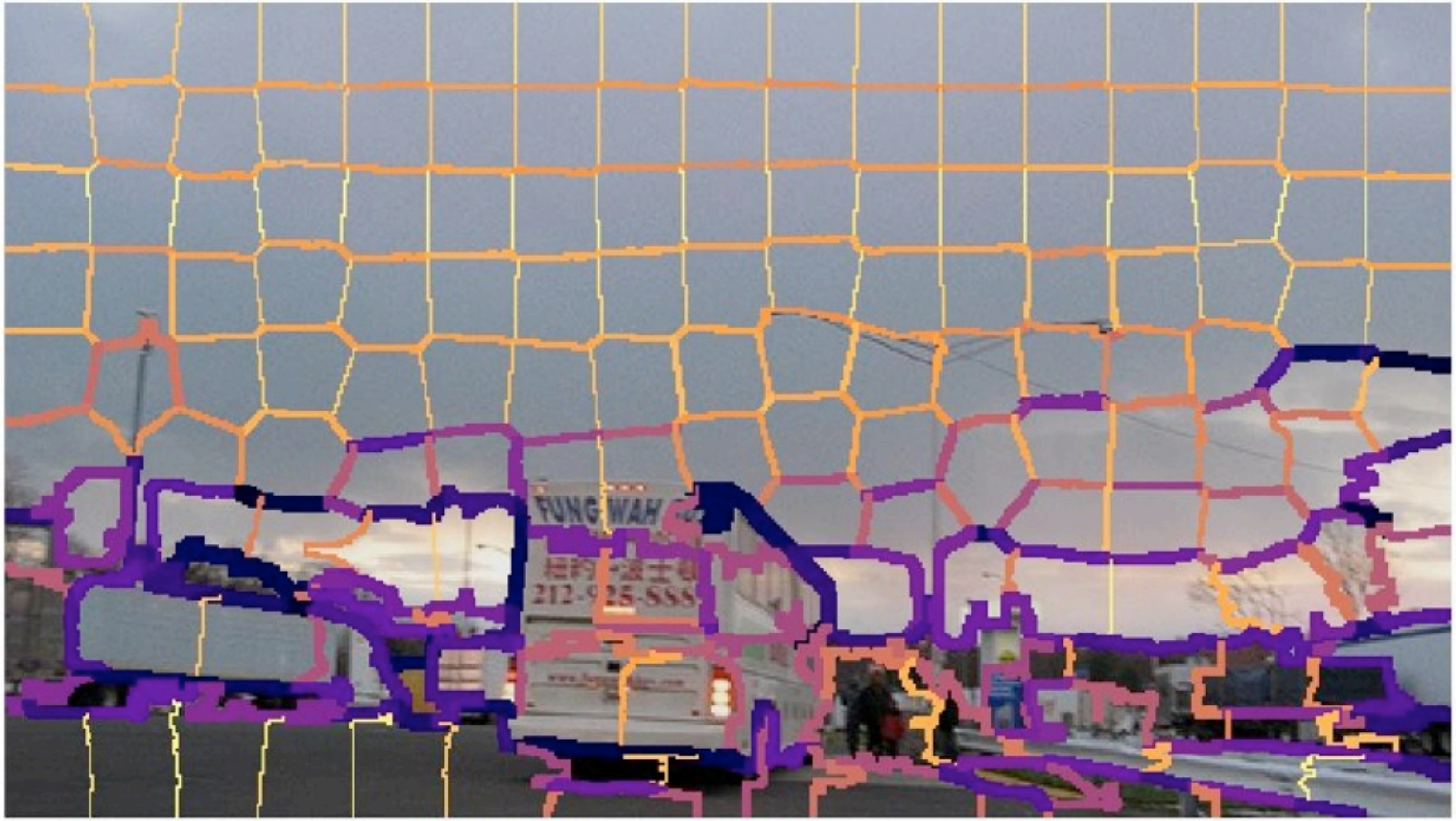
LabSIFTPairwise

LabSIFTPairwise+LabSIFTUnary

LabSIFTPairwise



LabSIFTPairwise+LabSIFTUnary



Overview

Objects in Detail
Parts & attributes

- A new dataset
- An object lexicon
- Localising parts
- Layouts
- Recognising attributes

Stuff in Detail
Texture

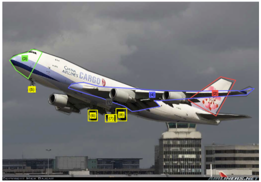
- A texture lexicon
- A new dataset
- Transformation invariant semantic

Parsing
Bottom-up inference

- Learning to merge
- Cascading
- Scoring regions by attributes

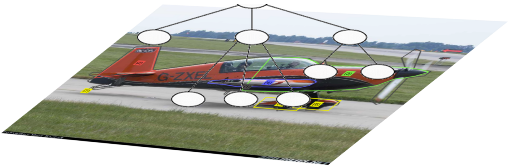
Part/Attribute Queries

- A person may be interested in querying a set of images for objects that have certain properties
  - An aeroplane with a red, pointy nose

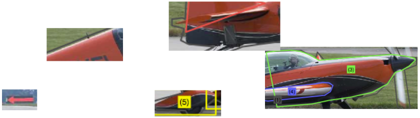


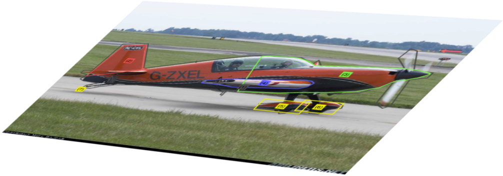
  - A furry cat

Bottom Up Proposals of Parts/Attributes



single-prop aeroplane with a red pointy nose





Scoring Functions

- First approach: train a discriminative classifier for every possible class/part/attribute

I	$f_{\text{cat}}(I)$	$f_{\text{furry}}(I)$	$f_{\text{furry+cat}}(I)$	$\dots$
	17.3	16.8	19.2	$\dots$
	14.6	-3.2	-0.6	$\dots$

A Naïve Independence Assumption

- k mutually-exclusive class/parts, m binary attributes $\rightarrow$ $(k+1)2^m - 1$ possible scoring functions

- Insufficient sample of complex part/attribute combinations
- Exponential training cost

$$\begin{aligned} p(brown, furry, cat) &\propto e^{f_{\mathsf{brown}}(I)} \cdot e^{f_{\mathsf{furry}}(I)} \cdot e^{f_{\mathsf{cat}}(I)} \\ &\Longrightarrow \\ \ln p(brown, furry, cat) &= f_{\mathsf{brown}}(I) + f_{\mathsf{furry}}(I) + f_{\mathsf{cat}}(I) + b \end{aligned}$$

- Linear training cost
- Disregards the high statistical dependence between cat and furry

Joint Discriminative Training

- Formulation as regularized risk

$$\min_f \lambda\Omega(f) + \sum_{q\in\mathcal{Q}} \ell(f, X, Y, q)$$

- $|\mathcal{Q}|$ is exponential, and we therefore need to sample a subset of *basis queries*, Q

$$\min_{f_Q} \lambda\Omega(f_Q) + \sum_{q\in Q} \ell(f_Q, X, Y, q)$$

- Q is a very general parametrization of discriminative models

Basis Queries

- For simplicity, consider only conjunctions: brown $\wedge$ furry $\wedge$ cat
- Encode as a binary matrix

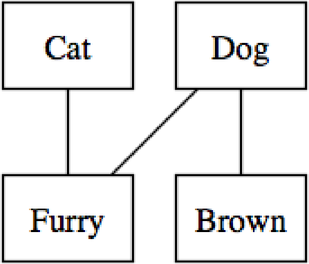
	cat	dog	brown	furry
q_1	1	0	0	0
q_2	0	1	0	0
q_3	0	0	1	0
q_4	0	0	0	1
q_5	1	0	0	1
q_6	0	1	1	0
q_7	0	1	0	1

Relationship to Graphical Models

- Hammersley-Clifford theorem

$$\ln p(x) = \sum_{C \in \mathrm{cl}(\mathcal{G})} f_C(x_C) + b$$

	cat	dog	brown	furry
q_1	1	0	0	0
q_2	0	1	0	0
q_3	0	0	1	0
q_4	0	0	0	1
q_5	1	0	0	1
q_6	0	1	1	0
q_7	0	1	0	1



Vector Valued Functions / Query Covariances

- A vector valued function returns a vector ouput for any input.
- One may specify a covariance structure, B, between outputs.
- With a separable kernel, $k(x, y, i, x', y', j) = k(x, y, x', y')B_{i,j}$ and $K_S = K_{\text{joint}} \otimes B$

- $B_{i,j}$ should be large if outputs i and j are similar, and small otherwise.
- We will set each of our outputs to be the scoring function of a prediction for a given part/attribute query, and B will measure how similar those scoring functions should be.

Application to Part/Attribute Queries

- A part/attribute query can be encoded in a binary string as follows:

nose	...	wing	striped	red	pointy	...
1	...	0	0	1	1	...

  we will call the mapping of a query, q, to this binary string $\varphi(q)$

- Set $B_{i,j} = \langle \varphi(q_i), \varphi(q_j) \rangle$
- We specify a set of basis queries, $Q = \{q_1, \ldots, q_k\}$.
- Train vector valued regression with the submatrix B_Q corresponding to the basis queries
- Infer functions for novel queries using their relationship to basis queries

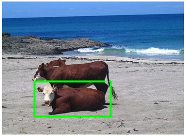
Joint Kernel between Images and Boxes: Restriction Kernel

- Note: $x|_y$ (the image restricted to the box region) is again an image.
- Compare two images with boxes by comparing the images within the boxes:

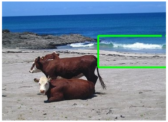
$$k_{joint}((x,y),(x',y')\ ) = k_{image}(x|_y, x'|_{y'},)$$

- Any common image kernel is applicable:
  - linear on cluster histograms: $k(h,h') = \sum_i h_i h'_i$,
  - χ^2-kernel: $k_{\chi^2}(h,h') = \exp\left(-\frac{1}{\gamma}\sum_i \frac{(h_i-h'_i)^2}{h_i+h'_i}\right)$
  - pyramid matching kernel, ...
- The resulting joint kernel is positive definite.

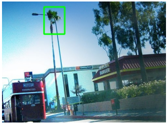
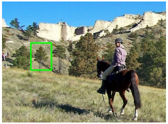
Restriction Kernel: Examples

$k_{joint}\Big($  ,  $\Big) = k\Big($  ,  $\Big)$

is large.

$k_{joint}\Big($  ,  $\Big) = k\Big($  ,  $\Big)$

is small.

$k_{joint}\Big($  ,  $\Big) = k\Big($  ,  $\Big)$

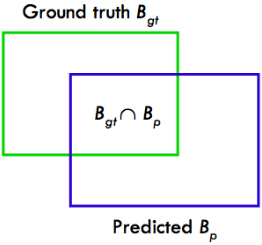
could also be large.

- Note: This behaves differently from the common tensor products

$$k_{joint}(\ (x,y),(x',y')\ ) \neq k(x,x')k(y,y'))\ !$$

Evaluating Bounding Boxes

- Area of Overlap (AO) Measure



$$AO(B_{gt}, B_p) = \frac{|B_{gt} \bigcap B_p|}{|B_{gt} \bigcup B_p|}$$

- Set a threshold such that $AO(B_{gt}, B_p) > t$ indicates a correct detection: 0.5

PASCAL VOC

- Define a loss function $\Delta(B_{gt}, B_p) = 1 - AO(B_{gt}, B_p)$.

Structured Output Ranking

- Given a joint kernel map, φ, learn an objective that orders outputs correctly

$$\min_{w\in\mathcal{H},\xi} \qquad \lambda\Omega(w) + \frac{1}{|\mathcal{E}|}\sum_{(i,j)\in\mathcal{E}} \xi_{ij} \tag{1}$$

$$\begin{aligned}
\text{s.t.} \qquad & \langle w, \varphi(x_i, y_i)\rangle - \langle w, \varphi(x_j, y_j)\rangle \geq \overbrace{\Delta_j - \Delta_i - \xi_{ij}}^{\text{margin rescaling}} \\
\text{or} \qquad & \langle w, \varphi(x_i, y_i)\rangle - \langle w, \varphi(x_j, y_j)\rangle \geq \underbrace{1 - \frac{\xi_{ij}}{\Delta_j - \Delta_i}}_{\text{slack rescaling}}
\end{aligned}$$

$$\xi_{ij} \geq 0 \tag{2}$$

Transferring to Previously Unseen Queries

- Given basis queries, we may jointly learn a set of functions by combining ranking objectives subject to a joint regularization of basis queries: $\Omega(f_1, \ldots, f_k) = \alpha^T K \otimes B\alpha$

- Using our covariance function, we may construct a ranking objective for previously unseen queries by taking a linear combination of basis queries:

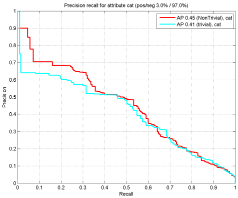
$$f_j = \sum_{i \in \text{basis}} B_{i,j} f_i$$

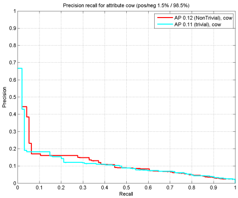
Results

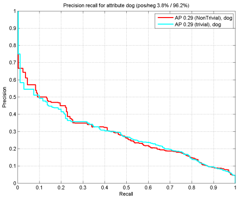
- VOC Dataset - 20 categories
- Features and attributes described in Farhadi et al., CVPR 2009
- Texture + Color + HOG $\approx$ 9K features
- 64 attributes - many of which are *highly* correlated with a specific class label

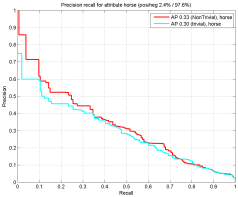
- We will focus on the "furry" attribute and related classes

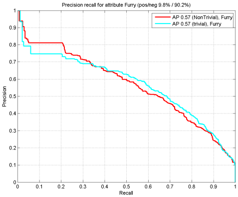
Results

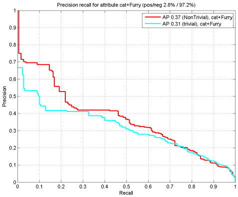

Precision recall for attribute cat (pos/neg 3.0% / 97.0%)
AP 0.45 (NonTrivial), cat
AP 0.41 (trivial), cat
Precision
Recall

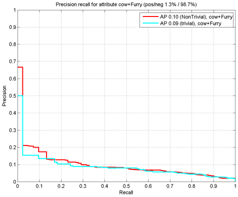

Precision recall for attribute cow (pos/neg 1.5% / 98.5%)
AP 0.12 (NonTrivial), cow
AP 0.11 (trivial), cow
Precision
Recall

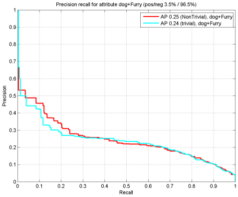

Precision recall for attribute dog (pos/neg 3.8% / 96.2%)
AP 0.29 (NonTrivial), dog
AP 0.29 (trivial), dog
Precision
Recall

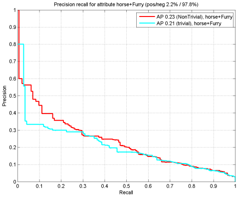

Precision recall for attribute horse (pos/neg 2.4% / 97.6%)
AP 0.33 (NonTrivial), horse
AP 0.30 (trivial), horse
Precision
Recall


Precision recall for attribute Furry (pos/neg 9.8% / 90.2%)
AP 0.57 (NonTrivial), Furry
AP 0.57 (trivial), Furry
Precision
Recall

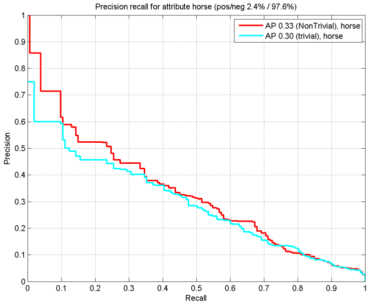

Precision recall for attribute cat+Furry (pos/neg 2.8% / 97.2%)
AP 0.37 (NonTrivial), cat+Furry
AP 0.31 (trivial), cat+Furry
Precision
Recall


Precision recall for attribute cow+Furry (pos/neg 1.3% / 98.7%)
AP 0.10 (NonTrivial), cow+Furry
AP 0.09 (trivial), cow+Furry
Precision
Recall

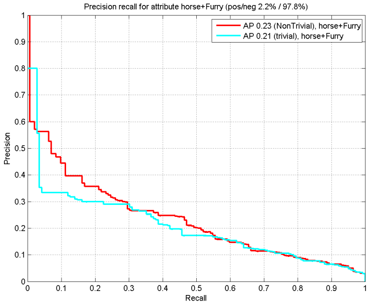

Precision recall for attribute dog+Furry (pos/neg 3.5% / 96.5%)
AP 0.25 (NonTrivial), dog+Furry
AP 0.24 (trivial), dog+Furry
Precision
Recall


Precision recall for attribute horse+Furry (pos/neg 2.2% / 97.8%)
AP 0.23 (NonTrivial), horse+Furry
AP 0.21 (trivial), horse+Furry
Precision
Recall

Results

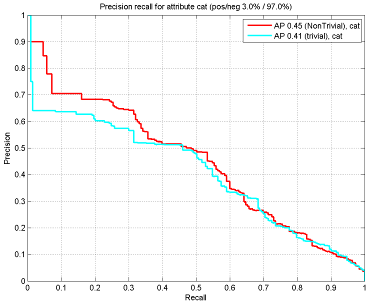


Precision recall for attribute horse (pos/neg 2.4% / 97.6%)
AP 0.33 (NonTrivial), horse
AP 0.30 (trivial), horse
Precision
Recall

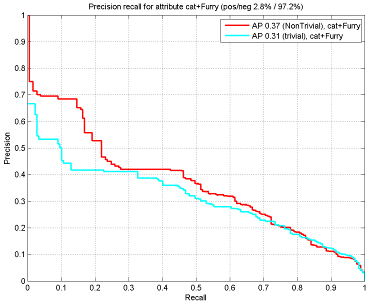


Precision recall for attribute horse+Furry (pos/neg 2.2% / 97.8%)
AP 0.23 (NonTrivial), horse+Furry
AP 0.21 (trivial), horse+Furry
Precision
Recall

Results



Precision recall for attribute cat (pos/neg 3.0% / 97.0%)
AP 0.45 (NonTrivial), cat
AP 0.41 (trivial), cat
Precision
Recall



Precision recall for attribute cat+Furry (pos/neg 2.8% / 97.2%)
AP 0.37 (NonTrivial), cat+Furry
AP 0.31 (trivial), cat+Furry
Precision
Recall

Overview and Future Outlook

- Discriminative training of a scoring system for object/part+attributes queries
- A general regularized risk framework that relates *basis queries* to a graphical model structure
- Natural extension to novel queries at test time
- Significantly improved performance over a naïve independence assumption

- Extensions to queries beyond conjunctions
- Automatic learning of basis query set (structure of graphical model)
  - Modeling accuracy + sparsity penalty
- Integration with top down inference system

Contact

Matthew Blaschko
Center for Visual Computing
École Centrale Paris & INRIA Saclay - Île-de-France

matthew.blaschko@inria.fr

Summary

Objects in Detail

Parts & attributes

- A new dataset
- An object lexicon
- Localising parts
- Layouts
- Recognising attributes
- The cost of data collection

Stuff in Detail

Texture

- A texture lexicon
- A new dataset
- Transformation invariant semantic

Parsing

Bottom-up inference

- Learning to merge
- Cascading
- Scoring regions by attributes

Contribution: A framework for annotation

Annotation software

Draw polygons, mark attributes, display instructions ...

JS magic

Submission software

Manage money, revisions, and data

Submitted ~200,000 Amazon Turk HITs

Validation software

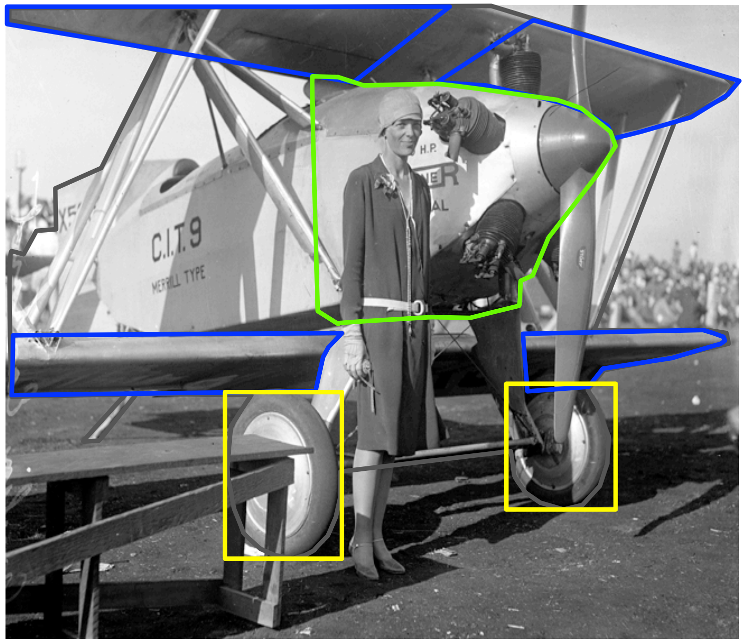
Coordinate people, fix errors

Validate more than 30,000 part annotations in a few days

A special thanks to Esa and Juho!

Contribution: A new part & attribute dataset

Problem	Data	Time frame	Progress
Image Classification	Caltech-101	2003-06	star models, BoW
Object Detection	PASCAL VOC	2006-12	DPMs, large scale learning
Parts & Attributes	**OID**	2012-?	?



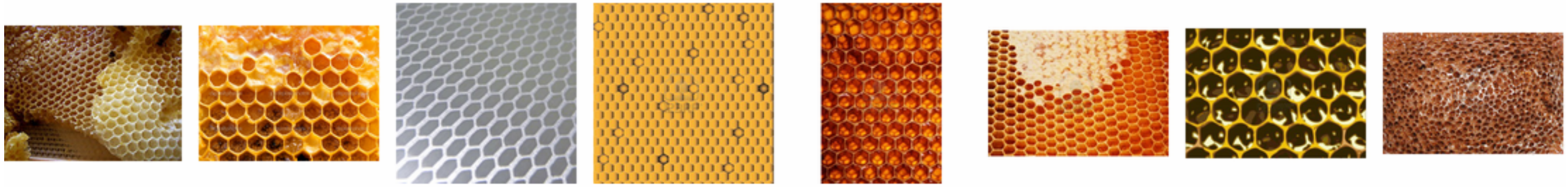
First dataset in this class

New benchmark and challenges

See it grow in the future!

Contribution: a new semantic texture dataset

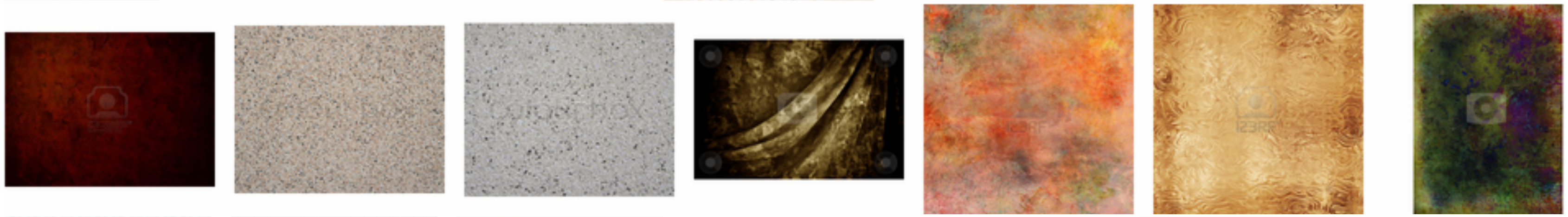
honeycombed



latticed

netlike

mottled



meshed

Contribution: models & methods

Parts and geometry

Part models, semantic
clustering boxes & shapes

Part layouts
improving part detection with context

Learning to merge

Generic
metric learning

Class specific
union & ambiguous labels

Attributes

Attributes from appearance
local-global appearance and attribute interactions

Attributes from geometry
many attributes can be predicted from layouts

Proposals
covariant attribute modelling

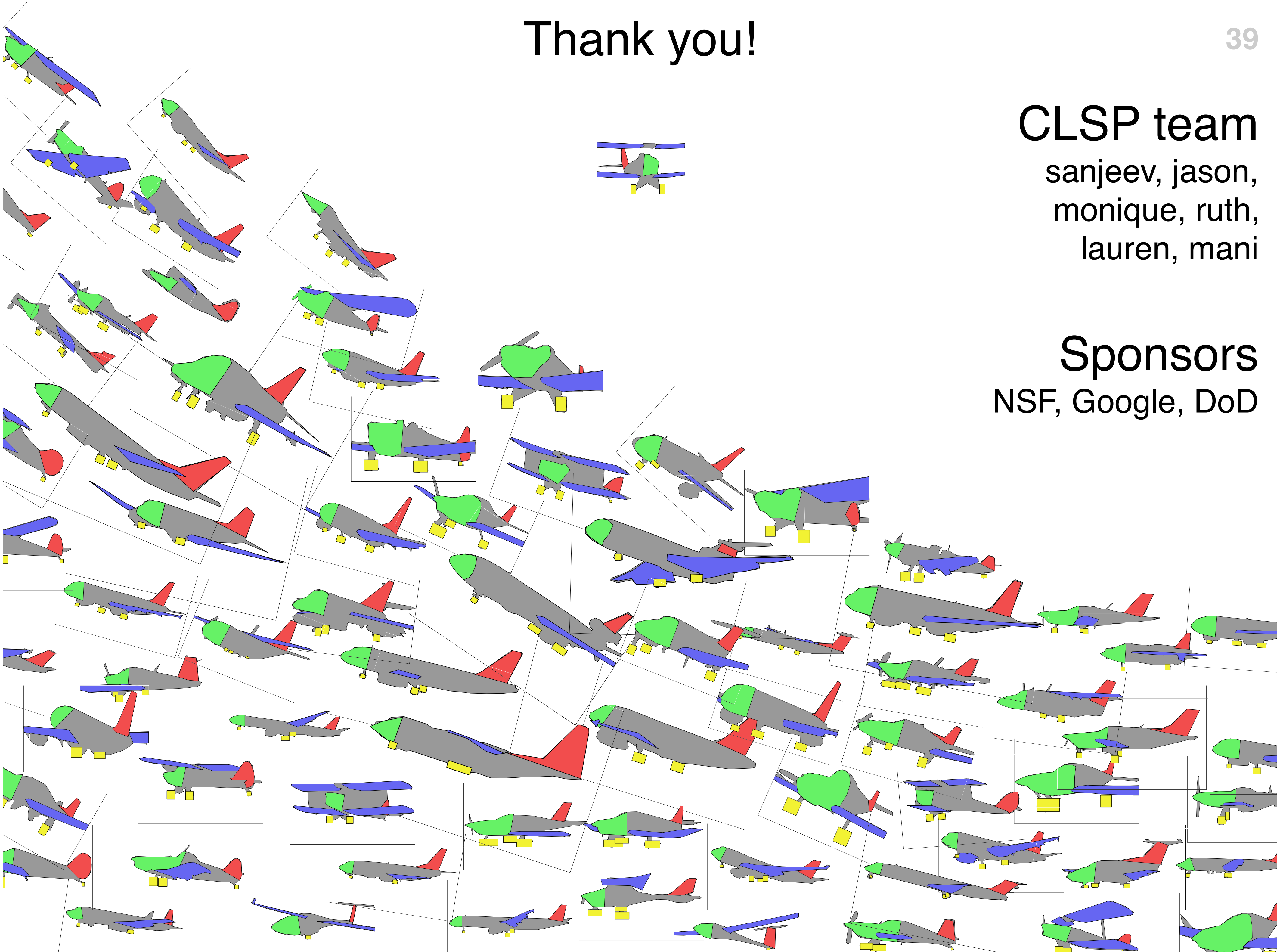
Texture
nuisance-invariant models

Future

- **The start of a new challenge**
  - the life after 7 years of PASCAL VOC
    - large scale but basic understanding (*e.g.*, ImageNet)
    - detailed understanding
  - **Objects in detail**
    - a multi-year challenge
  - **Texture in detail**

- **Pushing the technical barrier**
  - modelling local & global information
  - fast inference
  - detailed features for subtle attributes

Thank you!



CLSP team

sanjeev, jason,
monique, ruth,
lauren, mani

Sponsors

NSF, Google, DoD