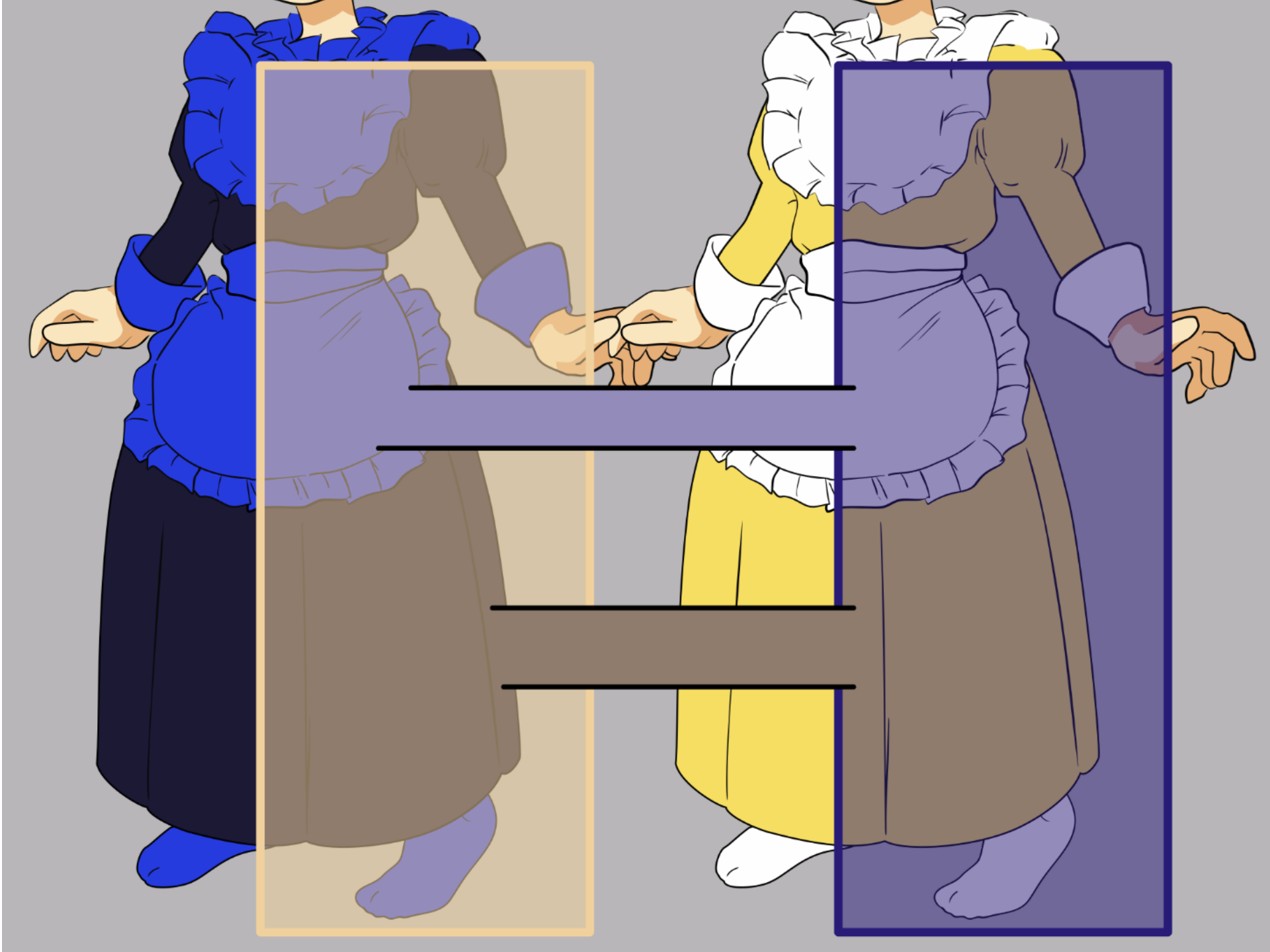
https://en.wikipedia.org/wiki/The_dress



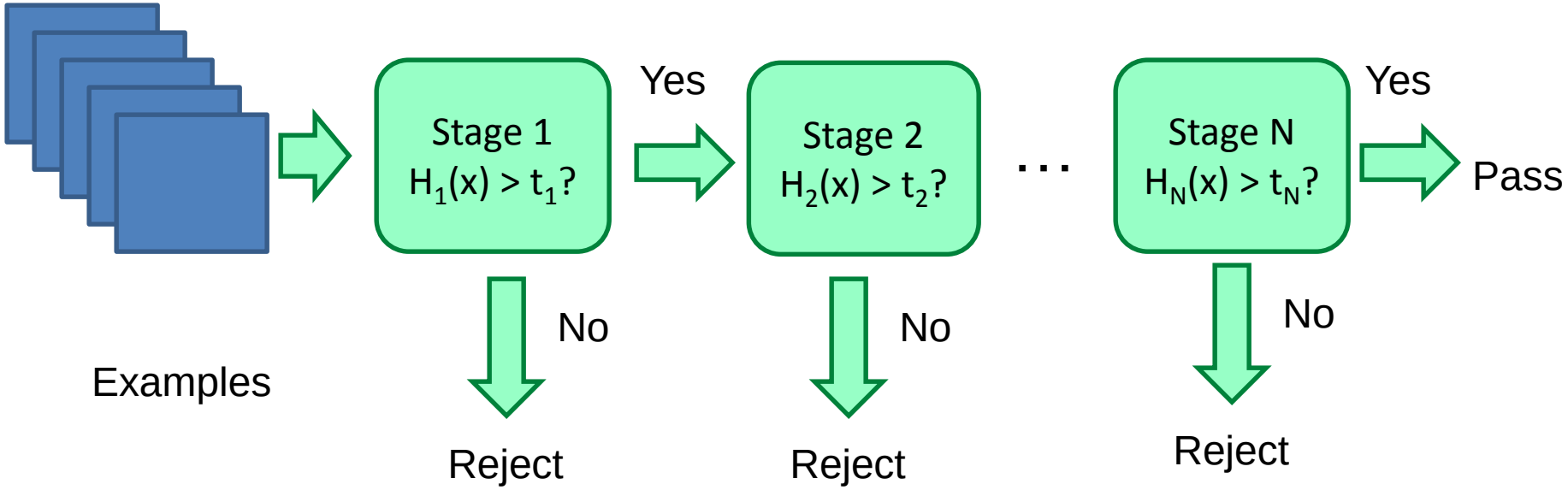
Recap: Viola-Jones sliding window detector

Fast detection through two mechanisms

- Quickly eliminate unlikely windows
- Use features that are fast to compute

Viola and Jones. Rapid Object Detection using a Boosted Cascade of Simple Features (2001).

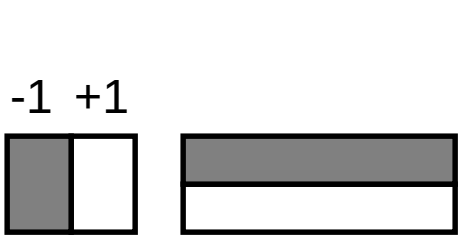
Cascade for Fast Detection



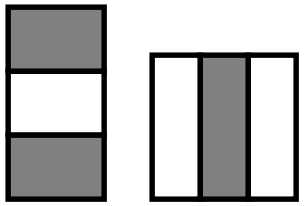
- Choose threshold for low false negative rate
- Fast classifiers early in cascade
- Slow classifiers later, but most examples don't get there

Features that are fast to compute

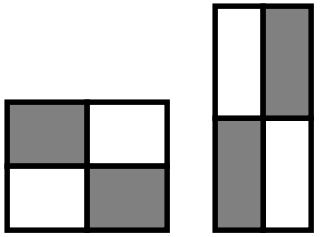
- "Haar-like features"
  - Differences of sums of intensity
  - Thousands, computed at various positions and scales within detection window



Two-rectangle features



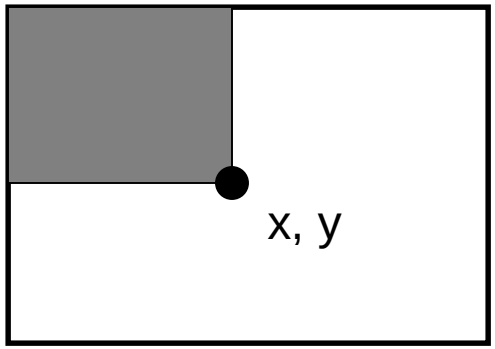
Three-rectangle features



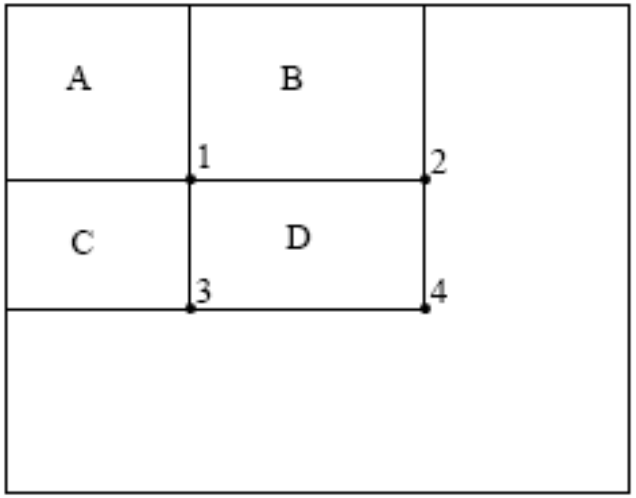
Etc.

Integral Images

- `ii = cumsum(cumsum(im, 1), 2)`



ii(x,y) = Sum of the values in the grey region



SUM within Rectangle D is
ii(4) - ii(2) - ii(3) + ii(1)

Feature selection with Adaboost

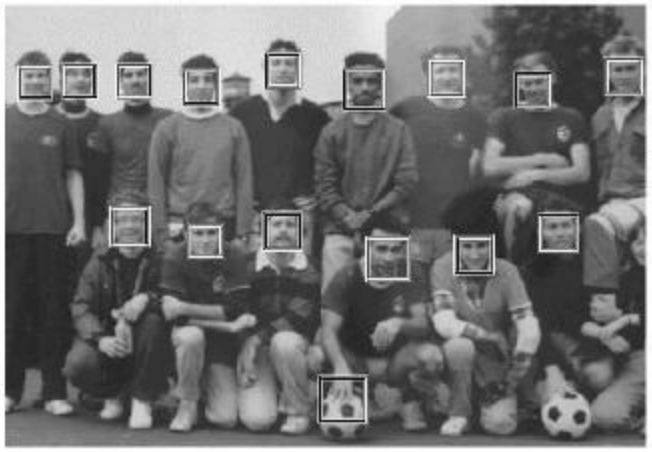
- Create a large pool of features (180K)
- Select features that are discriminative and work well together
  - "Weak learner" = feature + threshold + parity

$$h_j(x) = \begin{cases} 1 & \text{if } p_j f_j(x) < p_j \theta_j \\ 0 & \text{otherwise} \end{cases}$$

  - Choose weak learner that minimizes error on the weighted training set
  - Reweight

Viola Jones Results

Speed = 15 FPS (in 2001)



False detections / Detector	10	31	50	65	78	95	167
Viola-Jones	76.1%	88.4%	91.4%	92.0%	92.1%	92.9%	93.9%
Viola-Jones (voting)	81.1%	89.7%	92.1%	93.1%	93.1%	93.2 %	93.7%
Rowley-Baluja-Kanade	83.2%	86.0%	-	-	-	89.2%	90.1%
Schneiderman-Kanade	-	-	-	94.4%	-	-	-
Roth-Yang-Ahuja	-	-	-	-	(94.8%)	-	-

MIT + CMU face dataset

Object Detection

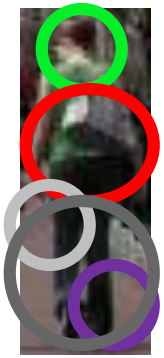
- Overview
- Viola-Jones
- Dalal-Triggs
- Deformable models
- Deep learning

Statistical Template

Object model = sum of scores of features at fixed positions

+3 +2 -2 -1 -2.5 = -0.5 ? > 7.5

Non-object



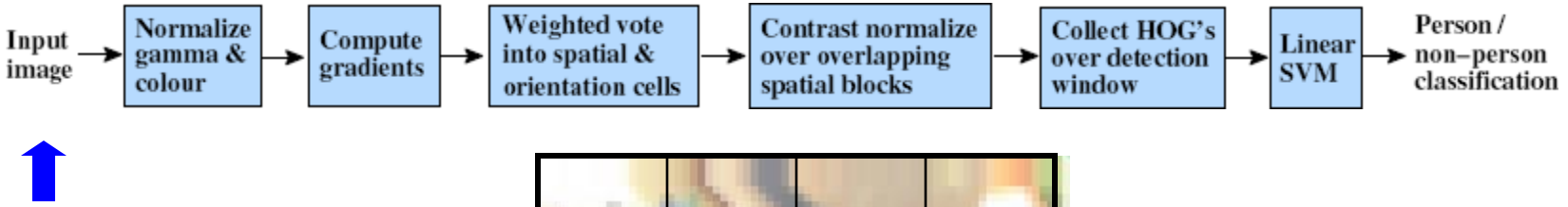
+4 +1 +0.5 +3 +0.5 = 10.5 ? > 7.5

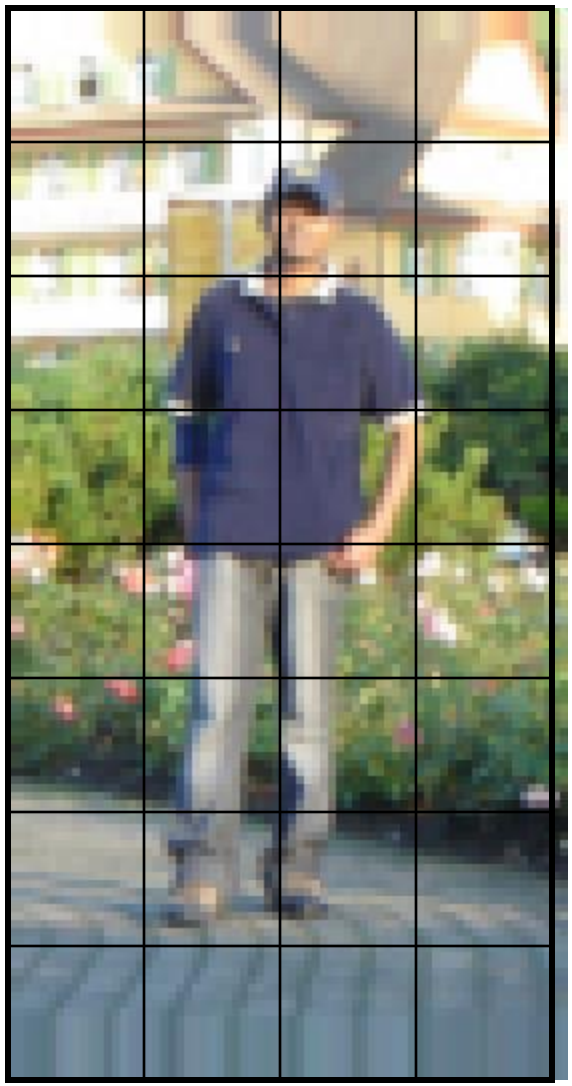
Object

Example: Dalal-Triggs pedestrian detector

1. Extract fixed-sized (64x128 pixel) window at each position and scale
2. Compute HOG (histogram of gradient) features within each window
3. Score the window with a linear SVM classifier
4. Perform non-maxima suppression to remove overlapping detections with lower scores

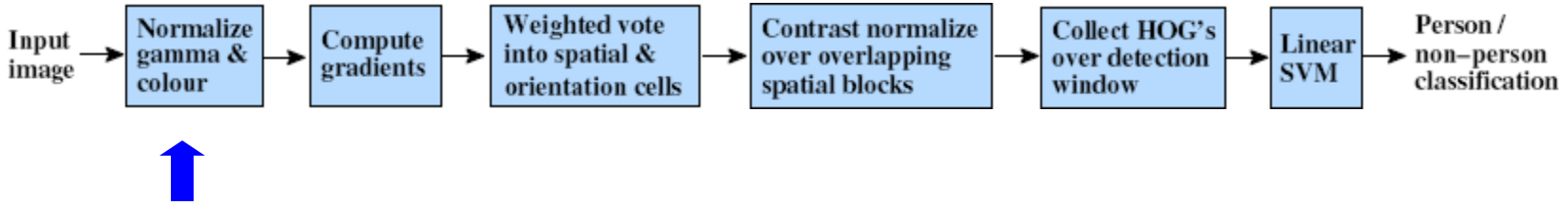
Navneet Dalal and Bill Triggs, Histograms of Oriented Gradients for Human Detection, CVPR05





Slides by Pete Barnum

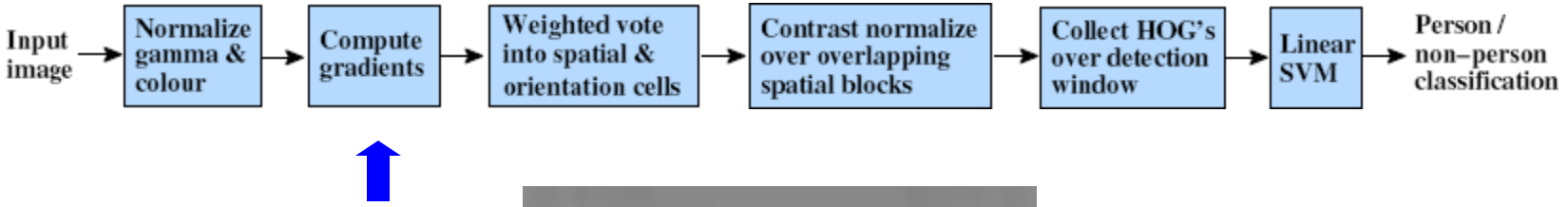
Navneet Dalal and Bill Triggs, Histograms of Oriented Gradients for Human Detection, CVPR05



- Tested with
  - RGB
  - LAB
  - Grayscale

  } Slightly better performance vs. grayscale
- Gamma Normalization and Compression
  - Square root
  - Log

  } Very slightly better performance vs. no adjustment



Outperforms

-1	0	1

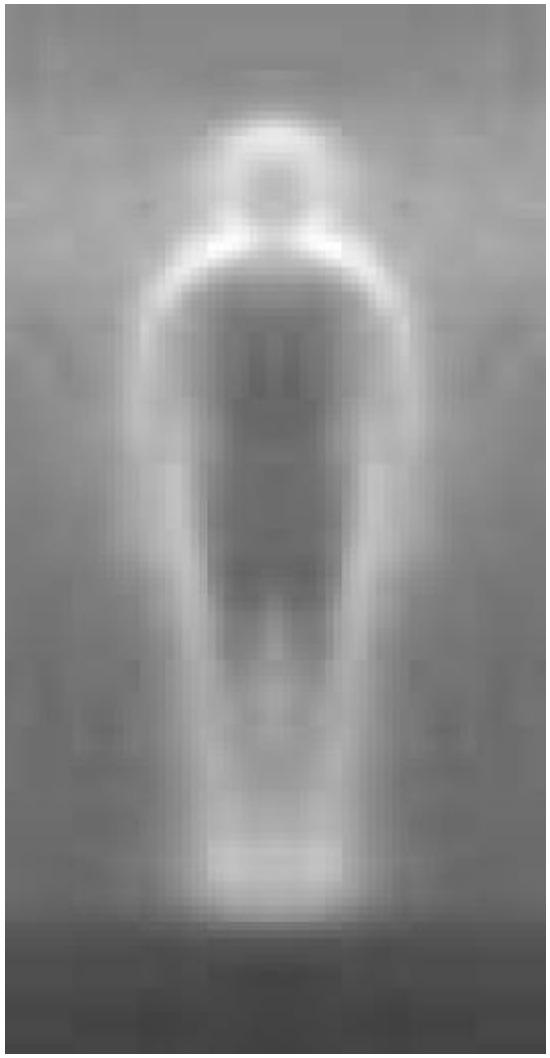
centered

-1	1

uncentered

1	-8	0	8	-1

cubic-corrected



0	1
-1	0

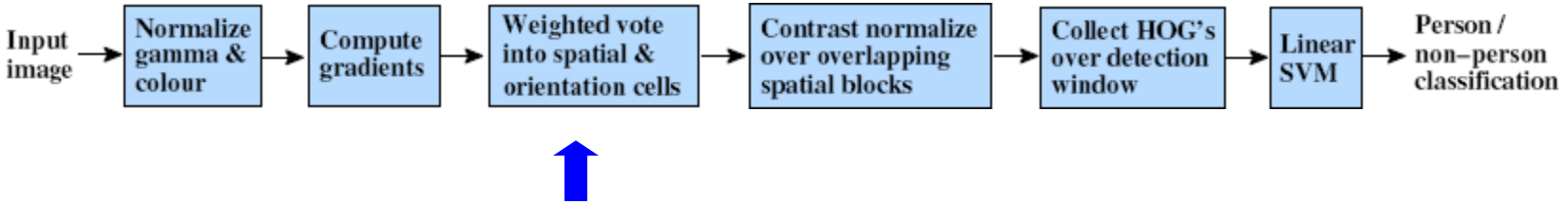
diagonal

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

Sobel

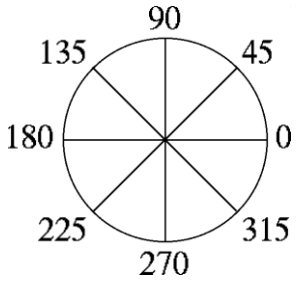
Slides by Pete Barnum

Navneet Dalal and Bill Triggs, Histograms of Oriented Gradients for Human Detection, CVPR05

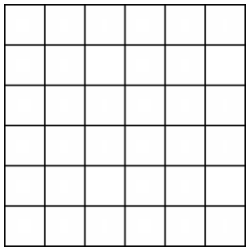


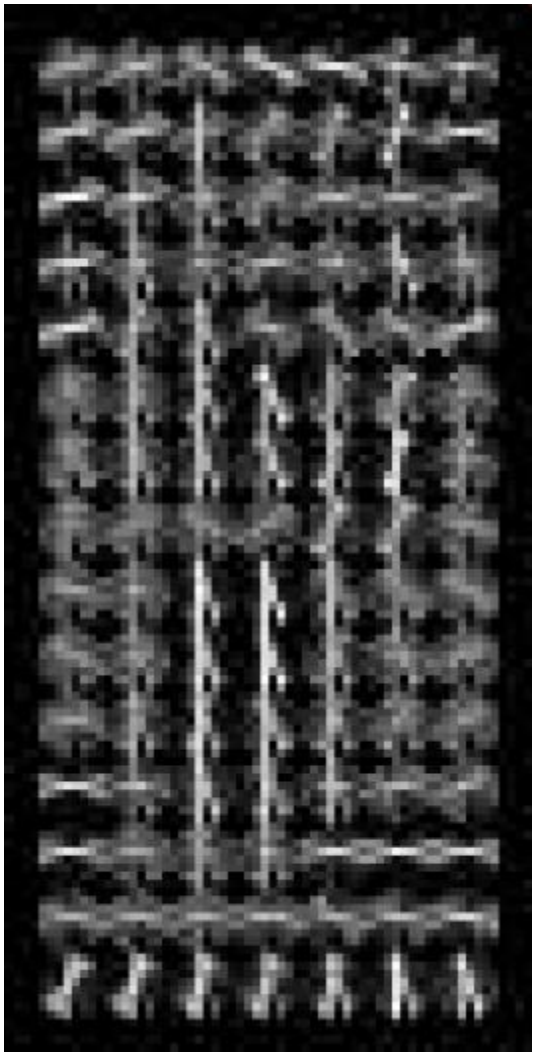
- Histogram of gradient orientations

Orientation: 9 bins
(for unsigned angles
0 -180)



Histograms in
k x k pixel cells

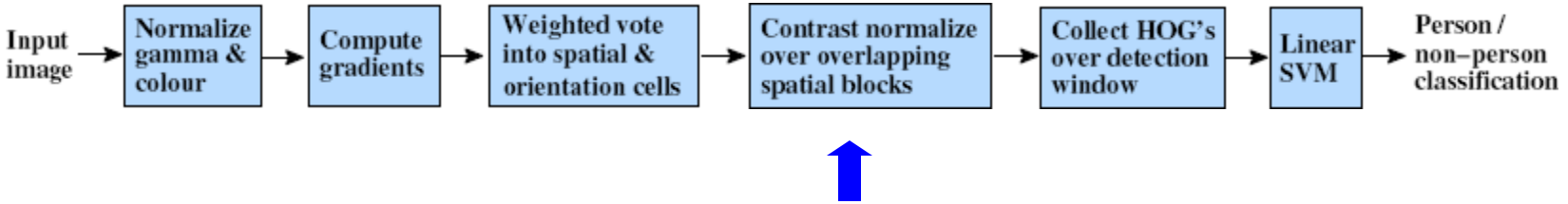




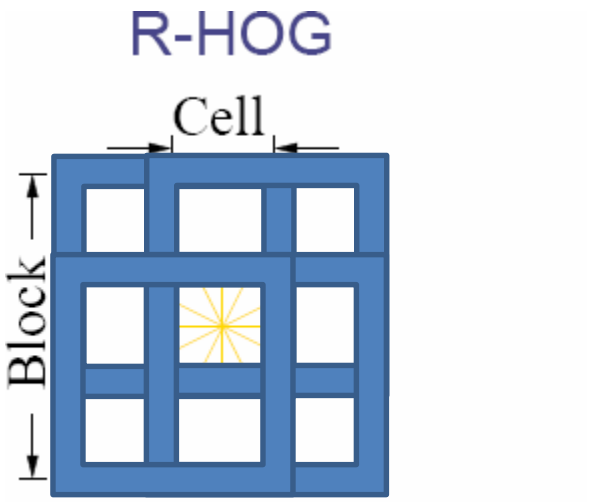
  - Votes weighted by magnitude
  - Bilinear interpolation between cells

Slides by Pete Barnum

Navneet Dalal and Bill Triggs, Histograms of Oriented Gradients for Human Detection, CVPR05

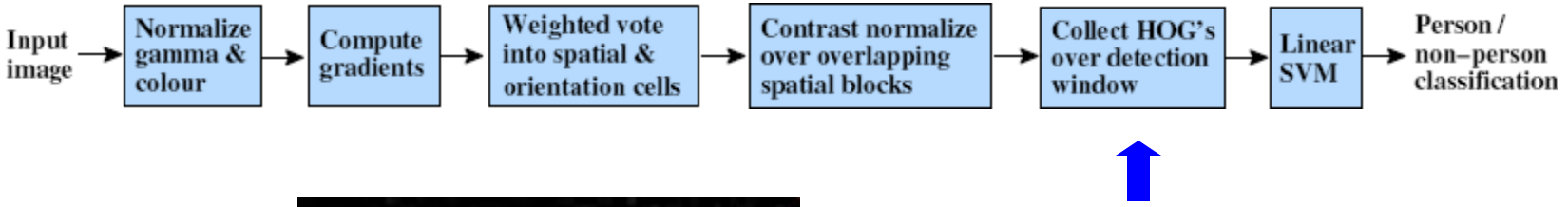


Normalize with respect to surrounding cells

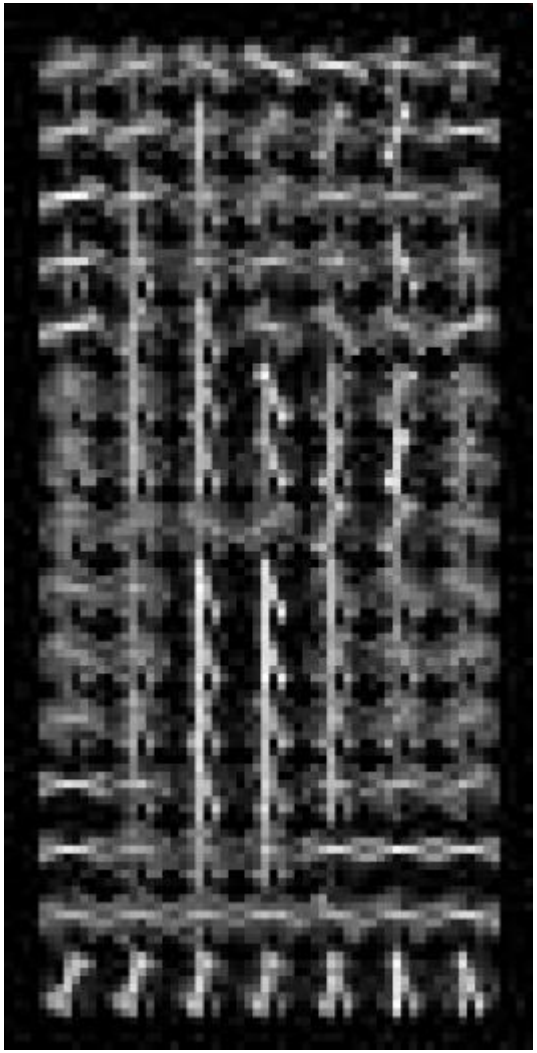


$$L2-norm: v \longrightarrow v/\sqrt{||v||_2^2+\epsilon^2}$$

Navneet Dalal and Bill Triggs, Histograms of Oriented Gradients for Human Detection, CVPR05



X=



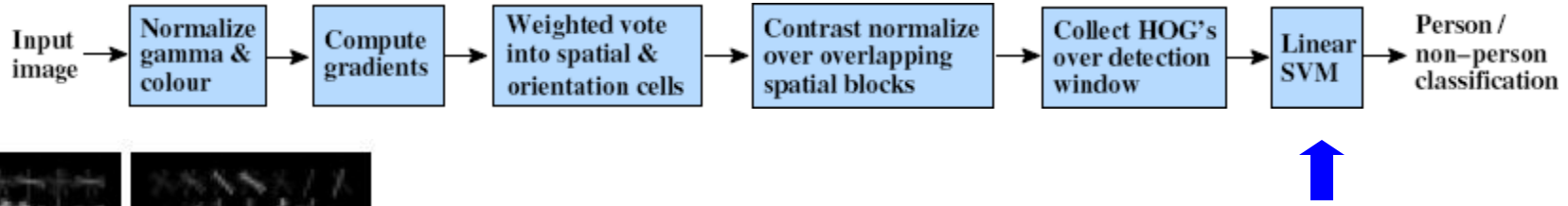
Original Formulation

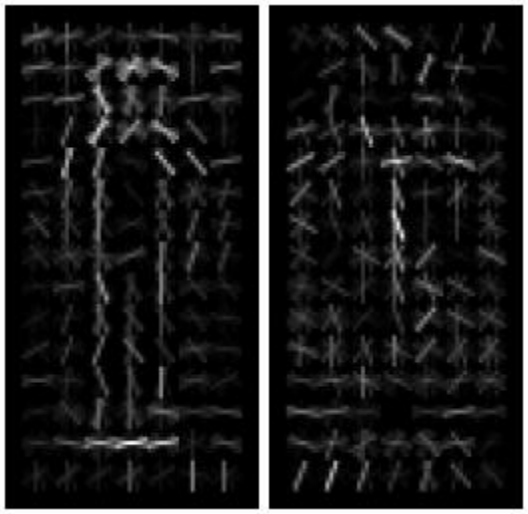
orientations

features = 15 x 7 x 9 x 4 = 3780

cells

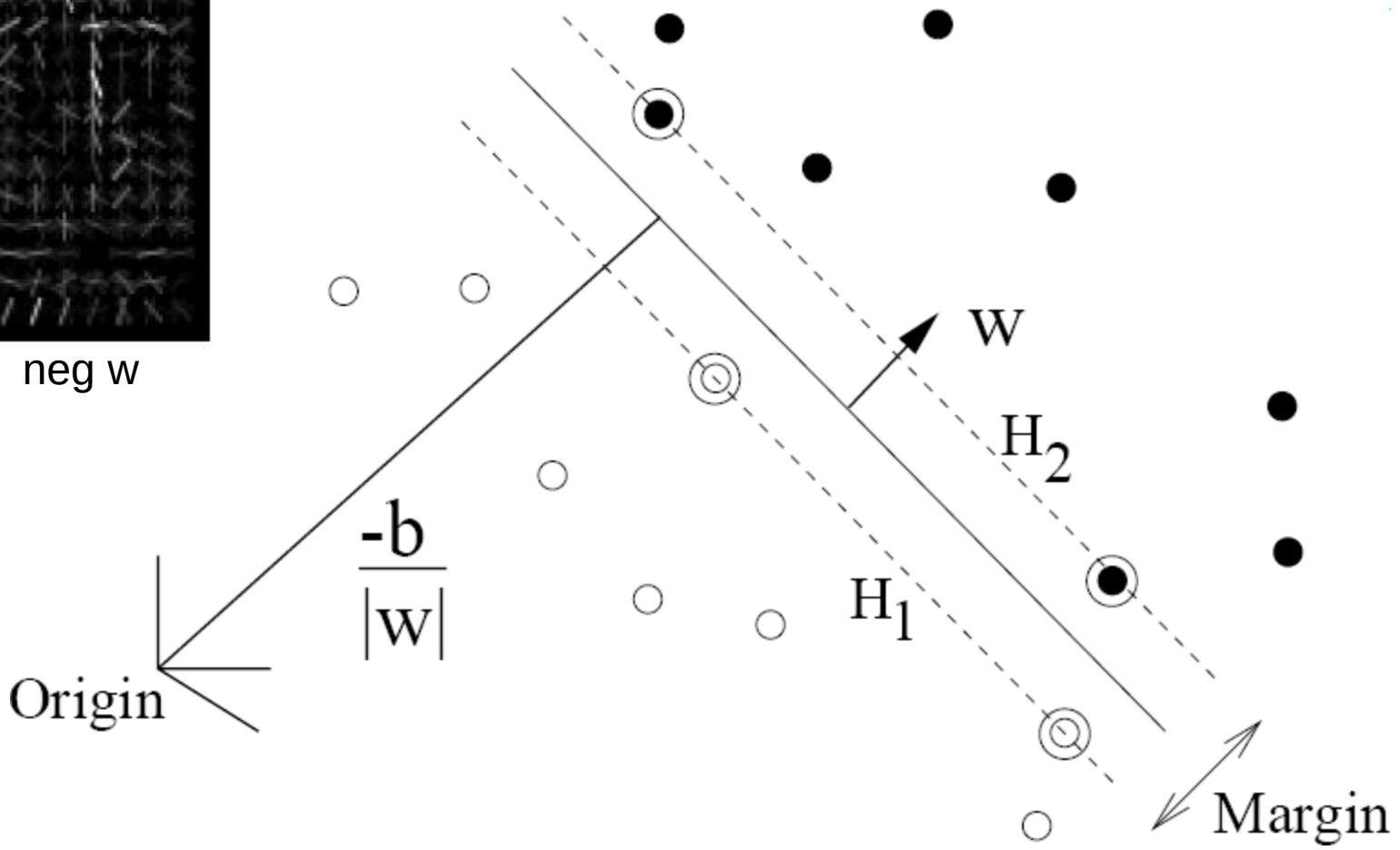
normalizations by neighboring cells





pos w

neg w



Slides by Pete Barnum

Navneet Dalal and Bill Triggs, Histograms of Oriented Gradients for Human Detection, CVPR05

$$0.16 = w^T x - b$$

$$sign(0.16) = 1$$

$$=>$$ pedestrian

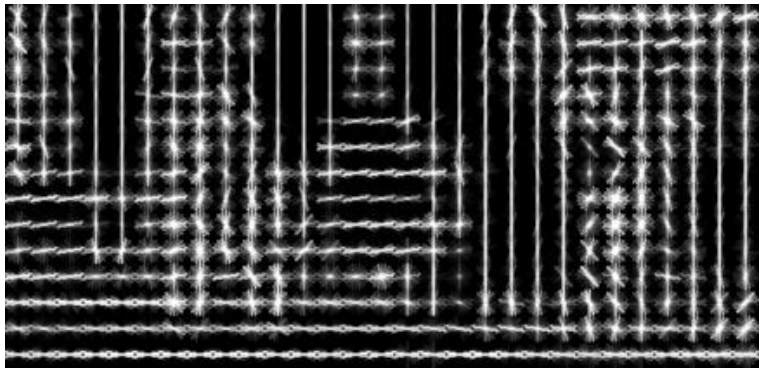
Slides by Pete Barnum

Navneet Dalal and Bill Triggs, Histograms of Oriented Gradients for Human Detection, CVPR05

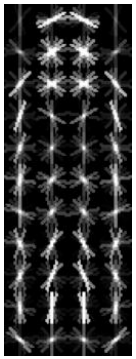
Pedestrian detection with HOG

- Train a pedestrian template using a linear support vector machine
- At test time, convolve feature map with template
- Find local maxima of response
- For multi-scale detection, repeat over multiple levels of a HOG *pyramid*

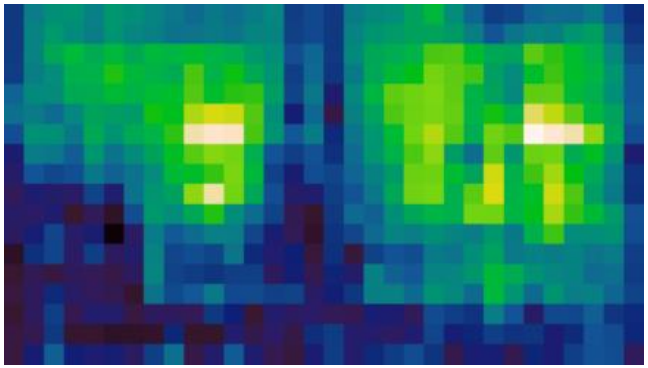
HOG feature map



Template



Detector response map



N. Dalal and B. Triggs, Histograms of Oriented Gradients for Human Detection, CVPR 2005

Something to think about…

- Sliding window detectors work
  - *very well* for faces
  - *fairly well* for cars and pedestrians
  - *badly* for cats and dogs
- Why are some classes easier than others?

Strengths and Weaknesses of Statistical Template Approach

Strengths

- Works very well for non-deformable objects with canonical orientations: faces, cars, pedestrians
- Fast detection

Weaknesses

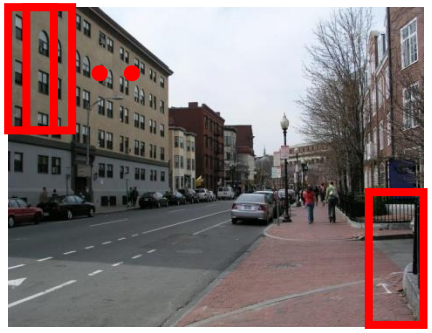
- Not so well for highly deformable objects or “stuff”
- Not robust to occlusion
- Requires lots of training data

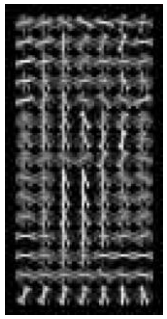
Tricks of the trade

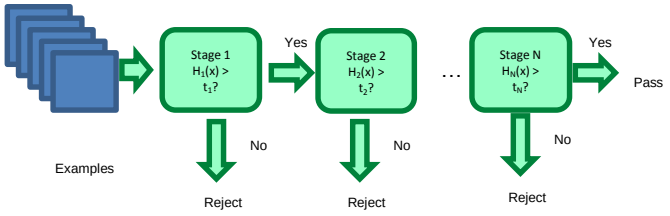
- Details in feature computation really matter
  - E.g., normalization in Dalal-Triggs improves detection rate by 27% at fixed false positive rate
- Template size
  - Typical choice is size of smallest detectable object
- “Jittering” to create synthetic positive examples
  - Create slightly rotated, translated, scaled, mirrored versions as extra positive examples
- Bootstrapping to get hard negative examples
  1. Randomly sample negative examples
  2. Train detector
  3. Sample negative examples that score > -1
  4. Repeat until all high-scoring negative examples fit in memory

Things to remember

- Sliding window for search
- Features based on differences of intensity (gradient, wavelet, etc.)
  - Excellent results require careful feature design
- Boosting for feature selection
- Integral images, cascade for speed
- Bootstrapping to deal with many, many negative examples





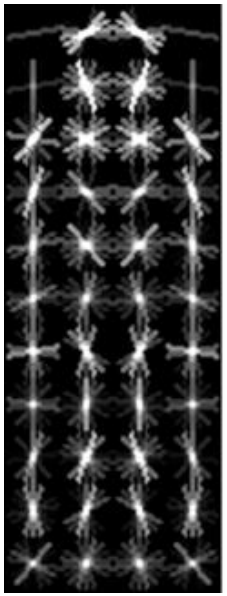


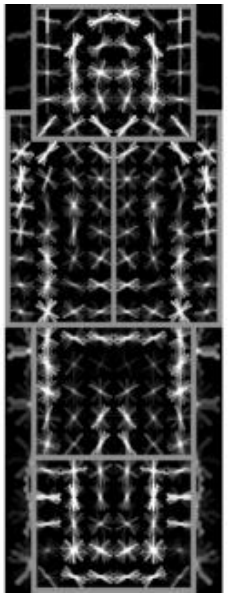
Discriminative part-based models

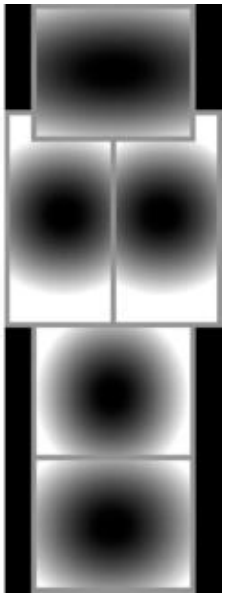
Root filter

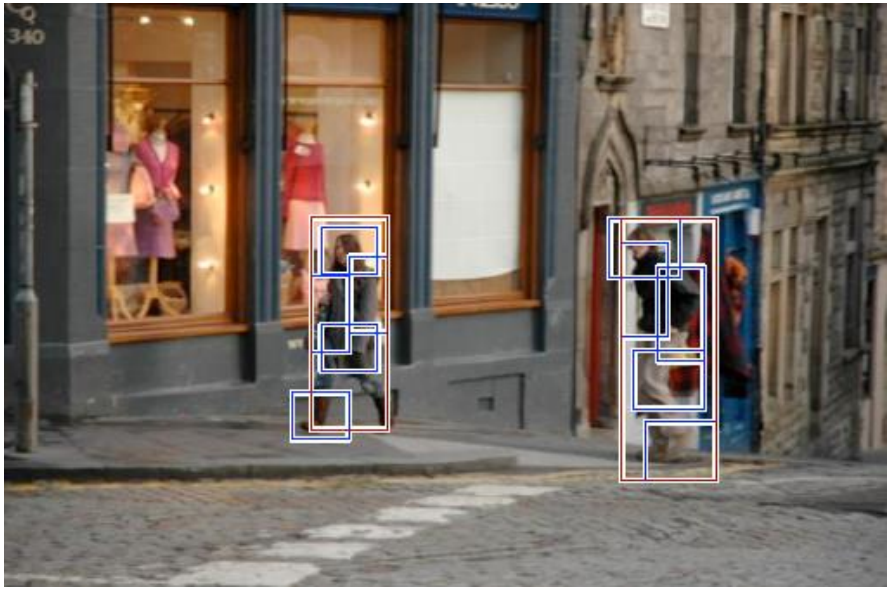
Part filters

Deformation weights





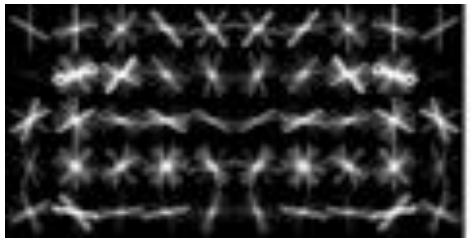
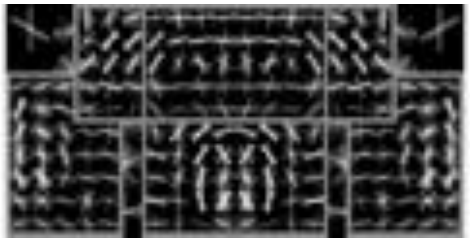
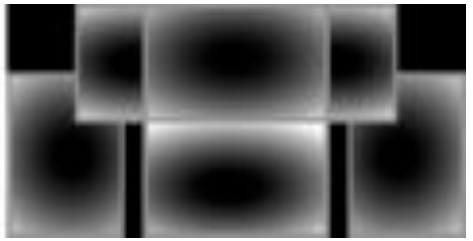




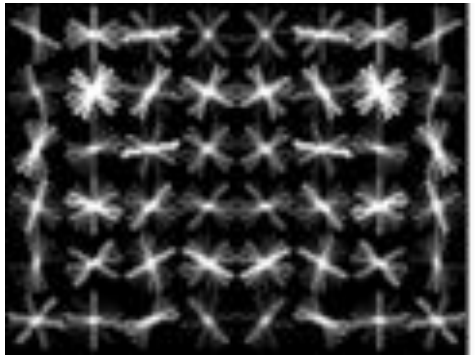
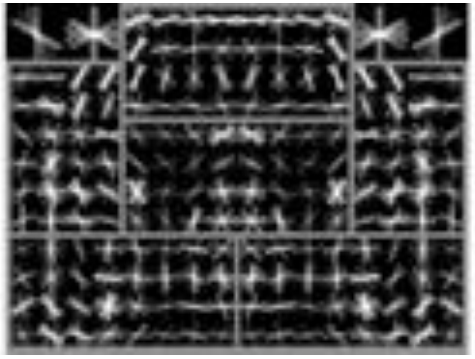
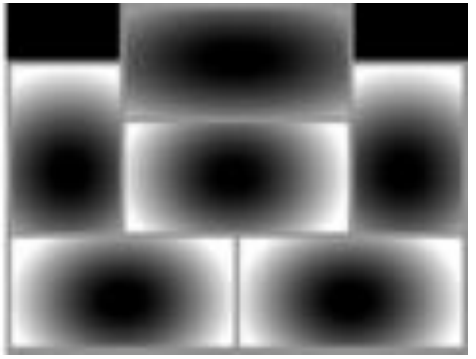
P. Felzenszwalb, R. Girshick, D. McAllester, D. Ramanan, Object Detection with Discriminatively Trained Part Based Models, PAMI 32(9), 2010

Car model

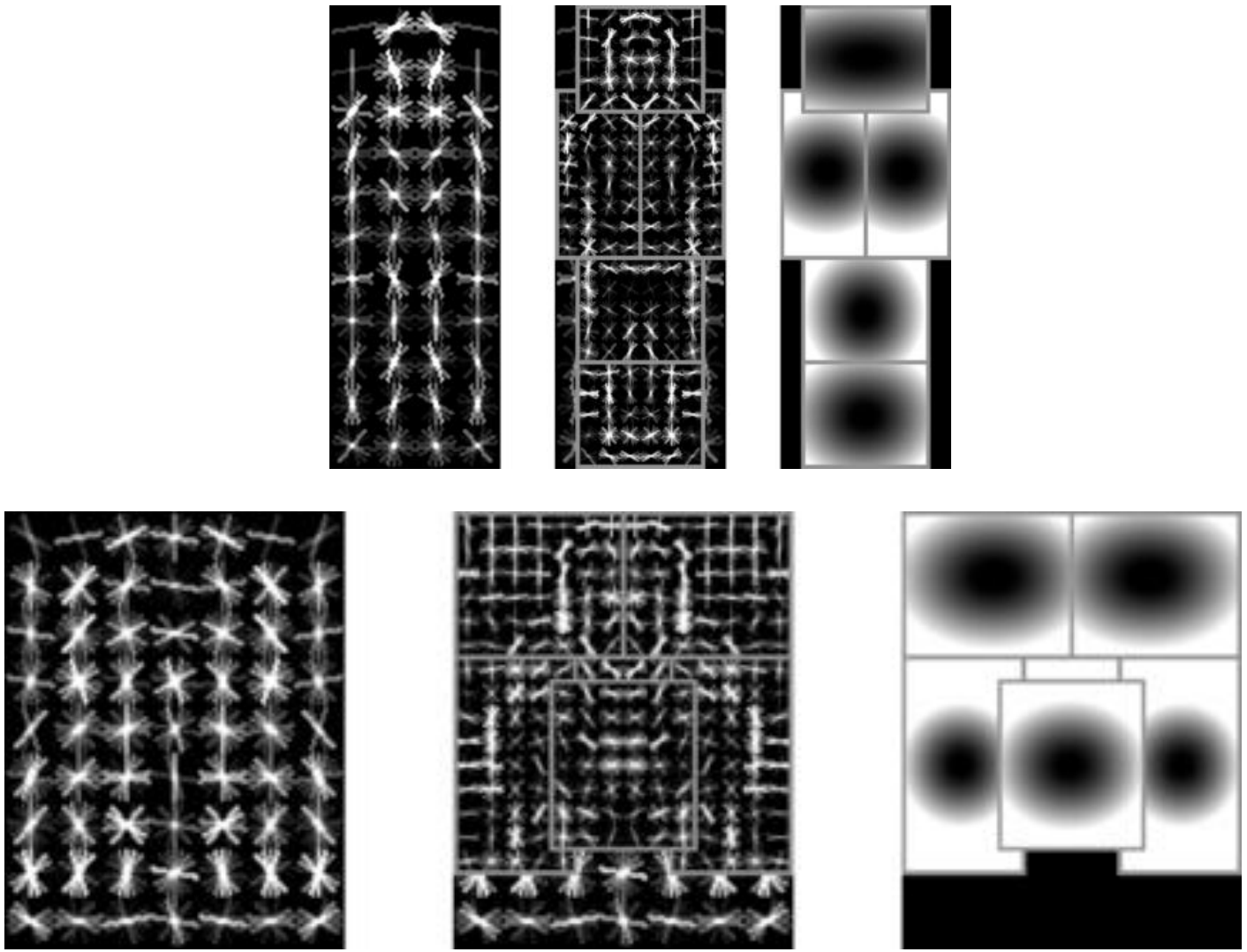
Component 1

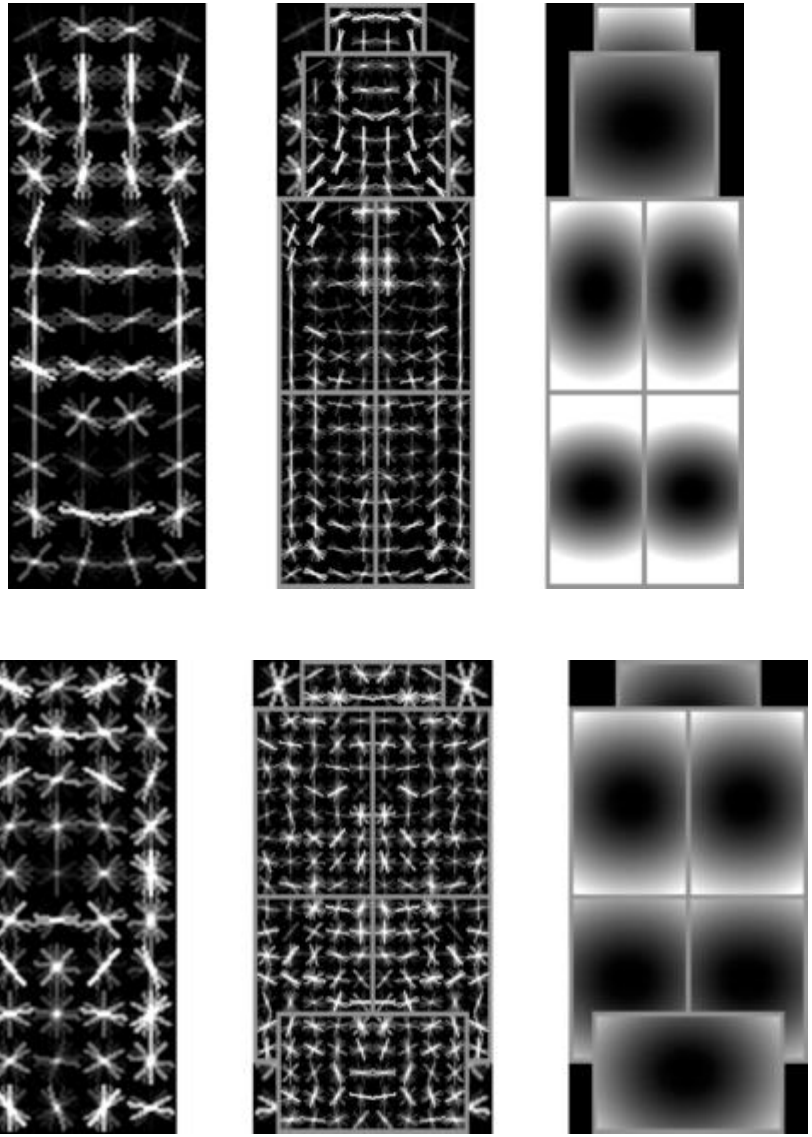
Component 2

Person model

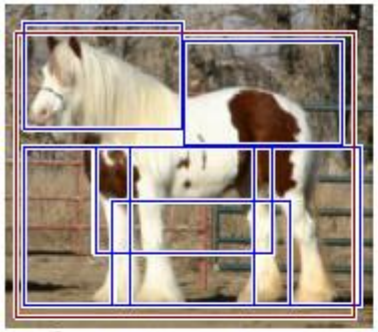
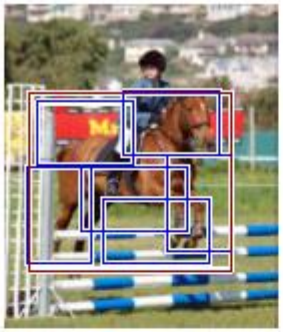
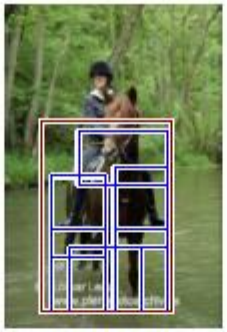
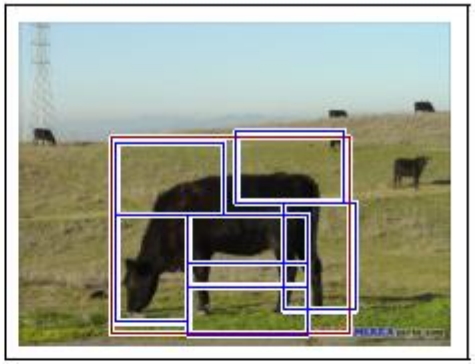
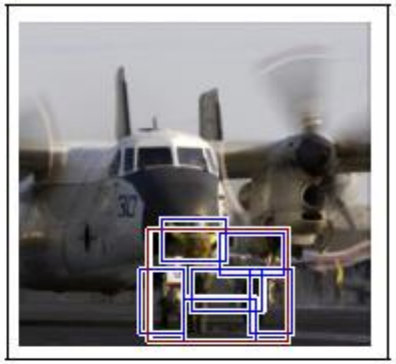


Bottle model

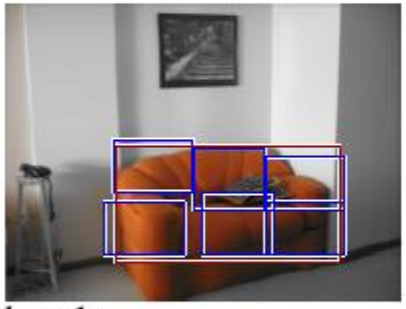


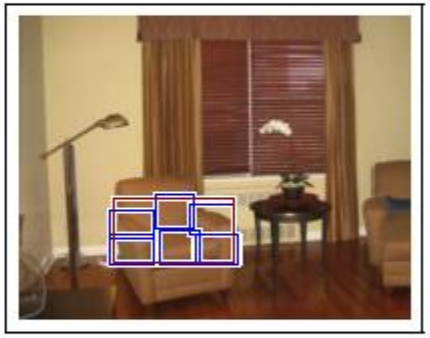
More detections

horse

sofa



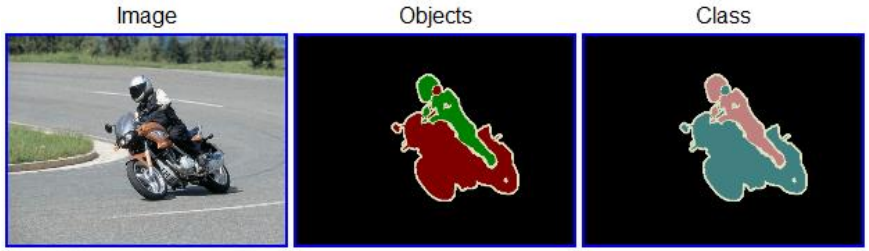



bottle




The PASCAL Visual Object Classes Challenge 2009 (VOC2009)

- Twenty object categories (aeroplane to TV/monitor)
- Three challenges:
  - Classification challenge (is there an X in this image?)
  - Detection challenge (draw a box around every X)
  - Segmentation challenge



Slides from Noah Snavely

Dataset: Collection

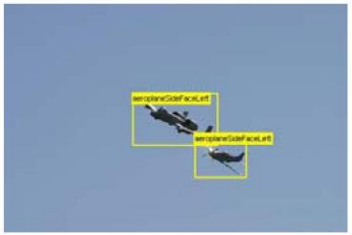
- Images downloaded from **flickr**
  - 500,000 images downloaded and random subset selected for annotation

Dataset: Annotation

- Complete annotation of all objects
- Annotated over web with written guidelines
  - High quality (?)

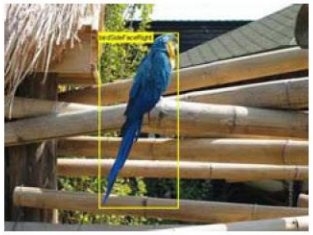
Examples

Aeroplane

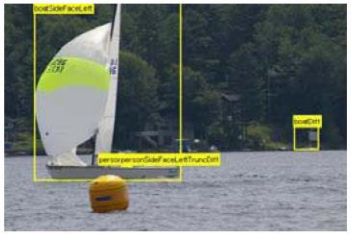


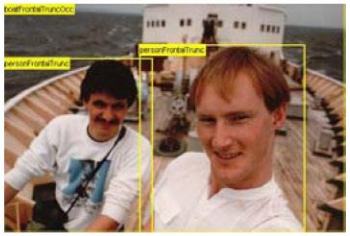
Bicycle

Bird



Boat





Bottle

Bus

Car

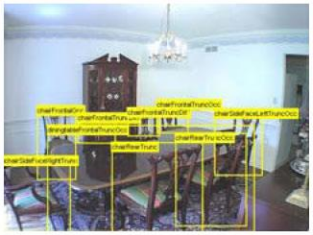
Cat

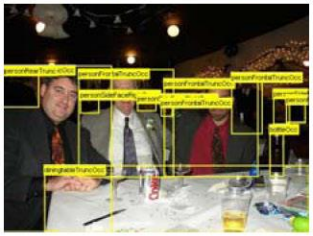
Chair

Cow

Examples

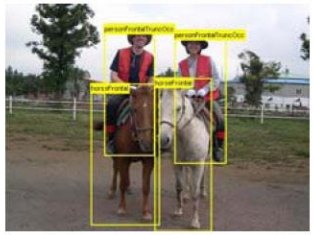
Dining Table



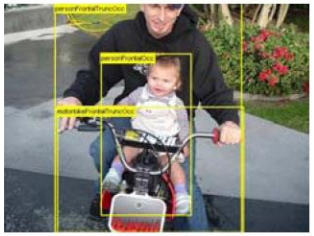


Dog

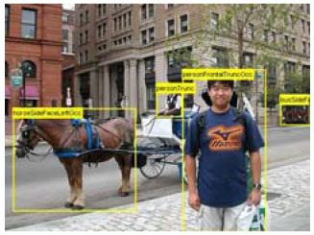
Horse

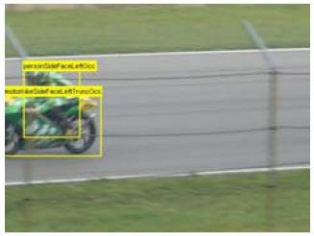


Motorbike

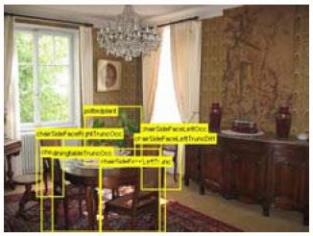


Person

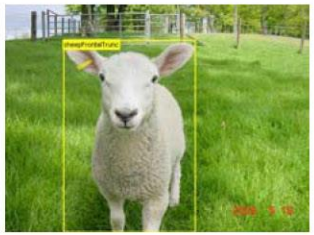




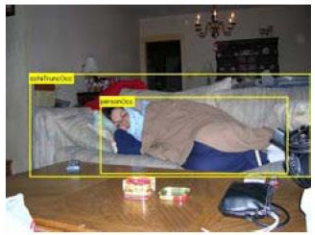
Potted Plant



Sheep

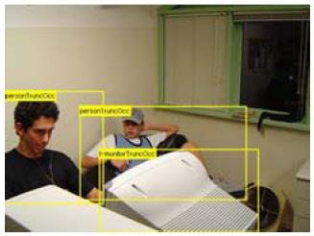


Sofa



Train

TV/Monitor



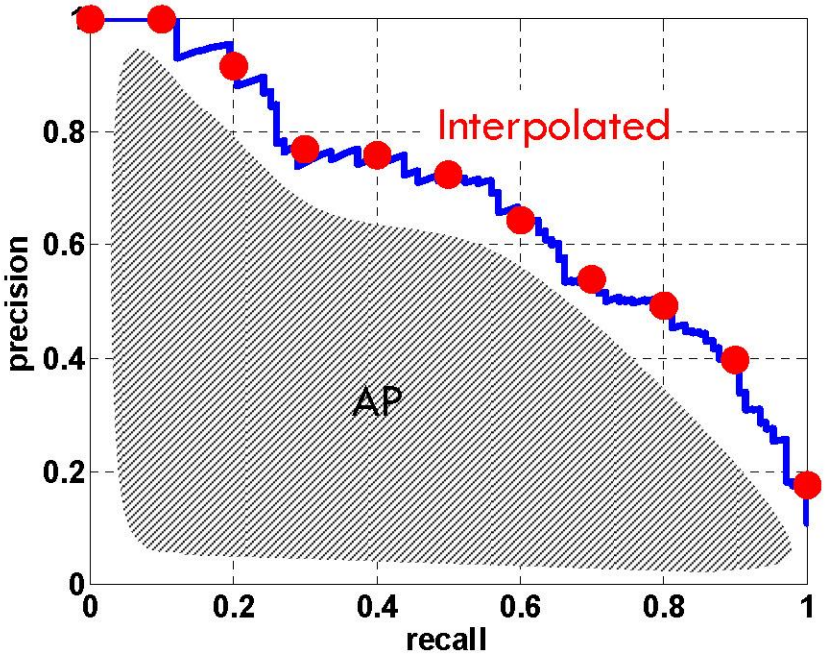
Classification Challenge

- Predict whether at least one object of a given class is present in an image

is there a cat?

Evaluation

- Average Precision [TREC] averages precision over the entire range of recall
  - Curve interpolated to reduce influence of "outliers"



- A good score requires both high recall and high precision
- Application-independent
- Penalizes methods giving high precision but low recall

Participation

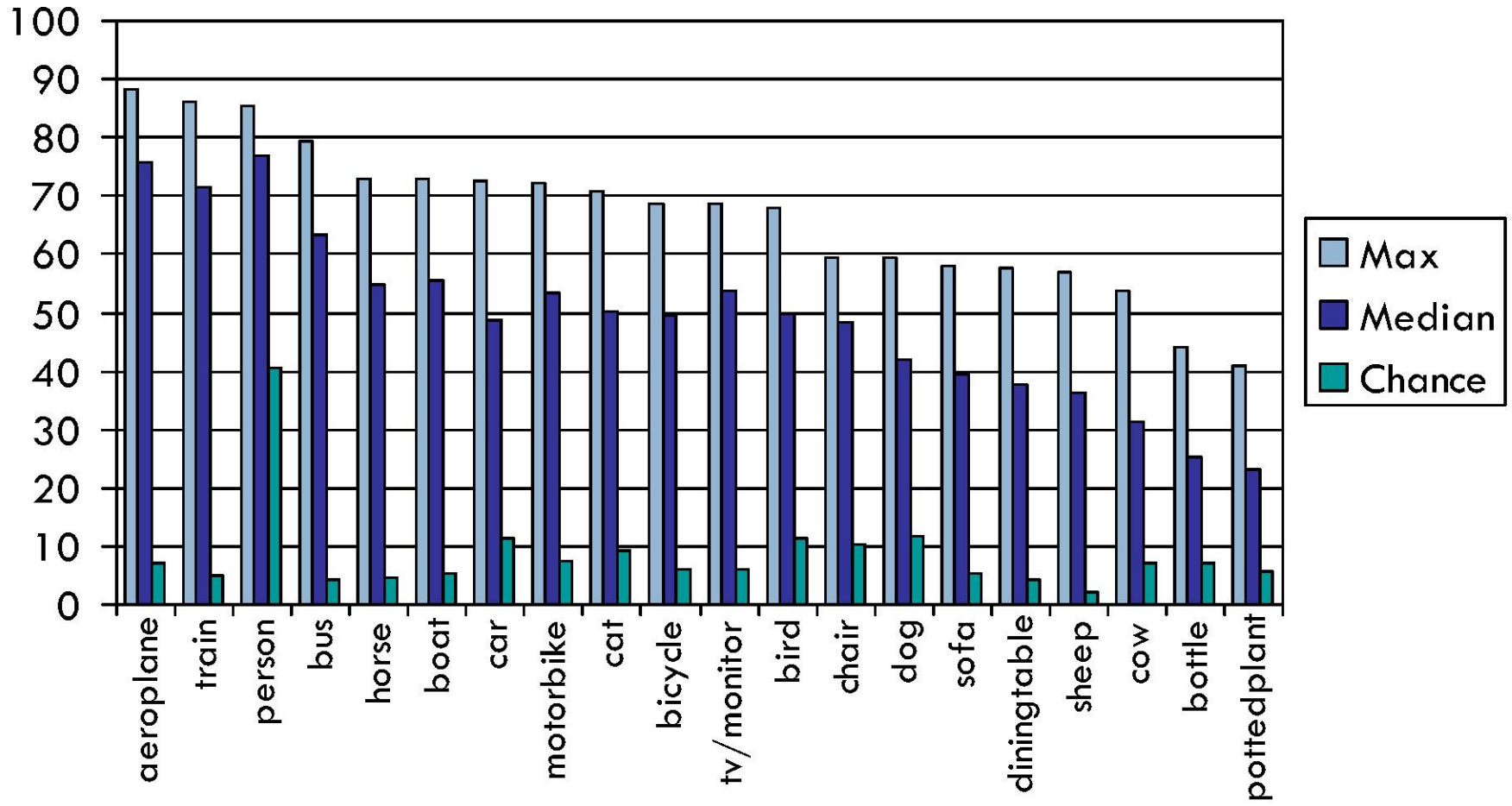
- 48 Methods, 20 Groups

Results: AP by Method and Class

	aero plane	bicycle	bird	boat	bottle	bus	car	cat	chair	cow	dining table	dog	horse	motor bike	person	potted plant	sheep	sofa	train	tv/ monitor
CVC_FLAT	85.3	57.8	66.0	66.1	36.2	70.6	60.6	63.5	55.1	44.6	53.4	49.1	64.4	66.8	84.8	37.4	44.1	47.9	81.9	67.5
CVC_FLAT-HOG-ESS	86.3	60.7	66.4	65.3	41.0	71.7	64.7	63.9	55.5	40.1	51.3	45.9	65.2	68.9	85.0	40.8	49.0	49.1	81.8	68.6
CVC_PLUS	86.6	58.4	66.7	67.3	34.8	70.4	60.0	64.2	52.5	43.0	50.8	46.5	64.1	66.8	84.4	37.5	45.1	45.4	82.1	67.0
FIRSTNIKON_AVGSRKDA	83.3	59.3	62.7	65.3	30.2	71.6	58.2	62.2	54.3	40.7	49.2	50.0	66.6	62.9	83.3	34.2	48.2	46.1	83.4	65.5
FIRSTNIKON_AVGSVM	83.8	58.2	62.6	65.2	32.0	69.8	57.7	61.1	54.5	44.0	50.3	49.6	64.6	61.7	83.2	33.4	46.5	48.0	81.6	65.3
FIRSTNIKON_BOOSTSRKDA	83.0	59.2	61.4	64.6	33.2	71.1	57.5	61.0	54.8	40.7	48.3	50.0	65.5	63.4	82.8	32.8	47.0	47.1	83.3	64.6
FIRSTNIKON_BOOSTSVMS	83.5	56.8	61.8	65.5	33.2	69.7	57.3	60.5	54.6	43.1	48.3	50.3	64.3	62.4	82.3	32.9	46.9	48.4	82.0	64.2
LEAR_CHI-SVM-MULT-LOC	79.5	55.5	54.5	63.9	43.7	70.3	66.4	56.5	54.4	38.8	44.1	46.2	58.5	64.2	82.2	39.1	41.3	39.8	73.6	66.2
NECUIUC_CDCV	88.1	68.0	68.0	72.5	41.0	78.9	70.4	70.4	58.1	53.4	55.7	59.3	73.1	71.3	84.5	32.3	53.3	56.7	86.0	66.8
NECUIUC_CLS-DTCT	88.0	68.6	67.9	72.9	44.2	79.5	72.5	70.8	59.5	53.6	57.5	59.0	72.6	72.3	85.3	36.6	56.9	57.9	85.9	68.0
NECUIUC_LL-CDCV	87.1	67.4	65.8	72.3	40.9	78.3	69.7	69.7	58.5	50.1	55.1	56.3	71.8	70.8	84.1	31.4	51.5	55.1	84.7	65.2
NECUIUC_LN-CDCV	87.7	67.8	68.1	71.1	39.1	78.5	70.6	70.7	57.4	51.7	53.3	59.2	71.6	70.6	84.0	30.9	51.7	55.9	85.9	66.7
UVASURREY_BASELINE	84.1	59.2	62.7	65.4	35.7	70.6	59.8	61.3	56.7	45.3	52.4	50.6	66.1	66.6	83.7	34.8	47.2	47.7	80.8	65.9
UVASURREY_MKFDA+BOW	84.7	63.9	66.1	67.3	37.9	74.1	63.2	64.0	57.1	46.2	54.7	53.5	68.1	70.6	85.2	38.5	47.2	49.3	83.2	68.1
UVASURREY_TUNECOLORKERNELSEL	85.0	62.8	65.1	66.5	37.6	73.5	62.1	62.0	57.4	45.1	54.5	52.5	67.7	69.8	84.8	39.1	46.8	49.9	82.9	68.1
UVASURREY_TUNECOLORSPECKDA	84.6	62.4	65.6	67.2	39.4	74.0	63.4	62.8	56.7	43.8	54.7	52.7	67.3	70.6	85.0	38.8	46.9	50.0	82.2	66.2

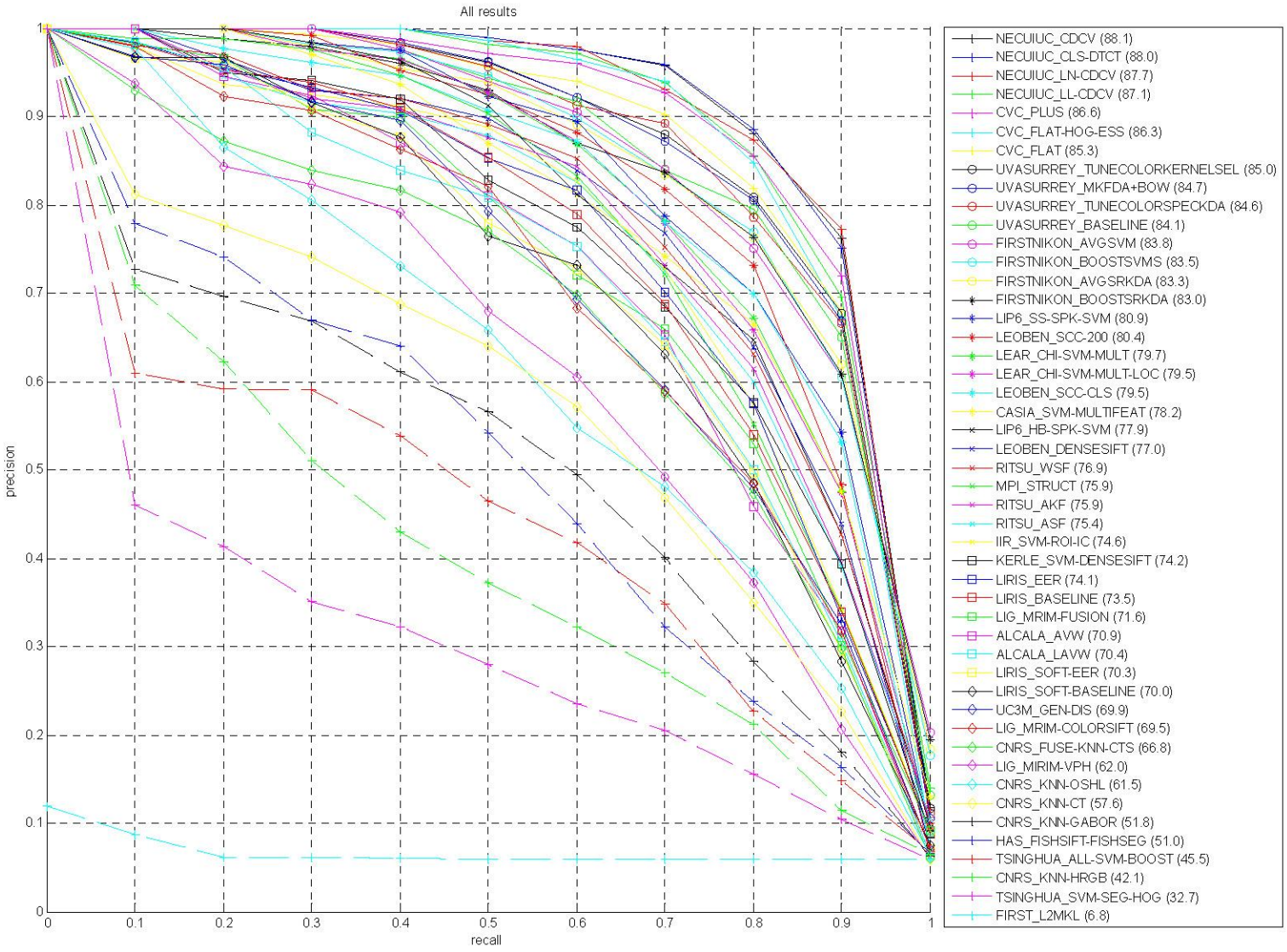
- Only methods in 1st, 2nd or 3rd place by group shown
- Groups: CVC, FIRST/Nikon, NEC/UIUC, UVA/Surrey

AP by Class

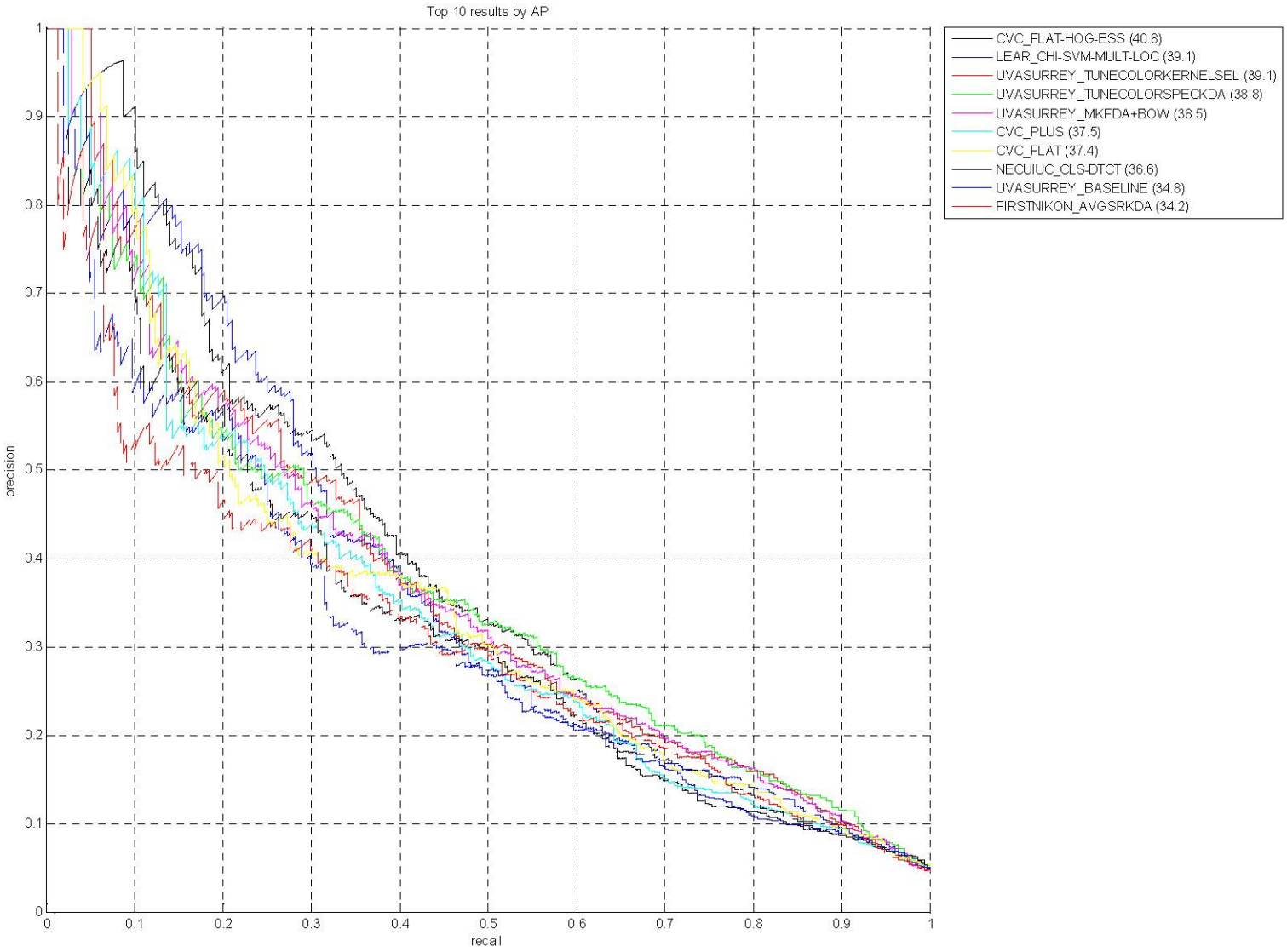


- Max AP: 88.1% (aeroplane) ... 40.8% (potted plant)

Precision/Recall: Aeroplane (All)



Precision/Recall: Potted plant (Top 10 by AP)

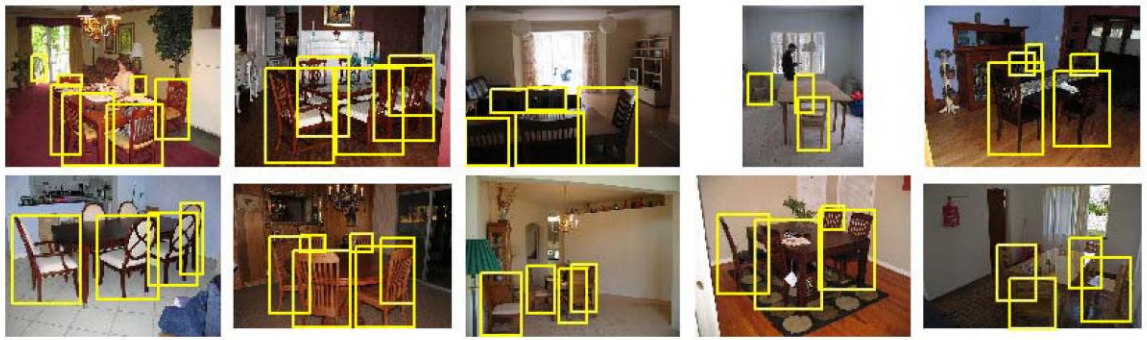


Ranked Images: Aeroplane

- Class images:
  Highest ranked

Ranked Images: Chair

- Class images:
  Highest ranked

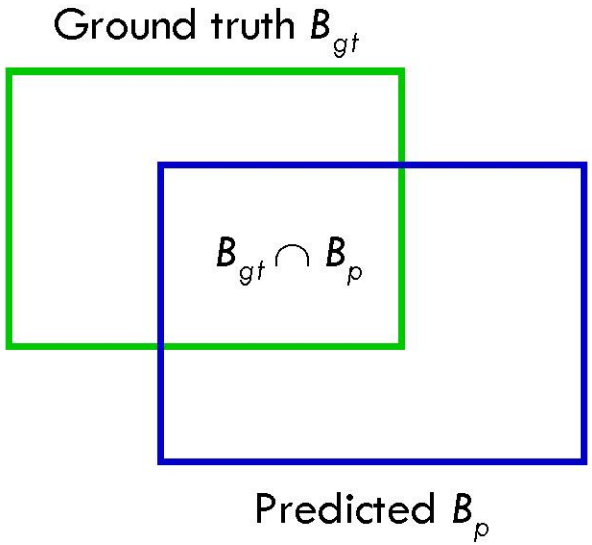


Detection Challenge

- Predict the bounding boxes of all objects of a given class in an image (if any)

Evaluating Bounding Boxes

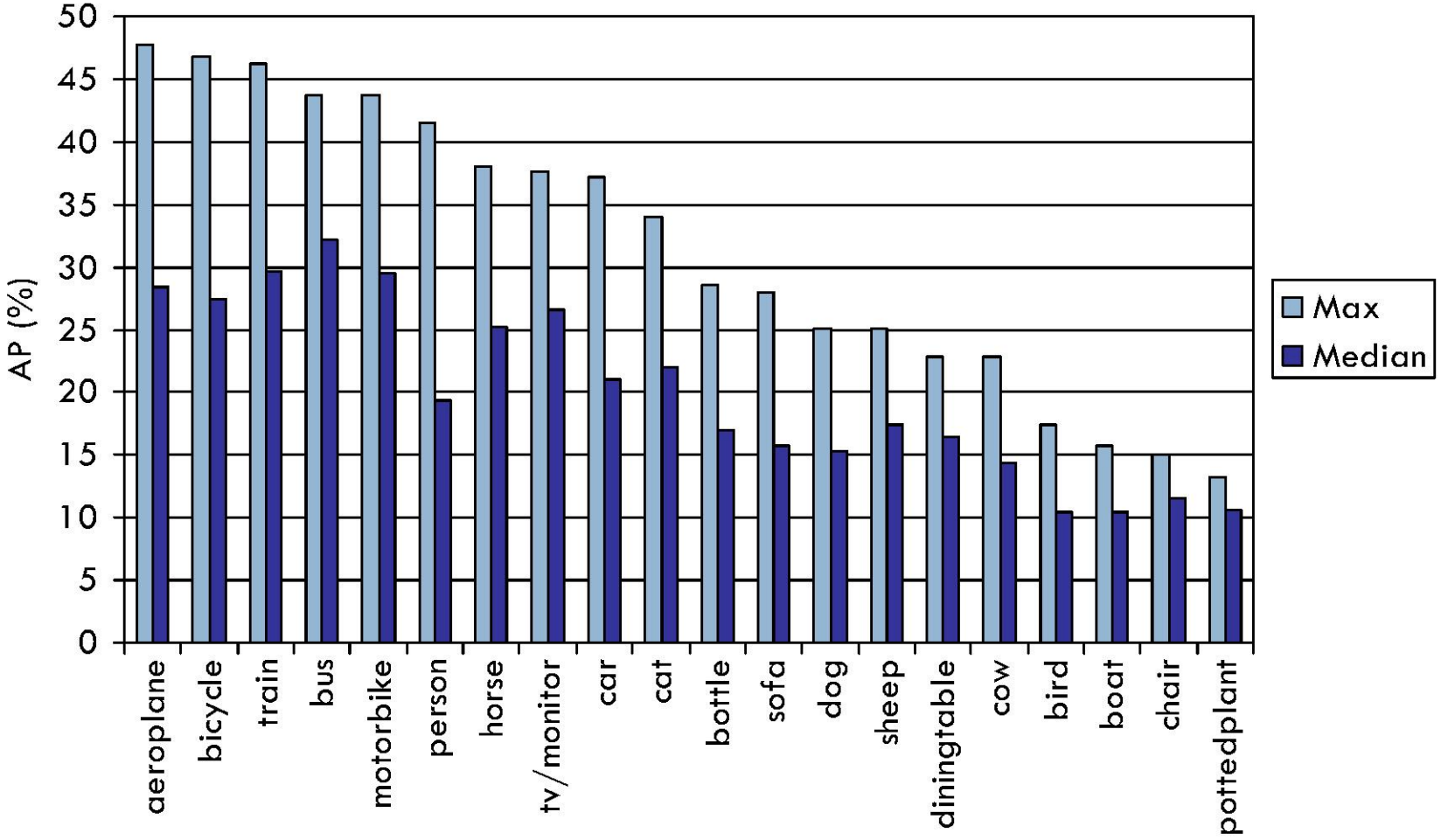
- Area of Overlap (AO) Measure



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

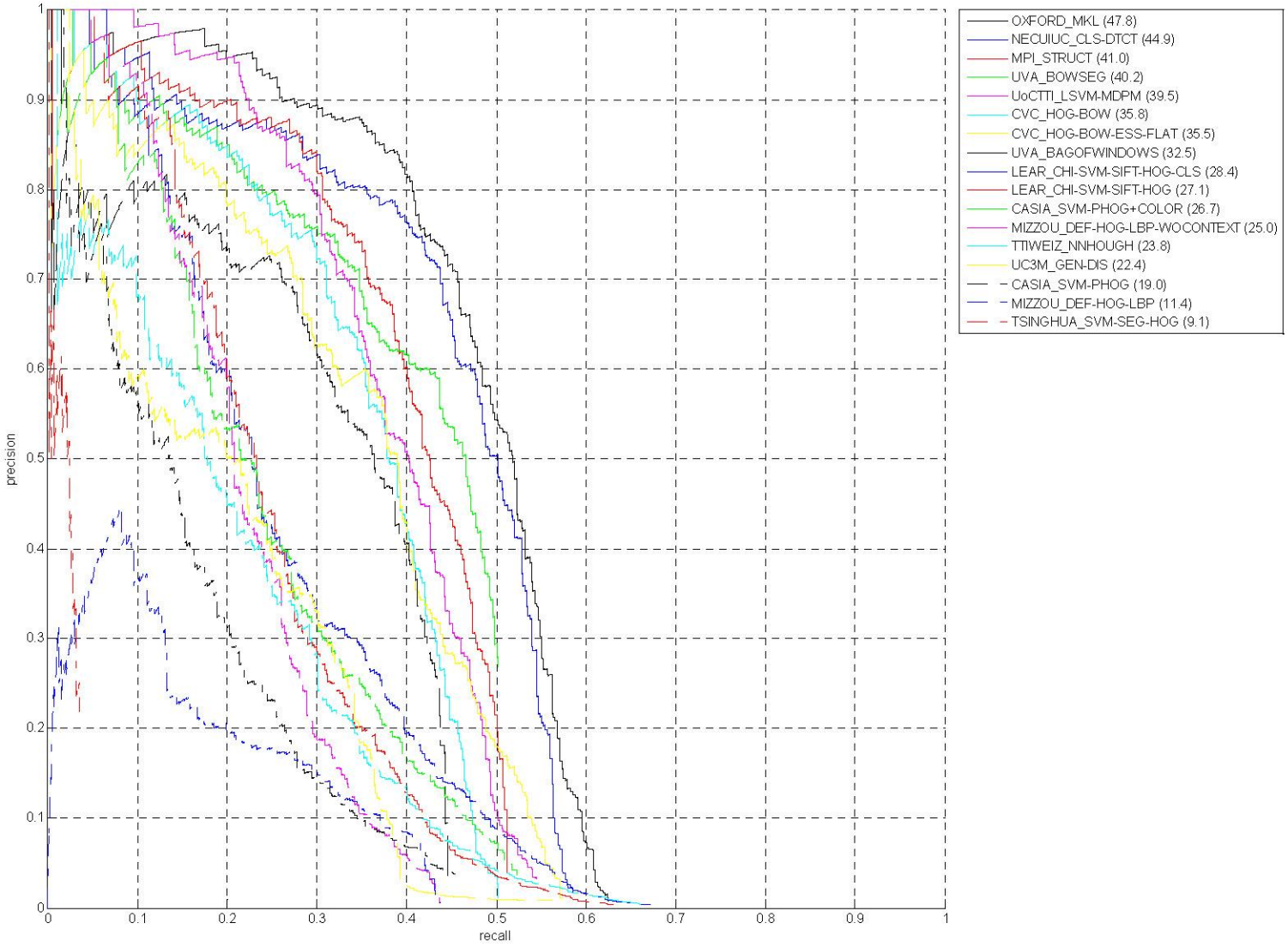
- Need to define a threshold *t* such that $AO(B_{gt},B_p)$ implies a correct detection: 50%

AP by Class

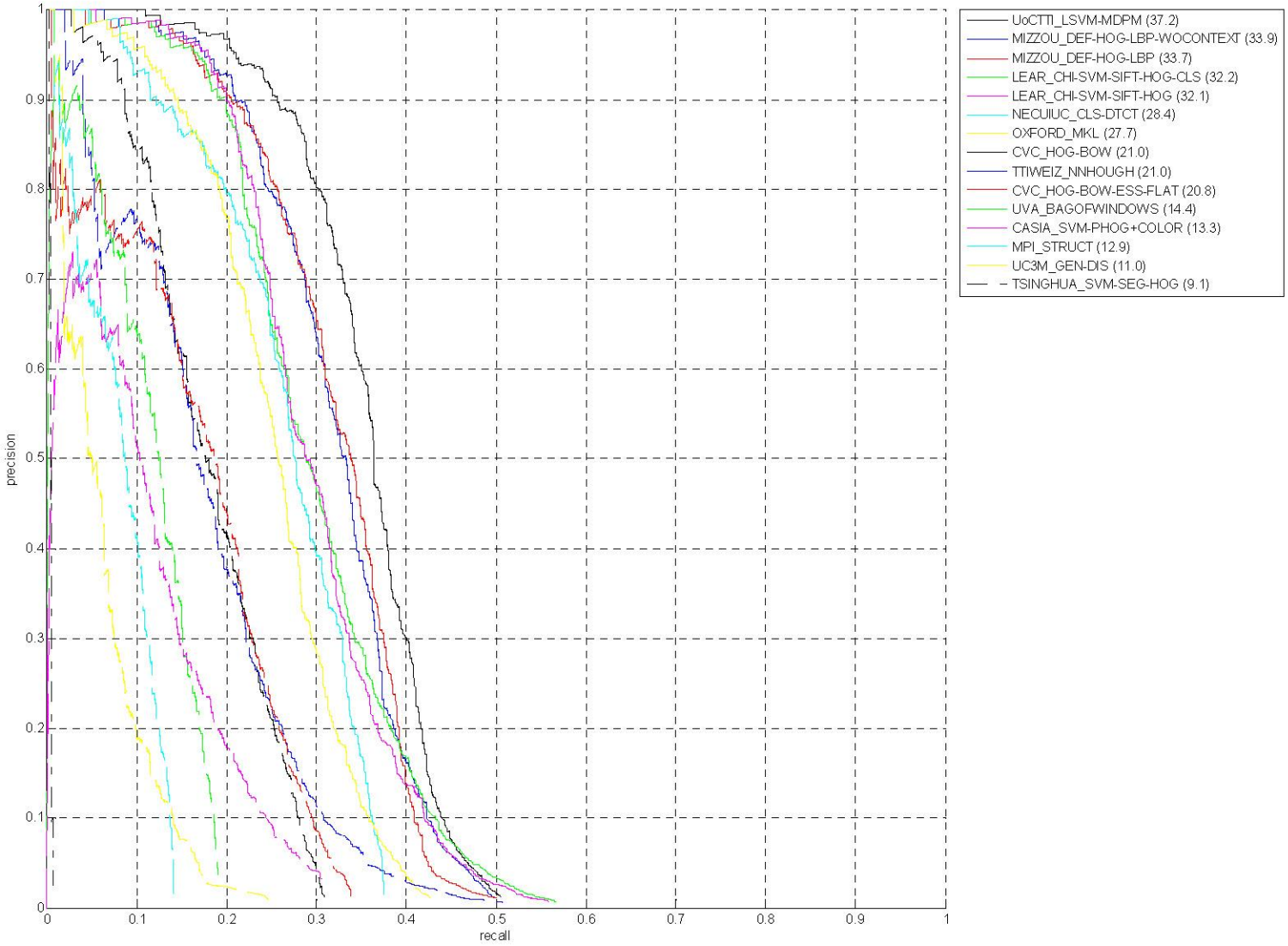


Chance essentially 0

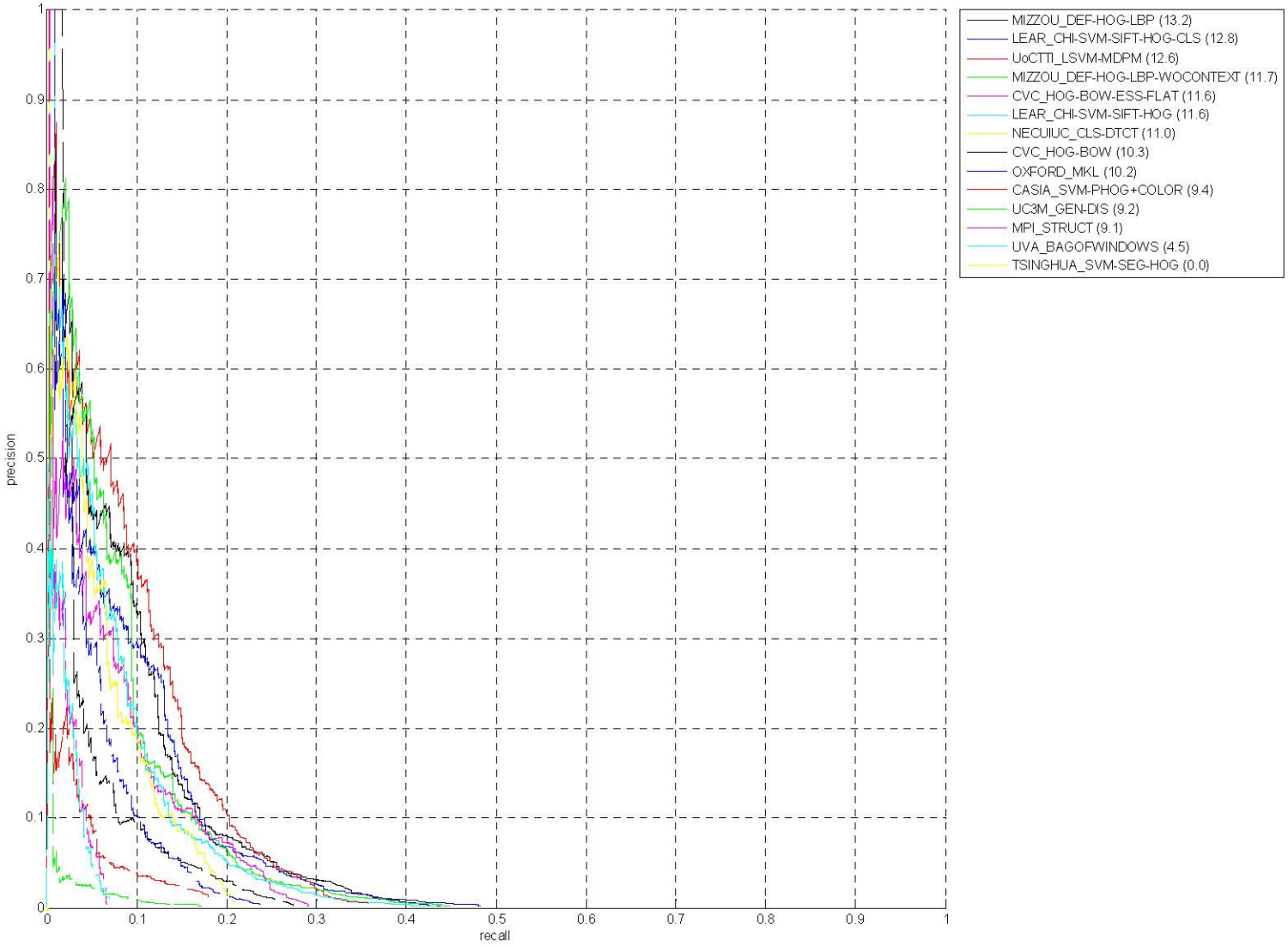
Precision/Recall - Aeroplane



Precision/Recall - Car



Precision/Recall – Potted plant



True Positives - Person

UoCTTI_LSVM-MDPM

MIZZOU_DEF-HOG-LBP

NECUIUC_CLS-DTCT

False Positives - Person

UoCTTI_LSVM-MDPM



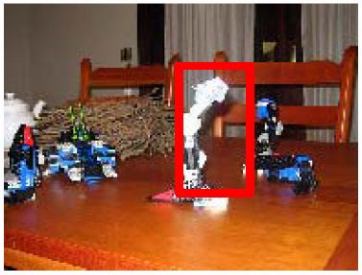


MIZZOU_DEF-HOG-LBP





NECUIUC_CLS-DTCT


“Near Misses” - Person

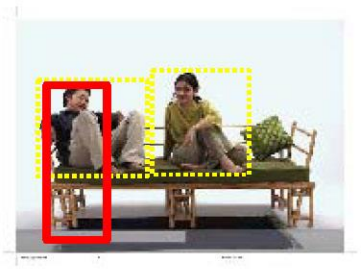
UoCTTI_LSVM-MDPM





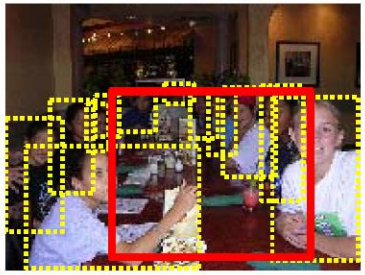
MIZZOU_DEF-HOG-LBP




NECUIUC_CLS-DTCT





True Positives - Bicycle

UoCTTI_LSVM-MDPM





OXFORD_MKL





NECUIUC_CLS-DTCT





False Positives - Bicycle

UoCTTI_LSVM-MDPM





OXFORD_MKL





NECUIUC_CLS-DTCT