max planck institut
informatik

UNIVERSITÄT
DES
SAARLANDES

High Level Computer Vision

Sliding Window Detection: Viola-Jones-Detector & Histogram of Oriented Gradients (HOG)

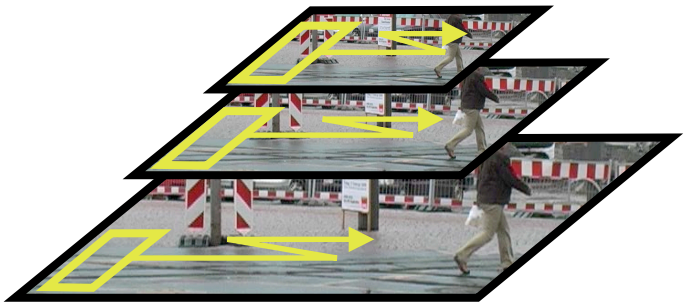
Bernt Schiele - schiele@mpi-inf.mpg.de
Mario Fritz - mfritz@mpi-inf.mpg.de

http://www.d2.mpi-inf.mpg.de/cv

Sliding Window Methods - Overview

- Sliding Window Based People Detection:

Scan Image → Extract Feature Vector → Classify Feature Vector → Non-Maxima Suppression



'slide' detection window
over all positions & scales

Two Important Questions:
1) which feature vector
2) which classifier

Today:
- Viola Jones Face Detector
  - Haar Features
  - Boosting (cascade)
- HOG Pedestrian Detector
  - HOG descriptor
  - linear SVM

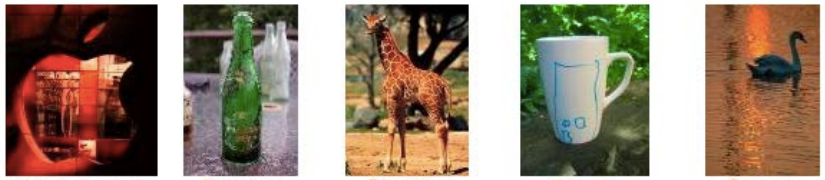
Overview

- Detection task
  - datasets
  - issues
  - evaluation
- Sliding Window Detection
- Viola Jones Face Detector
- HOG Pedestrian Detector

Some Detection Benchmarks

- UIUC car database: only car sideviews (single or multi-scale)

- ETH shape database (5 object classes)



- PASCAL Visual Object Class (VOC) challenge:
  - since 2005; now 20 classes about 10k images with 23k objects
  - recently also segmentation and action classification

Benchmark Database - Discussion

- Positive effects
  - Important tool to benchmark and **compare** methods
  - Important scientific progress has been possible due to well defined challenges
- Caution
  - Results are always conditioned on the database
  - Database has to capture relevant problem
  - Be aware of limitations of database such as bias, annotation scheme and scope of database
- Databases are usually only useful for a certain time:
  - too hard: no progress measurable
  - too easy: fiddling with database artifacts, no significance anymore
- Recommended read:
  - “Dataset Issues in Object Recognition” paper
  - “Unbiased Look at Dataset Bias” paper

Databases: selection and annotation of images needs careful work !

VOC2011 Annotation Guidelines

Guidelines on what and how to label.

What to label

All objects of the defined categories, unless:

- you are **unsure** what the object is.
- the object is **very small** (at your discretion).
- less than 10-20% of the object is **visible**, *such that you cannot be sure what class it is*. e.g. if only a tyre is visible it may belong to car or truck so cannot be labelled car, but feet/faces can only belong to a person.

If this is not possible because too many objects, mark image as bad.

Viewpoint

Record the viewpoint of the 'bulk' of the object e.g. the **body** rather than the **head**. Allow viewpoints within 10-20 degrees.

If ambiguous, leave as 'Unspecified'. Unusually rotated objects e.g. upside-down people should be left as 'Unspecified'.

Bounding box

Mark the bounding box of the **visible area** of the object (*not* the estimated total extent of the object).

Bounding box should **contain all visible pixels**, **except** where the bounding box would have to be made **excessively large to include a few additional pixels (<5%)** e.g. a car aerial.

Truncation

If more than **15-20% of the object lies outside the bounding box** mark as Truncated. The flag indicates that the bounding box does not cover the total extent of the object.

Occlusion

If **more than 5%** of the object is occluded within the bounding box, mark as Occluded. The flag indicates that the object is not totally visible within the bounding box.

Image quality/ illumination

Images which are poor quality (e.g. excessive **motion blur**) should be marked bad. However, **poor illumination** (e.g. objects in silhouette) should not count as poor quality unless objects cannot be recognised.

Images made up of multiple images (e.g. **collages**) should be marked bad.

Clothing/mud/ snow etc.

If an object is 'occluded' by a **close-fitting occluder e.g. clothing, mud, snow** etc., then the occluder should be treated as part of the **object**.

Transparency

Do label objects visible through **glass**, but treat reflections on the glass as occlusion.

Mirrors

Do label objects in **mirrors**.

How to Evaluate?

- Typical evaluation:
  - confusion matrix:

ground truth

classifier prediction	class A	not class A
class A	true positives (TP)	false positives (FP)
not class A	false negative (FN)	true negative (TN)

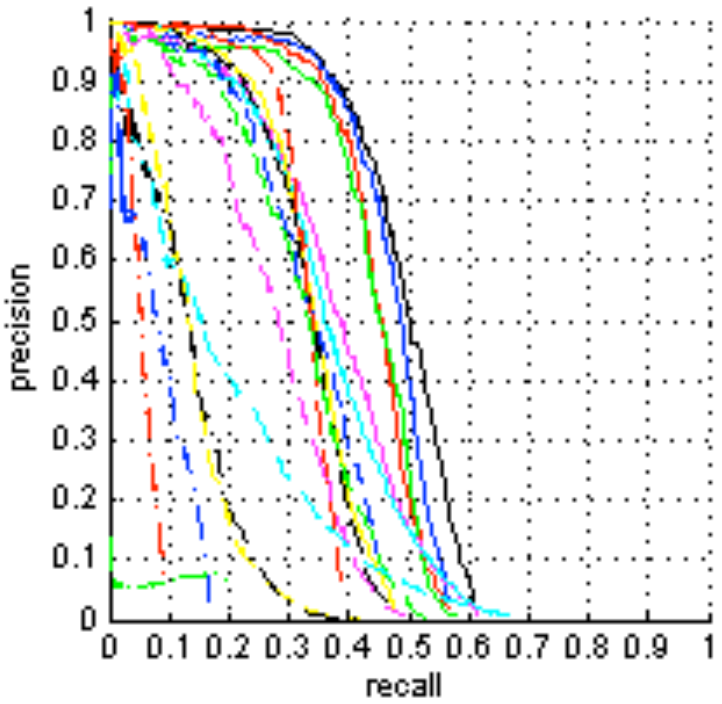
- Precision and recall:

$$precision = \frac{truepositives}{totalnumber\ of\ predictions} = \frac{TP}{TP+FP}$$

$$recall = \frac{truepositives}{totalnumber\ of\ positives} = \frac{TP}{TP+FN}$$

How to evaluate algorithms? Precision-Recall Curve

- Parameterized over score/confidence
- average precision (AP)
  - calculate the precision for many / all recall levels and average
- comments:
  - full recall is hard to achieve
  - random guessing does NOT lead to diagona



How to read result plots in object detection literature?

- Typically evaluated with Precision-Recall Plots and Average Precision (examples from the PASCAL VOC challenge)

Detection Results: VOC2010 data

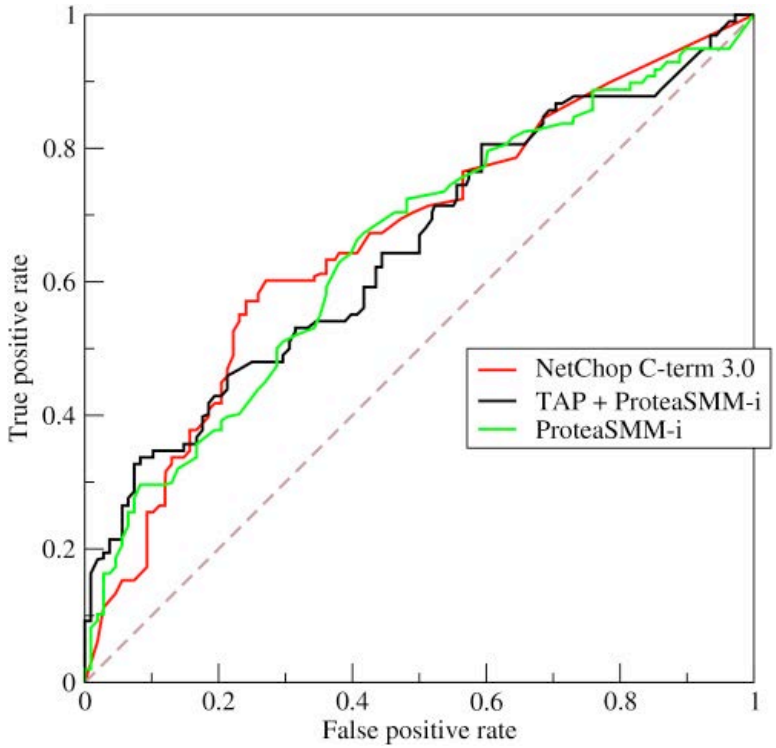
Competition "comp3" (train on VOC2010 data)

Average Precision (AP %)

	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
BONN_FGT_SEGM	52.7	33.7	13.2	11.0	14.2	43.1	31.9	35.6	5.7	25.4	14.4	20.6	38.1	41.7	25.0	5.8	26.3	18.1	37.6	28.1
BONN_SVR_SEGM	50.5	24.4	17.1	13.3	10.9	39.5	32.9	36.5	5.6	16.0	6.6	22.3	24.9	29.0	29.8	6.7	28.4	13.3	32.1	27.2
CMIC_SYNTHTRAIN	-	28.9	-	-	-	30.2	13.3	-	-	-	-	-	26.2	28.1	13.2	-	-	-	18.8	25.7
CMIC_VARPARTS	-	28.2	-	-	-	26.9	13.7	-	-	-	-	-	23.5	24.7	16.1	-	-	-	18.8	24.5
CMU_RANDPARTS	23.8	31.7	1.2	3.4	11.1	29.7	19.5	14.2	0.8	11.1	7.0	4.7	16.4	31.5	16.0	1.1	15.6	10.2	14.7	21.0
CMU_RANDPARTS_MAXSCORE	-	-	2.7	-	-	-	-	16.2	-	10.6	8.5	-	-	-	17.9	-	-	-	15.7	-
LJKINPG_HOG_LBP_LTP_PLS2ROOTS	32.7	29.7	0.8	1.1	19.8	39.4	27.5	8.6	4.5	8.1	6.3	11.0	22.9	34.1	24.6	3.1	24.0	2.0	23.5	27.0
MITUCLA_HIERARCHY	54.2	48.5	15.7	19.2	29.2	55.5	43.5	41.7	16.9	28.5	26.7	30.9	48.3	55.0	41.7	9.7	35.8	30.8	47.2	40.8
NLPR_HOGLBP_MC_LCEGCHLC	53.3	55.3	19.2	21.0	30.0	54.4	46.7	41.2	20.0	31.5	20.7	30.3	48.6	55.3	46.5	10.2	34.4	26.5	50.3	40.3
NUS_HOGLBP_CTX_CLS_RESCORE_V2	49.1	52.4	17.8	12.0	30.6	53.5	32.8	37.3	17.7	30.6	27.7	29.5	51.9	56.3	44.2	9.6	14.8	27.9	49.5	38.4
TIT_SIFT_GMM_MKL	10.5	1.6	1.2	0.9	0.1	2.8	1.6	6.7	0.1	2.0	0.4	3.0	2.0	4.4	2.0	0.3	1.1	1.2	2.1	1.9
TIT_SIFT_GMM_MKL2	20.0	14.5	3.8	1.2	0.5	17.6	8.1	28.5	0.1	2.9	3.1	17.5	7.2	18.8	3.3	0.8	2.9	6.3	7.6	1.1
UC3M_GENDISC	15.8	5.5	5.6	2.3	0.3	10.2	5.4	12.6	0.5	5.6	4.5	7.7	11.3	12.6	5.3	1.5	2.0	5.9	9.1	3.2
UCI_DPM_SP	46.1	52.6	13.8	15.5	28.3	53.2	44.5	26.6	17.6	-	16.1	20.4	45.5	51.2	43.5	11.6	30.9	20.3	47.6	-
UMNECUIUC_HOGLBP_DHOGBOW_SVM	40.4	34.7	2.7	8.4	26.0	43.1	33.8	17.2	11.2	14.3	14.4	14.9	31.8	37.3	30.0	6.4	25.2	11.6	30.0	35.7
UMNECUIUC_HOGLBP_LINSVM	37.9	33.7	2.7	6.5	25.3	37.5	33.1	15.5	10.9	12.3	12.5	13.7	29.6	34.5	33.8	7.2	22.9	9.9	28.9	34.1
UOCTTI_LSVM_MDPM	52.4	54.3	13.0	15.6	35.1	54.2	49.1	31.8	15.5	26.2	13.5	21.5	45.4	51.6	47.5	9.1	35.1	19.4	46.6	38.0
UVA_DETMONKEY	56.7	39.8	16.8	12.2	13.8	44.9	36.9	47.7	12.1	26.9	26.5	37.2	42.1	51.9	25.7	12.1	37.8	33.0	41.5	41.7
UVA_GROUPLOC	58.4	39.6	18.0	13.3	11.1	46.4	37.8	43.9	10.3	27.5	20.8	36.0	39.4	48.5	22.9	13.0	36.8	30.5	41.2	41.9

Receiver Operator Characteristic (ROC - curve)

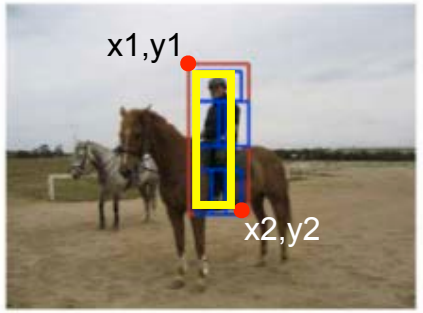
- Parameterized over score/ confidence:
  - one point on the curve (operation point) corresponds to a threshold on score
- true positive rate = true positives / all positives
- false positive rate = false positives / all negatives
- area under curve = average false positive rate
- random guessing leads to diagonal
- not well suited for detection:
  - #negatives not well defined
  - very difficult to get all true positives by random guessing



Typical output of a object detector

- List of image bounding boxes with score or confidence
- Looks something like this:

imageId x1 y1 x2 y2 score
imageId x1 y1 x2 y2 score
imageId x1 y1 x2 y2 score
.
.
.



0.9562

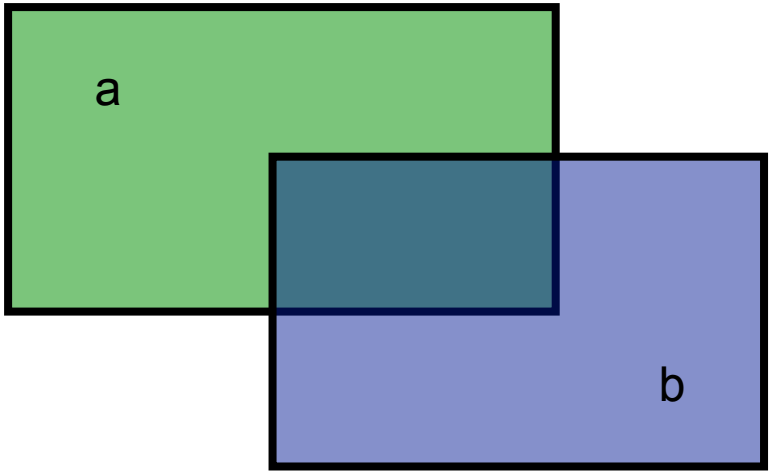
detection in red
ground truth in yellow

- Evaluation:
  - What is a correct detection?
  - How to compare those list stemming from different detectors?

What is a true positive?

- Detection has to be matched to ground truth
- If enough “overlap” according to a score -> true positive
- Double detections count as false positives
- Most popular criterion:

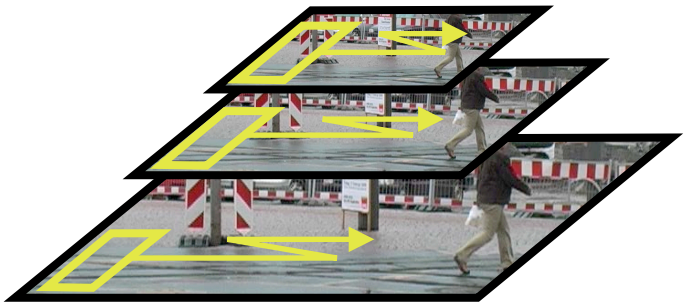
$$s(a,b) = \frac{a \cap b}{a \cup b} \geq 0.5$$



Sliding Window Methods - Overview

- Sliding Window Based People Detection:

Scan Image → Extract Feature Vector → Classify Feature Vector → Non-Maxima Suppression



'slide' detection window
over all positions & scales

Two Important Questions:
1) which feature vector
2) which classifier

Today:
- Viola Jones Face Detector
  - Haar Features
  - Boosting (cascade)
- HOG Pedestrian Detector
  - HOG descriptor
  - linear SVM

Rapid Object Detection Using a Boosted Cascade of Simple Features

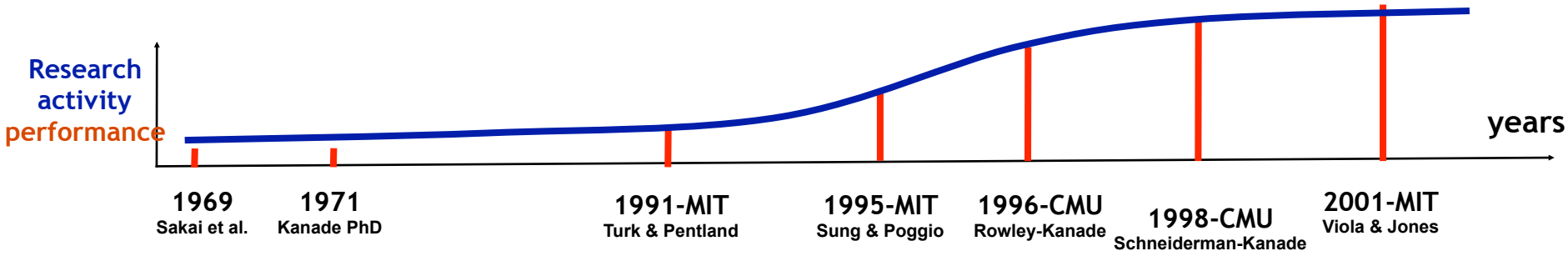
Paul Viola Michael J. Jones
Mitsubishi Electric Research Laboratories (MERL)
Cambridge, MA

Most of this work was done at Compaq CRL before the authors moved to MERL

Face Detection - Historical account

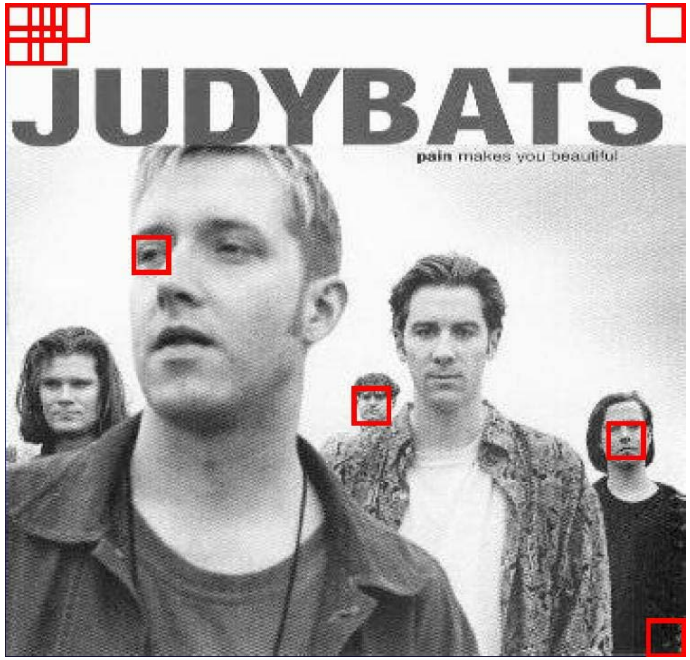
- 1991 – learning based approach
- 96-99 - most active years
  - Better machines,
  - Better understanding of the problems resulting in a diversity of approaches
  - Use of learning techniques & training and test data available
- 98 – two quite successful approaches
  - Excellent detection results but low efficiency
- 2001 – successful approach
  - Both, detection rate and efficiency

Literature: Yang, Kriegman, Ahuja, "Detecting Faces in Images: A Survey", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 24, No 1, 2oo2



Appearance-Based Methods
Sliding Window Search Strategy

- Search over Space and Scale



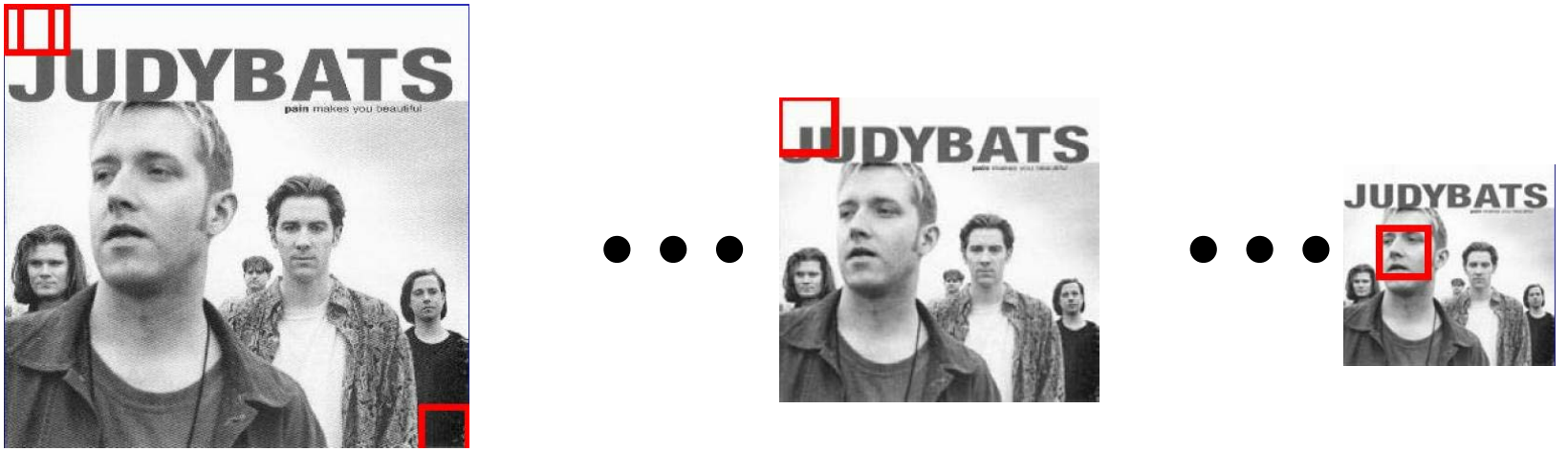
Scan an input image at one-pixel increments horizontally and vertically

• • •

Downsample the input image by a factor of 1.2 and continue to search

Appearance-Based Methods
Sliding Window Search Strategy

- Search over Space and Scale



Continue to downsample the input image and search until the image size is too small

Classifier is Learned from Labeled Data

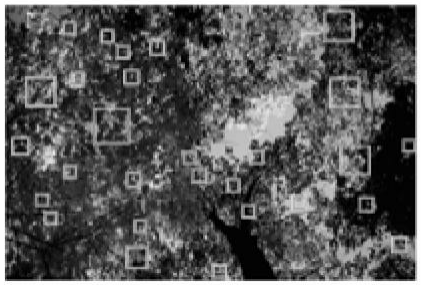
- Training Data
  - 5000 faces
    - All frontal
  - 10^8 non faces
  - Faces are normalized
    - Scale, translation
- Many variations
  - Across individuals
  - Illumination
  - Pose (rotation both in plane and out)

Viola and Jones, Robust object detection using a boosted cascade of simple features, CVPR 2001

Appearance-Based Methods
Training data

- Positive examples
  - Get as much variation as possible
  - Manually crop and normalize each face image into a standard size (e.g., 19 × 19 pixels)
  - Create virtual examples

- Negative examples
  - Use any image that do not contain faces
  - Explore large image subspace

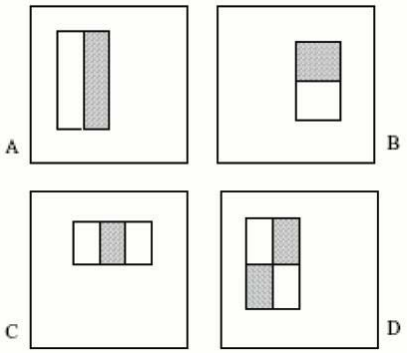


Appearance-Based Methods
AdaBoost based face detector [Viola & Jones 01]

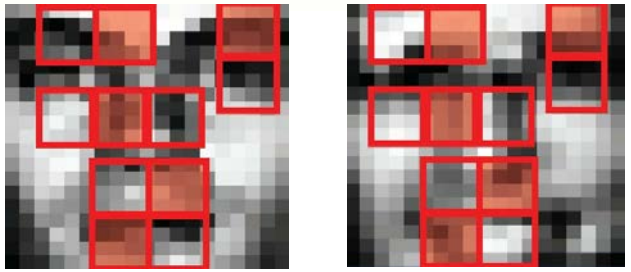
- Three ingredients:
  - Use simple features
  - Select important features and combine them into strong classifiers using a learning method - AdaBoost
    - Each feature can be used as a weak classifier
    - Sort them in the order of importance (performance) and combine
  - A cascade of classifiers
    - Combine the strong classifiers to do a difficult task
    - Focus on potential regions
    - Filter out the regions that most likely do not contain faces filter

Appearance-Based Methods
AdaBoost based face detector [Viola & Jones 01]

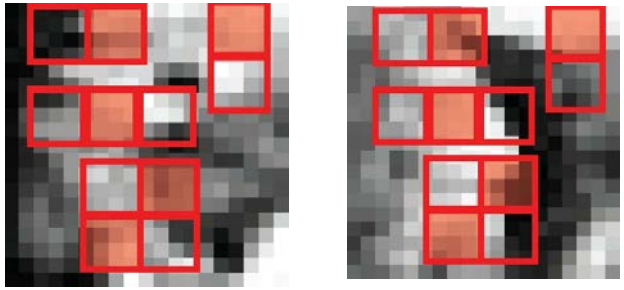
- Boxlet / rectangle features:
  - Compute the difference between sums of pixels within rectangular regions
  - Compute boxlets all over a pattern
  - Harr-like wavelets
  - Over-complete representation: lots of boxlet features
- Each feature f, is used as a weak classifier
- For each boxlet f compute responses on positive and negative examples and threshold them
  if f(x)>T face=1,
  otherwise face=0
- Find threshold for each feature so that most samples are classified correctly



Boxlet:
2-rectangle,
3-rectangle,
4-rectangle



Face examples

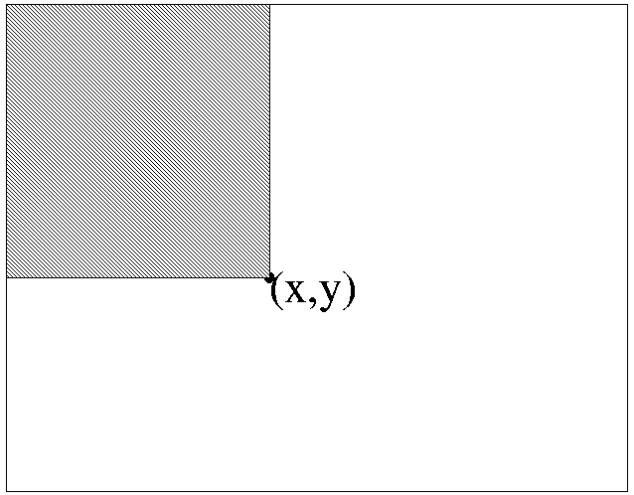


Non-Face examples

Integral Image

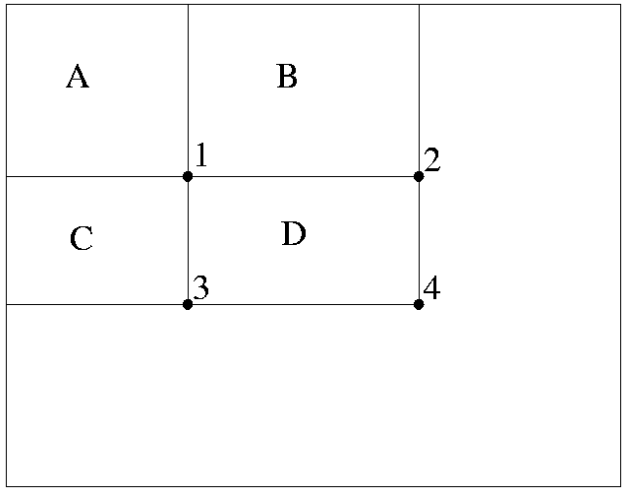
- Define the Integral Image

$$I'(x,y) = \sum_{\substack{x' \le x \\ y' \le y}} I(x',y')$$



- Any rectangular sum can be computed in constant time:

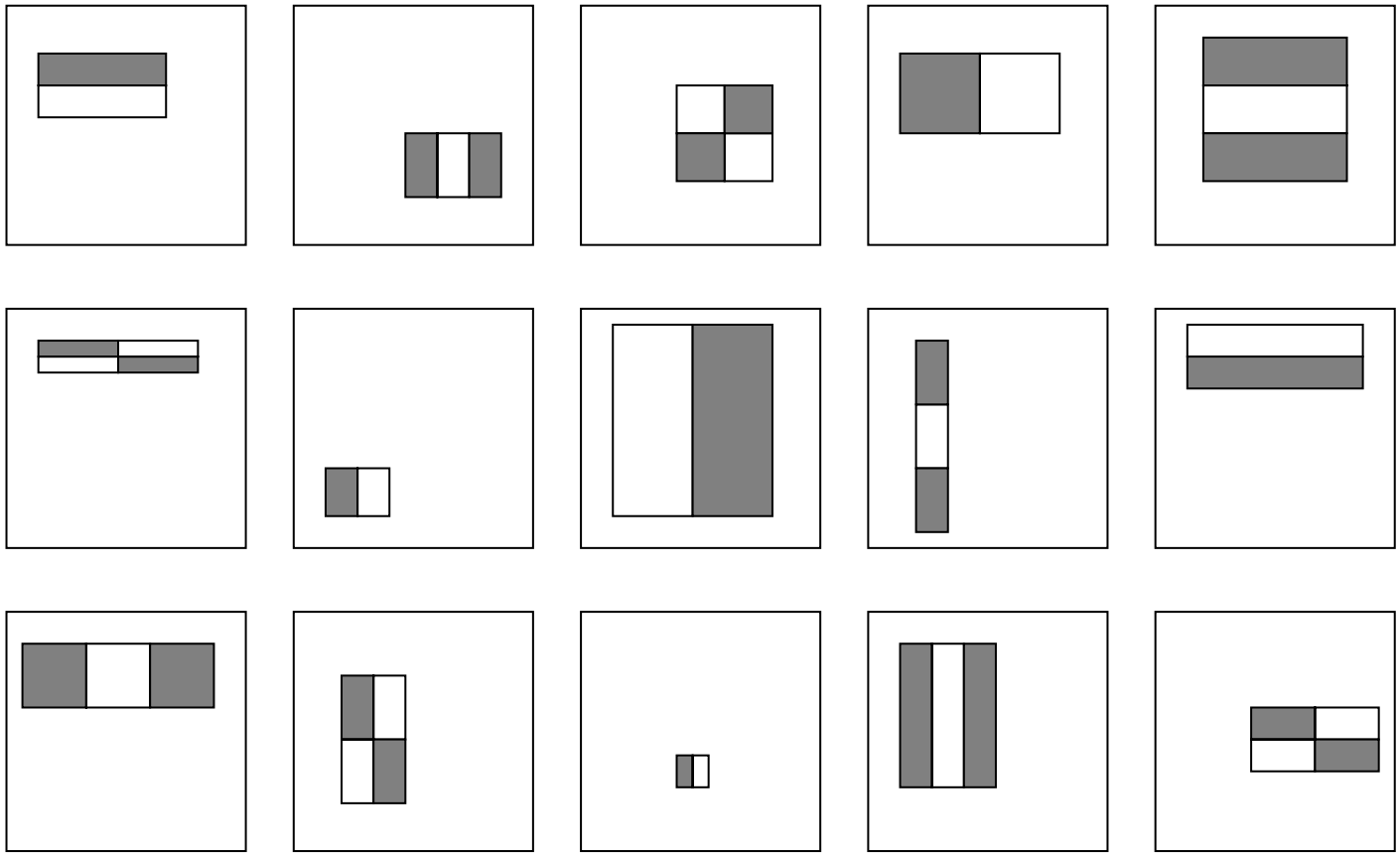
$$\begin{aligned} D &= 1 + 4 - (2 + 3) \\ &= A + (A + B + C + D) - (A + C + A + B) \\ &= D \end{aligned}$$



- Rectangle features can be computed as differences between rectangles

Viola and Jones, Robust object detection using a boosted cascade of simple features, CVPR 2001

Huge “Library” of Filters



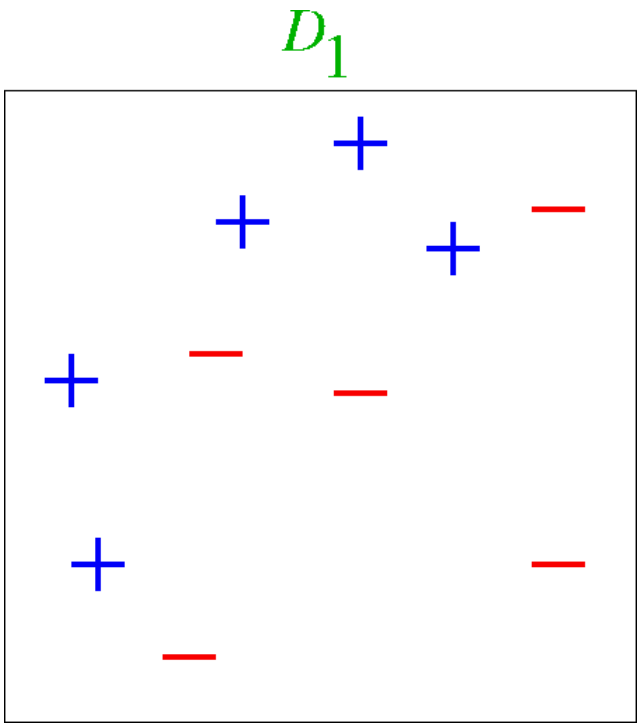
Viola and Jones, Robust object detection using a boosted cascade of simple features, CVPR 2001

Appearance-Based Methods
AdaBoost based face detector [Viola & Jones 01]

- Construct a strong classifier using single features (weak classifiers)
  1. For each classifier, estimate the probability of an error
  2. Choose the classifier, with the minimum error
  3. Update the weight:
     - increase the importance of examples where the classifier makes an error
     - decrease for correctly classified examples
  4. Go to step 1

Appearance-Based Methods
AdaBoost –toy example

Positive and negative examples

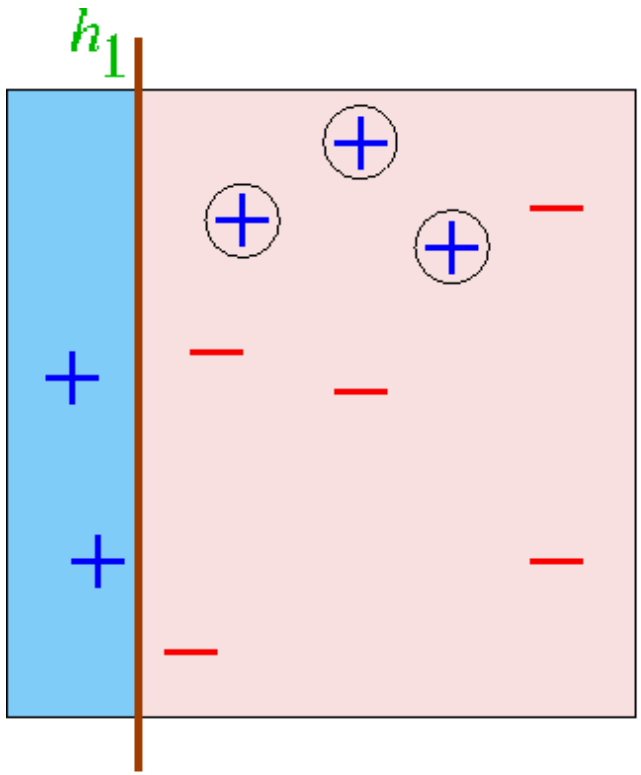


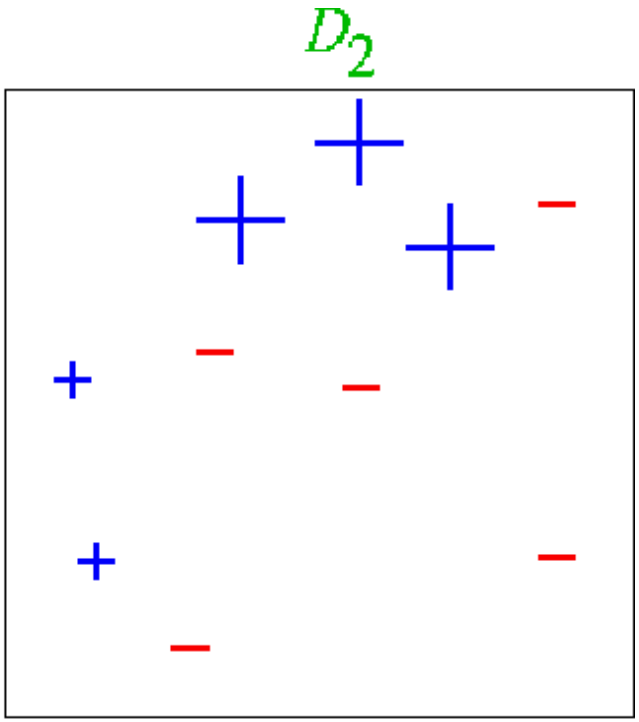
Weak classifier

h

Appearance-Based Methods
AdaBoost –toy example

- Iteration 1



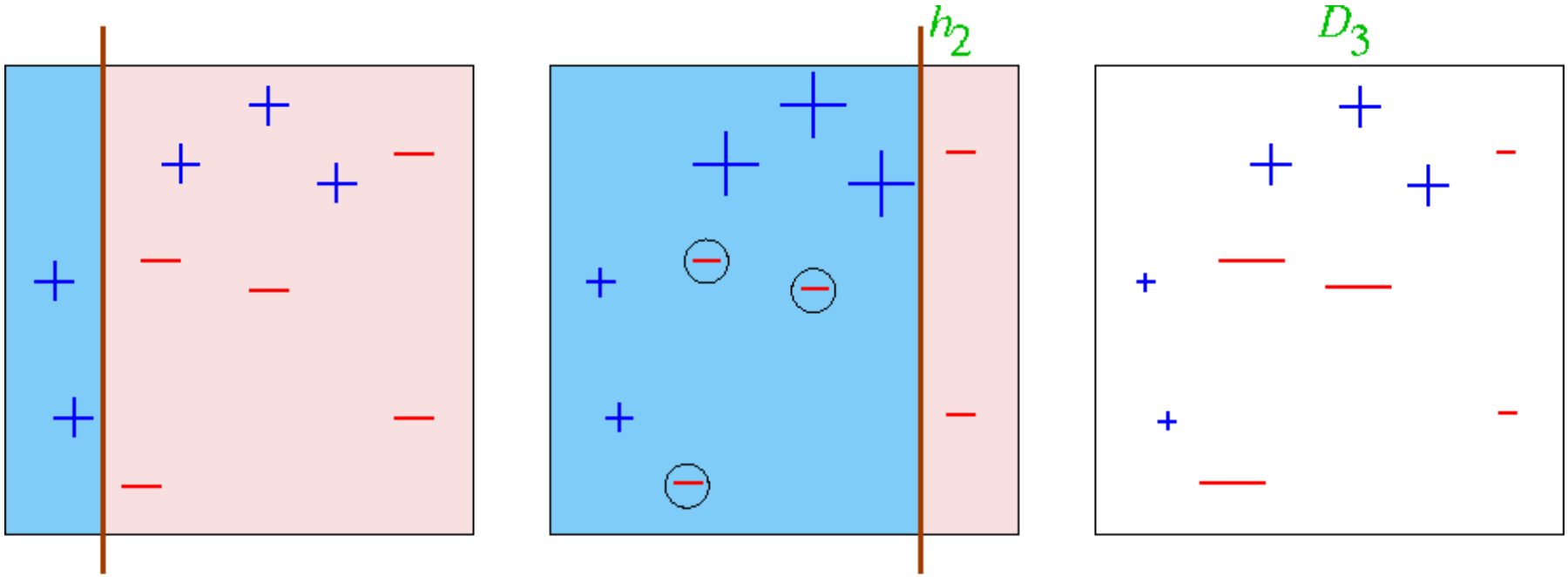


$\varepsilon_1 = 0.30$

$\alpha_1 = 0.42$

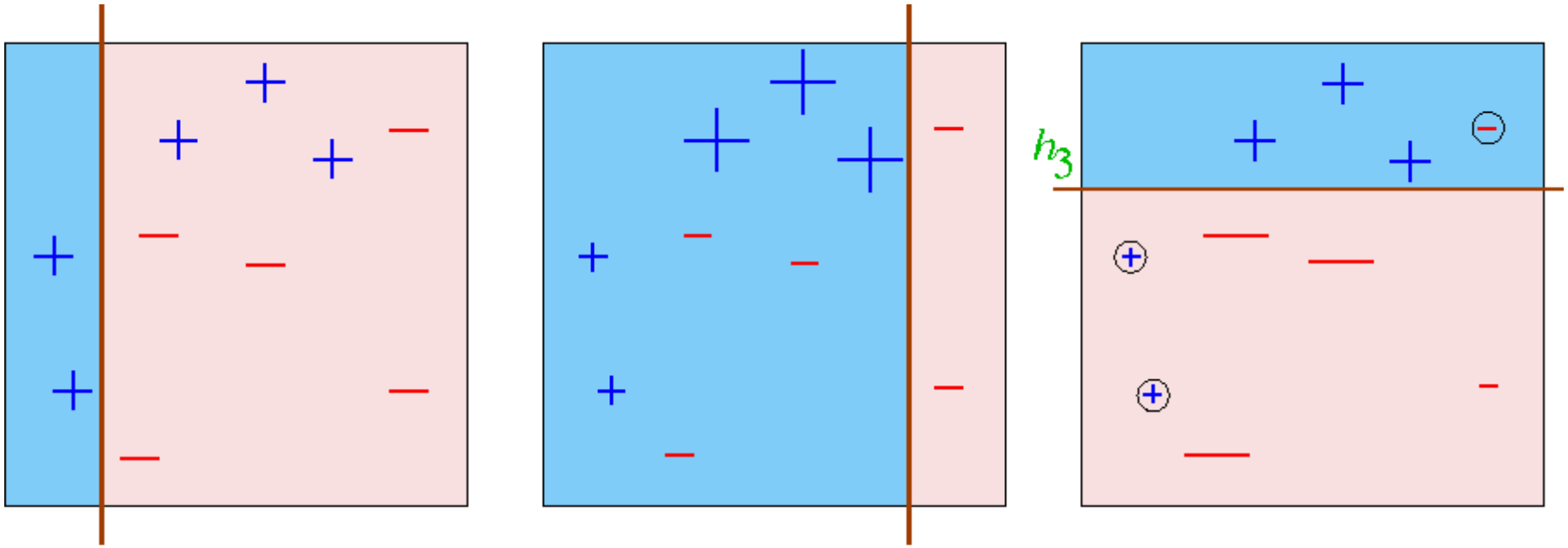
Appearance-Based Methods
AdaBoost –toy example

- Iteration 2



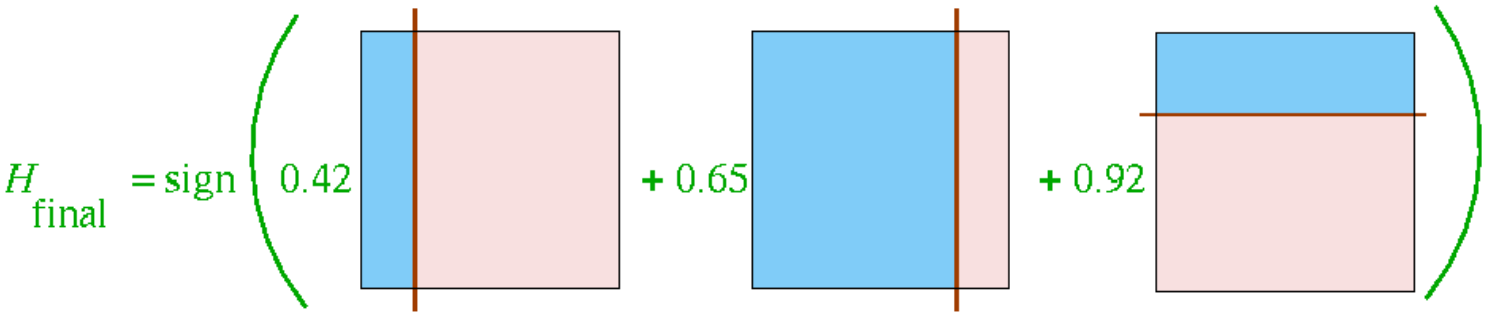
Appearance-Based Methods
AdaBoost –toy example

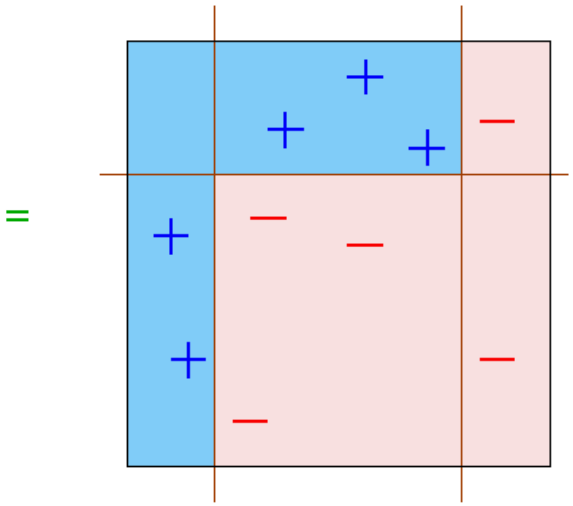
- Iteration 3



Appearance-Based Methods
AdaBoost –toy example

- Strong Classifier





AdaBoost (Freund & Shapire 95)

•Given examples (x_1, y_1), …, (x_N, y_N) where $y_i = 0,1$ for negative and positive examples respectively.
•Initialize weights $w_{t=1,i} = 1/N$

•For t=1, …, T

•Normalize the weights, $w_{t,i} = w_{t,i} / \sum_{j=1}^{N} w_{t,j}$

•Find a weak learner, i.e. a hypothesis, $h_t(x)$ with weighted error less than .5

•Calculate the error of h_t : $e_t = \sum w_{t,i} \mid h_t(x_i) - y_i \mid$

•Update the weights: $w_{t,i} = w_{t,i} B_t^{(1-d_i)}$ where $B_t = e_t / (1- e_t)$ and $d_i = 0$ if example x_i is classified correctly, $d_i = 1$ otherwise.

•The final strong classifier is

$$h(x) = \begin{cases} 1 & \text{if } \sum_{t=1}^{T} \alpha_t h_t(x) > 0.5 \sum_{t=1}^{T} \alpha_t \\ 0 & \text{otherwise} \end{cases}$$

where $\alpha_t = \log(1/ B_t)$

Viola and Jones, Robust object detection using a boosted cascade of simple features, CVPR 2001

AdaBoost for Efficient Feature Selection

- Our Features = Weak Classifiers
- For each round of boosting:
  - Evaluate each rectangle filter on each example
  - Sort examples by filter values
  - Select best threshold for each filter (min error)
    - Sorted list can be quickly scanned for the optimal threshold
  - Select best filter/threshold combination
  - Weight on this feature is a simple function of error rate
  - Reweight examples
  - (There are many tricks to make this more efficient.)

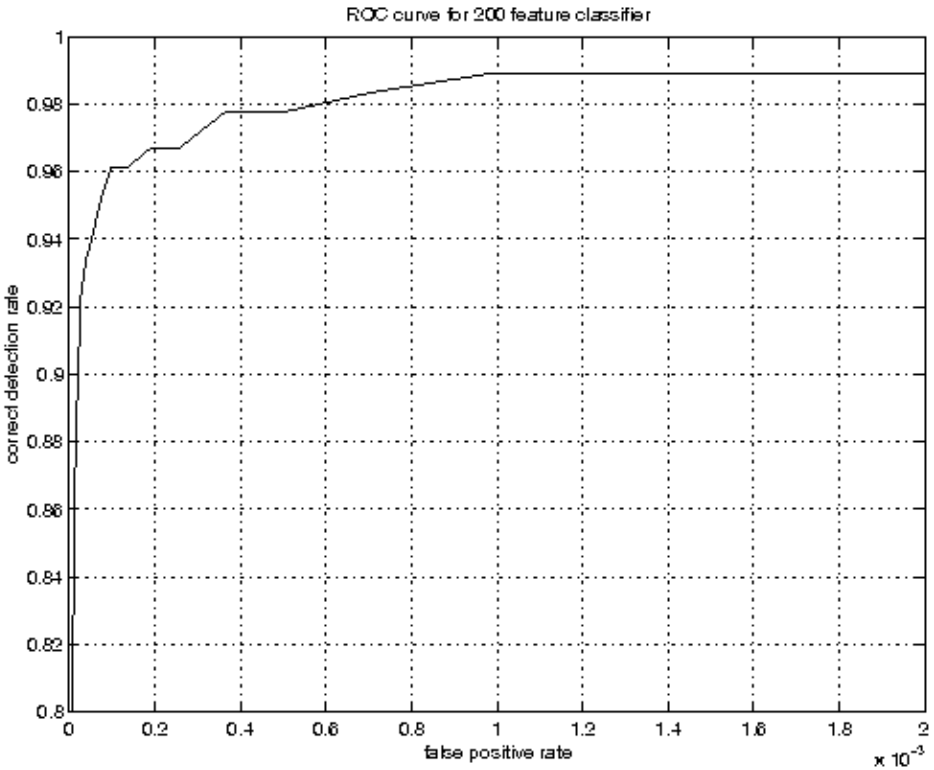
Viola and Jones, Robust object detection using a boosted cascade of simple features, CVPR 2001

Example Classifier for Face Detection

A classifier with 200 rectangle features was learned using AdaBoost

95% correct detection on test set with 1 in 14084 false positives.

Not quite competitive...

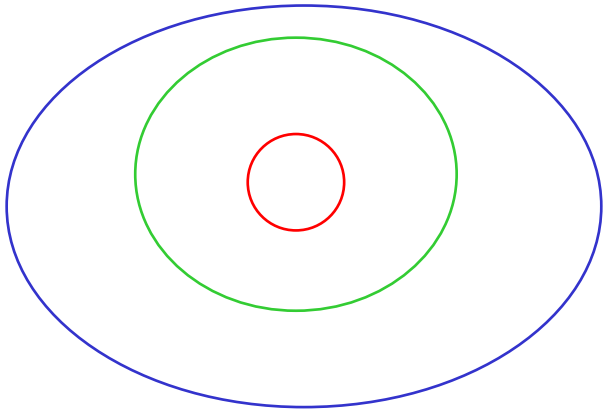


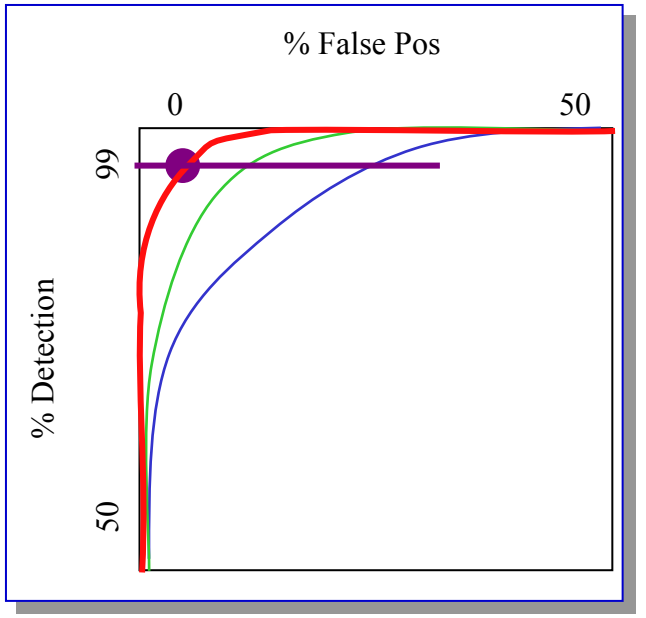
ROC curve for 200 feature classifier

Viola and Jones, Robust object detection using a boosted cascade of simple features, CVPR 2001

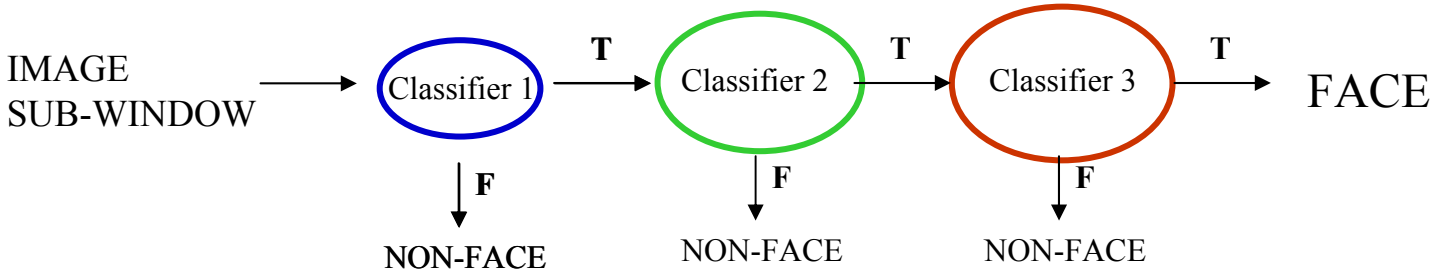
Trading Speed for Accuracy

- Given a nested set of classifier hypothesis classes



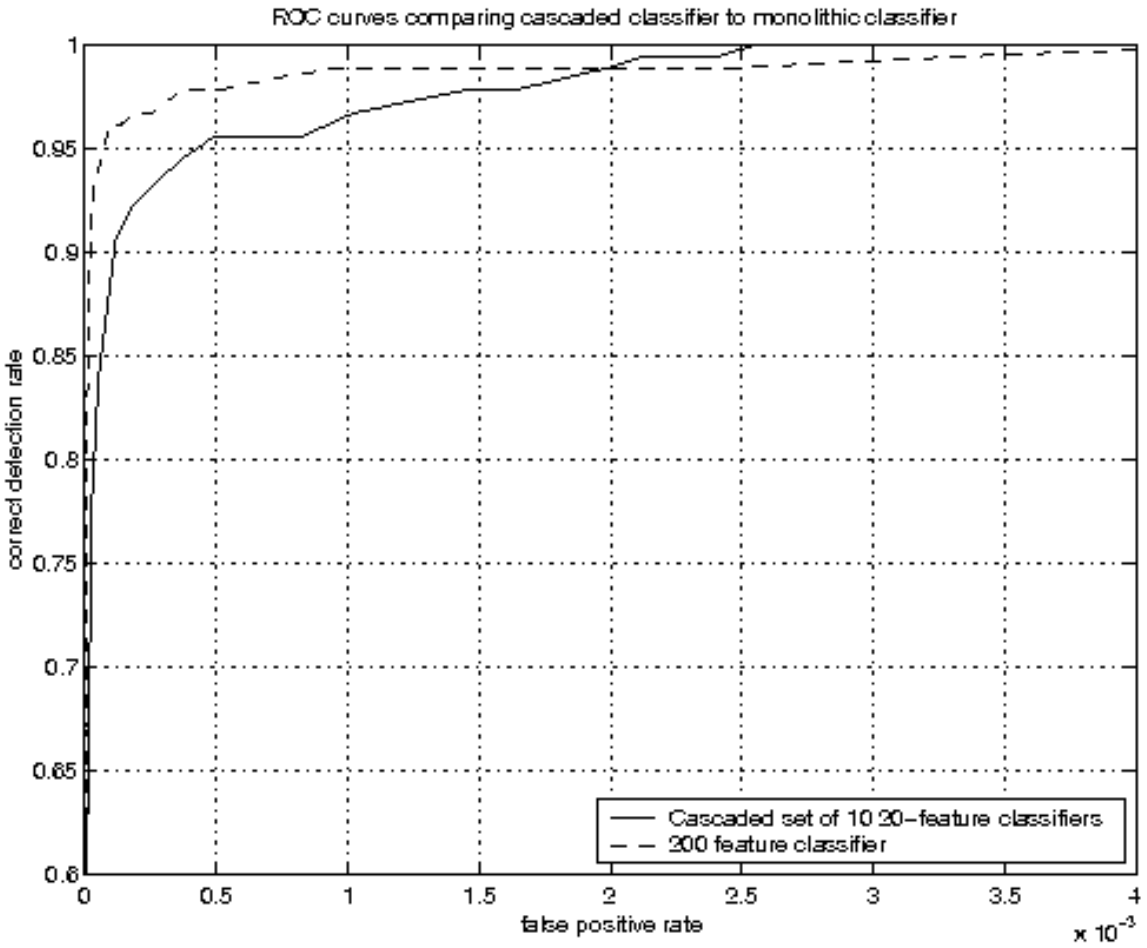


- Computational Risk Minimization



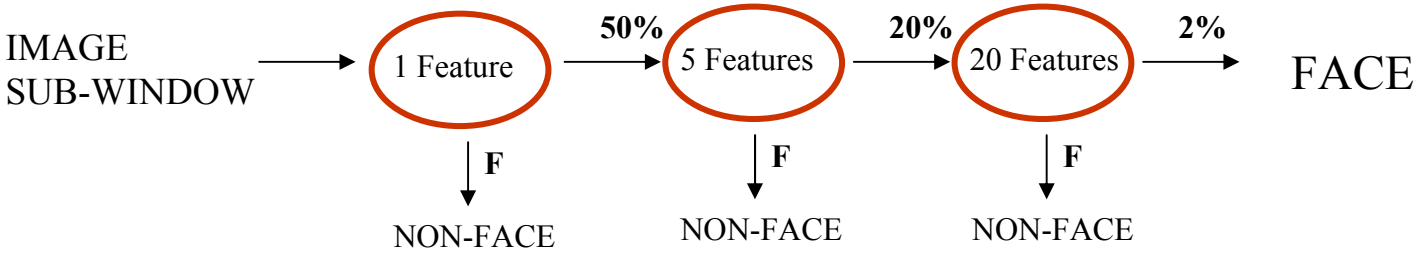
Viola and Jones, Robust object detection using a boosted cascade of simple features, CVPR 2001

Experiment: Simple Cascaded Classifier



Viola and Jones, Robust object detection using a boosted cascade of simple features, CVPR 2001

Cascaded Classifier



- A 1 feature classifier achieves 100% detection rate and about 50% false positive rate.
- A 5 feature classifier achieves 100% detection rate and 40% false positive rate (20% cumulative)
  - using data from previous stage.
- A 20 feature classifier achieve 100% detection rate with 10% false positive rate (2% cumulative)

Viola and Jones, Robust object detection using a boosted cascade of simple features, CVPR 2001

A Real-time Face Detection System

Training faces: 4916 face images (24 x 24 pixels) plus vertical flips for a total of 9832 faces

Training non-faces: 350 million sub-windows from 9500 non-face images

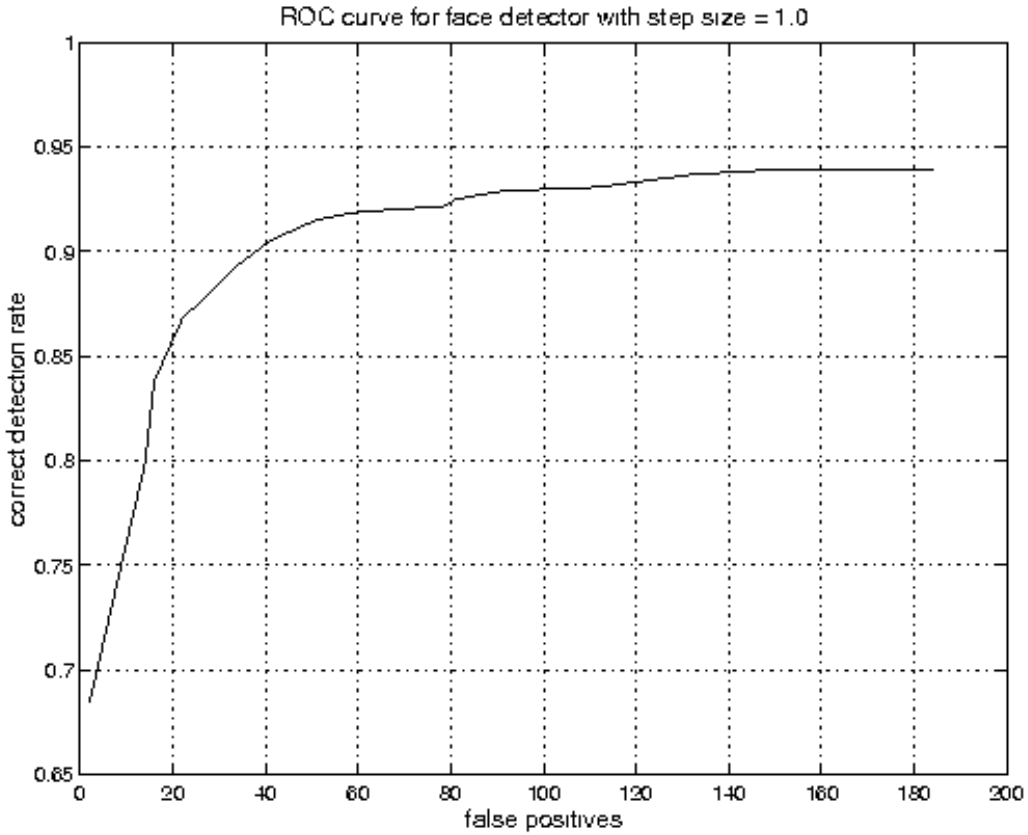
Final detector: 38 layer cascaded classifier
The number of features per layer was 1, 10, 25, 25, 50, 50, 50, 75, 100, …, 200, …

Final classifier contains 6061 features.

Viola and Jones, Robust object detection using a boosted cascade of simple features, CVPR 2001

Accuracy of Face Detector

Performance on MIT+CMU test set containing 130 images with 507 faces and about 75 million sub-windows.



Viola and Jones, Robust object detection using a boosted cascade of simple features, CVPR 2001

Comparison to Other Systems

Detector \ False Detections	10	31	50	65	78	95	110	167
Viola-Jones	76.1	88.4	91.4	92.0	92.1	92.9	93.1	93.9
Viola-Jones (voting)	81.1	89.7	92.1	93.1	93.1	93.2	93.7	93.7
Rowley-Baluja-Kanade	83.2	86.0				89.2		90.1
Schneiderman-Kanade				94.4				

Viola and Jones, Robust object detection using a boosted cascade of simple features, CVPR 2001

Speed of Face Detector

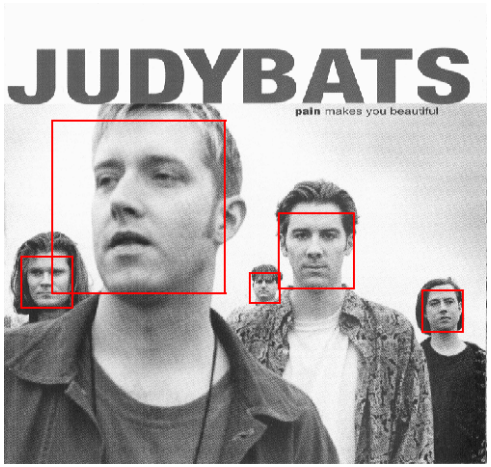
Speed is proportional to the average number of features computed per sub-window.

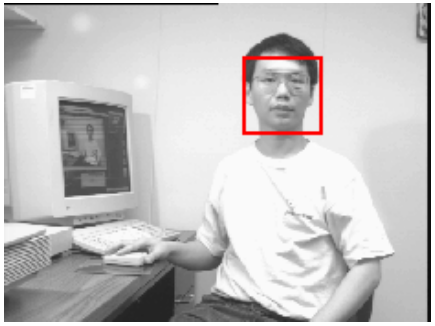
On the MIT+CMU test set, an average of 9 features out of a total of 6061 are computed per sub-window.

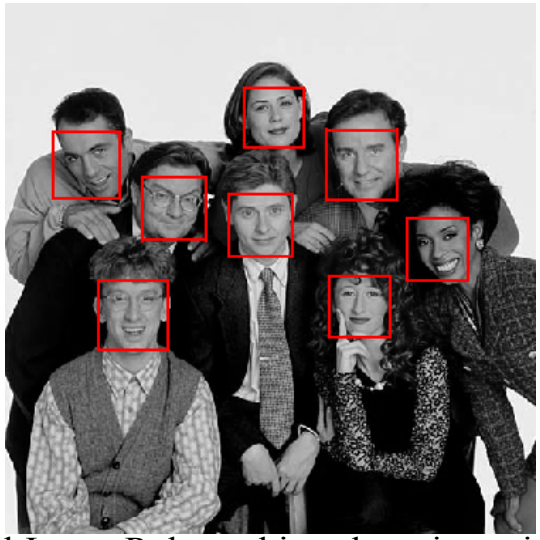
On a 700 Mhz Pentium III, a 384x288 pixel image takes about 0.067 seconds to process (15 fps).

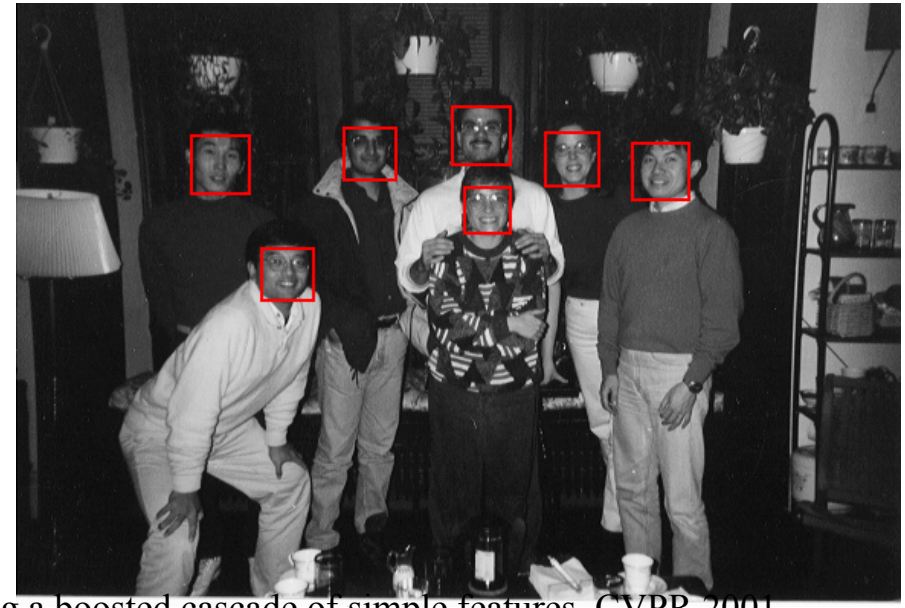
Roughly 15 times faster than Rowley-Baluja-Kanade and 600 times faster than Schneiderman-Kanade.

Output of Face Detector on Test Images



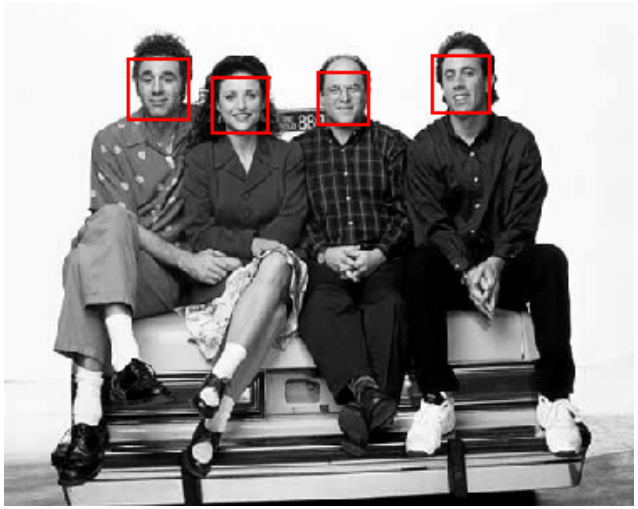


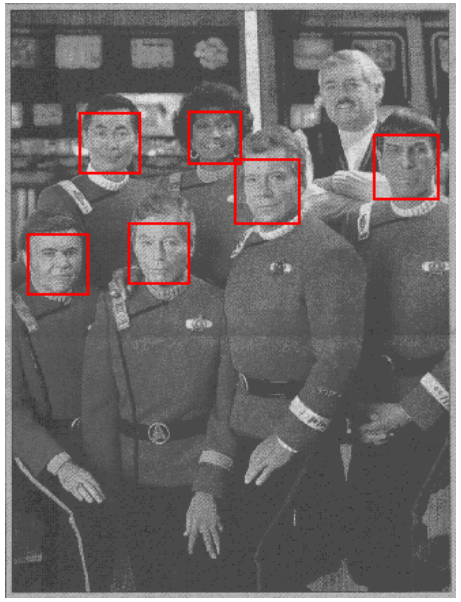


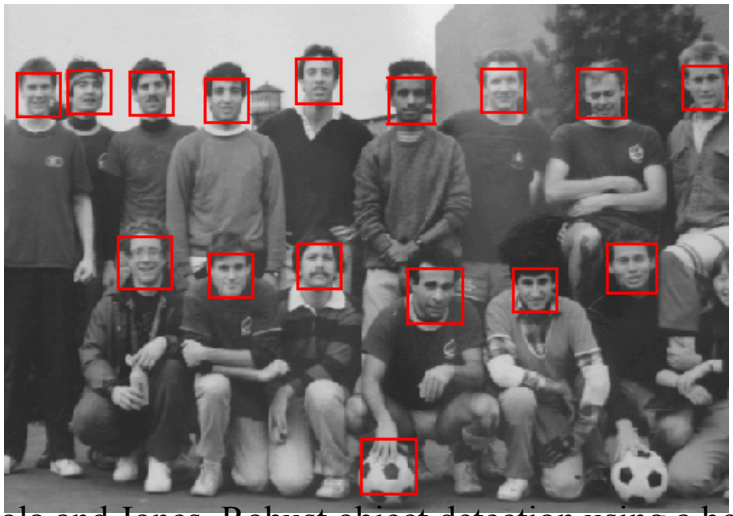


Viola and Jones, Robust object detection using a boosted cascade of simple features, CVPR 2001

More Examples







Viola and Jones, Robust object detection using a boosted cascade of simple features, CVPR 2001

Conclusions

- We [they] have developed the fastest known face detector for gray scale images
- Three contributions with broad applicability
  - Cascaded classifier yields rapid classification
  - AdaBoost as an extremely efficient feature selector
  - Rectangle Features + Integral Image can be used for rapid image analysis

Appearance-Based Methods
AdaBoost based face detector - Summary

- Three main components
  - Simple features, Adaboost for feature selection, Cascade of strong classifiers
- Pros
  - Fast and fairly robust; runs in real time; simple + easy to program
  - Flexible: can be combined with any classifier (neuronal net, C4.5, …)
- Cons
  - Requires lots of engineering work, many features, many training examples
  - Very time consuming in training stage (may take days in training)
  - Performance depends on data & weak classifier
  - AdaBoost can fail if
    - weak classifier is too complex (overfitting)
    - weak classifier is too weak ($\gamma_t \rightarrow 0$ too quickly),
      - underfitting
      - Low margins → overfitting
  - Empirically, AdaBoost seems especially susceptible to noise

Viola Jones Detector

- try it out
  - implementations available, e.g. opencv
  - works in real-time on reasonable image sizes

Overview

- Detection task
  - datasets
  - issues
  - evaluation
- Sliding Window Detection
- Viola Jones Face Detector
- **HOG Pedestrian Detector**

Goals & Applications of HOG

- Original Goal:
  - Detect and Localize people in Images and Videos
- Applications:
  - Images, films & multi-media analysis
  - Pedestrian detection for autonomous cars
  - Visual surveillance, behavior analysis

Difficulties of People / Object Detection

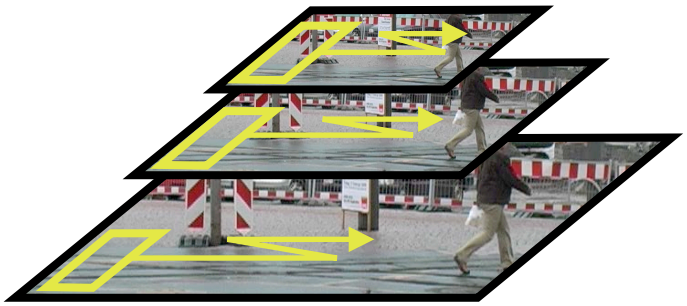
- Some of the Difficulties
  - Wide variety of articulated poses
  - Variable appearance and clothing
  - Complex backgrounds
  - Unconstrained illumination
  - Occlusions, different scales
  - Videos sequences involves motion of the subject, the camera and the objects in the background

- Main assumption for HOG: upright fully visible people

Sliding Window Methods - Overview

- Sliding Window Based People Detection:

Scan Image → Extract Feature Vector → Classify Feature Vector → Non-Maxima Suppression



'slide' detection window
over all positions & scales

Two Important Questions:
1) which feature vector
2) which classifier

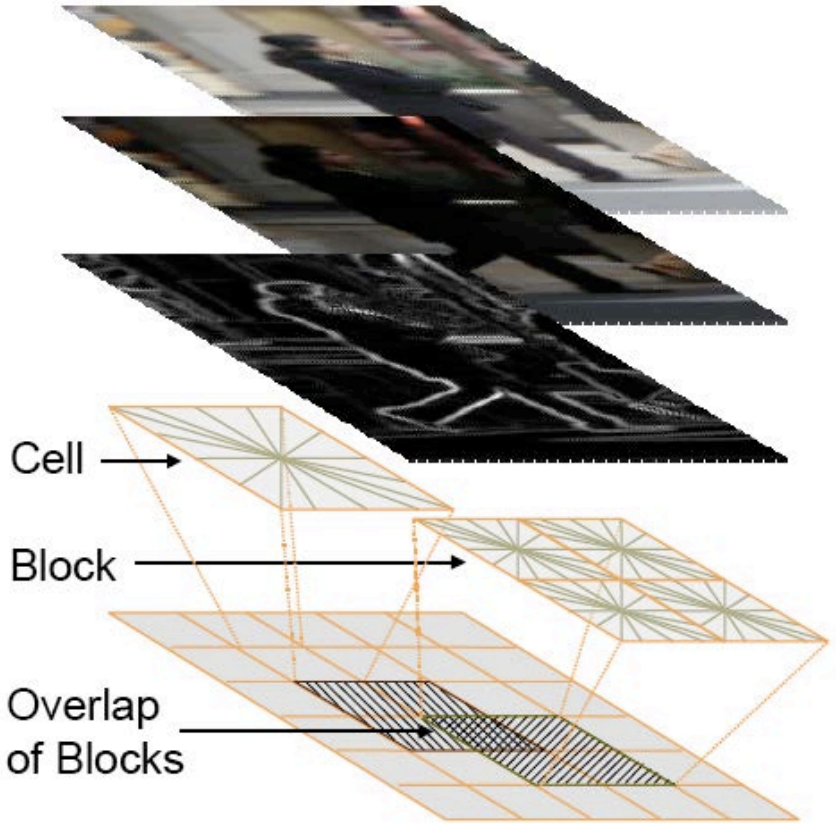
Today:
- Viola Jones Face Detector
  - Haar Features
  - Boosting (cascade)
- HOG Pedestrian Detector
  - HOG descriptor
  - linear SVM

HOG: Static Feature Extraction

Input Image

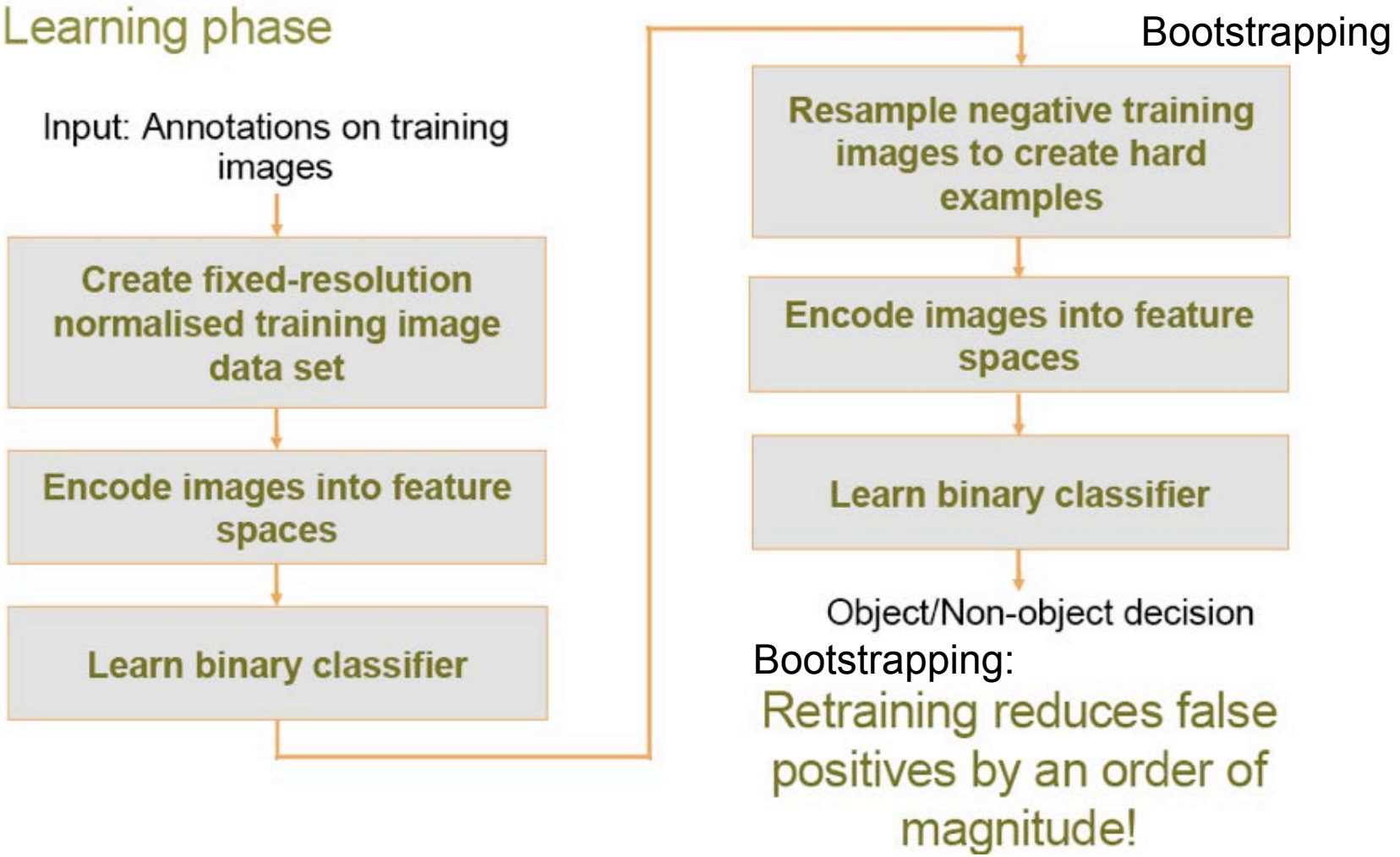
Detection Window

- Normalise gamma
- Compute gradients
- Weighted vote in spatial & orientation cells
- Contrast normalise over overlapping spatial cells
- Collect HOGs over detection window
- Linear SVM

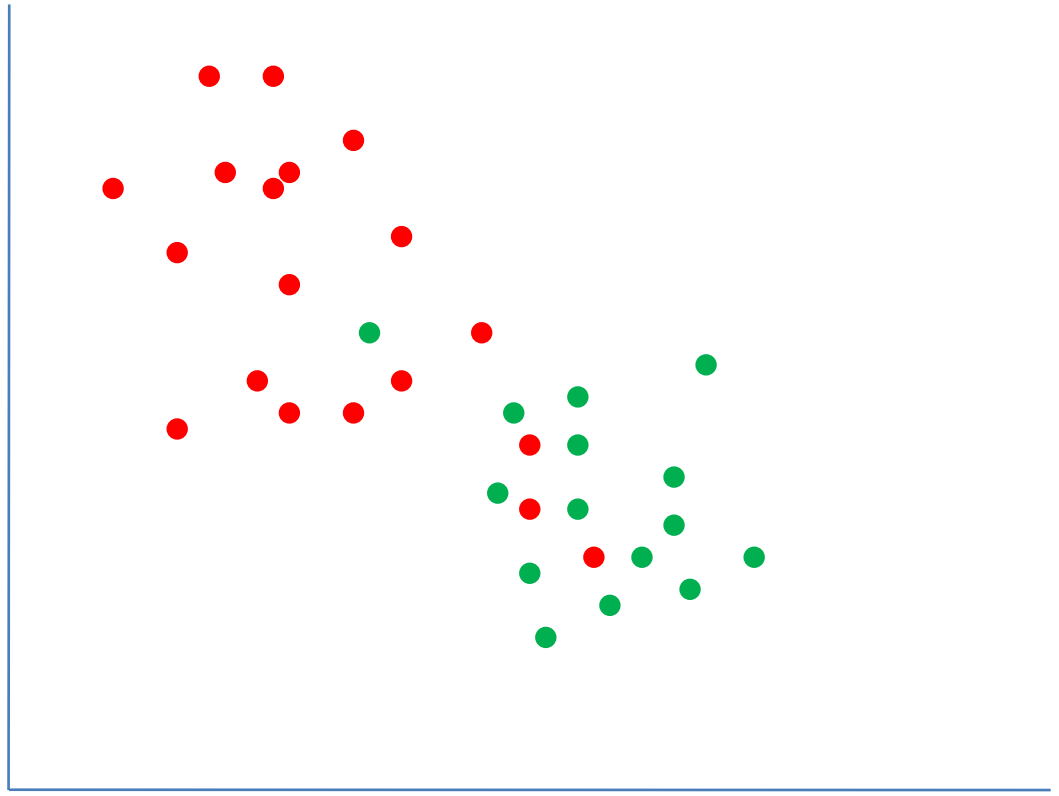


Feature vector f = [..., ..., ...]

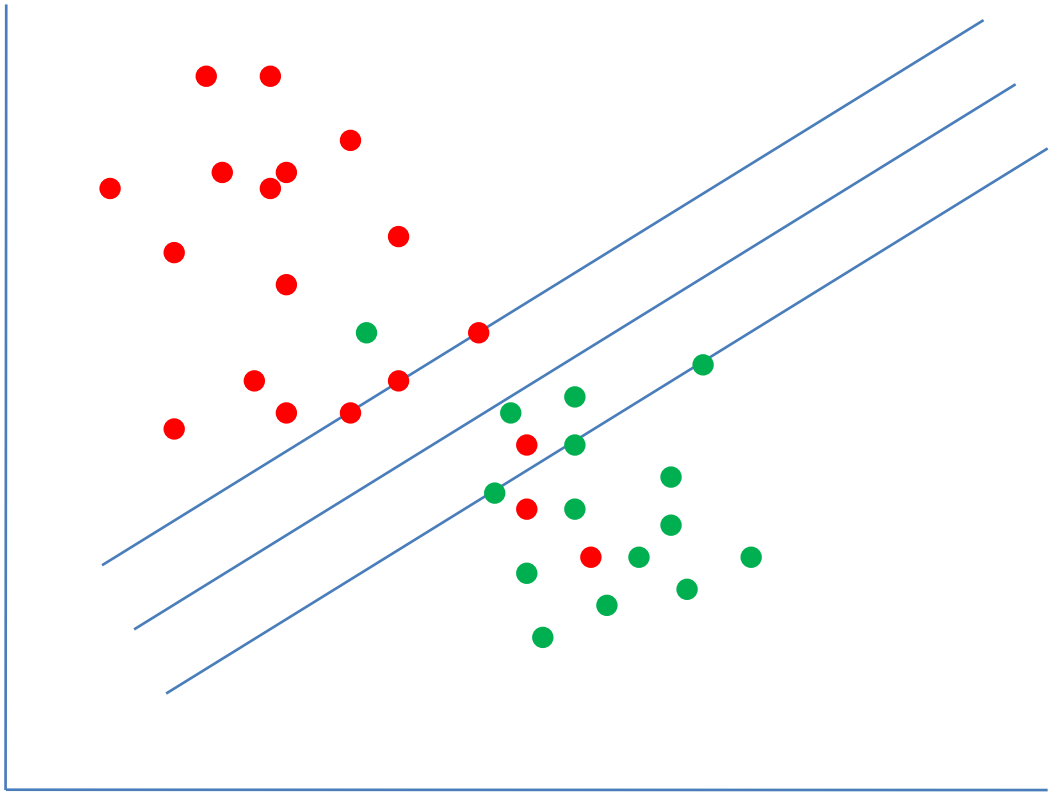
Overview of Learning



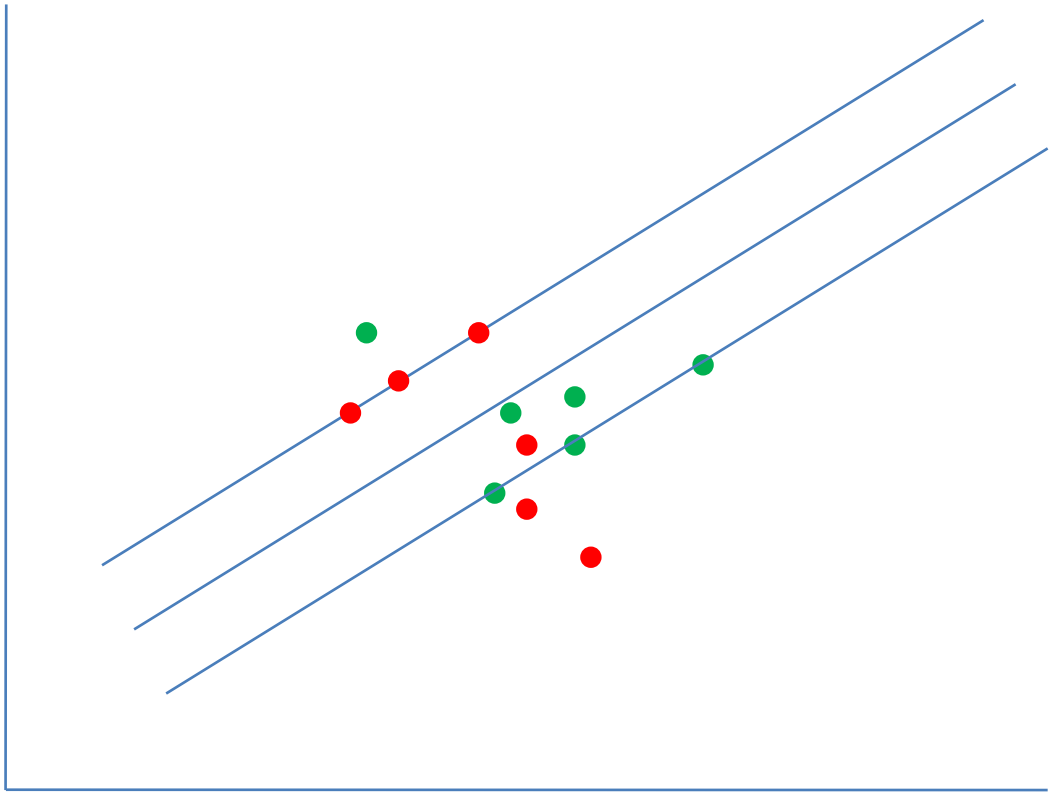
Data Mining Hard Negatives



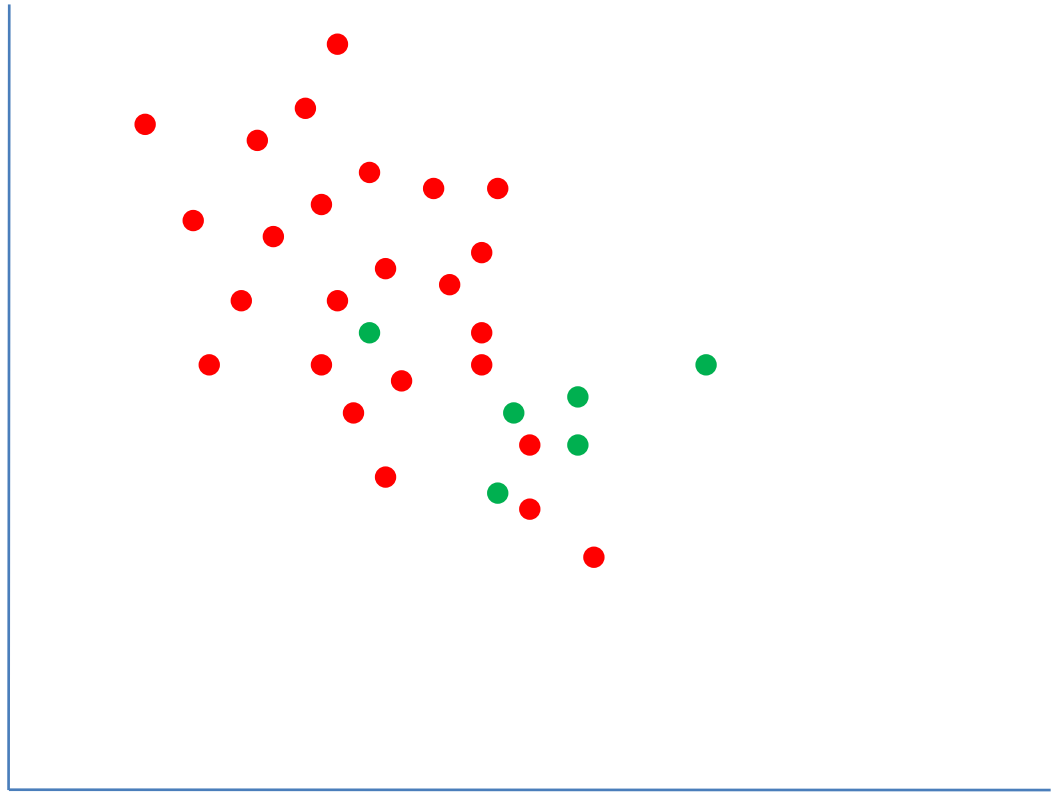
Data Mining Hard Negatives



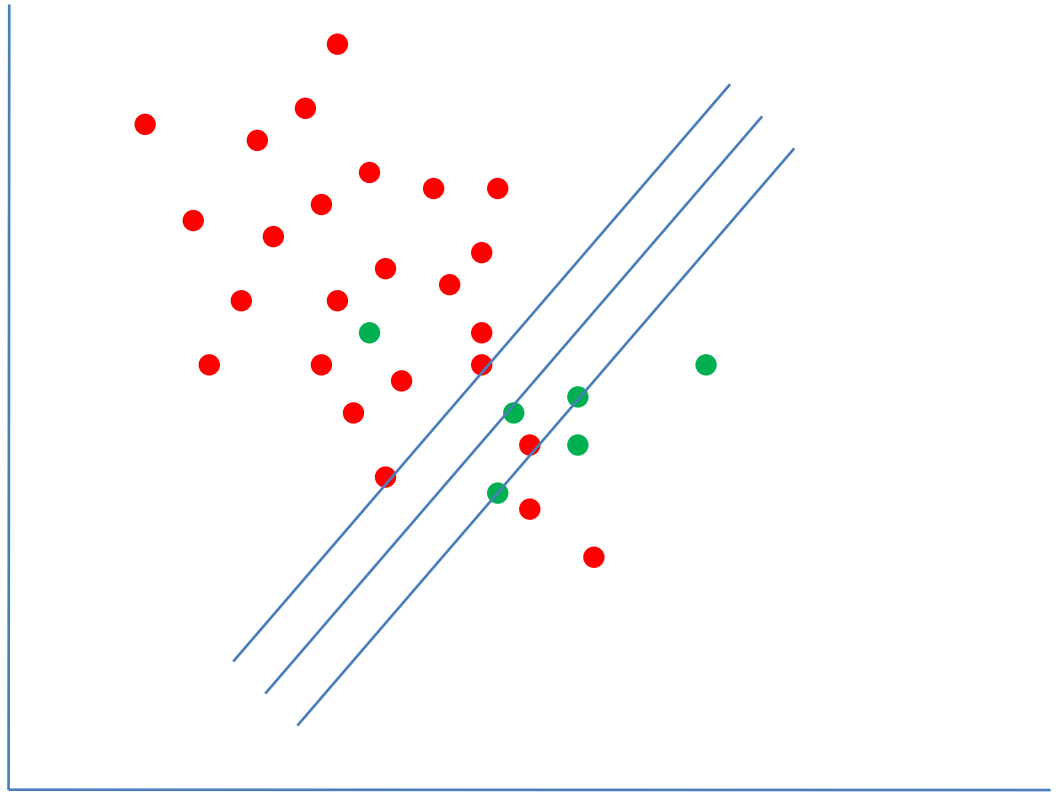
Data Mining Hard Negatives



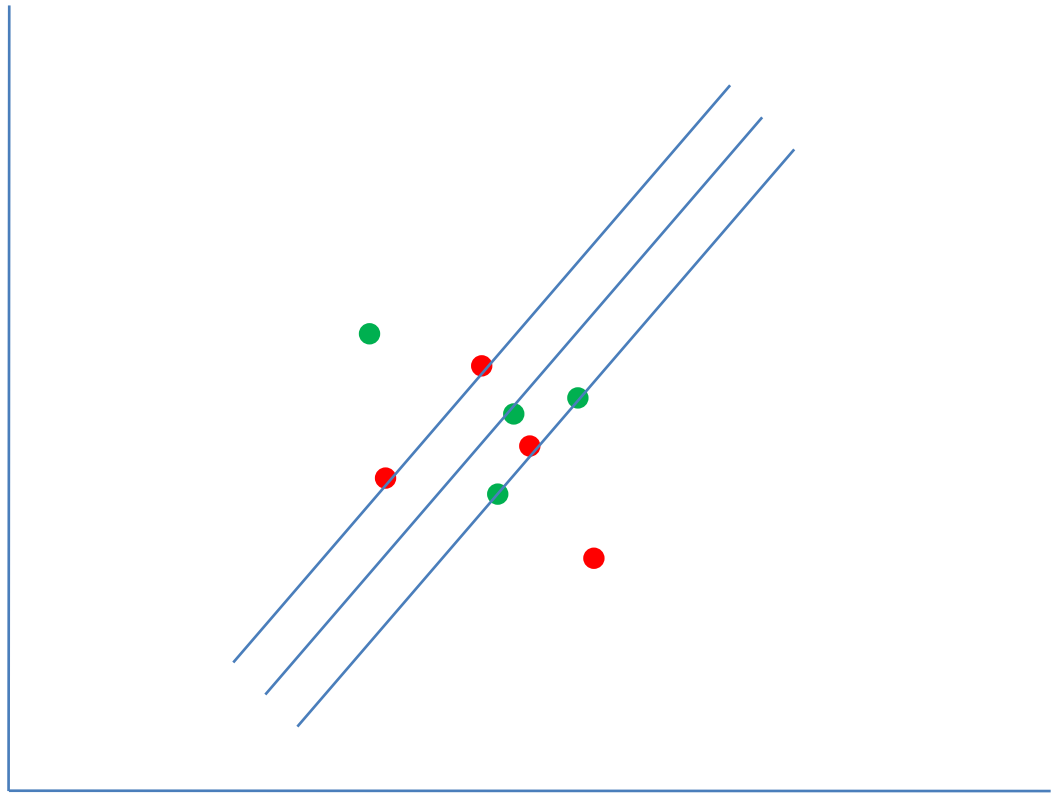
Data Mining Hard Negatives



Data Mining Hard Negatives



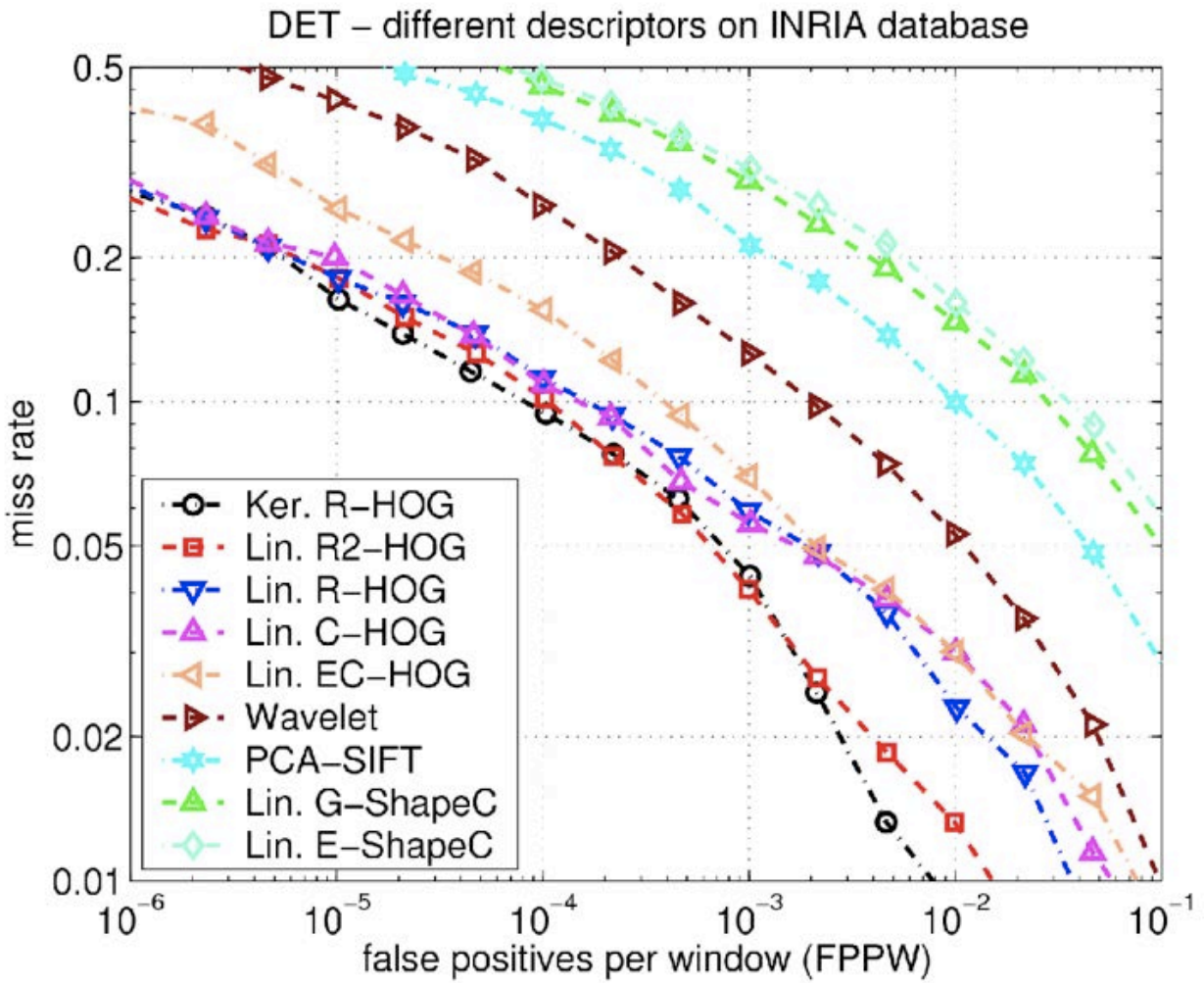
Data Mining Hard Negatives



Evaluation Data Sets

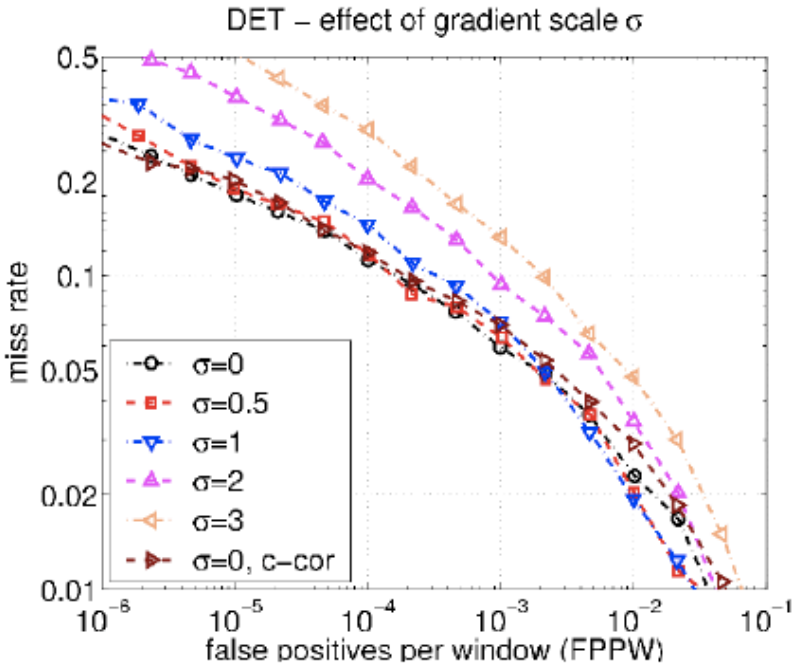
MIT pedestrian database	INRIA person database
	
Train: 507 positive windows Negative data unavailable	Train: 1208 positive windows 1218 negative images
Test: 200 positive windows Negative data unavailable	Test: 566 positive windows 453 negative images
Overall 709 annotations+ reflections	Overall 1774 annotations+ reflections

Performance on INRIA Dataset



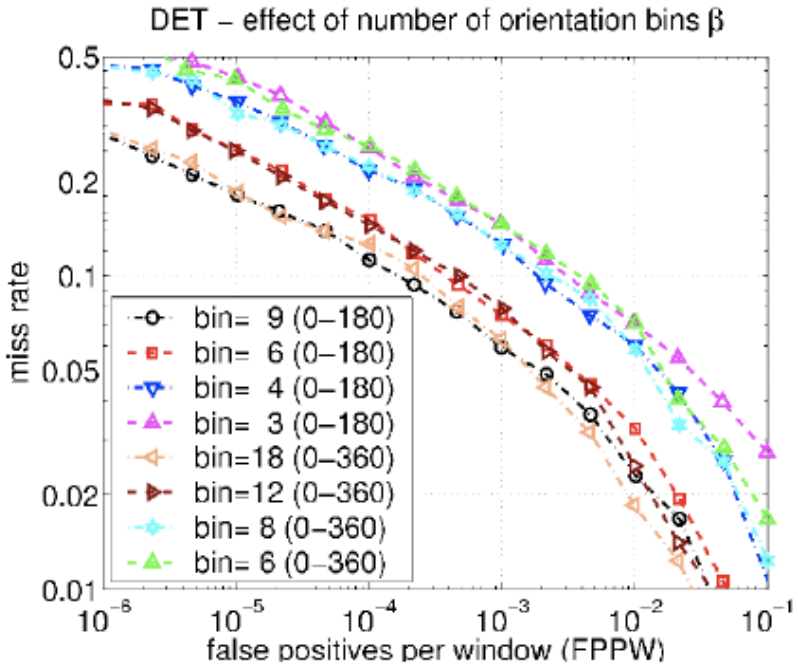
Effects of Parameters

Gradient smoothing, σ



Reducing gradient scale from 3 to 0 decreases false positives by 10 times

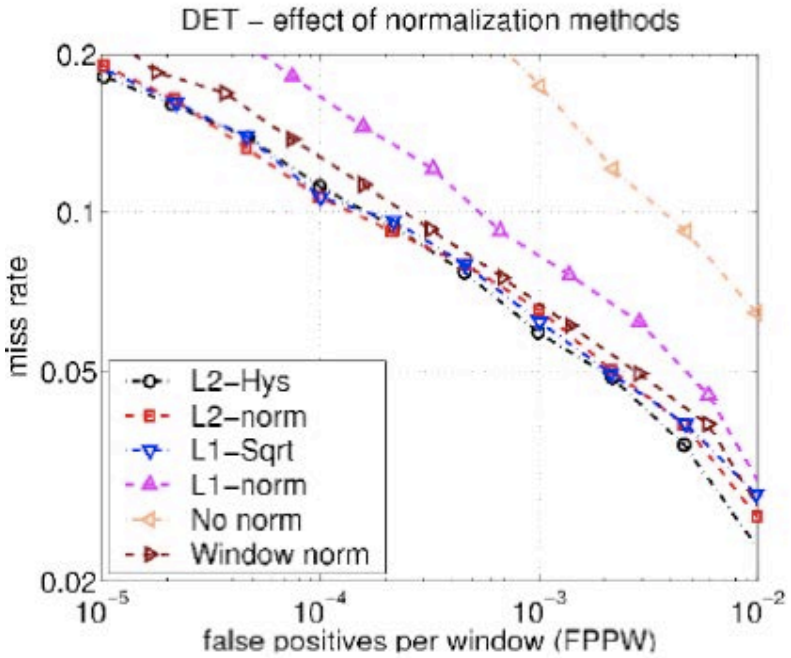
Orientation bins, β



Increasing orientation bins from 4 to 9 decreases false positives by 10 times

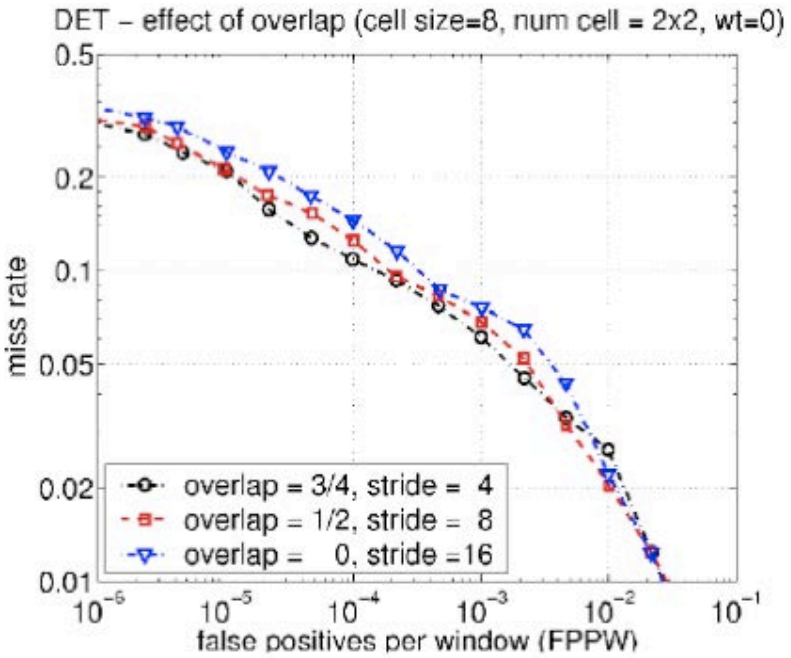
Normalization Method & Block Overlap

Normalisation method



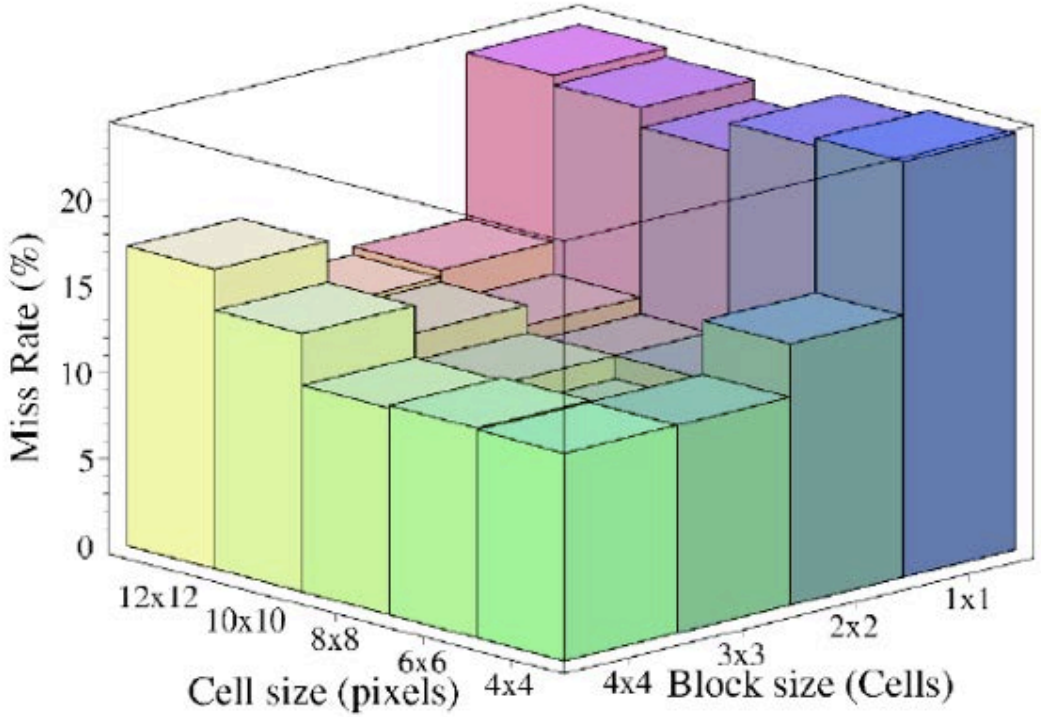
Strong local normalisation is essential

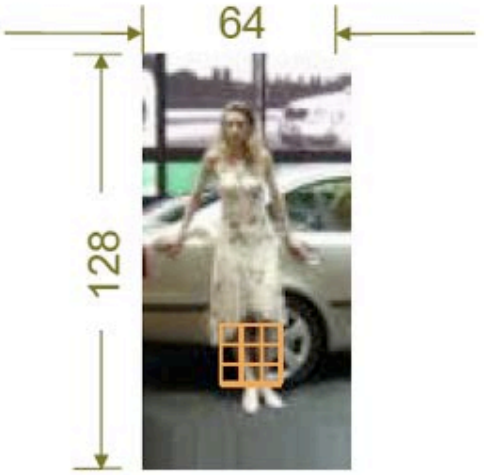
Block overlap



Overlapping blocks improve performance, but descriptor size increases

Effect of Block and Cell Size

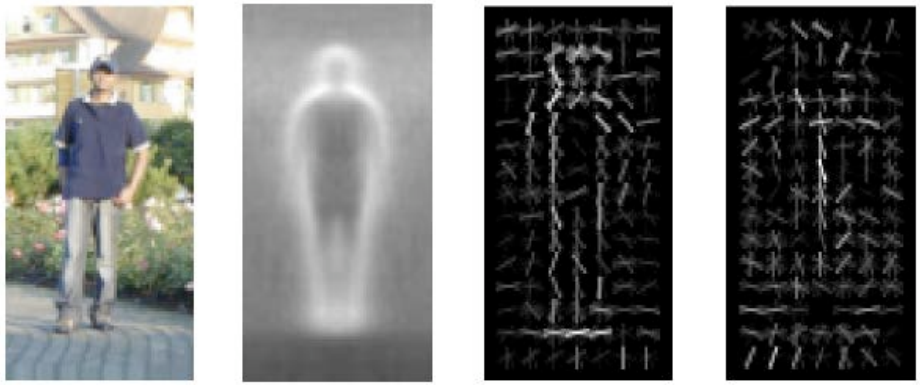




Trade off between need for local spatial invariance and need for finer spatial resolution

Descriptor Cues

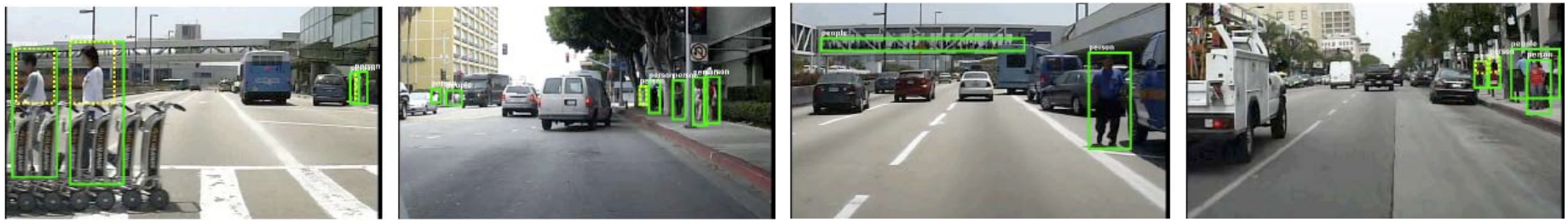
- Most Important Cues:
  - Head, shoulder, leg silhouettes
  - vertical gradients inside a person are counted as negative
  - overlapping blocks just outside the contour are most important
- "Local Context" Use:
  - Note that Dalal & Triggs obtain best performance by including quite substantial context/background around the person:



Input example | Average gradients | Weighted pos wts | Weighted neg wts

Pedestrian Detection: A New Benckmark

- Features of the new Pedestrian Dataset:
  - 11h of 'normal' driving in urban environment (greater LA area)
  - annotation:
    - 250'000 frames (~137 min) annotated with 350'000 labeled bounding boxes of 2'300 unique pedestrians
    - occlusion annotation: 2 bounding boxes for entire pedestrian & visible region
    - difference between 'single person' and 'groups of people'



[Dollar,Wojek,Perona,Schiele@CVPR-09]

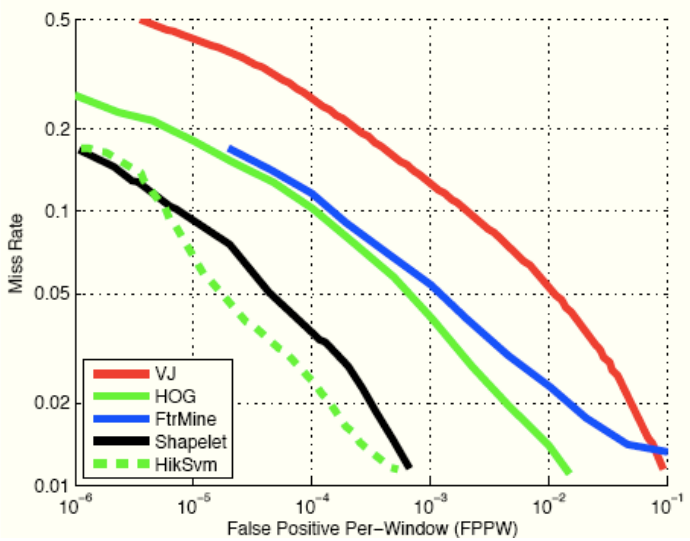
Comparison to Existing Datasets

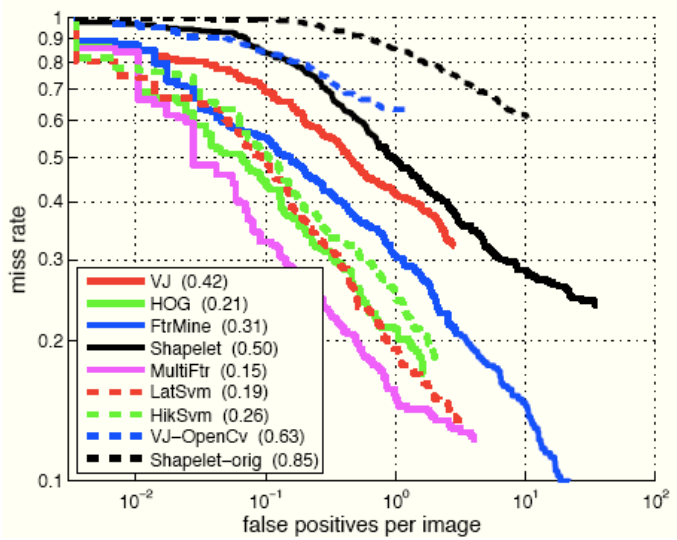
- New Pedestrian Dataset [Dollar,Wojek,Perona,Schiele@CVPR-09]
  - 1 - 2 orders of magnitude larger than any existing dataset
  - new features: temporal correlation of pedestrians, occlusion labeling, ...

	Training			Testing			Height			Properties					
	# pedestrians	# neg. images	# pos. images	# pedestrians	# neg. images	# pos. images	10% quantile	median	90% quantile	color images	per-image ev.	no selec. bias	video seqs.	temporal corr.	occ. labels
MIT[25]	924	–	–	–	–	–	128	128	128	✓					
USC-A[38]	–	–	–	313	–	205	70	98	133		✓				
USC-B[38]	–	–	–	271	–	54	63	90	126		✓				
USC-C[39]	–	–	–	232	–	100	74	108	145		✓	✓			
CVC[13]	1000	6175†	–	–	–	–	46	83	164	✓		✓			
TUD-det[1]	400	–	400	311	–	250	133	218	278	✓	✓				
INRIA[4]	1208	1218	614	566	453	288	139	279	456	✓					
DC[23]	2.4k	15k†	–	1.6k	10k†	–	36	36	36			✓			
ETH[7]	2388	–	499	12k	–	1804	50	90	189	✓	✓	✓	✓		
Proposed	192k	61k	67k	155k	56k	65k	27	48	97	✓	✓	✓	✓	✓	✓

Evaluation Criteria

- Different Approaches:
  - False Positive Per Window (FFPW) vs.
  - False Positives Per Image (FFPI)
- comparison of different algorithms on INRIA-person:
  - ordering of algorithms largely differs between FFPW and FFPI !

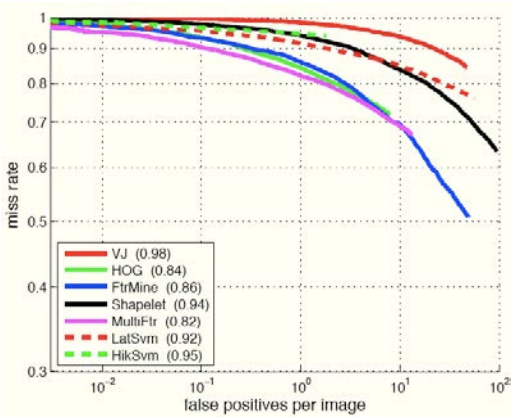




Comparison of Algorithms

- 7 Algorithms tested (FPPI: False-Positives-per-Image):

overall performance

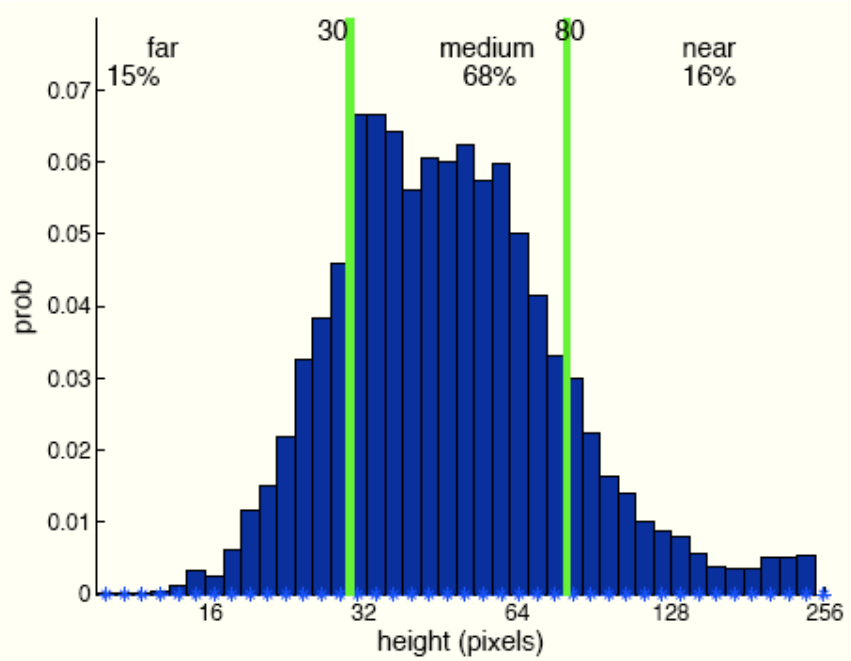


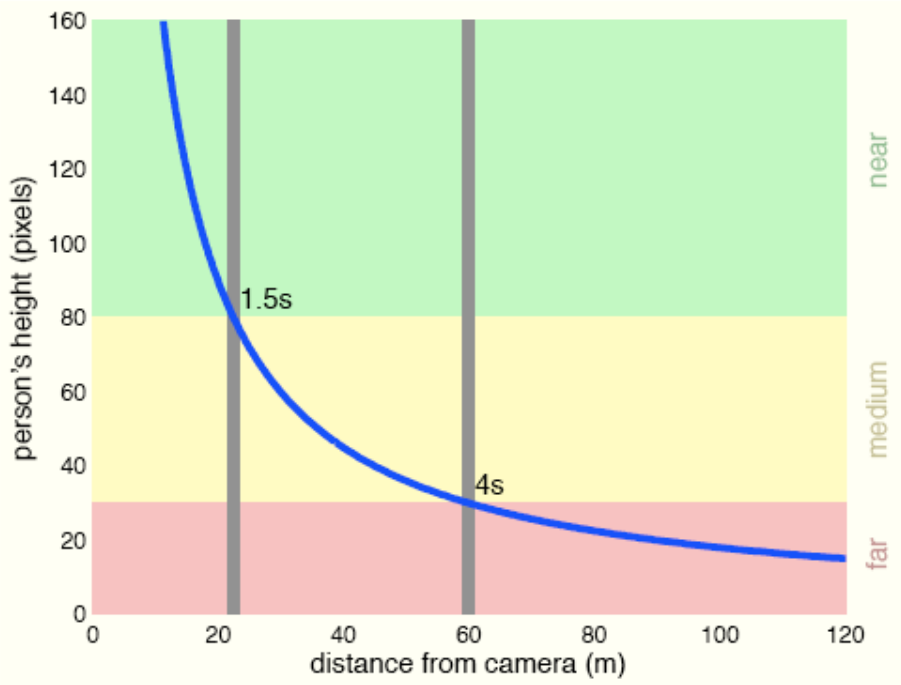
	low-level features	classifier	original implementation	trained on INRIA data	per-image evaluation	sec / frame	model height (in pixels)	scale stride	publication
VJ[35]	Haar	AdaBoost		✓		7.0	100	1.05	'04
HOG[4]	HOG	linear SVM	✓	✓		13.3	100	1.05	'05
FtrMine[6]	gen. Haar	AdaBoost	✓	✓		45	100	1.20	'07
Shapelet[28]	gradients	AdaBoost		✓		60.1	100	1.05	'07
MultiFtr[37]	HOG+Haar	AdaBoost	✓	✓	✓	18.9	100	1.05	'08
LatSvm[10]	HOG	latent SVM	✓		✓	6.3	80	1.05	'08
HikSvm[20]	HOG-like	HIK SVM	✓	✓		140	100	1.20	'08

[Dollar,Wojek,Perona,Schiele@CVPR-09]

Distribution of Pedestrian Sizes

- Differentiation between different sizes:
  - far : pedestrians < 30 pixels large
  - medium : 30 - 80 pixels large
  - near : > 80 pixels large

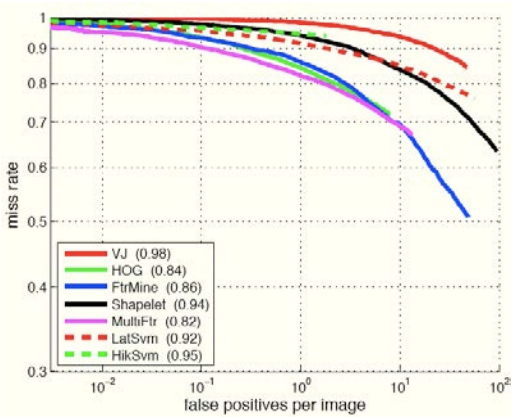




Comparison of Algorithms

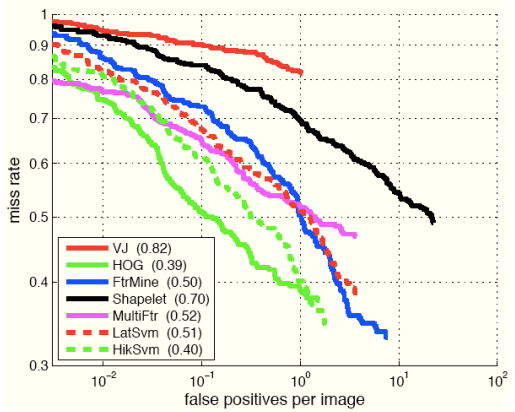
- 7 Algorithms tested (FPPI: False-Positives-per-Image):

overall performance

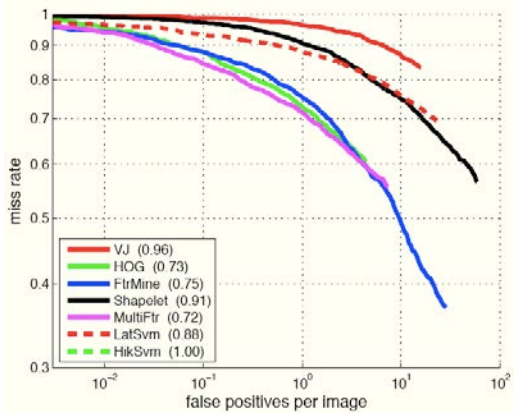


	low-level features	classifier	original implementation	trained on INRIA data	per-image evaluation	sec / frame	model height (in pixels)	scale stride	publication
VJ[35]	Haar	AdaBoost		✓		7.0	100	1.05	'04
HOG[4]	HOG	linear SVM	✓	✓		13.3	100	1.05	'05
FtrMine[6]	gen. Haar	AdaBoost	✓	✓		45	100	1.20	'07
Shapelet[28]	gradients	AdaBoost		✓		60.1	100	1.05	'07
MultiFtr[37]	HOG+Haar	AdaBoost	✓	✓	✓	18.9	100	1.05	'08
LatSvm[10]	HOG	latent SVM	✓		✓	6.3	80	1.05	'08
HikSvm[20]	HOG-like	HIK SVM	✓	✓		140	100	1.20	'08

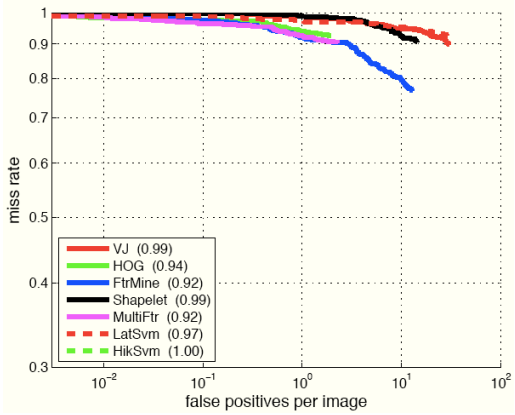
near scale



medium scale



far scale



[Dollar,Wojek,Perona,Schiele@CVPR-09]

Remaining Failure Cases (for INRIA-people dataset)

- Missing Detections: [Wojek,Schiele@DAGM-08]

(a) Unusual articulation (b) Difficult contrast (c) Occlusion (d) Person carrying goods

- 149 missing detections:
  - 44 difficult contrast & backgrounds
  - 43 occlusion & carried bags
  - 37 unusual articulations
  - 18 over- / underexposure
  - 7 wrong scale (too small/large)

Remaining Failure Cases (for INRIA-people dataset)

- False Positives

[Wojek,Schiele@DAGM-08]

(e) Detection on parts (f) Too large scale (g) Detection on vertical structures (h) Cluttered background (i) Missing annotation

- 149 false positives:
  - 54 vertical structures / street signs
  - 31 cluttered background
  - 28 too small scale (body parts)
  - 24 too large scale detections
  - 12 people that are not annotated :-)

Application to other Classes (e.g. PASCAL'o5)

Parameter Settings

- Most HOG parameters are stable across different classes

- Parameters that change:
  - Gamma compression
  - Normalization methods
  - Signed/un-signed gradients

Results e.g. from PASCAL VOC 2oo6

	Person	Car	Motorbike	Bicycle	Bus	Sheep	Horse	Cow	Cat	Dog
Cambridge	0.030	0.254	0.178	0.249	0.138	0.131	0.091	0.149	0.151	**0.118**
ENSMP	-	0.398	-	-	-	-	-	0.159	-	-
HOG	**0.164**	**0.444**	**0.390**	0.414	0.117	**0.251**	-	0.212	-	-
Laptev= HOG+ Ada-boost	0.114	-	0.318	**0.440**	-	-	**0.140**	0.224	-	-
TUD	0.074	-	0.153	-	-	-	-	-	-	-
TKK	0.039	0.222	0.265	0.303	**0.169**	0.227	0.137	**0.252**	**0.160**	0.113

HOG outperformed other methods for 4 out of 10 classes

Its adaBoost variant outperformed other methods for 2 out of 10 classes

Overview

- Detection task
  - datasets
  - issues
  - evaluation
- Sliding Window Detection
- Viola Jones Face Detector
- HOG Pedestrian Detector