



HA NOI UNIVERSITY OF SCIENCE AND TECHNOLOGY  
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

# Computer Vision

Chapter 7 (part 2): Object detection

Nguyễn Thị Oanh  
[oanhnt@soict.hust.edu.vn](mailto:oanhnt@soict.hust.edu.vn)

# Computer Vision

Chapter 7 (part 2): Object detection

Nguyễn Thị Oanh  
[oanhnt@soict.hust.edu.vn](mailto:oanhnt@soict.hust.edu.vn)



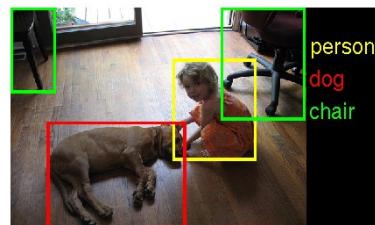
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

# Contents

- Window-based generic object detection: basic pipeline
  - Boosting classifiers
- Face detection as case study
  - Haar features
  - Adaboost
- SVM + HOG for human detection as case study
- Object proposals
- [DPM]
- Evaluation

# Object Detection

- **Problem:** Detecting and localizing generic objects from various categories, such as cars, people, etc.
- **Challenges:**
  - Illumination,
  - viewpoint,
  - deformations,
  - Intra-class variability



---

# Window-based generic object detection

Basic pipeline

---

## Basic Pipeline

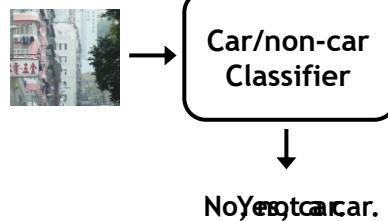
- Build/train object model
  - Choose a representation
  - Learn or fit parameters of model / classifier
- Generate candidates in new image
- Score the candidates

## Basic Pipeline

- **Build/train object model**

- Choose a representation
- Learn or fit parameters of model / classifier

Given the representation,  
train a binary classifier



## Basic Pipeline

- **Generate candidates** in new image
- **Score** the candidates



## Window-based models: Window size?

- Slide through the image and check if there is an object at every location



YES!! Person match found

## Window-based models: Window size ?

- But what if we were looking for buses?



No bus found!

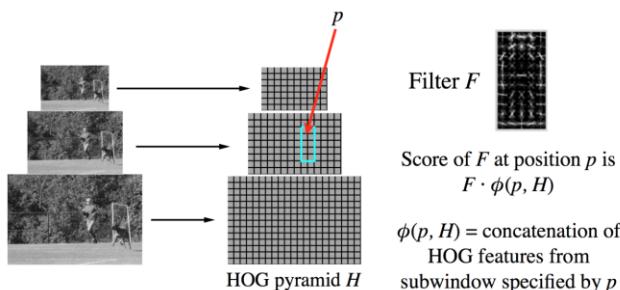
- We will never find the object if we don't choose our window size wisely!



Bus found

## Multi-scale sliding window

- Work with multiple window sizes
- Create a feature pyramid



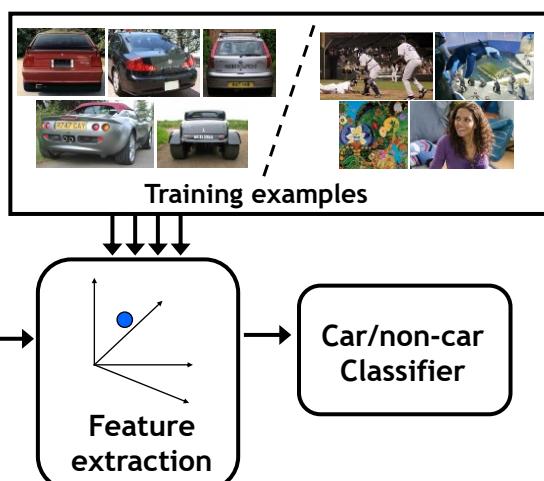
## Window-based object detection: recap

### Training:

1. Obtain training data
2. Define features
3. Define classifier

### Given new image:

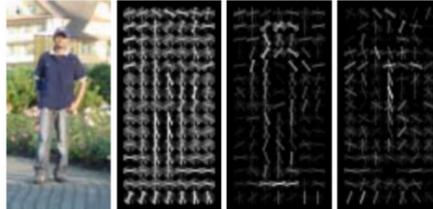
1. Slide window
2. Score by classifier



## Features

13

- HOG



- Bags of visual words

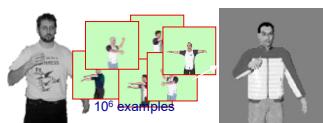
Bag of 'words'



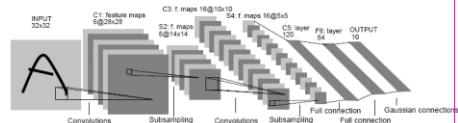
- Haar features, ...

## Discriminative classifier construction

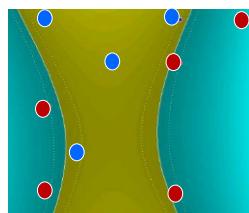
### Nearest neighbor



### Neural networks



### Support Vector Machines



### Boosting



### Conditional Random Fields

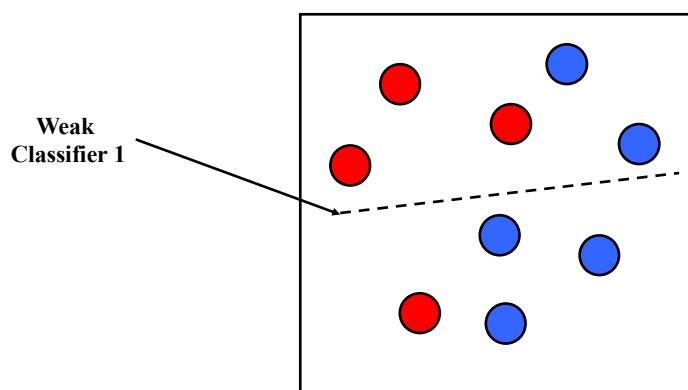


---

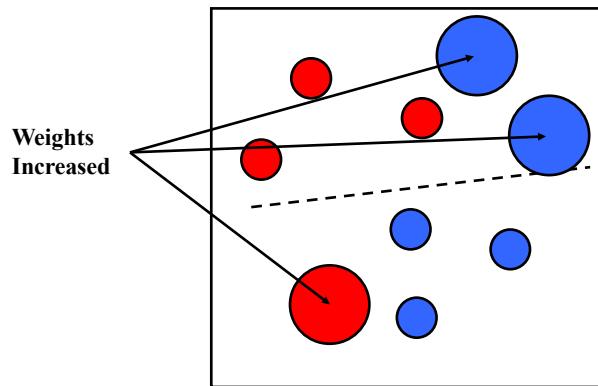
# Boosting classifiers

---

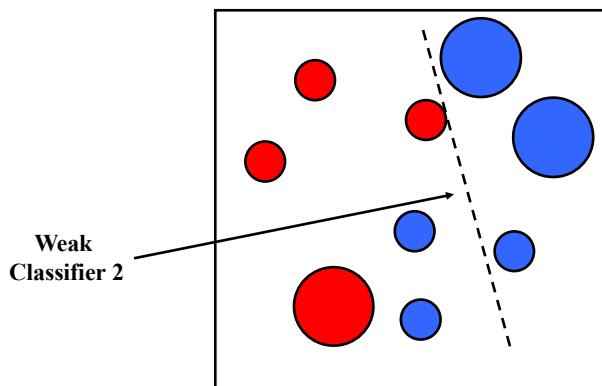
## Boosting intuition



## Boosting illustration

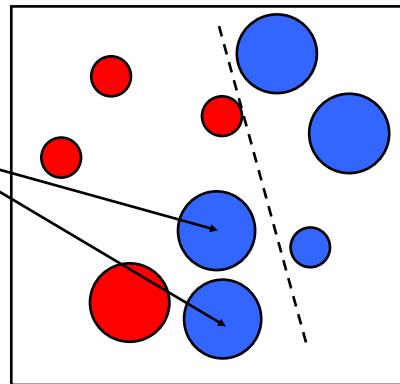


## Boosting illustration



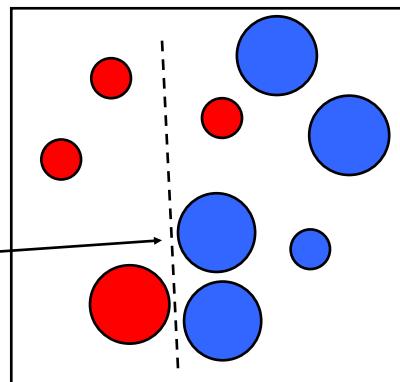
## Boosting illustration

Weights  
Increased



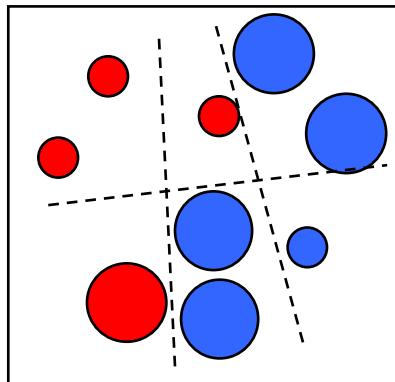
## Boosting illustration

Weak  
Classifier 3



## Boosting illustration

Final classifier is  
a combination of weak  
classifiers



## Boosting: training

- Initially, weight each training example equally
- In each boosting round:
  - Find the **weak learner** that achieves the lowest *weighted* training error
  - **Raise weights of training examples misclassified** by current weak learner
- Compute final classifier as linear combination of all weak learners
  - (weight of each learner is directly proportional to its accuracy)
- Exact formulas for re-weighting and combining weak learners **depend on the particular boosting scheme** (e.g., *AdaBoost*)

# Face detection as case study

## Viola-Jones face detector

ACCEPTED CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION 2001

### Rapid Object Detection using a Boosted Cascade of Simple Features

Paul Viola  
 viola@merl.com  
 Mitsubishi Electric Research Labs  
 201 Broadway, 8th FL  
 Cambridge, MA 02139

Michael Jones  
 mjones@crl.dec.com  
 Compaq CRL  
 One Cambridge Center  
 Cambridge, MA 02142

#### Abstract

*This paper describes a machine learning approach for vi-*

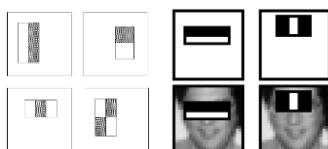
tected at 15 frames per second on a conventional 700 MHz Intel Pentium III. In other face detection systems, auxiliary information, such as image differences in video sequences,

## Viola-Jones face detector

### Main idea:

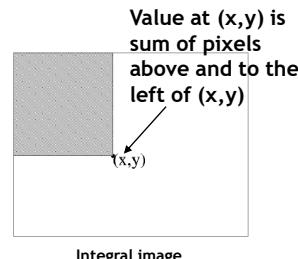
- Represent **local texture** with efficiently computable “rectangular” features within window of interest
- Select discriminative features to be weak classifiers
- Use boosted combination of them as final classifier
- Form a cascade of such classifiers, rejecting clear negatives quickly

## Viola-Jones detector: features

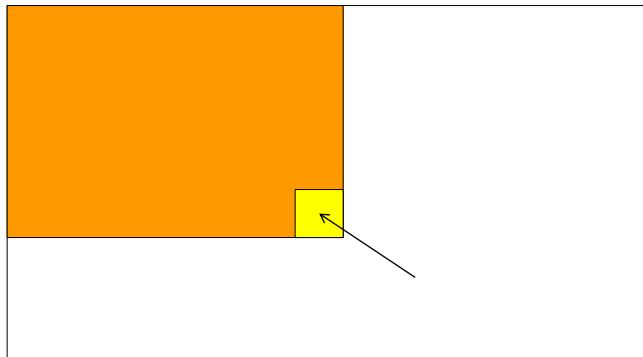


- “Rectangular” filters  
Feature output is difference between adjacent regions

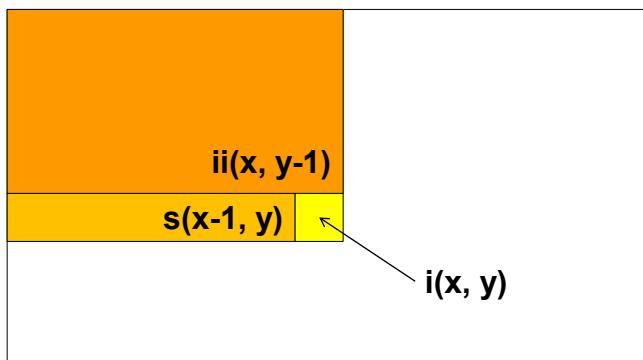
- Efficiently computable with **integral image**: any sum can be computed in constant time.



## Computing the integral image



## Computing the integral image

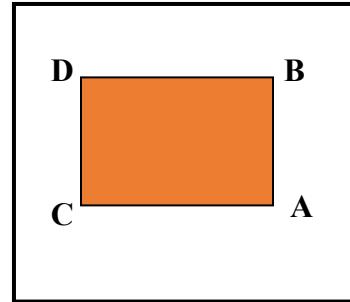


- Cumulative row sum:  $s(x, y) = s(x-1, y) + i(x, y)$
- Integral image:  $ii(x, y) = ii(x, y-1) + s(x, y)$

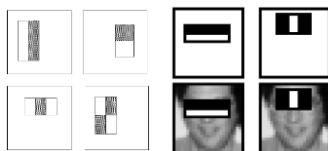
## Computing sum within a rectangle

- Let **A,B,C,D** be the values of the integral image at the corners of a rectangle
- Then the sum of original image values within the rectangle can be computed as:  

$$\text{sum} = A - B - C + D$$
- Only 3 additions are required for any size of rectangle!

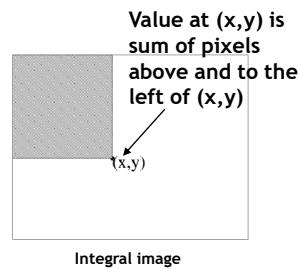


## Viola-Jones detector: features

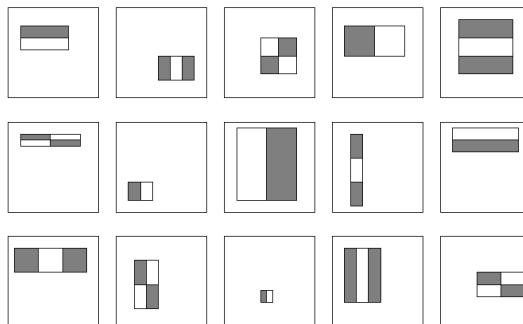


- “Rectangular” filters  
Feature output is difference between adjacent regions

- Efficiently computable with integral image: any sum can be computed in constant time  
Avoid scaling images → scale features directly for same cost



## Viola-Jones detector: features



Considering all possible filter parameters: position, scale, and type:

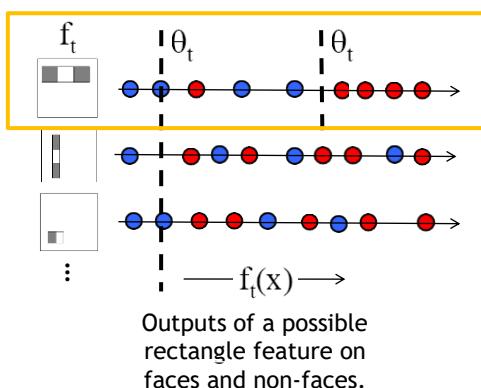
**180,000+** possible features associated with each 24 x 24 window

*Which subset of these features should we use to determine if a window has a face?*

Use AdaBoost both to select the informative features and to form the classifier

## Viola-Jones detector: AdaBoost

- Want to select the single rectangle feature and threshold that best separates **positive** (faces) and **negative** (non-faces) training examples, in terms of **weighted error**.



Resulting weak classifier:

$$h_t(x) = \begin{cases} +1 & \text{if } f_t(x) > \theta_t \\ -1 & \text{otherwise} \end{cases}$$

For next round, reweight the examples according to errors, choose another filter/threshold combo.

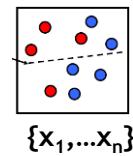
- Given example images  $(x_1, y_1), \dots, (x_n, y_n)$  where  $y_i = 0, 1$  for negative and positive examples respectively.
  - Initialize weights  $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$  for  $y_i = 0, 1$  respectively, where  $m$  and  $l$  are the number of negatives and positives respectively.
  - For  $t = 1, \dots, T$ :
    - Normalize the weights,
$$w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^n w_{t,j}}$$

so that  $w_t$  is a probability distribution.

    - For each feature,  $j$ , train a classifier  $h_j$  which is restricted to using a single feature. The error is evaluated with respect to  $w_t$ ,  $\epsilon_j = \sum_i w_i |h_j(x_i) - y_i|$ .
    - Choose the classifier,  $h_t$ , with the lowest error  $\epsilon_t$ .
    - Update the weights:
$$w_{t+1,i} = w_{t,i} \beta_t^{1-e_i}$$

where  $e_i = 0$  if example  $x_i$  is classified correctly,  $e_i = 1$  otherwise, and  $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$ .
  - The final strong classifier is:
- $$h(x) = \begin{cases} 1 & \sum_{t=1}^T \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \alpha_t \\ 0 & \text{otherwise} \end{cases}$$
- where  $\alpha_t = \log \frac{1}{\beta_t}$

Start with  
uniform weights  
on training  
examples



For T rounds

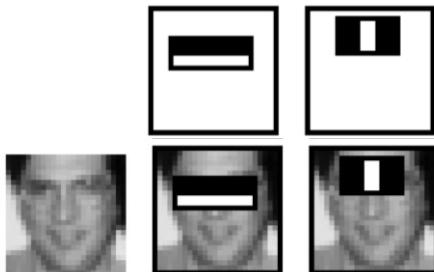
Evaluate  
weighted error  
for each feature,  
pick best.

Re-weight the examples:  
Incorrectly classified -> more weight  
Correctly classified -> less weight

Final classifier is combination of the  
weak ones, weighted according to error  
they had.

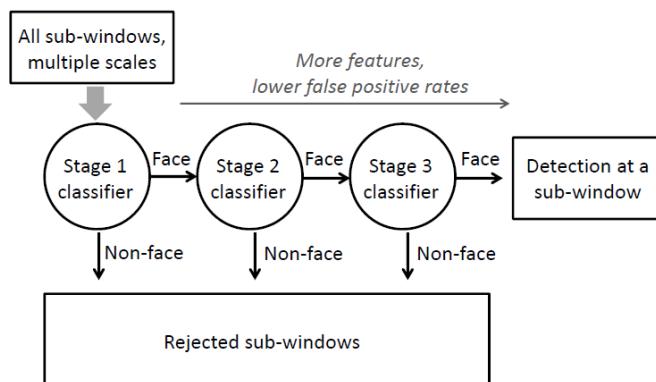
33

## Viola-Jones Face Detector: Results



- Even if the filters are fast to compute, each new image has a lot of possible windows to search.
- How to make the detection more efficient?

## Cascading classifiers for detection

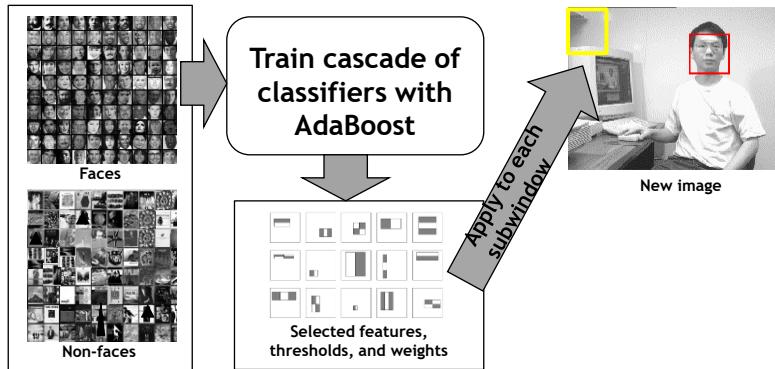


- Form a *cascade* with **low false negative rates early on**
- Apply less accurate but faster classifiers first **to immediately discard** windows that **clearly appear to be negative**

## Training the cascade

- Set **target detection** and **false positive rates** for each stage
- Keep **adding features** to the current stage until its target rates have been met
  - Need to **lower AdaBoost threshold to maximize detection** (as opposed to minimizing total classification error)
  - Test on a **validation set**
- If the **overall false positive rate is not low enough**, then add another stage
- **Use false positives** from current stage as **the negative training examples for the next stage**

## Viola-Jones detector: summary



- Train with 5K positives, 350M negatives
- Real-time detector using 38 layer cascade
- 6061 features in all layers

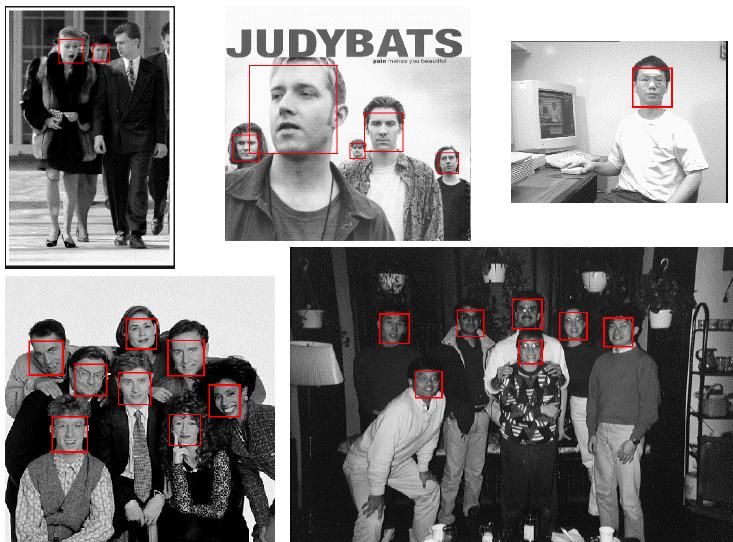
## Viola-Jones detector: summary

- A seminal approach to real-time object detection
  - 16,165 citations and counting
- Training is slow, but detection is very fast
- Key ideas
  - Integral images for fast feature evaluation
  - Boosting for feature selection
  - Attentional cascade of classifiers for fast rejection of non-face windows

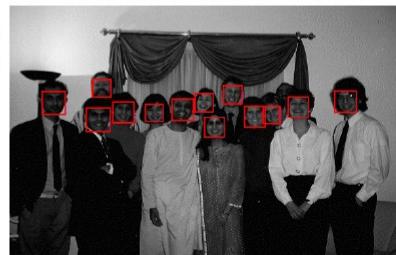
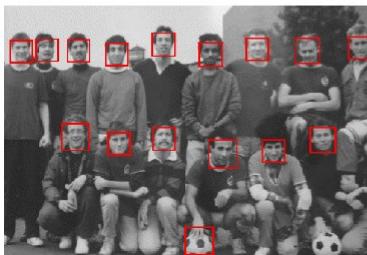
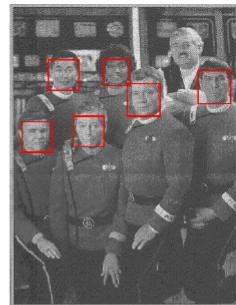
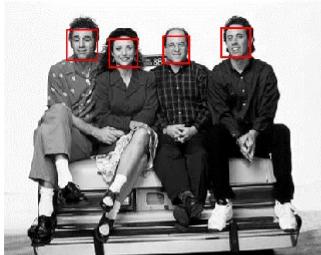
P. Viola and M. Jones. [\*Rapid object detection using a boosted cascade of simple features.\*](#) CVPR 2001.

P. Viola and M. Jones. [\*Robust real-time face detection.\*](#) IJCV 57(2), 2004.

## Viola-Jones Face Detector: Results



## Viola-Jones Face Detector: Results

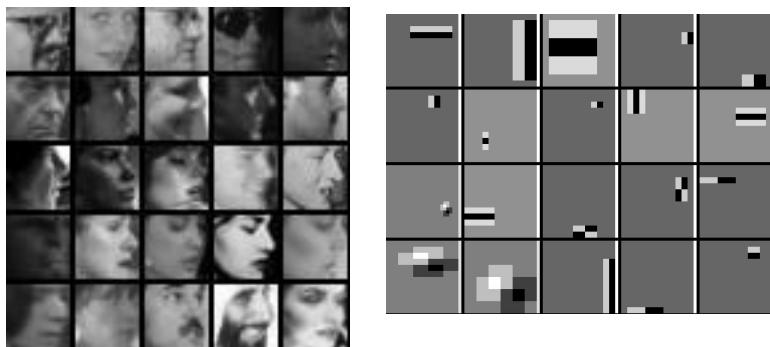


## Viola-Jones Face Detector: Results



## Detecting profile faces?

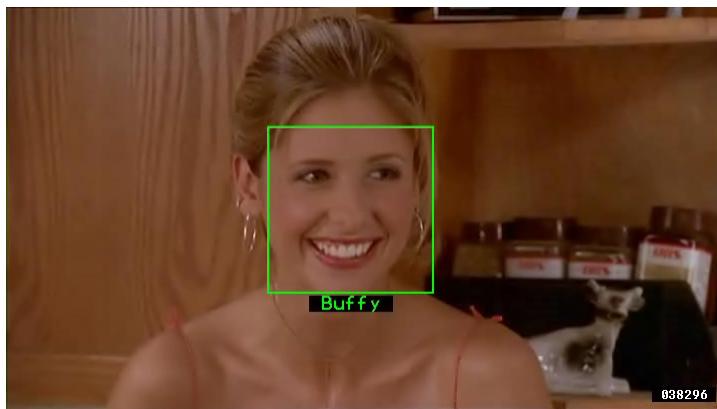
*Can we use the same detector?*



## Viola-Jones Face Detector: Results



## Example using Viola-Jones detector



Frontal faces detected and then tracked, character names inferred with alignment of script and subtitles.

Everingham, M., Sivic, J. and Zisserman, A.

"Hello! My name is... Buffy" - Automatic naming of characters in TV video,  
BMVC 2006. <http://www.robots.ox.ac.uk/~vgg/research/nface/index.html>

 Where Technology Means Business

 See how he stays with Cisco Collaboration Solutions [WATCH](#)

Home News Insight Reviews TechGuides Jobs Blogs Videos Community Downloads IT Library

Software Hardware Security Communications Business Internet Photos Search ZDNet Asia

News > Internet

**Google now erases faces, license plates on Map Street View**

By Elinor Mills, CNET News.com  
Friday, August 24, 2007 01:37 PM

Google has gotten a lot of flack from privacy advocates for photographing faces and license plate numbers and displaying them on the Street View in Google Maps. Originally, the company said only people who identified themselves could ask the company to remove their image.

But Google has quietly changed that policy, partly in response to criticism, and now anyone can alert the company and have an image of a license plate or a recognizable face removed, not just the owner of the face or car, says Marissa Mayer, vice president of search products and user experience at Google.

"It's a good policy for users and also clarifies the intent of the product," she said in an interview following her keynote at the Search Engine Strategies conference in San Jose, Calif., Wednesday.

The policy change was made about 10 days after the launch of the product in late May, but was not publicly announced, according to Mayer. The company is removing images only when someone notifies them and not proactively, she said. "It was definitely a big policy change inside."

News from Countries/Region

» Singapore	» India	» China/HK/R
» Malaysia	» Philippines	» ASEAN
» Thailand	» Indonesia	» Asia Pacific

What's Hot Latest News

- Is eBay facing seller revolt?
- Report: Amazon may again be mulling Netflix buyout
- Mozilla maps out Jetpack add-on transition plan
- Google begins search for Middle East lobbyist
- Google still thinks it can change China

▼ advertisement

Brought to you by CIS

**Cisco Collaboration Solutions**

Home > Technology

## Google street view blurs face of cow to protect its identity

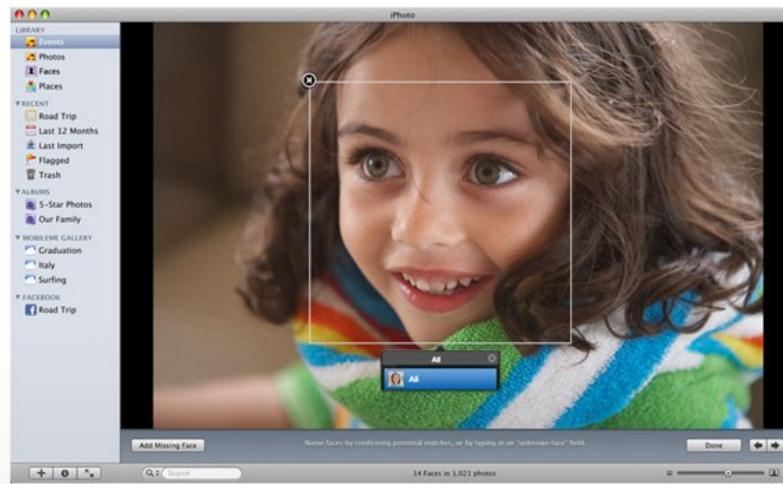


SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Slide: Kristen Grauman

47

## Consumer application: iPhoto



<http://www.apple.com/ilife/iphoto/>



SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Slide credit: Lana Lazebnik

48

## Consumer application: iPhoto



Things iPhoto thinks are faces



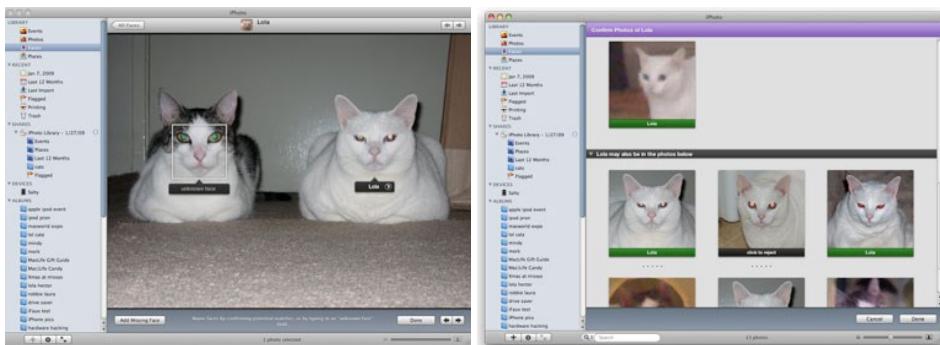
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Slide credit: Lana Lazebnik

49

## Consumer application: iPhoto

- Can be trained to recognize pets!



[http://www.maclife.com/article/news/iphotos\\_faces\\_recognizes\\_cats](http://www.maclife.com/article/news/iphotos_faces_recognizes_cats)



SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Slide credit: Lana Lazebnik

50

## Privacy Gift Shop – CV Dazzle

- <http://www.wired.com/2015/06/facebook-can-recognize-even-dont-show-face/>
- Wired, June 15, 2015



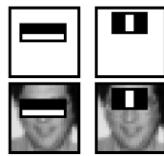
Slide: Kristen Grauman

51

## Boosting: pros and cons

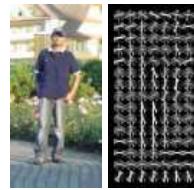
- Advantages of boosting
  - Integrates classification with feature selection
  - Complexity of training is linear in the number of training examples
  - Flexibility in the choice of weak learners, boosting scheme
  - Testing is fast
  - Easy to implement
- Disadvantages
  - Needs many training examples
  - Other discriminative models may outperform in practice (SVMs, CNNs,...)
    - especially for many-class problems

## Window-based models: Two case studies



Boosting + face  
detection

Viola & Jones



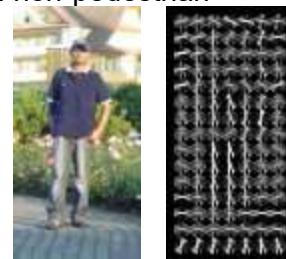
SVM + person  
detection

e.g., Dalal & Triggs

## SVM + HOG for human detection as case study

## Person detection with HoG's & linear SVM's

- Histogram of oriented gradients (HoG):
  - Map each grid cell in the input window to a histogram counting the gradients per orientation.
- Train a linear SVM
  - using training set of pedestrian vs. non-pedestrian windows.



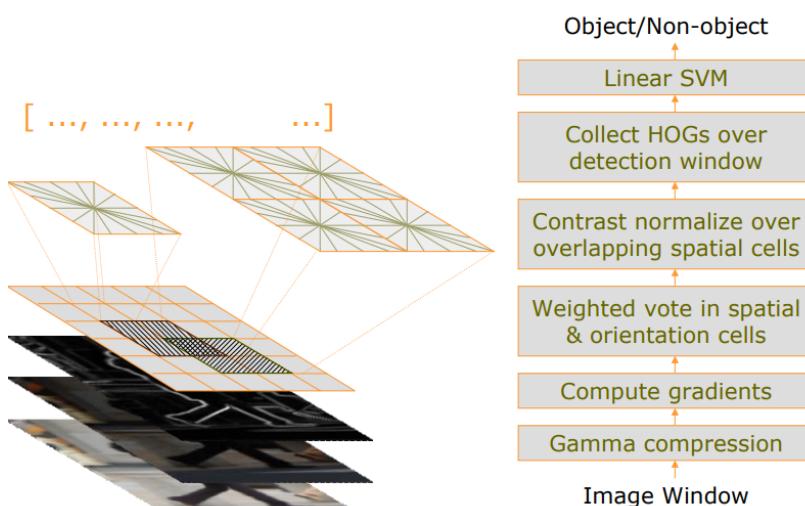
<https://lear.inrialpes.fr/people/triggs/pubs/Dalal-cvpr05.pdf> Dalal & Triggs, CVPR 2005



SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

64

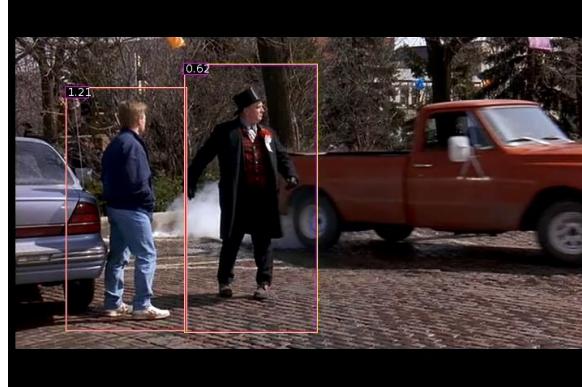
## Person detection with HoG's & linear SVM's



<http://host.robots.ox.ac.uk/pascal/VOC/voc2006/slides/dalal.pdf>

65

## Person detection with HoGs & linear SVMs



- For more detail about HoG:
  - Histograms of Oriented Gradients for Human Detection, [Navneet Dalal, Bill Triggs](#), International Conference on Computer Vision & Pattern Recognition - June 2005
  - <http://lear.inrialpes.fr/pubs/2005/DT05/>

## Key Descriptor Parameters

Class	Window Size	Avg. Size	# of Orientation Bins	Orientat-ion Range	Gamma Compre-ssion	Normal-isation Method
Person	64×128	Height 96	9	0°–180°	√RGB	L2-Hys
Car	104×56	Height 48	18	0°–360°	√RGB	L1-Sqrt
Bus	120×80	Height 64	18	0°–360°	√RGB	L1-Sqrt
Motorbike	120×80	Width 112	18	0°–360°	√RGB	L1-Sqrt
Bicycle	104×64	Width 96	18	0°–360°	√RGB	L2-Hys
Cow	128×80	Width 56	18	0°–360°	√RGB	L2-Hys
Sheep	104×60	Height 56	18	0°–360°	√RGB	L2-Hys
Horse	128×80	Width 96	9	0°–180°	RGB	L1-Sqrt
Cat	96×56	Height 56	9	0°–180°	RGB	L1-Sqrt
Dog	96×56	Height 56	9	0°–180°	RGB	L1-Sqrt

## Window-based detection: strengths

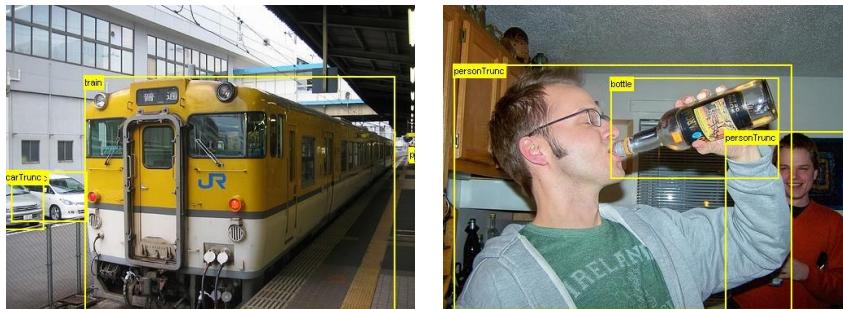
- Sliding window detection and global appearance descriptors:
  - Simple detection protocol to implement
  - Good feature choices critical
  - Past successes for certain classes

## Window-based detection: Limitations

- High computational complexity
  - For example: 250,000 locations x 30 orientations x 4 scales = 30,000,000 evaluations!
  - If training binary detectors independently, means cost increases linearly with number of classes
- With so many windows, false positive rate better be low

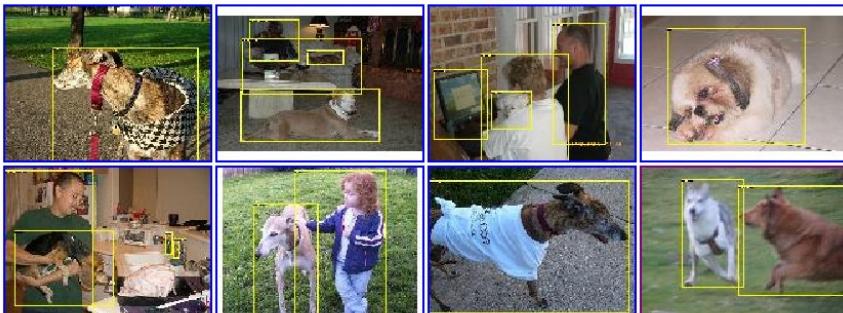
## Limitations (continued)

- Not all objects are “box” shaped



## Limitations (continued)

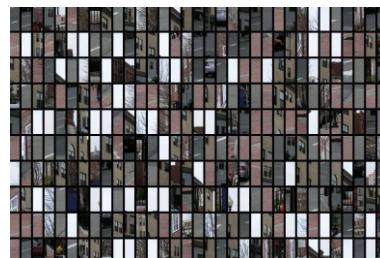
- Non-rigid, deformable objects not captured well with representations assuming a fixed 2d structure; or must assume fixed viewpoint
- Objects with less-regular textures not captured well with holistic appearance-based descriptions



## Limitations (continued)



Sliding window



Detector's view

If considering windows in isolation,  
context is lost

## Limitations (continued)

- In practice, often entails large, cropped training set (**expensive**)
- Requiring good match to a global appearance description can lead to **sensitivity to partial occlusions**

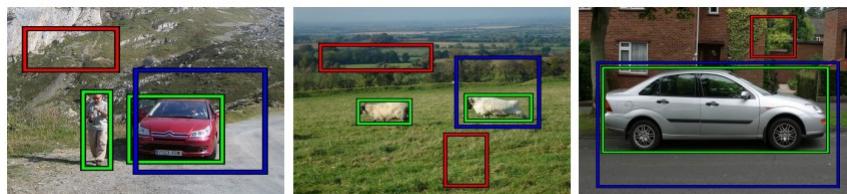


# Object proposals

# Object proposals

## Main idea:

- Learn to generate category-independent regions/boxes that have **object-like** properties.
- Let object detector **search over “proposals”**, not exhaustive sliding windows



## Object proposals



Alexe et al. Measuring the objectness of image windows, PAMI 2012

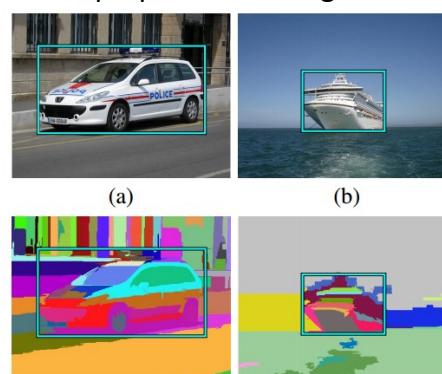
76

## Object proposals

Edge density



Superpixel straddling

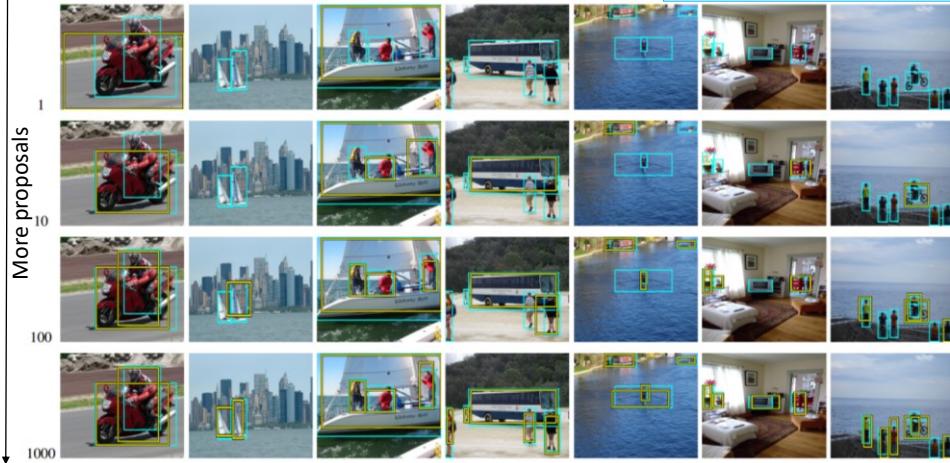


Alexe et al. Measuring the objectness of image windows, PAMI 2012

77

## Object proposals

Yellow box: object detected  
Cyan box: groundtruth



Alexe et al. Measuring the objectness of image windows, PAMI 2012

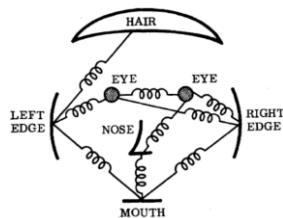


SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

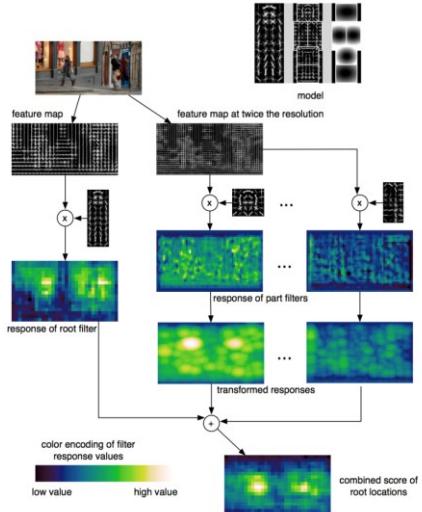
78

## Deformable Part Model (DPM)

- Represents an object as a **collection of parts** arranged in a deformable configuration
- Each part represents **local appearances**
- Spring-like connections between certain pairs of parts



Fischler and Elschlager, Pictorial Structures,  
1973



Felzenszwalb et al., PAMI 2010

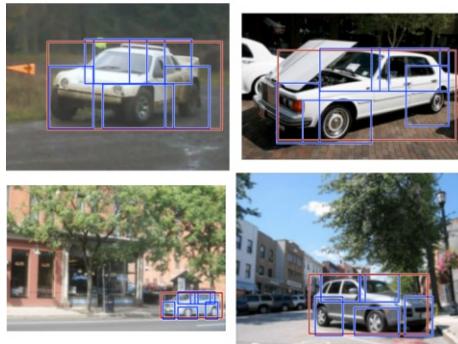


SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

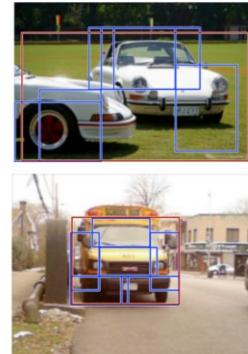
80

## Deformable Part Model (DPM)

high scoring true positives



high scoring false positives



## Deformable Part Model (DPM)

- References

- Pedro F. Felzenszwalb & Daniel P. Huttenlocher, Pictorial Structures for Object Recognition, IJCV 2005
  - <https://www.cs.cornell.edu/~dph/papers/pict-struct-ijcv.pdf>
- P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(9):1627–1645, 2010

# Object detection: Evaluation

## Object Detection Benchmarks

- PASCAL VOC Challenge
- ImageNet Large Scale Visual Recognition Challenge (ILSVR)
  - 200 Categories for detection
- Common Objects in Context (COCO)
  - 80 Object categories



## How do we evaluate object detection?



— predictions  
— ground truth

**True positive:**

- The overlap of the prediction with the ground truth is **MORE** than a threshold value (0.5)

## How do we evaluate object detection?



— predictions  
— ground truth

**True positive:**

**False positive:**

- The overlap of the prediction with the ground truth is **LESS** than a threshold value (0.5)

## How do we evaluate object detection?



— predictions

— ground truth

**True positive:**

**False positive:**

**False negative:**

- The objects that our model doesn't find

## How do we evaluate object detection?



— predictions

— ground truth

**True positive:**

**False positive:**

**False negative:**

- The objects that our model doesn't find

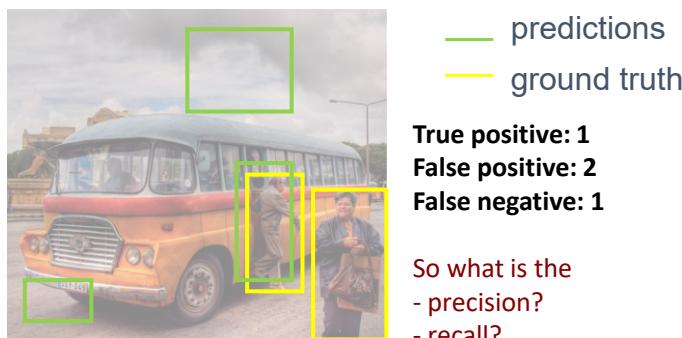
What is a **True Negative**?

	<u>Predicted 1</u>	<u>Predicted 0</u>	
<u>True 1</u>	true positive	false negative	<u>Predicted 1</u>
<u>True 0</u>	false positive	true negative	<u>Predicted 0</u>
			TP      FN
			FP      TN

	<u>Predicted 1</u>	<u>Predicted 0</u>	
<u>True 1</u>	hits	misses	<u>Precision</u>
<u>True 0</u>	false alarms	correct rejections	<u>Recall</u>
			$precision = \frac{TP}{TP + FP}$
			$recall = \frac{TP}{TP + FN}$

## How do we evaluate object detection?



# Precision versus recall

- Precision:

- how many of the object detections are correct?

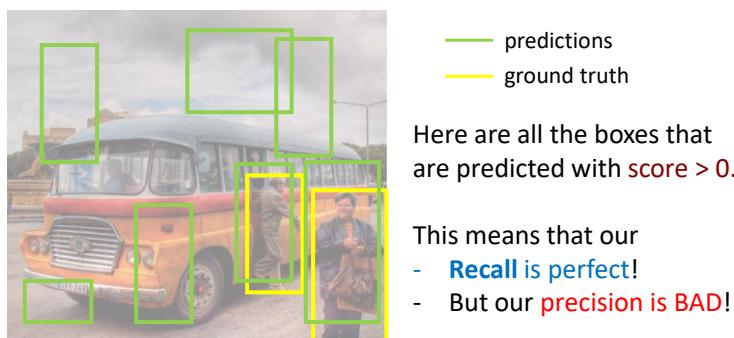
$$\text{precision} = \frac{TP}{TP + FP}$$

- Recall:

- how many of the ground truth objects can the model detect?
- True Positive Rate (TPR)

$$\text{recall} = \frac{TP}{TP + FN}$$

- In reality, our model makes a lot of predictions with varying scores between 0 and 1



## How do we evaluate object detection?

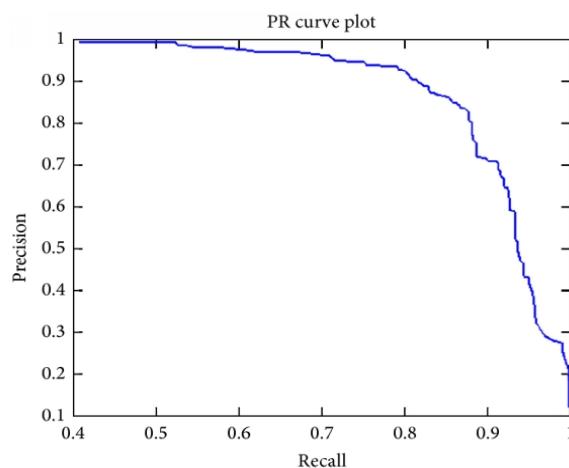


— predictions  
— ground truth

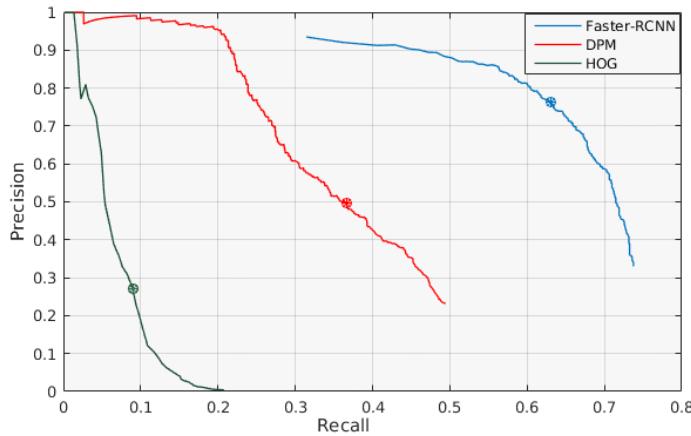
Here are all the boxes that are predicted with score > 0.5

We are setting a **threshold** of 0.5

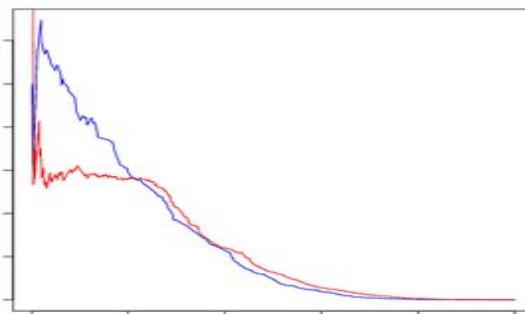
## Precision – recall curve (PR curve)



## Which model is the best?

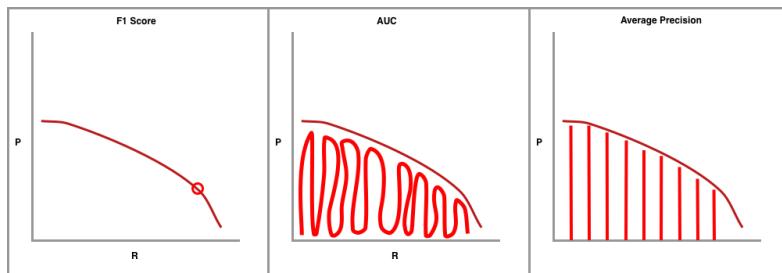


## Which model is the best?



- Area under curve (AUC), average precision (AP)
- F1-score (highest value at optimal confidential score)

## Which model is the best?



AP: The metric calculates the average precision (AP) for each class individually across all of the IoU thresholds

$$\begin{aligned} AP &= \frac{1}{11} \sum_{r \in \{0.0, 0.1, \dots, 0.9, 1\}} p_{inter,p}(r) \\ \text{mAP: the average of AP} &= \frac{1}{11} (1 + 1 + 1 + 1 + 0.67 + 0.67 + 0.67 + 0.5 + 0.5 + 0.5 + 0.5) \\ &\approx 0.728 \end{aligned}$$

## Summary

- Object detection as classification task
  - Boosting (face detection ex)
  - Support vector machines and HOG (human detection ex)
  - Sliding window search paradigm
    - Pros and cons
    - Speed up with attentional cascade
    - Object proposals, proposal regions as alternative

# References

Most of these slides were adapted from:

1. Kristen Grauman (CS 376: Computer Vision, Spring 2018, The University of Texas at Austin)



99



Thank  
you!