Image Processing Technology and Applications:

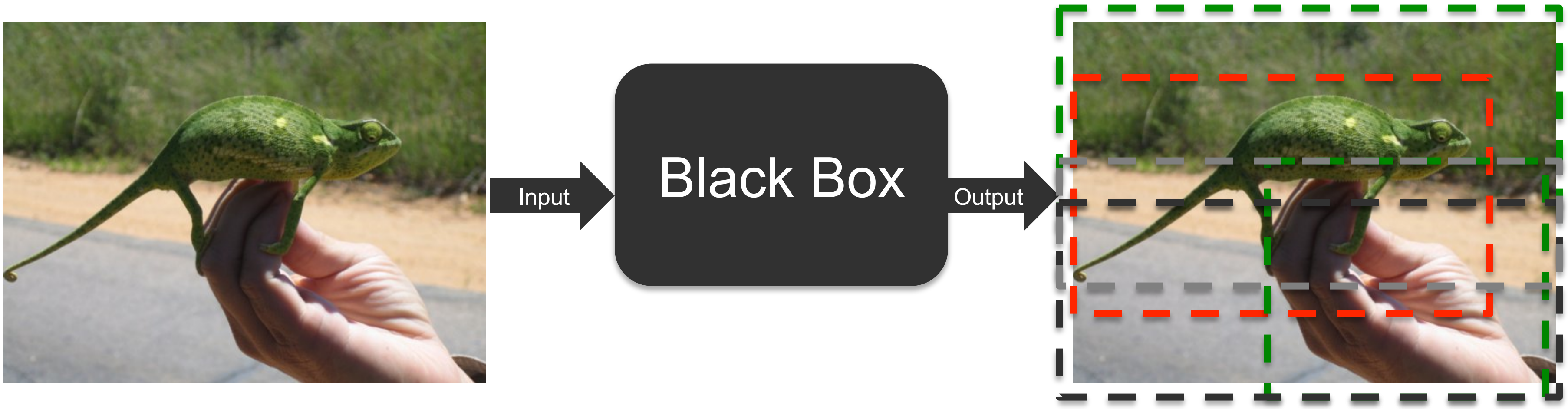
Object Recognition and Detection in Natural Images

Katherine Bouman – MIT

November 26, 2012

Goal

- To be able to automatically understand the content of an image



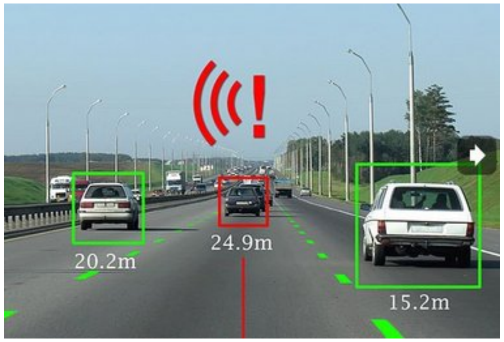
Chameleon
Hand
Grass
Road
Sand

Motivation

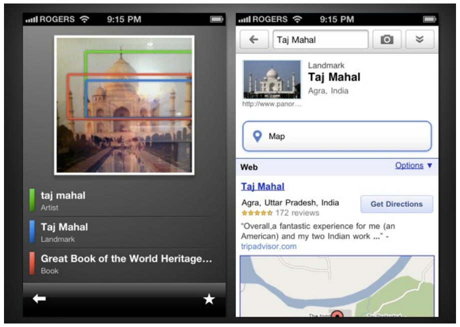
- Recently object detection in natural images is starting to have a lot of commercial success!

Automatic Focus

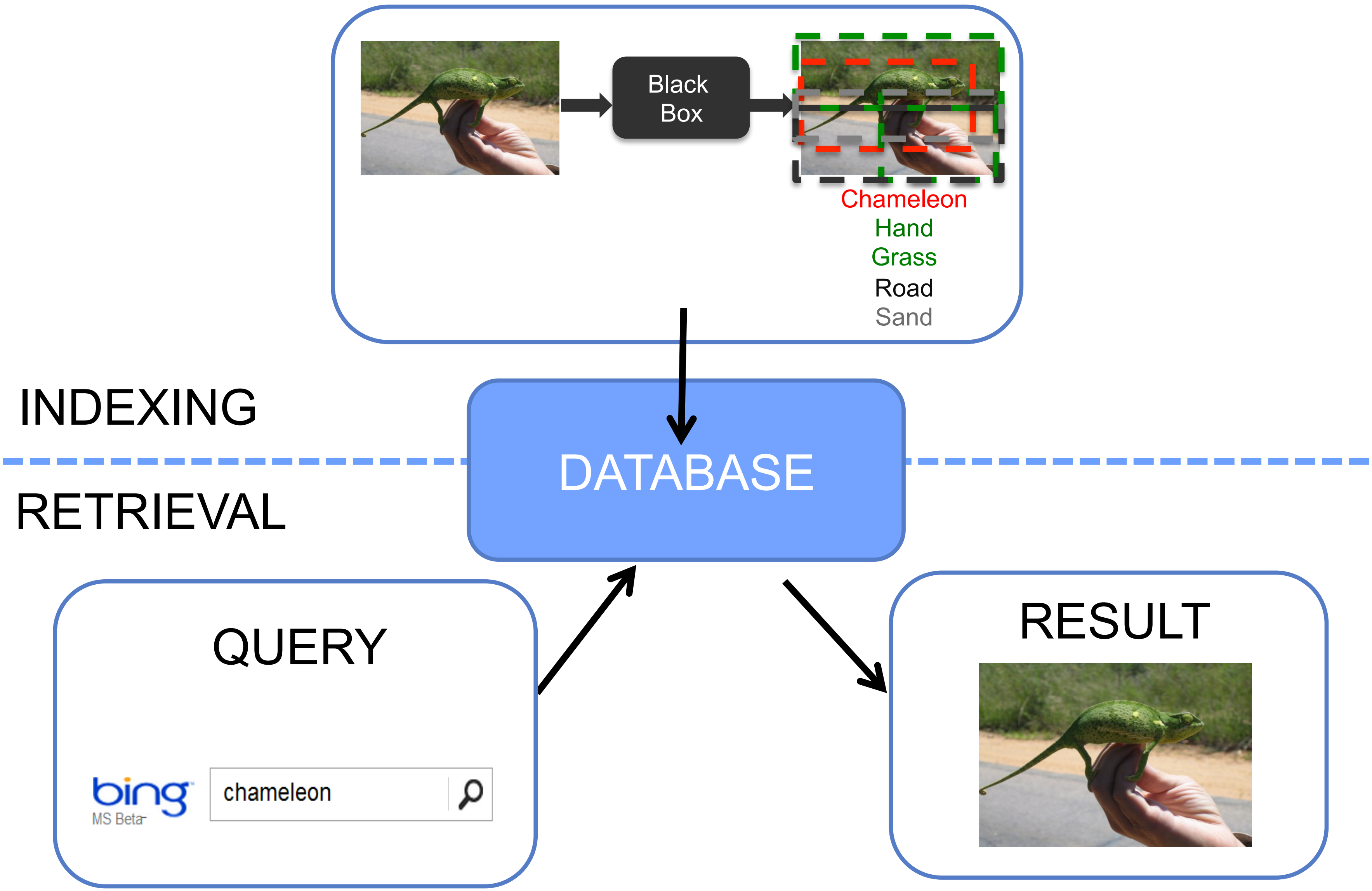
MobilEye



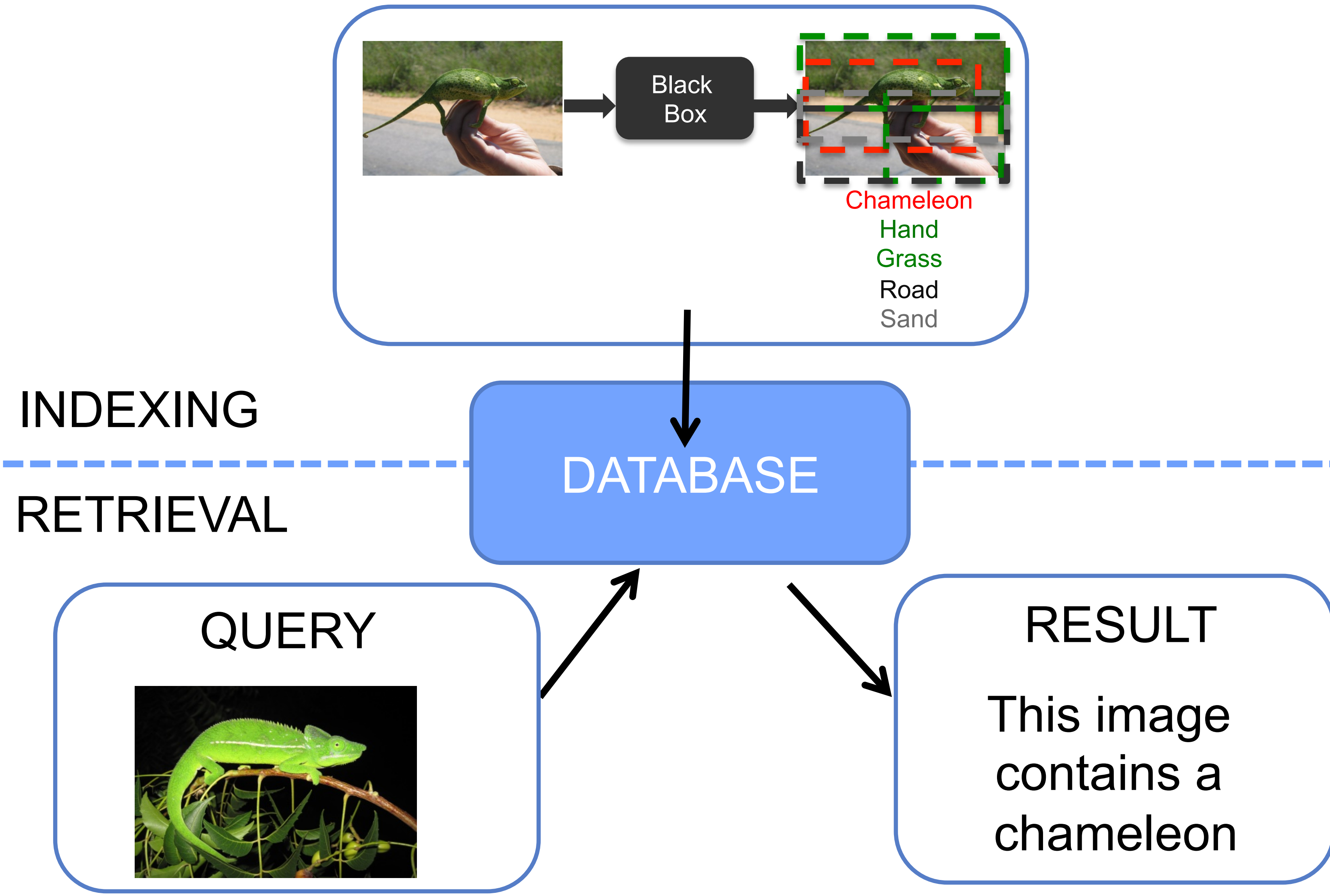
Google Goggles



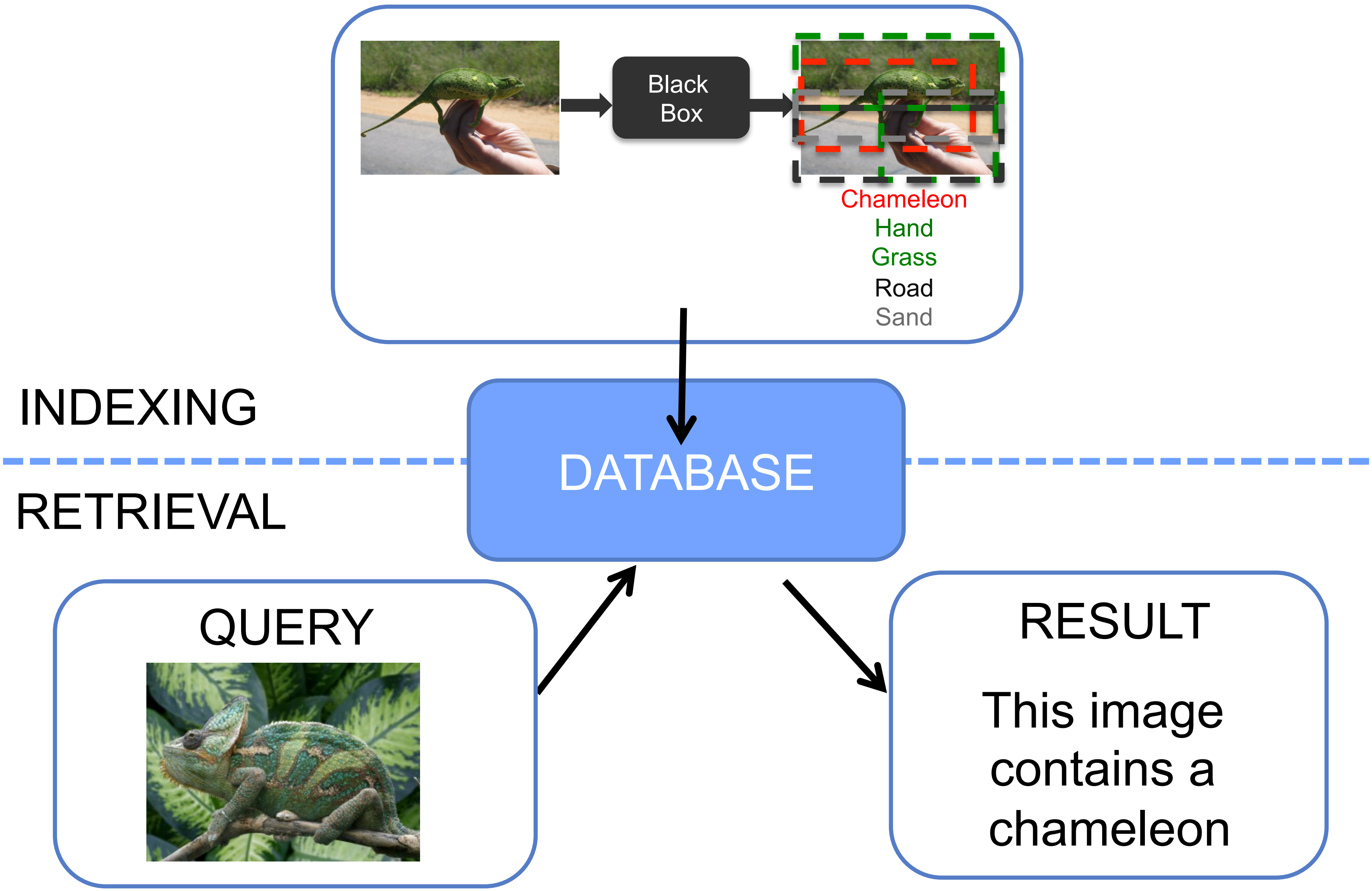
Motivation


Black
Box
Chameleon
Hand
Grass
Road
Sand
INDEXING
DATABASE
RETRIEVAL
QUERY
bing
MS Beta
chameleon
RESULT

Motivation

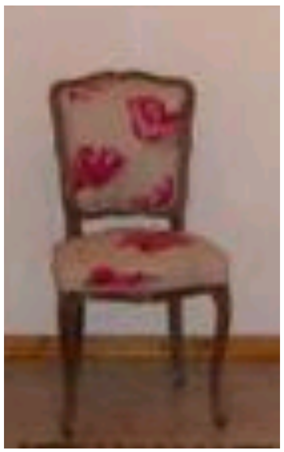

Black Box
Chameleon
Hand
Grass
Road
Sand
INDEXING
DATABASE
RETRIEVAL
QUERY
RESULT
This image contains a chameleon

Motivation


Black Box
Chameleon
Hand
Grass
Road
Sand
INDEXING
DATABASE
RETRIEVAL
QUERY
RESULT
This image contains a chameleon

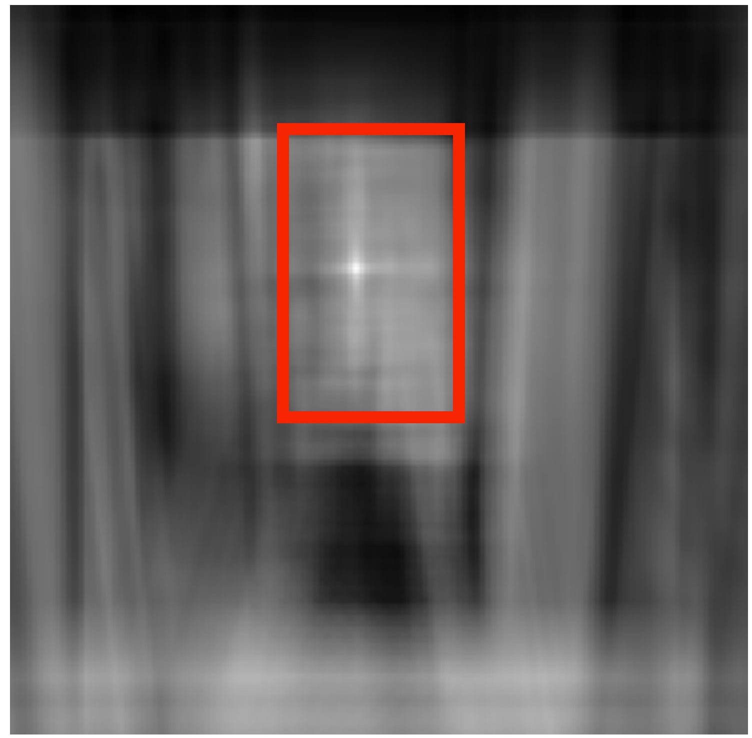
Is Object Detection Really that Hard?

This is a chair



Find the chair in this image

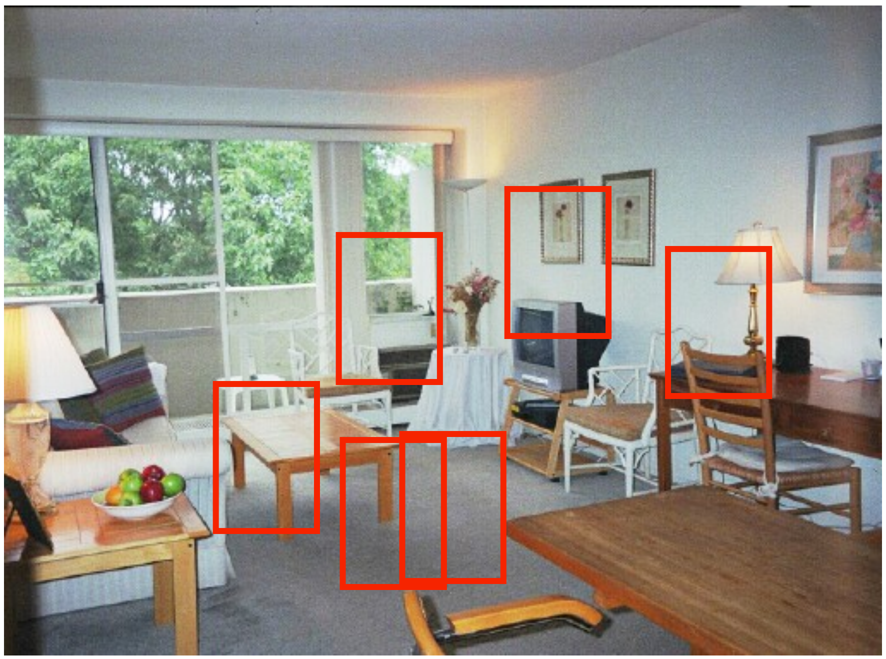
Output of normalized correlation



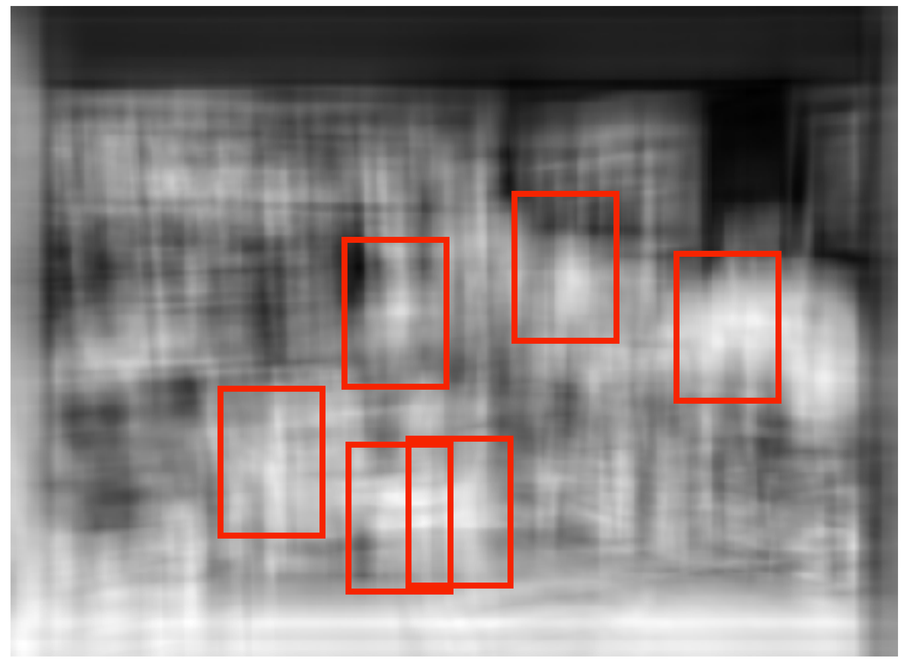
Slide Credit: A. Torralba

Is Object Detection Really that Hard?

Find the chairs in this image



Output of normalized correlation



Garbage!

Slide Credit: A. Torralba

Challenges of Object Detection

- **View Point Variation**

Challenges of Object Detection

- View Point Variation
- **Illumination**

Image Credit: S. Ullman

Challenges of Object Detection

- View Point Variation
- Illumination
- **Occlusion**

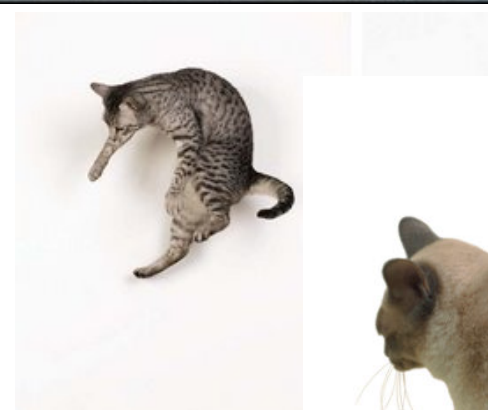


Challenges of Object Detection

- View Point Variation
- Illumination
- Occlusion
- **Scale**

Challenges of Object Detection

- View Point Variation
- Illumination
- Occlusion
- Scale
- **Deformation/Articulation**



Challenges of Object Detection

- View Point Variation
- Illumination
- Occlusion
- Scale
- Deformation/Articulation
- **Intra-Class Variation**



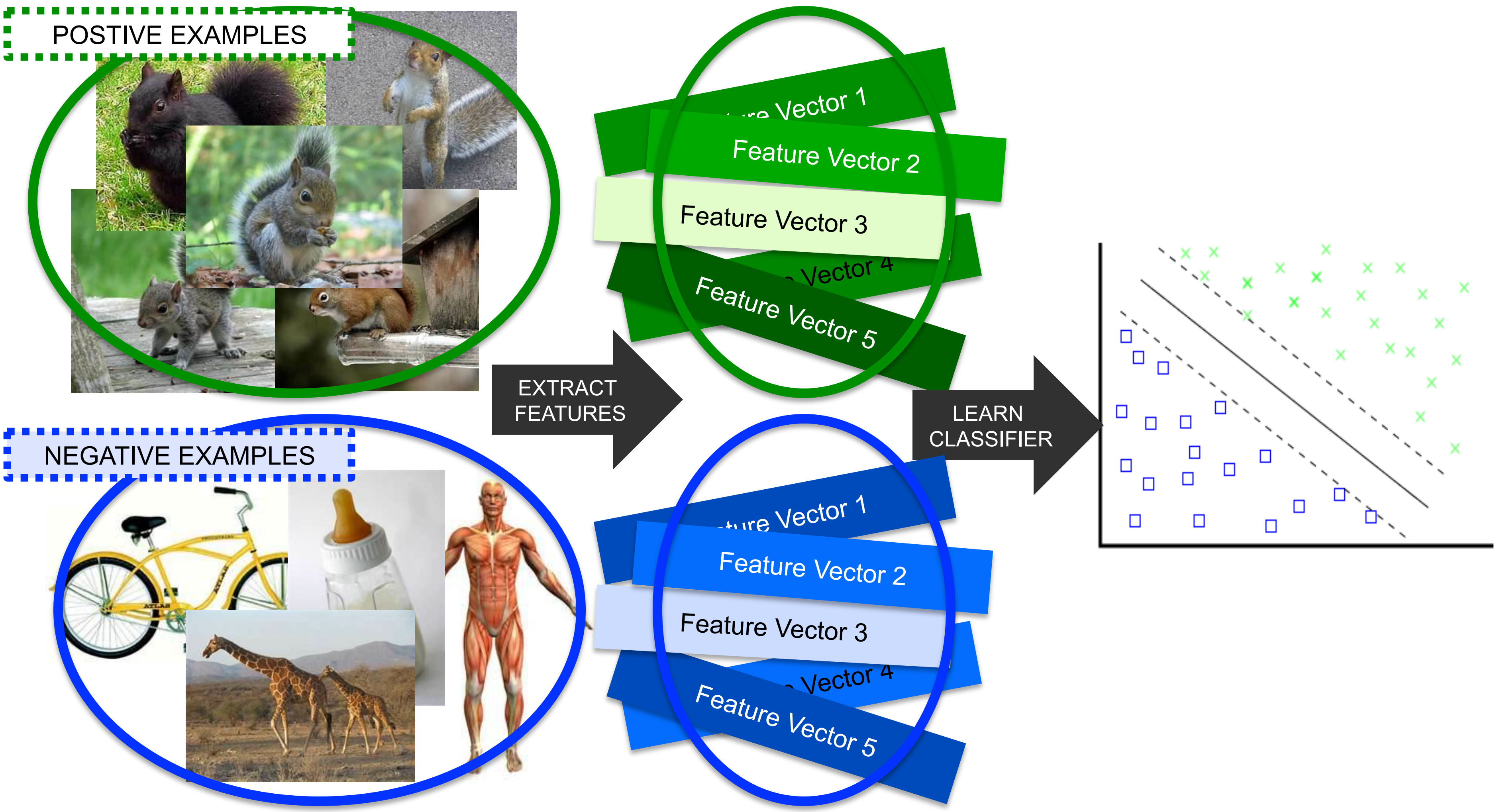


Challenges of Object Detection

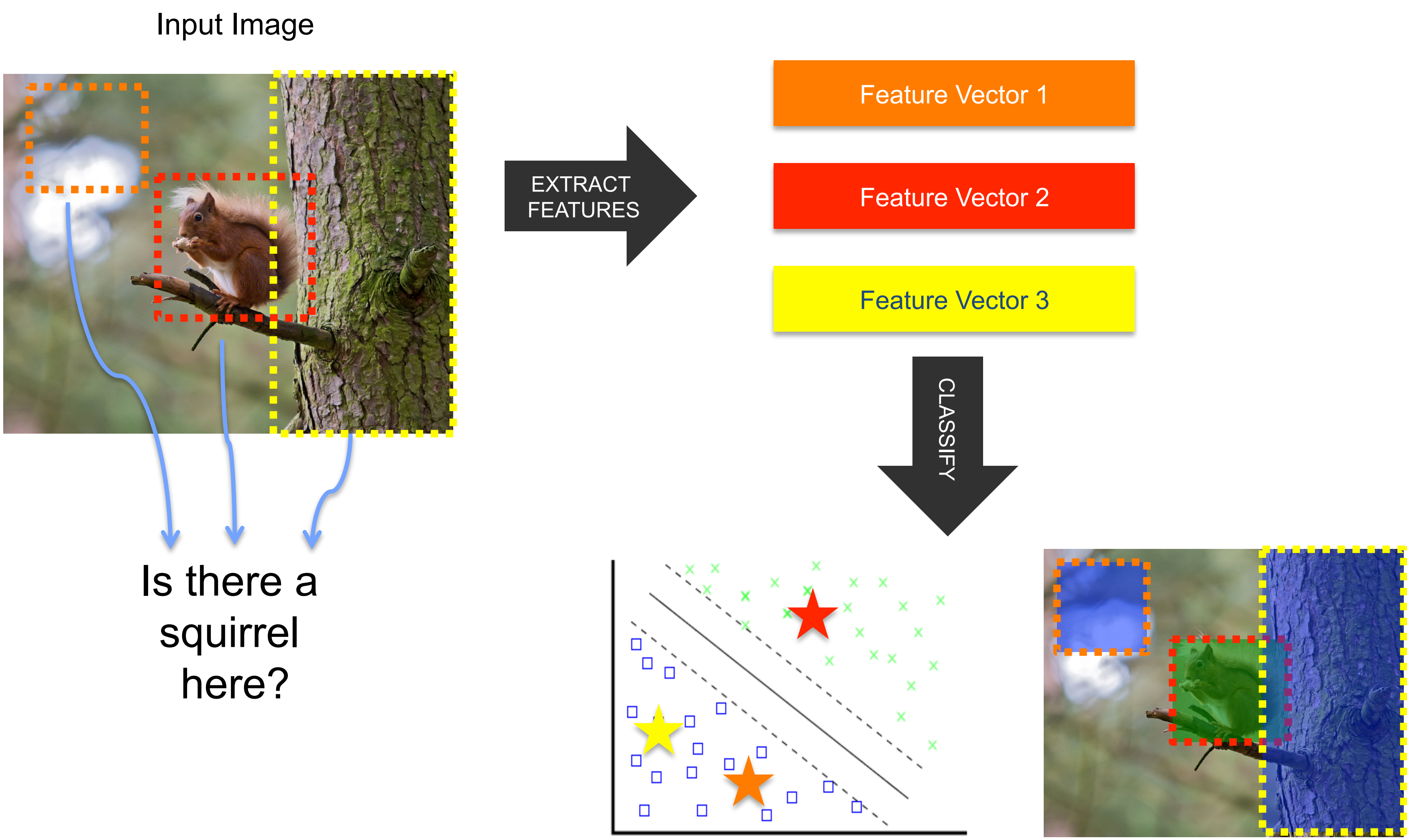
- View Point Variation
- Illumination
- Occlusion
- Scale
- Deformation/Articulation
- Intra-Class Variation
- **Background Clutter**

Image Credit: A. Torralba

General Approach Pipeline: Learning Model



General Approach Pipeline: Detecting Objects



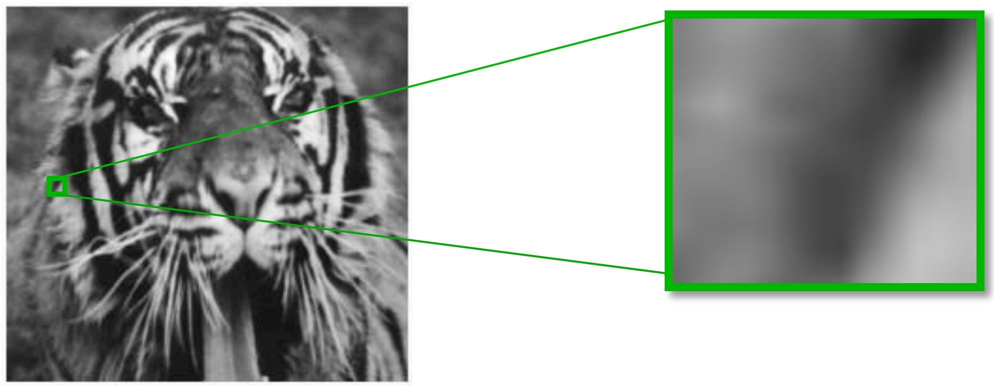
Questions To Answer

- What features should we use?
  - Intensity, color, gradient information, etc…
- What models should we use?
- How can we realistically implement an algorithm?

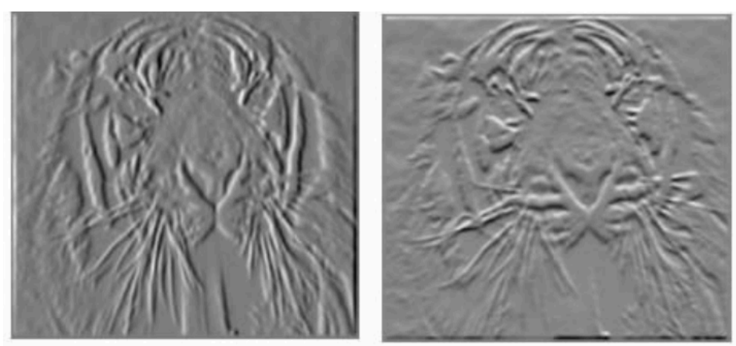
What features should we use?

Common Image Descriptors for Detection

- Descriptors encode local neighboring window around keypoints



- Commonly descriptors in object detection try to capture gradient information
  - Human Perception is sensitive to gradient orientation
  - Invariant to changes in lighting and small deformations



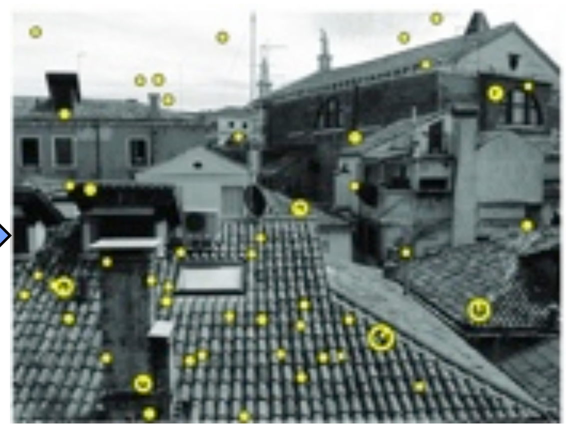
Common Image Descriptors for Detection

- Most common image descriptors currently used in object detection
  - SIFT – Scale Invariant Feature Transform
  - HOG – Histogram of Oriented Gradients
  - and many variants of these…

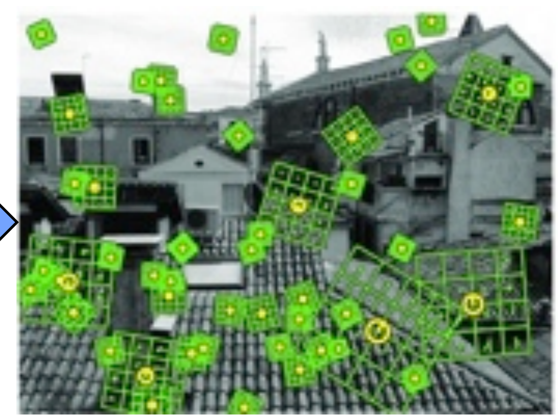
SIFT – Scale Invariant Feature Transform

- Input an Image
- Extract Keypoints
  - Finds “corners”
  - Determines scale and orientation of the keypoint
- Compute Descriptor for each Keypoint
  - Histogram of gradients in Gaussian window around keypoint

Input Image



Extract Keypoints

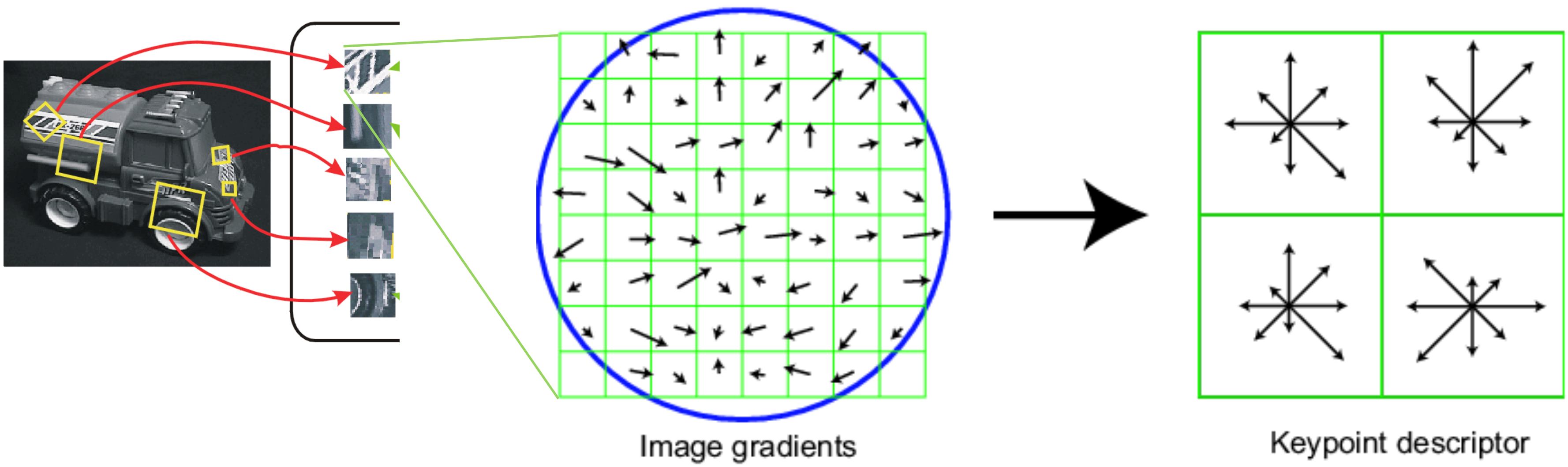


Compute Descriptors

Image Credit: VLFeat

SIFT – Scale Invariant Feature Transform

- Compute the gradient for each pixel in local neighboring window
  - Typically 8 gradient directions
  - Neighboring window is determined by scale of the keypoint
- Pool Gradients into a 4x4 histogram
  - Weight each magnitude by a Gaussian window centered around the keypoint
- 8x4x4=128 dimensional output vector normalized to 1



showing only 2x2 here but is generally 4x4

Image Credit: D. Lowe

SIFT – Scale Invariant Feature Transform

- Match groups of keypoints across images
  - Invariant to scale and some changes in lighting and orientation
- Great for finding the same instance of an object!

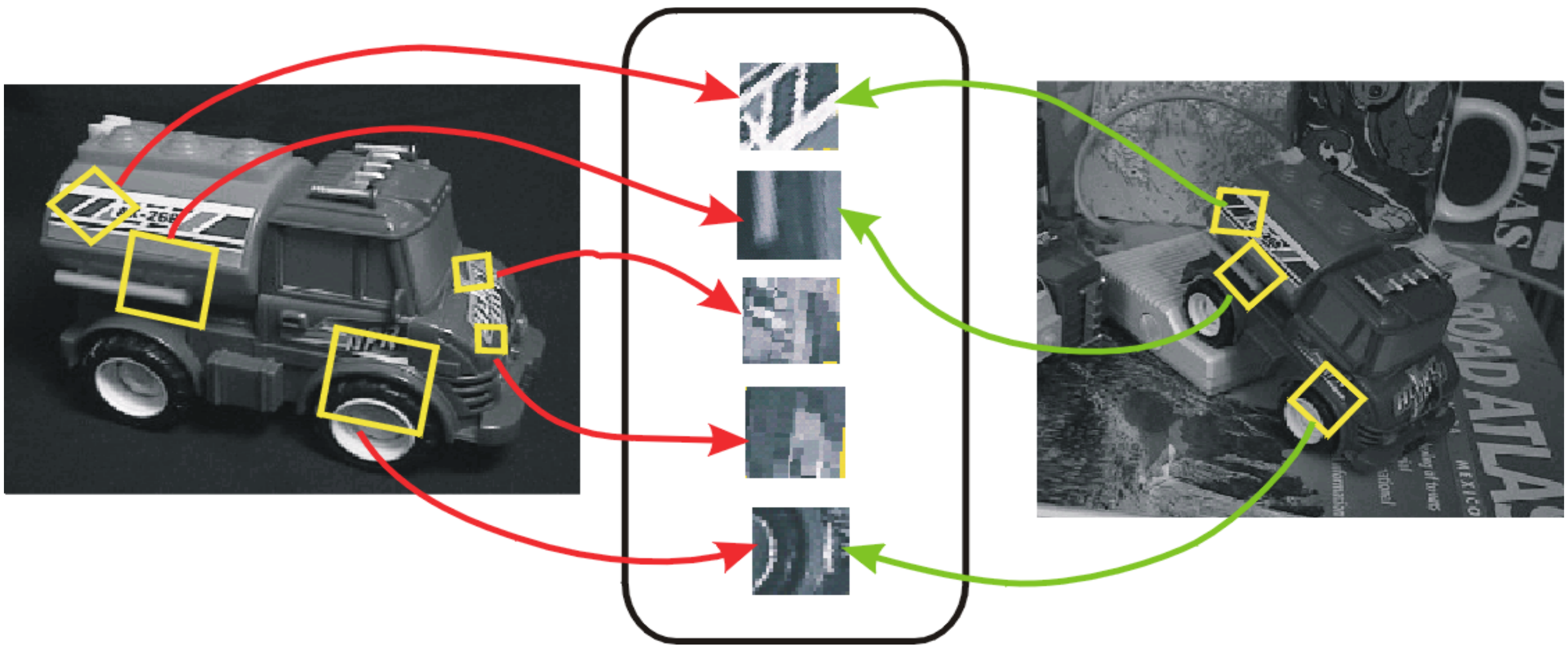


Image Credit: D. Lowe

SIFT – Scale Invariant Feature Transform

- Not good at finding different instances of an object

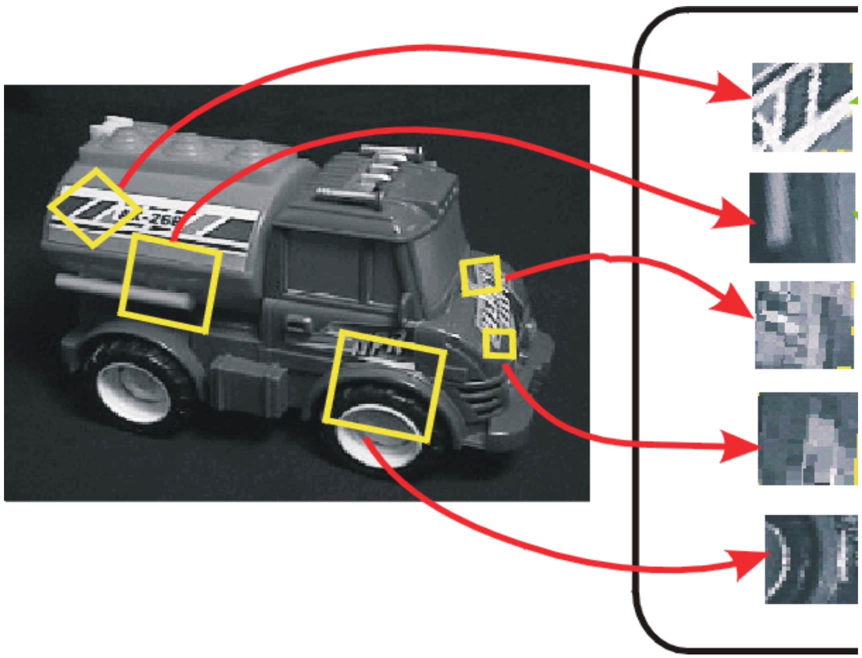


Image Credit: D. Lowe

SIFT – Scale Invariant Feature Transform

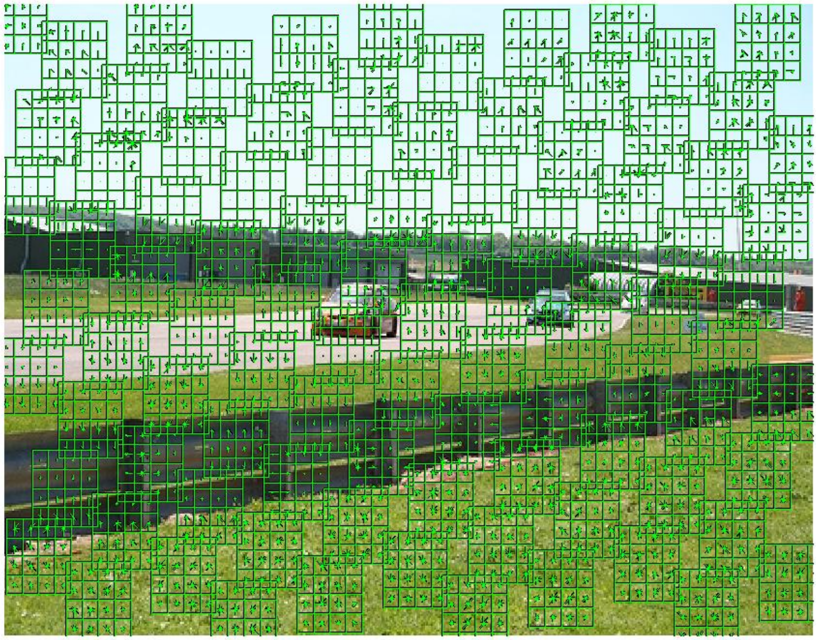
- Input an Image
- Extract Keypoints
  - Finds “corners”
  - Determines scale and orientation of the keypoint
- Compute Descriptor for each Keypoint
  - Histogram of gradients in Gaussian window around keypoint

Input Image Extract Keypoints Compute Descriptors

Image Credit: VLFeat

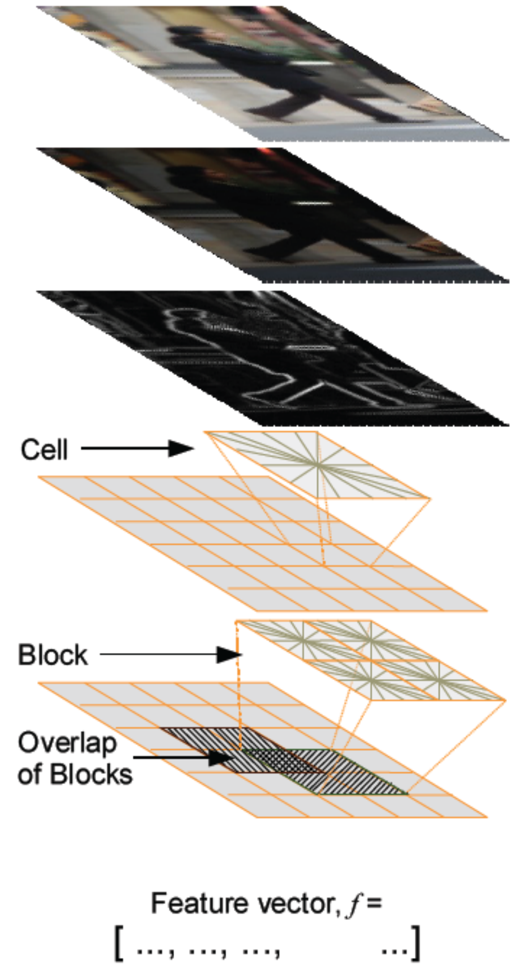
DSIFT – Dense ~~Scale Invariant~~ Feature Transform

- Input an Image
- Compute Descriptor for each k pixels
  - Use a fixed scale to calculate each descriptor
  - No longer scale invariant

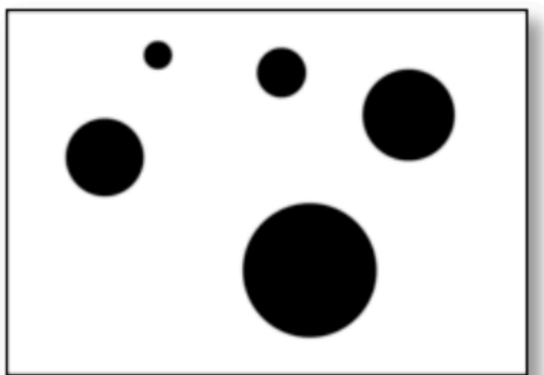


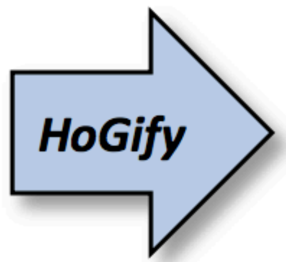
HOG – Histogram of Oriented Gradients

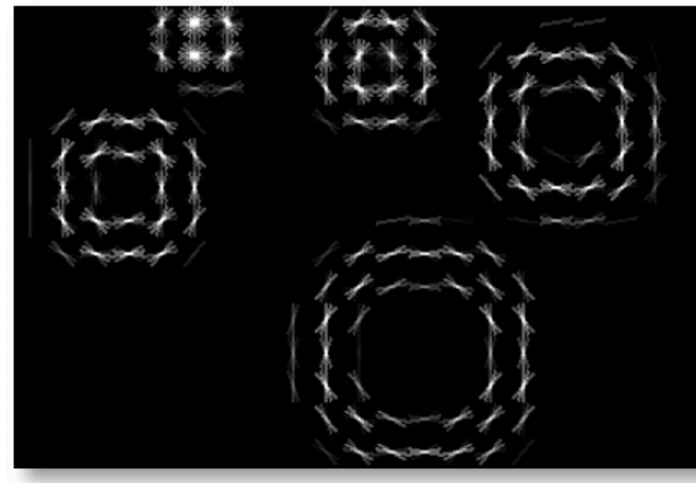
- Input an Image
- Normalize Gamma and Color
- Compute Gradients
- Accumulate weighted votes for gradient orientation over spatial bins
- Normalize contrast within overlapping blocks of cells
- Collect HOGs for all blocks over image



HOG – Histogram of Oriented Gradients



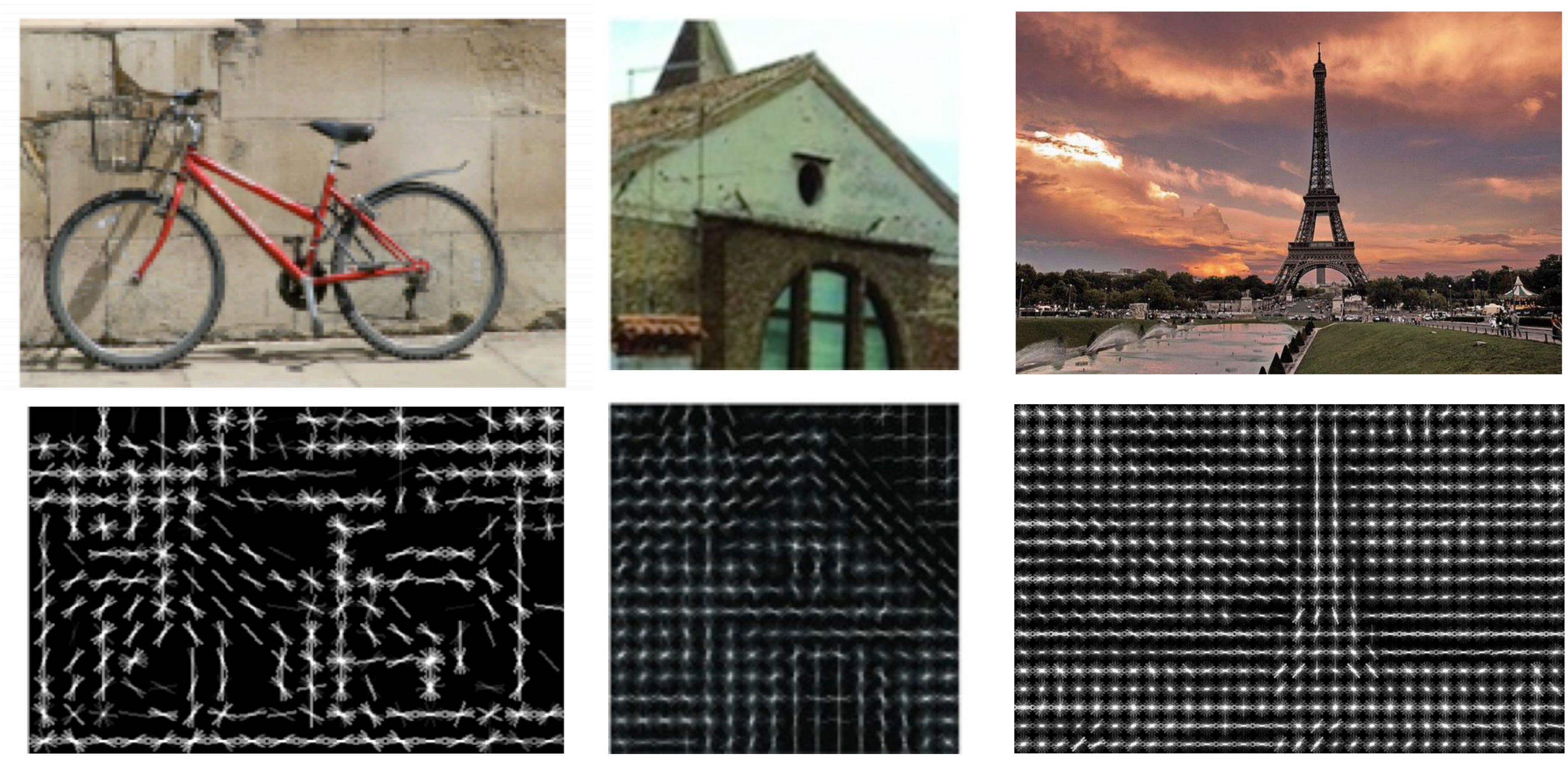




10x10 cells

20x20 cells

HOG – Histogram of Oriented Gradients



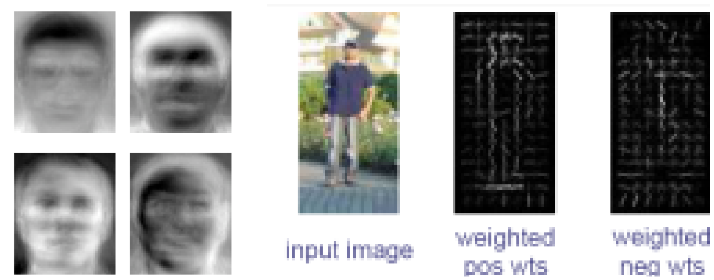
What models should we use?

Families of Detection/Recognition Models

Bag of Words Models

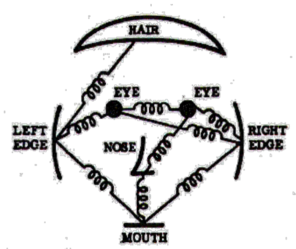
Csurka, Dance, Fan, Willamowski, and Bray 2004
Sivic, Russell, Freeman, Zisserman, ICCV 2005

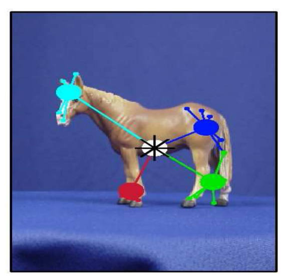
Rigid Template Models



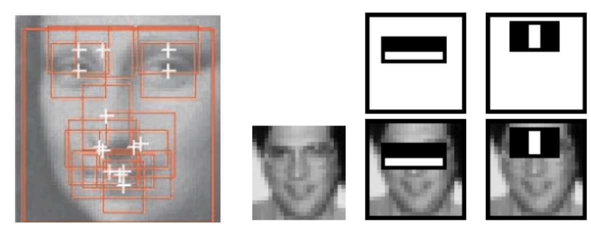
Sirovich and Kirby 1987
Turk, Pentland, 1991
Dalal & Triggs, 2006

Structure Models -Part and Voting Models





Fischler and Elschlager, 1973
Burl, Leung, and Perona, 1995
Weber, Welling, and Perona, 2000
Fergus, Perona, & Zisserman, CVPR 2003



Viola and Jones, ICCV 2001
Heisele, Poggio, et. al., NIPS 01
Schneiderman, Kanade 2004
Vidal-Naquet, Ullman 2003

Image Credit: A. Torralba

Families of Detection/Recognition Models

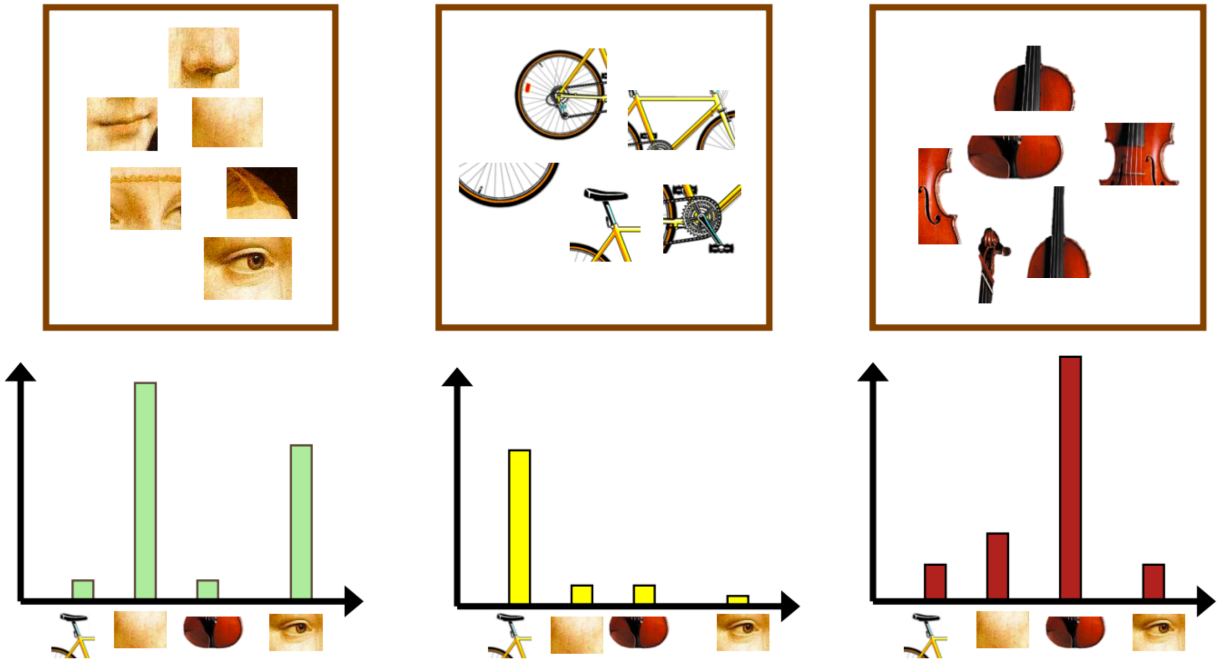
- Models capture varying degrees of spatial relationships between features
  - Bag of Words
  - Structure Models
  - Rigid Template Models

Bag of Words

Image Credit: L. Fei Fei

Bag of Words

- Extract local descriptors from image
- Learn a "visual vocabulary" (codebook) of local descriptors
- Quantize the local descriptors using the codebook
- Represent images by frequencies of visual words



Slide Credit: J. Hays

Bag of Words: Learning a Codebook

- Extract Features from all images and then cluster
  - SIFT, DSIFT
  - K-means
- How to choose vocabulary size?
  - Too small: visual words not representative of all patches
  - Too large: quantization artifacts and overfitting

Image Credit: J. Hays

Bag of Words: Coding Local Descriptors

- Vector Quantization Coding
  - Map each feature to the index of the nearest visual word in the codebook
- Locally-Constrained Linear Coding
  - Write each feature as a linear combination of the visual words

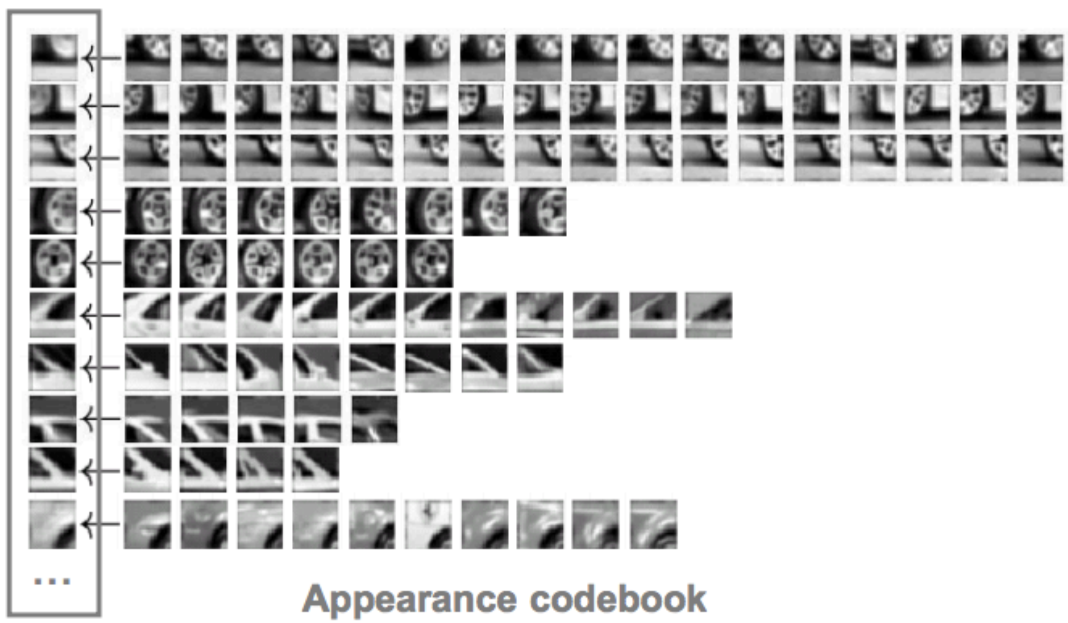


Image Credit: J. Hays

Bag of Words: Coding Local Descriptors

Original Image

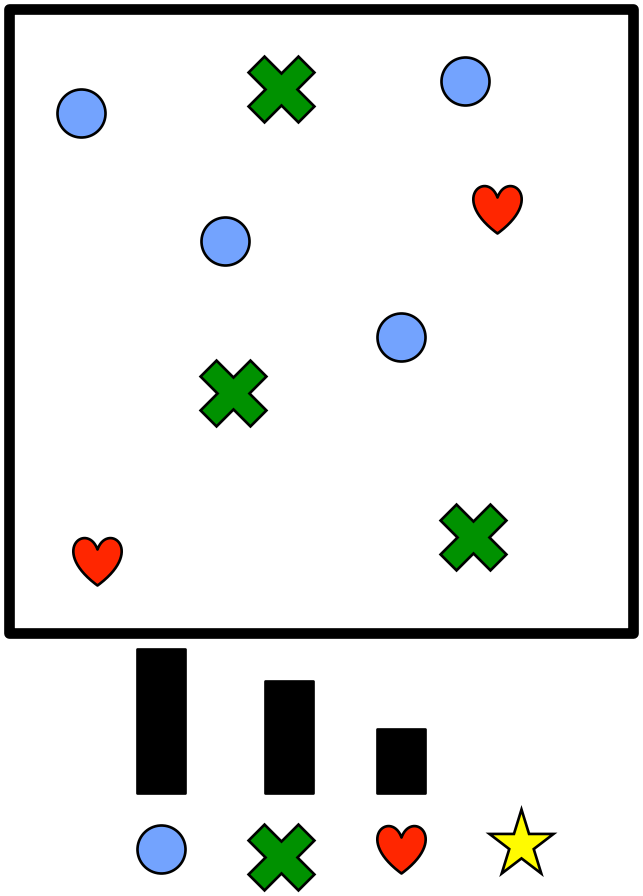
Vector Quantization Coding

Locality-Constrained Linear Coding

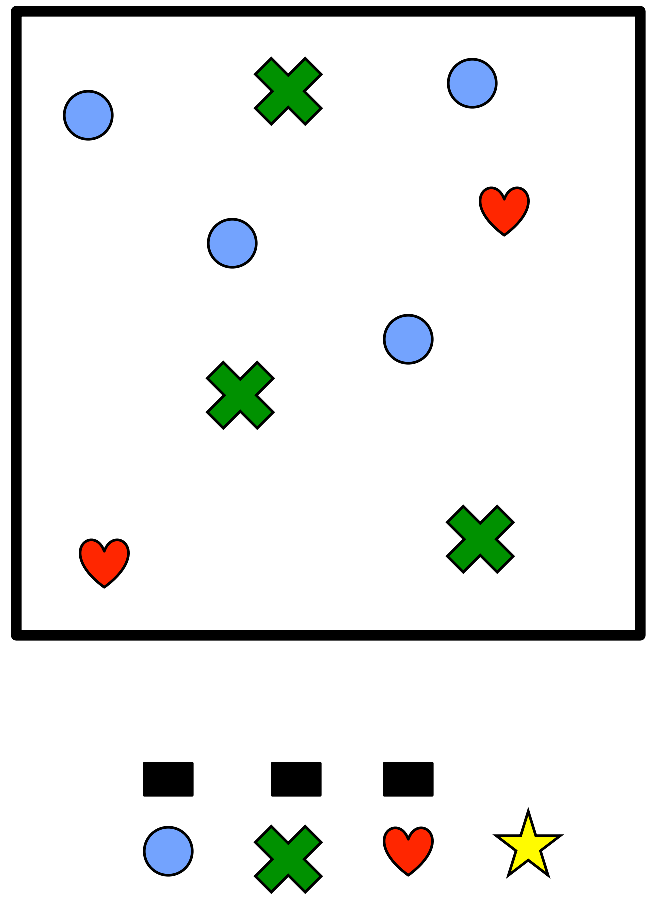
Generated Images from the average patch associated with each visual word

Bag of Words: Pooling

Sum Pooling

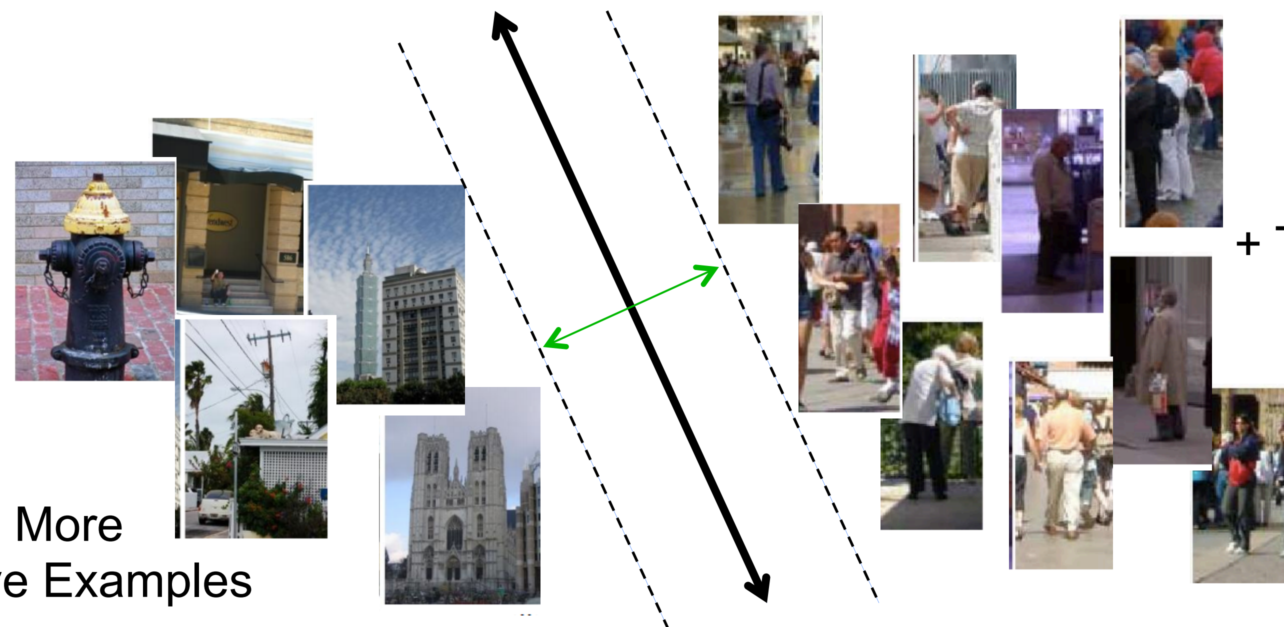


Max Pooling



Learning Detection Classifiers

- Extract Regions from Images
  - Containing the desired object
  - Containing everything other than the desired object
- Compute Feature Vector for Each Region
- Train SVM
  - Linear vs. Non-linear Kernel



+ Thousands More Positive Examples

+ Millions More Negative Examples

Bag of Words: Where it Works

- Texture Recognition
  - Texture is characterized by the repetition of basic elements or textons
  - For stochastic textures, it is the identify of the textons and not their spatial arrangement that matters

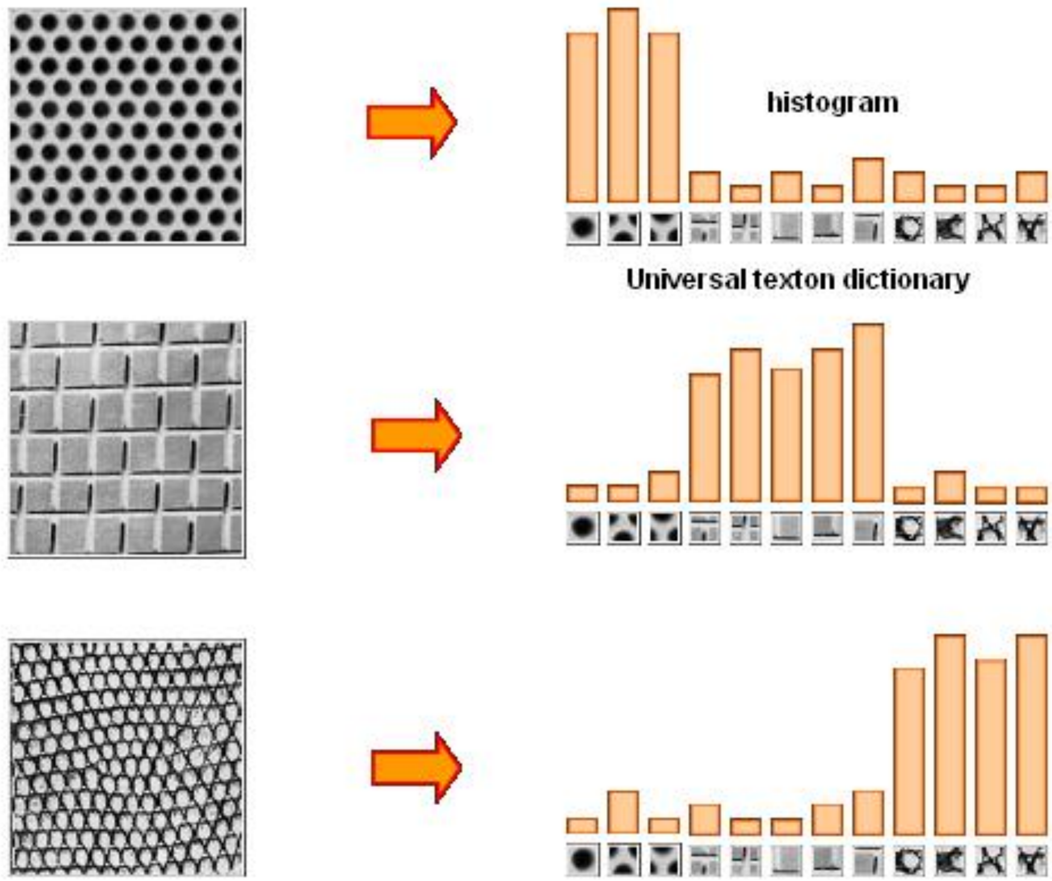
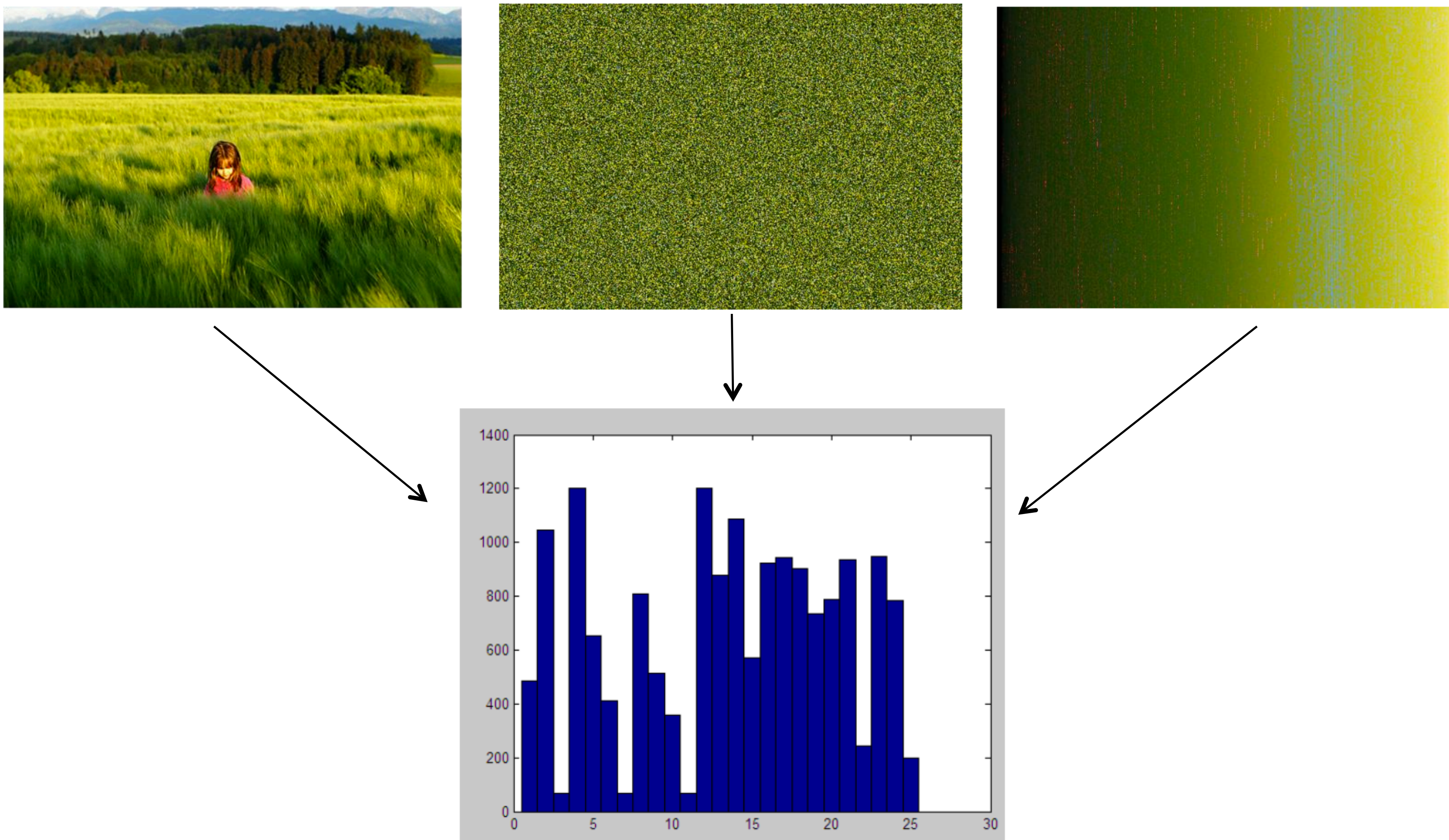


Image Credit: J. Malik

Bag of Words: Where it Fails



Slide Credit: J. Hays

Spatial Pyramids

Slide Credit: J. Hays

Spatial Pyramids

- Extension of the bag of words model
- Locally orderless representation at several levels of resolution
  - Some spatial information

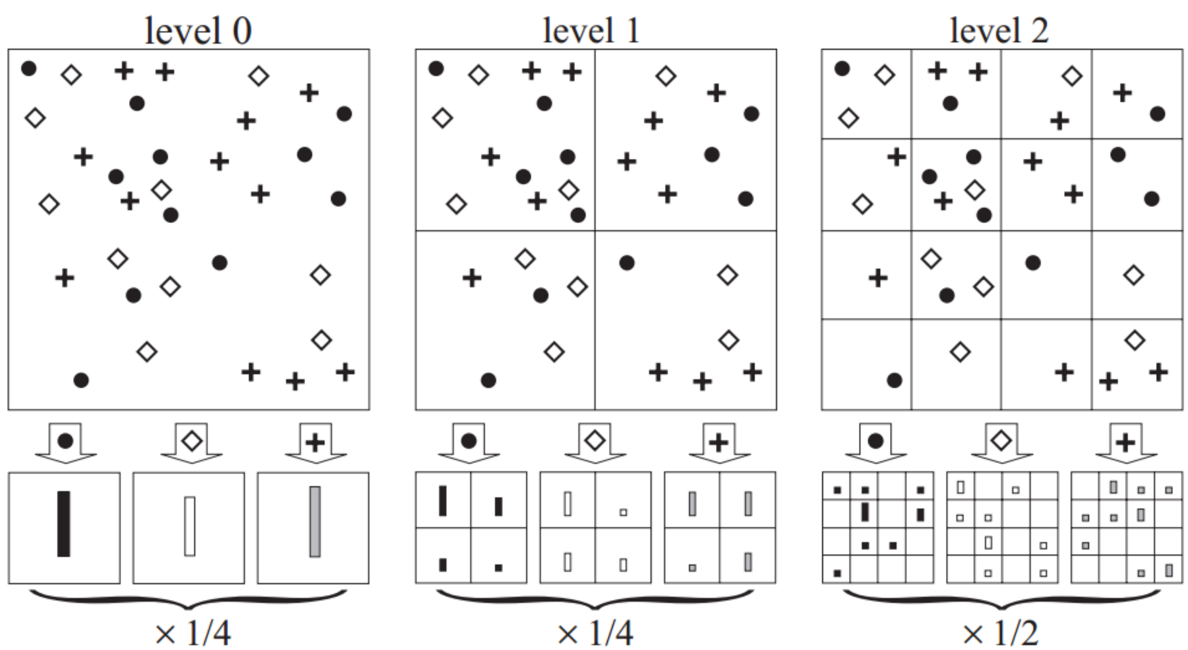
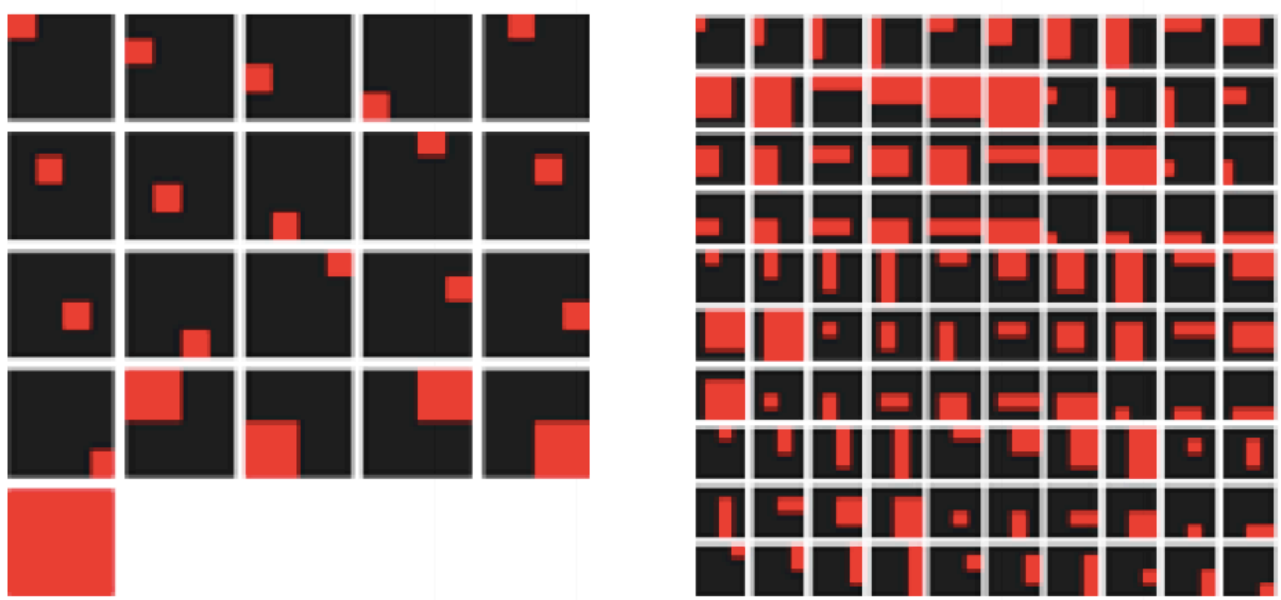


Image Credit: S. Lazebnik

Spatial Pyramids: Learning Receptive Fields

- Recent work in learning Receptive Fields rather than using a regular grid
  - Example Motivation: Sunset and Highway Images



Traditional Receptive Fields　　Learned Receptive Fields

Image Credit: Y. Jia

Rigid Template Models

- Find the HOG feature vector for each image
- Originally developed to used for pedestrian detection by Dalal and Triggs

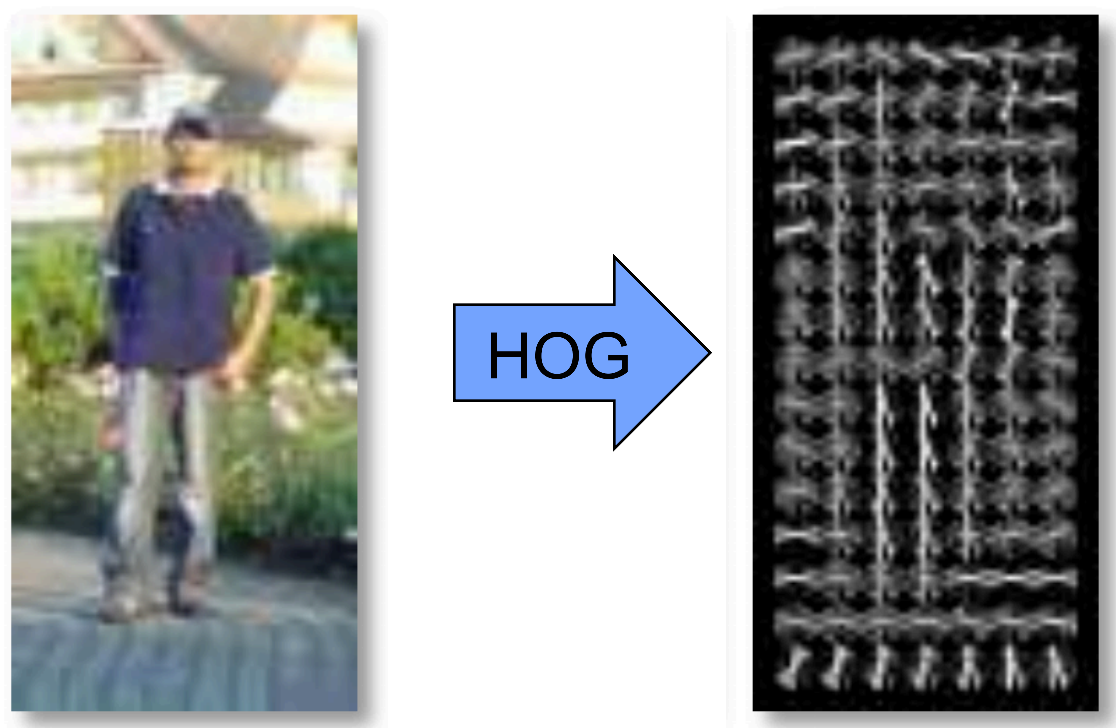


Image Credit: Dalal

Rigid Template Models: Detecting Objects

- Convolve low resolution HOG with the rigid template

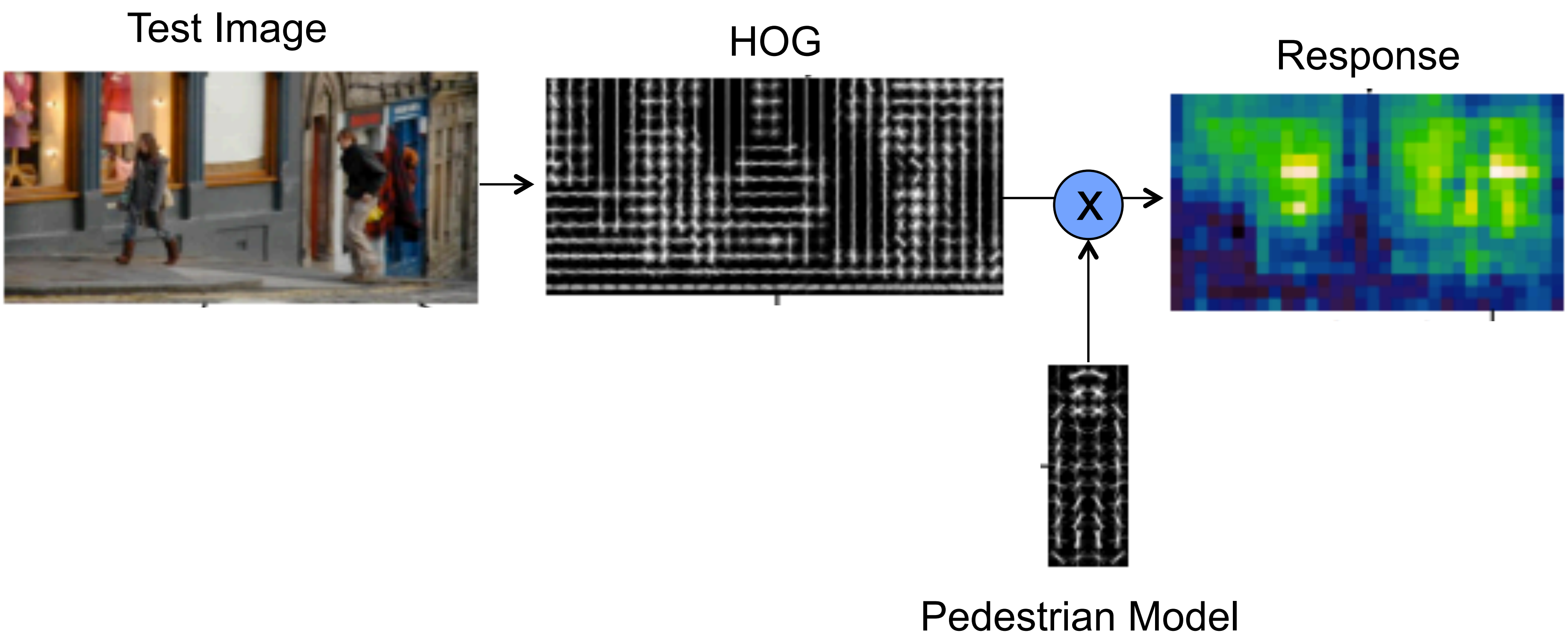
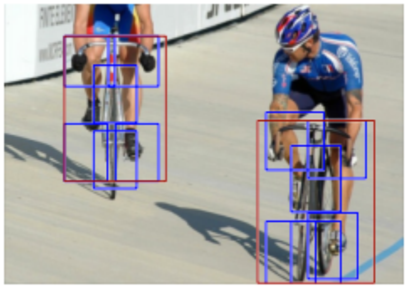
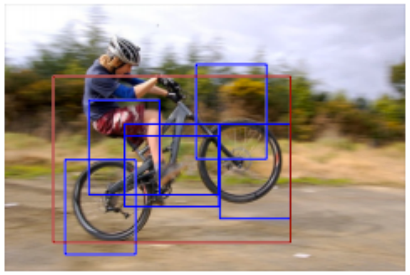


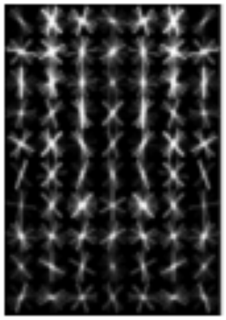
Image Credit: P. Felzenszwalb

Rigid Template Models: Mixture Models

- Objects take on different appearances
  - Pose
  - Multiple types of the same object
- For each object create a mixture model to capture the various appearances of the same object







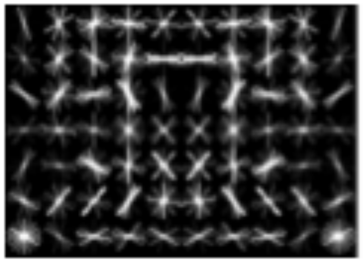
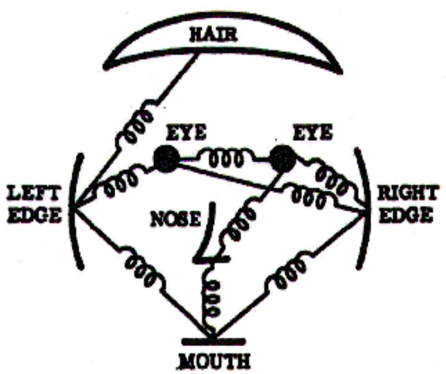


Image Credit: P. Felzenszwalb

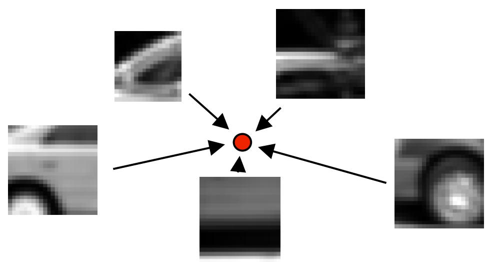
Structure Models

Part Models



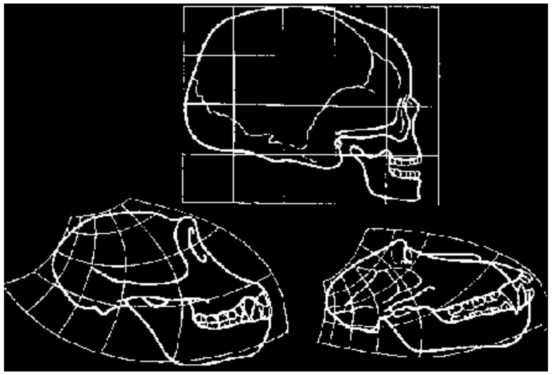
- Few parts (~6)

Voting models



- Many patches (>100)

Deformable Models



- No parts

Slide Credit: A. Torralba

Part Based Models

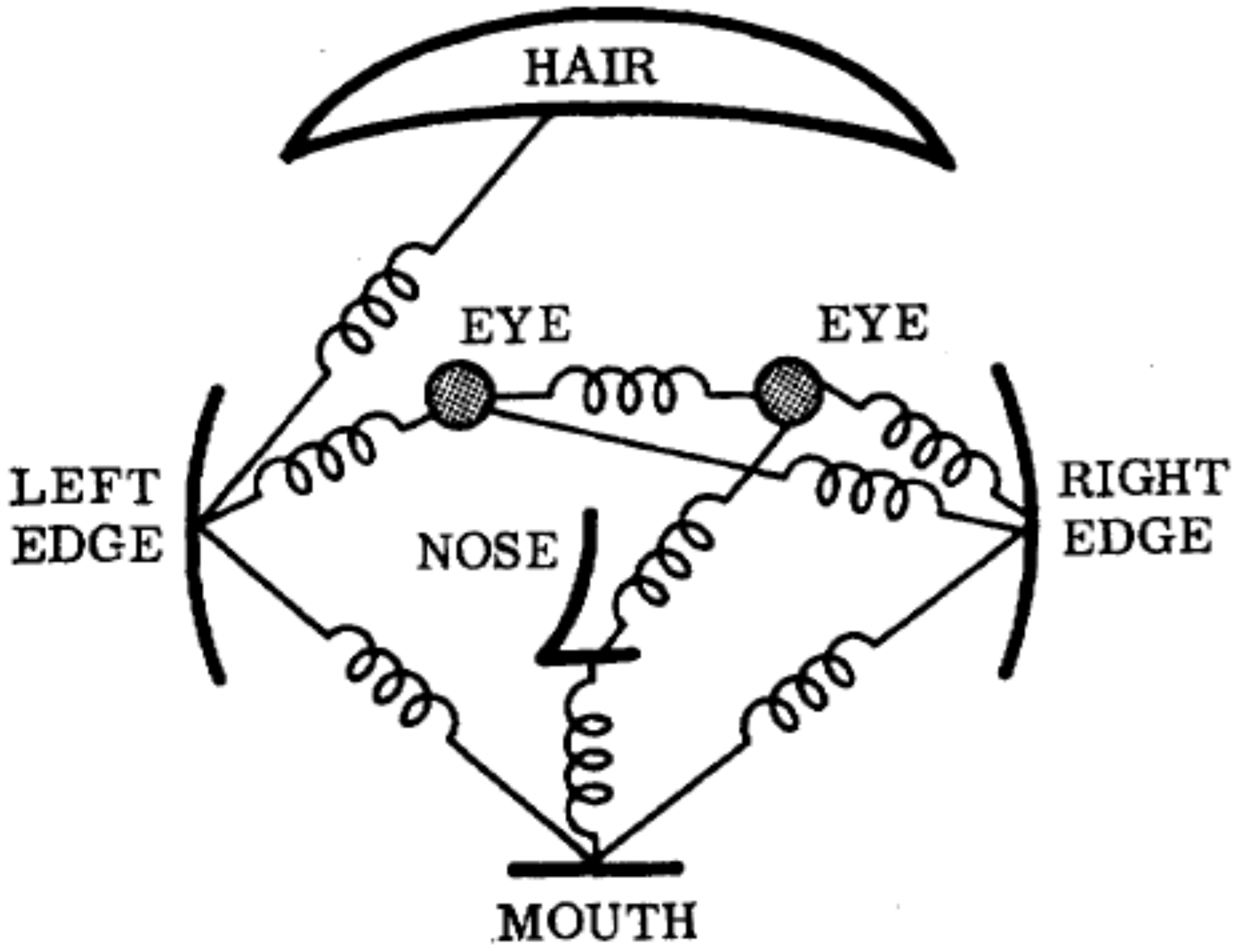
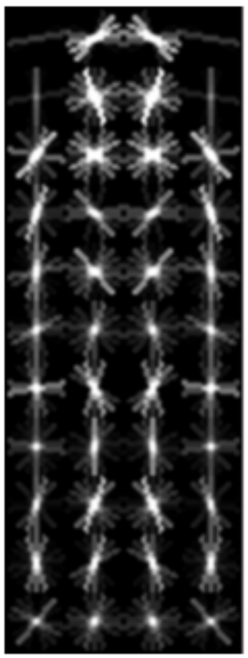


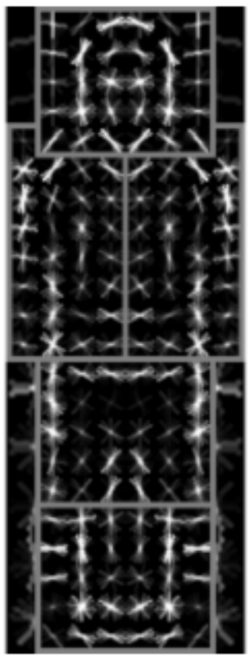
Image Credit: Fischler & Elschlager

Part Based Model

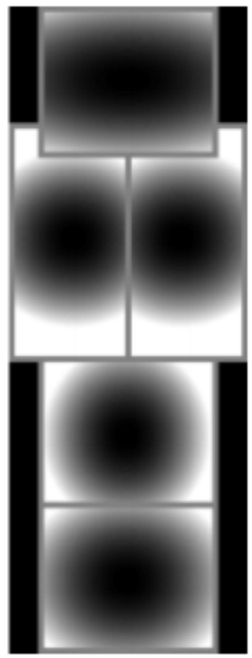
- Models an object as a number of smaller parts that are allowed to deviate slightly from "average" appearance
  - Star model - coarse root and higher resolution part filters



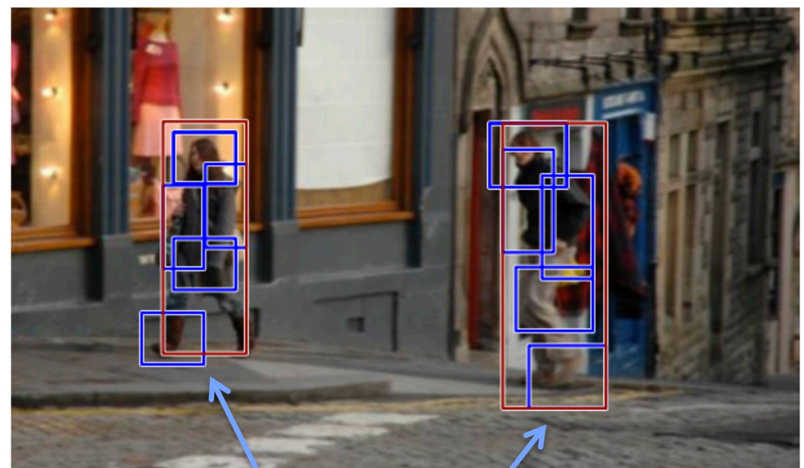
Root Filter



Part Filters



Spatial Model for location of each part relative to root

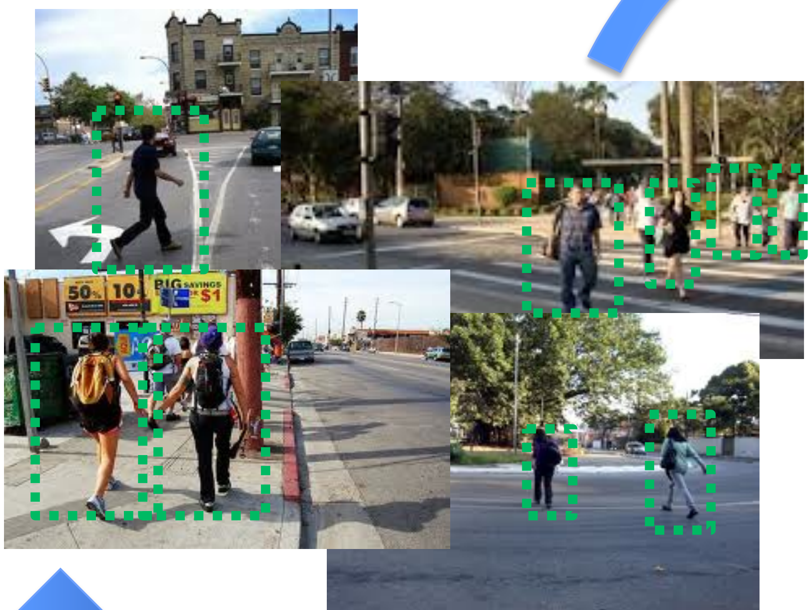


Parts are not positioned the same in each detection

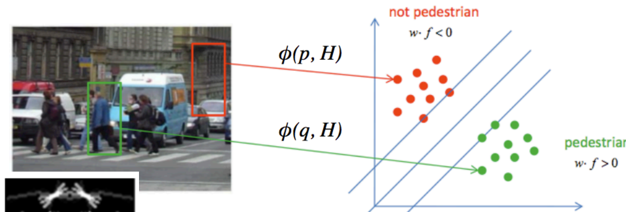
Image Credit: P. Felzenszwalb

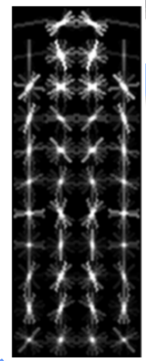
Part Based Model: Training using Latent SVM

START

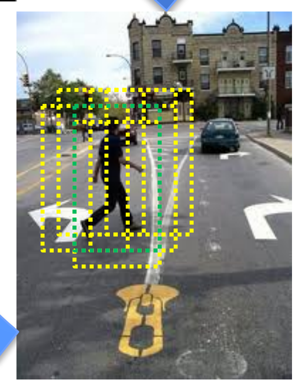


Initialize Part Locations





Use SVM to find classifier

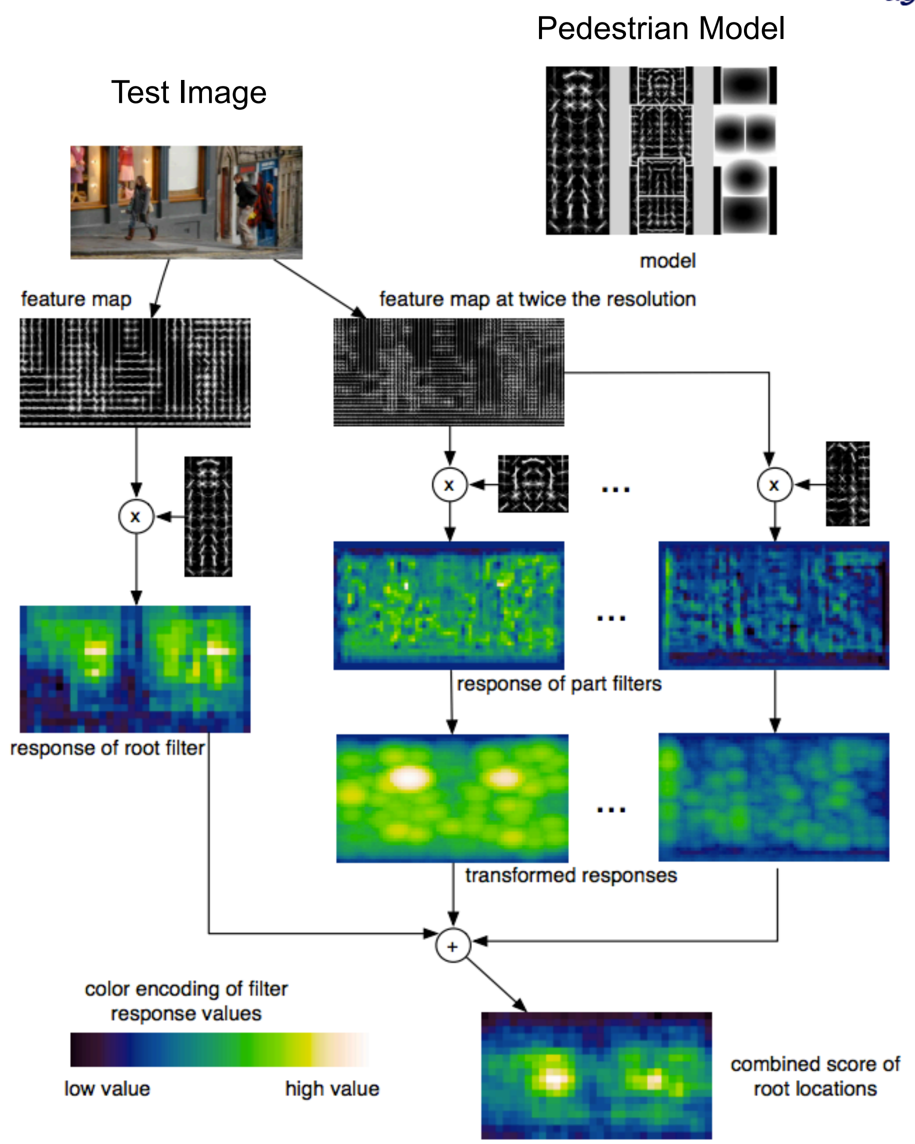


Use classifier to estimate the latent variable – part locations

Part Based Model: Detecting Objects

Extract HOG Features at multiple scales

Convolve low resolution HOG with the root filter and high resolution HOG with the parts filters



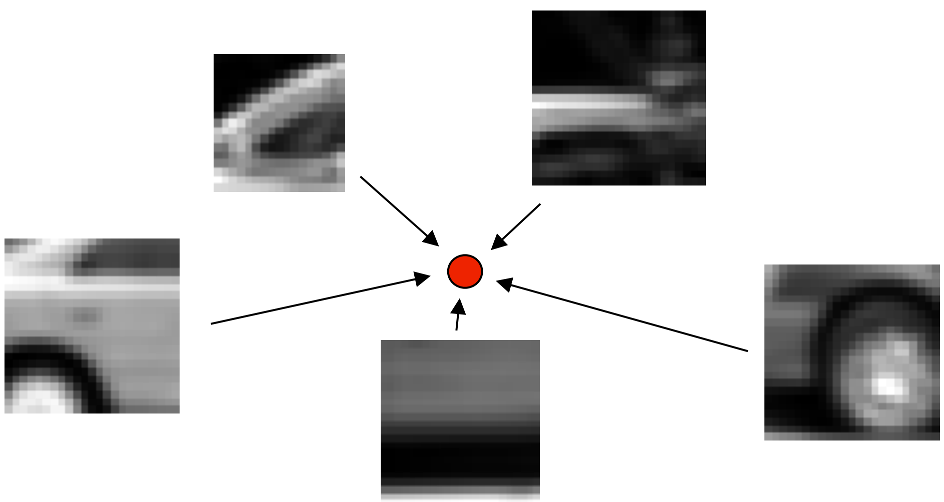
Transform part responses to point to where the star's root should be located

Add responses to find score for root locations

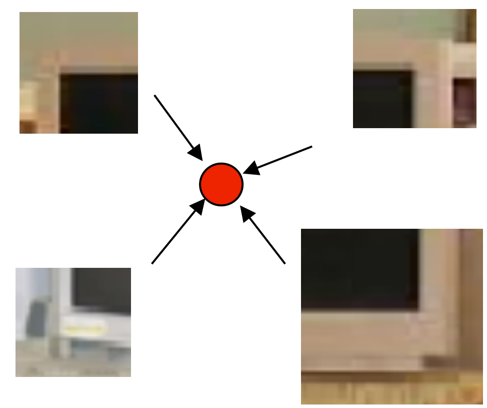
Image Credit: P. Felzenszwalb

Voting Models

- Create weak detectors by using parts and voting for the object's center location



Car model

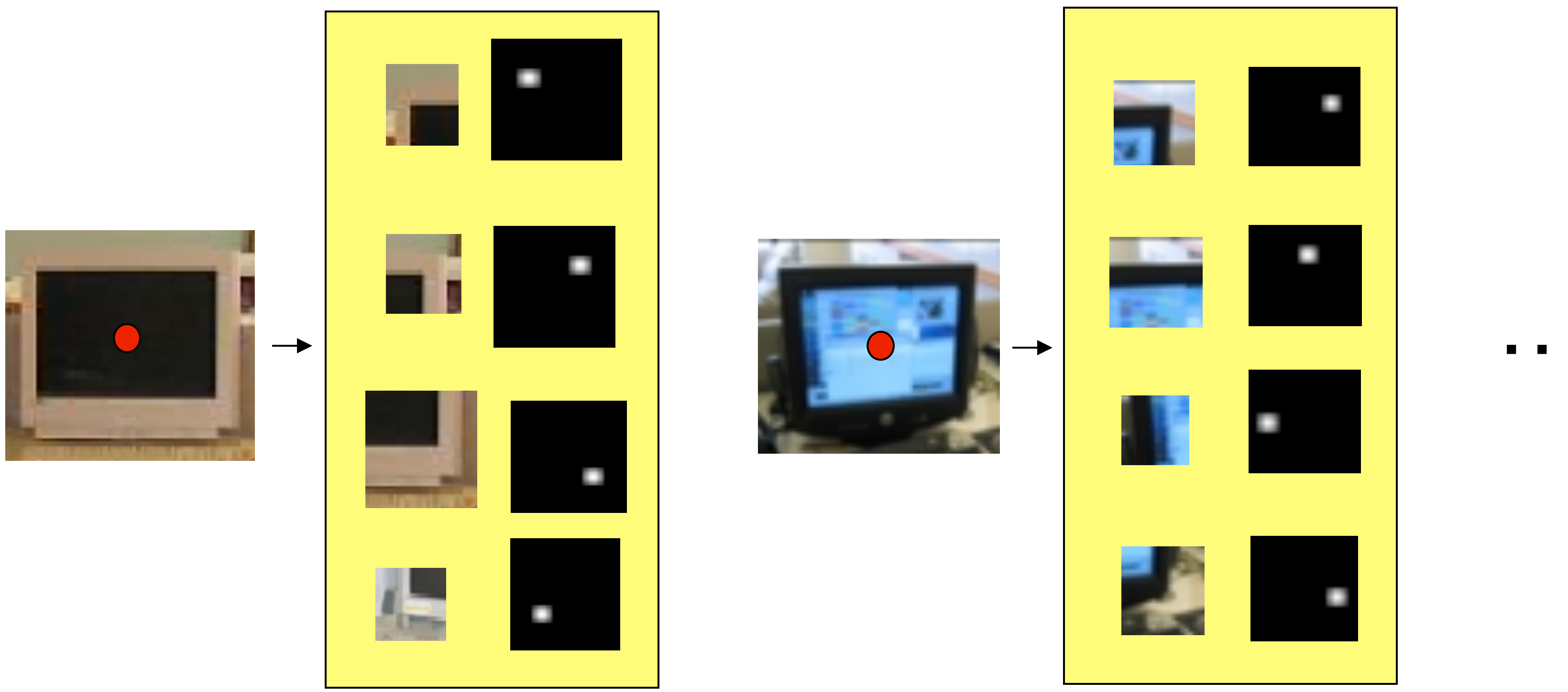


Screen model

Slide Credit: A. Torralba

Voting Models: Collecting Parts

First we collect a set of part templates from a set of training objects.

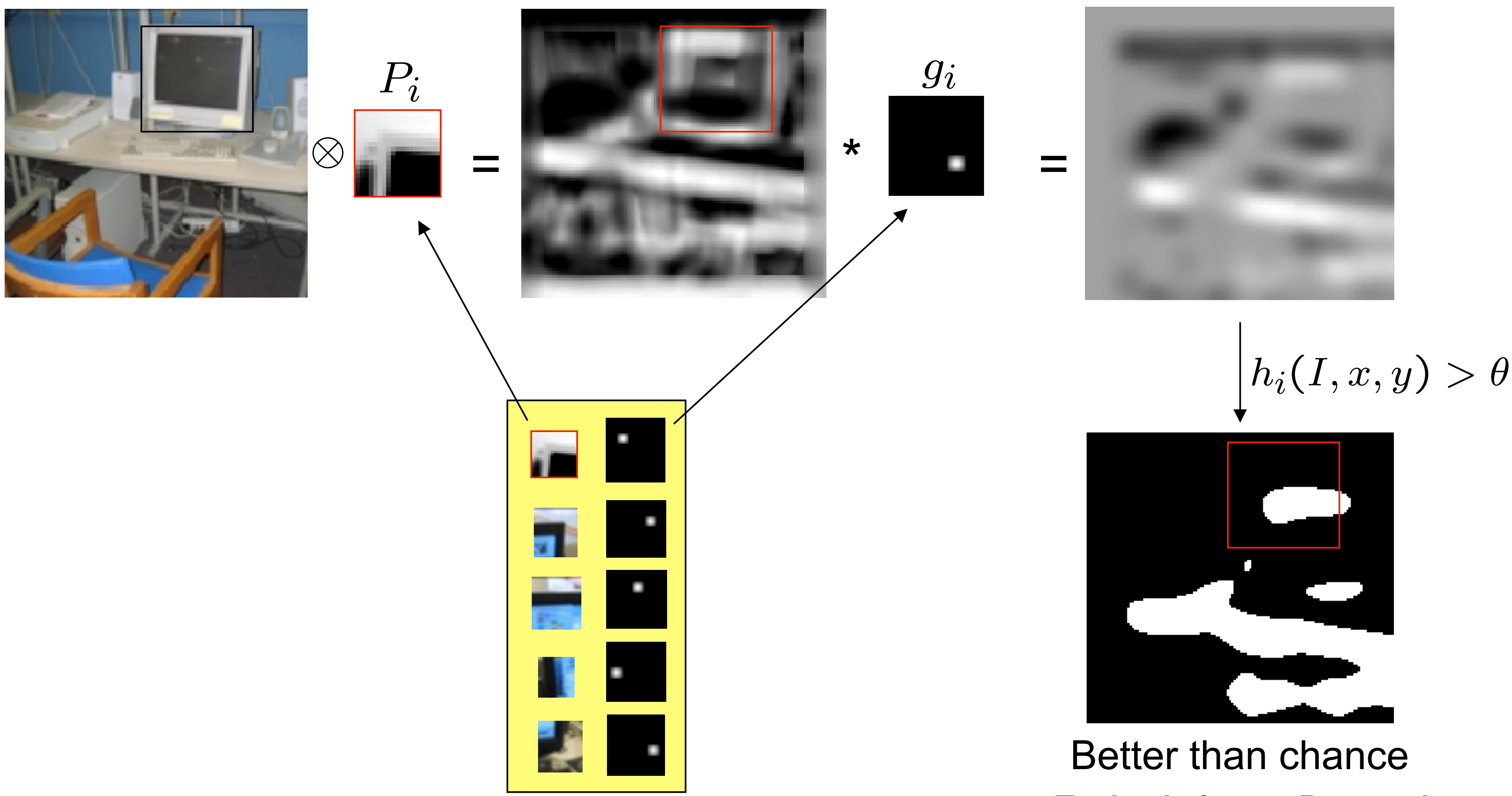


Slide Credit: A. Torralba

Voting Models: Weak Part Detectors

We now define a family of "weak detectors" as:

$$h_i(I, x, y) = [I \otimes P_i] * g_i$$



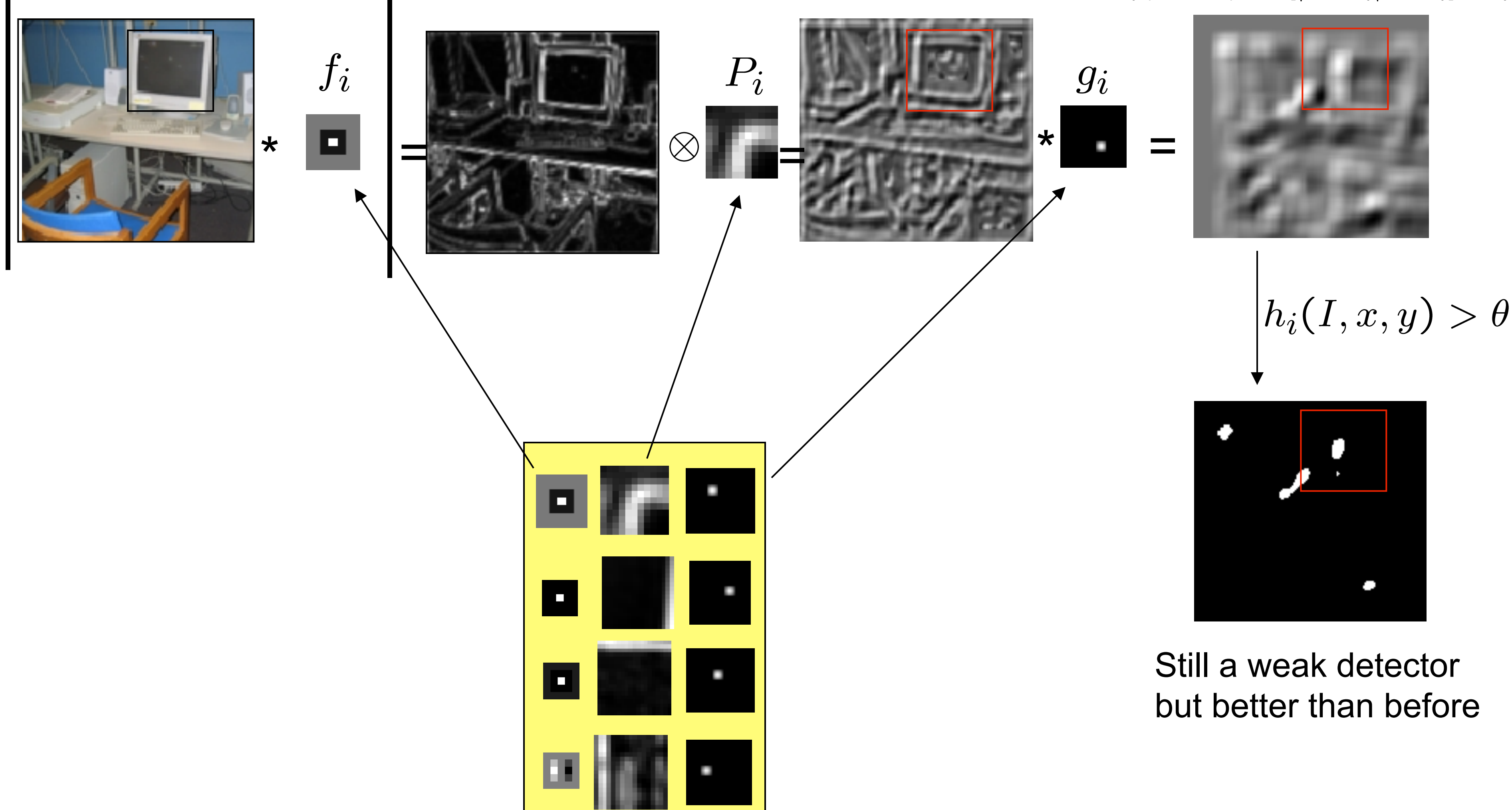
Better than chance

Slide Credit: A. Torralba

Voting Models: Weak Part Detectors

We can do a better job using filtered images

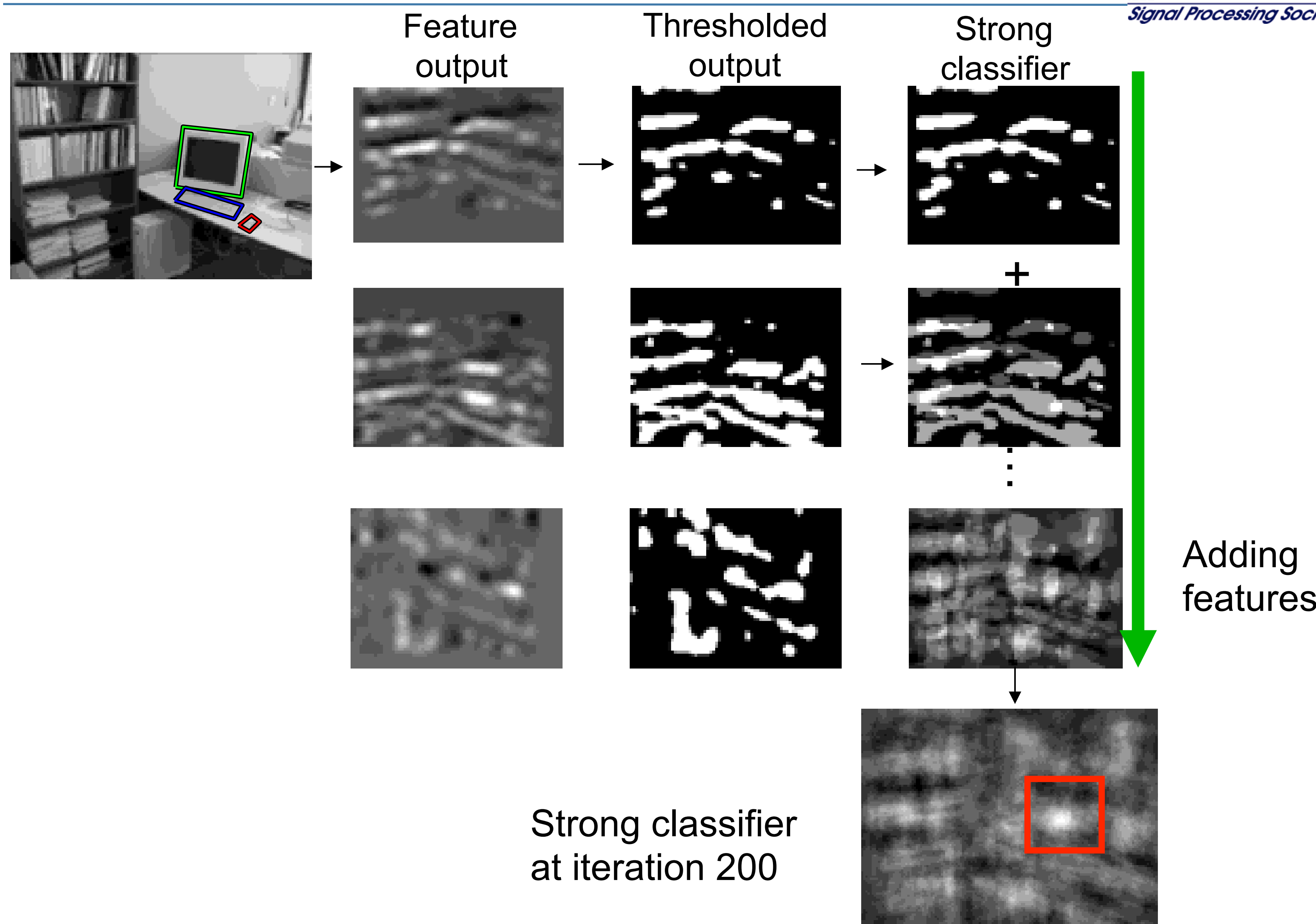
$$h_i(I, x, y) = [|I * f_i| \otimes P_i] * g_i$$



Still a weak detector
but better than before

Slide Credit: A. Torralba

Voting Model Example: Screen Detection



Slide Credit: A. Torralba

Boosting

- Defines a classifier using an additive model

$$F(x) = \alpha_1 f_1(x) + \alpha_2 f_2(x) + \alpha_3 f_3(x) + \ldots$$

Strong classifier

Features vector

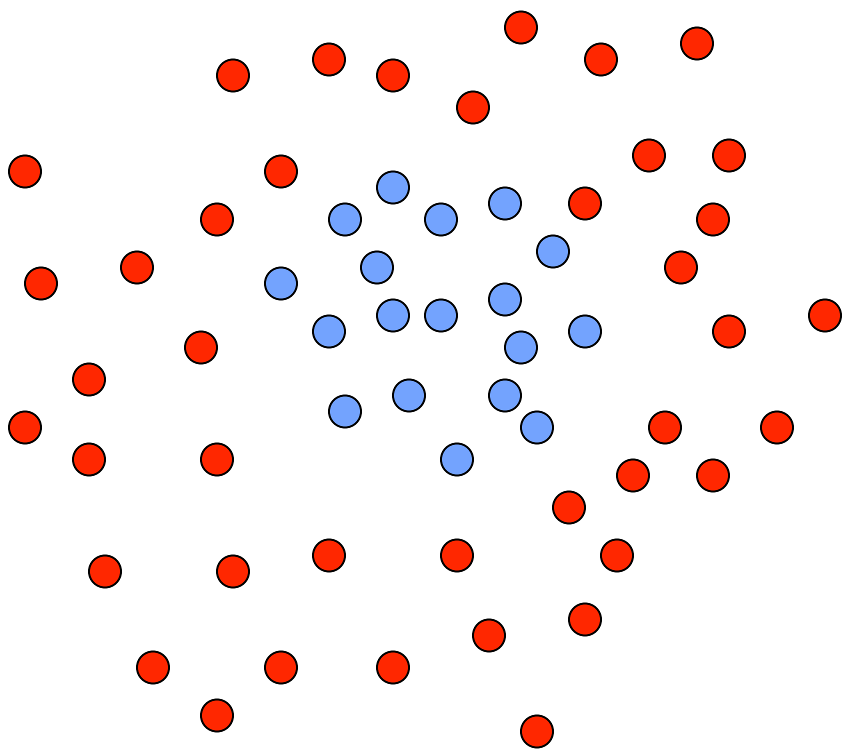
Weight

Weak classifier

Slide Credit: A. Torralba

Boosting Example

- **It is a sequential procedure:**



Each data point has a class label:

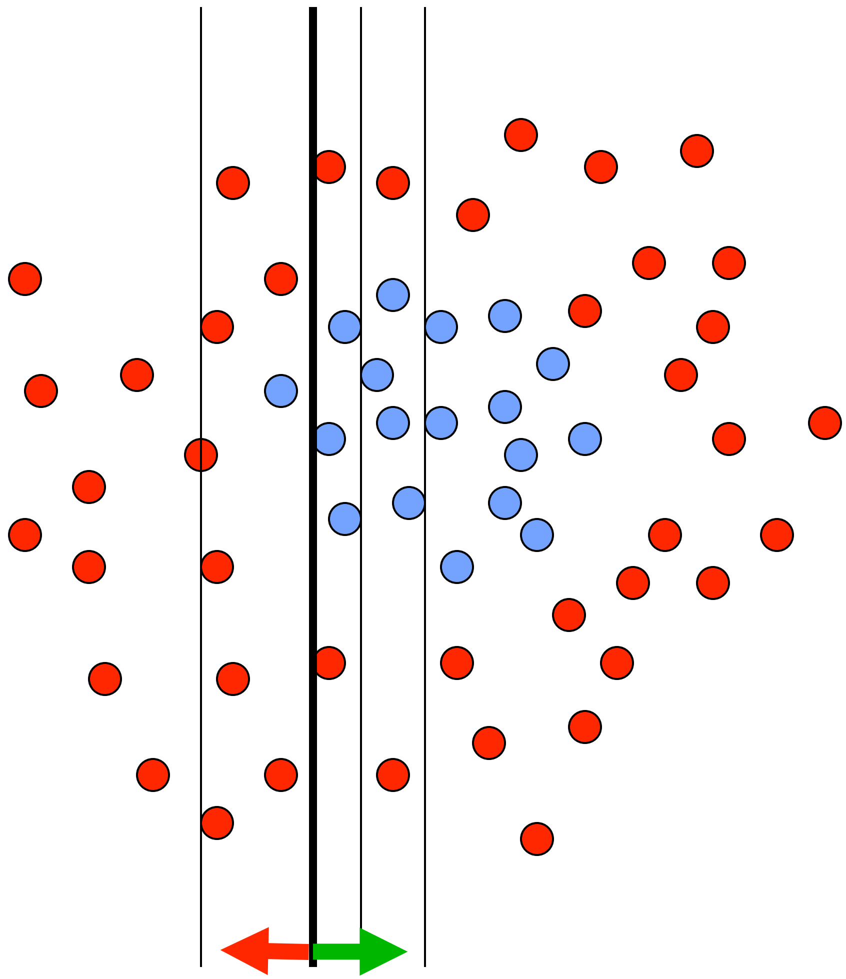
$$y_t = \begin{cases} +1 & (\text{red}) \\ -1 & (\text{blue}) \end{cases}$$

and a weight:

$$w_t = 1$$

Slide Credit: A. Torralba

Boosting Example



Each data point has a class label:

$$y_t = \begin{cases} +1 \ (\text{red}) \\ -1 \ (\text{blue}) \end{cases}$$

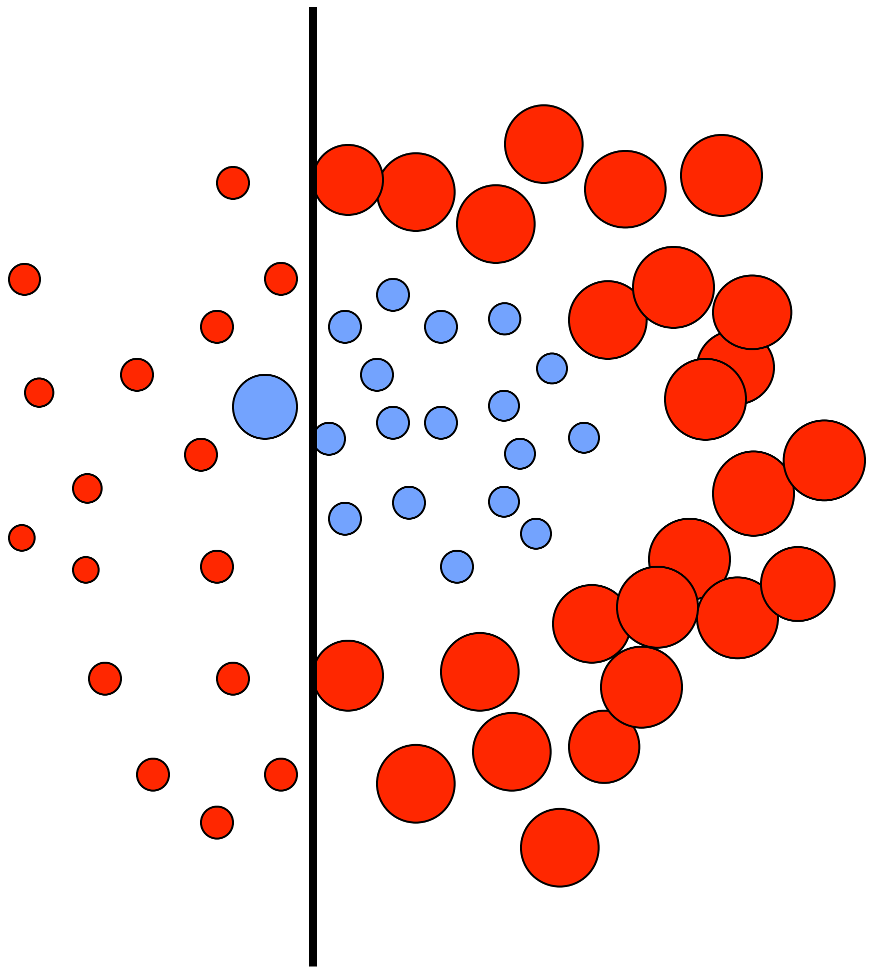
and a weight:

$$w_t = 1$$

This classifier seems to be the best

Slide Credit: A. Torralba

Boosting Example



Each data point has a class label:

$$y_t = \begin{cases} +1 & (\text{red}) \\ -1 & (\text{blue}) \end{cases}$$

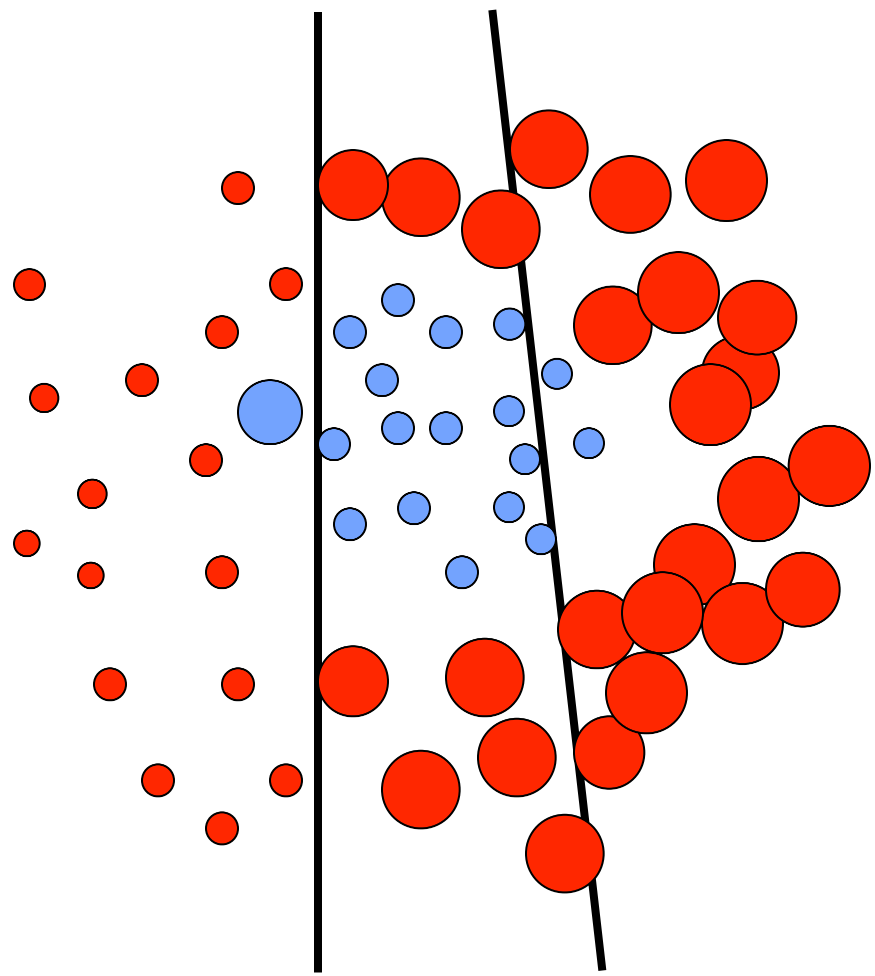
We update the weights:

$$w_t \leftarrow w_t \exp\{-y_t H_t\}$$

We set a new problem for which the previous weak classifier performs at chance again

Slide Credit: A. Torralba

Boosting Example



Each data point has a class label:

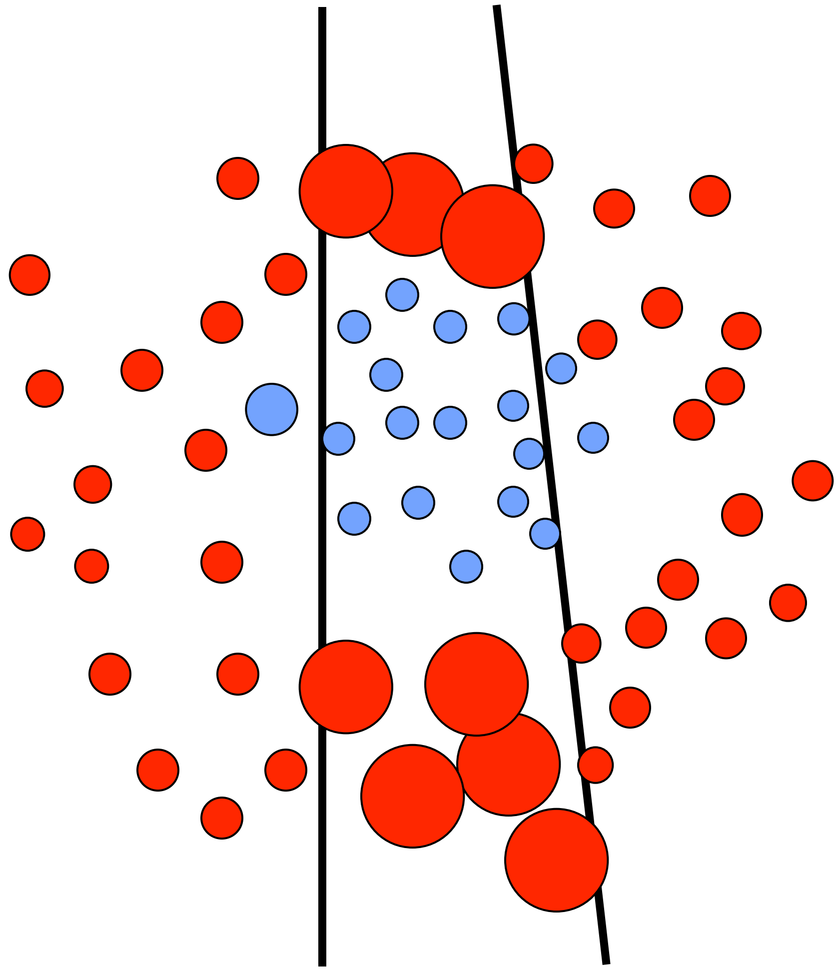
$$y_t = \begin{cases} +1 & (\text{red}) \\ -1 & (\text{blue}) \end{cases}$$

$$w_t \leftarrow w_t \exp\{-y_t\, H_t\}$$

We set a new problem for which the previous weak classifier performs at chance again

Slide Credit: A. Torralba

Boosting Example



Each data point has a class label:

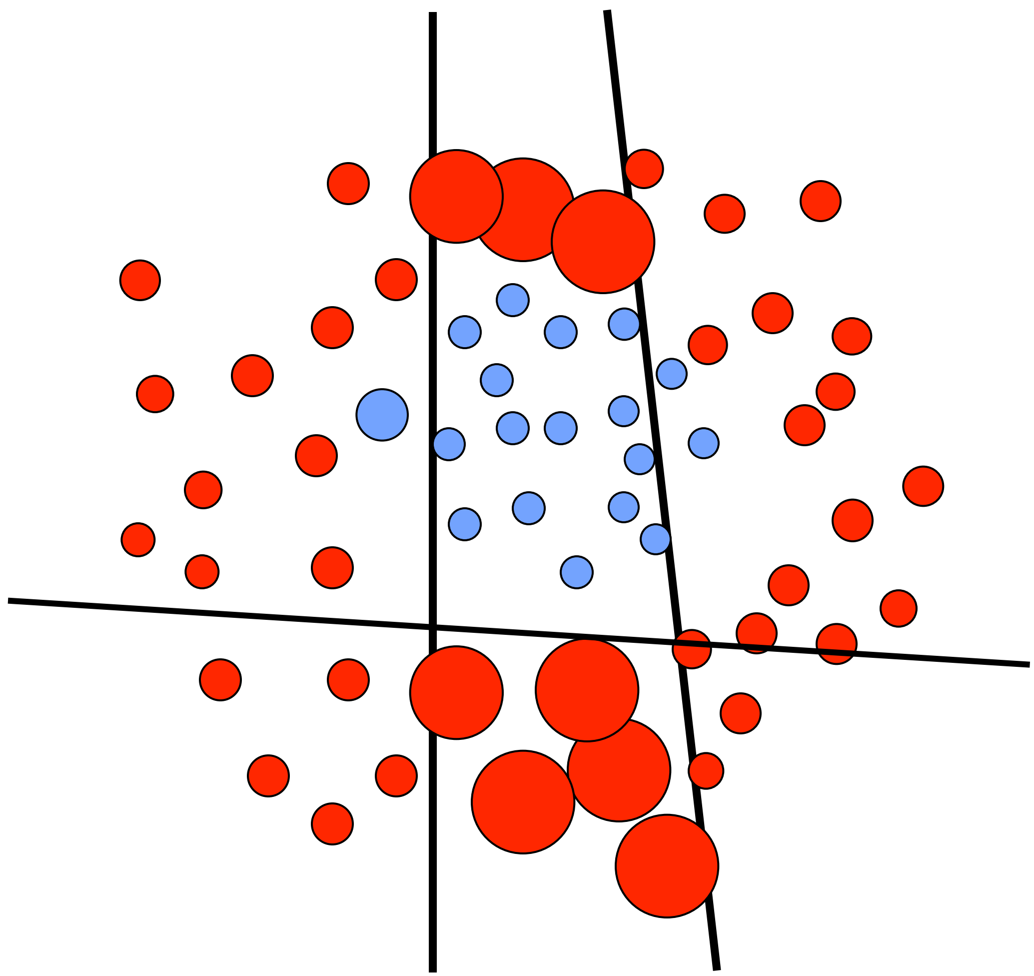
$$y_t = \begin{cases} +1 \ (\text{red}) \\ -1 \ (\text{blue}) \end{cases}$$

$$w_t \leftarrow w_t \exp\{-y_t H_t\}$$

We set a new problem for which the previous weak classifier performs at chance again

Slide Credit: A. Torralba

Boosting Example



Each data point has a class label:

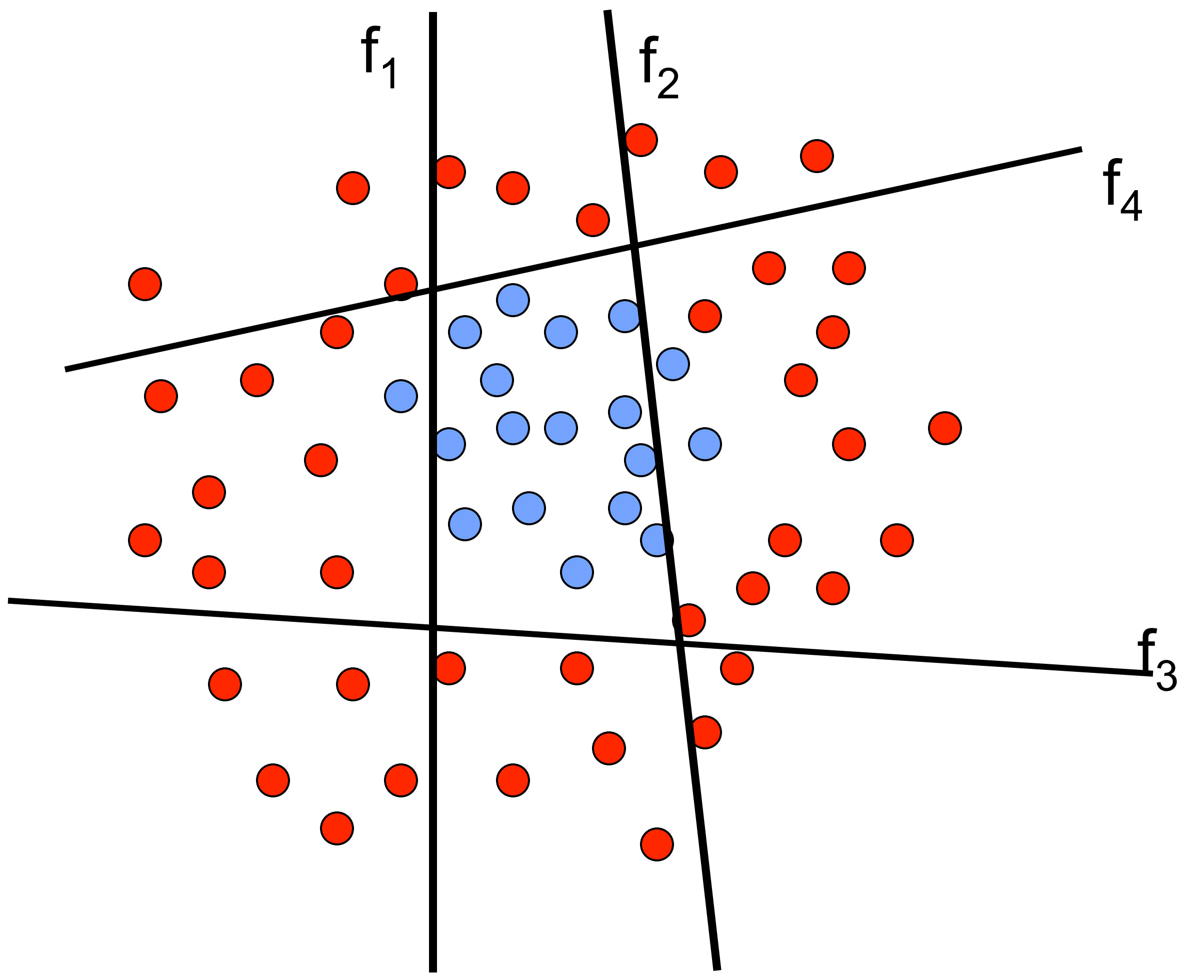
$$y_t = \begin{cases} +1 \ (\ ) \\ -1 \ (\ ) \end{cases}$$

$$w_t \leftarrow w_t \exp\{-y_t H_t\}$$

We set a new problem for which the previous weak classifier performs at chance again

Slide Credit: A. Torralba

Boosting Example



The strong (non- linear) classifier is built as the combination of all the weak (linear) classifiers.

Slide Credit: A. Torralba

Learning Detection Classifiers: Exemplar SVMs

- Learns a separate classifier for EVERY positive example (and millions of negative examples)
- At test time each classifier is applied to the image

vs.

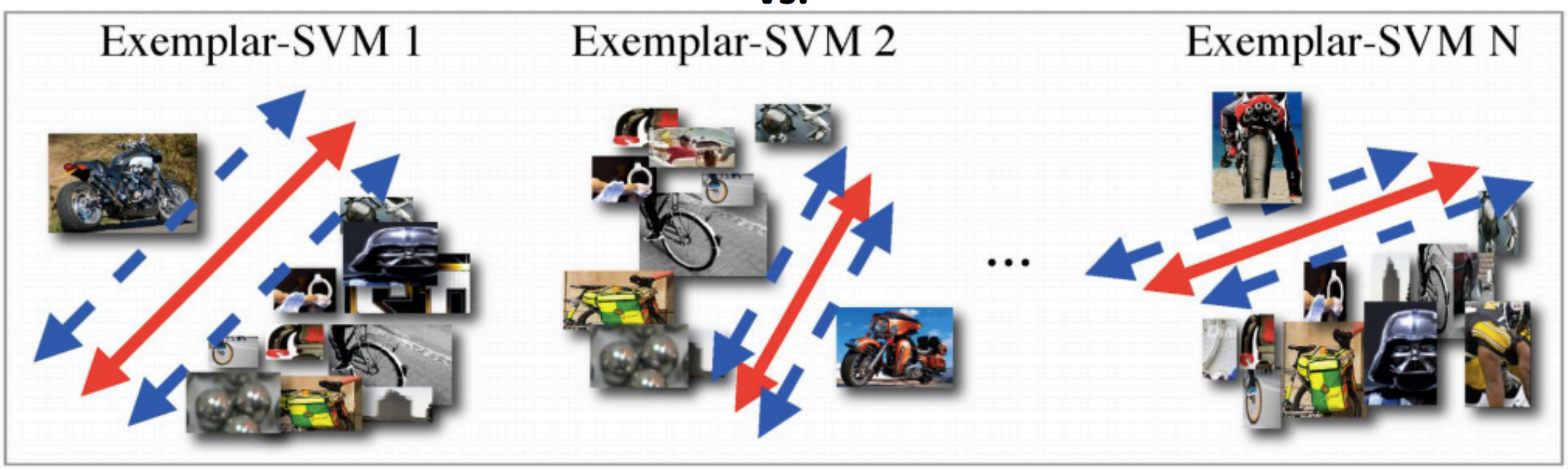


Image Credit: A. Malisiewicz

Learning Detection Classifiers: Exemplar SVMs

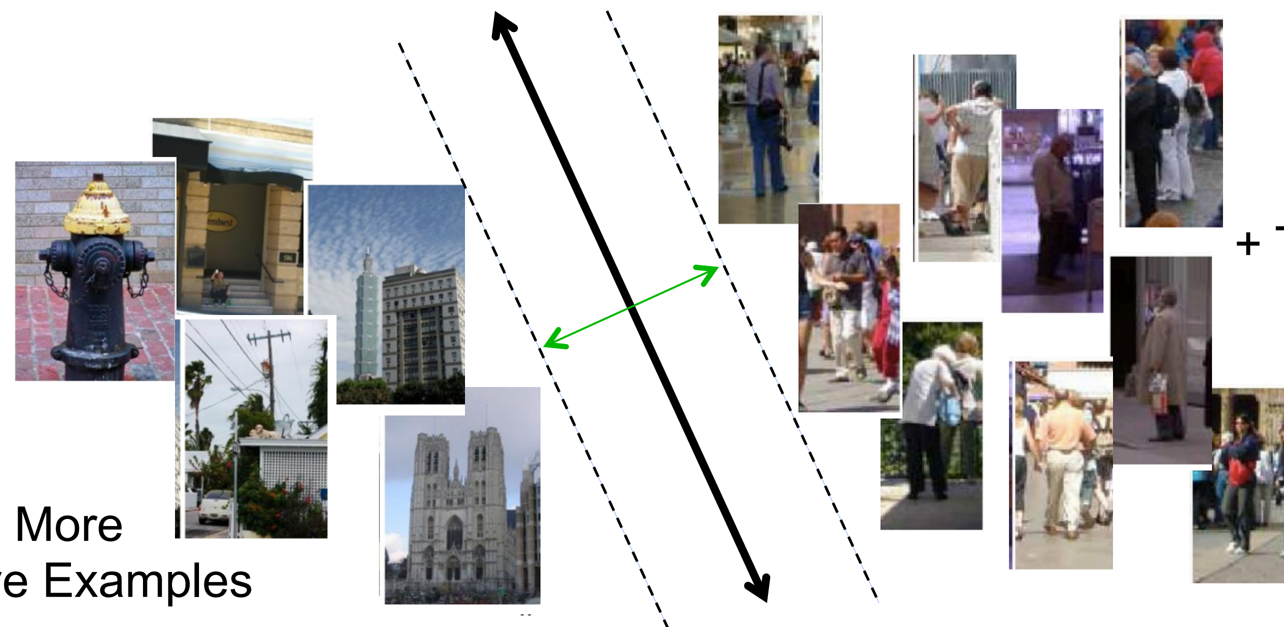
- Allows for more accurate correspondence and information transfer

Image Credit: A. Malisiewicz

How can we realistically implement an algorithm?

Learning Detection Classifiers

- Extract Regions from Images
  - Containing the desired object
  - Containing everything other than the desired object
- Compute Feature Vector for Each Region
- Train SVM
  - Linear vs. Non-linear Kernel



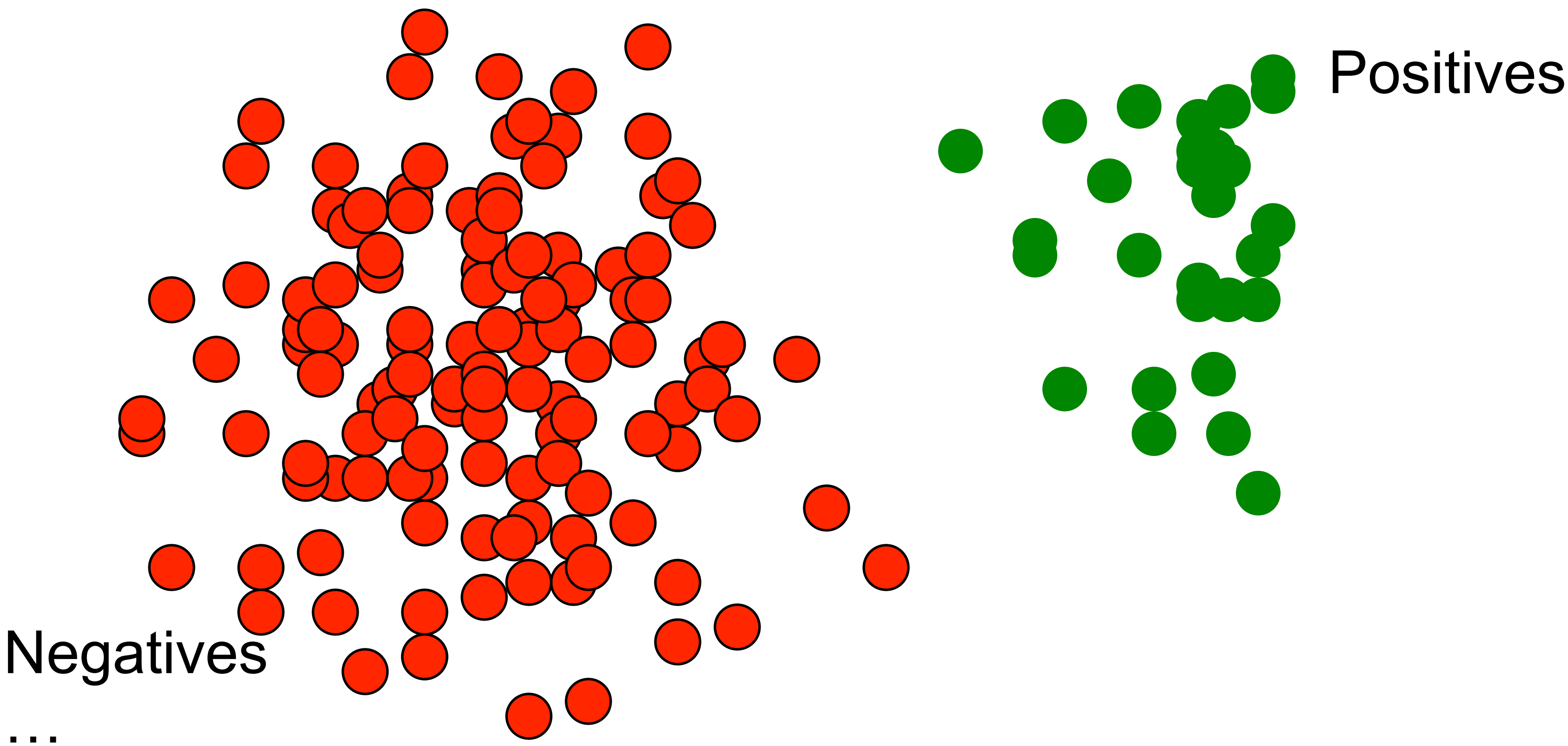
+ Thousands More Positive Examples

+ Millions More Negative Examples

Training SVMs using Hard Negative Mining

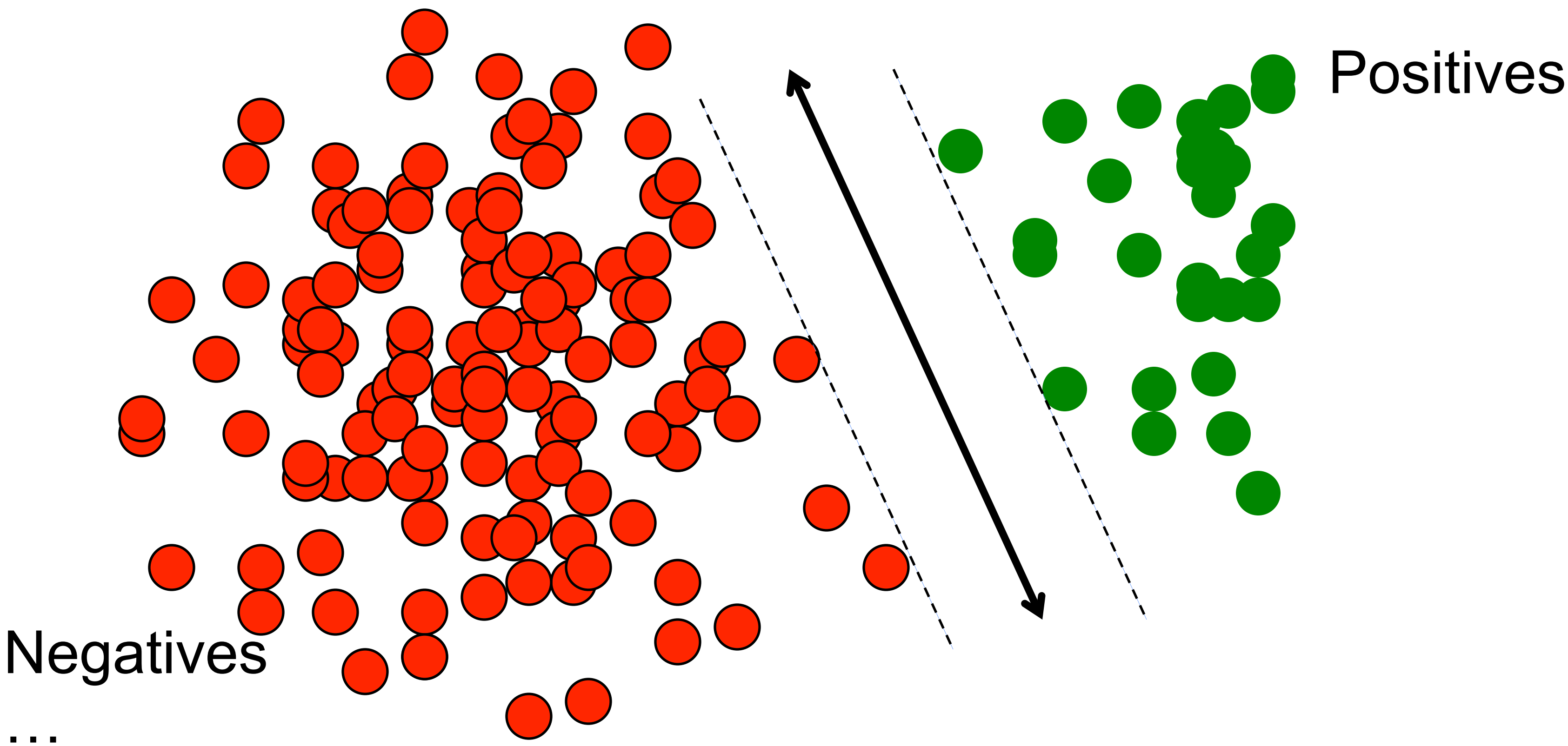
IEEE Signal Processing Society

- Too many negatives examples to train against all of them
  - Time and Memory Constraints
- Be smart about how you choose what negatives to train against



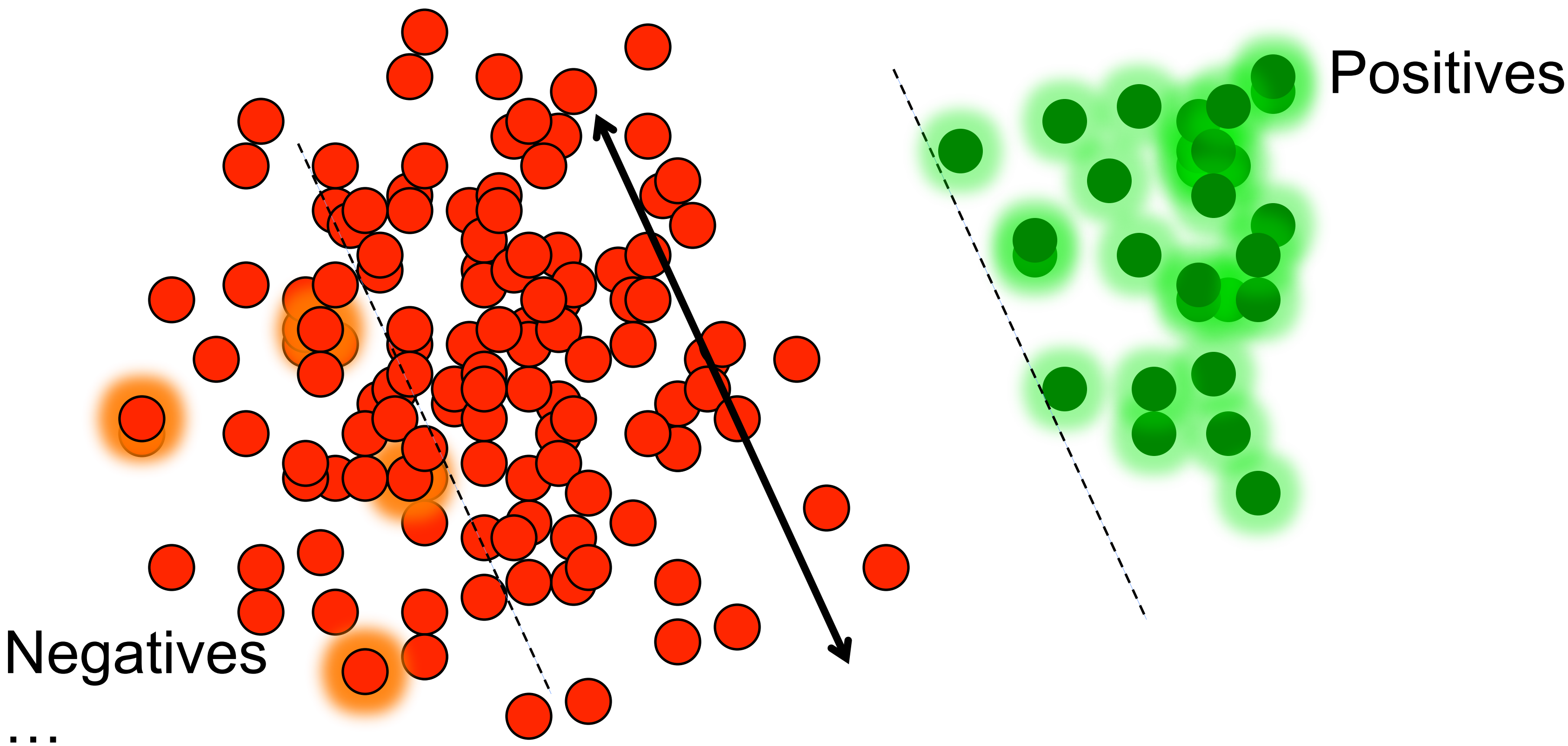
Training SVMs using Hard Negative Mining

- Too many negatives examples to train against all of them
  - Time and Memory Constraints
- Be smart about how you choose what negatives to train against



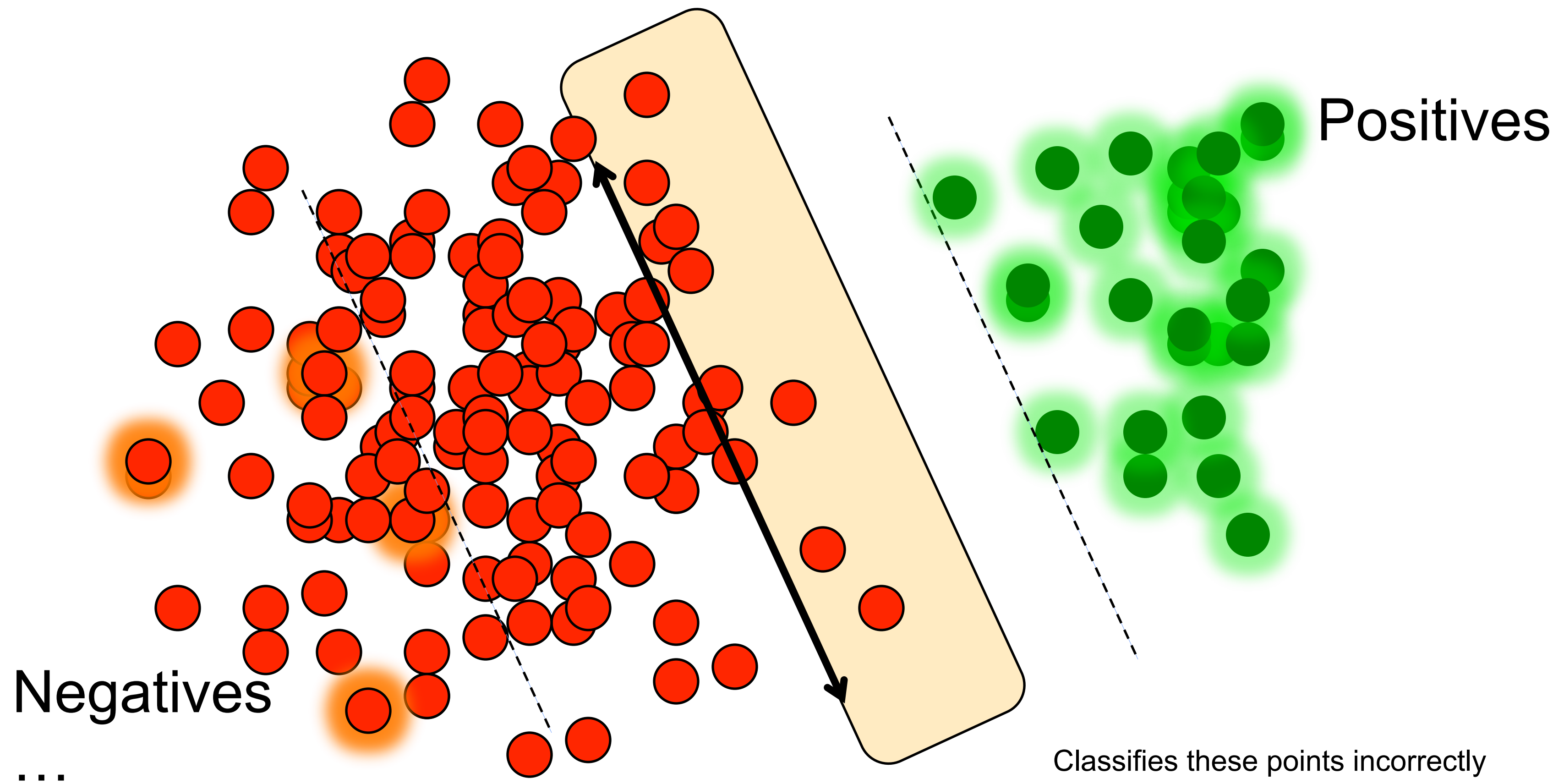
Training SVMs using Hard Negative Mining

- Too many negatives examples to train against all of them
  - Time and Memory Constraints
- Be smart about how you choose what negatives to train against



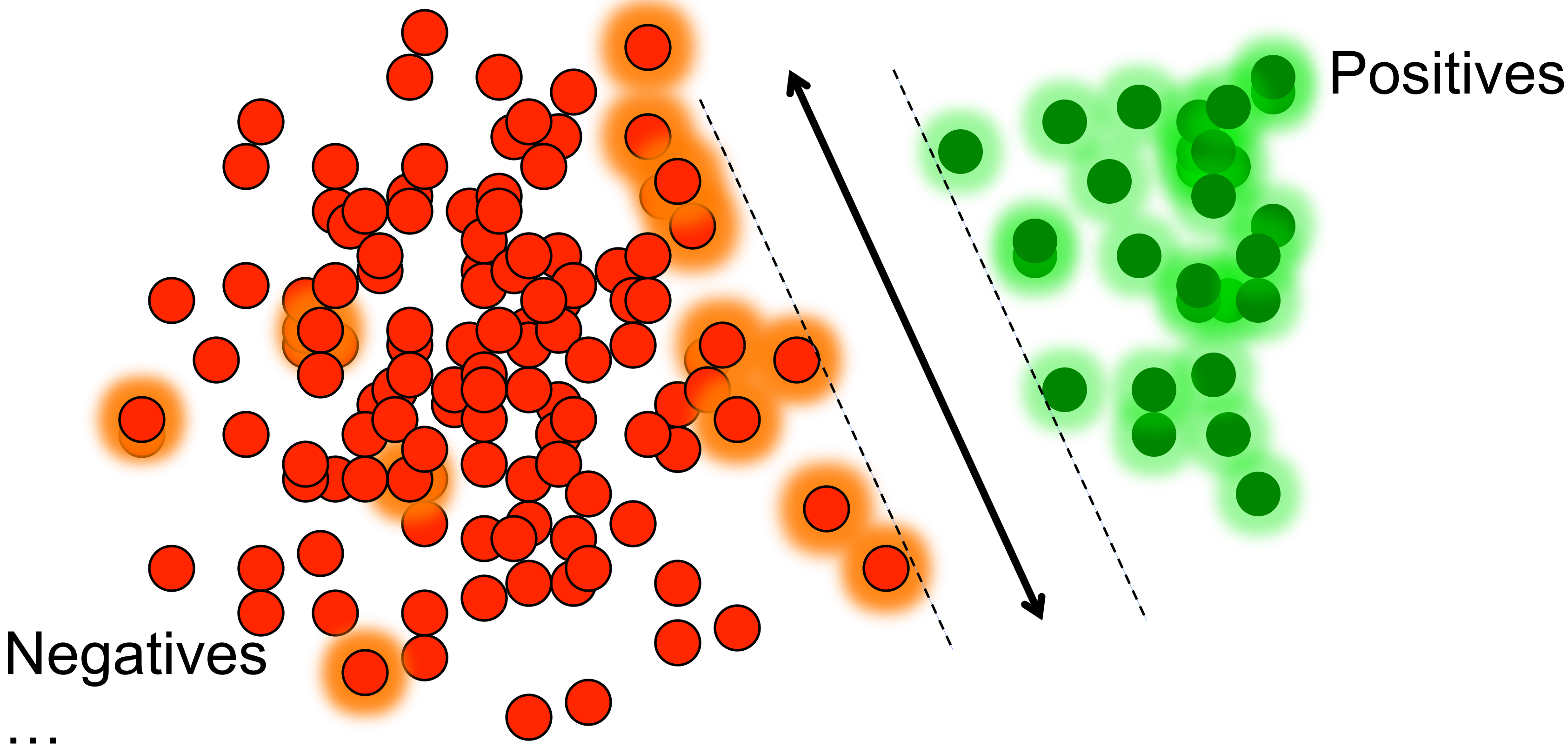
Training SVMs using Hard Negative Mining

- Too many negatives examples to train against all of them
  - Time and Memory Constraints
- Be smart about how you choose what negatives to train against



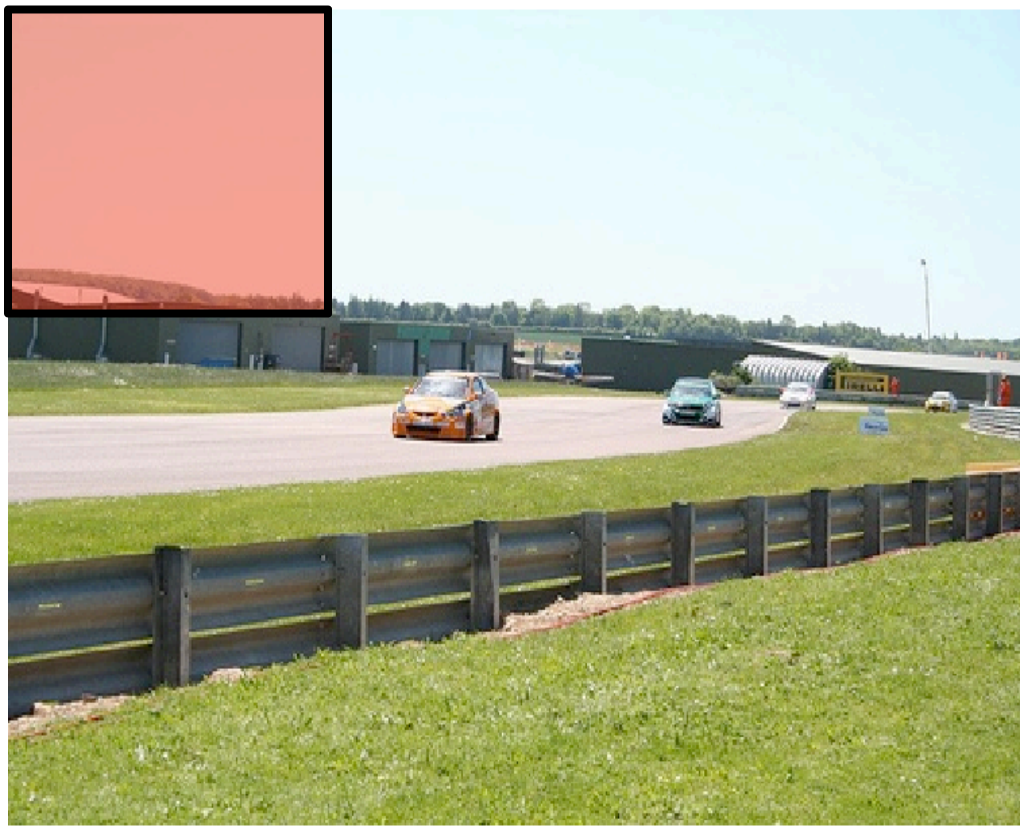
Training SVMs using Hard Negative Mining

- Too many negatives examples to train against all of them
  - Time and Memory Constraints
- Be smart about how you choose what negatives to train against



Detecting Objects

- Must run a classifier at every position at every scale in order to detect object



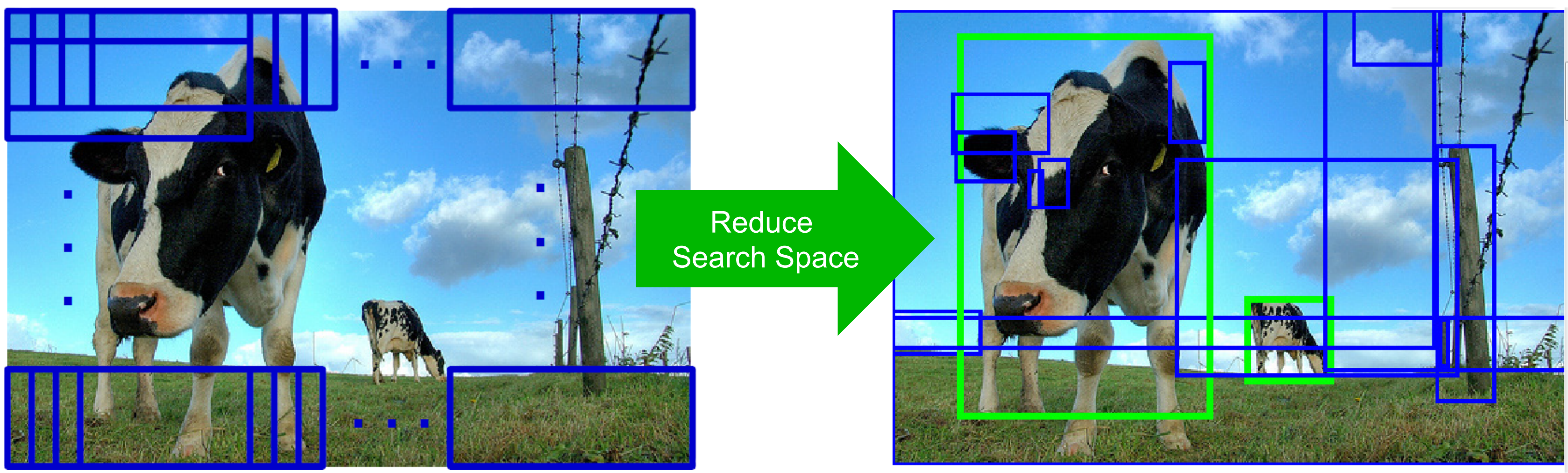
Detecting Objects

Detecting Objects

Detecting Objects

Generating Potential Object Regions

- Most algorithms rely on an exhaustive search to find object detections
  - Number of Pixels X Number of Scales
- Quickly find a smaller number of potential object bounding boxes to search in



Exhaustive Search

Selective Search

Image Credit: K. Van de Sande

Generating Potential Object Regions

- Segmentation Algorithm
  - Start with oversegmentation in a variety of color spaces

  - Group regions for each color space in a greedy fashion until the image is a single region

    Size and Texture

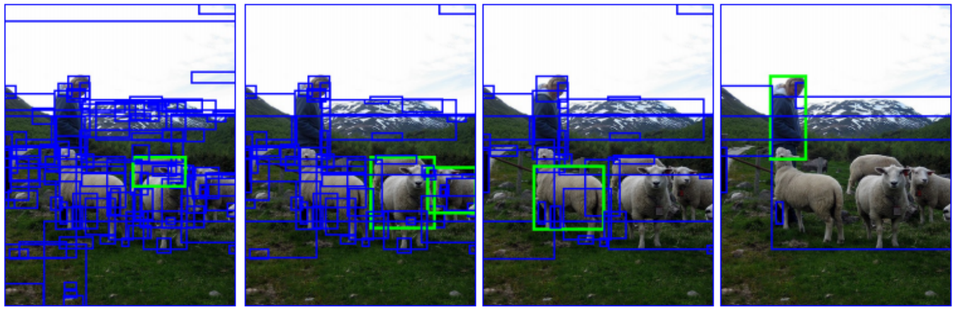
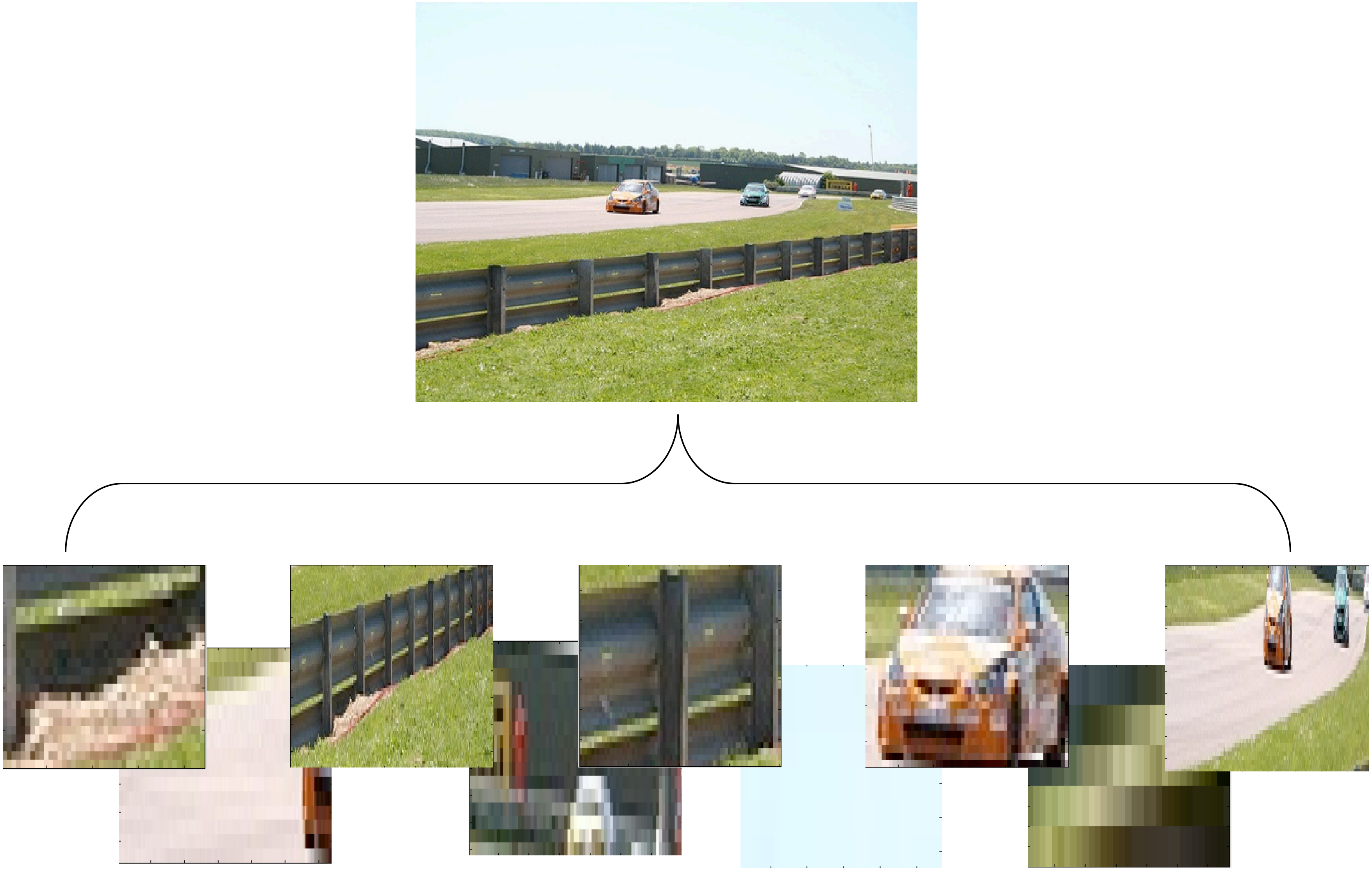
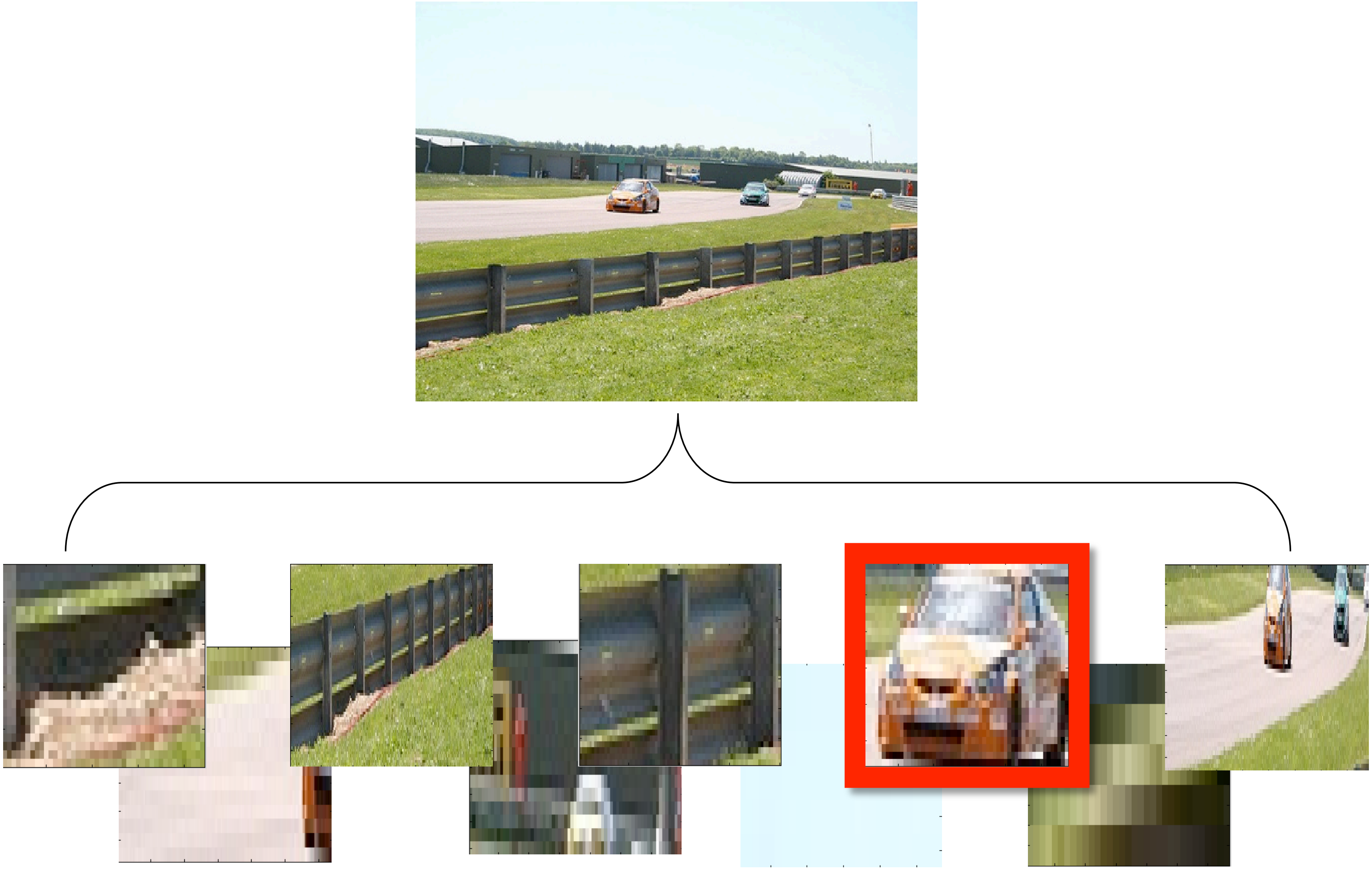


Image Credit: K. Van de Sande

Generating Potential Object Regions



Generating Potential Object Regions



Datasets for Object Classification/Detection

- Caltech101
- Caltech256
- PASCAL
- ImageNET
- LabelMe

Questions?