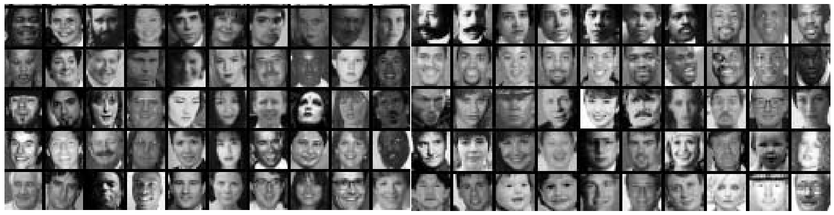


Sliding windows and face detection

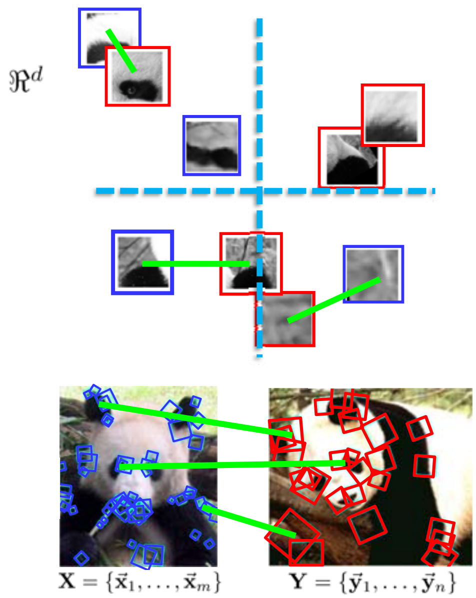
Tuesday, Nov 10

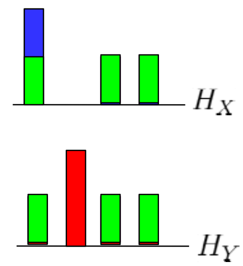
Kristen Grauman
UT-Austin

Last time

- Modeling categories with local features and spatial information:
  - Histograms, configurations of visual words to capture global or local layout in the bag-of-words framework
    - Pyramid match, semi-local features

Pyramid match



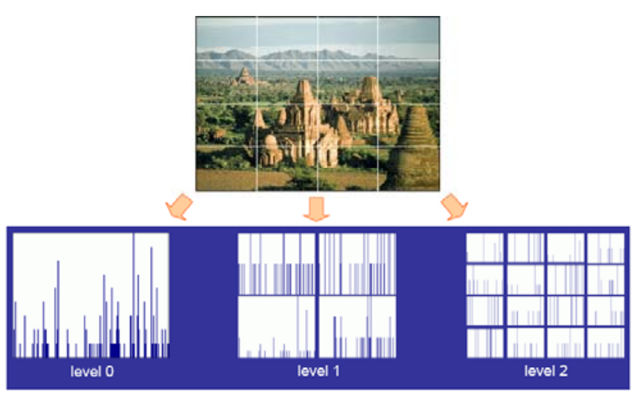


$$\mathcal{I}(H_X, H_Y) = \sum_j \min\left(H_X(j), H_Y(j)\right)$$
$$= 3$$

Histogram intersection counts number of possible matches at a given partitioning.

Spatial pyramid match

- Make a pyramid of bag-of-words histograms.
- Provides some loose (global) spatial layout information



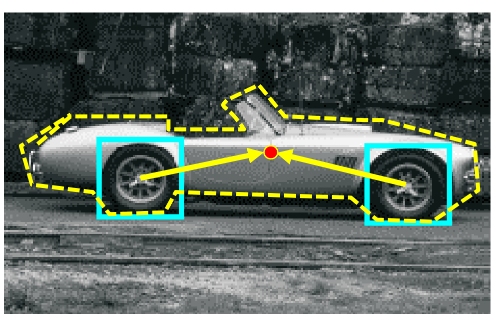
[Lazebnik, Schmid & Ponce, CVPR 2006]

Last time

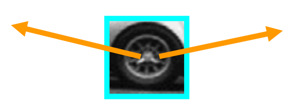
- Modeling categories with local features and spatial information:
  - Histograms, configurations of visual words to capture global or local layout in the bag-of-words framework
    - Pyramid match, semi-local features
  - Part-based models to encode category's part appearance together with 2d layout,
  - Allow detection within cluttered image
    - "implicit shape model", Generalized Hough for detection
    - "constellation model": exhaustive search for best fit of features to parts

Implicit shape models

- Visual vocabulary is used to index votes for object position [a visual word = "part"]



training image annotated with object localization info



visual codeword with displacement vectors

B. Leibe, A. Leonardis, and B. Schiele, Combined Object Categorization and Segmentation with an Implicit Shape Model, ECCV Workshop on Statistical Learning in Computer Vision 2004

Implicit shape models

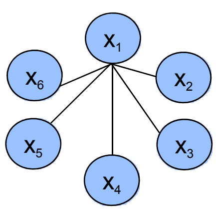
- Visual vocabulary is used to index votes for object position [a visual word = "part"]

test image

B. Leibe, A. Leonardis, and B. Schiele, Combined Object Categorization and Segmentation with an Implicit Shape Model, ECCV Workshop on Statistical Learning in Computer Vision 2004

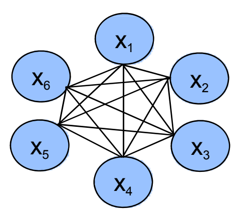
Shape representation in part-based models

"Star" shape model



- **e.g. implicit shape model**
- **Parts mutually independent**
- **Recognition complexity: O(NP)**
- **Method: Gen. Hough Transform**

Fully connected constellation model



- **e.g. Constellation Model**
- **Parts fully connected**
- **Recognition complexity: $O(N^P)$**
- **Method: Exhaustive search**

N image features, P parts in the model

Slide credit: Rob Fergus

Visual Object Recognition Tutorial

Coarse genres of recognition approaches

- Alignment: hypothesize and test
  - Pose clustering with object instances
  - Indexing invariant features + verification
- Local features: as parts or words
  - Part-based models
  - Bags of words models
- Global appearance: “texture templates”
  - With or without a sliding window

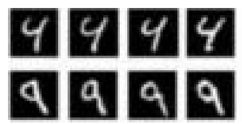
Today

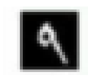
- Detection as classification
  - Supervised classification
    - Skin color detection example
  - Sliding window detection
    - Face detection example

Supervised classification

- Given a collection of *labeled* examples, come up with a function that will predict the labels of new examples.

“four”
“nine”





?

Training examples

Novel input

- How good is some function we come up with to do the classification?
- Depends on
  - Mistakes made
  - Cost associated with the mistakes

Supervised classification

- Given a collection of *labeled* examples, come up with a function that will predict the labels of new examples.

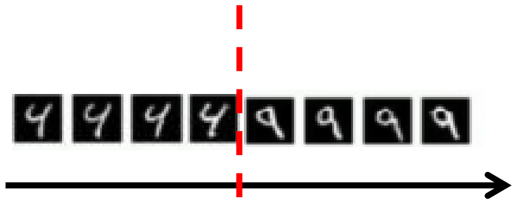
- Consider the two-class (binary) decision problem
  - L(4→9): Loss of classifying a 4 as a 9
  - L(9→4): Loss of classifying a 9 as a 4

- **Risk** of a classifier *s* is expected loss:

$$R(s) = \Pr(4 \to 9 \mid \text{using } s)L(4 \to 9) + \Pr(9 \to 4 \mid \text{using } s)L(9 \to 4)$$

- We want to choose a classifier so as to minimize this total risk

Supervised classification



Feature value **x**

Optimal classifier will minimize total risk.

At decision boundary, either choice of label yields same expected loss.

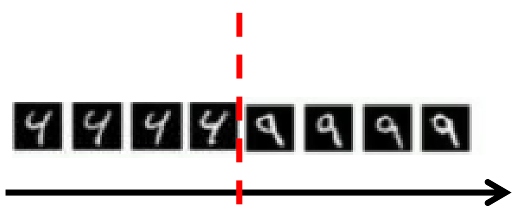
If we choose class "four" at boundary, expected loss is:

$$= P(\text{class is } 9 \mid \mathbf{x})\, L(9 \to 4) + P(\text{class is } 4 \mid \mathbf{x}) L(4 \to 4)$$

If we choose class "nine" at boundary, expected loss is:

$$= P(\text{class is } 4 \mid \mathbf{x})\, L(4 \to 9)$$

Supervised classification



Feature value **x**

Optimal classifier will minimize total risk.

At decision boundary, either choice of label yields same expected loss.

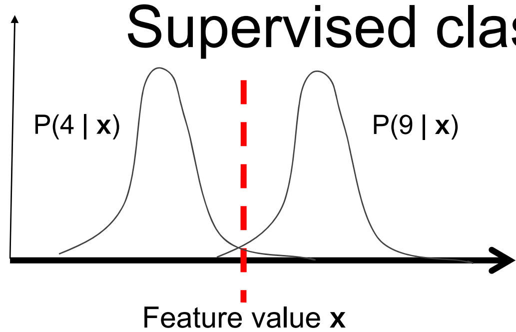
So, best decision boundary is at point **x** where

$$P(\text{class is } 9 \mid \mathbf{x})\, L(9 \to 4) = \mathrm{P}(\text{class is } 4 \mid \mathbf{x}) L(4 \to 9)$$

To classify a new point, choose class with lowest expected loss; i.e., choose "four" if

$$P(4 \mid \mathbf{x}) L(4 \to 9) > P(9 \mid \mathbf{x}) L(9 \to 4)$$

Supervised classification



Optimal classifier will minimize total risk.

At decision boundary, either choice of label yields same expected loss.

So, best decision boundary is at point **x** where

$$P(\text{class is } 9 \mid \mathbf{x})\, L(9 \to 4) = \mathrm{P}(\text{class is } 4 \mid \mathbf{x}) L(4 \to 9)$$

To classify a new point, choose class with lowest expected loss; i.e., choose "four" if

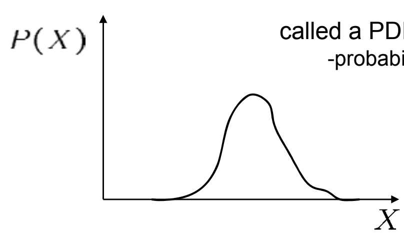
$$\boxed{P(4 \mid \mathbf{x})} L(4 \to 9) > \boxed{P(9 \mid \mathbf{x})} L(9 \to 4)$$

How to evaluate these probabilities?

Probability

Basic probability

- X is a random variable
- P(X) is the probability that X achieves a certain value



called a PDF
-probability distribution/density function

- $0 \leq P(X) \leq 1$
- $\int_{-\infty}^{\infty} P(X)dX = 1$ or $\sum P(X) = 1$

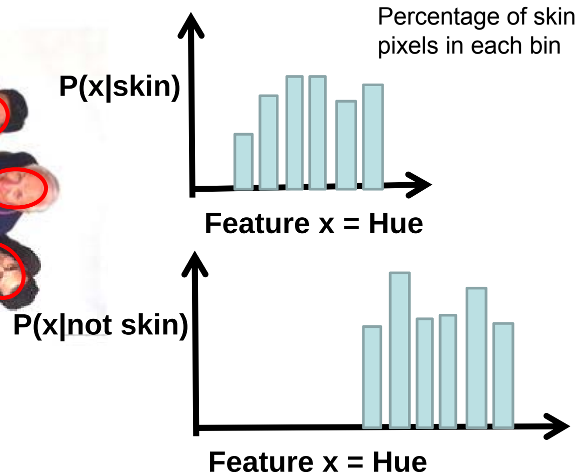
continuous X discrete X

- Conditional probability: P(X | Y)
  - probability of X given that we already know Y

Source: Steve Seitz

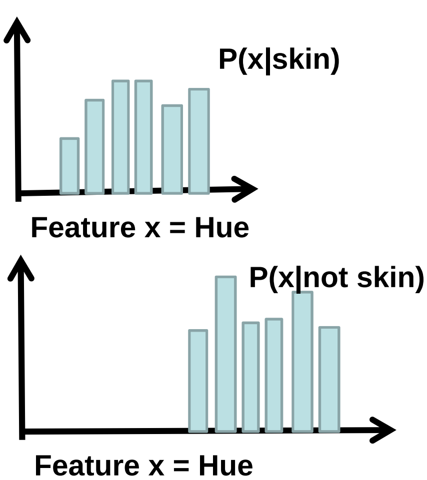
Example: learning skin colors

- We can represent a class-conditional density using a histogram (a “non-parametric” distribution)



Example: learning skin colors

- We can represent a class-conditional density using a histogram (a “non-parametric” distribution)



Now we get a new image, and want to label each pixel as skin or non-skin.

What’s the probability we care about to do skin detection?

Bayes rule

$$\overbrace{P(skin \mid x)}^{\text{posterior}} = \frac{\overbrace{P(x \mid skin)}^{\text{likelihood}}\overbrace{P(skin)}^{\text{prior}}}{P(x)}$$

$$P(skin \mid x)\ \alpha\ P(x \mid skin)P(skin)$$

Where does the prior come from?

Why use a prior?

Example: classifying skin pixels

Now for every pixel in a new image, we can estimate probability that it is generated by skin.

Brighter pixels → higher probability of being skin

Classify pixels based on these probabilities

- if $p(\text{skin}|\boldsymbol{x}) > \theta$, classify as skin
- if $p(\text{skin}|\boldsymbol{x}) < \theta$, classify as not skin

Example: classifying skin pixels

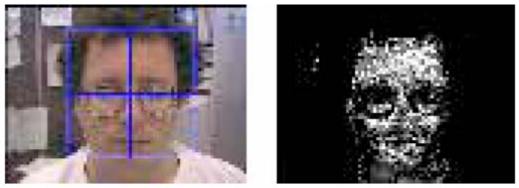


Figure 6: A video image and its flesh probability image

Figure 7: Orientation of the flesh probability distribution marked on the source video image

Gary Bradski, 1998

Example: classifying skin pixels

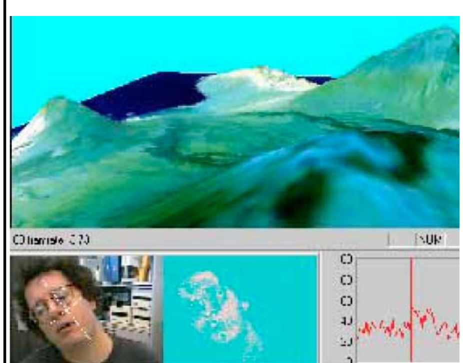


Figure 13: CAMSHIFT-based face tracker used to over a 3D graphic's model of Hawaii

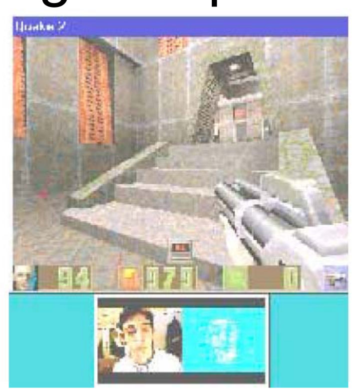


Figure 12: CAMSHIFT-based face tracker used to play Quake 2 hands free by inserting control variables into the mouse queue

Using skin color-based face detection and pose estimation as a video-based interface

Gary Bradski, 1998

Supervised classification

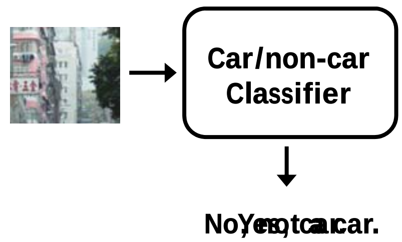
- Want to minimize the expected misclassification
- Two general strategies
  - Use the training data to build representative probability model; separately model class-conditional densities and priors (*generative*)
  - Directly construct a good decision boundary, model the posterior (*discriminative*)

Today

- Detection as classification
  - Supervised classification
    - Skin color detection example
  - Sliding window detection
    - Face detection example

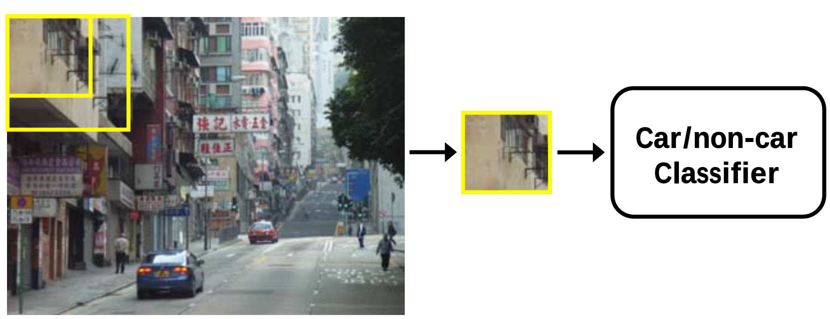
Detection via classification: Main idea

Basic component: a binary classifier



Detection via classification: Main idea

If object may be in a cluttered scene, slide a window around looking for it.



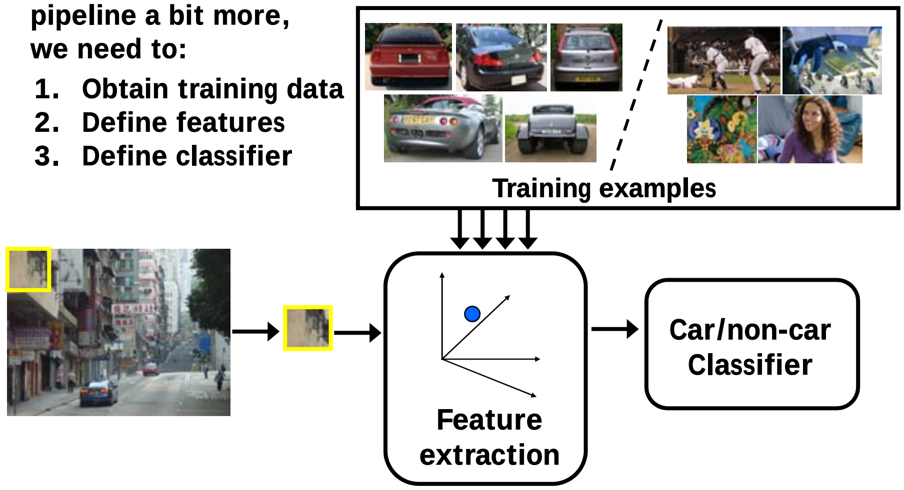
(Essentially, our skin detector was doing this, with a window that was one pixel big.)

Visual Object Recognition Tutorial

Detection via classification: Main idea

Fleshing out this pipeline a bit more, we need to:

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

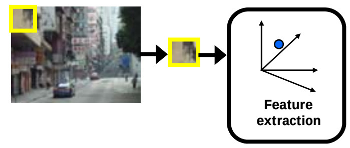


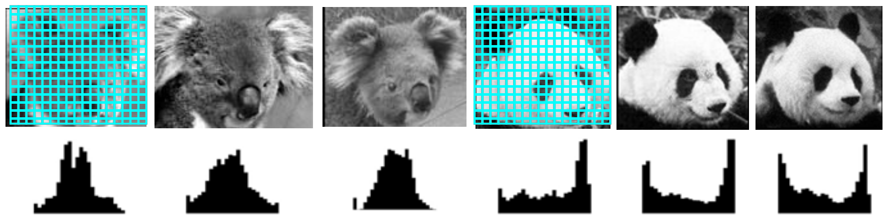
Visual Object Recognition Tutorial

Detection via classification: Main idea

- Consider all subwindows in an image
  - Sample at multiple scales and positions (and orientations)
- Make a decision per window:
  - "Does this contain object category X or not?"

Feature extraction: global appearance





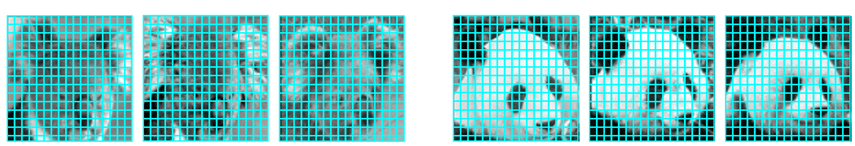
Simple holistic descriptions of image content

- grayscale / color histogram
- vector of pixel intensities

Visual Object Recognition Tutorial

Feature extraction: global appearance

- Pixel-based representations sensitive to small shifts



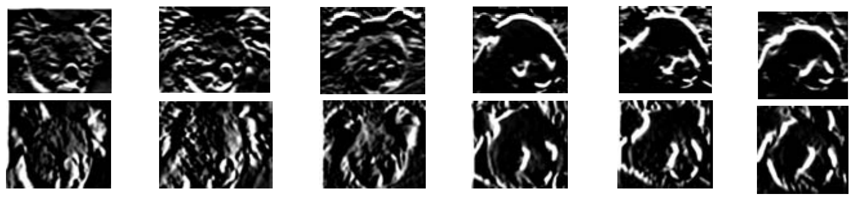
- Color or grayscale-based appearance description can be sensitive to illumination and intra-class appearance variation

Visual Object Recognition Tutorial

Visual Object Recognition Tutorial

Gradient-based representations

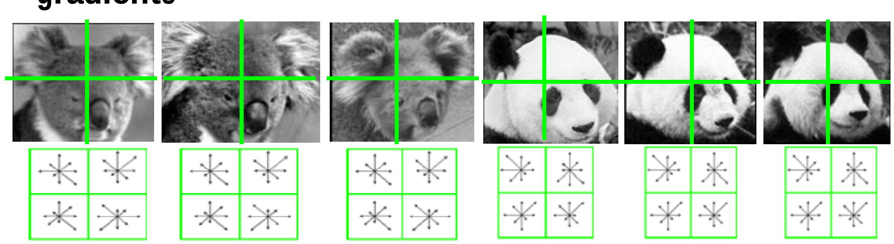
- Consider edges, contours, and (oriented) intensity gradients



Visual Object Recognition Tutorial

Gradient-based representations

- Consider edges, contours, and (oriented) intensity gradients



- Summarize local distribution of gradients with histogram
  - Locally orderless: offers invariance to small shifts and rotations
  - Contrast-normalization: try to correct for variable illumination

Classifier construction

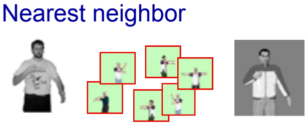
- **How to compute a decision for each subwindow?**

Image feature

Visual Object Recognition Tutorial

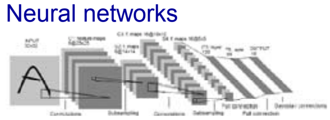
Discriminative classifier construction: many choices...

Nearest neighbor



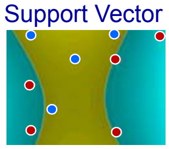
Shakhnarovich, Viola, Darrell 2003
Berg, Berg, Malik 2005...

Neural networks



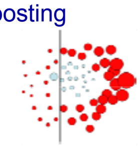
LeCun, Bottou, Bengio, Haffner 1998
Rowley, Baluja, Kanade 1998
…

Support Vector Machines



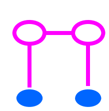
Guyon, Vapnik
Heisele, Serre, Poggio, 2001,…

Boosting



Viola, Jones 2001, Torralba et al. 2004, Opelt et al. 2006,…

Conditional Random Fields



McCallum, Freitag, Pereira 2000; Kumar, Hebert 2003
…

K. Grauman, B. Leibe

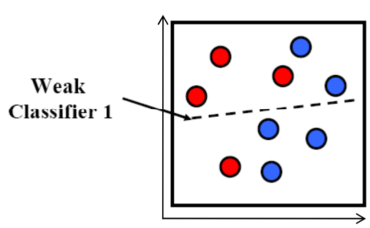
Slide adapted from Antonio Torralba

Boosting

- Build a strong classifier by combining number of "weak classifiers", which need only be better than chance
- Sequential learning process: at each iteration, add a weak classifier
- Flexible to choice of weak learner
  - including fast simple classifiers that alone may be inaccurate
- We'll look at the *AdaBoost* algorithm
  - Easy to implement
  - Base learning algorithm for Viola-Jones face detector

Visual Object Recognition Tutorial

AdaBoost: Intuition



Consider a 2-d feature space with positive and negative examples.

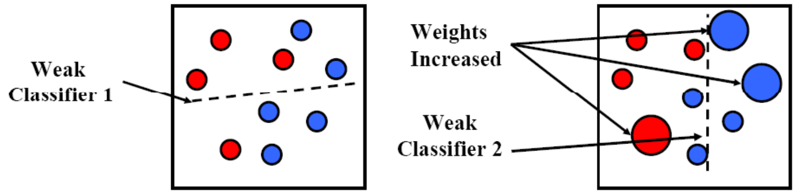
Each weak classifier splits the training examples with at least 50% accuracy.

Examples misclassified by a previous weak learner are given more emphasis at future rounds.

Figure adapted from Freund and Schapire

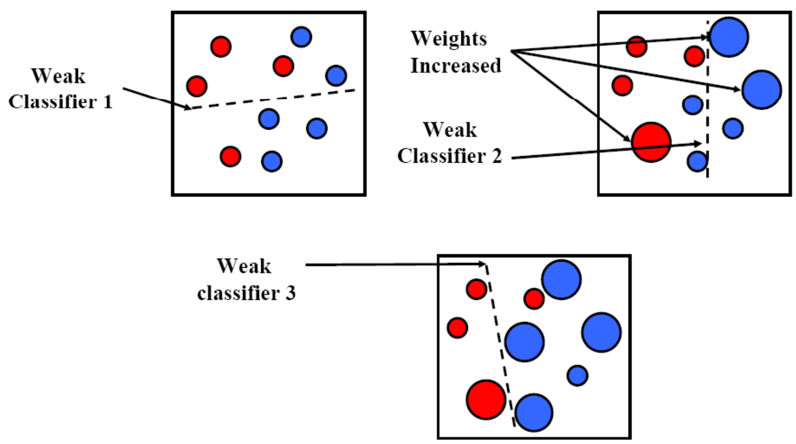
Visual Object Recognition Tutorial

AdaBoost: Intuition



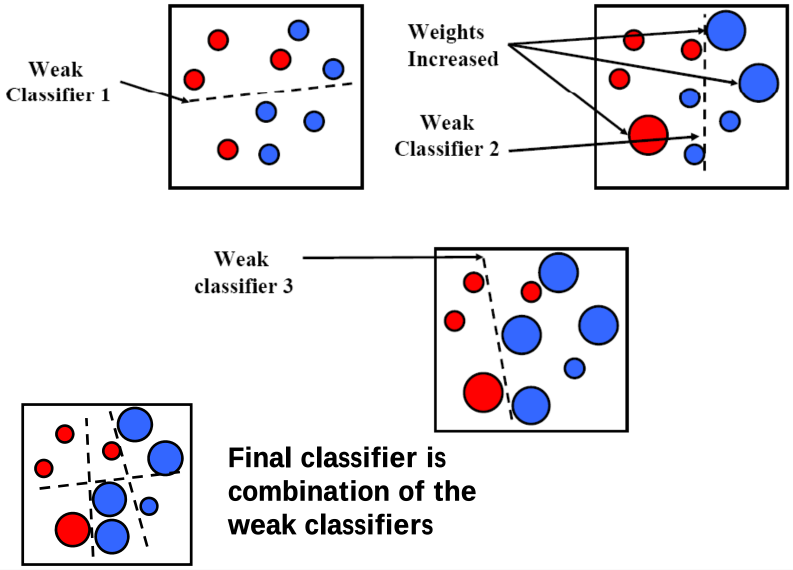
Visual Object Recognition Tutorial

AdaBoost: Intuition



Visual Object Recognition Tutorial

AdaBoost: Intuition



Final classifier is combination of the weak classifiers

Visual Object Recognition Tutorial

Boosting: Training procedure

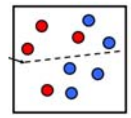
- Initially, weight each training example equally
- In each boosting round:
  - Find the weak learner that achieves the lowest *weighted* training error
  - Raise the weights of training examples misclassified by current weak learner
- Compute final classifier as linear combination of all weak learners (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)

Slide credit: Lana Lazebnik

AdaBoost Algorithm

- Given example images $(x_1, y_1), \ldots, (x_n, y_n)$ 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 $t = 1, \ldots, T$:
  1. 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.
  2. 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, $\epsilon_j = \sum_i w_i |h_j(x_i) - y_i|$.
  3. Choose the classifier, h_t, with the lowest error ϵ_t.
  4. 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



$\{x_1, \ldots x_n\}$

For T rounds

Evaluate *weighted* error for each feature, pick best.

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

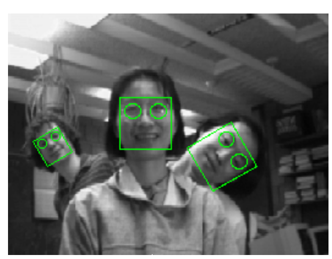
Final classifier is combination of the weak ones, weighted according to error they had.

Freund & Schapire 1995

Visual Object Recognition Tutorial

Faces : terminology

- *Detection*: **given an image, where is the face?**
- *Recognition*: **whose face is it?**



Visual Object Recognition Tutorial

Example: Face detection

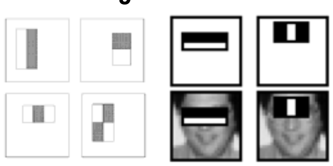
- **Frontal faces are a good example of a class where global appearance models + a sliding window detection approach fit well:**
  - **Regular 2D structure**
  - **Center of face almost shaped like a "patch"/window**

- **Now we'll take AdaBoost and see how the Viola-Jones face detector works**

Visual Object Recognition Tutorial

Feature extraction

"Rectangular" filters

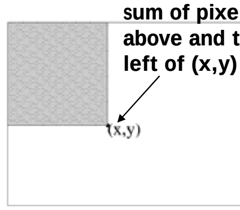


Feature output is difference between adjacent regions

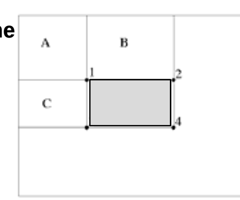
Efficiently computable with integral image: any sum can be computed in constant time

Avoid scaling images → scale features directly for same cost

Value at (x,y) is sum of pixels above and to the left of (x,y)



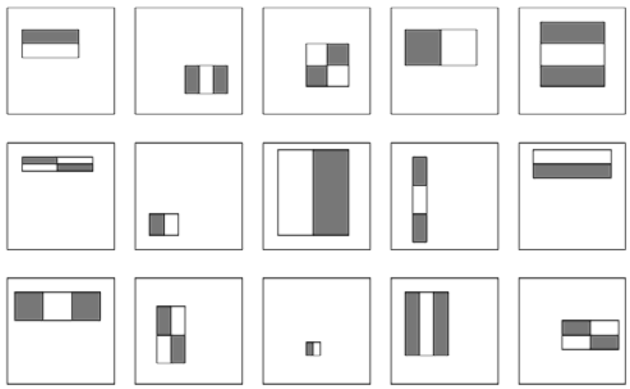
Integral image



$$D = 1 + 4 - (2 + 3)$$
$$= A + (A + B + C + D) - (A + C + A + B)$$
$$= D$$

Viola & Jones, CVPR 2001

Large library of filters



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

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

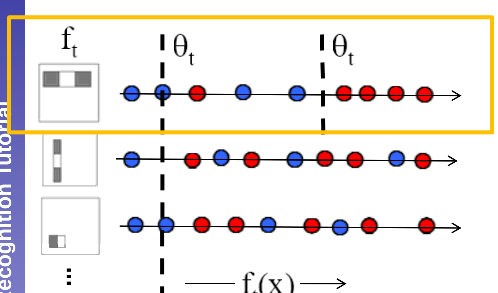
Which subset of these features should we use to determine if a window has a face?

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

Visual Object Recognition Tutorial

AdaBoost for feature+classifier selection

- **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.**



Outputs of a possible rectangle feature on faces and non-faces.

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.

- 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

For efficiency, apply less accurate but faster classifiers first to immediately discard windows that clearly appear to be negative; e.g.,

- **Filter for promising regions with an initial inexpensive classifier**
- **Build a chain of classifiers, choosing cheap ones with low false negative rates early in the chain**

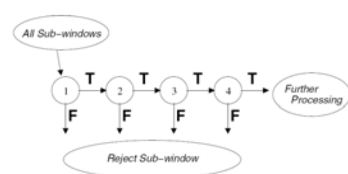
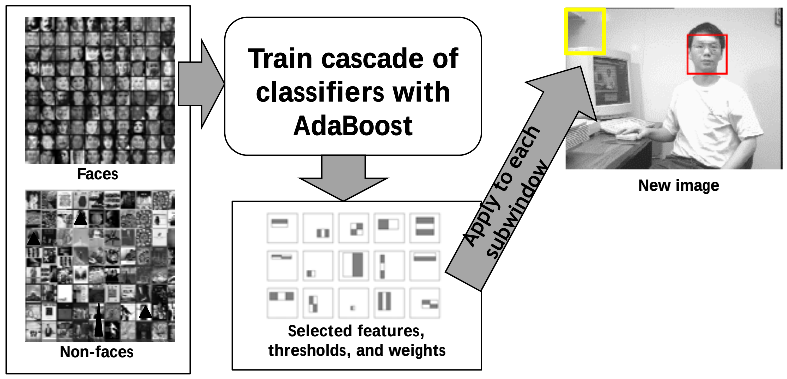


Figure from Viola & Jones CVPR 2001

Visual Object Recognition Tutorial

Viola-Jones Face Detector: Summary



- Train with 5K positives, 350M negatives
- Real-time detector using 38 layer cascade
- 6061 features in final layer
- [Implementation available in OpenCV: http://www.intel.com/technology/computing/opencv/]

Visual Object Recognition Tutorial

Viola-Jones Face Detector: Summary

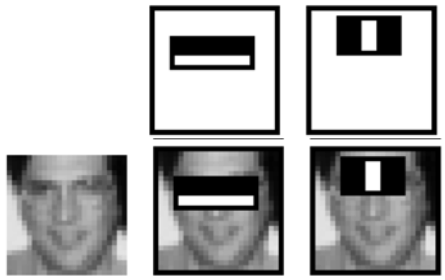
- A seminal approach to real-time object detection
- Training is slow, but detection is very fast
- Key ideas
  - *Integral images* for fast feature evaluation
  - *Boosting* for feature selection
  - *Attentional cascade* for fast rejection of non-face windows

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.

Slide credit: Lana Lazebnik

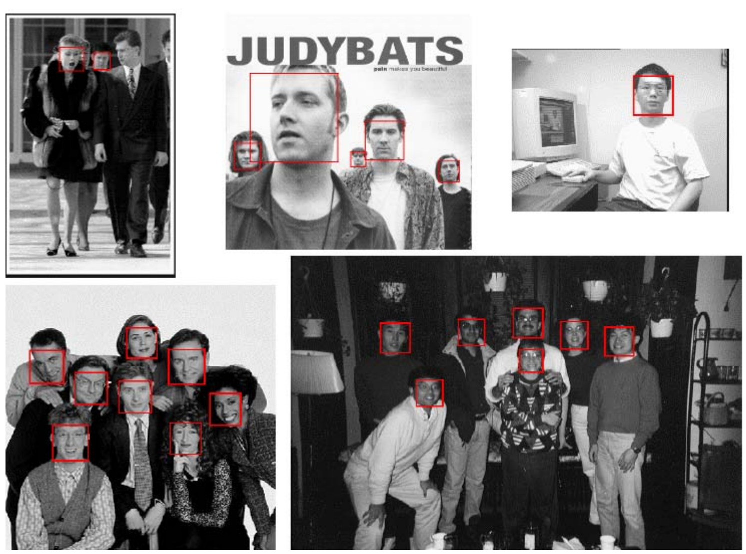
Viola-Jones Face Detector: Results



First two features selected

Visual Object Recognition Tutorial

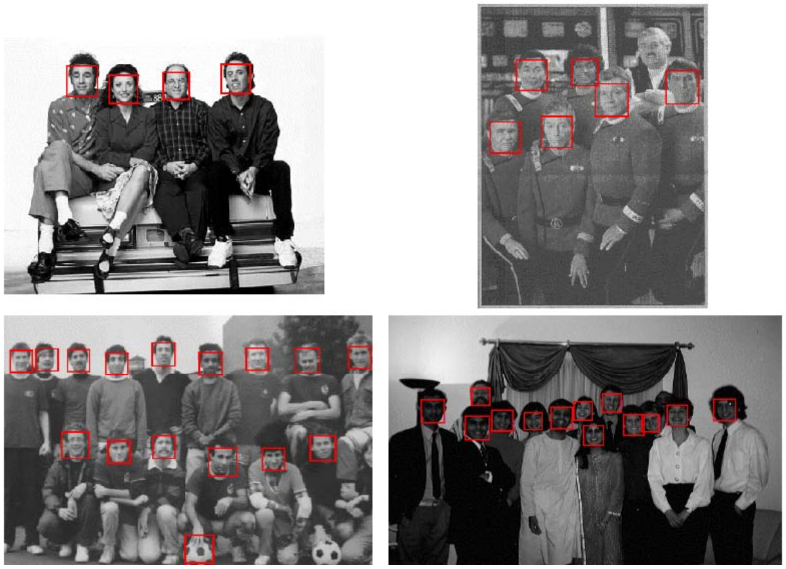
Viola-Jones Face Detector: Results



Visual Object Recognition Tutorial

Visual Object Recognition Tutorial

Viola-Jones Face Detector: Results



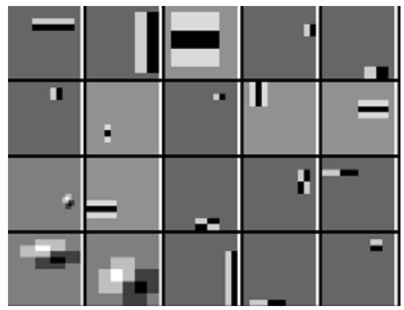
Visual Object Recognition Tutorial

Viola-Jones Face Detector: Results

Visual Object Recognition Tutorial

Detecting profile faces?

Can we use the same detector?



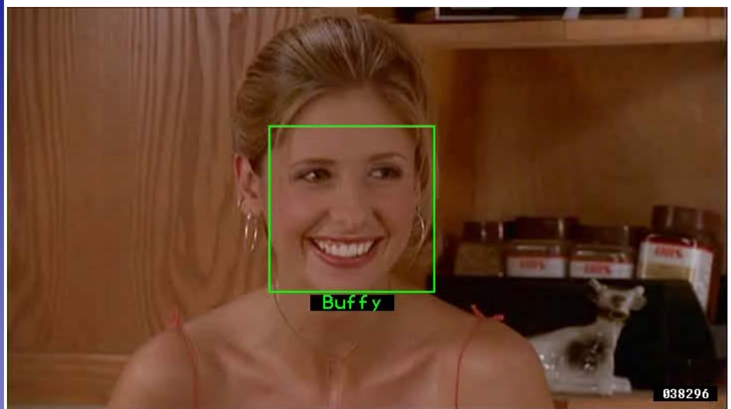
Visual Object Recognition Tutorial

Pau

Viola-Jones Face Detector: Results

Visual Object Recognition Tutorial

Example application

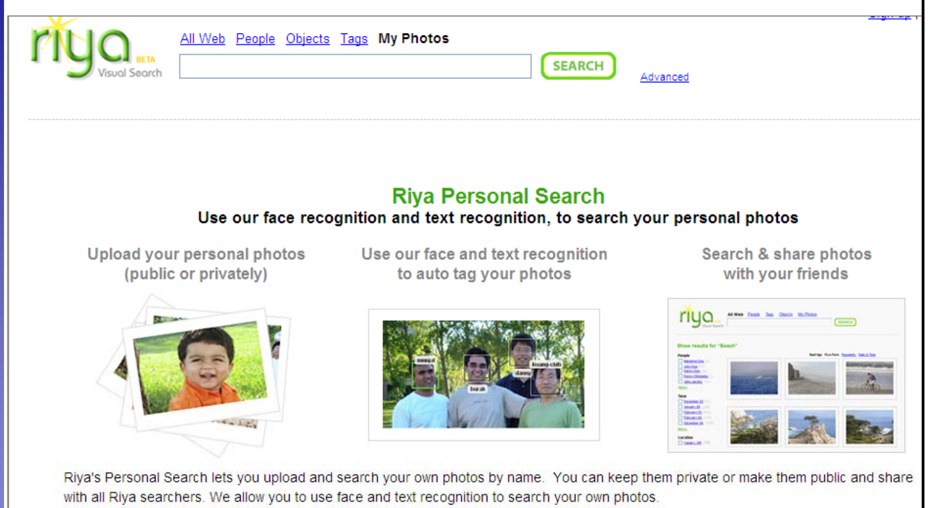


Frontal faces detected and then tracked, character names inferred with alignment of script and subtitles.

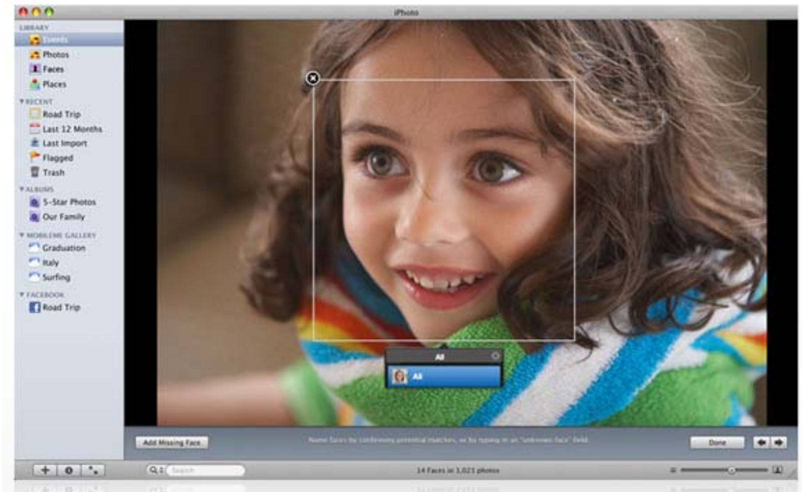
Everingham, M., Sivic, J. and Zisserman, A.
"Hello! My name is... Buffy" - Automatic naming of characters in TV video, BMVC 2006.
http://www.robots.ox.ac.uk/~vgg/research/nface/index.html

Visual Object Recognition Tutorial

Example application: faces in photos



Consumer application: iPhoto 2009

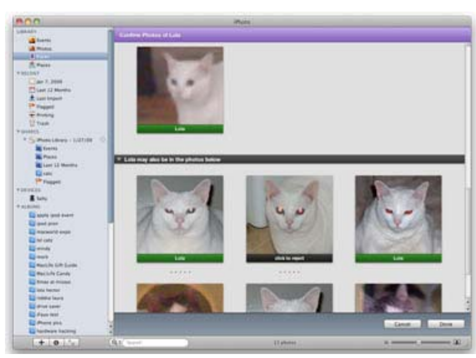


http://www.apple.com/ilife/iphoto/

Slide credit: Lana Lazebnik

Consumer application: iPhoto 2009

Can be trained to recognize pets!



http://www.maclife.com/article/news/iphotos_faces_recognizes_cats

Slide credit: Lana Lazebnik

Consumer application: iPhoto 2009

Things iPhoto thinks are faces

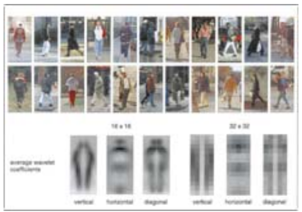
Slide credit: Lana Lazebnik

- Other classes that might work with global appearance in a window?

Pedestrian detection

Visual Object Recognition Tutorial

- Detecting upright, walking humans also possible using sliding window's appearance/texture; e.g.,

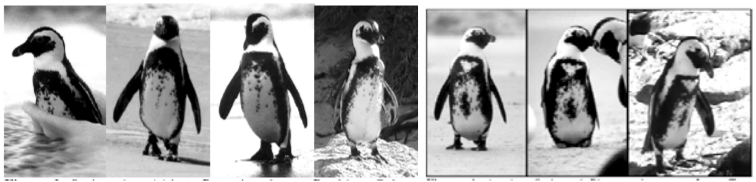


SVM with Haar wavelets [Papageorgiou & Poggio, IJCV 2000]

Space-time rectangle features [Viola, Jones & Snow, ICCV 2003]

SVM with HoGs [Dalal & Triggs, CVPR 2005]

- Other classes that might work with global appearance in a window?



Fifth International Penguin Conference, Ushuaia, Tierra del Fuego, Argentina, September 2004

Fifth International Penguin Conference
Ushuaia, Tierra del Fuego, Argentina

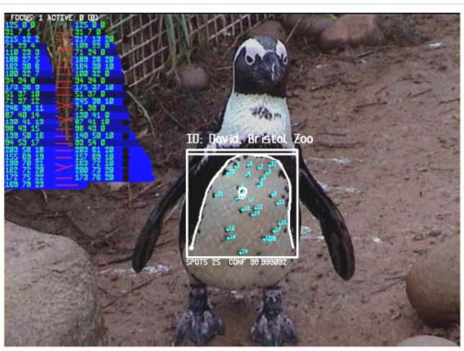
Automated Visual Recognition of Individual African Penguins

Tilo Burghardt,
Barry Thomas, Peter J Barham, Janko Ćalić

University of Bristol, Department of Computer Science,
MVB Woodland Road, Bristol BS8 1UB, United Kingdom,
September 2004

burghard@cs.bris.ac.uk

Penguin detection & identification



African penguins *(Spheniscus demersus)* carry a pattern of black spots on their chests that does not change from season to season during their adult life. Further, as far as we can tell, no two penguins have exactly the same pattern. We have developed a real-time system that can confidently locate African penguins whose chests are visible within video sequences or still images. An extraction of the chest spot pattern allows the generation of a unique biometrical identifier for each penguin. Using these identifiers an authentication of filmed or photographed African penguins against a population database can be performed. This paper provides a detailed technical description of the developed system and outlines the scope and the conditions of application ■

This project uses the Viola-Jones Adaboost face detection algorithm to detect penguin chests, and then matches the pattern of spots to identify a particular penguin.

Burghart, Thomas, Barham, and Calic. Automated Visual Recognition of Individual African Penguins , 2004.

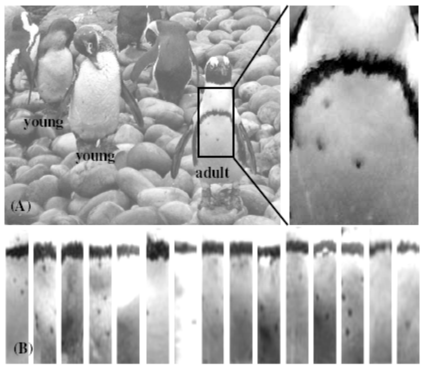
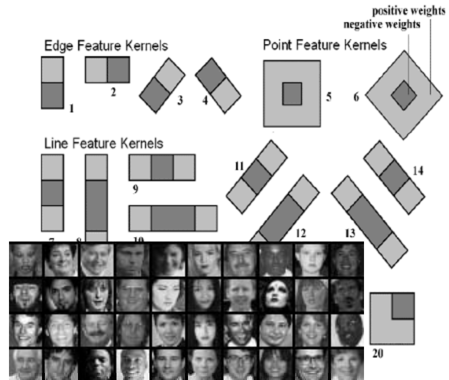


Figure 6. *Distinctive Chest Stripe of Adult African Penguins:* **(A)** Adult African penguins carry a distinctive and stable black stripe on their chest whilst young members of the species still change the colour of their chest feathers. **(B)** Various chests of adult African penguins under different lighting conditions. (figure source [19])



Use rectangular features, select good features to distinguish the chest from non-chests with Adaboost

Burghart, Thomas, Barham, and Calic. Automated Visual Recognition of Individual African Penguins , 2004.

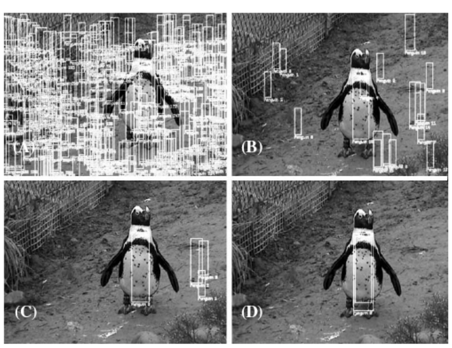


Figure 12. *Application of Attentional Cascades on Chests:* **(A)** Image areas that are accepted as likely to represent a chest after one stage are marked as white rectangles. **(B)** After three stages... **(C)** After five stages... **(D)** ...and after seven stages with final result. (figure source [18], [19])

Attentional cascade

Figure 10. *AoI Detector Spotting Frontal Penguin Chests:* The detector was tested on a series of black and white still images and footage. Some result images are shown above. The detector might fire several times on one and the same chest instance. (figure

Penguin chest detections

Burghart, Thomas, Barham, and Calic. Automated Visual Recognition of Individual African Penguins , 2004.

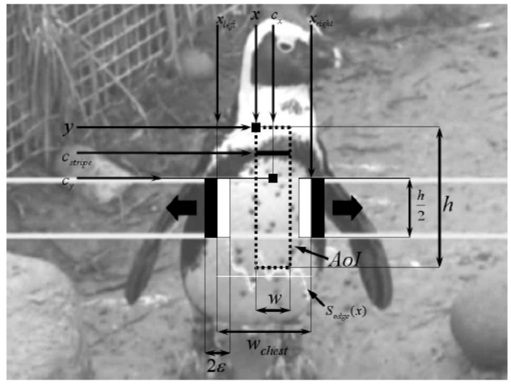
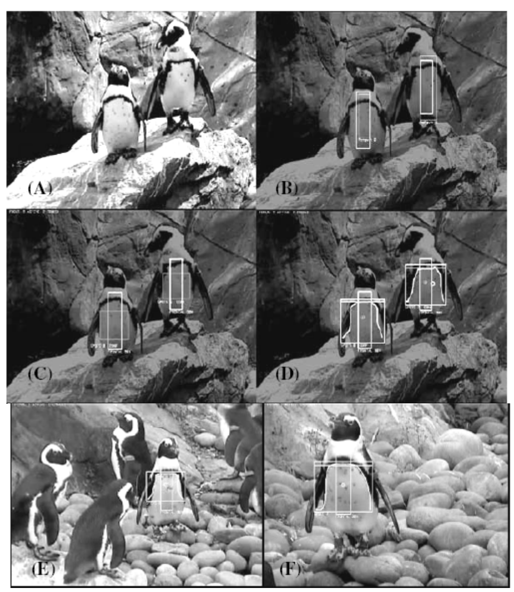


Figure 14. *Visual Description of the Chest Width Measurement:* Starting from an upper central point of the chest AoI two locally operating edge detectors moving apart search for the left and right boundary of the assumed chest. (figure source [17])

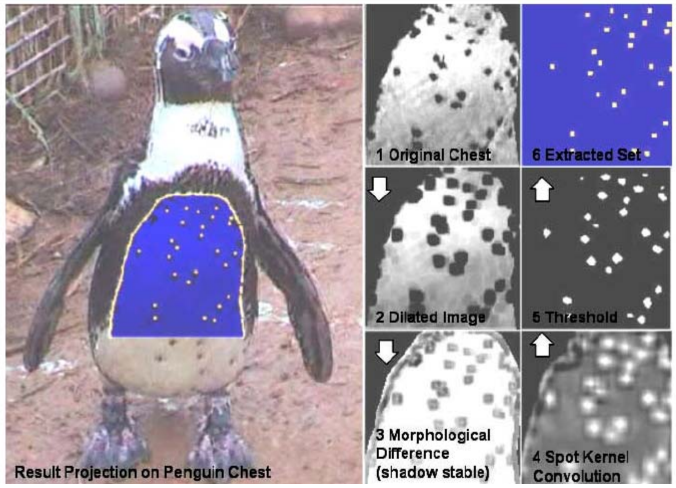
Given a detected chest, try to extract the whole chest for this particular penguin.

Burghart, Thomas, Barham, and Calic. Automated Visual Recognition of Individual African Penguins , 2004.



Example detections

Burghart, Thomas, Barham, and Calic. Automated Visual Recognition of Individual African Penguins , 2004.



Perform **identification** by matching the pattern of spots to a database of known penguins.

Burghart, Thomas, Barham, and Calic. Automated Visual Recognition of Individual African Penguins , 2004.

Penguin detection & identification

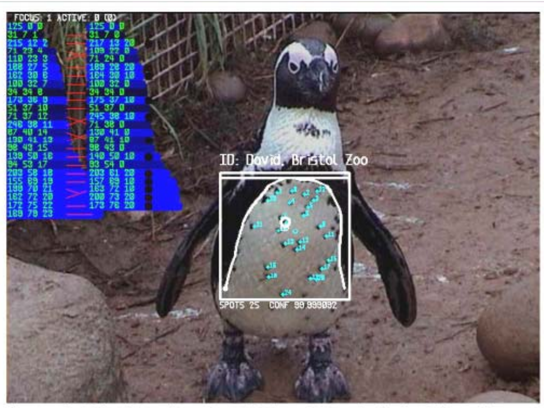


Figure 1. *Identification of an African Penguin by its Chest Pattern:* Screenshot of Software Prototype; African penguins carry a unique pattern of black spots on their chest. The detection of the chest location and the decomposition of the spot pattern allow checking a photographed individual (here penguin 'David' from Bristol Zoo) against a population database. (figure source [18], [19])

Burghart, Thomas, Barham, and Calic. Automated Visual Recognition of Individual African Penguins , 2004.

Highlights

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

Visual Object Recognition Tutorial

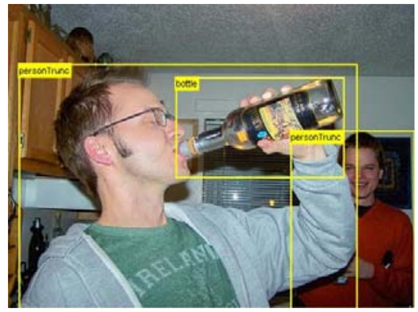
Limitations

- 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

Visual Object Recognition Tutorial

Limitations (continued)

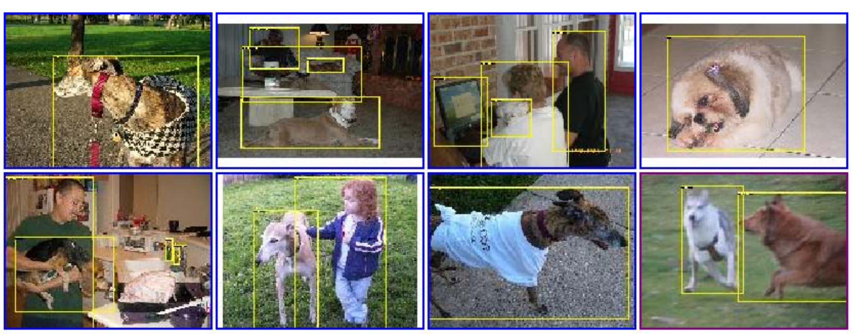
- Not all objects are “box” shaped



Visual Object Recognition Tutorial

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



Visual Object Recognition Tutorial

Limitations (continued)

- If considering windows in isolation, context is lost

Sliding window

Detector's view

Figure credit: Derek Hoiem

Visual Object Recognition Tutorial

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

Image credit: Adam, Rivlin, & Shimshoni

Summary:
Detection as classification

- Supervised classification
  - Loss and risk, Bayes rule
  - Skin color detection example
- Sliding window detection
  - Classifiers, boosting algorithm, cascades
  - Face detection example
- Limitations of a global appearance description
- Limitations of sliding window detectors