Face Detection

The “Margaret Thatcher Illusion”, by Peter Thompson
Funny Nikon ads

"The Nikon S60 detects up to 12 faces."
Funny Nikon ads

"The Nikon S60 detects up to 12 faces."
Consumer application: Apple iPhoto

Things iPhoto thinks are faces
Applications: Computational photography

[Face priority AE] When a bright part of the face is too bright
Consumer application: Apple iPhoto

http://www.apple.com/ilife/iphoto/
Consumer application: Apple iPhoto

Can be trained to detect pets!

Exploring Photobios

Ira Kemelmacher-Shlizerman
University of Washington

Eli Shechtman
Adobe Systems

SIGGRAPH 2011

Rahul Garg
Steven M. Seitz
University of Washington
Google Inc.
Tyson and Jasper
Challenges of face detection

• Sliding window detector must evaluate tens of thousands of location/scale combinations

• Faces are rare: 0–10 per image
  • Should make quick decisions at non-face regions
  • A megapixel image has $\sim 10^6$ pixels and a comparable number of candidate face locations
  • To avoid having a false positive in every image image, our false positive rate has to be less than $10^{-6}$
One simple method: skin detection

Skin pixels have a distinctive range of colors
- Corresponds to region(s) in RGB color space
  - for visualization, only R and G components are shown above

Skin classifier
- A pixel $X = (R, G, B)$ is skin if it is in the skin region
- But how to find this region?
Skin detection

Learn the skin region from examples
- Manually label pixels in one or more “training images” as skin or not skin
- Plot the training data in RGB space
  - skin pixels shown in orange, non-skin pixels shown in blue
  - some skin pixels may be outside the region, non-skin pixels inside. Why?

Skin classifier
- Given $X = (R,G,B)$: how to determine if it is skin or not?
Skin classification techniques

Skin classifier

- Given $X = (R,G,B)$: how to determine if it is skin or not?
- Nearest neighbor
  - find labeled pixel closest to $X$
  - choose the label for that pixel
- Data modeling
  - fit a model (curve, surface, or volume) to each class
- Probabilistic data modeling
  - fit a probability model to each class
Skin detection results

Figure 25.3. The figure shows a variety of images together with the output of the skin detector of Jones and Rehg applied to the image. Pixels marked black are skin pixels, and white are background. Notice that this process is relatively effective, and could certainly be used to focus attention on, say, faces and hands. Figure from “Statistical color models with application to skin detection,” M.J. Jones and J. Rehg, Proc. Computer Vision and Pattern Recognition, 1999 © 1999, IEEE.
The Viola/Jones Face Detector

• 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


Image Features

“Rectangle filters”

Value =

\[ \sum \text{(pixels in white area)} - \sum \text{(pixels in black area)} \]
Example

Source

Result
Example
Fast computation with integral images

- The integral image pixel \((x,y)\) stores the sum of the values above and left.
- This can quickly be computed in one pass through the image.
Integral images

What’s the sum of pixels in the blue rectangle?

input image

integral image
Integral images
Integral images

integral image
Integral images

integral image
Integral images

integral image
Integral images
Integral images

What’s the sum of pixels in the rectangle?
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!
Computing a rectangle feature

Integral Image
Feature selection

• For a 24x24 detection region, the number of possible rectangle features is \(~160,000\)!
Feature selection

• For a 24x24 detection region, the number of possible rectangle features is ~160,000!
• At test time, it is impractical to evaluate the entire feature set
• Can we create a good classifier using just a small subset of all possible features?
• How to select such a subset?
Boosting

- *Boosting* is a classification scheme that combines *weak learners* into a more accurate *ensemble classifier*.
- Weak learners based on rectangle filters:

\[
h_t(x) = \begin{cases} 
1 & \text{if } f_t(x) > \theta_t \\
0 & \text{otherwise}
\end{cases}
\]
Boosting

- *Boosting* is a classification scheme that combines weak learners into a more accurate ensemble classifier.

- Weak learners based on rectangle filters:

\[ h_t(x) = \begin{cases} 
1 & \text{if } f_t(x) > \theta_t \\
0 & \text{otherwise}
\end{cases} \]

- Ensemble classification function:

\[ C(x) = \begin{cases} 
1 & \text{if } \sum_{t=1}^{T} \alpha_t h_t(x) > \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\
0 & \text{otherwise}
\end{cases} \]
Boosting

Slide credit: Paul Viola
Boosting

Weights Increased
Boosting
Boosting

Weights Increased
Boosting

Weak Classifier 3
Boosting

Final classifier is a combination of weak classifiers
Boosting: training

• Initially, weight each training example equally

• In each boosting round:
  – find the best weak classifier
  – raise weights of misclassified training examples

• Final classifier is linear combination of all weak classifiers
  – weight of each learner is directly proportional to its accuracy)

• Exact formulas for re-weighting and combining weak classifiers depend on the particular boosting scheme
Boosting for face detection

• First two features selected by boosting:

This feature combination can yield 100% detection rate and 50% false positive rate
Boosting for face detection

- A 200-feature classifier can yield 95% detection rate and a false positive rate of 1 in 14084
• 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

Stage 1 classifier: Face \rightarrow Non-face
Stage 2 classifier: Face \rightarrow Non-face
Stage 3 classifier: Face \rightarrow Non-face

Rejected sub-windows

More features, lower false positive rates

All sub-windows, multiple scales

Detection at a sub-window
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
Attentional cascade

• Start with simple classifiers which reject many negatives but do not miss positives
• Positive response moves to the next
• Negative response means immediate rejection
Attentional cascade

- Chain classifiers that are progressively more complex and have lower false positive rates:

![Diagram of chain classifiers]

```
IMAGE SUB-WINDOW → Classifier 1 → Classifier 2 → Classifier 3 → FACE
```

Receiver operating characteristic

- % False Pos vs % Detection

- Graph showing the relationship between false positive rate and detection rate for different classifiers.
Training the cascade

• Set *target detection* and *false positive rates* for each stage

• For each stage
  • Keep adding features to achieve the target detection
  • Lower AdaBoost threshold to maximize detection (as opposed to minimizing total classification error)
  • Test on a validation set
  • If false positive rate is not low enough, then add another stage

• Note: Use false positives from current stage as the negative training examples for the next stage
The implemented system

• **Training Data**
  • 5000 faces
    – All frontal, rescaled to 24x24 pixels
  • 300 million non-faces
    – 9500 non-face images
  • Faces are normalized
    – Scale, translation

• **Many variations**
  • Across individuals
  • Illumination
  • Pose
System performance

- Training time: “weeks” on 466 MHz Sun workstation
- 38 layers, total of 6061 features
- Average of 10 features evaluated per window on test set
- “On a 700 MHz Pentium III processor, the face detector can process a 384 by 288 pixel image in about .067 seconds”
  - 15 Hz
  - 15 times faster than previous detector of comparable accuracy (Rowley et al., 1998)
Output of Face Detector on Test Images
Other detection tasks

Facial Feature Localization

Profile Detection

Male vs. female
Profile Detection
Profile Features
Viola–Jones detector: summary

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

[Implementation available in OpenCV: http://www.intel.com/technology/computing/opencv/]

Kristen Grauman
Application: streetview
Application: streetview
Application: streetview
What other categories are amenable to window-based representation?
Pedestrian detection

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

SVM with HoG [Dalal & Triggs, CVPR 2005]
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)

- If considering windows in isolation, context is lost

![Sliding window](image1)
![Detector’s view](image2)

Figure credit: Derek Hoiem
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

Kristen Grauman
Deep Learning