Presenting
“Robust Real-time Object Detection”
Paper by Paul Viola and Michael Jones (2001)

CMPS 242: Graduate Machine Learning

Presented by Damian Eads
Face Detection

• Find bounding rectangles around faces
Face Detection

• **Very fast** but *simple* features

• Large space of them

• Each pixel in the feature is computed in constant time (almost like accessing the image itself!)
Haar-like Features

- Try every combination of 1x2, 2x1, 1x3, 3x1, 2x2 with detection window
Integral Image

- \( \text{IntImage}(x,y) = \text{sum of pixels } x' \leq x \text{ and } y' \leq y \)
Integral Image

- \( \text{Integral Image}(x,y) = \text{sum of pixels } x' \leq x \text{ and } y' \leq y \)

\[
\sum (A+D)-(C+B)
\]

4 pixel accesses!
Why VJ features work well?
Cascade

- Filter out large non-face regions
Cascade

• 80% of pixels involve about $7 \times 4 = 28$ integral image accesses!

• Cascaded rejection makes this very fast! 16 frames/second.
Cascade

• 80% of pixels involve about $7 \times 4 = 28$ integral image accesses!

• Cascaded rejection makes this very fast! 16 frames/second.

For each level of cascade, train AdaBoost on full set!
Summary

• **very fast features**: convolution not needed, constant time features

• **Haar features**: good for describing *face* primitives

• **cascades**:
  
  • eliminate most of the image with just a few features (2-5 features)
Summary

• **very fast features**: convolution not needed, constant time features

• **Haar features**: good for describing *face* primitives

• **cascades**:
  
  • eliminate most of the image with just a few features (2-5 features)

  = robust, real-time face detection
Summary

- **very fast features**: convolution not needed, constant time features

- **Haar features**: good for describing *face* primitives

- **cascades**:
  - eliminate most of the image with just a few features (2-5 features)

  = robust, real-time face detection with boosting