MANCHESTER 1824

The University of Manchester

Short Lecture:
Face Detection:
The Viola-Jones Algorithm

Spring 2021

Slides by: Dr Carole Twining

Presented by: Terence Morley

Handouts & Lecture Notes

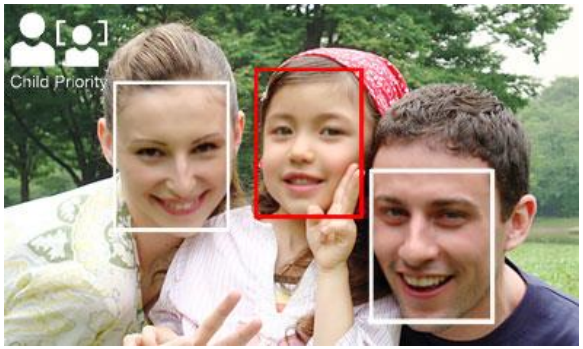
- Report in Scientific American (June 2014):

  "In each study, however, those who wrote out their notes by hand had a stronger conceptual understanding and were more successful in applying and integrating the material than those who used [sic] took notes with their laptops."

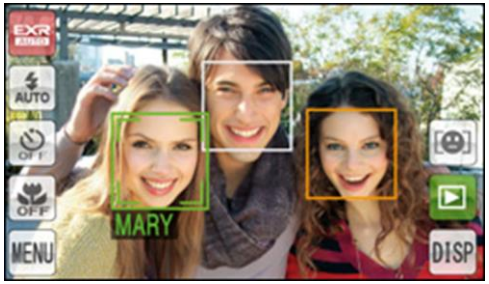
  The Pen Is Mightier Than the Keyboard
  P. A. Mueller, D. M. Oppenheimer, *Psychological Science,* Vol 25,Issue 6, pp. 1159 – 1168, April-23-2014.

- Handouts are to aid note taking, not a total replacement for note taking
- Podcasts, slides, pdfs etc on BlackBoard

Commercial Applications



Sony Cyber-Shot



Images from Fujifilm

- Face **Detection**:
  - Red-eye removal, focus and exposure
  - Tracking
  - Adult/**Child** detection (Sony)
- Face **Recognition**:
  - Save person-specific settings
  - Identity-Based Image Retrieval
- Cat and Dog **Detection**:
  - Focus and exposure
  - Take picture when face-on
  - Pet recognition as well! (Pentax)

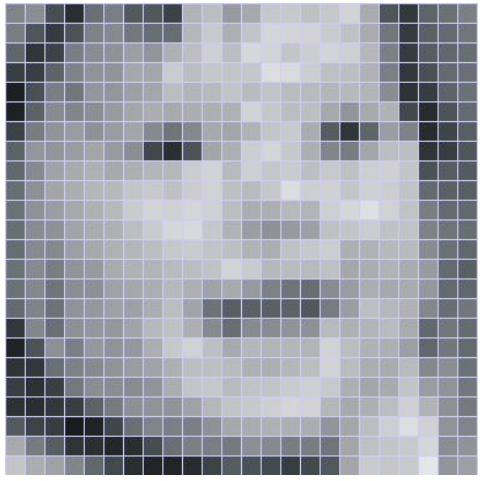
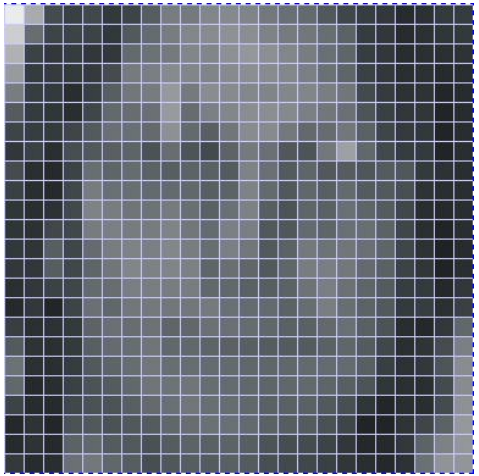
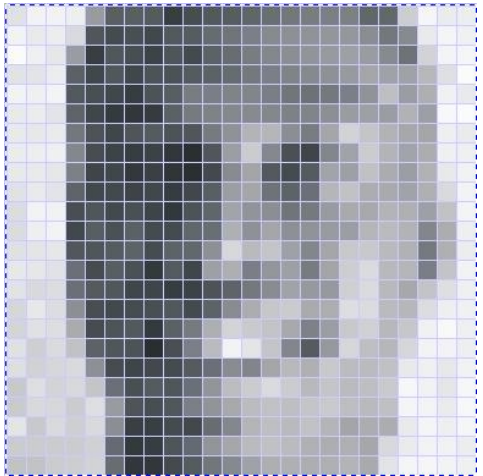
Clues to Detecting (Human) Faces?

- Basic features and their relationship
- Simple pattern in blurred, grayscale, contrast-adjusted patches
- Dark & light patches, simple horizontal & vertical patterns

Task:

- **Scan images, and find regions that contain examples of the object of interest**

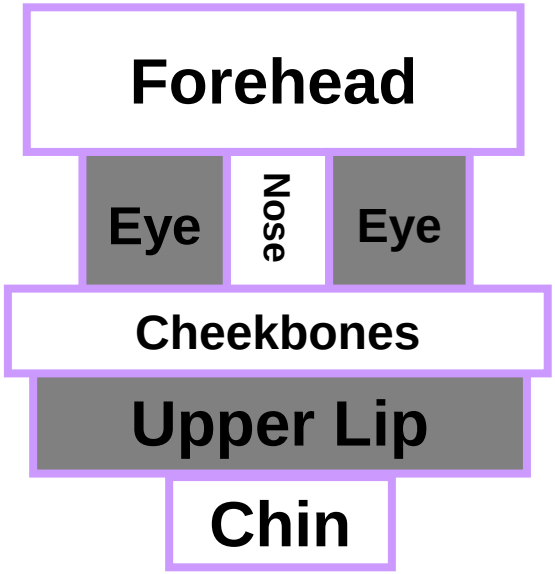

- **Learning Approach: Given positive and negative examples, learn best features and best classifier**

Fast Face Detection

Requirements:

- Speed of computation
  (15 frames per second)
- ~ 90% - 95% detection rate
- ~ 10^{-5} false positive rate

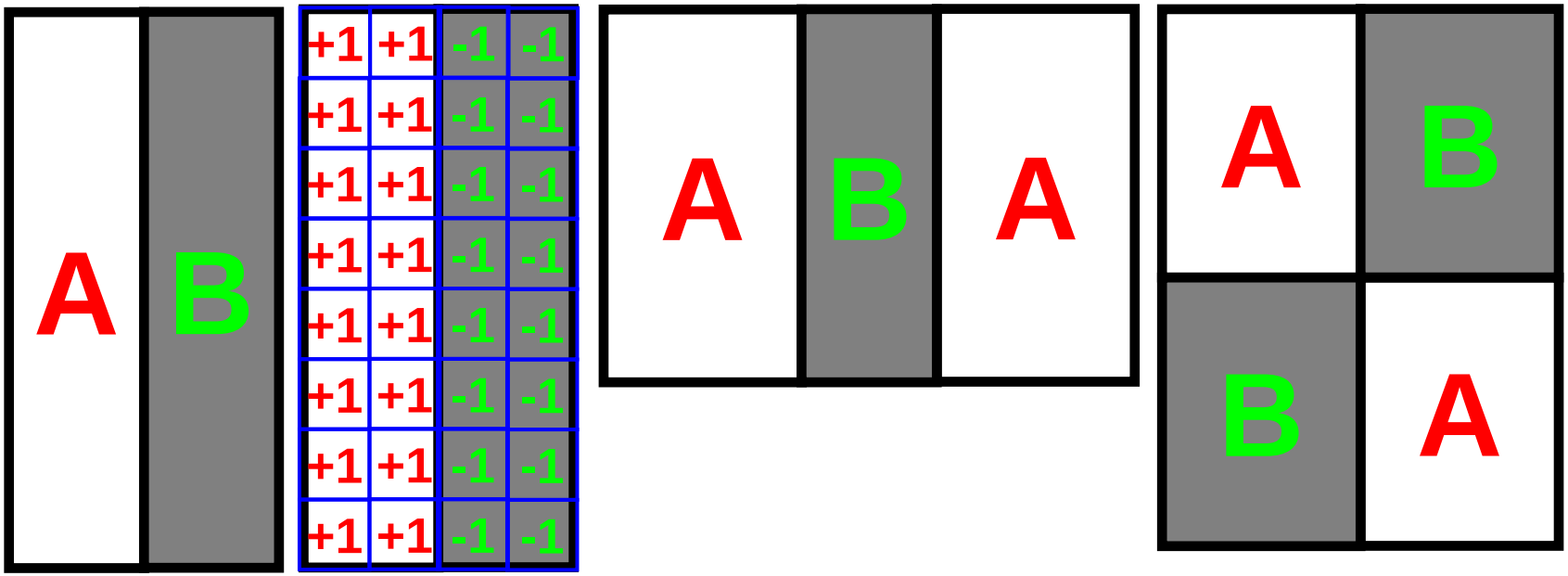


Approach:

- Limit yourself to frontal, upright faces
- Efficient-to-compute features
- Efficient image representation
- Adaboost for efficient choice of features
- Cascade of classifiers

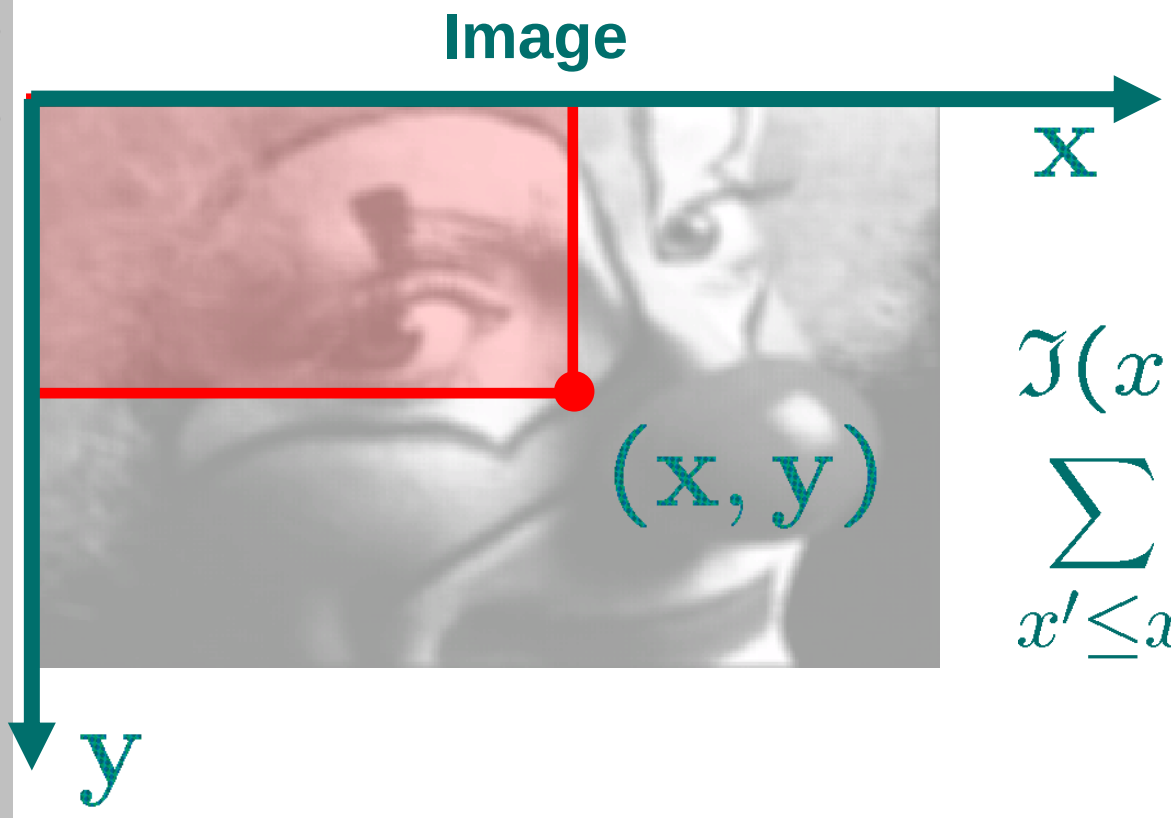
Rectangle Features: Definition

$$\mathcal{F} = \sum_{\vec{r} \in A} \mathcal{I}(\vec{r}) - \sum_{\vec{r} \in B} \mathcal{I}(\vec{r})$$



- Vertical/Horizontal/Diagonal feature detectors
- Implementation as convolution too inefficient

Image Representation: Integral Image

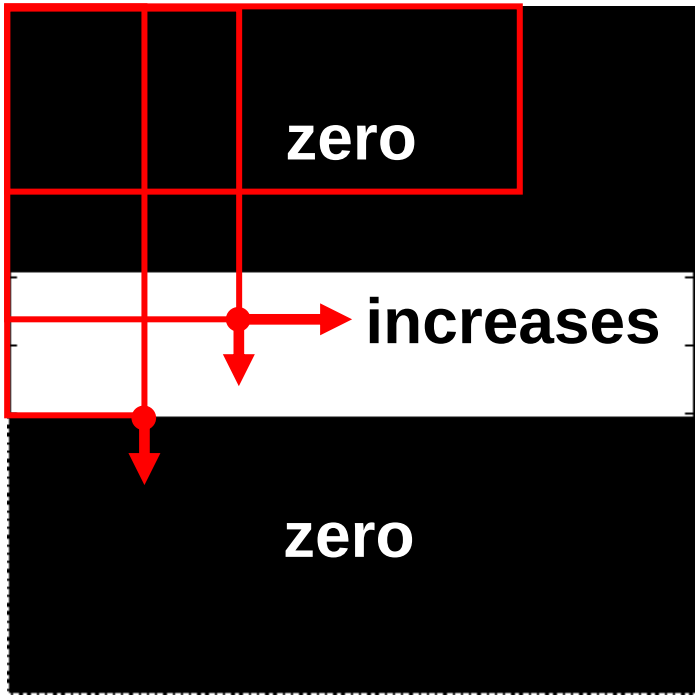


$$\mathfrak{I}(x,y) \doteq \sum_{x' \leq x} \sum_{y' \leq y} \mathcal{I}(x',y')$$

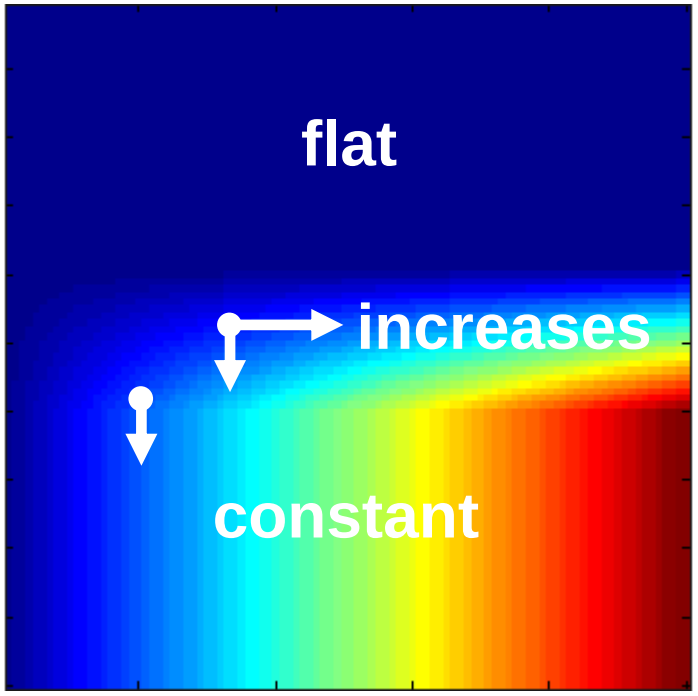
- Integral image can be computed in one pass

Image Representation: Integral Image

$\mathcal{I}(x, y)$

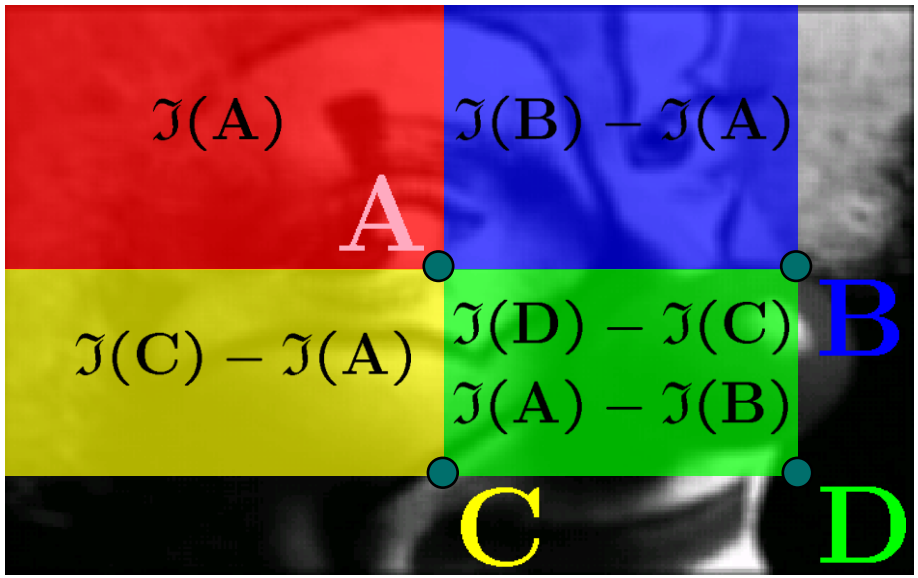


$\mathfrak{I}(x, y)$



- Integral image contains all information in original image

Rectangle Features: Integral Image



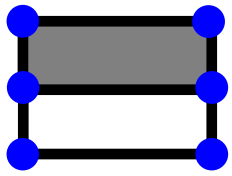
$$\sum_{\vec{r} \in ABCD} \mathcal{I}(\vec{r}) =$$

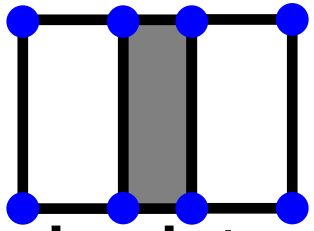
$$\mathfrak{I}(D) - \mathfrak{I}(B)$$

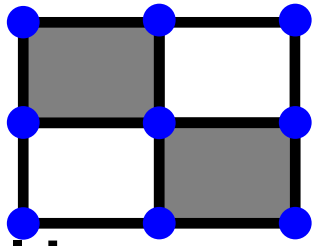
$$+\mathfrak{I}(A) - \mathfrak{I}(C)$$

A is calculated twice, need to add back

- Single rectangle: 4 array references
- Two-rectangle: 6 array references (<8, shared points)
- Three-rectangle: 8
- Four-rectangle: 9

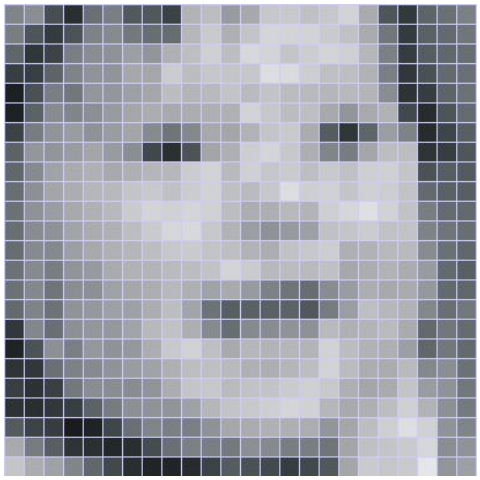
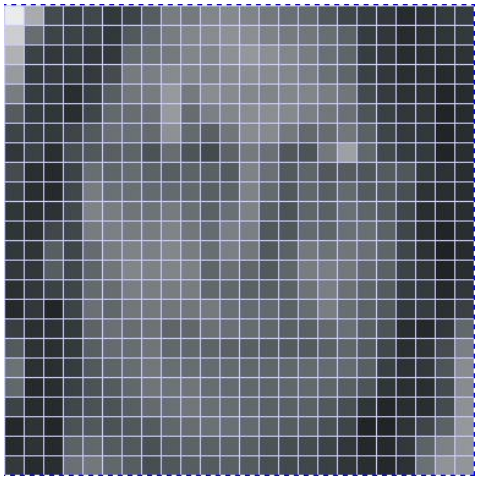
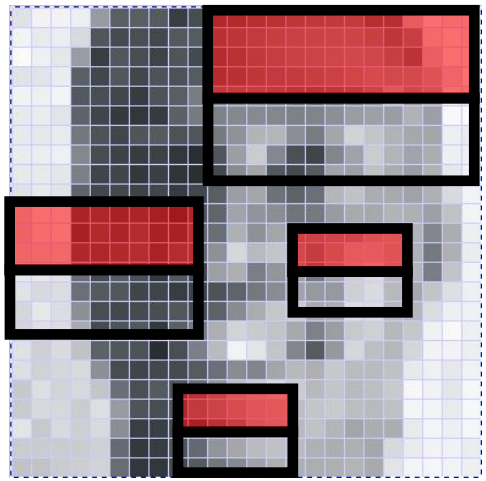




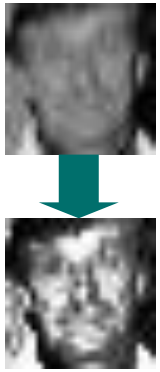


- Rectangle features efficient using integral image

Rectangle Features on Faces

- 24 x 24 frontal faces, roughly aligned
- 2,3 & 4 rectangle features -> 160,000 in total
- Intensity normalisation using variance:
  - sum of values & sum of values squared
  - integral image of image & integral image of squared image
- Which rectangle features?



See Appendix B

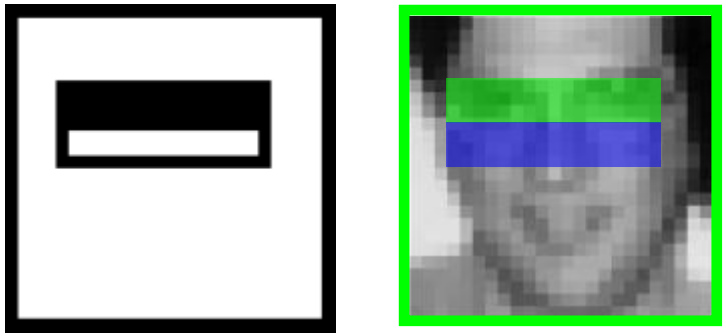
AdaBoost Face Classifier

- **Training Data:**

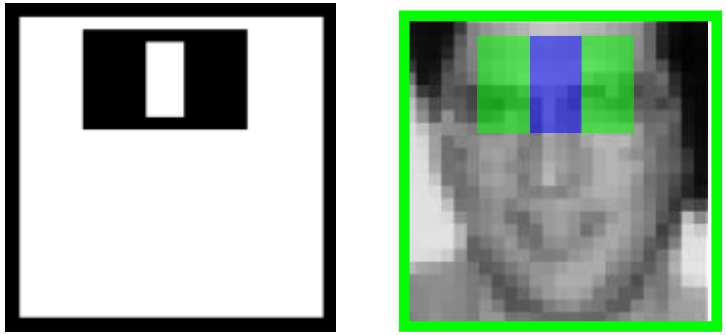
  5000 24 x 24 faces & 10,000 24 x 24 non-faces

- **Result: 200 features, 95% detection rate, 1 in 14,084 false positive rate**

- **AdaBoost first two features:**

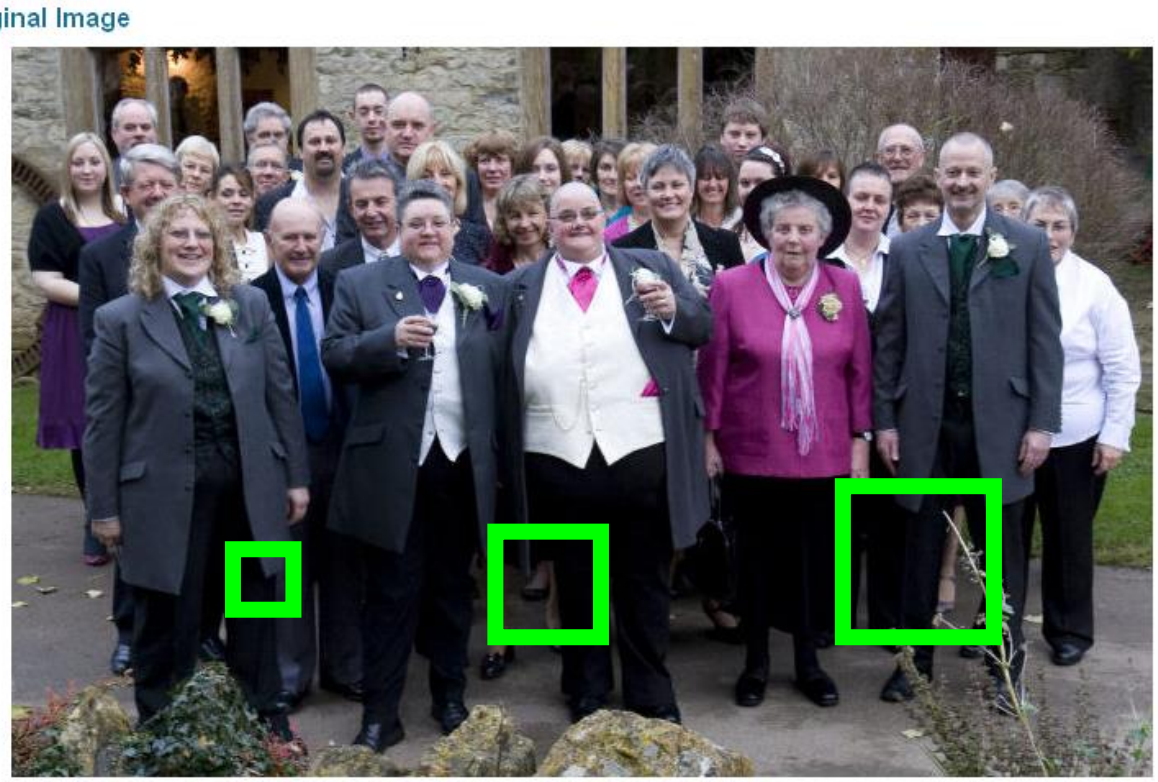


Eyes darker compared to cheekbones



Eyes darker compared to bridge of nose

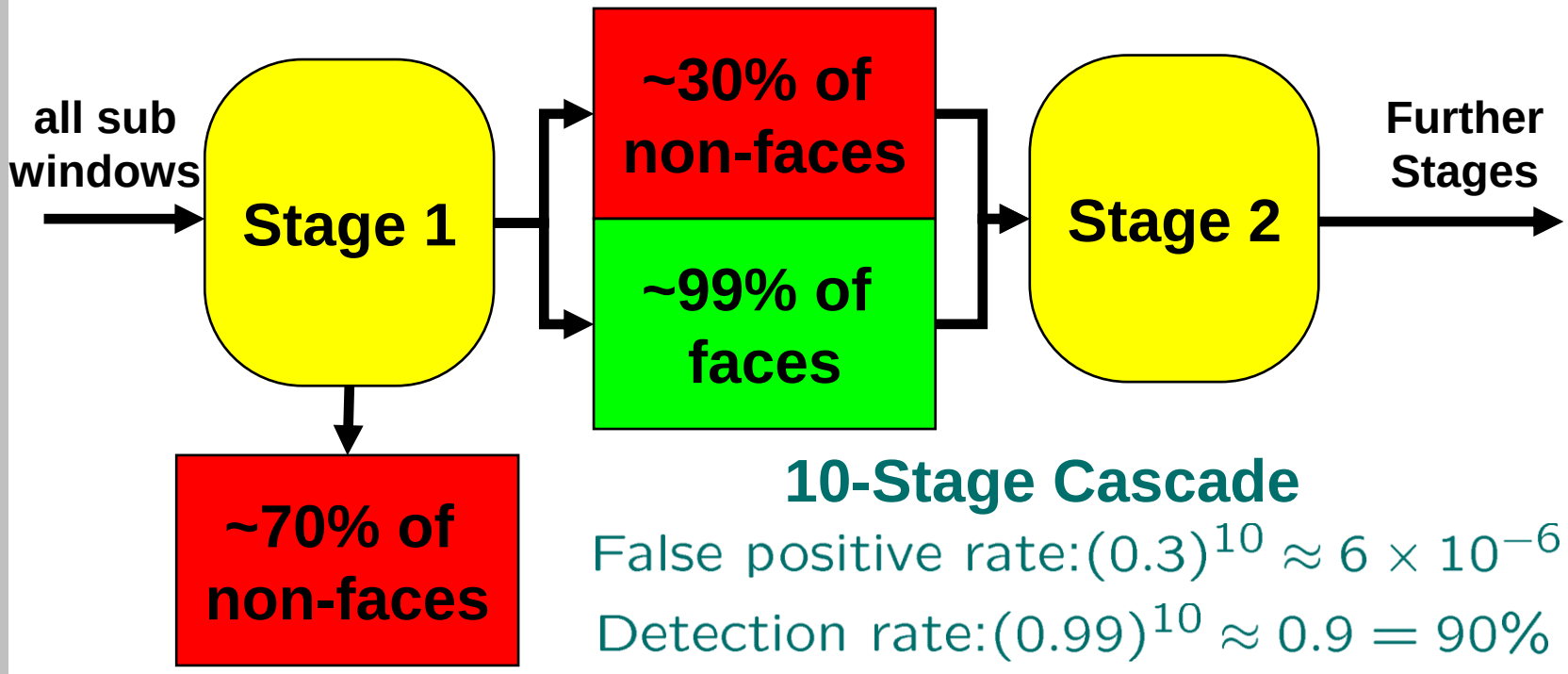
Searching the Whole Image:



- Have to consider a range of possible scales
- Have to consider all possible patch positions
- Most patches NOT faces!

Cascade Classifier

- Single-stage AdaBoost classifier too slow, but shows that rectangle features can detect faces
- In any image, most sub-windows are not faces



10-Stage Cascade

False positive rate: $(0.3)^{10} \approx 6 \times 10^{-6}$

Detection rate: $(0.99)^{10} \approx 0.9 = 90\%$

don't waste time on further processing

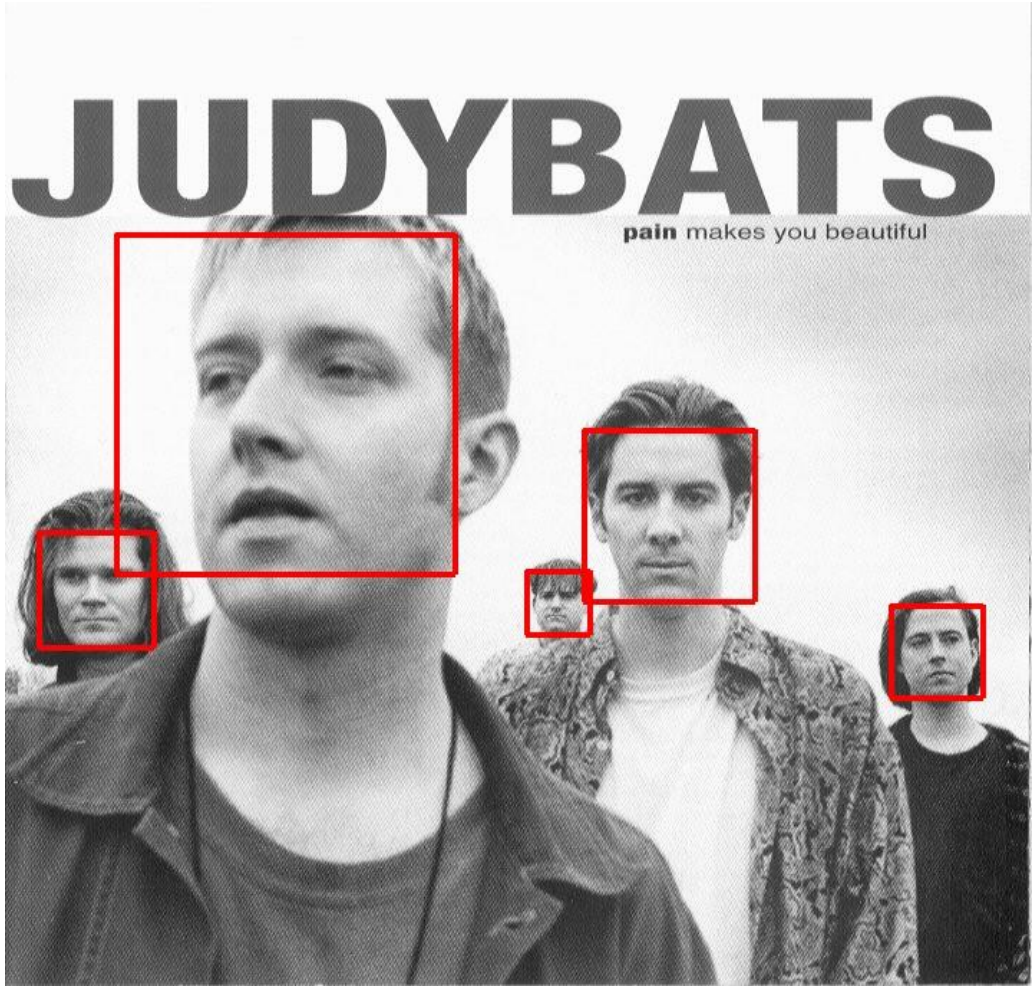
Viola & Jones Cascade Classifier

Paul Viola & Michael Jones, *Robust Real-Time Face Detection,* International Journal of Computer Vision, 57(2), 137—154 (2004)

- **38 stages of varying complexity, 6060 features in all**
- **Trained in a day using parallel processing**
- **Searches 384x288 pixel image in 0.067 seconds**
- **Applies features at multiple scales and locations**
- **Fifteen times faster than previous approaches**
- **Can be generalised to other objects (people/cars/cats/dogs)**

Original Face Detection Results:

- Works at various scales and small rotation

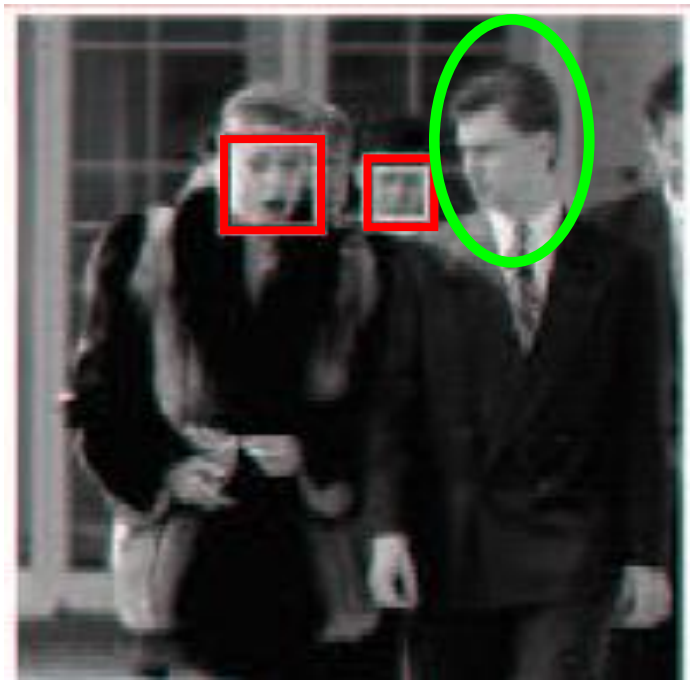


Original Face Detection Results:

- Finds multiple faces within a single image

Original Face Detection Results:

- Fails for faces in profile (as expected)
- Occasional mistakes



Online Demo: Upload your own image

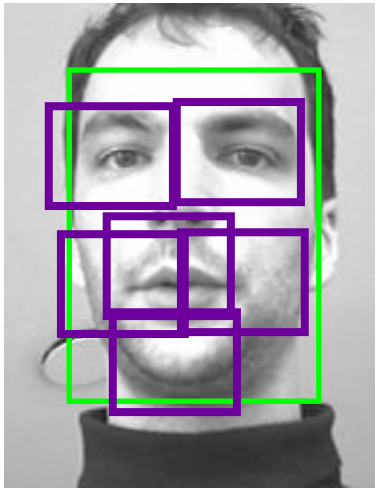
Image with Faces Detected

- Tudor Dobrila http://www.airtudor.com/ **Now defunct!**

Face Detection: Summary

- Fast detectors became available
  - Vary the features (Haar -> SURF etc), Vary classifiers
- Sufficient for commercial applications
- Face detection not the same as facial recognition
  - Identifying a person & their expression
- Apply similar feature detection to detected faces
- Try a different approach: shape models

Training Features



MANCHESTER 1824

The University of Manchester

Recording pause

MANCHESTER 1824

The University of Manchester

Appendix A:
Dogs & Cats

Detecting Cats:

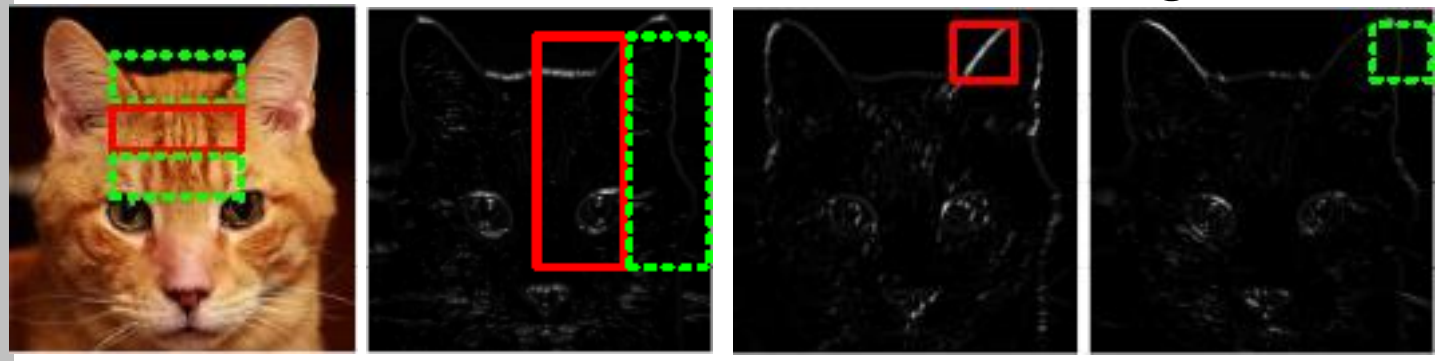
- **Cat faces: greater appearance variation than human**

  Cat Head Detection – Zhang, Sun, and Tang, ECCV 2008

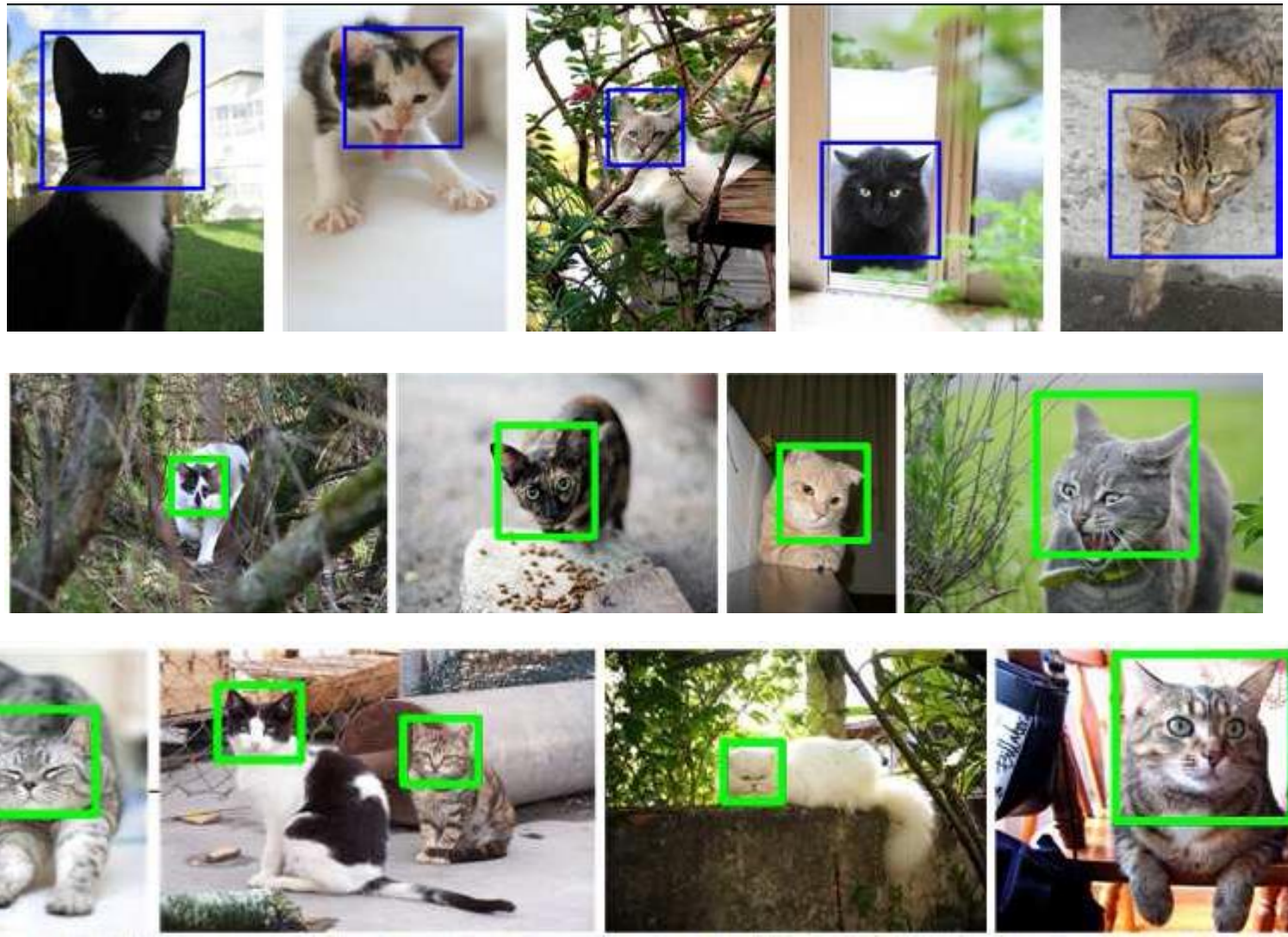
- **Use image intensity and image gradient channels**

Intensity Vertical Horizontal Two Diagonals

- **Within and between channel rectangle/Haar features**



Cat Head Detection: Results



MANCHESTER 1824

The University of Manchester

Recording pause

MANCHESTER 1824

The University of Manchester

Appendix B:
Intensity Normalisation & AdaBoost

Intensity Normalisation:

Mean within window: $\bar{\mathcal{I}} = \frac{1}{\mathrm{N}} \sum_{\tilde{\mathbf{r}} \in \mathbf{W}} \mathcal{I}(\tilde{\mathbf{r}})$

Variance within window: $\sigma^2 = \frac{1}{\mathrm{N}} \sum_{\tilde{\mathbf{r}} \in \mathbf{W}} (\mathcal{I}(\tilde{\mathbf{r}}) - \bar{\mathcal{I}})^2$

$$= -(\bar{\mathcal{I}})^2 + \frac{1}{\mathrm{N}} \sum_{\tilde{\mathbf{r}} \in \mathbf{W}} (\mathcal{I}(\tilde{\mathbf{r}}))^2$$

Intensity normalisation using mean & variance:

Shift to fixed mean & scale to unit variance

- sum of values

  & sum of values squared

- integral image of image

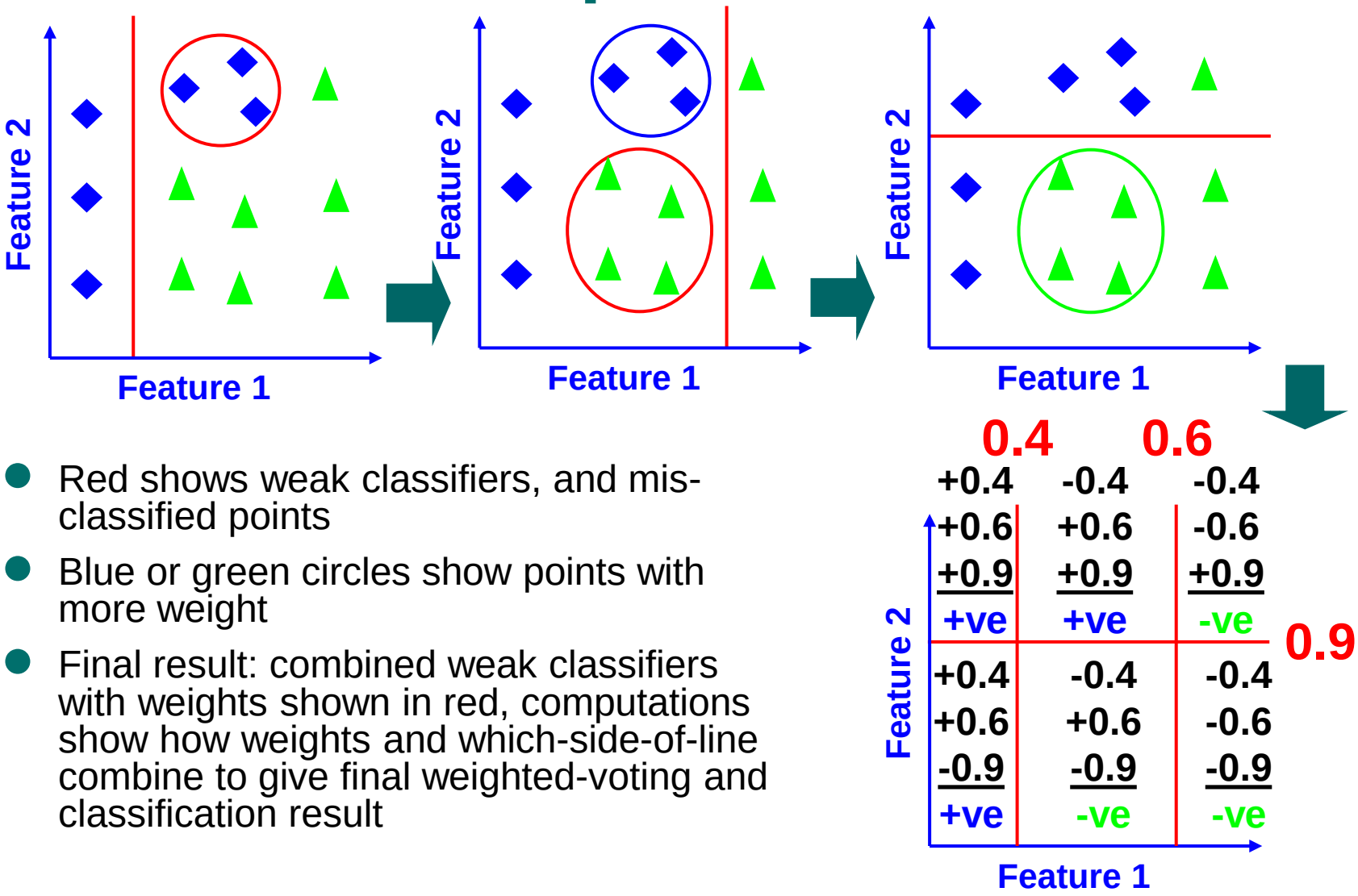
  & integral image of squared image

AdaBoost (Adaptive Boosting) Classifier:

- Combination of weak classifiers
- Each weak classifier is based on value of one feature

  value < threshold, or value > threshold
- Binary output from each weak classifier: ±1
- To build: add new weak classifier by concentrating on mis-classified examples at last round
- Output is computed by weighted voting from ALL of the weak classifiers
- Means you have to evaluate ALL of the chosen weak classifiers on each patch to get a decision

Adaboost Example:



- Red shows weak classifiers, and mis-classified points
- Blue or green circles show points with more weight
- Final result: combined weak classifiers with weights shown in red, computations show how weights and which-side-of-line combine to give final weighted-voting and classification result