Overview

- Real-time detection of faces in video
- Challenge: faces are rare.
- Study the widely-cited paper
- Basic concepts
  - Very fast image feature extraction.
  - Series of classifiers (nodes) with each classifier having a very low false negative rate.
  - Each classifier depends on a set of simply-computed image features.
  - Training occurs with an algorithm called AdaBoost.
  - Training is repeated several times over using a selection of positive and negative training examples. Each time, new (and more difficult) negative training examples are selected and added.
- In 2001, they achieved 15 frames per second detection on 384x288 video, apparently without any multi-threading.

Features — Integral Images

- Block-like feature extraction, similar to our initial box filters
- An integral image is
  \[ ii(x, y) = \sum_{x' \leq x, y' \leq y} I(x', y') \]  
  which is the sum of the intensity values above and to the left of location \((x, y)\).
- Fast computation in one pass over the entire image:
  \[ s(x, y) = s(x, y - 1) + I(x, y) \]  
  \[ ii(x, y) = ii(x - 1, y) + s(x, y) \]
- Once we have the integral image, we may quickly compute combinations of two, three and four rectangle features, mimicking the computation of first and second derivatives. We will derive some of these in class (and they may be read in the paper).
- The authors state that there are about 160,000 possible features in a 24x24 pixel region.
AdaBoost Classifier — Overview

- Starting point: a set of simple ("weak") classifiers, sometimes called “decision stumps”.
  - In our case they will be a single integral image feature, together with a threshold — very weak indeed.

Each of these classifiers can be trained.

- Requirement is that each have greater than a 50% success rate.
  - In our case, this is trivially-easy to achieve, since we can control the threshold and the sign of each weak classifier.

- Goal is to automatically learn, in a greedy fashion, both (a) a set of classifiers and (b) the weights of these classifiers.

- In the Viola/Jones paper, a series of AdaBoost classifiers is trained and applied in what is called a “cascade”. Our discussion will focus first on a single AdaBoost classifier and then on the cascade.

AdaBoost Classifier

- Goal is to learn a function

$$C(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$$  \hspace{1cm} (4)

which is a linear combination of weak classifiers.

  - $h_t(\cdot)$ is a trained classifier
  - $\alpha_t$ is a weight.

- We then apply a threshold to $C(x)$ to make the actual classification.

- Training algorithm given in paper consists of the following steps (see paper / lecture for details) in each iteration, starting at $t = 1$

  1. Normalize the weights
  2. Select the weak classifier that gives the best results based on the current weights.
  3. Define $h_t(\cdot)$ from the weak classifier.
  4. Update the weights.

- We will discuss the details in class, focusing on the training of the weak classifiers and the weights.

Cascade of Classifiers

- Series of classifiers, $i$, with $n_i$ integral image features per classifier.

- AdaBoost training for each.

- Start with an AdaBoost classifier applied to each region of the entire image, but using a small number, $n_0$, of integral image features.
• Apply each successive AdaBoost classifier, \( i \), to \textit{only to the regions that passed the previous classifier}.

• If \( p_i \) is the fraction of regions that pass the \( i \) classifier, then we can show that the example number of features evaluated per region is

\[
N = n_0 + \sum_{i=1}^{K} \left( n_i \prod_{j<i} p_j \right)
\]

which is a dramatic reduction in the number of features calculated.

**Training the Cascade**

We will go over the training in class, but here are some important properties

- User sets overall false positive rate
- Each cascade is trained by adding one feature at a time until a per classifier error rate is achieved.
- False detections from a validation set are added to the training set in the next round.

**Experimental Details and Results**

- Web crawl obtained images
- 38-layer cascade with 6060 features, overall
- Manually selected the number of features per layer in the first seven layers.
- Later layers added 25 features at a time to increase the speed of training
- Training time was weeks
- Features at multiple locations and scales.
- Shifts between regions at different scales
- Post-processing to avoid multiple detections
- Final results on best two-out-of-three voting scheme
- Failure modes:
  - More than 15° in plane rotation
  - More than 45° out-of-plane rotation
  - Backlighting
  - Occlusion

**Looking Ahead**

- Transformations
- Cameras and calibration
- Image registration