CS 461: Computer Vision
Professor Hager

Face Detection
Raphael Sznitman
Face Detection

- Goal: To identify and locate human faces in images.
- Faces may show variation in their location, scale, rotation and illumination.
  - Will focus here on location, scale and illumination
- A natural step for automatic face recognition.
- A very difficult problem!
Classification and Localization

▶ **Classification** Is this a face?
Classification and Localization

- **Classification** Is this a face?
- **Localization** Where is the face?
Classification and Localization
Classification and Localization
Classification and Localization
Classification and Localization
Classification and Localization
Classification and Localization
Classification and Localization
Face Detection

A little bit of history:

- A problem studied for the last 40 years.
- A myriad of methods have been proposed
  - Subspace based (Turk and Pentland, 1991)
  - Distribution based (Sung and Poggio, 1995)
  - Neural Network based (Rowley et al., 1998)
  - Boosting based (Viola and Jones, 2001)
  - And many many more...
Given a training set: \( \mathcal{L} = \{(x_1, y_1), \ldots, (x_n, y_n)\} \), our goal is to build a classifier:

\[
C_{\mathcal{L}} : \mathcal{X} \rightarrow \{+1, -1\}
\]

We want our classifier to be of the form:

\[
F(x) = \text{sign} \left( \sum_{m=1}^{M} \alpha_m h_m(x) \right)
\]

- \( h_m(x) \) is a “weak classifier” - does slightly better than chance when classifying \( x \).
- \( \alpha \in \mathbb{R} \) is weight associated with the strength of the “weak classifier”.
- \( M \) is number of classifier we will use.
- Assume that we have a family, \( H \), of such “weak classifiers”.

A Very Short Introduction to Boosting
A Very Short Introduction to Boosting

We want our classifier to be of the form:

\[ F(x) = \text{sign}(\sum_{m=1}^{M} \alpha_m h_m(x)) \]

We will do this iteratively:

\[ m = 1 : f_0(x) = \alpha_1 h_1(x) \]
\[ m = 2 : f_1(x) = f_0(x) + \alpha_1 h_1(x) \]
\[ \vdots \]
\[ m = M : f_M(x) = f_{M-1}(x) + \alpha_m h_m(x) \]

Associate with each training example a weight, such that

\[ \mathcal{L} = \{(x_1, y_1, w_1), \ldots, (x_n, y_n, w_n)\} \]
A Very Short Introduction to Boosting

- Picking $h_m$ and $\alpha_m$ involves minimizing the exponential loss function

\[
L(f_m) = \sum_{i=1}^{n} e^{-y_if_m(x_i)} = \sum_{i=1}^{n} e^{-y_i(f_{m-1}(x_i) + \alpha_m h_m(x_i))} = \sum_{i=1}^{n} w_i^{m-1} e^{-y_i \alpha_m h_m(x_i)}
\]

with respect to $h_m$ and $\alpha_m$ (see additional handout for derivation).

- Could use other loss functions.
Discrete Adaboost  (Freund & Schapire, 1996)

- Input: $H, \mathcal{L}, M$
- Output: $F : \mathcal{X} \rightarrow \{+1, -1\}$

1. Initialize: $w_i^1 = 1/n$
2. For $m = 1, \ldots, M$
   i. Choose a weak classifier
      $$h_m = \arg\min_h \sum_{i=1}^{n} w_i^{t-1} I(y_i \neq h(x_i))$$
   ii. Compute the training error
       $$e_m = \frac{\sum_{i=1}^{n} w_i^m I(y_i \neq h(x_i))}{\sum_{i=1}^{n} w_i^m}$$
   iii. Compute $\alpha_m = \frac{1}{2} \ln\left(\frac{1-e_m}{e_m}\right)$
   iv. Update weights by:
       $$w_i^{m+1} = w_i^m e^{-\alpha_m y_i h_m(x_i)}$$
A simple example

**Toy Example**

*weak classifiers = vertical or horizontal half-planes*

Example Taken from Robert Schapire
A simple example

Round 1

\[ n_1 \]

\[ \epsilon_1 = 0.30 \]
\[ \alpha_1 = 0.42 \]

\[ n_2 \]
A simple example

Round 2

\[ \epsilon_2 = 0.21 \]
\[ \alpha_2 = 0.65 \]
A simple example

Round 3

$\epsilon_3 = 0.14$
$\alpha_3 = 0.92$
A simple example

Final Classifier

$H_{\text{final}} = \text{sign}(0.42 + 0.65 + 0.92)$
Can initialize the weights in different ways depending on the data:

\[
w_i^1 = \begin{cases} 
\frac{1}{2} |L^+| & y_i = 1 \\
\frac{1}{2} |L^-| & y_i = -1 
\end{cases}
\]

\(\alpha_m\) is a measure of performance of \(h_m\).

Weights measure the difficulty to classify the data point:

\[
w^{m+1} = \begin{cases} 
w^m e^{-\alpha_m} & y_i = h_m(x_i) \\
w^m e^{\alpha_m} & y_i \neq h_m(x_i) \end{cases}
\]
Viola and Jones Face Detection

- A widely used method for real-time object detection. Can also be applied to a wider range of objects.
- Training is slow, but detection is very fast:
  - Training is done using Boosting
  - Detection is done by a Cascade of Boosted Classifiers.
Training Data

- 5000 faces (all frontal)
- 10000 non-faces
- Faces are normalized for scale and translation.

Many variations

- Across individuals
- Illumination
- Pose
Creating a Boosted Classifier for Faces

- Using the data presented, we want to build:

\[ C_L(x) = \text{sign}\left( \sum_{m=1}^{M} \alpha_m h_m(x) \right) \]

- We will use Adaboost to do this:
  - Need to specify a family of “weak learners”, \( H \).
  - Need to specify how many “weak learners” to use, \( M \).
Features

- “Weak Learners” are face features!
- Inspired from Haar Basis Functions
- These are rectangular features, summing pixel values such that light regions are added and dark regions are subtracted.
- Allows for an over complete set of weak learners (∼160’000).
In order to compute our features quickly, we convert our image into an Integral Image.

The Integral Image computes a value at each pixel \((x, y)\) that is the sum of the pixel values above and to the left of \((x, y)\), inclusive.
Integral Images

- Using the integral image, any rectangle can be computed by 3 additions

\[ \text{Sum} = A - B - C + D \]

- Can compute integral images for edges, edge oriented histograms, ...

- Great tool for real-time processing.
Learned Classifier and Features Selected

- Build a 200 feature classifier from the procedure described.
- Features selected are (some) representative of the task at hand.
ROC Curve

▶ ROC (Receiver Operating Characteristic) curves provide a good measure of how well a classifier is performing.
▶ Each point on the curve displays the performance of the classifier for some threshold value.
▶ Plots the true positive rate (tpr) against the false positive rate (fpr).
  ▶ \( tpr = \frac{TP}{TP + FN} \)
  ▶ \( fpr = \frac{FP}{FP + TN} \)
Attentional Cascade

- A 1 feature classifier achieves 100% detection rate and about 50% false positive rate.
- A 5 feature classifier achieves 100% detection rate and 40% false positive rate (20% cumulative) using data from previous stage.
- A 20 feature classifier achieve 100% detection rate with 10% false positive rate (2% cumulative)
Speed up by a factor of 10 and performance does not suffer much
Some Results (MIT+CMU Dataset)
For Face Profiles
So is Face Detection solved?

- Viola and Jones provide a method to detect faces accurately in images. Detects faces with 95%.
  - Note: If you really think about it, this is NOT so good! Image you are speaking with someone for 10 minutes, for a whole 30 seconds, you can’t see who you are talking to.
- What happens for more complex objects?
  - End up fragmenting the data suing this technique. This is problematic for complex objects.
- Training in is slow, and requires a lot of samples.
Example: Human concept learning (J. Tenenbaum)
Example: Human concept learning (J. Tenenbaum)
If you are to remember anything from today...

- Boosting works well for faces.
- Integral Images are key for real-time applications.
- Cascaded Classifiers reduce computation by spending more time on difficult examples.
- ROC Curves allow us to visualize classifier performance.
- Face detection is not a solved problem.
Any Last Questions?