Histograms of Oriented Gradients

COMPSCI 527 — Computer Vision
Outline

1. Hand-Crafted Features and Deep Neural Networks
2. Features for Object Detection
3. Histogram of Oriented Gradients
Before 2010, the feature vector $x$ was typically designed by hand

After 2010, features are typically learned from examples

We will look at a good hand-design example to see what is expected of a feature vector

*Histograms of Oriented Gradients*, Dalal and Triggs, 2005 (paper is mandatory reading)

Insights from hand design can guide design of feature learners

In many applications, the cost of learning features is still prohibitive
Features for Object Detection

- Pedestrians are easier to detect than people (template-like)
- Compromise window size: $128 \times 64$. Pyramids for scale
- Check (almost) all possible window positions in the pyramid
- Ideal feature vector: Remove what’s irrelevant to detection, keep what’s essential

“distribution of local intensity gradients [...] without precise knowledge of [...] positions; [...] contrast-normalize the responses”
Simplified Gradient

- We only want to know where there is “sudden change” of brightness, to find edges
- Actual values are not too important
- Smoothing is harmful
- Central differences: $I_x(r, c) = I(r, c + 1) - I(r, c - 1)$
  and $I_y(r, c) = I(r - 1, c) - I(r + 1, c)$
- Don’t bother dividing by 2, because we’ll normalize later
- Transform to polar coordinates because we care about orientations:
  $\mu = \sqrt{I_x^2 + I_y^2}$ and
  $\theta = \frac{180}{\pi} \left( \tan^{-1} \left( \frac{I_y}{I_x} \right) \mod \pi \right)$
- $\theta \in [0, 180)$, so contrast direction does not matter
Cells

- The $128 \times 64$ window is divided into $16 \times 8$ cells of $8 \times 8$ pixels each.
- Each cell contributes 64 magnitudes $\mu$ and orientations $\theta$.
- Build a 9-bin histogram $c$ of $\theta$, weighed by $\mu$:
  
  $0 \leq \theta < 20$, $20 \leq \theta < 40$, $\ldots$, $160 \leq \theta < 180$
“Bilinear” Voting

- Each pixel in a cell gets a vote equal to $\mu$
- If $\theta$ changes slightly, the vote can fall in a different cell

To reduce this quantization issue, a vote is partitioned over two adjacent cells (with wrap-around), in amounts inversely proportional to distance to cell centers.
Contrast Normalization

- A $2 \times 2$ array of cells is a block
- All possible blocks are considered $\Rightarrow$ 15 $\times$ 7 blocks
- Concatenate four histograms in a block into $\mathbf{b}$ (36 entries)
- Normalize each block histogram $\mathbf{b}$ to unit norm
- Concatenate the 105 $\mathbf{b}$ histograms into $\mathbf{h}$ (3780 entries)
- Normalize $\mathbf{h}$ once more, saturate entries, normalize again
Contrast Normalization Details

\[
\mathbf{b} = [\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \mathbf{c}_4]^T
\]

\[
\mathbf{b} \leftarrow \frac{\mathbf{b}}{\sqrt{||\mathbf{b}||^2 + \epsilon}}
\]

\[
\mathbf{h} = [\mathbf{b}_1, \ldots, \mathbf{b}_{105}]^T
\]

\[
\mathbf{h} \leftarrow \frac{\mathbf{h}}{\sqrt{||\mathbf{h}||^2 + \epsilon}}
\]

\[
h_n \leftarrow \min(h_n, \tau) \quad (\tau = 0.2)
\]

\[
\mathbf{h} \leftarrow \frac{\mathbf{h}}{\sqrt{||\mathbf{h}||^2 + \epsilon}}
\]

- Each of the 128 cell histograms shows up about 4 times, normalized differently
- Compromise: preserve some relative contrast (within each block), but normalize away overall image contrast
Summary

- A vector with 3780 features describes a window with $128 \times 64 = 8129$ pixels
- Precise color and contrast position and direction are abstracted away by cells and bins
- Overall contrast is abstracted away but some local contrast information is retained
Performance

- HOG descriptors work quite well
- Dalal and Triggs use them as inputs to a Support Vector Machine classifier (most classifiers would do)
- Non-maximum suppression required