CPSC 425: Computer Vision

Instructor: Jim Little
little@cs.ubc.ca

Department of Computer Science
University of British Columbia

Lecture Notes 2016/2017 Term 2
Menu March 28, 2017

Topics:
   Image Classification (cont.)
   Object Detection

Reading:
   Today: Forsyth & Ponce (2nd ed.) 17.1–17.2

Handouts:
   Assignment 7: Video Segmentation

Reminders:
   Assignment 6 due now
   Assignment 7 due Thursday, April 6
Today’s Fun Example

Video clip: ‘Visual Microphone’
Follow-up work to previous lecture’s example of Eulerian video magnification
Lecture 20: Re-cap

- A decision tree passes a data point through a sequence of feature tests. A random forest is an ensemble of decision trees.

- Factors that make image classification hard
  - intra-class variation, viewpoint, illumination, clutter, occlusion...

- A codebook of **visual words** contains representative local patch descriptors
  - can be constructed by clustering local descriptors (e.g. SIFT) in training images
Lecture 20: Re-cap

- The **bag of words** model accumulates a histogram of occurrences of each visual word

- The **spatial pyramid** partitions the image and counts visual words within each grid box; this is repeated at multiple levels
VLAD (Vector of Locally Aggregated Descriptors)

- There are more advanced ways to ‘count’ visual words than incrementing its histogram bin.

- For example, it might be useful to describe how local descriptors are quantized to their visual words.

- In the VLAD representation, instead of incrementing the histogram bin by one, we increment it by the residual vector $\mathbf{x} - \mathbf{c(x)}$. 
VLAD (Vector of Locally Aggregated Descriptors)
VLAD (Vector of Locally Aggregated Descriptors)

- The dimensionality of a VLAD descriptor is $Kd$
  - $K$: number of codewords
  - $d$: dimensionality of the local descriptor

- VLAD characterizes the distribution of local descriptors with respect to the codewords
Object Detection: Introduction

- We have been discussing image classification, where we pass a whole image into a classifier and obtain a class label as output.

- We assumed the image contained a single, central object.

- The task of object detection is to detect and localize all instances of a target object class in an image.
  — Localization typically means putting a tight bounding box around the object.
Sliding Window

- Train an image classifier as described previously. ‘Slide’ a fixed-sized detection window across the image and evaluate the classifier on each window.

Image credit: KITTI Vision Benchmark
Train an image classifier as described previously. ‘Slide’ a fixed-sized detection window across the image and evaluate the classifier on each window.

This is a search over location
   — We have to search over scale as well
   — We may also have to search over aspect ratios

Image credit: KITTI Vision Benchmark
Example 1: Face Detection

- The Viola-Jones face detector is a classic sliding window detector that learns both efficient features and a classifier.

- A key strategy is to use features that are fast to evaluate to reject most windows early.

- The Viola-Jones detector computes ‘rectangular’ features within each window.
Example 1: Face Detection

- A ‘rectangular’ feature is computed by summing up pixel values within rectangular regions and then differencing those region sums.

Figure credit: P. Viola and M. Jones, 2001
Integral Image

- Rectangular features can be computed quickly using an integral image
- Let $\hat{I}$ denote the integral image of $I$. Then the value of $\hat{I}$ at $(x,y)$ is the sum of all values in $I$ above and to the left of $(x,y)$

$$\hat{I}_{ij} = \sum_{u=1}^{i} \sum_{v=1}^{j} I_{uv}$$ (1)

Figure credit: P. Viola
A Short Exercise

Find a general formula to calculate the sum within a rectangular region of an image, given the integral image. Here is a sample image:

\[
\begin{array}{cccc}
0.8 & 0.6 & 0.9 & 0.9 \\
0.9 & 0.1 & 0.9 & 0.5 \\
0.2 & 0.3 & 0.2 & 0.8 \\
0.9 & 0.5 & 0.9 & 0.2 \\
\end{array}
\]

and its integral image:

\[
\begin{array}{cccc}
0.8 & 1.4 & 2.3 & 3.2 \\
1.7 & 2.4 & 4.2 & 5.6 \\
1.9 & 2.9 & 4.9 & 7.1 \\
2.8 & 4.3 & 7.2 & 9.6 \\
\end{array}
\]

Your answer should be independent of the size of the region you are summing over.
Integral Image

- Given an integral image, the sum within a rectangular region in $I$ can be computed with just 3 additions.

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

- Constant time: does not depend on the size of the region.
- We can avoid scaling images - just scale features directly.

Figure credit: P. Viola
Example 1: Face Detection

- Many possible rectangular features

Figure credit: B. Freeman
Example 1: Face Detection

- Use boosting to both select the informative features and form the classifier. Each round chooses a weak classifier that simply compares a single rectangular feature against a threshold.

Figure credit: P. Viola and M. Jones, 2001
Cascading Classifiers

To make detection faster, features can be reordered by increasing complexity of evaluation and the thresholds adjusted so that the early (simpler) tests have few or no false negatives.

Any window that is rejected by early tests can be discarded quickly without computing the other features.

This is referred to as a cascade architecture.
Cascading Classifiers

A classifier in the cascade is not necessarily restricted to a single feature

Figure credit: P. Viola
Example 1: Face Detection

Summary:

Train cascade of classifiers with AdaBoost

Apply to each subwindow

Figure credit: K. Grauman
Example 1: Face Detection

Just for fun:

"CV Dazzle, a project focused on finding fashionable ways to thwart facial-recognition technology"

Figure source: Wired, 2015.
Example 2: Pedestrian Detection

- The sliding window approach applies naturally to pedestrian detection because pedestrians tend to take characteristic poses, (e.g. standing, walking)

![Image](image.png)

Image window; Visualisation of HOG features; HOG features weighted by positive weights; HOG features weighted by negative weights

Fig. 17.7 in Forsyth & Ponce (2nd ed). Original source: Dalal and Triggs, 2005.
Deformable Part Model

- Sliding window detectors tend to fail when the object is not well described by a rigid template

Felzenszwalb et al., 2010

- Many complex objects are better represented using a parts model
A deformable part model consists of a root and a set of parts
— Root: an approximate model that gives the overall location of the object
— Parts: object components that have reliable appearance but might appear at somewhat different locations on the root for different instances
Deformable Part Model

- Each part has an appearance model and a natural location relative to the root.
- Finding a window that looks a lot like the part close to that part’s natural location relative to the root yields evidence that the object is present.

Felzenszwalb et al., 2010
A parts model for a bicycle, containing a root and 6 parts

Figure source: Felzenszwalb et al., 2010
Summary

- VLAD accumulates the residual of each local descriptor with respect to its corresponding visual word

- Object detection: detect and localize all instances of an object class in an image

- The sliding window strategy moves a detection window across the image and evaluates a trained classifier on each window

- A deformable part model is useful when the object is not well represented by a rigid template
  — model consists of a root and a set of parts
  — detection scores based on both appearance and location