Object Category Detection

Andrew Zisserman

Visual Geometry Group
University of Oxford
http://www.robots.ox.ac.uk/~vgg
What we would like to be able to do…

• Visual scene understanding
• **What** is in the image and **where**

• Object categories, identities, properties, activities, relations, …
**Things vs. Stuff**

**Thing (n):** An object with a specific size and shape.

**Stuff (n):** Material defined by a homogeneous or repetitive pattern of fine-scale properties, but has no specific or distinctive spatial extent or shape.

---

Ted Adelson, Forsyth et al. 1996.

Slide: Geremy Heitz
Recognition Tasks

• **Image Classification**
  – Does the image contain an aeroplane?

• **Object Class Detection/Localization**
  – Where are the aeroplanes (if any)?

• **Object Class Segmentation**
  – Which pixels are part of an aeroplane (if any)?
Challenges: Background Clutter
Challenges: Occlusion and truncation
Challenges: Intra-class variation
Why detection?

- Spatial relationships for image understanding
- Spatial relationships for image retrieval
- Tracking by detection

“a cat riding a skateboard”
Motivation/Applications

Collision prevention

www.mobileye.com

Slide: Ross Girshick
Funny Nikon ads

"The Nikon S60 detects up to 12 faces."

Slide: Svetlana Lazebnik
Funny Nikon ads

"The Nikon S60 detects up to 12 faces."

Slide: Svetlana Lazebnik
Preview of typical results
Outline

Part I: Sliding window detectors & HOG
  • Introduce HOG
  • Train a sliding window detector
  • Speeding up inference

Part II: Evaluation and state of the art
Problem of background clutter

• Use a sub-window
  – At correct position, no clutter is present
  – Slide window to detect object
  – Change size of window to search over scale
Detection by Classification

- Basic component: binary classifier
Detection by Classification

• Detect objects in clutter by **search**

• **Sliding window**: exhaustive search over position and scale
Detection by Classification

• Detect objects in clutter by **search**

• **Sliding window**: exhaustive search over position and scale
Detection by Classification

- Detect objects in clutter by **search**

- **Sliding window**: exhaustive search over position and scale (can use same size window over a spatial pyramid of images)
Window (Image) Classification

Features usually engineered
Classifier learnt from data

\( P(c|x) \propto F(x) \)
Problems with sliding windows ...

- aspect ratio
- granularity (finite grid)
- partial occlusion
- multiple responses
Pedestrian detection

- Objective: detect (localize) standing humans in an image
- Sliding window classifier
- Train a binary SVM classifier on whether a window contains a standing person or not
- Histogram of Oriented Gradients (HOG) feature
- Although HOG + SVM originally introduced for pedestrians has been used very successfully for many object categories
Feature: Histogram of Oriented Gradients (HOG)

- tile 64 x 128 pixel window into 8 x 8 pixel cells
- each cell represented by histogram over 8 orientation bins (i.e. angles in range 0-180 degrees)
Histogram of Oriented Gradients (HOG) continued

• Adds a second level of overlapping spatial bins re-normalizing orientation histograms over a larger spatial area

• Feature vector dimension (approx) = 16 x 8 (for tiling) x 8 (orientations) x 4 (for blocks) = 4096
Window (Image) Classification

- HOG Features
- Linear SVM classifier

Training Data

Feature Extraction

Classifier $F(x)$

$x$

$P(c|x) \propto F(x)$

pedestrian/Non-pedestrian
Tiling defines (records) the spatial correspondence
Averaged examples
Advantages of linear SVM:

• Training (Learning)
  
  • Very efficient packages for the linear case, e.g. LIBLINEAR for batch training and Pegasos for on-line training.
  
  • Complexity $O(N)$ for $N$ training points (cf $O(N^3)$ for general SVM)

• Testing (Detection)

Non-linear

$$f(x) = \sum_{i}^{S} \alpha_i k(x_i, x) + b$$

S = # of support vectors

= (worst case ) $N$

size of training data

Linear

$$f(x) = \sum_{i}^{S} \alpha_i x_i^\top x + b$$

$$= w^\top x + b$$  Independent of size of training data
Learned model

\[ f(x) = w^\top x + b \]
What do negative weights mean?

\[ wx > 0 \]
\[ (w_+ - w_-)x > 0 \]
\[ w_+ > w_- x \]

pedestrian model > pedestrian background model

Complete system should compete pedestrian/pillar/doorway models

Discriminative models come equipped with own bg
(avoid firing on doorways by penalizing vertical edges)

Slide from Deva Ramanan
What is represented by HOG

Inverting and Visualizing Features for Object Detection

Carl Vondrick Aditya Khosla Tomasz Malisiewicz Antonio Torralba

http://web.mit.edu/vondrick/ihog/index.html
What is represented by HOG

HOG

HOG Inverse Original
Why does HOG + SVM work so well?

- Similar to SIFT, records spatial arrangement of histogram orientations
- Compare to learning only edges:
  - Complex junctions can be represented
  - Avoids problem of early thresholding
  - Represents also soft internal gradients
- Older methods based on edges have become largely obsolete

- HOG gives fixed length vector for window, suitable for feature vector for SVM
Training a sliding window detector

- Object **detection** is inherently asymmetric: much more “non-object” than “object” data

- Classifier needs to have very low false positive rate
- Non-object category is very complex – need lots of data
Bootstrapping

1. Pick negative training set at random
2. Train classifier
3. Run on training data
4. Add false positives to training set
5. Repeat from 2

- Collect a finite but diverse set of non-object windows
- Force classifier to concentrate on **hard negative** examples
- For some classifiers can ensure equivalence to training on entire data set
Example: train an upper body detector

- Training data – used for training and validation sets
  - 33 Hollywood2 training movies
  - 1122 frames with upper bodies marked

- First stage training (bootstrapping)
  - 1607 upper body annotations jittered to 32k positive samples
  - 55k negatives sampled from the same set of frames

- Second stage training (retraining)
  - 150k hard negatives found in the training data
Training data – positive annotations
Positive windows

Note: common size and alignment
Jittered positives
Jittered positives
Random negatives
Random negatives
Window (Image) first stage classification

- Jittered positives
- random negatives

HOG Feature Extraction

\[ f(x) = w^T x + b \]

Linear SVM Classifier

- find high scoring false positives detections
- these are the hard negatives for the next round of training

- cost = # training images x inference on each image
Hard negatives
Hard negatives
First stage performance on validation set
Reminder: Precision – Recall curve

- **Precision**: % of returned windows that are correct
- **Recall**: % of correct windows that are returned

Classifier score decreasing
First stage performance on validation set
Performance after retraining
Effects of retraining
Side by side

before retraining

after retraining
Side by side

before retraining

after retraining
Side by side

before retraining

after retraining
Tracked upper body detections
Accelerating Sliding Window Search

• Sliding window search is slow because so many windows are needed e.g. \(x \times y \times \text{scale} \approx 100,000\) for a 320\(\times\)240 image

Example: face detection

• Most windows are clearly not the object class of interest
• Can we speed up the search?
Cascaded Classification

- Build a sequence of classifiers with increasing complexity

More complex, slower, lower false positive rate

- Reject easy non-objects using simpler and faster classifiers
Cascaded Classification

- Slow expensive classifiers only applied to a few windows ⇒ significant speed-up

- Controlling classifier complexity/speed:
  - Number of support vectors [Romdhani et al, 2001]
  - Number of features [Viola & Jones, 2001]
  - Type of SVM kernel [Vedaldi et al, 2009]
  - Number of parts [Felzenszwalb et al, 2011]
Detection Proposals

• Propose image regions that contain objects (rather than stuff)

• Proposals can be boxes or segmented regions and are class agnostic

• Aim to cover all the objects in the image with a small number of proposals, e.g. 100-1000 per image

• “Objectness” Alexe et al, PAMI 2012
Selective Search for Object Recognition
J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, A. W. M. Smeulders
Detection Proposals – example methods 2

Edge Boxes: Locating Object Proposals from Edges
Larry Zitnick & Piotr Dollár,
ECCV 2014
Detection Proposals – example methods 3

Learning to propose Objects
Philipp Krähenbühl and Vladlen Koltun
CVPR 2015

Further reading:
What makes for effective detection proposals?
J. Hosang, R. Benenson, P. Dollár, and B. Schiele.
PAMI 2015.