Intro to Object Detection

CSC2548, 2018 Winter
Bin Yang
17 Jan. 2018

“If I have seen further it is by standing on the shoulders of giants.” - Isaac Newton

slides adopted from Ross Girshick, Chris McIntosh, Sanja Fidler, Mubarak Shah and many others
Object detection

- Introduction
- Face detection: from Viola-Jones to CNN
- General object detection
  - HOG detector
  - Deformable Part-based Model
  - Region-CNN
  - Fast versions of R-CNN
  - YOLO/SSD
- Future directions
Object detection

• Introduction
  • Face detection: from Viola-Jones to CNN
  • General object detection
    • HOG detector
    • Deformable Part-based Model
    • Region-CNN
    • Fast versions of R-CNN
    • YOLO/SSD
  • Future directions
Formalizing the object detection task

Many possible ways, this one is popular:

- Input
- Desired output

- cat, dog, chair, cow, person, motorbike, car, ...

slide credit: Ross Girshick
Formalizing the object detection task

Many possible ways, this one is popular:

Input

Desired output

Performance summary:

Average Precision (AP)
0 is worst  1 is perfect

slide credit: Ross Girshick
Object detection

- Introduction
- Face detection: from Viola-Jones to CNN
- General object detection
  - HOG detector
  - Deformable Part-based Model
  - Region-CNN
  - Fast versions of R-CNN
  - YOLO/SSD
- Future directions

Cascaded Classifier

- 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)
VJ face detection results
CNN based face detector (H. Qin, 2016)
Demo

Object detection

- Introduction
- Face detection: from Viola-Jones to CNN
- General object detection
  - HOG detector
  - Deformable Part-based Model
  - Region-CNN
  - Fast versions of R-CNN
  - YOLO/SSD
- Future directions
The HOG Detector

N. Dalal and B. Triggs

*Histograms of oriented gradients for human detection*

CVPR, 2005


cited by 17,502
HOG detector: pipeline

1. Scan image(s) at all scales and locations
2. Extract features over windows
3. Run linear SVM classifier on all locations
4. Fuse multiple detections in 3-D position & scale space
5. Object detections with bounding boxes
I. Sliding window

- Scan image(s) at all scales and locations
  - Extract features over windows
  - Run linear SVM classifier on all locations
  - Fuse multiple detections in 3-D position & scale space
  - Object detections with bounding boxes

Slide credit: Sanja Fidler
II. Histograms of Oriented Gradients

- Scan image(s) at all scales and locations
- Extract features over windows
- Run linear SVM classifier on all locations
- Fuse multiple detections in 3-D position & scale space
- Object detections with bounding boxes

Computes gradients → Weighted vote into spatial & orientation cells → Contrast normalize over overlapping spatial blocks

Slide credit: Sanja Fidler
II. Histograms of Oriented Gradients
II. Histograms of Oriented Gradients

1. Scan image(s) at all scales and locations
2. Extract features over windows
3. Run linear SVM classifier on all locations
4. Fuse multiple detections in 3-D position & scale space
5. Object detections with bounding boxes
6. Compute gradients
7. Weighted vote into spatial & orientation cells
8. Contrast normalize over overlapping spatial blocks
9. 9-dim feature vector
II. Histograms of Oriented Gradients

Scan image(s) at all scales and locations

Extract features over windows

Run linear SVM classifier on all locations

Fuse multiple detections in 3-D position & scale space

Object detections with bounding boxes

Compute gradients

Weighted vote into spatial & orientation cells

Contrast normalize over overlapping spatial blocks

Feature vector $f = [\ldots, \ldots, \ldots]$  

L2 normalization in each block:

$$ f = \frac{f}{\sqrt{||f||^2 + \epsilon^2}} $$

Slide credit: Sanja Fidler
III. SVM classifier

Training:
- Train a classifier (e.g., person vs no person)

Detection:
- Use the trained classifier to predict presence/absence of object class in each window in the image

Slide credit: Sanja Fidler
III. SVM classifier - training

Learning phase

- Input: Annotations on training images
- Create fixed-resolution normalised training image data set
- Encode images into feature spaces
- Learn binary classifier

Train classifier → Predict presence/absence of object class in each image window

Positive training examples

Negative training examples

- All image crops are scaled to the same size (for this example (15x8) x (7x8) pixels), where 8 is the width/height of each HOG cell in pixels
- **Cool trick**: take a bigger region than each annotated object to also capture context (works better!)
III. SVM classifier - training

Learning phase

Input: Annotations on training images

Create fixed-resolution normalised training image data set

Encode images into feature spaces

Learn binary classifier

Train classifier

Predict presence/absence of object class in each image window

positive training examples

negative training examples

*** These are just feature visualizations. Each "picture" is really a 15x7x31 feature matrix.

Before training a classifier, we vectorize each of these examples: $f = f(\cdot)$

Slide credit: Sanja Fidler
III. SVM classifier - training

Learning phase

Input: Annotations on training images

Create fixed-resolution normalised training image data set

Encode images into feature spaces

Learn binary classifier

Train classifier. SVM (Support Vector Machines) is typically used.

Slide credit: Sanja Fidler
Computing the score $w^T \cdot x + b$ in every location is the same as performing cross-correlation with template $w$ (and add $b$ to result).

Detection Phase

$$\text{score}(l, p) = w \cdot \phi(l, p)$$
IV. Non-Maxima Suppression (NMS)

- Scan image(s) at all scales and locations
- Extract features over windows
- Run linear SVM classifier on all locations
- Fuse multiple detections in 3-D position & scale space
- Object detections with bounding boxes

Non-maxima suppression (NMS)

\[
\text{overlap} = \frac{\text{area}(\text{box}_1 \cup \text{box}_2)}{\text{area}(\text{box}_1 \cap \text{box}_2)} > 0.5
\]

- Remove all boxes that overlap more than XX (typically 50%) with the chosen box

Slide credit: Sanja Fidler
IV. Non-Maxima Suppression (NMS)

- Scan image(s) at all scales and locations
- Extract features over windows
- Run linear SVM classifier on all locations
- Fuse multiple detections in 3-D position & scale space

Object detections with bounding boxes

Non-maxima suppression (NMS)

- Greedy algorithm.
- At each iteration pick the highest scoring box.
- Remove all boxes that overlap more than XX (typically 50%) with the chosen box

Slide credit: Sanja Fidler
HOG detector: summary

Dalal & Triggs ’05
- Histogram of Oriented Gradients (HOG)
- SVM training
- Sliding window detection

Slide credit: Sanja Fidler, Ross Girshick
Example: How can we deal with this guy?

**Dalal & Triggs ’05**
- Histrogram of Oriented Gradients (HOG)
- SVM training
- Sliding window detection

Slide credit: Sanja Fidler, Ross Girshick
HOG detector: limitations

**Dalal & Triggs ’05**
- Histrogram of Oriented Gradients (HOG)
- SVM training
- Sliding window detection

**We need flexible models!**

**Fischler & Elschlager ’73**
- Pictorial structures
- Weak appearance models
- Non-discriminative training

Slide credit: Sanja Fidler, Ross Girshick
The DPM Detector

P. Felzenszwalb, R. Girshick, D. McAllester, D. Ramanan

*Object Detection with Discriminatively Trained Part Based Models*

T-PAMI, 2010

**Paper:** http://cs.brown.edu/~pff/papers/lsvm-pami.pdf

**Code:** http://www.cs.berkeley.edu/~rbg/latent/

cited by 5,084
Deformable Part Model (DPM): key idea

Port the success of Dalal & Triggs into a part-based model

DPM 2005
D&T 2008
PS 2011

Slide credit: Ross Girshick
DPM: Model representation

- A model has a root filter $F_0$ and $n$ part models $(F_i, v_i, d_i)$
  - $F_i$: $i$-th part filter
  - $v_i$: anchor position of $i$-th part relative to the root
  - $d_i$: deformation parameters for $i$-th part

Slide credit: Mubarak Shah, Ross Girshick
DPM: Object Hypothesis

- In HOG feature pyramid
  - root filter - coarser scale
  - part filters - finer scale

\[ z = (p_0, ..., p_n) \]

- \( p_0 \): location of root
- \( p_1, ..., p_n \): location of parts

Score is sum of filter scores minus deformation costs

Slide credit: Mubarak Shah
DPM: Score of a Hypothesis

Score of a hypothesis \( z \) is

\[
score(z) = \beta \cdot \psi(H, z)
\]

where

\[
\beta = (F_0, \ldots, F_n, d_1, \ldots, d_n, b)
\]

\[
\psi(H, z) = (\phi(H, p_0), \ldots, \phi(H, p_n), -\phi_d(dx_1, dy_1), \ldots, -\phi_d(dx_n, dy_n), 1)
\]
DPM: Score of a Hypothesis

\[
score(p_0, ..., p_n) = \sum_{i=0}^{n} F_i \cdot \phi(H, p_i) - \sum_{i=1}^{n} d_i \cdot \phi_d(dx_i, dy_i) + b
\]

- **Data Term**
- **Spatial Prior**
- **Bias**
- **Filters**
- **Feature of subwindow at location \( p_i \)**
- **Deformation parameters**
- **Displacement of part \( i \) relative to its anchor position**

Initial Value (to be learned): \( d_i = (0, 0, 1, 1) \)

Deformation models

Slide credit: Mubarak Shah
DPM: Detection

• The overall score of a root location is computed according to the best possible placement of the parts
  \[ score(p_0) = \max_{p_1, \ldots, p_n} score(p_0, \ldots, p_n) \]

• High-scoring root locations define detections
• Sliding-window approach
• Efficient computation (O(nk)): dynamic programming + generalized distance transforms
DPM: Detection

- **Distance transform**
  - Response of the $i$-th part filter in the $l$-th level of the feature pyramid:
    \[ R_{i,l}(x, y) = F_i \cdot \phi(H, (x, y, l)) \]
  - Transformed response, given root is at $(x, y)$:
    \[ D_{i,l}(x, y) = \max_{dx, dy} (R_{i,l}(x + dx, y + dy) - d_i \cdot \phi_d(dx, dy)) \]

\[
\phi_d(dx, dy) = (dx, dy, dx^2, dy^2)
\]

\[
d_i = (0, 0, 1, 1)
\]
DPM: Detection

Slide credit: Mubarak Shah, Ross Girshick
DPM: Training

- Positive training examples are labeled with bounding boxes
- No part location is available during training (latent)
- Aim: learn model parameters $\beta = (F_0, \ldots, F_n, d_1, \ldots, d_n, b)$
DPM: Latent Variables

- The positions of the parts are not given in both the training and the testing images.
- The variables that exist but not known in training samples are called latent variables.
- The learning algorithm must be able to find/discover the optimal values for the latent variables, namely the position of the parts.

Slide credit: Mubarak Shah
DPM: Training

- The classifier scores an example $x$ by
  $$f_{\beta}(x) = \max_{z \in Z(x)} \beta \cdot \Phi(x, z)$$
  - $\beta$: the model parameters
  - $z$: latent values
  - $Z(x)$: the possible latent values for example $x$
DPM: Training

• Minimize the objective function

\[ L_D(\beta) = \frac{1}{2} \| \beta \|^2 + C \sum_{i=1}^{n} \max(0, 1 - y_i f_\beta(x_i)) \]

• Labeled training examples \( D = \{(x_1, y_1), \ldots, (x_n, y_n)\} \)
• \( y_i \in \{-1, 1\} \)
DPM: Latent SVM

- A latent SVM is semi-convex
  - $f_\beta(x) = \max_{z \in \mathcal{Z}(x)} \beta \cdot \Phi(x, z)$ is convex in $\beta$
  - For negative examples ($y_i = -1$), the hinge loss is convex
    \[
    \max\left(0, 1 - y_i f_\beta(x_i)\right) = \max(0, 1 + f_\beta(x_i))
    \]
    (the maximum of two convex functions)
  - For positive examples ($y_i = 1$), the hinge loss is not convex
    \[
    \max\left(0, 1 - y_i f_\beta(x_i)\right) = \max(0, 1 - f_\beta(x_i))
    \]
    (the maximum of a convex function and a concave function)
  - If the latent value for positive examples are fixed, the hinge loss is convex

Slide credit: Mubarak Shah
DPM: Latent SVM

- Initialize $\beta$ using standard SVM by assuming the same parts locations for all the positive examples
- Iterative optimization:
  - Relabel positive examples: fix $\beta$, find the best $z$ for each positive example (exactly the same with detection!)
  - Optimize $\beta$: fix $z$, optimize $\beta$ by solving the convex problem
DPM: Mixture model

- A mixture model consists of $m$ components
- Captures extreme intra-class variation
- Split the positive bounding boxes into $m$ groups by aspect ratio
DPM on PASCAL VOC

[Source: http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc20{07,08,09,10,11,12}/results/index.html]

Slide credit: Ross Girshick
Object detection

• Introduction
• Face detection: from Viola-Jones to CNN
• General object detection
  • HOG detector
  • Deformable Part-based Model
  • Region-CNN
  • Fast versions of R-CNN
  • YOLO/SSD
• Future directions
Object detection renaissance (2013-present)

Slide credit: Renjie Liao
Deep object detection

Object Detection

Jump to...
- Leaderboard
- Papers
  - R-CNN
  - MultiBox
  - SPP-Net
    - DeepID-Net
    - NoC
    - Fast R-CNN
  - DeepBox
  - MR-CNN
- Faster R-CNN
- YOLO
  - AttentionNet
  - DenseBox

- SSD
- Inside-Outside Net (ION)
- G-CNN
- HyperNet
- MultiPathNet
- CRAFT
- OHEM
- R-FCN
- MS-CNN
- PVANET
- GBD-Net
- StuffNet
- Feature Pyramid Network (FPN)
- YOLOv2
- DSSD

Slide credit: https://handong1587.github.io/deep_learning/2015/10/09/object-detection.html
**R-CNN: Regions with CNN features**

1. Pre-train CNN for **image classification**

large auxiliary dataset (ImageNet)
Training

1. Pre-train CNN for **image classification**
   - large auxiliary dataset (ImageNet)
   - train CNN

2. Fine-tune CNN on **target dataset** and **task**
   - small target dataset (PASCAL VOC)
   - fine-tune CNN

Slide credit: Ross Girshick
Training

1. Pre-train CNN for **image classification**
   - large auxiliary dataset (ImageNet)

2. Fine-tune CNN on **target dataset**
   - small target dataset (PASCAL VOC)
   (optional)

3. Train linear predictor for **detection**
   - region proposals
   - small target dataset (PASCAL VOC)
   - $\sim$2000 warped windows / image
   - CNN features
   - training labels
   - per class SVM

Slide credit: Ross Girshick
## R-CNN Results

<table>
<thead>
<tr>
<th>Method</th>
<th>VOC2007</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPM v5 (Girshick et al. 2011)</td>
<td>33.7%</td>
</tr>
<tr>
<td>Regionlets (Wang et al. 2013)</td>
<td>41.7%</td>
</tr>
<tr>
<td>R-CNN (AlexNet)</td>
<td>54.2%</td>
</tr>
<tr>
<td>R-CNN (AlexNet) + BB</td>
<td>58.5%</td>
</tr>
<tr>
<td>R-CNN (VGGNet)</td>
<td>62.2%</td>
</tr>
<tr>
<td>R-CNN (VGGNet) + BB</td>
<td>66.0%</td>
</tr>
</tbody>
</table>
### R-CNN Results

<table>
<thead>
<tr>
<th>Method</th>
<th>VOC2007</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPM v5 (Girshick et al. 2011)</td>
<td>33.7%</td>
</tr>
<tr>
<td>Regionlets (Wang et al. 2013)</td>
<td>41.7%</td>
</tr>
<tr>
<td>R-CNN (AlexNet)</td>
<td>54.2%</td>
</tr>
<tr>
<td>R-CNN (AlexNet) + BB</td>
<td>58.5%</td>
</tr>
<tr>
<td>R-CNN (VGGNet)</td>
<td>62.2%</td>
</tr>
<tr>
<td>R-CNN (VGGNet) + BB</td>
<td>66.0%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train</td>
<td>84 hours</td>
</tr>
<tr>
<td>Test</td>
<td>47 s/im</td>
</tr>
</tbody>
</table>
Slow R-CNN

- Apply bounding-box regressors
- Classify regions with SVMs
- Forward each region through ConvNet
- Warped image regions
-Regions of Interest (RoI) from a proposal method (~2k)

Slide credit: Ross Girshick
Object Detection System

Getting Proposals  Feature Extraction  Classifier
Object Detection System

Getting Proposals → Feature Extraction → Classifier
Spatial Pyramid Pooling

Spatial Pyramid Pooling

SPP-net

Slide credit: Ross Girshick
## SPP-net Results

<table>
<thead>
<tr>
<th>Model Type</th>
<th>VOC2007</th>
<th>Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>R-CNN (ZFNet)</td>
<td>59.2%</td>
<td>14.5 s/im</td>
</tr>
<tr>
<td>R-CNN (VGGNet)</td>
<td>66.0%</td>
<td>47.0 s/im</td>
</tr>
<tr>
<td>SPP (ZFNet)</td>
<td>59.2%</td>
<td>0.38 s/im</td>
</tr>
<tr>
<td>SPP (VGGNet)</td>
<td>63.1%</td>
<td>2.3 s/im</td>
</tr>
</tbody>
</table>
Object Detection System

Getting Proposals  Feature Extraction  Classifier

SPP
Object Detection System

Getting Proposals  Feature Extraction  Classifier
Fast R-CNN

Totally end-to-end!

Log loss + smooth L1 loss

Linear + softmax

Linear

Multi-task loss

Trainable

ConvNet

Slide credit: Ross Girshick
## Fast R-CNN Results

<table>
<thead>
<tr>
<th>Model</th>
<th>VOC2007</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPPNet BB</td>
<td>63.1%</td>
</tr>
<tr>
<td>R-CNN BB</td>
<td>66.0%</td>
</tr>
<tr>
<td>Fast RCNN</td>
<td>66.9%</td>
</tr>
<tr>
<td>Fast RCNN (07+12)</td>
<td>70.0%</td>
</tr>
</tbody>
</table>
Object Detection System

Getting Proposals → Feature Extraction → Classifier

Fast R-CNN
Faster R-CNN

Region Proposal Network

- Sliding window style
- Multi-scale predictions on fix-sized window for efficiency (take advantage of the large receptive field of CNN features)
- Same loss as R-CNN (cls+bbox)
Region Proposal Network

Figure 2: Recall vs. IoU overlap ratio on the PASCAL VOC 2007 test set.
Faster R-CNN Results

- Fewer and better proposals not only bring speed-up, but also detection performance boost.

<table>
<thead>
<tr>
<th>method</th>
<th># proposals</th>
<th>data</th>
<th>mAP (%)</th>
<th>time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SS</td>
<td>2k</td>
<td>07</td>
<td>66.9</td>
<td>1830</td>
</tr>
<tr>
<td>SS</td>
<td>2k</td>
<td>07+12</td>
<td>70.0</td>
<td>1830</td>
</tr>
<tr>
<td>RPN+VGG, unshared</td>
<td>300</td>
<td>07</td>
<td>68.5</td>
<td>342</td>
</tr>
<tr>
<td>RPN+VGG, shared</td>
<td>300</td>
<td>07</td>
<td>69.9</td>
<td>196</td>
</tr>
<tr>
<td>RPN+VGG, shared</td>
<td>300</td>
<td>07+12</td>
<td>73.2</td>
<td>196</td>
</tr>
</tbody>
</table>
Object Detection System

Getting Proposals

Feature Extraction

Classifier

Faster R-CNN
Efficient Object Detection System

Getting Proposals
- Faster R-CNN

Feature Extraction
- SPP

Classifier
- Fast R-CNN

Accuracy Improvement:
- 66.0% → 73.2%
- 47 s/im → 0.2 s/im
## Example: Driving

<table>
<thead>
<tr>
<th></th>
<th>Pascal 2007 mAP</th>
<th>Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPM v5</td>
<td>33.7</td>
<td>.07 FPS</td>
</tr>
<tr>
<td>R-CNN</td>
<td>66.0</td>
<td>.05 FPS</td>
</tr>
</tbody>
</table>

\(\frac{1}{3}\) Mile, 1760 feet

Slide credit: Joseph Chet Redmon
Example: Driving

<table>
<thead>
<tr>
<th></th>
<th>Pascal 2007 mAP</th>
<th>Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPM v5</td>
<td>33.7</td>
<td>.07 FPS</td>
</tr>
<tr>
<td>R-CNN</td>
<td>66.0</td>
<td>.05 FPS</td>
</tr>
<tr>
<td>Fast R-CNN</td>
<td>70.0</td>
<td>.5 FPS</td>
</tr>
</tbody>
</table>

Slide credit: Joseph Chet Redmon
Example: Driving

<table>
<thead>
<tr>
<th></th>
<th>Pascal 2007 mAP</th>
<th>Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPM v5</td>
<td>33.7</td>
<td>.07 FPS</td>
</tr>
<tr>
<td>R-CNN</td>
<td>66.0</td>
<td>.05 FPS</td>
</tr>
<tr>
<td>Fast R-CNN</td>
<td>70.0</td>
<td>.5 FPS</td>
</tr>
<tr>
<td>Faster R-CNN</td>
<td>73.2</td>
<td>7 FPS</td>
</tr>
</tbody>
</table>

8 feet

12 feet

Slide credit: Joseph Chet Redmon
Real-time object detectors?

- **VOC2007 test mAP**
  - R-CNN, Girshick 2014: 66% mAP / 0.02 fps
  - Fast R-CNN, Girshick 2015: 70% mAP / 0.4 fps
  - Faster R-CNN, Ren 2015: 73% mAP / 7 fps
  - YOLO, Redmon 2016: 66% mAP / 21 fps
  - SSD300: 77% mAP / 46 fps
  - SSD512: 80% mAP / 19 fps

- **Two-Stage**
  - box proposal + post-classify

- **Single Shot**

Slide credit: Wei Liu
YOLO: You Only Look Once
YOLO: output parameterization

Each cell predicts:

- For each bounding box:
  - 4 coordinates \((x, y, w, h)\)
  - 1 confidence value
- Some number of class probabilities

For Pascal VOC:

- 7x7 grid
- 2 bounding boxes / cell
- 20 classes

\[7 \times 7 \times (2 \times 5 + 20) = 7 \times 7 \times 30 \text{ tensor} = 1470 \text{ outputs}\]
YOLO: limitations

- Small objects
- Objects with different shapes/sizes
- Occluded objects
SSD: Single Shot MultiBox Detector

SSD: Single Shot MultiBox Detector

SSD: YOLO + default box shape + multi-scale

(a) Image with GT boxes  
(b) 8 × 8 feature map  
(c) 4 × 4 feature map

SSD: YOLO + default box shape + multi-scale

Object detection

- Introduction
- Face detection: from Viola-Jones to CNN
- General object detection
  - HOG detector
  - Deformable Part-based Model
  - Region-CNN
  - Fast versions of R-CNN
  - YOLO/SSD
- Future directions
Future directions:
Future directions:

- Real-time 3D object detection
Future directions:

• Real-time 3D object detection
• Adversarial examples for object detection
Future directions:

- Real-time 3D object detection
- Adversarial examples for object detection
- Weakly-supervised instance segmentation
Future directions:

- Real-time 3D object detection
- Adversarial examples for object detection
- Weakly-supervised instance segmentation
- High-level scene graph construction
Object detection

• Introduction
• Face detection: from Viola-Jones to CNN
• General object detection
  • HOG detector
  • Deformable Part-based Model
  • Region-CNN
  • Fast versions of R-CNN
  • YOLO/SSD
• Future directions