Object Recognition II

Linda Shapiro

EE/CSE 576

with CNN slides from Ross Girshick
Outline

• Object detection
  • the task, evaluation, datasets

• Convolutional Neural Networks (CNNs)
  • overview and history

• Region-based Convolutional Networks (R-CNNs)
Image classification

- $K$ classes
- Task: assign correct class label to the whole image

Digit classification (MNIST)  Object recognition (Caltech-101)
Classification vs. Detection

✓ Dog
Problem formulation

\{ airplane, bird, motorbike, person, sofa \}
Evaluating a detector

Test image (previously unseen)
First detection ...

☐ ‘person’ detector predictions
Second detection ...

‘person’ detector predictions
Third detection ...

☐ ‘person’ detector predictions
Compare to ground truth

- ‘person’ detector predictions
- ground truth ‘person’ boxes
Sort by confidence

0.9 ✓
true positive (high overlap)

0.8 X

0.6 ✓

0.5 ✓

0.2 X
false positive (no overlap, low overlap, or duplicate)

0.1 X
Evaluation metric

\[ \text{precision}@t = \frac{\#\text{true positives}@t}{\#\text{true positives}@t + \#\text{false positives}@t} \]

\[ \text{recall}@t = \frac{\#\text{true positives}@t}{\#\text{ground truth objects}} \]
Evaluation metric

Average Precision (AP)
0% is worst
100% is best
mean AP over classes (mAP)
Pedestrians

Histories of Oriented Gradients for Human Detection,
Dalal and Triggs, CVPR 2005

AP ~77%
More sophisticated methods: AP ~90%

(a) average gradient image over training examples
(b) each “pixel” shows max positive SVM weight in the block centered on that pixel
(c) same as (b) for negative SVM weights
(d) test image
(e) its R-HOG descriptor
(f) R-HOG descriptor weighted by positive SVM weights
(g) R-HOG descriptor weighted by negative SVM weights
Overview of HOG Method

1. Compute gradients in the region to be described
2. Put them in bins according to orientation
3. Group the cells into large blocks
4. Normalize each block
5. Train classifiers to decide if these are parts of a human
Details

• Gradients
  
  [-1 0 1] and [-1 0 1]^T were good enough filters.

• Cell Histograms
  
  Each pixel within the cell casts a weighted vote for an orientation-based histogram channel based on the values found in the gradient computation. (9 channels worked)

• Blocks
  
  Group the cells together into larger blocks, either R-HOG blocks (rectangular) or C-HOG blocks (circular).
More Details

• **Block Normalization**

  They tried 4 different kinds of normalization.

  • L1-norm
  • sqrt of L1-norm
  • L2 norm
  • L2-norm followed by clipping

• If you think of the block as a vector $\mathbf{v}$, then the normalized block is $\mathbf{v}/\text{norm}(\mathbf{v})$
Example: Dalal-Triggs pedestrian

1. Extract fixed-sized (64x128 pixel) window at each position and scale
2. Compute HOG (histogram of gradient) features within each window
3. Score the window with a linear SVM classifier
4. Perform non-maxima suppression to remove overlapping detections with lower scores
Outperforms

-1 0 1
centered

-1 1
uncentered

1 -8 0 8 -1
cubic-corrected

0 1
diagonal

-1 0 1
-2 0 2
-1 0 1
Sobel

Slides by Pete Barnum

Navneet Dalal and Bill Triggs, Histograms of Oriented Gradients for Human Detection, CVPR05
- Histogram of gradient orientations

Orientation: 9 bins (for unsigned angles)

Histograms in 8x8 pixel cells

- Votes weighted by magnitude
- Bilinear interpolation between cells
Normalize with respect to surrounding cells

$$L_2 - norm : v \rightarrow \frac{v}{\sqrt{\|v\|^2_2 + \epsilon^2}}$$
Input image → Normalize gamma & colour → Compute gradients → Weighted vote into spatial & orientation cells → Contrast normalize over overlapping spatial blocks → Collect HOG’s over detection window → Linear SVM → Person / non-person classification

# features = 15 × 7 × 9 × 4 = 3780

# cells

# orientations

# normalizations by neighboring cells

X =
Training set
0.16 = w^T x - b

\text{sign}(0.16) = 1

\implies \text{pedestrian}
Detection examples
Deformable Parts Model

• Takes the idea a little further
• Instead of one rigid HOG model, we have multiple HOG models in a spatial arrangement
• One root part to find first and multiple other parts in a tree structure.
The Idea

Articulated parts model

- Object is configuration of parts
- Each part is detectable
Deformable objects

Images from Caltech-256
Deformable objects

Images from D. Ramanan’s dataset
How to model spatial relations?

• Tree-shaped model
Model Overview

Model has a root filter plus deformable parts
Hybrid template/parts model

Detections

Template Visualization

root filters
coarse resolution

part filters
finer resolution

deformation models

Felzenszwalb et al. 2008
Pictorial Structures Model

\[ P(L|I, \theta) \propto \left( \prod_{i=1}^{n} p(I|l_i, u_i) \prod_{(v_i, v_j) \in E} p(l_i, l_j|c_{ij}) \right) \]

Appearance likelihood

Geometry likelihood
Results for person matching
Results for person matching
2012 State-of-the-art Detector: Deformable Parts Model (DPM)

1. Strong low-level features based on HOG
2. Efficient matching algorithms for deformable part-based models (pictorial structures)
3. Discriminative learning with latent variables (latent SVM)

Why did gradient-based models work?
Generic categories

Can we detect people, chairs, horses, cars, dogs, buses, bottles, sheep ...?
PASCAL Visual Object Categories (VOC) dataset
Generic categories

Why doesn’t this work (as well)?

Can we detect people, chairs, horses, cars, dogs, buses, bottles, sheep ...?
PASCAL Visual Object Categories (VOC) dataset
Quiz time
(Back to Girshick)
Warm up

This is an average image of which object class?
Warm up

pedestrian
A little harder
A little harder

?  

Hint: airplane, bicycle, bus, car, cat, chair, cow, dog, dining table
A little harder

bicycle (PASCAL)
A little harder, yet
A little harder, yet

Hint: white blob on a green background
A little harder, yet

sheep (PASCAL)
Impossible?
Impossible?

dog (PASCAL)
Impossible?

Why does the mean look like this?
There’s no alignment between the examples!
How do we combat this?

dog (PASCAL)
PASCAL VOC detection history

year

mean Average Precision (mAP)

DPM

DPM, HOG+

DPM, MKL

DPM++, MKL

Selective Search

Selective Search, DPM++, MKL


17% 23% 28% 37% 41% 41%
Part-based models & multiple features (MKL)

mean Average Precision (mAP) vs year

- **DPM**
- **DPM, HOG+ BOW**
- **DPM, MKL**
- **DPM++, MKL, Selective Search**

Year progression:
- 2006: 17%
- 2007: 23%
- 2008: 28%
- 2009: 37%
- 2010: 41%
- 2011: 41%

Rapid performance improvements
Kitchen-sink approaches

Increasing complexity & plateau

- DPM
- DPM++, MKL
- Selective Search
- DPM++, MKL
Region-based Convolutional Networks (R-CNNs)

![Graph showing the improvement in mean Average Precision (mAP) over years. The graph plots different years on the x-axis and mean Average Precision (%) on the y-axis. Each year has a percentage value indicating the improvement in performance.]

[R-CNN. Girshick et al. CVPR 2014]
Region-based Convolutional Networks (R-CNNs)

mean Average Precision (mAP)

year

~1 year

~5 years

[R-CNN. Girshick et al. CVPR 2014]
Convolutional Neural Networks

• Overview
Standard Neural Networks

\[ x = (x_1, \ldots, x_{784})^T \]

\[ z_j = g(w_j^T x) \]

\[ g(t) = \frac{1}{1 + e^{-t}} \]
From NNs to Convolutional NNs

- Local connectivity
- Shared (“tied”) weights
- Multiple feature maps
- Pooling
Convolutional NNs

• Local connectivity

• Each green unit is only connected to (3) neighboring blue units
Convolutional NNs

• Shared ("tied") weights

• All green units share the same parameters \( w \)

• Each green unit computes the same function, but with a different input window
Convolutional NNs

- Convolution with 1-D filter: \([w_3, w_2, w_1]\)

- All green units share the same parameters \(w\)

- Each green unit computes the same function, but with a different input window
Convolutional NNs

- Convolution with 1-D filter: \([w_3, w_2, w_1]\)

- All green units share the same parameters \(w\)

- Each green unit computes the same function, but with a different input window
Convolutional NNs

• Convolution with 1-D filter: \([w_3, w_2, w_1]\]

- All green units **share** the same parameters \(w\)
- Each green unit computes the **same function**, but with a **different input window**
Convolutional NNs

• Convolution with 1-D filter: \([w_3, w_2, w_1]\)

• All green units *share* the same parameters \(w\)

• Each green unit computes the *same function*, but with a *different input window*
Convolutional NNs

- Convolution with 1-D filter: \([w_3, w_2, w_1]\)

- All green units share the same parameters \(w\)

- Each green unit computes the same function, but with a different input window
Convolutional NNs

• Multiple feature maps

• All orange units compute the same function but with a different input windows

• Orange and green units compute different functions

Feature map 1 (array of green units)

Feature map 2 (array of orange units)
Convolutional NNs

• Pooling (max, average)

- Pooling area: 2 units
- Pooling stride: 2 units
- Subsamples feature maps
2D input
Backpropagation applied to handwritten zip code recognition, Lecun et al., 1989
Historical perspective – 1980

Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position

Kunihiko Fukushima
NHK Broadcasting Science Research Laboratories, Kinuta, Setagaya, Tokyo, Japan
Historical perspective – 1980

Hubel and Wiesel
1962

Included basic ingredients of ConvNets, but no supervised learning algorithm
Supervised learning – 1986

Gradient descent training with error backpropagation

Learning Internal Representations by Error Propagation

D. E. RUMELHART, G. E. HINTON, and R. J. WILLIAMS

Early demonstration that error backpropagation can be used for supervised training of neural nets (including ConvNets)
Supervised learning – 1986

“T” vs. “C” problem

Simple ConvNet
Practical ConvNets

Gradient-Based Learning Applied to Document Recognition,
Lecun et al., 1998
Demo

- [http://cs.stanford.edu/people/karpathy/convnetjs/demo/mnist.html](http://cs.stanford.edu/people/karpathy/convnetjs/demo/mnist.html)
- ConvNetJS by Andrej Karpathy (Ph.D. student at Stanford)

Software libraries

- Caffe (C++, python, matlab)
- Torch7 (C++, lua)
- Theano (python)
The fall of ConvNets

• The rise of Support Vector Machines (SVMs)
• Mathematical advantages (theory, convex optimization)
• Competitive performance on tasks such as digit classification
• Neural nets became unpopular in the mid 1990s
The key to SVMs

- It’s all about the features

HOG features SVM weights
(+) (-)

Histograms of Oriented Gradients for Human Detection,
Dalal and Triggs, CVPR 2005
Core idea of “deep learning”

• Input: the “raw” signal (image, waveform, ...)

• Features: hierarchy of features is *learned* from the raw input
• If SVMs killed neural nets, how did they come back (in computer vision)?
What’s new since the 1980s?

• More layers
  • LeNet-3 and LeNet-5 had 3 and 5 learnable layers
  • Current models have 8 – 20+
• “ReLU” non-linearities (Rectified Linear Unit)
  • \( g(x) = \max(0, x) \)
  • Gradient doesn’t vanish
• “Dropout” regularization
• Fast GPU implementations
• More data
What else? Object Proposals

• Sliding window based object detection
  
• Object proposals
  • Fast execution
  • High recall with low # of candidate boxes

Iterate over window size, aspect ratio, and location
The number of contours wholly enclosed by a bounding box is indicative of the likelihood of the box containing an object.
Ross’s Own System: Region CNNs

**R-CNN: Regions with CNN features**

1. Input image
2. Extract region proposals (~2k)
3. Compute CNN features
4. Classify regions

- aeroplane? no.
- person? yes.
- tvmonitor? no.
Competitive Results

<table>
<thead>
<tr>
<th>VOC 2010 test</th>
<th>aero</th>
<th>bike</th>
<th>bird</th>
<th>boat</th>
<th>bottle</th>
<th>bus</th>
<th>car</th>
<th>cat</th>
<th>chair</th>
<th>cow</th>
<th>table</th>
<th>dog</th>
<th>horse</th>
<th>mbike</th>
<th>person</th>
<th>plant</th>
<th>sheep</th>
<th>sofa</th>
<th>train</th>
<th>tv</th>
<th>mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPM v5 [20]†</td>
<td>49.2</td>
<td>53.8</td>
<td>13.1</td>
<td>15.3</td>
<td>35.5</td>
<td>53.4</td>
<td>49.7</td>
<td>27.0</td>
<td>17.2</td>
<td>28.8</td>
<td>14.7</td>
<td>17.8</td>
<td>46.4</td>
<td>51.2</td>
<td>47.7</td>
<td>10.8</td>
<td>34.2</td>
<td>20.7</td>
<td>43.8</td>
<td>38.3</td>
<td>33.4</td>
</tr>
<tr>
<td>UVA [39]</td>
<td>56.2</td>
<td>42.4</td>
<td>15.3</td>
<td>12.6</td>
<td>21.8</td>
<td>49.3</td>
<td>36.8</td>
<td>46.1</td>
<td>12.9</td>
<td>32.1</td>
<td>30.0</td>
<td>36.5</td>
<td>43.5</td>
<td>52.9</td>
<td>32.9</td>
<td>15.3</td>
<td>41.1</td>
<td>31.8</td>
<td>47.0</td>
<td>44.8</td>
<td>35.1</td>
</tr>
<tr>
<td>Regionlets [41]</td>
<td>65.0</td>
<td>48.9</td>
<td>25.9</td>
<td>24.6</td>
<td>24.5</td>
<td>56.1</td>
<td>54.5</td>
<td>51.2</td>
<td>17.0</td>
<td>28.9</td>
<td>30.2</td>
<td>35.8</td>
<td>40.2</td>
<td>55.7</td>
<td>43.5</td>
<td>143</td>
<td>43.9</td>
<td>32.6</td>
<td>54.0</td>
<td>45.9</td>
<td>39.7</td>
</tr>
<tr>
<td>SegDPM [18]†</td>
<td>61.4</td>
<td>53.4</td>
<td>25.6</td>
<td>25.2</td>
<td>35.5</td>
<td>51.7</td>
<td>50.6</td>
<td>50.8</td>
<td>19.3</td>
<td>33.8</td>
<td>26.8</td>
<td>40.4</td>
<td>48.3</td>
<td>54.4</td>
<td>47.1</td>
<td>14.8</td>
<td>38.7</td>
<td>35.0</td>
<td>52.8</td>
<td>43.1</td>
<td>40.4</td>
</tr>
<tr>
<td>R-CNN</td>
<td>67.1</td>
<td>64.1</td>
<td>46.7</td>
<td>32.0</td>
<td>30.5</td>
<td>56.4</td>
<td>57.2</td>
<td>65.9</td>
<td>27.0</td>
<td>47.3</td>
<td>40.9</td>
<td>66.6</td>
<td>37.8</td>
<td>65.9</td>
<td>53.6</td>
<td>26.7</td>
<td>56.5</td>
<td>38.1</td>
<td>52.8</td>
<td>30.2</td>
<td>50.2</td>
</tr>
<tr>
<td>R-CNN BB</td>
<td>71.8</td>
<td>65.8</td>
<td>53.0</td>
<td>36.8</td>
<td>35.9</td>
<td>59.7</td>
<td>60.0</td>
<td>69.9</td>
<td>27.9</td>
<td>50.6</td>
<td>41.4</td>
<td>70.0</td>
<td>62.0</td>
<td>69.0</td>
<td>58.1</td>
<td>29.5</td>
<td>59.4</td>
<td>39.3</td>
<td>61.2</td>
<td>52.4</td>
<td>53.7</td>
</tr>
</tbody>
</table>

Table 1: Detection average precision (%) on VOC 2010 test. R-CNN is most directly comparable to UVA and Regionlets since all methods use selective search region proposals. Bounding-box regression (BB) is described in Section C. At publication time, SegDPM was the top-performer on the PASCAL VOC leaderboard. †DPM and SegDPM use context rescoring not used by the other methods.

Figure 3: (Left) Mean average precision on the ILSVRC2013 detection test set. Methods preceeded by * use outside training data (images and labels from the ILSVRC classification dataset in all cases). (Right) Box plots for the 200 average precision values per method. A box plot for the post-competition OverFeat result is not shown because per-class APs are not yet available (per-class APs for R-CNN are in Table 8 and also included in the tech report source uploaded to arXiv.org; see R-CNN–ILSVRC2013–APs.txt). The red line marks the median AP, the box bottom and top are the 25th and 75th percentiles. The whiskers extend to the min and max AP of each method. Each AP is plotted as a green dot over the whiskers (best viewed digitally with zoom).
Top Regions for Six Object Classes

Figure 4: Top regions for six pool$_2$ units. Receptive fields and activation values are drawn in white. Some units are aligned to concepts, such as people (row 1) or text (4). Other units capture texture and material properties, such as dot arrays (2) and specular reflections (6).
Finale

• Object recognition has moved rapidly in the last 12 years to becoming very appearance based.

• The HOG descriptor lead to fast recognition of specific views of generic objects, starting with pedestrians and using SVMs.

• Deformable parts models extended that to allow more objects with articulated limbs, but still specific views.

• CNNs have become the method of choice; they learn from huge amounts of data and can learn multiple views of each object class.