Object recognition (part 2)

CSE P 576
Larry Zitnick (larryz@microsoft.com)

Convolutional Nets

Yann Le Cun
The Courant Institute of Mathematical Sciences
New York University
http://yann.lecun.com

Good Internal Representations are Hierarchical

- Feature Transform
- Feature Transform
- Classifier

- Low-level features - mid-level features - high-level features - categories
- Representations are increasingly abstract, global, and invariant.

- In Vision: part-whole hierarchy
  - Pixels -> Edges -> Textons -> Parts -> Objects -> Scenes
- In Language: hierarchy in syntax and semantics
  - Words -> Parts of Speech -> Sentences -> Text
  - Objects, Actions, Attributes... -> Phrases -> Statements -> Stories

An Old Idea for Image Representation with Distortion Invariance

- [Hubel & Wiesel 1962]:
  - simple cells detect local features
  - complex cells "pool" the outputs of simple cells within a retinotopic neighborhood.

- Multiple convolutions
- Retinotopic Feature Maps
**Convolutional Net Architecture**

- Convolutional net for handwriting recognition (400,000 synapses)
- Convolutional layers (simple cells): all units in a feature plane share the same weights
- Pooling/subsampling layers (complex cells): for invariance to small distortions.
- Supervised gradient-descent learning using back-propagation
- The entire network is trained end-to-end. All the layers are trained simultaneously.

**Learned Features on natural patches: VI-like receptive fields**

**MNIST Handwritten Digit Dataset**

- Handwritten Digit Dataset MNIST: 60,000 training samples, 10,000 test samples

**Some Results on MNIST (from raw images: no preprocessing)**

<table>
<thead>
<tr>
<th>CLASSIFIER</th>
<th>DEFORMATION</th>
<th>ERROR</th>
<th>REFERENCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Knowledge-free methods (a fixed permutation of the pixels would make no difference)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2-layer NN, 800-H2, CE</td>
<td>Affine</td>
<td>1.50</td>
<td>Simard et al., CDAR 2003</td>
</tr>
<tr>
<td>3-layer NN, 500×500-H1, CE, reg</td>
<td>Affine</td>
<td>1.53</td>
<td>Hinton, in press, 2005</td>
</tr>
<tr>
<td>SVM, Gaussian kernel</td>
<td></td>
<td>1.40</td>
<td>Cortes 92 = Many others</td>
</tr>
<tr>
<td>Convolutional nets</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Convolutional net (Laffin-5)</td>
<td>Affine</td>
<td>0.80</td>
<td>Narasato et al., NIPS 2006</td>
</tr>
<tr>
<td>Convolutional net (Laffin-6)</td>
<td>Affine</td>
<td>0.70</td>
<td>Narasato et al., NIPS 2006</td>
</tr>
<tr>
<td>Training set augmented with Affine Distortions</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2-layer NN, 800-H1, CE</td>
<td>Affine</td>
<td>1.10</td>
<td>Simard et al., CDAR 2003</td>
</tr>
<tr>
<td>Virtual SVM deg-3 poly</td>
<td>Affine</td>
<td>0.80</td>
<td>Scholkopf</td>
</tr>
<tr>
<td>Training set augmented with Elastic Distortions</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2-layer NN, 800-H1, CE</td>
<td>Elastic</td>
<td>0.70</td>
<td>Simard et al., CDAR 2003</td>
</tr>
<tr>
<td>Convolutional net, CE</td>
<td>Elastic</td>
<td>0.40</td>
<td>Simard et al., CDAR 2003</td>
</tr>
</tbody>
</table>

Note: some groups have found good results using various amounts of preprocessing such as deskewing (e.g. 0.50% using an SVM with a constant kernel [deCoste and Scholkopf]), hand-designed feature representations (e.g. 0.63% with “shape context” and nearest neighbor [Belongie].
Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories

To appear in CVPR 2006

Svetlana Lazebnik (slazebni@uiuc.edu)
Beckman Institute, University of Illinois at Urbana-Champaign

Cordelia Schmid (cordelia.schmid@inrialpes.fr)
INRIA Rhône-Alpes, France

Jean Ponce (ponce@di.ens.fr)
École Normale Supérieure, France

http://www-cvr.ai.uiuc.edu/ponce_grp

Overview

- A “pre-attentive” approach: recognize the scene as a whole without examining its constituent objects
- Inspiration: locally orderless images
  - Kaempffert & Van Duzer (1993)
- Previous work: “subdivide-and-disorder” strategy

Spatial pyramid representation

- Extension of a bag of features
- Locally orderless representation at several levels of resolution
- Based on pyramid match kernels
  - Grauman & Darrell (2005)
    - Our approach: build pyramid in image space, discard spatial information
    - Our approach: build pyramid in image space, quantize feature space

Feature extraction

- Weak features
  - Edge points at 2 scales and 8 orientations (vocabulary size 16)
- Strong features
  - SIFT descriptors of 16x16 patches sampled on a regular grid, quantized to form visual vocabulary (size 200, 400)
Support Vector Machines

Modified from the slides by Dr. Andrew W. Moore
http://www.cs.cmu.edu/~awm/tutorials

Copyright © 2001, 2003, Andrew W. Moore
Nov 23rd, 2001

Linear Classifiers

\[ f(x, w, b) = \text{sign}(w \cdot x - b) \]

How would you classify this data?

Copyright © 2001, 2003, Andrew W. Moore
Support Vector Machines: Slide 14

Copyright © 2001, 2003, Andrew W. Moore
Support Vector Machines: Slide 15

Copyright © 2001, 2003, Andrew W. Moore
Support Vector Machines: Slide 16
Linear Classifiers

\[ f(x, w, b) = \text{sign}(w \cdot x - b) \]

- denotes +1
- denotes -1

How would you classify this data?

Classifier Margin

Define the margin of a linear classifier as the width that the boundary could be increased by before hitting a datapoint.

Maximum Margin

The maximum margin linear classifier is the linear classifier with the, um, maximum margin. This is the simplest kind of SVM (Called an LSVM).

Any of these would be fine...

...but which is best?
Support Vector Machines: Slide 21

Maximum Margin

- denotes +1
- denotes -1

$f(x, w, b) = \text{sign}(w \cdot x - b)$

The maximum margin linear classifier is the linear classifier with the, um, maximum margin. This is the simplest kind of SVM (Called an LSVM)

Support Vectors are those datapoints that the margin pushes up against

Why Maximum Margin?

1. Intuitively this feels safest.
2. If we've made a small error in the location of the boundary (it's been jolted in its perpendicular direction) this gives us least chance of causing a misclassification.
3. LOOCV is easy since the model is immune to removal of any non-support-vector datapoints.
4. There's some theory (using VC dimension) that is related to (but not the same as) the proposition that this is a good thing.
5. Empirically it works very very well.

Support Vector Machines: Slide 22

Nonlinear Kernel (I)

Example: SVM with Polynomial of Degree 2

Kernel: $k(x, x') = [x \cdot x' + 1]^d$

Support Vector Machines: Slide 23

Nonlinear Kernel (II)

Example: SVM with RBF-Kernel

Kernel: $k(x, x') = \exp(-\|x - x'\|^2)$

Support Vector Machines: Slide 24
Scene category dataset
Fei-Fei & Perona (2005), Oliva & Torralba (2001)
http://www-vibe.stanford.edu/papers/imr/data

Multi-class classification results (100 training images per class)

<table>
<thead>
<tr>
<th>Level</th>
<th>Weak features (vocabulary size: 16)</th>
<th>Strong features (vocabulary size: 200)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Single-level</td>
<td>Pyramid</td>
</tr>
<tr>
<td>0 (1 x 1)</td>
<td>45.3 ±0.5</td>
<td>72.2 ±0.6</td>
</tr>
<tr>
<td>1 (2 x 2)</td>
<td>53.6 ±0.3</td>
<td>77.9 ±0.6</td>
</tr>
<tr>
<td>2 (4 x 4)</td>
<td>61.7 ±0.6</td>
<td>79.4 ±0.3</td>
</tr>
<tr>
<td>3 (8 x 8)</td>
<td>63.3 ±0.8</td>
<td>77.2 ±0.4</td>
</tr>
</tbody>
</table>

Fei-Fei & Perona: 69.2%

Caltech101 dataset
Fei-Fei et al. (2005)

Multi-class classification results (30 training images per class)

<table>
<thead>
<tr>
<th>Level</th>
<th>Weak features (16)</th>
<th>Strong features (200)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Single-level</td>
<td>Pyramid</td>
</tr>
<tr>
<td>0</td>
<td>15.5 ±0.9</td>
<td>41.2 ±1.2</td>
</tr>
<tr>
<td>1</td>
<td>31.4 ±1.2</td>
<td>32.8 ±1.3</td>
</tr>
<tr>
<td>2</td>
<td>47.2 ±1.1</td>
<td>49.3 ±1.4</td>
</tr>
<tr>
<td>3</td>
<td>52.2 ±0.8</td>
<td>54.0 ±1.1</td>
</tr>
</tbody>
</table>

Caltech-101: Drawbacks

- Smallest category size is 31 images: \( N_{\text{min}} \leq 30 \)
- Too easy?
  - left-right aligned
  - Rotation artifacts
- Soon will saturate performance
• Jump to Nicolas Pinto’s slides. (page 29)
Papers


Object Detection

**Probabilistic Framework**

\[
P(O | v) = \frac{P(v | O)}{P(v)} P(O)
\]

(Single Object Likelihood)

Given all image features (local/object and scene/context)

\[v = v_{\text{Local}} + v_{\text{Contextual}}\]
Contextual Priming for Object Detection: Probabilistic Framework

\[
P(O \mid v) = \frac{P(O, v)}{P(v)} = \frac{P(v_L \mid O, v_C) \cdot P(O \mid v_C)}{P(v)}
\]

Local measurements (a lot in the literature)

Contextual features
Contextual Priming for Object Detection: Contextual Features

Use PCA on filter output images to reduce the number of features (< 64)

Gabor filters at 4 scales and 6 orientations

Use Mixture of Gaussians to model the probabilities. (Other alternatives include KNN, parzen window, logistic regression, etc)

Contextual Priming for Object Detection: Object Priming Results

\( o_1 = \text{people}, o_2 = \text{furniture}, o_3 = \text{vehicles} \) and \( o_4 = \text{trees} \)

Contextual Priming for Object Detection: Focus of Attention Results

Heads

Contextual Priming for Object Detection: Conclusions

• Proves the relation btw low level features and scene/context
• Can be seen as a computational evidence for the (possible) existence of low-level feature based biological attention mechanisms
• Also a warning: Whether an object recognition system understands the object or works by lots bg features.
Do classification (find label probabilities) in each segment only with local info.

Most consistent labeling according to object co-occurrences & local label probabilities.
Objects in Context:
Local Categorization

- Extract random patches on zero-padded segments
- Calculate SIFT descriptors
- Use BoF:
  - Cluster patches in training (Hier. K-means, K=10x3)
  - Histogram of words in each segment
  - NN classifier (returns a sorted list of categories)
- Each segment is classified independently

Objects in Context:
Contextual Refinement

Contextual model based on co-occurrences
Try to find the most consistent labeling with high posterior probability and high mean pairwise interaction.
Use CRF for this purpose.

\[ p(c_1 \ldots c_k | S_1 \ldots S_k) = \frac{B(c_1 \ldots c_k) \prod_{i=1}^{k} A(i)}{Z(\phi, S_1 \ldots S_k)} \]

\( R(c_1 \ldots c_k) = \exp \left( \sum_{i<k} \phi(i,j) \right) \)
Mean interaction of all label pairs
\( \phi(i,j) \) is basically the observed label co-occurrences in training set.

Objects in Context:
Learning Context

Using labeled image datasets (MSRC, PASCAL)

Using labeled text based data (Google Sets): Contains list of related items
\( \rightarrow \) A large set turns out to be useless! (anything is related)

Objects in Context:
Results

<table>
<thead>
<tr>
<th></th>
<th>No Context</th>
<th>Google Sets</th>
<th>Using Training</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSRC</td>
<td>45.0%</td>
<td>58.1%</td>
<td>68.4%</td>
</tr>
<tr>
<td>PASCAL</td>
<td>61.8%</td>
<td>63.4%</td>
<td>74.2%</td>
</tr>
</tbody>
</table>

Table 1. Average Categorization Accuracy.
"Objects in Context" – Limitations: Context modeling

Segmentation → Categorization without context (Local information only)

With co-occurrence context

Means: \( P(\text{person,dog}) > P(\text{person,cow}) \)

(Bonus Q: How did it handle the background?)

But why? Isn’t it only a dataset bias? We have seen in the previous example that \( P(\text{person,dog}) \) is common too.

"Objects in Context" – Limitations: Context modeling

Segmentation → Categorization without context (Local information only)

With co-occurrence context

\( P(\text{person,horse}) > P(\text{person,dog}) \)

"Objects in Context" – Limitations: Context modeling

Object-Object or Stuff-Object ?

Stuff

Stuff

Stuff

Looks like "background" stuff - object (such as water-boat) does help rather than "foreground" object co-occurrences (such as person-horse)

But still car-person-motorbike is useful in PASCAL

Labels with high co-occurrences with other labels

"Objects in Context" – Limitations: Segmentation

• Too good: A few or many? How to select a good segmentation in multiple segmentations?
• Can make object recognition & contextual reasoning (due to stuff detection) much easier.
“Objects in Context” - Limitations

• No cue by unknown objects
• No spatial relationship reasoning
• Object detection part heavily depends on good segmentations
• Improvements using object co-occurrences are demonstrated with images where many labels are already correct. → How good is the model?

Contextual Priming vs. Objects in Context

Scene→Object
Simpler training data
(Scene information is view-dependent)

[Object,Stuff] ↔ [Object,Stuff]
May need huge amount of labeled data

Can be more generic than scene→object with a very good model

Contextual model is object detector independent, in theory. But:
+ uses segmentation easier to detect stuff
- uses segmentation can be unreliable

Object recognition

We’ve come a long way...

Fischler and Elschlager, 1973
Dollar et al., BMVC 2009
Still a ways to go…

Part-based person detector
4 main components:

- Feature selection
- Part detection
- Spatial model
- NMS / context
How can we help?

- Humans supply training data...
  - 100,000s labeled images
- We design the algorithms.
  - Going on 40 years.
- Can we use humans to debug?

Human debugging

Human performance

- Humans ~90% average precision
- Machines ~46% average precision
Part detections

Humans

Machine

Humans

Machine

Humans

Machine

Humans

Machine

Humans

Machine

Part detections

Humans

Machine

Part detections

Humans

Machine

Part detections

Humans

Machine

Part detections

Humans

Machine
AP results

Spatial model

Feature selection → Part detection → NMS / context

Person

Not a person

Spatial model

Context/NMS

vs.

PASCAL

machine parts
human parts

HWW MWW HD

machine spatial model human spatial model

CH GH NH CL GL NL MP

machine nms human nms + context

PASCAL

0.8

0.6

0.4

0.2

0
Conclusion

http://www.ted.com/talks/lang/en/pawan_sinha_on_how_brains_learn_to_see.html 7:00min

Pawan Sinha on how brains learn to see