Object Detection

Hao Xu

Advisor: Prof. C.-C. Jay Kuo
Table of Content

• My Method
  – Motivation
  – Related Work
    • DVW&DVP
    • Spatial Encoding
  – Framework
  – Post Processing
  – Dataset
  – Experiment Result

• CNN
  – Overview
  – Best Architecture
  – Several Remarks
  – Caffe
  – R-CNN
    • Hypotheses Generation
  – Performance

• Comparison
  – Structure
  – Feature
Motivation

Bounding boxes in training images

Each point is a visual words, together they build visual phrase

landmark

component

Each point is a visual words, together they build visual phrase
Descriptive Visual Words and Phrases

SIFT Quantization -> VW represents Image -> Occurrence

Example

Group into VP -> Select Good VW & VP

Shiliang Zhang, Qi Tian, Gang Hua, Qingming Huang, Wen Gao, Generating Descriptive Visual Words and Visual Phrases for Large-Scale Image Applications
Descriptive Visual Words and Phrases

Algorithm 1: VisualWordRank

Input: \( R^{(C)} \); maximum iteration time: \( maxiter \).

Output: The rank value of each DVW candidate to the category \( C \):

\[
\text{Rank}_i^{(C)} = 1, \ldots, VW\text{num}^{(C)}
\]

Initialize each element in the \( VW\text{num}^{(C)} \times 1 \) sized rank vector: \( Old\text{Rank}^{(C)} \) as 1; Normalize the sum of each column of \( R^{(C)} \) as 1 [2]; Set \( iter = 0 \)

While \( iter < maxiter \)

\[
\text{NewRank}^{(C)} = R^{(C)} \cdot \text{OldRank}^{(C)}
\]

If \( |\text{NewRank}^{(C)} - \text{OldRank}^{(C)}| \leq \varepsilon \) break

\[
\text{OldRank}^{(C)} = \text{NewRank}^{(C)}\]

\( iter++ \)

End

\[
\text{Rank}^{(C)} = \text{NewRank}^{(C)}
\]
Descriptive Visual Words and Phrases

- Performance Evaluation

Figure 3. Occurrence Frequency of Visual Words

Figure 4. Demonstration of Regular VW and dVW and the Matching Performance
Spatial Information – Log Polar Code

- Using Spatial Information to Verify SIFT Matches

Example:

<table>
<thead>
<tr>
<th></th>
<th>$\mathbf{V}_1$</th>
<th>$\mathbf{V}_2$</th>
<th>$\mathbf{V}_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathbf{V}_1$</td>
<td>0</td>
<td>10011010</td>
<td>00010111</td>
</tr>
<tr>
<td>$\mathbf{V}_2$</td>
<td>11011110</td>
<td>0</td>
<td>11011110</td>
</tr>
<tr>
<td>$\mathbf{V}_3$</td>
<td>00010011</td>
<td>10011010</td>
<td>0</td>
</tr>
</tbody>
</table>
When you have outliers you may face much frustration if you include them in a model fitting operation. But if your model’s fit to a sample set of minimal size, the probability of the set being outlier-free will rise. Brute force tests of all sets will cause computational constipation.

$N$ random samples will provide an example of a fitted model uninfluenced by outliers. No need to test all combinations!

Each random trial should have its own unique sample set and make sure that the sets you choose are not degenerate. $N$, the number of sets, to choose is based on the probability of a point being an outlier, and of finding a set that’s outlier free. Updating $N$ as you go will minimize the time spent.

So if you gamble that $N$ samples are ample to fit a model to your set of points, it’s likely that you will win the bet.

Select the set that boasts that its number of inliers is the most (you’re almost there). Fit a new model just to those inliers and discard the rest, an estimated model for your data is now possessed! This marks the end point of your model fitting quest.

Framework - Training

- Bottom-up view
Framework - Detecting
Why Post Processing

• How VOC Evaluates a Bounding Box?

\[ a_o = \frac{\text{area}(B_p \cap B_{gt})}{\text{area}(B_p \cup B_{gt})} > 50\% \]
Post Processing

- Compute the Detection Window’s Saliency Score
  \[
  MS(w, \theta_{MS}) = \sum_{p \in w} I_{MS}(p) \times \frac{|\{p \in w | I_{MS}(p) \geq \theta_{MS}\}|}{|w|}
  \]

- Compute the Color Contrast in/outside Detection Window
  \[
  \frac{|\text{Surf}(w, \theta_{CC})|}{|w|} - \theta_{CC} - 1
  \]
  \[
  CC(w, \theta_{CC}) = \chi^2(h(w), h(\text{Surf}(w, \theta_{CC})))
  \]

- Compute Edge Density around Detection Window
  \[
  \frac{|\text{Lim}(w, \theta_{ED})|}{|w|} - \frac{1}{\theta_{ED}}
  \]
  \[
  ED(w, \theta_{ED}) = \frac{\sum_{p \in \text{Lim}(w, \theta_{ED})} I_{ED}(p)}{\text{Lim}(w, \theta_{ED})}
  \]

- Compute Super Pixel Straddling
  \[
  SS(w, \theta_{SS}) = 1 - \sum_{s \in S(w)} \frac{\min(|w \setminus s|, |s \setminus w|)}{|w|}
  \]
Increase/Decrease each side of the bounding box by 16%, 8%, 4%, 2%.
Evaluate 4 scores, and Select the best one
Improve mAP by 0.8-1.2
How To Evaluate Performance

- True Positive Rate: ratio between detected true positive and actual positive
- False Positive Rate: ratio between detected false positive and actual negative
- ROC: allowing certain false positive rate, what is the true positive rate
- mAP: averaging precision at a list of predefined recall level

<table>
<thead>
<tr>
<th></th>
<th>actual positive</th>
<th>actual negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>predicted positive</td>
<td>TP</td>
<td>FN</td>
</tr>
<tr>
<td>predicted negative</td>
<td>FP</td>
<td>TN</td>
</tr>
</tbody>
</table>

Recall = \( \frac{TP}{TP+FN} \)

Precision = \( \frac{TP}{TP+FP} \)

True Positive Rate = \( \frac{TP}{TP+FN} \)

False Positive Rate = \( \frac{FP}{FP+TN} \)

mAP = \( \frac{1}{11} \sum_{r \in [0.0, 1.0]} \text{precision}(r) \)
### Pascal vs Image-Net

<table>
<thead>
<tr>
<th>PASCAL</th>
<th>ILSVRC</th>
</tr>
</thead>
<tbody>
<tr>
<td>birds</td>
<td>flamingo, cock, ruffed grouse, quail, partridge</td>
</tr>
<tr>
<td></td>
<td>pill bottle, beer bottle, wine bottle, water bottle, pop bottle</td>
</tr>
<tr>
<td>bottles</td>
<td>bottle</td>
</tr>
<tr>
<td>cars</td>
<td>car</td>
</tr>
<tr>
<td></td>
<td>race car, wagon, minivan, jeep, cab</td>
</tr>
</tbody>
</table>
Pascal vs Image-Net
Performance

Figure 5. Performance comparison w.r.t DPM using VOC 2012 data

Figure 6. Performance comparison w.r.t DPM using high resolution data
Summary

• Visual Words + Spatial Information = Visual Phrase
• Match Visual Words and Fit Spatial Information
• Performance
  – Roughly on par with DPM for rigid object
  – Worse than DPM for non-rigid object
  – Better than DPM for high resolution dataset
• Comment and Future Work
  – Train better (define more spatial template for non-rigid object)
  – Fix low resolution performance (Hopefully)
Convolutional Neural Network (CNN)

- A Glimpse on the Black Box
Convolutional Neural Network (CNN)

- Most Advanced CNN for Visual Applications:

  - **Input**
    - 227x227x3 image
    - 96 11x11x3 kernel
    - 4 pixels stride

  - 55x55x48 input
    - 256 5x5x48 kernel
    - 1 pixels stride
    - 384 3x3x25 kernel
    - 1 pixels stride
    - 384 3x3x1 kernel
    - 1 pixels stride

  - 13x13x192 input
    - 256 3x3x192 kernel
    - 1 pixels stride
    - 384 3x3x192 kernel
    - 1 pixels stride

  - 13x13x192 input
    - 256 3x3x192 kernel
    - 1 pixels stride
    - 192 13x13x192 kernel
    - 1 pixels stride

  - Output

---

Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, ImageNet Classification with Deep Convolutional Neural Networks
Convolutional Neural Network (CNN)

- A few remarks on the architecture:
  - Non-linear Function:
  - Multiple GPU Communication
  - Overlapping Pooling

Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, ImageNet Classification with Deep Convolutional Neural Networks
Convolutional Neural Network (CNN)

- Conquer Overfitting:
  - Data Augmentation:
    - Image Translation and Reflection:
      - From 256x256, select 5 227x227 patches and their reflection.
    - Altering RGB Channels:
      - Adding Engineered Noise to Image.
  - Dropout:
    - At training time, random select half of the architecture
    - At testing time, use entire architecture but give each unit half weight.
Convolutional Neural Network (CNN)

- Performance on ILSVRC 2010 Classification Benchmark:

<table>
<thead>
<tr>
<th>Model</th>
<th>Top-1</th>
<th>Top-5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sparse coding [2]</td>
<td>47.1%</td>
<td>28.2%</td>
</tr>
<tr>
<td>SIFT + FVs [24]</td>
<td>45.7%</td>
<td>25.7%</td>
</tr>
<tr>
<td>CNN</td>
<td>37.5%</td>
<td>17.0%</td>
</tr>
</tbody>
</table>

Top-\(n\) Error Rate: The correct Result is not among the First \(n\) labels.
Convolutional Neural Network (CNN)

- Caffe Architecture
  - Caffe was created by Yangqing Jia, and is in active development by the Berkeley Vision and Learning Center.
  - Caffe fits industry needs, with blazing fast C++/CUDA code for GPU computation.
  - Caffe is currently the fastest GPU CNN implementation publicly available, and is able to process more than 40 million images per day with a single NVIDIA K40 or Titan GPU.
  - Caffe provides seamless switching between CPU and GPU.
  - Public available on Github.

This is a BORING slide, I know!

But Check this:
http://demo.caffe.berkeleyvision.org/
Region Proposal with CNN

- R-CNN Flowchart:

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

Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik, Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation
Region Proposal with CNN

- Region Proposal:

  Constraint Parametric Min-Cut
  Selective Search

J. Carreira and C. Sminchisescu. CPMC Automatic object segmentation using constrained parametric min-cut.

J.R.R. Uijlings, K.E.A. Van De Sande, T. Gevers, and A.W.M. Smeulders, Selective Search for Object Recognition
Region Proposal with CNN

- CNN:

- Dilate the region propose
- Warp image into 227x227
Region Proposal with CNN

- Performance:

<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 [18]^1</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 [34]</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 [37]</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>14.3</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 [16]^1</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>57.8</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>50.2</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>
Region Proposal with CNN

- Performance:

DPM and Our performance is based on VOC 2012 benchmark.
R-CNN is based on VOC 2010 benchmark.
Comparison

• Structure

Bounding boxes in training image

landmark

Each point is a visual words, together they build visual phrase

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

warped region

acoaplane? no.

person? yes.

tvmonitor? no.
Comparison

- Feature

Object Component Spatial Configuration

SIFT Spatial Configuration

Pool5 feature: (3,3,4) (top 1 - 24)

Pool5 feature: (3,3,6) (top 1 - 24)

Pool5 feature: (3,3,8) (top 1 - 24)
Future Plan

• Develop MAN structure
• Try Caffe implementation
• Adopt the features trained in CNN in my research
Thank You