Segmentation and structured deep learning

July 4th, 2014

Christian Wolf
Université de Lyon, INSA-Lyon
LIRIS UMR CNRS 5205

Mingyuan Jiu
Doct., soutenu le 3.4.2014

Natalia Neverova
Doct. 2ième année

Graham W. Taylor,
Université de Guelph, Canada
Segmentation for visual recognition

Applications:
- Pose estimation (body, hand)
- Semantic full scene labelling

Hard complexity constraints (real time!)

PhD of Mingyuan Jiu
PhD of Natalia Neverova
PhD of Prisca Bonnet
(Deep) representation learning
Segmentation and spatial relationships

**Non**
Pixelwise classification (independant)

**Oui**
MRF/CRF/BN. Inference of a global solution with high computational complexity

**Oui**
Auto-context models

**Oui**
Pixelwise classification. The prior improves the classifier

---

Features

- $F_i$
- $l_i$

Labels

- $F_1$
- $l_1$
- $F_2$
- $l_2$
- $F_3$
- $l_3$
- $F_4$
- $l_4$

---

**References**

- [ICPR 2002a]
  - 5e/43 à DIBCO 2009
- [IEEE-Tr-PAMI 2010]
- [Neurocomputing 2010]
- [ICPR 2010]
- [EG-W-3DOR 2008]
- [ICPR 2002b]
- [ICPR 2008]
- [Travaux en cours T. Kekec]
- [Travaux en cours R. Khan]
- [Pattern recognition letters 2014]
- [DIBCO 2009]
“Spatial learning”

Application:
- Calculate human pose: set of joint positions
- Use an intermediate representation: body part segmentation

Figure: Shotton et al., CVPR 2011
Jiu, Wolf, Baskurt, 2013

PhD of Mingyuan Jiu
Spatial relationships: labels

Additional information: neighboring pixels are likely
- to have similar labels, or
- to have labels which are adjacent in the object layout (!!)

Could also be solved by MRF + discrete optimization

\[ E(l_1, \ldots, l_N) = \sum_i U(l_i, Z_i) + \alpha \sum_{(i,j) \in \mathcal{E}} D(l_i, l_j) \]
Structured models ... w/o structure

- It is **not** possible to include pairwise terms into a classifier which classifies pixels independently.
- Pairwise terms lead to combinatorial problems.
- Alternative strategy:
  - do **not** proceed by pairs
  - change the loss function for pixelwise classification
  - punish errors (classically), but:
    - punish errors **less**, if the misclassified label is a **neighbor** of the groundtruth label
- It will be shown that this strategy decreases “pure” **classical** (!!!) classification error.
M images \{X^1, \ldots, X^M\}

- A parametric function maps pixels \(i\) (and their receptive fields) to a feature representation
  \[Z^m_i \in \mathbb{R}^Q\]
  \[Z^m_i = f(X^m_i | \theta_f)\]

- A classifier predicts part labels
  \[\hat{l}_i = g(Z^m_i | \theta_g)\]
Classical supervised learning

Stimulated network output:

\[ \hat{l}_i = g(Z^m_i | \theta_g) \]

Target output (groundtruth):

\[ \bar{l}_1 \quad \bar{l}_2 \quad \bar{l}_3 \]

Classical loss function: cross entropy

\[ E(w) = - \sum_n \left\{ \bar{l}_n \ln \hat{l}_n + (1 - \bar{l}_n) \ln (1 - \hat{l}_n) \right\} \]
Learning to rank class labels

- The groundtruth class label is supposed to be ranked first (highest classifier response)
- The neighboring class labels are supposed to ranked next
- The non-neighboring class labels are ranked last
- The rankings inside the groups (gt, nb, non-nb) are irrelevant
Learning to rank class labels

Similar to (Burges, NIPS 2006), the loss function is decomposed into terms over pairs. For each pair, differences in network output are mapped to probabilities:

\[ o_{uv} = g(Z_{i,u}) - g(Z_{i,v}) \]
\[ P_{uv} = \frac{e^{o_{ij}}}{1 + e^{o_{ij}}} \]

A target probability is defined according to desired ranking: \( \tilde{P}_{uv} \) is set to \( \lambda > 0.5 \) if \( u \) is ranked higher than \( v \), and \( 1 - \lambda \) otherwise.

Output and target probability are compared with cross-entropy loss:

\[ C_{uv} = -\tilde{P}_{uv} \log P_{uv} - (1 - \tilde{P}_{uv}) \log (1 - P_{uv}) \]
Results

Figure 4: Classification examples from the CDC4CV dataset. (a) input depth image; (b) groundtruth segmentation; (c) appropriate baseline: randomized forest for CDC4CV; (d) DrLIM+LR without spatial learning; (e) our method (spatial pre-training and spatial LR learning).

Input | Groundtruth | Random forest (Shotton et al., CVPR 2011) | ConvNet w/ DrLIM pretraining (Hadsell/Chopra/Lecun, CVPR 2006) + classical backprop | ConvNet w/ spatial pretraining + spatial backprop (Our method)

CDC4CV Poselets dataset (Holt et al., 2011)
Experimental results: accuracy

<table>
<thead>
<tr>
<th>Methods</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Randomized forest (Shotton et al., 2011)</td>
<td>60.30%</td>
</tr>
<tr>
<td><strong>Spatial Randomized forest (Jiu et al., 2013)</strong></td>
<td><strong>61.05%</strong></td>
</tr>
<tr>
<td>Single-scale (vanilla) ConvNet (LeCun et al., 1998)</td>
<td>47.17%</td>
</tr>
<tr>
<td>Multi-scale ConvNet (Farabet et al., 2012)</td>
<td>62.54%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Convolutional layers</th>
<th>LR</th>
<th>Fine-tuning</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>DrLIM (Hadsell et al., 2006)</td>
<td>classical</td>
<td>no</td>
<td>35.10%</td>
</tr>
<tr>
<td>DrLIM (Hadsell et al., 2006)</td>
<td>spatial</td>
<td>no</td>
<td>41.05%</td>
</tr>
<tr>
<td></td>
<td>spatial</td>
<td>classical</td>
<td>38.60%</td>
</tr>
<tr>
<td></td>
<td>spatial</td>
<td>spatial</td>
<td><strong>41.65%</strong></td>
</tr>
<tr>
<td>DrLIM (Hadsell et al., 2006)</td>
<td>classical</td>
<td>yes</td>
<td>64.39%</td>
</tr>
<tr>
<td>DrLIM (Hadsell et al., 2006)</td>
<td>spatial</td>
<td>yes</td>
<td>65.12%</td>
</tr>
<tr>
<td></td>
<td>spatial</td>
<td>classical</td>
<td>65.18%</td>
</tr>
<tr>
<td></td>
<td>spatial</td>
<td>yes</td>
<td><strong>66.92%</strong></td>
</tr>
</tbody>
</table>

CDC4CV Poselets dataset (Holt et al., 2011)
Hand part segmentation

- Structured Deep learning
- Real time necessary
- Training set: 600,000 frames
  - labelled synthetic data
  - Unlabelled real data

PhD of Natalia Neverova
Structural information

- A single region is supposed to exist for each label
- Unconnected outlier pixels are identified and punished
- No regularization during testing: pixelwise classification
Learning context

** INPUT DEPTH MAP 

\[ X(i,j) \]

** DIRECT LEARNER 

\[ f_d(\cdot; \theta_d) \]

\[ Y_d(i,j) \]

** CONTEXT LEARNER 

\[ f_c(\cdot; \theta_c) \]

\[ Y_c(i,j) \]

** SEGMENTATION MAP 1 

\[ n_{fd}(i,j) \]

** SEGMENTATION MAP 2 

\[ n_{G}(i,j) \]

** GROUND TRUTH MAP 

\[ G(i,j) \]

** GROUND TRUTH MAP 

\[ G(i,j) \]
Results

On 50 manually annotated frames (real data)

<table>
<thead>
<tr>
<th>Loss function</th>
<th>Training data</th>
<th>Test data</th>
<th>Accuracy</th>
<th>Average per class</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Q_{sd}$ (supervised baseline)</td>
<td>synth.</td>
<td>synth. real</td>
<td>85.90%</td>
<td>78.50%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>47.15%</td>
<td>34.98%</td>
</tr>
<tr>
<td>$Q_{sd} + Q_{loc} + Q_{glb}$ (semi-supervised, ours)</td>
<td>all</td>
<td>synth. real</td>
<td>85.49%</td>
<td>78.31%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>50.50%</td>
<td>43.25%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Terms</th>
<th>$Q_{loc}$</th>
<th>$Q_{glb}^+$</th>
<th>$Q_{glb}^+ + Q_{glb}^-$</th>
<th>$Q_{loc} + Q_{glb}^+ + Q_{glb}^-$</th>
<th>$Q_{sd}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requires labels</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>Gain in % points</td>
<td>+0.60</td>
<td>+0.36</td>
<td>+0.41</td>
<td>+0.82</td>
<td>+16.05</td>
</tr>
</tbody>
</table>
Results on real images: one step of unsupervised training
Results on real images
Many applications need highly efficient (real time) segmentation algorithms

Traditional graphical models are unsuited

Including structural terms into training (as opposed to testing) can help