DEEP CONVOLUTIONAL NETWORK CASCADE FOR FACIAL POINT DETECTION

Yi Sun, Xiaogang Wang, Xiaoou Tang

By 吴昊东
AUTHOR

- Xiaoou Tang

- Xiaogang Wang
PUBLICATIONS OF XIAOGANG WANG


Deep Learning in Vision

Joint Deep Learning for Pedestrian Detection
W. Ouyang and X. Wang

A Cascaded Deep Learning Architecture for Pedestrian Detection
X. Zeng, W. Ouyang and X. Wang

Hybrid Deep Learning for Computing Face Similarities
Y. Sun, X. Wang, and X. Tang

Deep Learning Identity Preserving Face Space
Z. Zhu, P. Luo, X. Wang, and X. Tang

A Deep Sum-Product Architecture for Robust Facial Attributes Analysis
P. Luo, X. Wang, and X. Tang

Pedestrian Parsing via Deep Decompositional Neural Network
P. Luo, X. Wang, and X. Tang

Deep Convolutional Network Cascade for Facial Point Detection
Y. Sun, X. Wang, and X. Tang

Hierarchical Face Parsing via Deep Learning
P. Luo, X. Wang, and X. Tang

Modeling Mutual Visibility Relationship with a Deep Model in Pedestrian Detection
W. Ouyang, X. Zeng, and X. Wang

A Discriminative Deep Model for Pedestrian Detection with Occlusion Handling
W. Ouyang and X. Wang
Facial Point Detection

- Eyes (Right, Left)
- Nose
- Mouth (Right side, Left side)
RELATED WORK

- Classifying search windows
  - Scan based on local region
  - Adaboost, SVM, random forest

- Directly predicting keypoint positions
  - No scanning, More efficient
  - Regressors
CONVOLUTIONAL NEURAL NETWORKS

- Convolutional Layers
  - Feature maps
  - Weight sharing
- Subsampling Layers
  - Pooling- Max Pooling
- Full connection Layers
  - Hidden layer + Logistic Regression
**Weight Sharing**

- Receptive field
  - Biological inspiration
- Locally connected
  - Decrease number of weight

![Image with matrix and convolved feature]

<table>
<thead>
<tr>
<th>1x1</th>
<th>1x0</th>
<th>1x1</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Image**

**Convolved Feature**
POOLING

- Subsampling
- Dimensionality reduction
- Max pooling or Average pooling or else...

Convolved feature

Pooled feature
CASCADE CONVOLUTIONAL NETWORKS

- 3 levels
  - Roughly prediction + fine tune
- Along the cascade
  - Input size: full -> local
  - Network structure: deep -> shallow
  - Task: high level but rough - > low level but fine
**ALL NEURAL NETWORKS**

- **Level 1:** F1 NE1 NM1

- **Level 2:** LE21, LE22, RE21, RE22, N21, N22, LM21, LM22, RM21, RM22;

- **Level 3:** LE31, LE32, RE31, RE32, N31, N32, LM31, LM32, RM31, RM32;
STRUCTURE SELECTION

- Absolute value rectification
  - In convolutional layer after nonlinear activation function
  - Effectively improve the performance (empirical but proved by experiment)

- Locally sharing weights of neurons on the same map
  - Traditional: Sharing weights of all neurons on one map
  - In Face, eyes and mouth may share low-level features but they are different at high-level
  - Improve the performance
MULTI-LEVEL REGRESSION

- The only prior knowledge for level 1
  - The face bounding box
  - Large to cover large pose variations and the instability of face detectors
  - But cause inaccuracy!
    - irrelevant areas included
MULTI-LEVEL REGRESSION

- Level 1 provide strong prior for the following levels
  - Facial point lies within small region around the prediction at level 1
  - Without context information
    - Should not cascade too many level and trust following levels too much
    - Adjust the initial prediction in a very small range

\[
x = \frac{x_1^{(1)} + \cdots + x_{l_1}^{(1)}}{l_1} + \sum_{i=2}^{n} \frac{\Delta x_1^{(i)} + \cdots + \Delta x_{l_i}^{(i)}}{l_i}
\]  

for an n-level cascade with \( l_i \) predictions at level \( i \). Note that predictions at the first level are absolute positions while predictions at the following levels are adjustments.
**Implementation Details**

- The input layer is denoted by $I(h,w)$
- Convolutional layer is denoted by $C(s, n, p, q)$ or $CR(s, n, p, q)$

\[
y_{i,j}^{(t)} = \tanh \left( \sum_{r=0}^{m-1} \sum_{k=0}^{s-1} \sum_{l=0}^{s-1} x_{i+k,j+l}^{(r)} \cdot w_{k,l}^{(r,u,v,t)} + b^{(u,v,t)} \right),
\]

for $i = \Delta h \cdot u, \ldots, \Delta h \cdot u + \Delta h - 1$, $j = \Delta w \cdot v, \ldots, \Delta w \cdot v + \Delta w - 1$, $t = 0, \ldots, n-1$. $\Delta h = \frac{h-s+1}{p}$, $\Delta w = \frac{w-s+1}{q}$, $u = 0, \ldots, p - 1$, and $v = 0, \ldots, q - 1$. $x$ and $y$ are the outputs of the previous and current layers. $w$ is weight and $b$ is bias. The $m$ maps in the previous layer are correlated with $m$ $s$ by $s$ kernels.
IMPLEMENTATION DETAILS

- Pooling layer is denoted by $P(s)$
  - $s$ is the side length of square pooling regions

\[
y_{i,j}^{(t)} = \tanh \left( g^{(u,v,t)} \cdot \max_{0 \leq k,l < s} \left\{ x_{i \cdot s + k, j \cdot s + l}^{(t)} \right\} + b^{(u,v,t)} \right)
\]

- Fully connected layer is denoted by $F(n)$
  - $n$ is the number of neurals

\[
y_j = \tanh \left( \sum_{i=0}^{m-1} x_i \cdot w_{i,j} + b_j \right), \text{ for } j = 0, \ldots, n - 1
\]
STRUCTURE

- Input Range

<table>
<thead>
<tr>
<th>net</th>
<th>left</th>
<th>right</th>
<th>top</th>
<th>bottom</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1</td>
<td>F1</td>
<td>-0.05</td>
<td>+1.05</td>
<td>-0.05</td>
</tr>
<tr>
<td></td>
<td>EN1</td>
<td>-0.05</td>
<td>+1.05</td>
<td>-0.04</td>
</tr>
<tr>
<td></td>
<td>NM1</td>
<td>-0.05</td>
<td>+1.05</td>
<td>+0.18</td>
</tr>
<tr>
<td>L2</td>
<td>*21</td>
<td>-0.16</td>
<td>+0.16</td>
<td>-0.16</td>
</tr>
<tr>
<td></td>
<td>*22</td>
<td>-0.18</td>
<td>+0.18</td>
<td>-0.18</td>
</tr>
<tr>
<td>L3</td>
<td>*31</td>
<td>-0.11</td>
<td>+0.11</td>
<td>-0.11</td>
</tr>
<tr>
<td></td>
<td>*32</td>
<td>-0.12</td>
<td>+0.12</td>
<td>-0.12</td>
</tr>
</tbody>
</table>

Table 2: Summary of network input ranges, which are described by left, right, top, and bottom boundary positions. For networks at level 1 (L1), the four boundary positions are relative to the normalized face bounding box with boundary positions (0, 1, 0, 1). For networks at level 2 (L2) and level 3 (L3), the four boundary positions are relative to the predicted facial point position.
EXPERIMENT

- Average detection error
  \[ err = \frac{\sqrt{(x - x')^2 + (y - y')^2}}{l} \]

- The failure rate
  - Err > 5% -> failed
**EXPERIMENT**

- **Investigate Structure**
  - Only use result of F1

<table>
<thead>
<tr>
<th></th>
<th>layer 0</th>
<th>layer 1</th>
<th>layer 2</th>
<th>layer 3</th>
<th>layer 4</th>
<th>layer 5</th>
<th>layer 6</th>
<th>layer 7</th>
<th>layer 8</th>
<th>layer 9</th>
</tr>
</thead>
<tbody>
<tr>
<td>S0</td>
<td>I(39,39)</td>
<td>CR(4,20,2,2)</td>
<td>P(2)</td>
<td>CR(3,40,2,2)</td>
<td>P(2)</td>
<td>CR(3,60,3,3)</td>
<td>P(2)</td>
<td>CR(2,80,2,2)</td>
<td>F(120)</td>
<td>F(10)</td>
</tr>
<tr>
<td>S1</td>
<td>I(31,39)</td>
<td>CR(4,20,1,1)</td>
<td>P(2)</td>
<td>CR(3,40,2,2)</td>
<td>P(2)</td>
<td>CR(3,60,2,3)</td>
<td>P(2)</td>
<td>CR(2,80,1,2)</td>
<td>F(100)</td>
<td>F(6)</td>
</tr>
<tr>
<td>S2</td>
<td>I(15,15)</td>
<td>CR(4,20,1,1)</td>
<td>P(2)</td>
<td>CR(3,40,1,1)</td>
<td>P(2)</td>
<td>F(60)</td>
<td>F(2)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S3</td>
<td>I(39,39)</td>
<td>CR(4,20,2,2)</td>
<td>P(2)</td>
<td>CR(3,40,2,2)</td>
<td>P(2)</td>
<td>CR(3,60,3,3)</td>
<td>P(2)</td>
<td>F(120)</td>
<td>F(10)</td>
<td></td>
</tr>
<tr>
<td>S4</td>
<td>I(39,39)</td>
<td>CR(4,20,2,2)</td>
<td>P(2)</td>
<td>CR(3,40,2,2)</td>
<td>P(2)</td>
<td>F(120)</td>
<td>F(10)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S5</td>
<td>I(39,39)</td>
<td>CR(4,20,2,2)</td>
<td>P(2)</td>
<td>F(120)</td>
<td>F(10)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S6</td>
<td>I(39,39)</td>
<td>C(4,20,2,2)</td>
<td>P(2)</td>
<td>C(3,40,2,2)</td>
<td>P(2)</td>
<td>C(3,60,3,3)</td>
<td>P(2)</td>
<td>C(2,80,2,2)</td>
<td>F(120)</td>
<td>F(10)</td>
</tr>
<tr>
<td>S7</td>
<td>I(39,39)</td>
<td>CR(4,20,1,1)</td>
<td>P(2)</td>
<td>CR(3,40,1,1)</td>
<td>P(2)</td>
<td>CR(3,60,1,1)</td>
<td>P(2)</td>
<td>CR(2,80,1,1)</td>
<td>F(120)</td>
<td>F(10)</td>
</tr>
</tbody>
</table>

Table 1: Summary of network structures. F1 adopts S0. Both EN1 and NM1 adopt S1. All the networks at the second and third levels share S2. To investigate different designs of network structures, we also compare different structures S3-S7 for F1 in experiments.
**EXPERIMENT**

- Investigate Structure
  - Only use result of F1

**Figure 3:** The structure of deep convolutional network F1. Sizes of input, convolution, and max pooling layers are illustrated by cuboids whose length, width, and height denote the number of maps, and the size of each map. Local receptive fields of neurons in different layers are illustrated by small squares in the cuboids.
EXPERIMENT

- Investigate Structure
  - Only use result of F1

Figure 4: Average detection errors and failure rates of convolutional network F1 with different structures.
EXPERIMENT

- Investigate Multi-Level
EXPERIMENT

- Compare with other methods

- Databases
  - BioID has 1, 521 images of 23 subjects
  - LFPW contains 1, 432 face images from the web
EXPERIMENT

- Compare with other methods

**Average errors on BioID**

**Failure rates on BioID**

**Average errors on LFPW**

**Failure rates on LFPW**

- Liang et al. [20]
- Valstar et al. [26]
- Luxand [1]
- Microsoft [2]
- Our Method
EXPERIMENT

- Compare with other methods

Accuracy improvements on BioID

Accuracy improvements on LFPW

Legend:
- Liang et al. [20]
- Valstar et al. [26]
- Luxand [1]
- Microsoft [2]
**EXPERIMENT**

- Compare with other methods

Figure 7: Compare with Belhumeur *et al.* [4] and Cao *et al.* [5] on LFPW test images.
CONCLUSION

- Cascade CNN
  - Deep cnn + shallow cnn
  - Roughly estimation + fine tune

- High performance
  - Beat state-of-the-art methods and latest commercial software