Deep Feature Learning for Contour Detection

Vision And Learning SEminar (VALSE)

Wei Shen (沈为)
http://wei-shen.weebly.com/
Shanghai University
Outline

- Contour Detection
  - Overview
  - Milestones
- Our Work
- Discussions
Introduction of Contour Detection

- Reduce dimensionality of data
- Preserve content information
- Useful in applications such as

Object Detection  Image Segmentation  Sketch-based Image Search
Introduction of Contour Detection

- **Edge Detection**
  - the characteristic changes in brightness, color, and texture

- **Contour Detection**
  - The changes in pixel ownership from one object or surface to another
To differentiate contour and non-contour is difficult, even for human beings!
Challenge

- Confusion between contour and non-contour changes caused by cluttered textures

changes correspond to object boundaries

changes caused by cluttered textures
Key Problems

- Contour detection is usually formulated as a per-pixel classification problem
  - How to extract discriminative contour features?
  - How to learn an efficient contour classifier?
Outline

- Contour Detection
  - Overview
  - Milestones
- Our Work
- Discussions
Edge Detection

- Use local differential template to compute gradient

\[
\begin{align*}
X &= F^* \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \\
Y &= F^* \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}
\end{align*}
\]

\[
G = \sqrt{(X^2 + Y^2)}
\]

The original

Sobel edge
Related Works

Learning Based Contour Detection

- Learn a classifier to combine different features

![Image](image_url)

**Image**

<table>
<thead>
<tr>
<th>Boundary Cues</th>
<th>Cue Combination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brightness</td>
<td>Model</td>
</tr>
<tr>
<td>Color</td>
<td></td>
</tr>
<tr>
<td>Texture</td>
<td></td>
</tr>
</tbody>
</table>

P_b
Discriminative features

- Spectral component (gPb - Arbeláez et al. PAMI 11)
- Sparse code gradients (SCG – Ren&Bo, NIPS 12)
Learning Based Contour Detection

- Efficient Detector
  - Structured Forest (SE – Dollar & Zitnick ICCV 13, PAMI 15)
Related Works

Use deep networks to extract contour features and predict contour map.
Related Works

Outline

- Contour Detection
  - Overview
  - Milestones
- Our Work
- Discussions
DeepContour: A Deep Convolutional Feature Learned by Positive-sharing Loss for Contour Detection

Wei Shen, Xinggang Wang, Yan Wang, Xiang Bai, Zhijiang Zhang

IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2015
Why to apply CNN for contour detection?
- Contour is hard to define
- Contour data are sufficient for CNN training (millions of local contour patches)
Given a color image patch $x \in \mathbb{R}^{n \times n \times 3}$, our goal is to determine whether its center pixel is passed through by contours or not.

$$x \in \mathbb{R}^{n \times n \times 3} \rightarrow z \in \{0, 1\}$$

Q: Is a good idea to directly use CNN as a blackbox to address this binary classification problem?
The large variations in the contour shapes

Solution: Partitioning contour patches into compact clusters to convert the binary classification problem to a multi-class classification problem

How to define the loss function?

Q: Is softmax a good choice?

- Softmax function penalizes the loss of each class equally
- The losses for contour versus non-contour should be emphasized

Solution: Adding a regularized term to focus on the end goal of binary classification
Pre-cluster contour patches according to their contour shapes.

Assign a label $y$ to each contour patch $x$ according to the pre-cluster index $\{1, \ldots, K\}$.

$$x \in \mathbb{R}^{n \times n \times 3} \rightarrow y \in \{0,1, \ldots, K\}$$

Negative  Positive
Method

- CNN Architecture

Input layer

COV1 layer

COV2 layer

COV3 layer COV4 layer FC1 layer FC2 layer

- Loss Function

Let \((a_j^{(i)}; j = 1, \ldots, K)\) be the output of unit \(j\) in FC2 for a image patch \(x_j^{(i)}\), the probability that the label is \(j\) is

\[
p_j^{(i)} = \frac{\exp(a_l^{(i)})}{\sum_{l=0}^{K} \exp(a_l^{(i)})}
\]
Method

\[ J = - \frac{1}{m} \sum_{i=1}^{m} \left( \sum_{j=0}^{K} 1(y^{(i)} = j) \log p_j^{(i)} \right) \rightarrow \text{Softmax loss} \]

\[ - \frac{1}{m} \left[ \sum_{i=1}^{m} \lambda \left( 1(y^{(i)} = 0) \log p_0^{(i)} + \sum_{j=1}^{K} 1(y^{(i)} = j) \log(1 - p_0^{(i)}) \right) \right] \]

Positive-sharing loss, the loss for positive class is shared among each pre-clustered contour classes
Method

- To apply standard back-propagation to optimize the parameters of the network

\[
\frac{\partial J}{\partial a_0^{(i)}} = \frac{1}{m} \left[ (\lambda + 1) \mathbf{1}(y^{(i)} = 0) (p_0^{(i)} - 1) + (\lambda + 1) \sum_{j=1}^{K} \mathbf{1}(y^{(i)} = j) p_0^{(i)} \right]
\]

\[
\frac{\partial J}{\partial a_l^{(i)}} = \frac{1}{m} \left[ (\lambda \mathbf{1}(y^{(i)} = 0) + 1)p_l^{(i)} - \mathbf{1}(y^{(i)} = l) - \lambda \sum_{j=1}^{K} \mathbf{1}(y^{(i)} = j) \frac{p_0^{(i)} p_l^{(i)}}{1 - p_0^{(i)}} \right]
\]
Method

- CNN model validation

$$\gamma = \frac{1}{m} \sum_{i=1}^{m} \left[ \left( 1(y^{(i)} = 0) - 1(y^{(i)} > 0) \right) \left( p_0^{(i)} - (1 - p_0^{(i)}) \right) \right]$$

$$\gamma \in [-1,1]$$, measuring the discrimination of the learned model between positive and negative samples.

- The learned features of FC1 will be fed into structured forest to perform contour detection.
Experimental results

- Deep Feature Visualization

Positive-sharing loss

Local gradient

Softmax loss
Results on BSDS500

Experimental results
Experimental results

- Results on BSDS500

<table>
<thead>
<tr>
<th>Method</th>
<th>ODS</th>
<th>OIS</th>
<th>AP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human</td>
<td>.80</td>
<td>.80</td>
<td>-</td>
</tr>
<tr>
<td>Canny [6]</td>
<td>.60</td>
<td>.63</td>
<td>.58</td>
</tr>
<tr>
<td>Felz-Hutt [16]</td>
<td>.61</td>
<td>.64</td>
<td>.56</td>
</tr>
<tr>
<td>Normalized Cuts [8]</td>
<td>.64</td>
<td>.68</td>
<td>.45</td>
</tr>
<tr>
<td>Mean Shift [7]</td>
<td>.64</td>
<td>.68</td>
<td>.56</td>
</tr>
<tr>
<td>Gb [28]</td>
<td>.69</td>
<td>.72</td>
<td>.72</td>
</tr>
<tr>
<td>ISCRA [39]</td>
<td>.72</td>
<td>.75</td>
<td>.46</td>
</tr>
<tr>
<td>gPb-owt-ucm [2]</td>
<td>.73</td>
<td>.76</td>
<td>.73</td>
</tr>
<tr>
<td>Sketch Tokens [29]</td>
<td>.73</td>
<td>.75</td>
<td>.78</td>
</tr>
<tr>
<td>DeepNet [23]</td>
<td>.74</td>
<td>.76</td>
<td>.76</td>
</tr>
<tr>
<td>SCG-[38]</td>
<td>.74</td>
<td>.76</td>
<td>.77</td>
</tr>
<tr>
<td>PMI+sPb [21]</td>
<td>.74</td>
<td>.77</td>
<td>.78</td>
</tr>
<tr>
<td>SE [11]</td>
<td>.74</td>
<td>.76</td>
<td>.78</td>
</tr>
<tr>
<td>SE-Var [12]</td>
<td>.75</td>
<td>.77</td>
<td>.80</td>
</tr>
<tr>
<td>N^4-Fields [19]</td>
<td>.75</td>
<td>.77</td>
<td>.78</td>
</tr>
<tr>
<td>DeepContour (ours)</td>
<td>.76</td>
<td>.78</td>
<td>.80</td>
</tr>
</tbody>
</table>
Experimental results

- Results on NYUD
### Experimental results

#### Cross Dataset Generalization

<table>
<thead>
<tr>
<th>DeepContour</th>
<th>ODS</th>
<th>OIS</th>
<th>AP</th>
</tr>
</thead>
<tbody>
<tr>
<td>BSDS/BSDS</td>
<td>.76</td>
<td>.78</td>
<td>.80</td>
</tr>
<tr>
<td>NYU/BSDS</td>
<td>.72</td>
<td>.74</td>
<td>.77</td>
</tr>
<tr>
<td>BSDS/NYU</td>
<td>.59</td>
<td>.60</td>
<td>.53</td>
</tr>
<tr>
<td>NYU/NYU</td>
<td>.62</td>
<td>.63</td>
<td>.57</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>BSDS/BSDS</td>
<td>.74</td>
<td>.76</td>
<td>.78</td>
</tr>
<tr>
<td>NYU/BSDS</td>
<td>.72</td>
<td>.73</td>
<td>.76</td>
</tr>
<tr>
<td>BSDS/NYU</td>
<td>.55</td>
<td>.57</td>
<td>.46</td>
</tr>
<tr>
<td>NYU/NYU</td>
<td>.60</td>
<td>.61</td>
<td>.56</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SCG [39]</th>
<th>ODS</th>
<th>OIS</th>
<th>AP</th>
</tr>
</thead>
<tbody>
<tr>
<td>NYU/NYU</td>
<td>.55</td>
<td>.57</td>
<td>.46</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>gPb [2]</th>
<th>ODS</th>
<th>OIS</th>
<th>AP</th>
</tr>
</thead>
<tbody>
<tr>
<td>NYU/NYU</td>
<td>.51</td>
<td>.52</td>
<td>.37</td>
</tr>
</tbody>
</table>
Experimental results

Parameter Discussion

![Graph showing the relationship between OSD and \( \lambda \) for different values of K (K=1, K=25, K=50, K=100, K=150).](image-url)
Outline

- Contour Detection
  - Overview
  - Milestones
- Our Work
- Discussions
Discussions

- **Speed**
  - Caffe – per-patch mean subtraction is the bottleneck

- **Accuracy**
  - Limitation may caused by the confusing labels
Thank you!

Q&A