Deep Networks with Shape Priors for Nucleus Detection
ICIP 2018 - Athens, Greece

Mohammad Tofighi, Tiantong Guo, Jairam K.P. Vanamala, Vishal Monga
Department of Electrical Engineering
Pennsylvania State University

Wednesday 10\textsuperscript{th} October, 2018
Table of contents

Introduction
   Cell Nuclei Detection
   Neural Networks
   CNN for Cell Nuclei Detection
   Challenges

Shape Priors with CNN
   Network Structure
   Cost Function

Experimental Results
   Dataset
   Performance Comparison

Conclusion
Medical Image Processing Applications

Cell detection

Cell segmentation
Xing et al.
TMI 2016

CT segmentation
Cherukuri et al.
TBME 2016
Cell Nucleus Detection and Challenges

▶ Morphological methods ¹

▶ Challenges:
  ▶ Overlapping cells,
  ▶ Different nucleus shapes

▶ Deep learning based methods are proposed ²

▶ Pros: Learned features can boost the performance

▶ Cons: Fail in challenging cases; naive learning of features

▶ Solution: Learn better!
  Guided by the expert domain knowledge

¹Y. Al-Kofahi et al., IEEE TBME 2010
²A. Cruz et al., MICCAI 2013
Introduction to Neural Networks

- Deep learning models inspired by the biological neural networks.
- They have been used for several applications: ³ ⁴
  - Classification: image segmentation, object detection, speech recognition, ...
  - Regression: Image super-resolution, denoising, ...

³ J. Long et al., CVPR 2015
⁴ Y. LeCun et al., Nature 2015
Introduction to Neural Networks

- One mostly used NN: Convolutional Neural Network (CNN)
- A mapping $Y = f(X, \Theta)$ is learned by minimizing the cost function $E(f(X, \Theta), G)$ between the output $Y$ and the ground truth $G$
- Using a stochastic gradient descent method and an error back-propagation algorithm

---

$^6$ Y. Lacun *at al.*, Neural Computation, 1989

Recent works on cell detection based on CNN/Deep Learning technique: SC-CNN\(^7\), SR-CNN\(^8\) SSAE\(^9\), LIPSyM\(^10\)

---

7 Sirinukunwattana et al., TMI 2016  
8 Xie et al., MICCAI 2015  
9 Xu et al., TMI 2016  
10 Kuse et al., JPI 2011
Challenges in Cell Nuclei Detection

- Overlapping cell: false positive and false negative detections
- Varying shapes of the nuclei: decrease detection and segmentation accuracy

Cell detection

Cell segmentation

Xing et al.
TMI 2016
Building Informative Priors

- Our Solution: Shape Prior Guided CNN
Proposed Method: SP-CNN Structure

Prior Information

Functionality

Raw Edges

$\hat{x}$

Edge Map

$\hat{x}$

Masked

$gp(\hat{y}) \odot \hat{x}$

$priors$

Window Map

$gp(\hat{y})$

Raw Input

$\times$

$D$-layer CNN

{$\Theta$}

Output

$\hat{y}$

Generate Raw Edges

Generate Window Map

Element-wise multiplication

Influencing CNN
Cost Function

- Suppose the shape set as $S = \{S_i \mid i = 1, 2, \ldots, n\}$
- CNN cost function
  \[ \Theta = \underset{\Theta}{\text{arg min}} \| f(x; \Theta) - y \|_2^2 \]  
  \hspace{1cm} (1)
- Cost term of the shape priors
  \[ \sum_{i=1}^{n} \| (g_p(\hat{y}) \odot \hat{x}) * S_i \|_2^2 \]  
  \hspace{1cm} (2)
- Overall, the cost function of the SP-CNN is given as:
  \[ \Theta = \underset{\Theta}{\text{arg min}} \| f(x; \Theta) - y \|_2^2 - \lambda \sum_{i=1}^{n} \| (g_p(\hat{y}) \odot \hat{x}) * S_i \|_2^2 \]  
  \hspace{1cm} (3)
SP-CNN Visual Illustrations

Raw input ($x$)  Raw edges ($\hat{x}$)  Window map $g_p(\hat{y})$

Masked edges ($g_p(\hat{y}) \odot \hat{x}$)  Output ($\hat{y}$)  Groundtruth ($y$)
Train CNN using the input image and the ground truth label

Using the CNN output, put masks on each detected local maxima (done by maxpooling): $g_p(\hat{y})$

Extract raw edge image from the raw input image using simple Canny edge detection filter: $\hat{x}$

Element-wise multiplication: $(g_p(\hat{y}) \odot \hat{x}) \Rightarrow$ masked edge map

Masks out the edges from $\hat{x}$ that are surrounding the detected location in $\hat{y}$: delete non-cell edges

Convolve masked edge map with each of the shapes in set $S$: shape prior information

Add them up and feed it back to CNN
Dateset & Assessment Methods

- **UW Dataset**¹¹: 100 H&E stained histology images of colorectal adenocarcinomas (≈30k cells)

- **PSU Dataset** – EE & Department of Food Science: 120 Colonic Mucosa images (≈26k cells)

- Test-Train split: UW (50 − 50, consistent with¹¹), PSU (20 − 80).
- For assessment Recall (R), Precision (P), and F1 Score are used:
  \[ P = \frac{TP}{TP+FP} , \quad R = \frac{TP}{TP+FN} , \quad \text{and} \quad F_1 = \frac{2PR}{P+R} \]

¹¹ K. Sirinukunwattana et al. – TMI’16
Assessment Methods & Experimental Results

All the results are obtained with same assessment procedure:

Table: Nucleus detection results for dataset of SC-CNN

<table>
<thead>
<tr>
<th>UW Dataset</th>
<th>Precision</th>
<th>Recall</th>
<th>F1 score</th>
</tr>
</thead>
<tbody>
<tr>
<td>SP-CNN</td>
<td>0.803</td>
<td>0.843</td>
<td>0.823</td>
</tr>
<tr>
<td>SC-CNN</td>
<td>0.781</td>
<td>0.823</td>
<td>0.802</td>
</tr>
<tr>
<td>CP-CNN</td>
<td>0.697</td>
<td>0.687</td>
<td>0.692</td>
</tr>
<tr>
<td>SR-CNN</td>
<td>0.783</td>
<td>0.804</td>
<td>0.793</td>
</tr>
<tr>
<td>SSAE</td>
<td>0.617</td>
<td>0.644</td>
<td>0.630</td>
</tr>
<tr>
<td>LIPSyM</td>
<td>0.725</td>
<td>0.517</td>
<td>0.604</td>
</tr>
<tr>
<td>CRImage</td>
<td>0.657</td>
<td>0.461</td>
<td>0.542</td>
</tr>
</tbody>
</table>

Table: Nucleus detection results for PSU Dataset

<table>
<thead>
<tr>
<th>PSU Dataset</th>
<th>Precision</th>
<th>Recall</th>
<th>F1 score</th>
</tr>
</thead>
<tbody>
<tr>
<td>SP-CNN</td>
<td>0.854</td>
<td>0.871</td>
<td>0.863</td>
</tr>
<tr>
<td>SC-CNN</td>
<td>0.821</td>
<td>0.830</td>
<td>0.825</td>
</tr>
<tr>
<td>SR-CNN</td>
<td>0.797</td>
<td>0.805</td>
<td>0.801</td>
</tr>
<tr>
<td>SSAE</td>
<td>0.665</td>
<td>0.634</td>
<td>0.649</td>
</tr>
</tbody>
</table>

---

12 Sirinukunwattana et al., TMI 2016
13 Xie et al., TMI 2016
14 Xu et al., TMI 2016
15 Kuse et al., JPI 2011
Precision-Recall Curve for Choosing the Optimal Threshold - UW Dataset

Figure: SC-CNN \textsuperscript{17}, SR-CNN \textsuperscript{18} SSAE \textsuperscript{19}, LIPSyM \textsuperscript{20}

\textsuperscript{17}Sirinukunwattana \textit{et al.}, TMI 2016
\textsuperscript{18}Xie \textit{et al.}, MICCAI 2015
\textsuperscript{19}Xu \textit{et al.}, TMI 2016
\textsuperscript{20}Kuse \textit{et al.}, JPI 2011
Precision-Recall Curve for Choosing the Optimal Threshold - PSU Dataset

![Precision-Recall Curve Diagram](image)

Figure: SC-CNN \(^{21}\), SR-CNN \(^{22}\) SSAE \(^{23}\)

\(^{21}\) Sirinukunwattana et al., TMI 2016

\(^{22}\) Xie et al., MICCAI 2015

\(^{23}\) Xu et al., TMI 2016
Example Results

(a) Example image (partial image)  
(b) Output of SP-CNN  
(c) Groundtruth  
(d) Detection by SP-CNN; F1-score = 0.843  
(e) Detection by SC-CNN; F1-score = 0.801  
(f) Detection by SR-CNN; F1-score = 0.784
Example Results

Groundtruth

Detection by SP-CNN; F1-score = 0.868

Detection by SC-CNN; F1-score = 0.815

Detection by SR-CNN; F1-score = 0.809
Conclusion

- Shape prior guided convolutional neural networks help improve the performance of cell nuclei detection.
- Future research will be focused on designing data adaptive learning shapes.
Thanks For Your Attention!
Back-propagation Analysis of SP-CNN

- Training a neural network with gradient descent requires the calculation of the gradient of the cost function.
- The cost function of SP-CNN is as follows:

\[
E(x; \Theta) = \|f(x; \Theta) - y\|^2_2 - \lambda \sum_{i=1}^{n} \|(g_p(\hat{y}) \odot \hat{x}) \ast S_i\|^2_2 \quad (4)
\]

- It has two terms: fidelity cost term and the cost term corresponding to the shape priors.
- Detection fidelity cost term is:

\[
L = \|f(x; \Theta) - y\|^2_2, \quad (5)
\]

- The cost term for shape priors is:

\[
P = -\lambda \sum_{i=1}^{n} \|(g_p(\hat{y}) \odot \hat{x}) \ast S_i\|^2_2. \quad (6)
\]
Back-propagation for Fidelity Cost Term

For detection fidelity cost term the back-propagation is performed by:

1. At iteration step $t$, weights are updated by:

$$\Theta^{t+1} = \Theta^t - \eta \frac{\partial L}{\partial \Theta^t}$$  \hspace{1cm} (7)

where, $\eta$ is learning rate for the stochastic gradient descent method and $\Theta^t$ is the values of weights at previous iteration.

2. $\Theta$ consists of weights from $D$ convolutional layers, following gradients are to be computed: $\frac{\partial L}{\partial W^d}$, where $d = 1, \ldots, D$.

3. For simplicity, we focus on filters and assume that output image $\hat{y}$ is of dimension $N \times N$.

4. For computation of the gradients of the weights at last layer:

$$\frac{\partial L}{\partial W^d} = -(y - \hat{y}) \cdot \frac{\partial \hat{y}}{\partial W^d}$$  \hspace{1cm} (8)

5. $\frac{\partial \hat{y}}{\partial W^d}$ is obtained according to $^{24}$.

$^{24}$Y. LeCun et al., Proc. of the IEEE, 1998
Back-propagation for Shape Priors Cost Term

To carry the shape priors cost term into the $\Theta$, we need to update Eq. (7) accordingly. Examining closely of the Eq. (6), we can re-write it as:

- Updated Eq. (7) will be:

$$
\Theta^{t+1} = \Theta^t - \eta \frac{\partial L}{\partial \Theta^t} - \eta \frac{\partial P}{\partial \Theta^t}.
$$

(9)

- Since, our network parameter $\Theta$ consists of weights from $D$ convolutional layers, following gradients are to be computed: $\frac{\partial P}{\partial W_{m,n}^l}$, where $l = 1, \ldots, D$ and $W$ is of dimension $k_1 \times k_2$ has $m$ by $n$ as the iterators.

- The equations for computing the gradients of weights at last layer are given by:

$$
\frac{\partial P}{\partial W_{m',n'}^l} = \sum_{i=0}^{N-k_1} \sum_{j=0}^{N-k_2} \frac{\partial P}{\partial x_{i,j}^l} \frac{\partial x_{i,j}^l}{\partial W_{m',n'}^l} = \sum_{i=0}^{N-k_1} \sum_{j=0}^{N-k_2} \delta_{i,j}^l \frac{\partial x_{i,j}^l}{\partial W_{m',n'}^l},
$$

(10)
Back-propagation for Shape Priors Cost Term

- where \( x^l_{i,j} \) is the convolved input vector at layer \( l \) plus the bias represented:

\[
x^l_{i,j} = \sum_m \sum_n W^l_{m,n} o^{l-1}_{i+m,j+n} + b^l,
\]

and the output vector at layer \( l \) given by \( o^l_{i,j} = max(x^l_{i,j}, 0) \).

- For \( l = D \) and \( x^D = \hat{y} \):

\[
\delta^D_{i,j} = \frac{\partial P}{\partial x^D_{i,j}} = -\sum_{i=1}^{n} g_p^{-1}(x^D_{i,j} \odot \hat{x}) \ast rot_{180^\circ} \{S_{m,n}\},
\]

where \( g_p^{-1}(\cdot) \) is assign the weights to where it comes from - the “winning unit” because other units in the previous layer’s pooling blocks did not contribute to it hence all the other assigned values of zero. For the mathematical notations please refer to \(^{25}\) and \(^{26}\).

\(^{25}\) V. Dumoulin et al., arXiv 2016
\(^{26}\) Y. LeCun et al., Proc. of the IEEE, 1998
Preparation of Training Data

- **Raw input (x)**
  - X in void region
  - ✓ on tissue
  - ✓ on tissue

- **Raw edge (x̂)**
  - X no edges
  - ✓ with edges
  - ✓ with edges

- **True labels (y)**
  - X no center
  - ✓ with centers
Example Results

Groundtruth

Detection by SP-CNN; F1-score = 0.881

Detection by SC-CNN; F1-score = 0.838

Detection by SR-CNN; F1-score = 0.827
Example Results

Groundtruth

SP-CNN

SC-CNN5

SR-CNN