Transcribing Content from Structural Images with Spotlight Mechanism

Yu Yin, Zhenya Huang, Enhong Chen*, Qi Liu, Fuzheng Zhang, Xin Xie, Guoping Hu
Anhui Province Key Laboratory of Big Data Analysis and Application, University of Science and Technology of China; Microsoft Research Asia; iFLYTEK Research

Reporter: Yu Yin
Date: 2018/8/21
Outline

• Introduction
• Preliminaries
• Spotlight Transcribing Network
• Experiments
• Conclusion
Outline

- **Introduction**
- Preliminaries
- Spotlight Transcribing Network
- Experiments
- Conclusion
Image Transcription

• Definition: recognizing semantic information in images into comprehensible forms (e.g., text)
• tldr: from image to text
• Examples:
Related Research

Related topics:
• Optical Character Recognition (OCR)
• Scene Text Recognition

Related methods:
• Rule-based / feature-based methods
• Encoder-decoder architecture

• Mainly focus on images with straightforward content
Structural Images

• Natural images / printed documents:

• Structural images:
Structural Image Transcription

• The problem: structural image transcription
  • From: structural images
  • To: source code in the specific language that fully defines the image (i.e. able to generate back the image)

\[
f(x) = \frac{\sqrt{x - 1}}{x - 2}
\]

- Music score
- LilyPond code
- Formula image
- TeX code
Challenges

- Unique characteristics:
  - Objects organized in a **complex manner**
  - **Much** semantics in **small** area
  - **Similar** objects

(a) Music score example

\[ f(x) = \frac{\sqrt{x-1}}{x-2} \]

(b) Formula example
Our Approach

• Spotlight mechanism:
  • Mimic human visual attention more intuitively
  • Focus on one spot, write down the content, then focus on another spot
  • Simulate this process by shedding a spotlight and modeling its movement
Outline

• Introduction
• Preliminaries
• Spotlight Transcribing Network
• Experiments
• Conclusion
Data Description

- Datasets: Melody, Formula, Multi-Line
- Compare with scene text recognition: SVT, IIIT5K
- Three main differences:
  - More information to preserve
  - Larger output space
  - Complete transcribing: reversible

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Image count</th>
<th>Token space</th>
<th>Token count</th>
<th>Avg. tokens per image</th>
<th>Avg. image pixels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Melody</td>
<td>4208</td>
<td>70</td>
<td>82,834</td>
<td>19.7</td>
<td>15,602.7</td>
</tr>
<tr>
<td>Formula</td>
<td>61649</td>
<td>127</td>
<td>607,061</td>
<td>9.7</td>
<td>1,190.7</td>
</tr>
<tr>
<td>Multi-Line</td>
<td>4595</td>
<td>127</td>
<td>182,112</td>
<td>39.8</td>
<td>9,016.6</td>
</tr>
<tr>
<td>SVT</td>
<td>618</td>
<td>26</td>
<td>3,796</td>
<td>5.9</td>
<td>12,733.5</td>
</tr>
<tr>
<td>IIIT5K</td>
<td>3000</td>
<td>36</td>
<td>15,269</td>
<td>5.0</td>
<td>11,682.0</td>
</tr>
</tbody>
</table>

![Graphs showing content length distribution for different datasets](image-url)
Problem Definition

• Input: $W \times H$ gray-scale image $x$
• Output: token sequence $y = \{y_1, \ldots, y_t\}$ representing source code
• Note: the process is reversible
Outline

- Introduction
- Preliminaries
- **Spotlight Transcribing Network**
- Experiments
- Conclusion
Spotlight Transcribing Network

At a glance:

• **Encoder-decoder** architecture

• Two-stage “where-to-what” solution:
  • *Spotlight* module for “where-to-look” problem
  • GRU based transcribing module for “what-to-write” problem

• Two implementations:
  • **STNM** with Markov property
  • **STNR** with Recurrent modeling

• Refinement through **reinforcement learning**
Spotlight vs. Attention

Both highlighting different parts when decoding by assigning different weights to feature vectors, but:

- Attention: has to learn to focus
- Spotlight: focus by design

- Attention weights are assigned based on “what” but not “where”
- Spotlight directly finds a reading path that follows the structure

- Attention and transcription are modeled together
- Separate modules for spotlight and transcription
STN Framework

• Spotlight Transcribing Network (STN):
  • Image encoder: CNN based feature extractor
  • Transcribing decoder: Spotlight Module and Transcription Module
Image Encoder

• Image encoder: feature extraction

• CNN: capture high-level semantic information

• Following ResNet
  • ReLU activation
  • Residual connections
  • Batch normalization

• Result: $V^{(i,j)}$, each vector at (i,j) represents local semantic information
Transcribing Decoder

• Transcribing decoder:
  • Generating one token at a time: $P(y_t|y_1, \ldots, y_{t-1}, V) = \text{Softmax}(h_t \oplus sc_t \oplus s_t; \theta_d)$,
Spotlight Mechanism

• Goal: **focus** on certain area when generating, get spotlight context $s_{ct}$

• Approach: shed a spotlight around a spotlight center
  • Spotlight center: *Spotlight handle* ($s_t$)
    • representing the spotlight: center position and width
      $$s_t = (x_t, y_t, \sigma_t)^T$$
  • Assign feature weights following Gaussian distribution:

$$\alpha_{t}^{(i,j)} \sim \mathcal{N}((i,j)^T|\mu_t, \Sigma_t),$$

$$\mu_t = (x_t, y_t)^T \quad \Sigma_t = \begin{bmatrix} \sigma_t & 0 \\ 0 & \sigma_t \end{bmatrix}.$$
Spotlight Mechanism (cont.)

- Weight calculation:
  - Make it differentiable:

\[
\alpha_{t}^{(i,j)} = \text{Softmax}(b_t) = \frac{\exp(b_{t}^{(i,j)})}{\sum_{u=1}^{W'} \sum_{v=1}^{H'} \exp(b_{t}^{(u,v)})},
\]

\[
b_{t}^{(i,j)} = \frac{(i-x_t)^2 + (j-y_t)^2}{\sigma_t^2},
\]

- Make it parallel:

\[
\begin{pmatrix}
1 & 2 & \cdots & W' \\
1 & 2 & \cdots & W' \\
\vdots & \vdots & \ddots & \vdots \\
1 & 2 & \cdots & W'
\end{pmatrix}
- \begin{pmatrix}
x_t & x_t & \cdots & x_t \\
x_t & x_t & \cdots & x_t \\
\vdots & \vdots & \ddots & \vdots \\
x_t & x_t & \cdots & x_t
\end{pmatrix}
\right)^2 + \begin{pmatrix}
1 & 1 & \cdots & 1 \\
2 & 2 & \cdots & 2 \\
\vdots & \vdots & \ddots & \vdots \\
y_t & y_t & \cdots & y_t
\end{pmatrix}
- \begin{pmatrix}
1 & 1 & \cdots & 1 \\
2 & 2 & \cdots & 2 \\
\vdots & \vdots & \ddots & \vdots \\
y_t & y_t & \cdots & y_t
\end{pmatrix}
\right)^2 / \sigma_t^2
\]
Spotlight Control

• Controlling spotlight: moving spotlight handle $s_t$
  • STNM: $s_t$ only depends on $s_{t-1}$

• STNR: $s_t$ are modeled as a sequence by another GRU
Training and Refining STN

- Training: standard backpropagation
- Refine **spotlight control** for a better reading path
- Reinforcement learning setup:
  - State: output history, spotlight context and handle: $h_t \oplus sc_t \oplus s_t$
  - Action: output generation at each step
  - Reward: *reconstruction similarity* (with grammar constraints)
  - Algorithm: Actor-Critic
Outline

• Introduction
• Preliminaries
• Spotlight Transcribing Network
• Experiments
• Conclusion
Experiments

• Data description:

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Image count</th>
<th>Token space</th>
<th>Token count</th>
<th>Avg. tokens per image</th>
<th>Avg. image pixels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Melody</td>
<td>4208</td>
<td>70</td>
<td>82,834</td>
<td>19.7</td>
<td>15,602.7</td>
</tr>
<tr>
<td>Formula</td>
<td>61649</td>
<td>127</td>
<td>607,061</td>
<td>9.7</td>
<td>1,190.7</td>
</tr>
<tr>
<td>Multi-Line</td>
<td>4595</td>
<td>127</td>
<td>182,112</td>
<td>39.8</td>
<td>9,016.6</td>
</tr>
</tbody>
</table>

• Melody and Formula are real-world structural image datasets
• Multi-line specifically for demonstrating complex structure

• Data partitioned into 60%/40%, 70%/30%, 80%/20%, 90%/10% as training/testing sets
• Available on my website
Experiments

• Baselines:
  • **Enc-Dec**: plain encoder-decoder model
  • **Attn-Dot**: encoder-decoder architecture with attention mechanism that calculate weights by inner production
  • **Attn-FC**: soft visual attention
  • **Attn-Pos**: attention with position embedding

• Metrics: transcription accuracy
Experimental Results

- **Accuracy:**
  - STN has better performance
  - STN is able to better capture structural information
  - STN better fits images with even more complex structure

<table>
<thead>
<tr>
<th>Baseline</th>
<th>Testing set percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>40%</td>
</tr>
<tr>
<td>EncDec</td>
<td>0.266</td>
</tr>
<tr>
<td>AttnDot</td>
<td>0.524</td>
</tr>
<tr>
<td>AttnFC</td>
<td>0.683</td>
</tr>
<tr>
<td>AttnPos</td>
<td>0.725</td>
</tr>
<tr>
<td>STNM</td>
<td>0.729</td>
</tr>
<tr>
<td><strong>STNR</strong></td>
<td><strong>0.738</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Baseline</th>
<th>Testing set percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>40%</td>
</tr>
<tr>
<td>EncDec</td>
<td>0.405</td>
</tr>
<tr>
<td>AttnDot</td>
<td>0.530</td>
</tr>
<tr>
<td>AttnFC</td>
<td>0.657</td>
</tr>
<tr>
<td>AttnPos</td>
<td>0.716</td>
</tr>
<tr>
<td>STNM</td>
<td>0.717</td>
</tr>
<tr>
<td><strong>STNR</strong></td>
<td><strong>0.739</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Baseline</th>
<th>Testing set percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>40%</td>
</tr>
<tr>
<td>EncDec</td>
<td>0.218</td>
</tr>
<tr>
<td>AttnDot</td>
<td>0.334</td>
</tr>
<tr>
<td>AttnFC</td>
<td>0.614</td>
</tr>
<tr>
<td>AttnPos</td>
<td>0.624</td>
</tr>
<tr>
<td>STNM</td>
<td>0.674</td>
</tr>
<tr>
<td><strong>STNR</strong></td>
<td><strong>0.712</strong></td>
</tr>
</tbody>
</table>
Experimental Results

• Loss:
  • Better performance
  • Fast convergence
Experimental Results

Ground Truth:

\[ \text{dis}16 \ a \ [ \ b \ c \ b \] \]
Experimental Results

Ground Truth:
\[ f(x) = \frac{\sqrt{x - 1}}{x - 2} \]

\[ f(x) = \frac{\sqrt{x - 1}}{x - 2} \]
Outline

• Introduction
• Preliminaries
• Spotlight Transcribing Network
• Experiments
• Conclusion
Conclusion

• Conclusion
  • Better performance
  • Fast convergence
  • Reasonable reading path: more explainable!

• Future directions:
  • Prior knowledge of the image or the specific language
  • Apply to tasks such as scene text recognition and image captioning
  • Further decouple reading and writing process
Thank you for listening!