Iterative and Adaptive Sampling with Spatial Attention for Black-Box Model Explanations

Bhavan Vasu

Kitware Inc,
Clifton Park, NY, USA 12065.
{bhavan.vasu, chengjiang.long}@kitware.com

Chengjiang Long
Explainable AI - Overview

- Test task
- Optimized Objective
- Learnable Model
- Training Data

Decision

Training

- Why did you do that?
- When do you succeed?
- When do you fail?
- When can I trust you?

Ambiguity in XAI

User: “Are the legs important?!”

“What the model thinks as important is not necessarily what the user thinks as important.”
Iterative and Adaptive Sampling (IAS) - Overview

- The input image is sampled coarsely using a sliding window to obtain an aggregated saliency map.
- Simultaneously, we obtain a spatial attention map of the input image using the LRSA module.
- An adjusted saliency map is obtained after combining the saliency map from previous iteration and attention map.

- This is iteratively repeated till there is little or no change in final saliency maps.
Long-Range Spatial Attention (LRSA) - Overview

- Receptive fields limit the area of consideration to a small window in the image.

- Long-range spatial attention lets us explore and combine long range inter-pixel dependencies to produce an affinity matrix.

- The output of the LRSA module is a spatial attention map. Note that our LRSA module does not contain any learnable parameters.
# Results comparison

Table 1. Comparative evaluation in terms of deletion (lower is better) and insertion (higher is better), F-1 (higher is better), IoU (higher is better), and Pointing Game (higher is better) scores at both image and pixel levels on the MS-COCO dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>Deletion ↓</th>
<th>Insertion ↑</th>
<th>F-1 ↑</th>
<th>IoU ↑</th>
<th>Pointing Game ↑</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Image-level</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LIME</td>
<td>0.900967</td>
<td>0.99</td>
<td>0.15390</td>
<td>0.09745</td>
<td>0.16461</td>
</tr>
<tr>
<td>RISE</td>
<td><strong>0.1847</strong></td>
<td><strong>1.0</strong></td>
<td>0.13837</td>
<td>0.13653</td>
<td>0.25</td>
</tr>
<tr>
<td>IASSA</td>
<td>0.18803</td>
<td><strong>1.0</strong></td>
<td><strong>0.23658</strong></td>
<td><strong>0.15153</strong></td>
<td><strong>0.4216</strong></td>
</tr>
<tr>
<td><strong>Pixel-level</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LIME</td>
<td>10.8526e-05</td>
<td>10.96158e-05</td>
<td>1.71177e-05</td>
<td>1.08447e-05</td>
<td>0.43671e-05</td>
</tr>
<tr>
<td>RISE</td>
<td>5.5423e-05</td>
<td>28.8669e-05</td>
<td>4.26672e-05</td>
<td>2.69240e-05</td>
<td>8.95937e-05</td>
</tr>
<tr>
<td>IASSA</td>
<td><strong>5.50534e-05</strong></td>
<td><strong>35.33639e-05</strong></td>
<td><strong>10.5960e-05</strong></td>
<td><strong>6.9282e-05</strong></td>
<td><strong>17.79331e-05</strong></td>
</tr>
</tbody>
</table>

---

![Input Image](image1.png) ![LIME](image2.png) ![RISE](image3.png) ![IASSA](image4.png) ![Input Image](image5.png) ![LIME](image6.png) ![RISE](image7.png) ![IASSA](image8.png)
Results across $k$ iterations

(a) Input Image  (b) $k = 5$  (c) $k = 10$  (d) $k = 15$  (e) $k = 20$  (f) $k = 25$
Thank you!

Please visit us at Poster 7 to know more.