Adversarial Scene Editing: Automatic Object Removal from Weak Supervision

Rakshith Shetty, Mario Fritz, and Bernt Schiele

Neur-IPS 2018

Presented by Deep Jigar Kotecha
Goal

- Automatic object removal using little supervision.
Contemporary problems

- Limited to object centric datasets.
- Image quality on scene-level images is mediocre.
- Assumes availability of masks.
Contribution

- End to end model for automatic object removal.
- Achieves on-par performance.
Approach

- 2 stage editor model
Stage 1 - Mask generator ( $G_M$ )

- Given the image and the target class generates a binary mask.

\[ m = G_M(x, c_t) \]
Stage 2 - Image inpainter ( $G_i$ )

- Does the inpainting.
- Learns to produce a realistic image.

$$y = \tilde{m} \cdot x + m \cdot G_I (\tilde{m} \cdot x)$$
Architecture - Mask generator

- Pretrained VGG-19
- Remove two maxpool layers
- Remove the FC layers
Architecture - Mask generator

● On top of that:

$$C^3_{512} - L_{0.1} - R_{512} - C^3_{256} - L_{0.1} - R_{256} - C^3_{128} - L_{0.1} - R_{128} - C^7_{21} - S$$
Architecture - Mask generator

- On top of that:

\[ C_{512}^3 - L_{0.1} - R_{512} - C_{256}^3 - L_{0.1} - R_{256} - C_{128}^3 - L_{0.1} - R_{128} - C_{21}^7 - S \]

where \( R_n = C_n^3 - I_n - L_{0.1} - C_n^3 - I_n - L_{0.1} \)
Architecture - Image inpainter

1. Downsampling block
2. Bottle-neck block
3. Upsampling block
Architecture - Image inpainter

- Downsampling block

\[ C^4_{64} - I_{64} - L_{0.1} - D^2_{128} - I_{128} - L_{0.1} - D^2_{256} - I_{256} - L_{0.1} - D^2_{512} - I_{512} - L_{0.1} \]
Architecture - Image inpainter

- Downsampling block

\[
C_{64}^4 - I_{64} - L_{0.1} - D_{128}^2 - I_{128} - L_{0.1} - D_{256}^2 - I_{256} - L_{0.1} - D_{512}^2 - I_{512} - L_{0.1}
\]

where \(D_{n}^2\) = conv layer - 4 x 4 - stride 2
Architecture - Image inpainter

- Bottleneck block

\[ R_{256} - R_{256} - R_{256} - R_{256} - R_{256} - R_{256} \]
Architecture - Image inpainter

- Upsampling block

\[ U^2 - C_{256}^3 - I_{256} - L_{0.1} - U^2 - C_{128}^3 - I_{128} - L_{0.1} - U^2 - C_{64}^3 - I_{64} - L_{0.1} - C_3^7 - T \]
Architecture - Image inpainter

- Upsampling block

\[ U^2 \circlearrowleft C_{256}^3 - I_{256} - L_{0.1} \circlearrowleft U^2 \circlearrowleft C_{128}^3 - I_{128} - L_{0.1} \circlearrowleft U^2 \circlearrowleft C_{64}^3 - I_{64} - L_{0.1} - C_3^7 - T \]
Training - Mask generator

- Trained to fool the object classifier for the target class.
Training - Mask generator
Training - Mask generator

- Mask Generator
- Editor
- Image In-painter
- Object Classifier: Is there a person?
- Real/Fake Classifier: Real/Fake?
Training - Mask generator

- Trained to fool the object classifier for the target class.

\[ L_{cls}(G_M) = - \mathbb{E}_x [\log(1 - D_{cls}(y, c_t))] \]
Training - Image inpainter

- Trained to only fool the real/fake classifier.
Training - Image inpainter
Training - Image inpainter
Training - Image inpainter

- Trained to only fool the real/fake classifier.

\[ L_{rf}(G_I) = - \mathbb{E}_x [D_{rf}(y)] \]
Mask priors
Mask priors

- Random rectangular boxes
- Unpaired data
Mask priors

Mask Generator

Samples from mask prior

Mask Discriminator

Prior / Generated

Class Label

to In-Painter
Training - Mask discriminator

- Minimize the Wasserstein distance using WGAN.

\[ L(D_M) = \mathbb{E}_{m^p \sim P(m^p|c_t)} [D_M(m^p, c_t)] - \mathbb{E}_x [D_M(G_M(x, c_t), c_t)] \]

\[ L_{\text{prior}}(G_M) = -\mathbb{E}_x [D_M(G_M(x, c_t), c_t)] \]
Mask generator loss

\[ L(G_M) = - \mathbb{E}_x \left[ \log(1 - D_{cls}(y, c_t)) \right] - \mathbb{E}_x \left[ D_M(m, c_t) \right] + \exp(\sum m_{ij}) \]

- **Fool object classifier**
- **Adversarial prior**
- **Size regularization**
Optimizing inpainter - reconstruction

- Reconstructs random image patches from fooling adversarial real/fake classifier
- Hence, reconstruction loss:

\[ L_{\text{recon}}(G_1) = \|G_1(\tilde{m}^r \cdot x) - x\|_1 + \sum_k \|\phi_k(G_1(\tilde{m}^r \cdot x)) - \phi_k(x)\|_1 \]

- L1
- Perceptual loss
Optimizing inpainter - local labels

- Drawback with $[D_{rf}(y)]$
- Overcome by providing local pixel-level real/fake labels.
Optimizing inpainter - local labels
Optimizing inpainter - local labels

- Use the least-square GAN loss.

\[
L(D_{rf}) = \frac{\sum_{ij} (1 - m_{ij}) (D_{rf}(y)_{ij} - 1)^2}{\sum_{ij} (1 - m_{ij})} + \frac{\sum_{ij} m_{ij} \cdot (D_{rf}(y)_{ij} + 1)^2}{\sum_{ij} m_{ij}}
\]
Optimizing inpainter - style & variation

- Incorporate the style-loss ($L_{sty}$) and the total variation loss ($L_{tv}$)
Final loss function - Mask generator

\[ L_{\text{total}}(G_M) = L_{\text{cls}} + L_{\text{prior}} + \exp(\sum_{i,j} m_{ij}) \]
Final loss function - Mask generator

\[ L_{\text{total}}(G_M) = \lambda_c L_{\text{cls}} + \lambda_p L_{\text{prior}} + \lambda_{sz} \exp(\Sigma_{ij} m_{ij}) \]
Final loss function - Image inpainter

\[ L_{\text{total}}(G_I) = L_{\text{rf}} + L_{\text{recon}} + L_{\text{tv}} + L_{\text{sty}} \]
Final loss function - Image inpainter

\[ L_{\text{total}}(G_I) = \lambda_{rf} L_{rf} + \lambda_r L_{\text{recon}} + \lambda_{tv} L_{tv} + \lambda_{sty} L_{sty} \]
Dataset - 1

- MS COCO - filter out images containing large objects (~covering more than 30%)
- Shared with Pascal-VOC 2012.
- 20 shared classes.
- Resized to 128x128.
Dataset - 2

- For application in automatic content filtering and visual privacy filtering.
- Flickr Logos
- Uses box prior
Evaluation metrics

- Removal performance
- Image quality
- Human evaluation
Removal performance

- Removal success rate - ↑
- False removal rate - ↓
Image quality assessment

- pSNR (Peak Signal to Noise ratio) - ↑
- ssim (Structural Similarity Index) - ↑
- Perceptual loss - ↓
Human evaluation

- Obtain human judgments of the performance.
- Show 100 edited images to 3 separate judges.
Quantitative results

- Comparison to GT and MaskRCNN baselines.

<table>
<thead>
<tr>
<th>Model</th>
<th>Supervision</th>
<th>Removal Performance</th>
<th>Image quality metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>removal success ↑</td>
<td>false ↓ removal</td>
</tr>
<tr>
<td></td>
<td></td>
<td>all</td>
<td>person</td>
</tr>
<tr>
<td>GT masks</td>
<td>-</td>
<td>66</td>
<td>72</td>
</tr>
<tr>
<td>Mask RCNN</td>
<td>Seg. masks &amp; bound boxes</td>
<td>68</td>
<td>73</td>
</tr>
<tr>
<td>Mask RCNN (dil. 7x7)</td>
<td></td>
<td>75</td>
<td>77</td>
</tr>
<tr>
<td>ours-pascal</td>
<td>image labels &amp; unpaired masks</td>
<td>73</td>
<td>81</td>
</tr>
</tbody>
</table>
Quantitative results

- Effect of different priors.
Quantitative results

- Effect of different priors.

<table>
<thead>
<tr>
<th>Prior</th>
<th>Removal Performance</th>
<th>Image quality metrics</th>
<th>Mask accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>removal success ↑</td>
<td>false ↓ removal</td>
<td>mIoU ↑</td>
</tr>
<tr>
<td></td>
<td>all</td>
<td>person</td>
<td>% masked area ↓</td>
</tr>
<tr>
<td>None</td>
<td>94</td>
<td>96</td>
<td>0.15</td>
</tr>
<tr>
<td>boxes</td>
<td>83</td>
<td>88</td>
<td>0.18</td>
</tr>
<tr>
<td>pascal (10)</td>
<td>67</td>
<td>59</td>
<td><strong>0.23</strong></td>
</tr>
<tr>
<td>pascal (100)</td>
<td>70</td>
<td>75</td>
<td>0.22</td>
</tr>
<tr>
<td>pascal (all)</td>
<td>73</td>
<td>81</td>
<td>0.22</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>37.7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>28.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>16.7</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>18.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>20.2</td>
</tr>
</tbody>
</table>

- Effect of different priors.

- For each prior, the table shows the removal success rate for all and person categories, as well as the false removal rate. The image quality metrics include percep. loss (↑), pSNR (↑), and ssim (↑), with higher values indicating better performance. The mask accuracy is measured by mIoU (↑) and the percentage of masked area (↓), with higher values indicating better performance.
Qualitative results

Input image

M-RCCN based

Ours

dog  person  tv  airplane  person  cow  person
Qualitative results
Failure cases
## Ablation study - 1

- Joint optimization

<table>
<thead>
<tr>
<th>Joint training</th>
<th>Removal success</th>
<th>mIoU</th>
<th>percep. loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>0.68</td>
<td>0.19</td>
<td>0.10</td>
</tr>
<tr>
<td>✓</td>
<td>0.73</td>
<td>0.22</td>
<td>0.08</td>
</tr>
</tbody>
</table>
Ablation study - 2

- Ablation of in-painting components

<table>
<thead>
<tr>
<th>Mask buffer</th>
<th>GAN</th>
<th>TV+ Style</th>
<th>percep. loss ↓</th>
<th>pSNR ↑</th>
<th>ssim ↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>G</td>
<td>-</td>
<td>0.13</td>
<td>20.0</td>
<td>0.730</td>
</tr>
<tr>
<td>✓</td>
<td>G</td>
<td>-</td>
<td>0.12</td>
<td>21.9</td>
<td>0.772</td>
</tr>
<tr>
<td>✓</td>
<td>L</td>
<td>-</td>
<td>0.10</td>
<td>21.5</td>
<td>0.758</td>
</tr>
<tr>
<td>✓</td>
<td>L</td>
<td>✓</td>
<td>0.10</td>
<td>21.6</td>
<td>0.763</td>
</tr>
</tbody>
</table>
Recapitulation

● Automatic object removal model.
● Uses only image level labels and unpaired data.
● Achieves on-par performance to fully-supervised segmenter based removal methods.
Questions?
References