Learning to Simplify: Fully Convolutional Networks for Rough Sketch Cleanup

Edgar Simo-Serra*, Satoshi Iizuka*, Kazuma Sasaki, Hiroshi Ishikawa (*equal contribution)

July 27th, 2016

Waseda University
Sketch Simplification

Play Speed x300
Video: David Revoy, www.davidrevoy.com
Sketch Simplification

Input: Rough Sketch

Output: Line Art
Sketch Simplification

Rough    Target    Rough    Target

Rough    Target    Rough    Target

Rough    Target

Rough    Target

Rough    Target
Related Work
Related Work

1. Sketch Simplification
   1.1 Progressive Online Modification
   1.2 Stroke Reduction
   1.3 Stroke Grouping

Liu et al. 2015
Related Work

1. Sketch Simplification
   1.1 Progressive Online Modification
   1.2 Stroke Reduction
   1.3 Stroke Grouping
   1.4 Vector input

Liu et al. 2015
Related Work

1. Sketch Simplification
   1.1 Progressive Online Modification
   1.2 Stroke Reduction
   1.3 Stroke Grouping
   1.4 Vector input

2. Vectorization
   2.1 Model Fitting (Bezier, ...)
   2.2 Gradient-based approaches

Noris et al. 2013
Related Work

1. Sketch Simplification
   1.1 Progressive Online Modification
   1.2 Stroke Reduction
   1.3 Stroke Grouping
   1.4 Vector input

2. Vectorization
   2.1 Model Fitting (Bezier, …)
   2.2 Gradient-based approaches
   2.3 Require fairly clean input sketches

Noris et al. 2013
Related Work

1. Sketch Simplification
   1.1 Progressive Online Modification
   1.2 Stroke Reduction
   1.3 Stroke Grouping
   1.4 Vector input

2. Vectorization
   2.1 Model Fitting (Bezier, ...)
   2.2 Gradient-based approaches
   2.3 Require fairly clean input sketches

3. Deep Learning
   3.1 Fully Convolutional Network

Long et al. 2015
Proposed Approach
Deep Learning

- Modern Neural Networks
  - Computational efficiency with GPU
  - Large scale datasets
Deep Learning

- Modern Neural Networks
  - Computational efficiency with GPU
  - Large scale datasets
- Learns input to output mapping
- Basic building block layer: $f(x) = \sigma(Wx)$

![Diagram of a neural network](image-url)
Deep Learning

- Modern Neural Networks
  - Computational efficiency with GPU
  - Large scale datasets
- Learns input to output mapping
- Basic building block layer: $f(x) = \sigma(Wx)$
- Parameters ($W \ldots$) are learnt
- Hyper-parameters are set by hand
Fully Convolutional Network

- Uses only convolutional layers
- Each layer convolves many filters
- Layer hyperparameters: kernel, padding, and stride
  - Weights expressed with kernels
  - Padding conserves the image size
  - Stride can change the output resolution
We create three building blocks by modifying the stride:

1. Flat-convolution
   1.1 3 × 3px kernel, 1 × 1px padding, 1px stride
2. Down-convolution
   2.1 3 × 3px kernel, 1 × 1px padding, 2px stride
3. Up-convolution
   3.1 4 × 4px kernel, 1 × 1px padding, 1/2px stride
Model

- 23 convolutional layers
- Output has the same resolution as the input
- Encoder-Decoder architecture
  - Reduces memory usage
  - Increases spatial resolution
Learning

- Trained from scratch
- Using 424 × 424px patches
- Weighted Mean Square Error loss
- Batch Normalization [Ioffe and Szegedy 2015] is critical
- Optimized with ADADELTA [Zeiler 2012]
Vectorization and Simplification

- Vectorization with potrace
  - Open source software
  - High pass filter and binarization
Vectorization and Simplification

- Vectorization with potrace
  - Open source software
  - High pass filter and binarization
- Scaling input changes simplification degree
Sketch Dataset
Sketch dataset

- 68 pairs of rough and target sketches
- 5 illustrators

[Diagram showing sketch dataset and extracted patches]
Inverse Dataset Creation

• Data quality is critical
• Creating target sketches from rough sketches has misalignments
• Creating rough sketches from target sketches properly aligns
Data Augmentation

- 68 pairs is insufficient
- Scaling training data
- Random cropping, flipping and rotation
- Additional augmentation: tone, slur, and noise
Results and Comparisons
Results

- Intel Core i7-5960X CPU (3.00GHz)
- NVIDIA GeForce TITAN X GPU
- 3 weeks training time

<table>
<thead>
<tr>
<th>Image Size</th>
<th>Pixels</th>
<th>CPU (s)</th>
<th>GPU (s)</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>320 × 320</td>
<td>102,400</td>
<td>2.014</td>
<td>0.047</td>
<td>42.9×</td>
</tr>
<tr>
<td>640 × 640</td>
<td>409,600</td>
<td>7.533</td>
<td>0.159</td>
<td>47.4×</td>
</tr>
<tr>
<td>1024 × 1024</td>
<td>1,048,576</td>
<td>19.463</td>
<td>0.397</td>
<td>49.0×</td>
</tr>
</tbody>
</table>
Comparison

Input

Potrace

Adobe Live Trace

Ours
User Study

• Comparison with 15 images
• 19 users participated (10 with illustration experience)
• Absolute rating (1 to 5 scale)
• Relative evaluation (best of two)

<table>
<thead>
<tr>
<th></th>
<th>Ours</th>
<th>Live Trace</th>
<th>Potrace</th>
</tr>
</thead>
<tbody>
<tr>
<td>Score</td>
<td>4.53</td>
<td>2.94</td>
<td>2.80</td>
</tr>
<tr>
<td>vs Ours</td>
<td>-</td>
<td>2.5%</td>
<td>2.8%</td>
</tr>
<tr>
<td>vs Live Trace</td>
<td>97.5%</td>
<td>-</td>
<td>30.3%</td>
</tr>
<tr>
<td>vs Potrace</td>
<td>97.2%</td>
<td>69.7%</td>
<td>-</td>
</tr>
</tbody>
</table>
Comparison

[Liu et al. 2015] Ours

(a) Fairy (b) Mouse (c) Duck (d) Car
Results
Results
Conclusions

• Automatic Sketch Simplification Approach
• Convolutional networks are suited to image processing
• Proper data is crucial for training
Conclusions

• Automatic Sketch Simplification Approach
• Convolutional networks are suited to image processing
• Proper data is crucial for training

Try Online: [http://hi.cs.waseda.ac.jp:8081/](http://hi.cs.waseda.ac.jp:8081/)
<table>
<thead>
<tr>
<th>type</th>
<th>kernel size</th>
<th>stride</th>
<th>output size</th>
</tr>
</thead>
<tbody>
<tr>
<td>input</td>
<td>-</td>
<td>-</td>
<td>$1 \times H \times W$</td>
</tr>
<tr>
<td>down-convolution</td>
<td>$5 \times 5$</td>
<td>$2 \times 2$</td>
<td>$48 \times H/2 \times W/2$</td>
</tr>
<tr>
<td>flat-convolution</td>
<td>$3 \times 3$</td>
<td>$1 \times 1$</td>
<td>$128 \times H/2 \times W/2$</td>
</tr>
<tr>
<td>flat-convolution</td>
<td>$3 \times 3$</td>
<td>$1 \times 1$</td>
<td>$128 \times H/2 \times W/2$</td>
</tr>
<tr>
<td>down-convolution</td>
<td>$3 \times 3$</td>
<td>$2 \times 2$</td>
<td>$256 \times H/4 \times W/4$</td>
</tr>
<tr>
<td>flat-convolution</td>
<td>$3 \times 3$</td>
<td>$1 \times 1$</td>
<td>$256 \times H/4 \times W/4$</td>
</tr>
<tr>
<td>flat-convolution</td>
<td>$3 \times 3$</td>
<td>$1 \times 1$</td>
<td>$256 \times H/4 \times W/4$</td>
</tr>
<tr>
<td>down-convolution</td>
<td>$3 \times 3$</td>
<td>$2 \times 2$</td>
<td>$256 \times H/8 \times W/8$</td>
</tr>
<tr>
<td>flat-convolution</td>
<td>$3 \times 3$</td>
<td>$1 \times 1$</td>
<td>$512 \times H/8 \times W/8$</td>
</tr>
<tr>
<td>flat-convolution</td>
<td>$3 \times 3$</td>
<td>$1 \times 1$</td>
<td>$1024 \times H/8 \times W/8$</td>
</tr>
<tr>
<td>flat-convolution</td>
<td>$3 \times 3$</td>
<td>$1 \times 1$</td>
<td>$1024 \times H/8 \times W/8$</td>
</tr>
<tr>
<td>flat-convolution</td>
<td>$3 \times 3$</td>
<td>$1 \times 1$</td>
<td>$1024 \times H/8 \times W/8$</td>
</tr>
<tr>
<td>flat-convolution</td>
<td>$3 \times 3$</td>
<td>$1 \times 1$</td>
<td>$512 \times H/8 \times W/8$</td>
</tr>
<tr>
<td>flat-convolution</td>
<td>$3 \times 3$</td>
<td>$1 \times 1$</td>
<td>$256 \times H/8 \times W/8$</td>
</tr>
<tr>
<td>up-convolution</td>
<td>$4 \times 4$</td>
<td>$1/2 \times 1/2$</td>
<td>$256 \times H/4 \times W/4$</td>
</tr>
<tr>
<td>flat-convolution</td>
<td>$3 \times 3$</td>
<td>$1 \times 1$</td>
<td>$256 \times H/4 \times W/4$</td>
</tr>
<tr>
<td>flat-convolution</td>
<td>$3 \times 3$</td>
<td>$1 \times 1$</td>
<td>$256 \times H/4 \times W/4$</td>
</tr>
<tr>
<td>up-convolution</td>
<td>$4 \times 4$</td>
<td>$1/2 \times 1/2$</td>
<td>$128 \times H/2 \times W/2$</td>
</tr>
<tr>
<td>flat-convolution</td>
<td>$3 \times 3$</td>
<td>$1 \times 1$</td>
<td>$128 \times H/2 \times W/2$</td>
</tr>
<tr>
<td>flat-convolution</td>
<td>$3 \times 3$</td>
<td>$1 \times 1$</td>
<td>$128 \times H/2 \times W/2$</td>
</tr>
<tr>
<td>up-convolution</td>
<td>$4 \times 4$</td>
<td>$1/2 \times 1/2$</td>
<td>$48 \times H \times W$</td>
</tr>
<tr>
<td>flat-convolution</td>
<td>$3 \times 3$</td>
<td>$1 \times 1$</td>
<td>$24 \times H \times W$</td>
</tr>
<tr>
<td>flat-convolution</td>
<td>$3 \times 3$</td>
<td>$1 \times 1$</td>
<td>$1 \times H \times W$</td>
</tr>
</tbody>
</table>