Learning Photographic Image Synthesis With Cascaded Refinement Networks

Jonathan Louie
Huy Doan
Siavash Motalebi
Introduction and Background
We are researching and re-implementing Photographic Image Synthesis with Cascaded Refinement Networks used for generating photorealistic images. This was originally done by Qifeng Chen and Vladlen Koltun. The Cascaded Refinement Networks uses an input semantic layout produced by humans and generates a realistic looking image in high resolution and in high fidelity. Unlike recent works, this approach does not rely on adversarial training. We show that photographic images can be synthesized from semantic layouts by a single feedforward network with appropriate structure, trained end-to-end with a direct regression objective.
Benefits (Rendering Engine)

One of the goals of implementing this program in the stated way is that by doing so the program would work as rendering engine which can help the user bypass the very difficult and complicated specification of detailed three-dimensional geometry and surface reflectance distributions. It is also able to avoid computationally intensive light transport simulation. Although this approach cannot replace the traditional rendering engines at the moment, it does show that there are other possible ways to implement these rendering engines.
Benefits (Human Cognition)

Another reason is that this program can also be used to give more insight into the role of mental imagery and simulation in human cognition. This is an important part of the decision making process that each human goes through. The level of detail and completeness of mental imagery is a matter of debate, but its role in human intelligence suggests that the ability to synthesize photo realistic images can help a significant amount in development of artificial intelligence.
History
The most common approach to image synthesis is based on generative adversarial networks. This method which was originally implemented in the paper “Generative Adversarial Nets (GAN)” was used to synthesize MNIST (Modified National Institute of Standards and Technology database) digits and $32 \times 32$ images. Each model is trained independently to synthesize details at its scale. Assembling separately trained models this way allows the program to synthesize smoother images and to push resolution up to $96 \times 96$.

GANs are known to be remarkably difficult to train and when approaching to solve this problem one still relies on heuristics that are extremely sensitive to modifications. Although this method is not fully similar to this project, it attempts to do a similar task.
Different Image Synthesis Methods

In the paper “Learning to Generate Chairs, Tables and Cars with Convolutional Networks.” the authors trained a ConvNet to generate images of 3D models, given a model ID and viewpoint. The network thus acts directly as a rendering engine for the 3D model. This is an important precursor to our re-implementation as it uses direct feedforward synthesis through a network trained with a regression loss.

The paper “Generating Images with Perceptual Similarity Metrics Based on Deep Networks” introduced a family of composite loss functions for image synthesis, which combine regression over the activations of a fixed “perceiver” network with a GAN loss. Networks trained using these composite loss functions were applied to synthesize preimages.
Another paper that discusses methods for synthesizing was “Image-to-Image Translation With Conditional Adversarial Networks.” The authors consider a family of problems that include the image synthesis problem we focus on. This method helps us to compare our re-implementation of the program to an alternative that was independently tested on the same data. Like a number of mentioned methods, it uses a composite loss that combines a GAN and a regression term. The authors use the Cityscapes dataset and synthesize 256×256 images for given semantic layouts.
Working With Different Inputs
Examples

- An example of image synthesis with a different input is the program described in the paper “Generative Adversarial Text to Image Synthesis” which synthesizes images of the size 64x64 of scenes that are described to it.
- The program introduced in “Conditional Image Generation From Visual Attributes” synthesizes images of faces and birds with given attributes.
- 128×128 images of birds and people conditioned on text descriptions and on spatial constraints such as bounding boxes or keypoints is created by the program described in “Learning What and Where to Draw”.
Architecture
Architecture

- Build a CNN with a cascade of “refinement modules”
  - Start with low-res feature maps, successively refine to higher resolution feature maps
  - Repeat until desired resolution reached
- For diverse photorealistic outputs, need extremely high model capacity
  - Model used 105 million parameters
Architecture
Architecture

- Each module is just three layers of a CNN
  - Each module doubles the resolution
  - Input layer: upsampled feature maps of previous module + downsampled semantic layout
  - Intermediate layer
  - Output layer
Architecture

- Model uses an initial resolution of 4 x 8
- First module has 1024 feature maps
- Reduce the number of feature maps as the cascade gets deeper until the final module outputs 3 channels
Architecture

- Loss function for training

\[
\mathcal{L}_{I,L}(\theta) = \sum_{l} \lambda_l \| \Phi_l(I) - \Phi_l(g(L; \theta)) \|_1
\]
Architecture

- Loss function modified
  - Want to generate multiple diverse images given a single semantic layout
  - Solution: network synthesizes $k$ images, define loss in terms of the best image
  - Encourages network to generate a wide spread of images

$$\min_u \sum_l \lambda_l \| \Phi_l(I) - \Phi_l(g_u(L; \theta)) \|_1$$
Architecture

- Loss function modified further
  - Another extension: redefine loss in terms of best generated image for each semantic class
  - Gives even more diverse collections

\[
\sum_{p=1}^{c} \min_u \sum_l \lambda_l \sum_j \left\| L^l_p \odot (\Phi^j_l(I) - \Phi^j_l(g_u(L; \theta))) \right\|_1
\]
Result
Results

<table>
<thead>
<tr>
<th></th>
<th>Image-space loss</th>
<th>GAN+SemSeg</th>
<th>Isola et al.</th>
<th>Encoder-decoder</th>
<th>Full-resolution network</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cityscapes</td>
<td>99.7%</td>
<td>98.5%</td>
<td>96.9%</td>
<td>78.3%</td>
<td>67.7%</td>
</tr>
<tr>
<td>NYU</td>
<td>91.4%</td>
<td>82.3%</td>
<td>77.2%</td>
<td>71.2%</td>
<td>65.8%</td>
</tr>
</tbody>
</table>
Demo
Demo
Demo
Demo
Demo
Demo
Conclusion
Conclusion

As we showed in our demo of the program discussed in the paper Photographic Image Synthesis with Cascaded Refinement Networks, the program is able to synthesize photo realistic images using a single feedforward network with appropriate structure, trained end-to-end with a direct regression objective. There are benefits for understanding and implementing new methods for image synthesis and we hope that by doing so we are able to learn more about this types of programs and the benefits that comes with it them.
Q&A