Machine Learning Image Enhancement

The following tweets, product of an earlier request for comments on an essay about ebooks started an interesting line of thinking.

> Re: image quality, as a book publisher whose backlist is pre-digital, our problem is generally that we don't have high-enough-res source art.
>
> — Teresa Elsey (@teresaelsey) December 11, 2017

> If we're *lucky* we have a digital file for what appears in the original print book! Often we're scanning a print book to recreate the assets ...
>
> — Teresa Elsey (@teresaelsey) December 11, 2017

After the initial thought that poor archival practices can bite you hard I started thinking about a solution to the actual problem. You're scanning from printed media and need a high resolution version that you can run through an image processing pipeline to create 2x and, maybe, 3x retina versions.

If you've seen the demos from Googles Machine Learning and Image processing they can do awesome things with image search and manipulation. They assume good quality images like those shot with phone cameras or some other mid to high quality cameras.

But what if the images are of lower quality, like those we scan from printed documents?

We'll look at Neural Enhance see if does what we need them to.

I'm looking to answer the following questions:

1. Can I improve the quality of a scanned image?
2. Is the quality good enough to use in Retina displays?
3. Will the image quality remain if I run it through an image processing pipeline?

Neural Enhance

As seen on TV! What if you could increase the resolution of your photos using technology from CSI laboratories? Thanks to deep learning and `NeuralEnhance`, it's now possible to train a neural network to zoom in to your images at 2x or even 4x. You'll get even better results by increasing the number of neurons or training with a dataset similar to your low resolution image.

The catch? The neural network is hallucinating details based on its training from example images. It's not reconstructing your photo exactly as it would have been if it was HD. That's only possible in Hollywood — but using deep learning as "Creative AI" works and it is just as cool! Here's how you can get started...

Rather than install all the requirements from scratch I'll use a Docker image to run the experiment. There instructions for installing Docker in Mac and Windows. The rest of this post will assume you have installed Docker and it's running on your system.

> The latest version of Docker for Windows requires Microsoft Windows 10 Professional or Enterprise 64-bit.

Download the Docker image using the `docker` CLI:

```
# Download the Docker image and show the help text to
# make sure it works.
docker run --rm -v `pwd`:/ne/input -it alexjc/neural-enhance --help
`pwd`
```

For Macintosh and Linux users add the following command to your `aliases` or `.bashrc`. This will allow users to run the command as enhance without a `.py` extension.

As the command indicates this should be written as a single line.

```
# All three lines below should be in one line. I've inserted hard returns
alias enhance='function ne() { docker run --rm -v \
```

```
"$(pwd)/`dirname ${@:$#}`":/ne/input -it alexjc/neural-enhance $ ${@:$#}$(
"$(pwd)/`dirname ${@:$#}`":/ne/input -it alexjc/neural-enhance $ \
{@:1:$#-1} "input/`basename ${@:$#}`"; }; ne'
```

The block below shows how to run enhance. The first example repairs the artifacts in an image keeping the resolution of the original.

The second example will create 2x images for all the `jpg` images inside the images folder.

```
enhance --zoom=1 --model=repair images/broken.jpg
# Process multiple images, make sure to quote the argument!
enhance --zoom=2 "images/*.jpg"
"images/*.jpg"
```

For this example we'll use an old black and white image I downloaded from an old article about Simnet. Figure one shows the original image, without retouch.

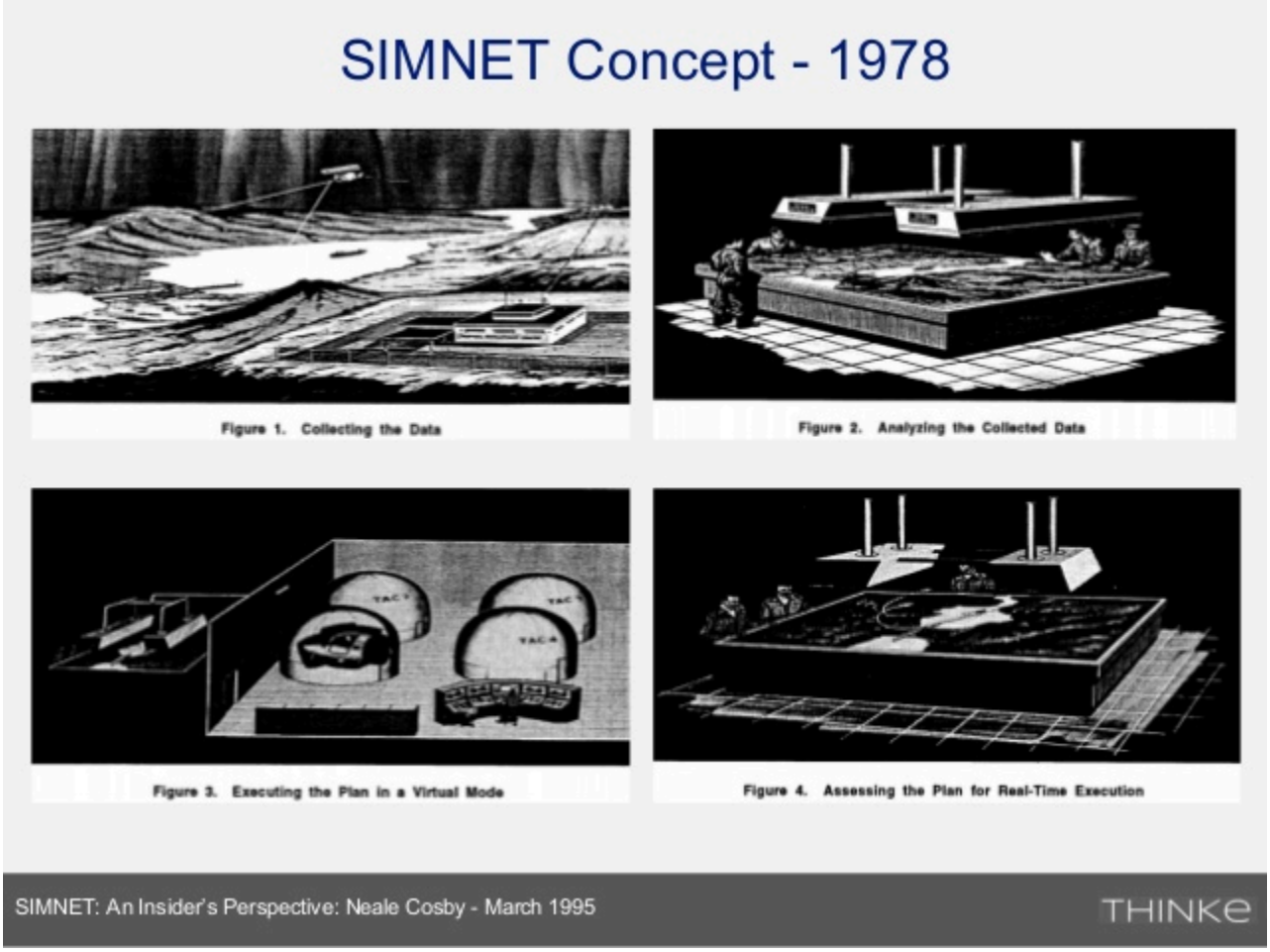

Figure 1: Simnet Concept Base Image

```
# Run the super-resolution script to repair JPEG artefacts, zoom factor 1
enhance --type=photo --model=repair --zoom=1 simnet-concept.jpg
```

Figure two shows the image with JPEG artifacts repaired but still at the same resolution as the original.

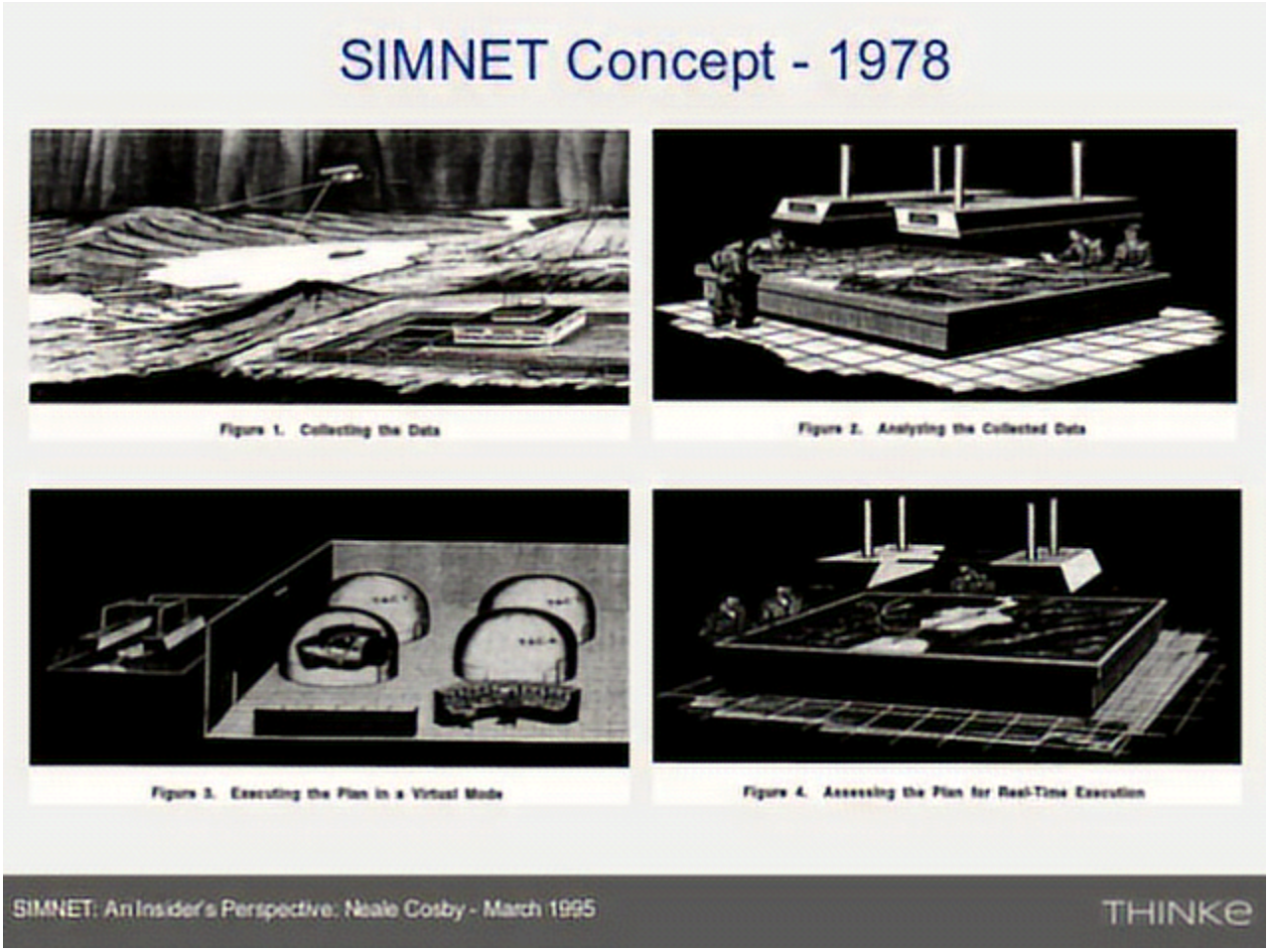

Figure 2: Simnet Concept Image Repaired at 1x Resolution

Figure 3 shows the image at twice the size of the original. It has eliminated the artifacts of the original (if it had any) and increased the size so it's now usable in a 2x Retina display.

```
# Process multiple good quality images with a single run, zoom factor 2:1
```

```
enhance.py --type=photo --zoom=2 simnet-concept.jpg
```

SIMNET Conce

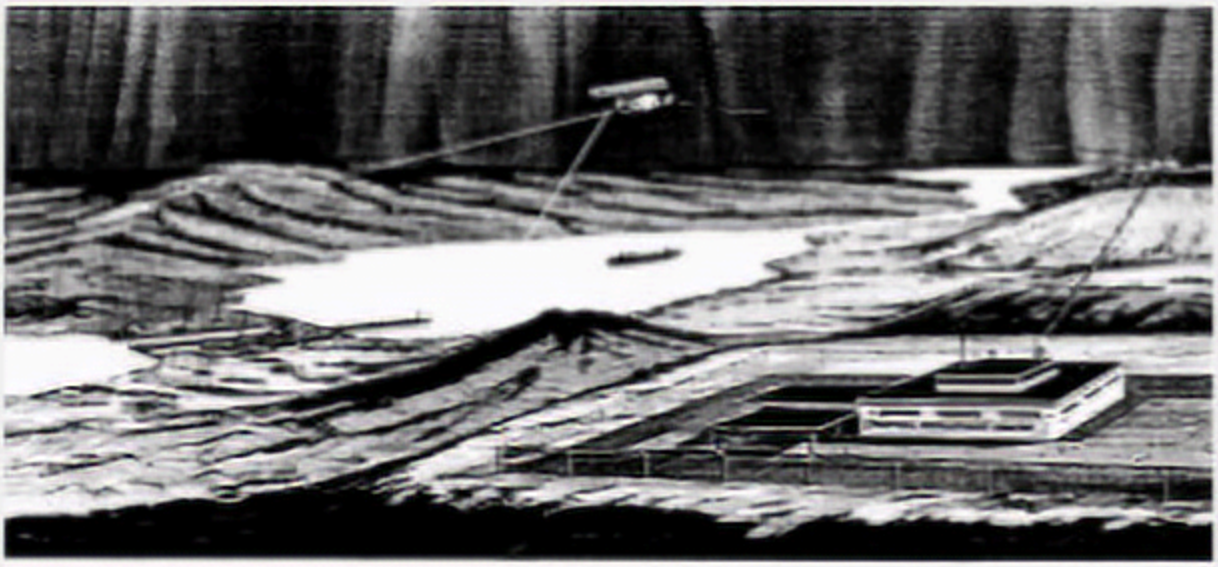

Figure 1. Collecting the Data

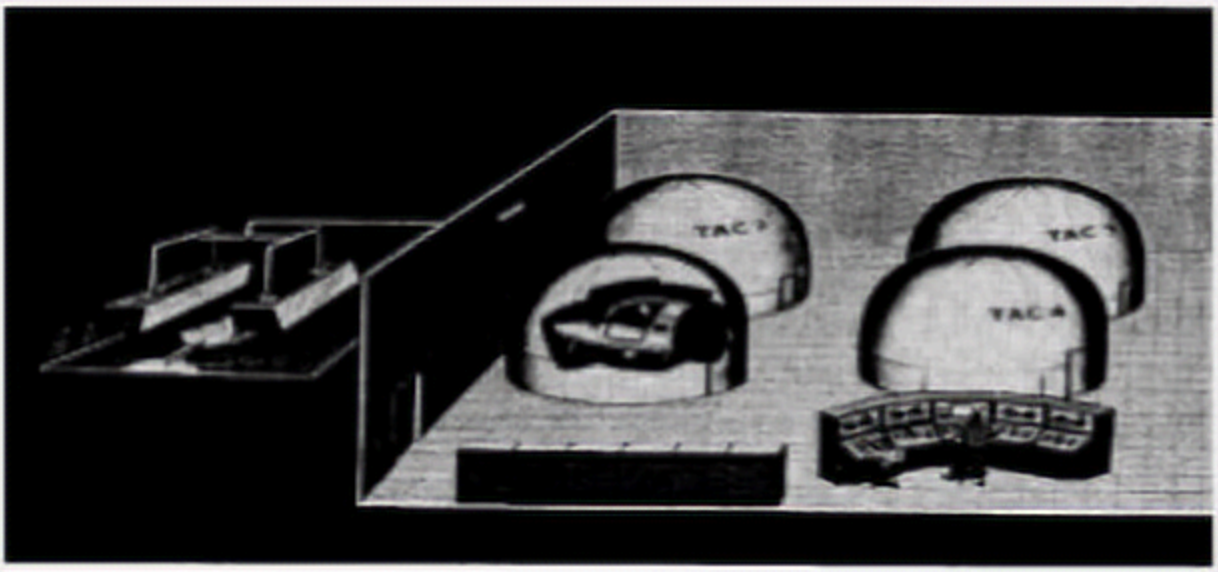

Figure 3. Executing the Plan in a Virtual Mode

Figure 3: Simnet Concept Image at 2x Resolution

Training your own models

The models bundled with Neural Enhance will only work with 1:1 and 2x resolutions. Anyting bigger will require to train your own model

If working with scanned images it may become necessary to create a new model for each book based on the scanned images. Some examples of commands to train models.

First we remove the model file. Don't want to reload the data to fine-tune it.

```
rm -f ne?x*.pkl.bz2
```

Pre-train the model using perceptual loss and algorithm that uses **convolutional neural networks** using a per-pixel loss and **perceptual loss functions** based on high-level features extracted from pretrained networks.

```
enhance --train "data/*.jpg" --model custom \
  --scales=2 --epochs=50 \
  --perceptual-layer=conv2_2 --smoothness-weight=1e7 \
  --adversary-weight=0.0 --generator-blocks=4 \
  --generator-filters=64
```

Train the model using an Generative adversarial network.

```
enhance --train "data/*.jpg" --model custom \
  --scales=2 --epochs=250 \
  --perceptual-layer=conv5_2 --smoothness-weight=2e4 \
  --adversary-weight=1e3 \
  --generator-start=5 --discriminator-start=0 \
  --adversarial-start=5 --discriminator-size=64
```

Additional models can be trained by playing with the different parameter available to the enhance tool. We can get a list of the parameters by running the same command we used to install and check the image:

```
# Download the Docker image and show the help text to
# make sure it works.
docker run --rm -v `pwd`:/ne/input -it alexjc/neural-enhance --help
`pwd`
```

The results look promising but to answer the questions I asked originally:

1. **Can I improve the quality of a scanned image?** I can fix artifacts in the low resolution image that will improve the quality but it's done through interpolation so the quality may not improve as much as you think it does
2. **Is the quality good enough to use in Retina displays?** Using the zoom parameter with a value of 2, it generates an image with the required number of pixels. If you want to work with higher image densities you'll have to train your own model
3. **Will the image quality remain if I run it through an image processing pipeline?** Unless we're generating a very large version of the image, we don't have enough pixels beyond 2x retina. We can train the model interpolate to higher resolutions but the result is far from guaranteed

Links and Resources

- Tools and Libraries
    - Neural Enhance
        - Github
    - Raisr
        - Google's prototype machine learning software lets you enhance low-res photos
        - RAISR: Rapid and Accurate Image Super Resolution
        - Supplementary Material for RAISR: Rapid and Accurate Image Super Resolution
        - Unofficial Python implementation of RAISR
    - Google Brain super-resolution image tech makes "zoom, enhance!" real
    - Website uses neural networks to enlarge small images, and the results are pretty magical

        - Let’s Enhance
    - Photo Enhancement is Starting to Get Crazy
        - EnhanceNet: Single Image Super-Resolution Through Automated Texture Synthesis
    - PRSR
        - Pixel Recursive Super Resolution
        - Github Repository
- Research Background
    - Perceptual Losses for Real-Time Style Transfer and Super-Resolution
    - Real-Time Super-Resolution Using Efficient Sub-Pixel Convolution
    - Deeply-Recursive Convolutional Network for Image Super-Resolution
    - Photo-Realistic Super-Resolution Using a Generative Adversarial Network