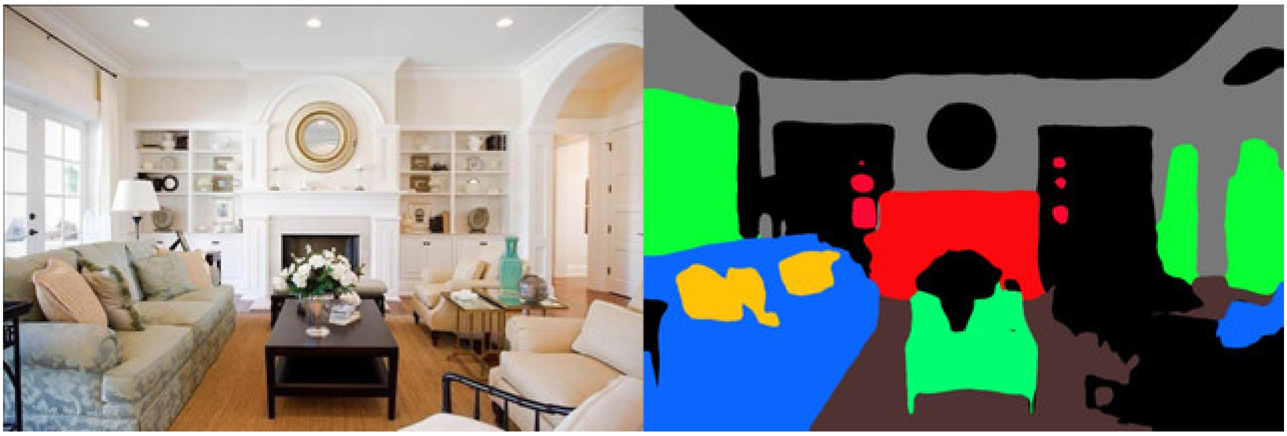
Transfer Learning for Custom Datasets in the Small-Data Regime for Basement

Milestone 4: Semantic Segmentation



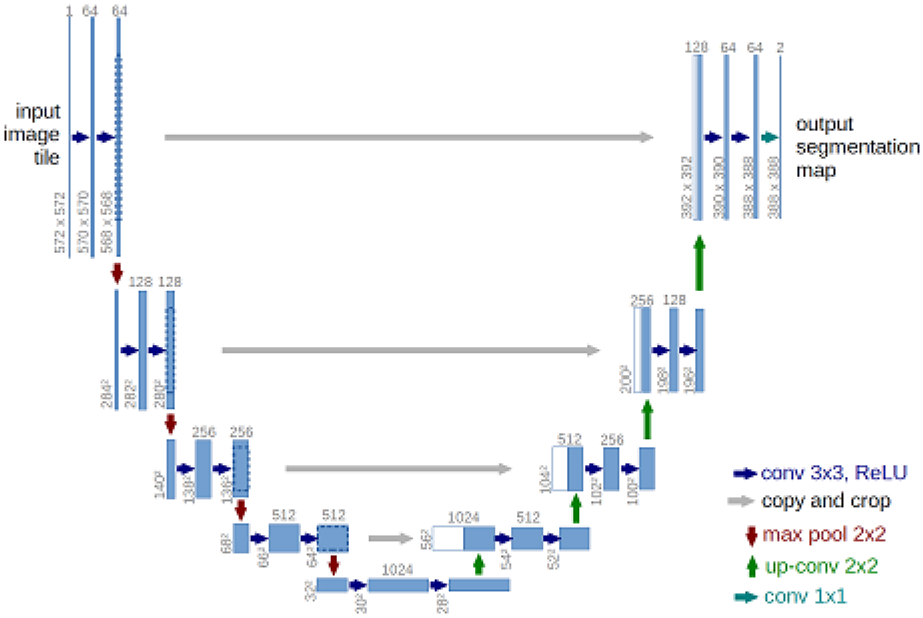
The architecture was inspired by U-Net: CNN for Biomedical Image Segmentation.

Network Description

Data

The pretrained data is from ADK20k images dataset fine tuned for our scraped images to be found here Basement Data Set

Model



This deep neural network is implemented with Keras functional API, which makes it extremely easy to experiment with different interesting architectures.

Output from the network is a 512*512 which represents mask that should be learned. Sigmoid activation function makes sure that mask pixels are in [0, 1] range.

Training

The model is trained for 5 epochs.

After 5 epochs, calculated accuracy is about 0.97.

Loss function for the training is basically just a binary crossentropy.

How to Use

Dependencies

This tutorial depends on the following libraries:

- Hardware: >=4 GPUs for training, >=1 GPU for testing (set `[--gpus GPUS]` accordingly)
- Software: Ubuntu 16.04.3 LTS, ***CUDA>=8.0, Python>=3.5, PyTorch>=0.4.0***
- Dependencies: numpy, scipy, opencv, yacs, tqdm

Also, this code should be compatible with Python versions 2.7-3.5.

1. Here is a simple demo to do inference on a single image:

```
chmod +x demo_test.sh
./demo_test.sh
```

This script downloads a trained model (UNet50dilated) and a test image, runs the test script, and saves predicted segmentation (.png) to the working directory.

2. To test on an image or a folder of images (`$PATH_IMG`), you can simply do the following:

```
python3 -u test.py --imgs $PATH_IMG --gpu $GPU --cfg $CFG
```

Results

Preformance

Architecture	Mean IoU	Pixel Accuracy(%)	Overall Score	Inference Speed(fps)
UNet50dilated	42.14	80.13	61.14	2.6
UNet18dilated	38.00	78.64	58.32	11.7

Image Output