

Assignment 2:

Deep monocular depth estimation – 30pts

1. Install and set-up a deep monocular depth estimation code.
2. Show the resulting images for the rabbit images. (10pt)

Discuss pros and cons of results compared with stereo depth images. (5pt)

3. Take an image which depth is difficult to be estimated.

Show and discuss the resulting image for the captured image. (10pt)

Describe why do you think the depth of image is difficult to be estimated. (5pt)

Submit the code (.ipynb) with resulting images and discussions above in it.

Monocular depth estimation

Train DNN with massive pairs of image + depth map.

There are many open codes, but a few codes share the weights (~ training with GPU is not necessary).

One example is

I. Alhashim and P. Wonka, "High Quality Monocular Depth Estimation via Transfer Learning," arXiv:1812.11941

<https://github.com/ialhashim/DenseDepth>

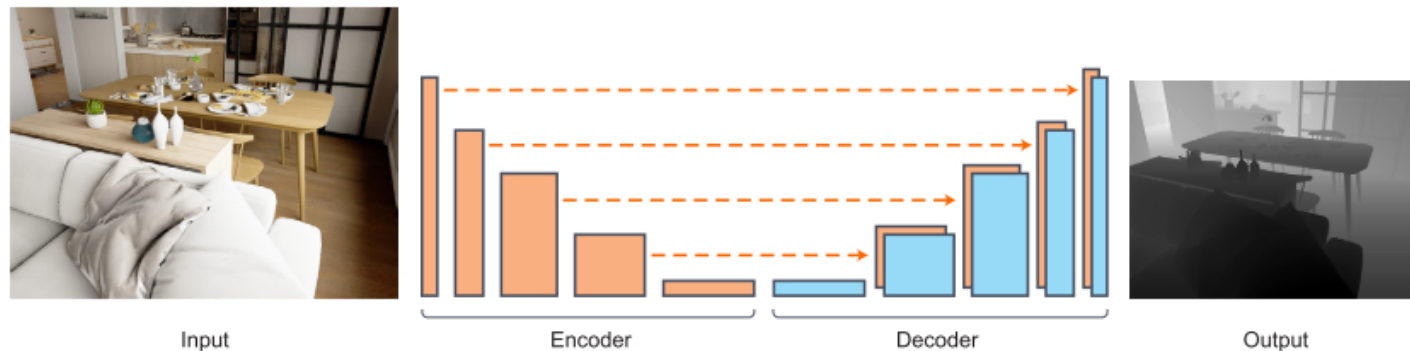


Figure 2. **Overview of our network architecture.** We employ a straightforward encoder-decoder architecture with skip connections. The encoder part is a pre-trained truncated DenseNet-169 [17] with no additional modifications. The decoder is composed of basic blocks of convolutional layers applied on the concatenation of the $2\times$ bilinear upsampling of the previous block with the block in the encoder with the same spatial size after upsampling.

Tips for <https://github.com/ialhashim/DenseDepth>

See “Demos”

(<https://github.com/ialhashim/DenseDepth/blob/master/DenseDepth.ipynb> is good example)

```
> git clone https://github.com/ialhashim/DenseDepth.git
```

```
> wget https://s3-eu-west-1.amazonaws.com/densedepth/nyu.h5 -O ./DenseDepth/nyu.h5
```

```
> python test.py --input input/*.bmp
```

(My experience:)

For Anaconda:

```
> conda install keras
```

```
> conda install tensorflow
```

```
> conda install scikit-image
```

Add code on `utils.py` (to solve OpenMP error #15):

```
import os
os.environ['KMP_DUPLICATE_LIB_OK'] = 'TRUE'
```