Monocular Depth Estimation

1. Introduction

Depth sensing is useful for tasks such as mapping, navigation, and obstacle detection, but historically it required stereo cameras or RGB-D cameras. There are now DNNs that are able to infer relative depth (aka monocular depth) from a single monocular image. See the MIT FastDepth paper for one approach to achieve this using a fully convolutional network (FCN).

The depthNet object accepts a monochrome image as input, and outputs a depth map. The depth map is colorized for visualization, but the raw depth field can also be used to directly access the depth. depthNet can be used from both Python and C++. As examples of using the depthNet class, we provide example programs in C++ and Python:

2. Mono Depth on Images

First, let's try running the depthnet example on some example images. In addition to the input/output paths, there are some additional command line options that are optional:

--network changes the network flags for the deep model being used

It is recommended to save the output images to a directory mounted under images/test. These images can then be easily viewed from the host device under jetson-inference/data/images/test

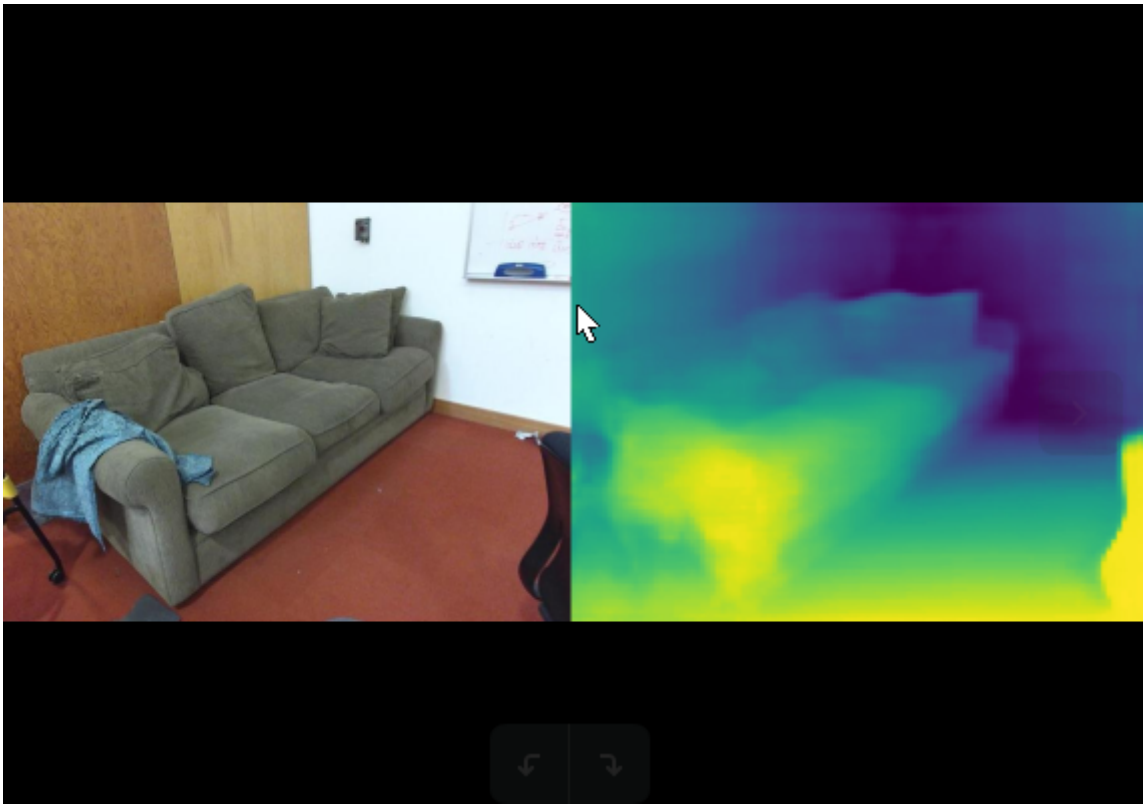
After building the project, make sure your terminal is in the aarch64/bin directory:

```
cd jetson-inference/build/aarch64/bin
```

Here are some examples of mono depth estimation for indoor scenes:

```
# C++
$ ./depthnet images/room_1.jpg images/test/depth_room_%i.jpg

# Python
$ ./depthnet.py images/room_1.jpg images/test/depth_room_%i.jpg
```



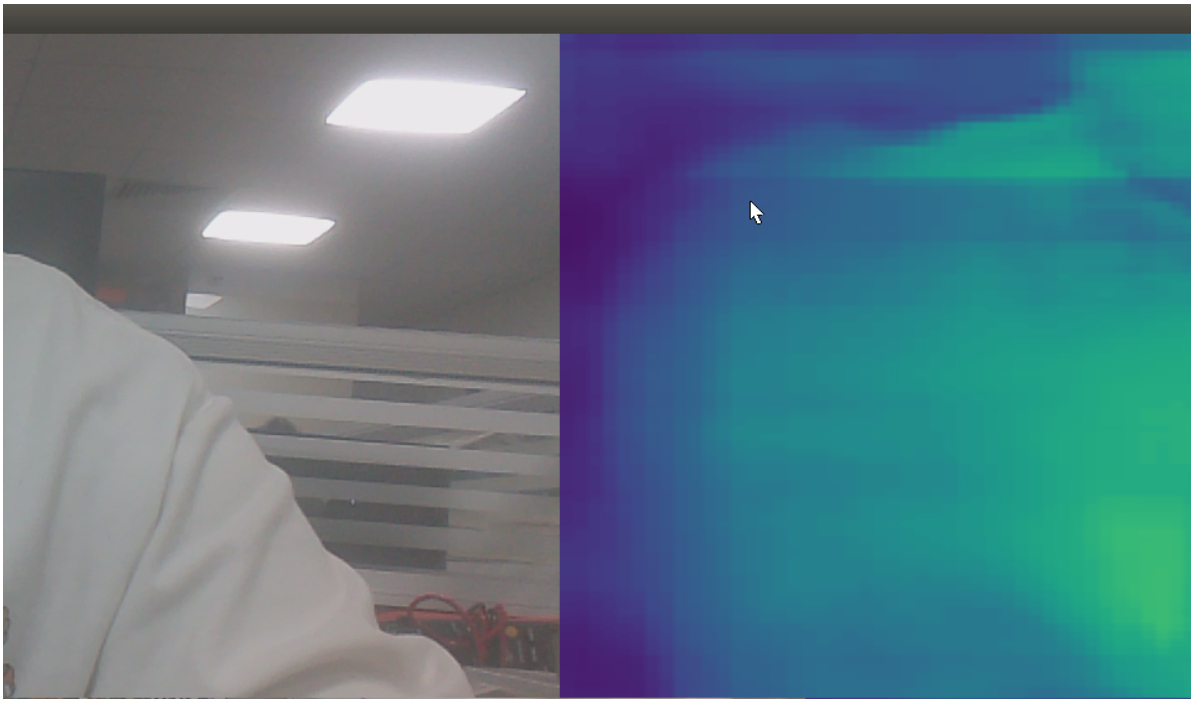
Note: The first time you run each model, TensorRT will take several minutes to optimize the network. This optimized network file is then cached to disk, so future runs using that model will load faster.

3. Video Mono Depth

To run mono depth estimation on a live camera stream or video, enter the device or file path from the Camera Stream and Multimedia page.

```
# C++
$ ./depthnet /dev/video0 # csi://0 if using MIPI CSI camera

# Python
$ ./depthnet.py /dev/video0 # csi://0 if using MIPI CSI camera
```



Note: If your screen is too small to accommodate the output, you can use --depth scale=0.5 to reduce the size of the depth image, or reduce the size of the camera, where --input width=X --input height=Y