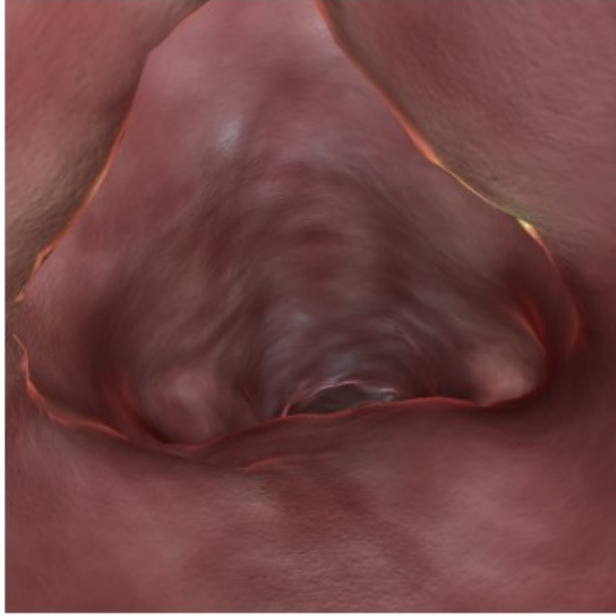
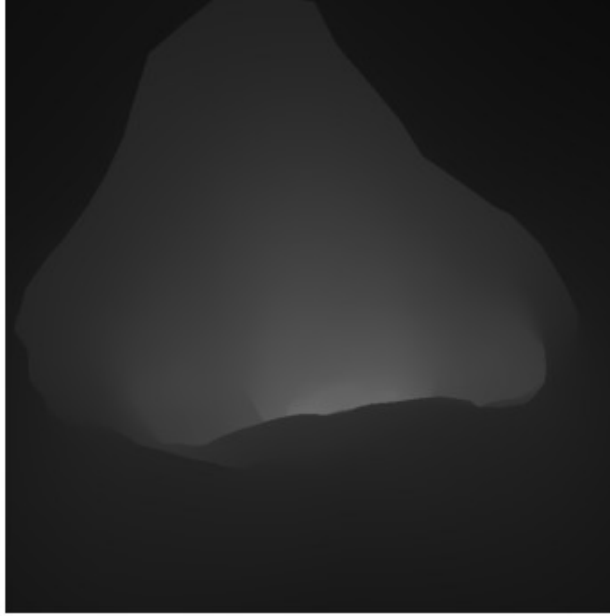


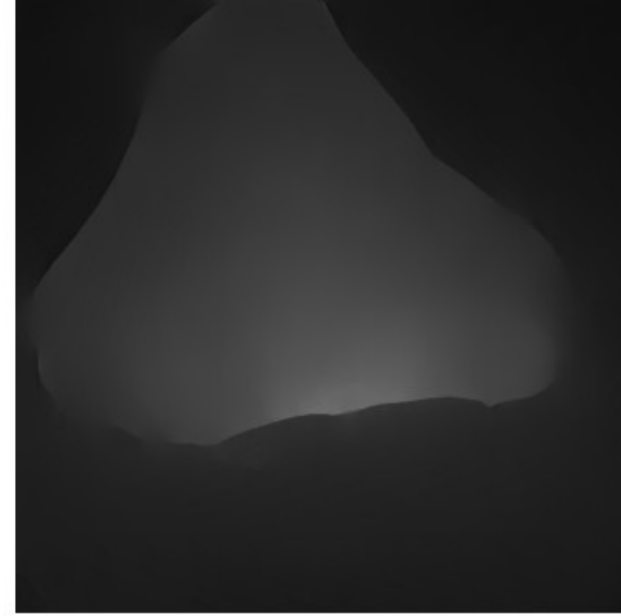
Input Image



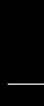
Ground Truth



Predicted Image



Pix2Pix for Monocular Depth Estimation



Background

It's a GAN

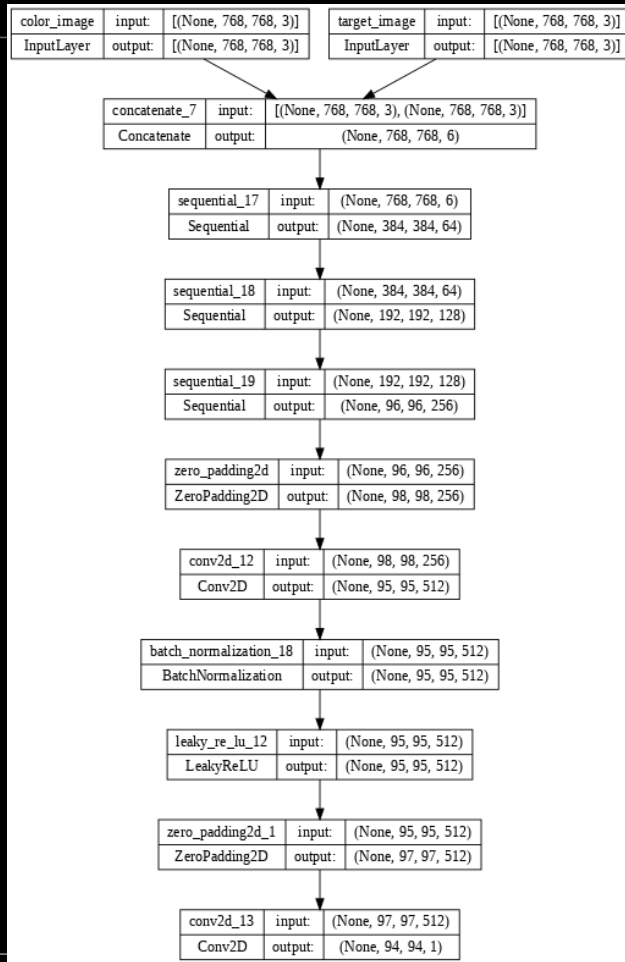
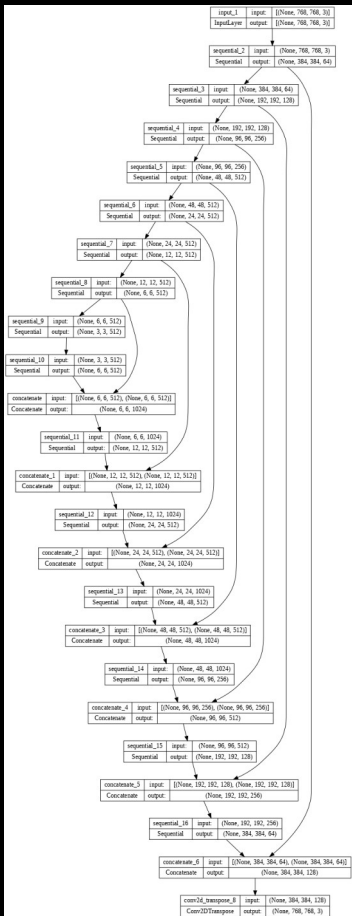
- Generator \leftrightarrow Discriminator
- Jittered input images
- Often effective with small datasets

Layers

That's a lot of layers!

- Downsampling
- Upsampling
- Skip-Connections

The values deviate from the paper's C64-C128-C256-C512 due to the image size



Checkpoints

ModelCheckpoint class

[source]

```
keras.callbacks.ModelCheckpoint(  
    filepath,  
    monitor="val_loss",  
    verbose=0,  
    save_best_only=False,  
    save_weights_only=False,  
    mode="auto",  
    save_freq="epoch",  
    initial_value_threshold=None,  
)
```

Callback to save the Keras model or model weights at some frequency.

No more lost epochs!

tf.train.Checkpoint



{}

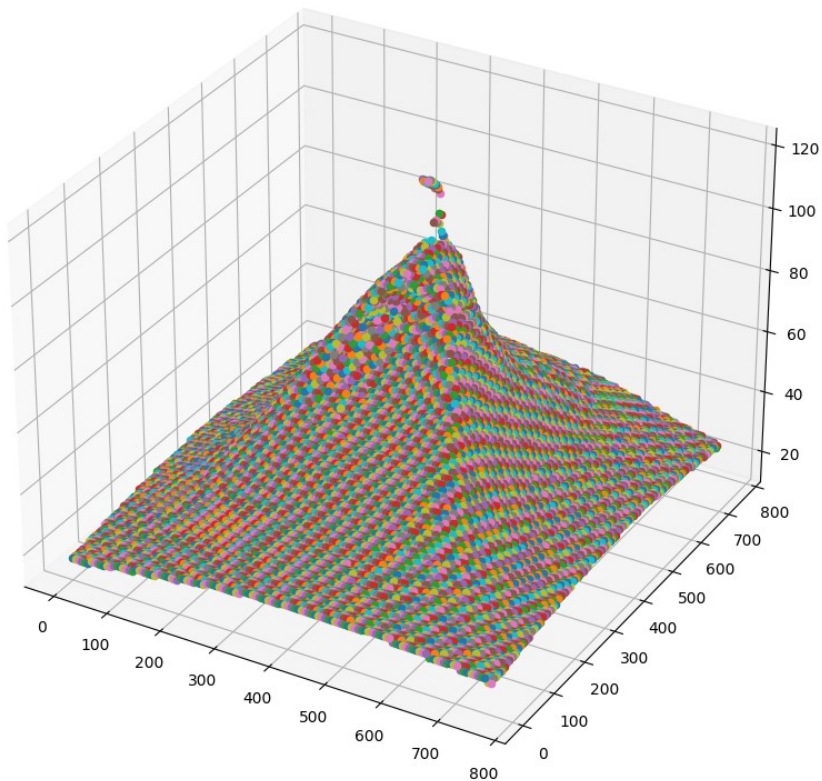


[View source on GitHub](#)

Manages saving/restoring trackable values to disk.

Rendering a Point Cloud

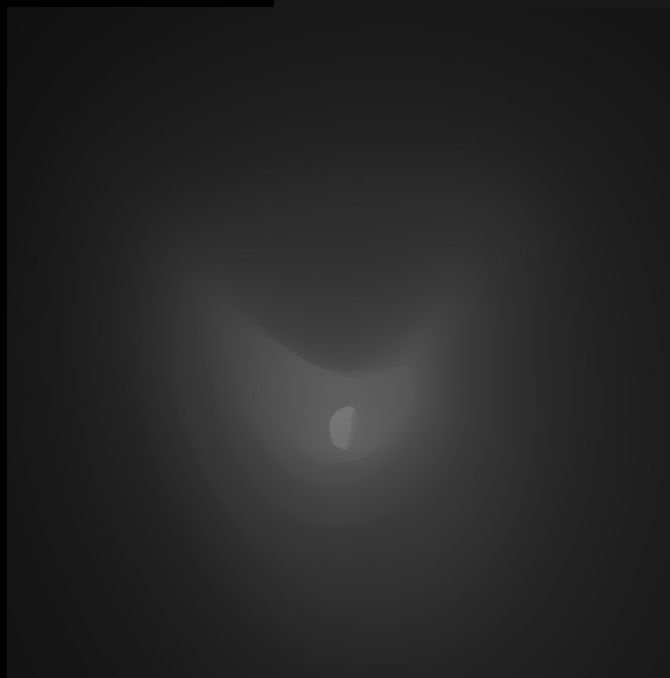
Just needs a few rotations...



```
#slicing the 3-tuple into a 2D array
darray = darray[:, :, 0]
print(darray.shape)

darray = pd.DataFrame(darray)

fig = plt.figure(figsize=(15, 10))
ax = plt.axes(projection="3d")
#Doing 768^2 operations on CPU is awful, let's do 48^2 instead
for x in range(0, 768, 6):
    for y in range(0, 768, 6):
        ax.scatter(x, y, darray[x][y])
print("Done")
```



Portable Colab Instances

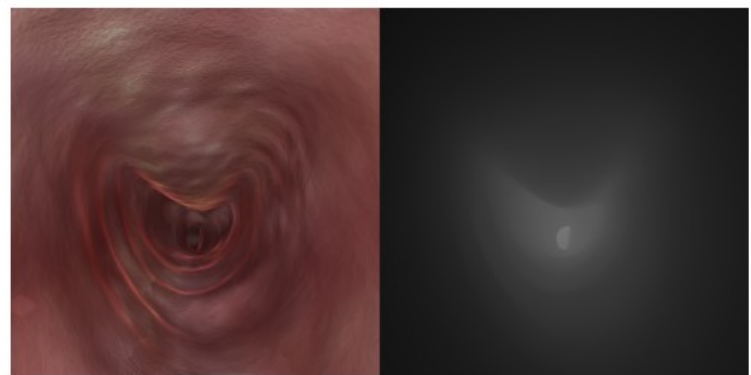
```
1 from google.colab import drive
2 drive.mount('/content/drive')
```

```
1 PATH = '/content/drive/MyDrive/Pix2Pix/Comb_Data/'
2 sample_image = tf.io.read_file(PATH + '0.png')
3 sample_image = tf.io.decode_png(sample_image)
4 print(sample_image.shape)
```

(768, 1536, 4)

```
[ ] 1 plt.figure()
    2 plt.axis('off')
    3 plt.imshow(sample_image)
    4
```

<matplotlib.image.AxesImage at 0x7b93c768f190>



Downloading my self hosted checkpoint and dataset

```
1 !wget http://daniel.win/CSCI_4800/pix2pixDWcomp.tar
2 !tar -xvzf pix2pixDWcomp.tar
```

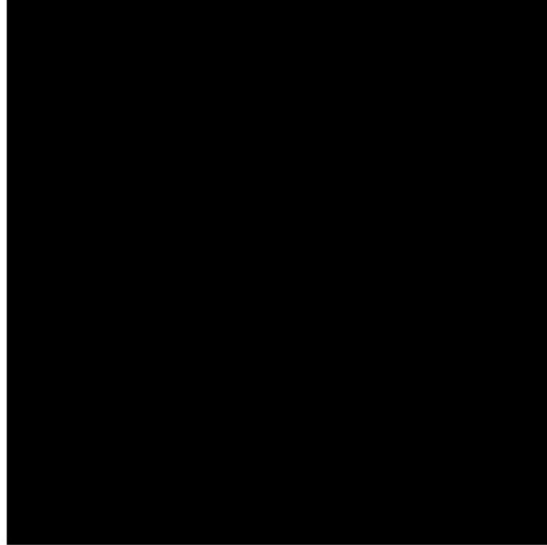
Comb_Data/22.png
Comb_Data/31.png
Comb_Data/59.png
Comb_Data/61.png

Oh.

Input Image



Ground Truth



Predicted Image



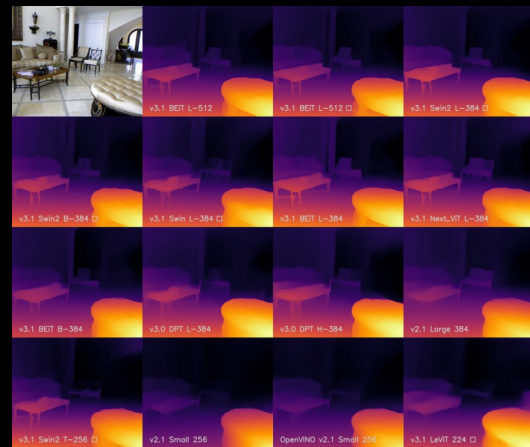
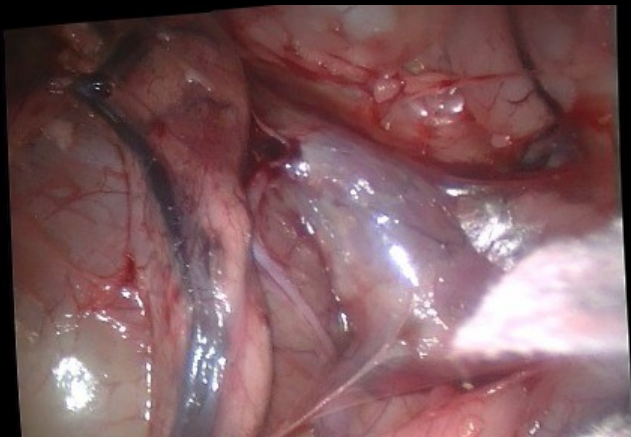
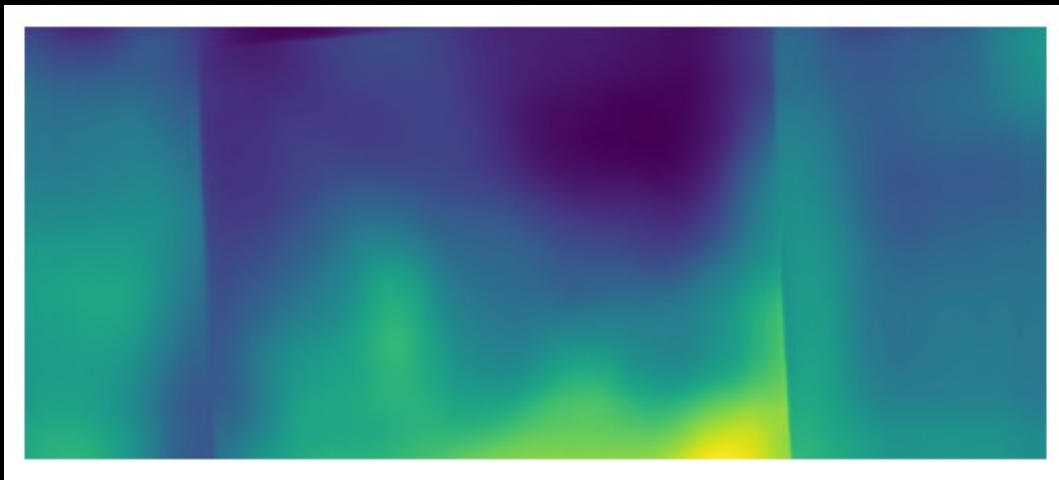
PyTorch

It seems like all the cool new depth estimation models use PyTorch.

- MiDaS (2019)
- ZoeDepth (2023)
- UniDepth (2024)

Some of these models offer metric depth estimation

MiDaS?



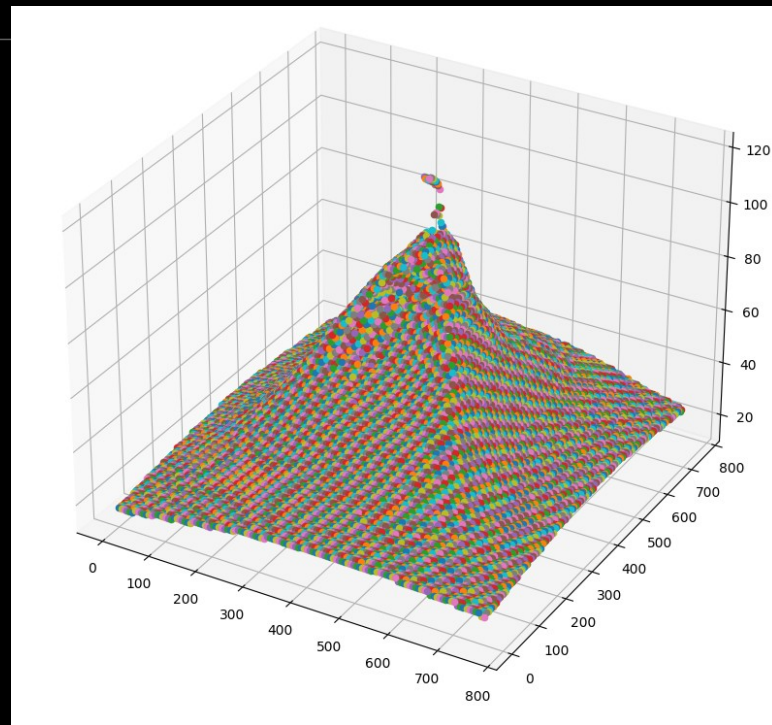
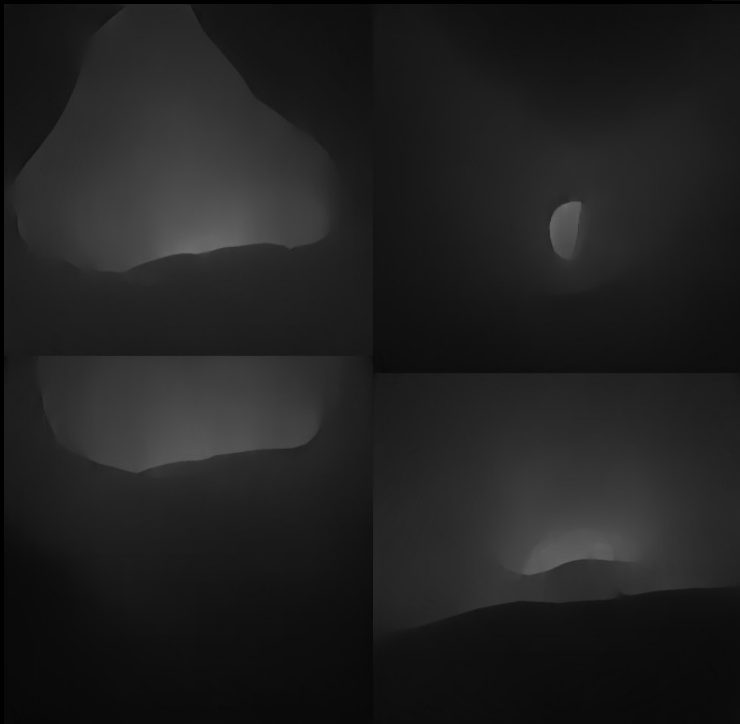
The data must be cleaned

Might be the hardest part of Phase II

- Dimensions
- Planar Distortion
- Not Phong shaded

It's not gonna be pretty.

Thank you!



Feel free to check out my Pix2Pix .ipynb at:
<http://daniel.win/pix2pix.ipynb>