**How to Use The “Data Prepper and VAE Trainer” Jupyter Notebook**

**Packages:**

If you can run the first code block with no problems, this section can be ignored.

First and foremost, OpenCV (and potentially other packages depending on your Anaconda version) may not be in the Jupyter Notebook environment by default. The easiest way I’ve found to handle this is to use the code I’ve commented out in the “imports” block (the first one). The commented out code does the following:

1. Imports sys, a package that I believe all recent Python versions have that allows for virtual environment manipulation
2. Uses sys to install opencv-python, which is OpenCV’s package for python

The line immediately following the commented out section imports cv2, which is what we need from opencv-python.

The code only needs to be run once and future running of this commented out code will cause an error along the lines of “you already have this package.” I know of no further issues this will cause, but I wouldn’t test it as environments can be tricky (also it makes doing the imports take substantially longer).

**Data Processing:**

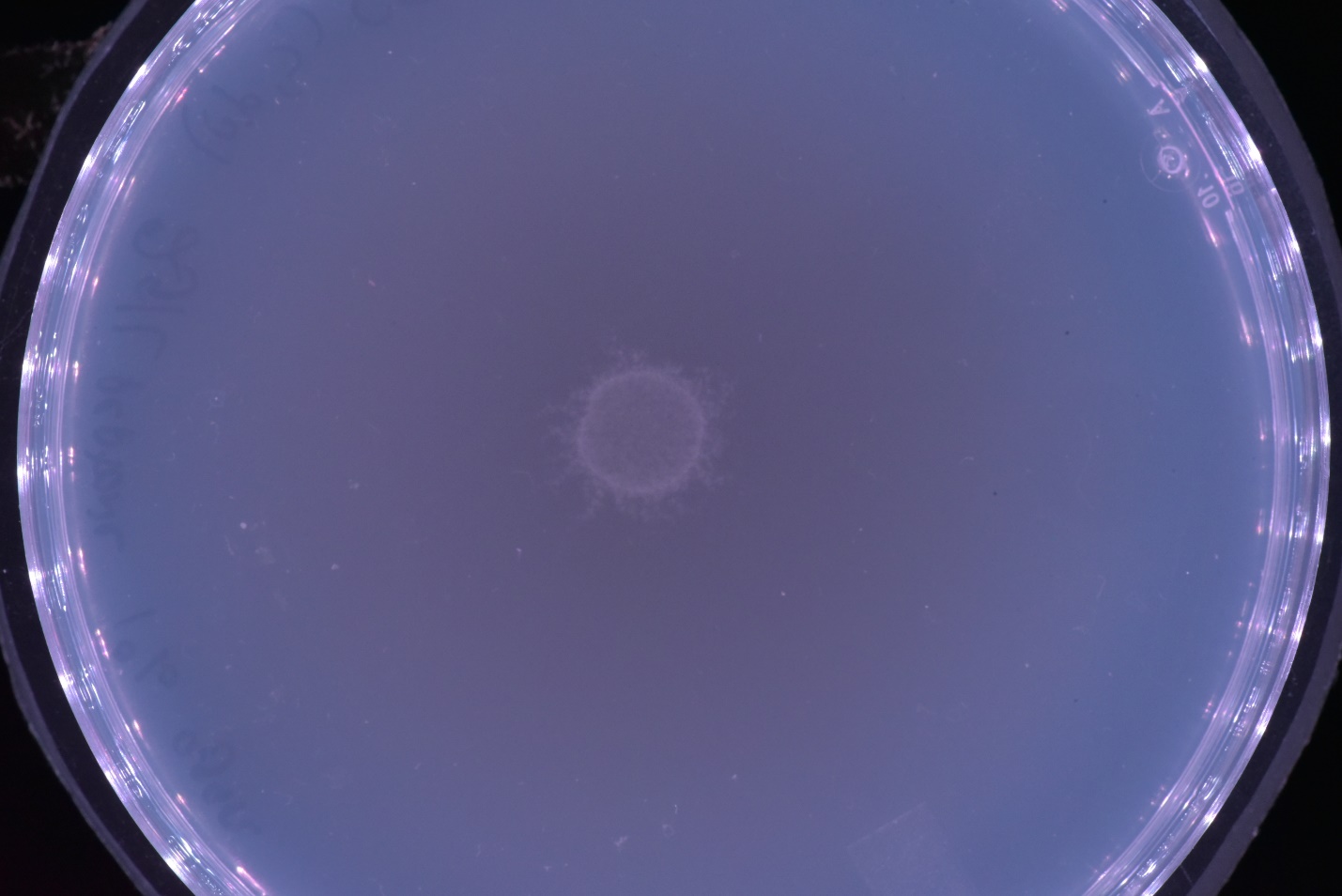
The data processor starts by changing directory to a folder called “images.” It is expected that the images folder has the following properties:

1. It is in the same directory as the notebook
2. Other directories in images (there can be other stuff in images too, and in fact we will save our processed data in this directory) are collections of .NET images (if you have saved the images in another file type, change the .NET expectation in the second code block. I recommend CTRL-F’ing to find it). Each .NET image will be a training example.
   1. Since we are prepping the data for the VAE, there are no further stipulations needed on how the images are organized in each directory. They could all be in one big directory for all this processor cares.

The processor then checks if “train\_data.z” already exists. If it does, it skips all cells relating to data processing, because it already has the processed data and the notebook assumes you just want to train the VAE. In this case, it loads the already saved train data. If you would like to make a new preprocessed version of the data, you could:

1. Delete the old preprocessed data
2. Rename the old preprocessed data
3. Move the old preprocessed data to another directory

Otherwise, the script will download all the images and reshape them into (128, 128, 1) np arrays. To ensure the preprocessing went as planned, look at the “Info on Picture Data” block and see if the results match the commented out section. This test assumes that the first image in the first directory (alphabetically) is the same as what I tested with, which is the following image:



(This is image 2021-02-27-1350-01-24-22)

(Honestly though, if it goes wrong the long printout will almost certainly be nonsense that is easily detectable by human eyes, such as all 0’s or all the same value).

Once the data is done being processed and you’ve confirmed it went correctly, some more reshaping needs to happen so the VAE can train on it.

**Making the VAE:**

This part of the code determines the architecture of the VAE. The variable “latent\_dim” controls the size of the latent vector produced by the encoder. I used 2048 exclusively, but feel free to test with larger numbers. Some architecture changes may be needed to accommodate larger dimensions, but I recommend check just changing this variable first.

**Training the VAE:**

This part trains and saves the vae. Training will take time depending on the number of samples, size of the latent dimension, batch size, and epochs. Verbose is on so keeping track of how long training will take should be clear after the first epoch. The model is saved in a folder called “models,” which needs to be made beforehand (I think) under the name “vae.” TensorFlow has a weird way of saving weights, so do not touch the odd files made when saving. **Note:** If the model is already saved, this save will overwrite the old one, so move the old model if you want to keep it.