References:

Similar code for mnist: <https://keras.io/examples/vision/conv_lstm/>

Custom dataset implementation: [Next-frame prediction with Conv-LSTM - Keras Code Examples](https://www.youtube.com/watch?v=P5yv8HDFc_M&ab_channel=HenryAILabs)

How to run:

This is an extension of the VAE model. It uses the VAE model to compress the data. This compressed data is further divided into “movies” with 20 frames each. For now we do not take the actions into account, we just want to see if the model is able to understand that the bacteria is growing.

The notebook simply runs linearly and if you have already trained the models it would be faster to load the model instead of running them every time.

Data:

The data for the RNN should be separated folders. Each folder contains a single growth of bacteria from start to finish. This code converts each folder into a growth movie to train the model.

Model:

It is a simple single layer RNN with 2048 nodes for each value in the latent vector. It is further connected to 2 dense layers. We have also added dropout between each layer to avoid overfitting and dead nodes.

Output:

| First prediction | Ground truth |
| --- | --- |
| Second prediction | 5th prediction |
| 10th prediction | 50th prediction |

As we can see in the output there isn’t much difference between predictions after the 10th prediction.

Speculation: The RNN models to the center of the latent space of the VAE. Hence whichever image type is at the center of this latent space is predicted.

We have 14 movies to train this model ie. 14 growths. The first and the last image in the growth is not very different hence it is difficult to make predictions.

Future work: This is not a very conclusive output. It would be interesting to see how this model would perform with more data which have the complete growth of bacteria from start to finish. At this moment the bacteria images are taken within a few days of time frame\* and hence we don't get the complete growth from the time it's just a circle to a full grown fractal pattern.