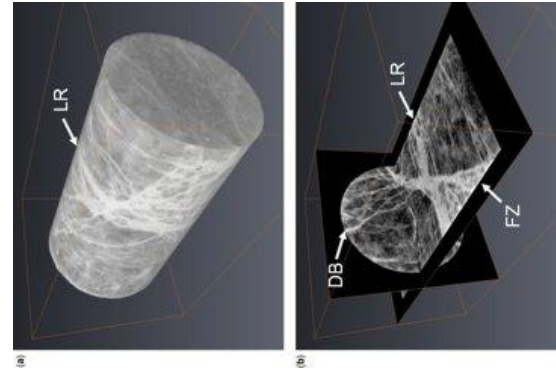# Time-Distributed Convolutional LSTM on Tomography Simulation

By: Alex Campero - Magruder High School
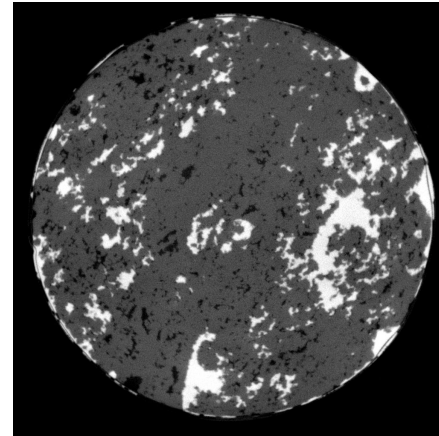
Mentor: Paul Kienzle

Geology: X-ray CT

- Tomography: imaging of an object by using waves from different angles and positions
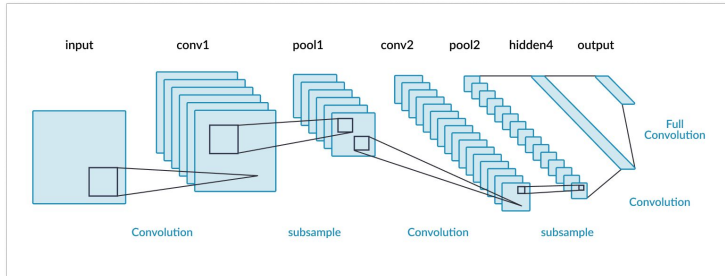

Oil and water on a rock

- Main Goal: Begin the project for frame prediction of tomographs
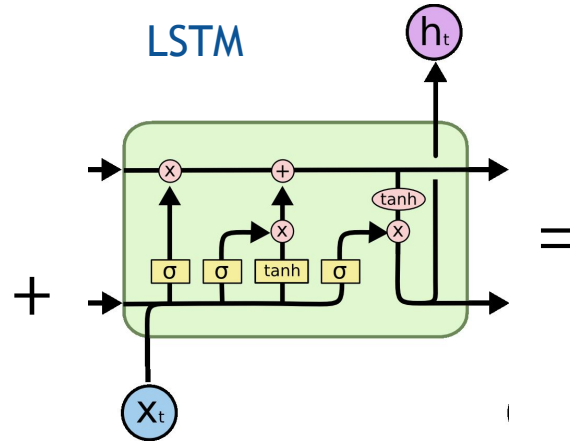- Action recognition

Neural Networks: Object Recognition

Artificial neural networks are computing systems inspired by the biological neural networks from animal brains

Convolutional Neural Network

- Used for video, and image recognition

LSTM

h_t

σ σ tanh σ

tanh

X_t

+

- Good for processing sequences of data

=

ConvLSTM

- Find the feature on each frame
- Check to see if an action is being done

Why is this important?

- Will help extract information from the
- Less time consuming than manual labor

Building a Simulation and Generator

- Simulation that simulates what the neutron tomography would look like

- never ending

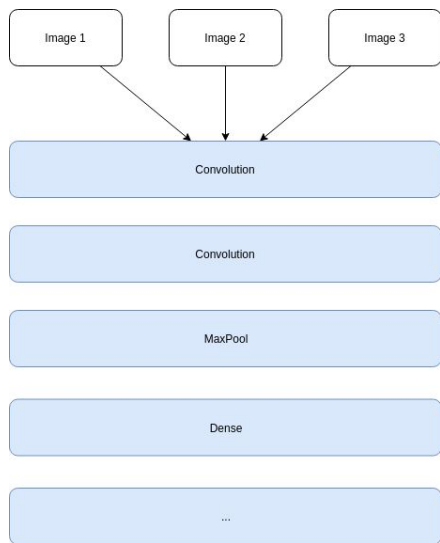- Generator used to feed frames into the Neural Network
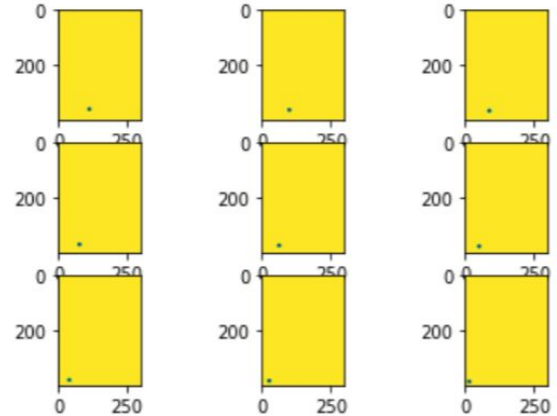
```
1 movies, labels = next_batch(3, 2)
2 for m in movies: print(id(m))
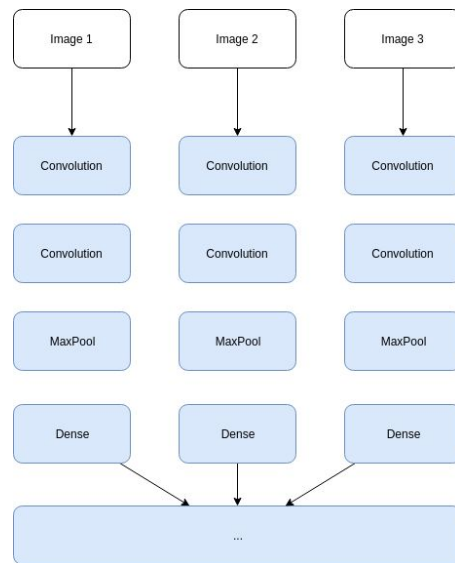```

```
140360593036432
140360593035312
140360593036432
```

Time Distributed Layer

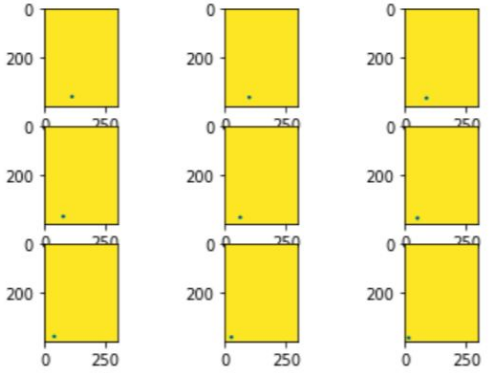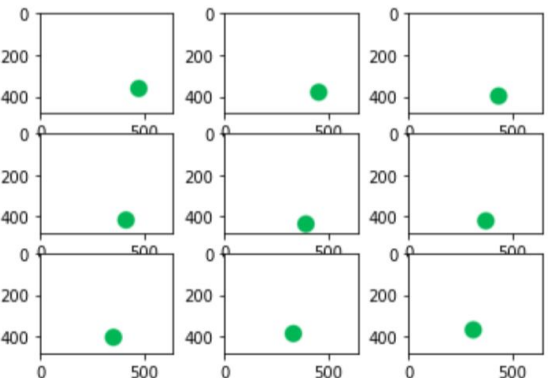Looks at every specific frame rather than combining multiple frames in a series of data

VS

- Merges images in layer

-Each frame is separated

Fitting the Generator to the Model

Issues:

Input requirements:
- Size requirements
- Channel numbers

Training the Neural Network

Issues:

- Was not learning at first
  - Changed number of layers and number of convolutions

- Batches and Epochs taking too long
  - Generate batches on GPU

```
Epoch 1/5

Epoch 00001: saving model to chkp/weights.01-0.56.hdf5
100/100 - 66s - loss: 0.7930 - val_loss: 0.5562
Epoch 2/5

Epoch 00002: saving model to chkp/weights.02-0.14.hdf5
100/100 - 66s - loss: 0.3490 - val_loss: 0.1352
Epoch 3/5

Epoch 00003: saving model to chkp/weights.03-0.56.hdf5
100/100 - 66s - loss: 0.3751 - val_loss: 0.5642
Epoch 4/5

Epoch 00004: saving model to chkp/weights.04-0.50.hdf5
100/100 - 64s - loss: 0.5102 - val_loss: 0.5011
Epoch 5/5

Epoch 00005: saving model to chkp/weights.05-0.43.hdf5
100/100 - 67s - loss: 0.5069 - val_loss: 0.4274
```

Loss: Prediction error of NN

How this will be Implemented in the Future

- Will add autoencoders, help clear up image

- Will be used for frame prediction

Questions?

Citations/acknowledgements: