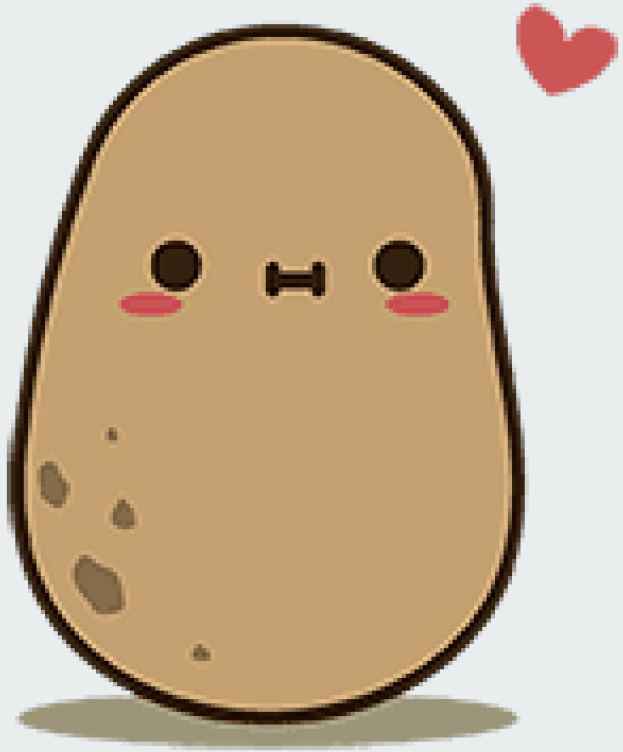
Welcome!



Inception based LSTM network for next frame prediction

Krishna Palle
U60025593

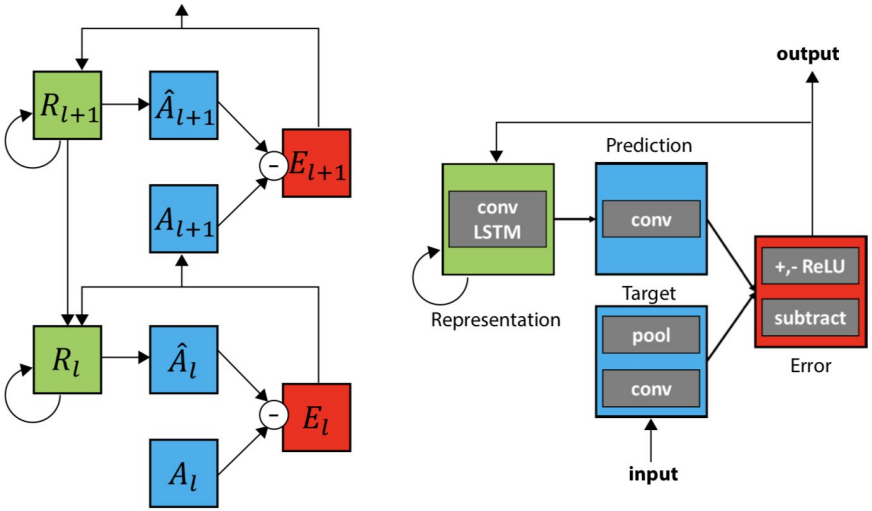
Project Definition

To implement a paper, ‘Inception-inspired LSTM for Next-frame Video Prediction’ which is an extension of another paper, ‘Deep Predictive Coding Networks for Video Prediction and Unsupervised Learning(PredNet)’ (cited 386 times) and run it against ‘The Kitti Vision benchmark Suite’. [Minimum Viable Product]

Extend the architecture to implement GAN loss and try to improve results.

PredNet

PredNet is a deep recurrent convolutional neural network that is inspired by the neuroscience concept of predictive coding (Rao and Ballard, 1999; Friston, 2005)

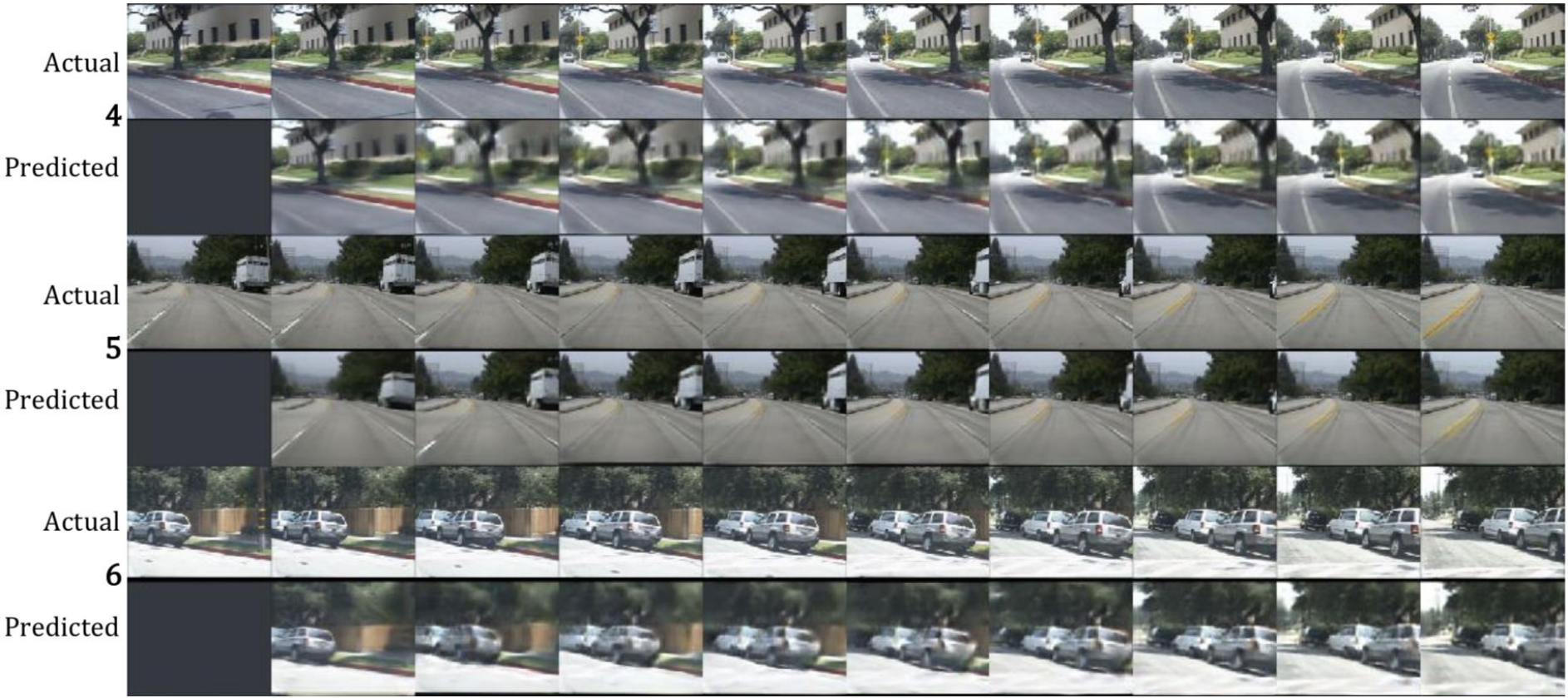


PredNet

Algorithm 1 Calculation of PredNet states

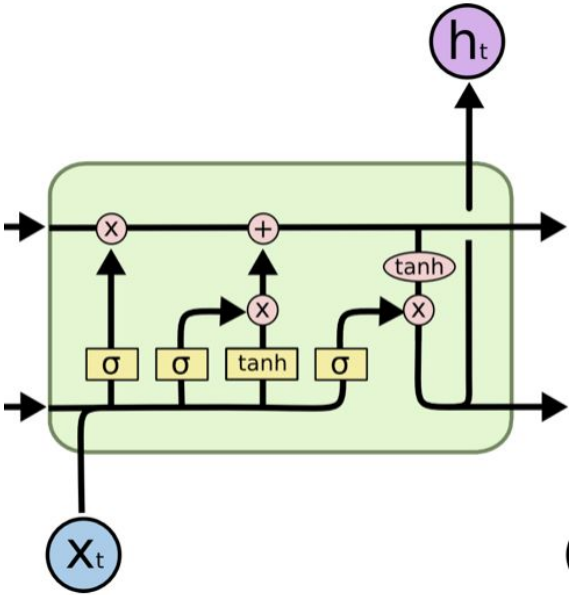
Require: x_t

$A_0^t \leftarrow x_t$
$E_l^0, R_l^0 \leftarrow 0$
for $t = 1$ **to** T **do**
  for $l = L$ **to** 0 **do** ▷ Update R_l^t states
    if $l = L$ **then**
      $R_L^t = \text{CONVLSTM}(E_L^{t-1}, R_L^{t-1})$
    else
      $R_l^t = \text{CONVLSTM}(E_l^{t-1}, R_l^{t-1}, \text{UPSAMPLE}(R_{l+1}^t))$
  for $l = 0$ **to** L **do** ▷ Update $\hat{A}_l^t, A_l^t, E_l^t$ states
    if $l = 0$ **then**
      $\hat{A}_0^t = \text{SATLU}(\text{RELU}(\text{CONV}(R_0^t)))$
    else
      $\hat{A}_l^t = \text{RELU}(\text{CONV}(R_l^t))$
    $E_l^t = [\text{RELU}(A_l^t - \hat{A}_l^t); \text{RELU}(\hat{A}_l^t - A_t^l)]$
    if $l < L$ **then**
      $A_{l+1}^t = \text{MAXPOOL}(\text{CONV}(E_t^l))$

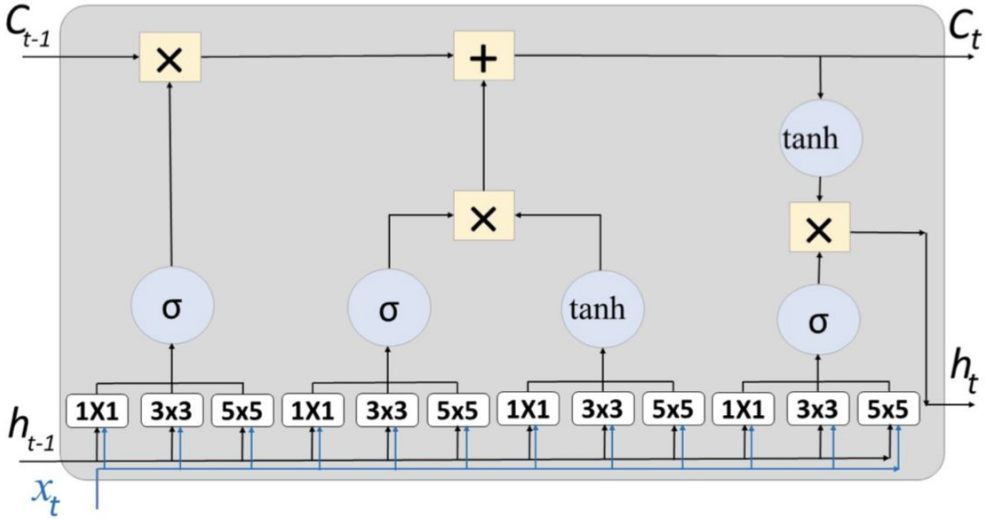

Actual
4
Predicted
Actual
5
Predicted
Actual
6
Predicted

Inception based LSTM network

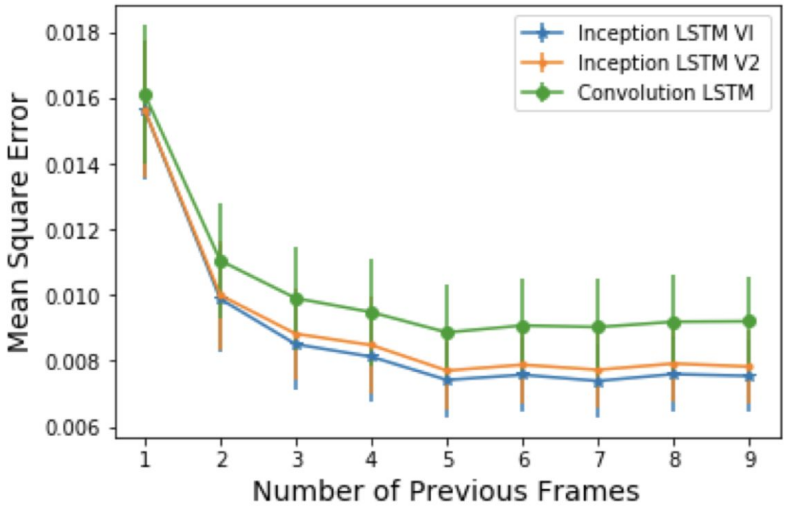
The general idea of inception networks is to implement wider networks instead of deeper networks.



VS



Inception based LSTM network



Implementation - PredNet

```
for i in range(self.layer_size):
	for c in ['i', 'f', 'c', 'o']:
		act = 'tanh' if c == 'c' else 'sigmoid'
		self.conv_layers[c].append(Conv2D(self.channels_r[i], self.glob_filter_size, activation=act))

	self.conv_layers['ahat'].append(Conv2D(self.channels_a[i], self.glob_filter_size, activation='relu'))

for i in range(1, self.layer_size):
	self.conv_layers['a'].append(Conv2D(self.channels_a[i], self.glob_filter_size, activation='relu')

self.upsample = UpSampling2D(data_format=self.data_format)
self.pool = MaxPooling2D(data_format=self.data_format)
```

Implementation - Inception inspired LSTM

```
for i in range(self.layer_size):
      for c in ['i1', 'f1', 'c1', 'o1']:
            act = 'tanh' if c == 'c' else 'hard_sigmoid'
            self.conv_layers[c].append(Conv2D(self.channels_r[i], 1, padding='same', activation=act)
      for c in ['i3', 'f3', 'c3', 'o3']:
            act = 'tanh' if c == 'c' else 'hard_sigmoid'
            self.conv_layers[c].append(Conv2D(self.channels_r[i], 3, padding='same', activation=act)
      for c in ['i5', 'f5', 'c5', 'o5']:
            act = 'tanh' if c == 'c' else 'hard_sigmoid'
            self.conv_layers[c].append(Conv2D(self.channels_r[i], 5, padding='same', activation=act)

      self.conv_layers['ahat'].append(Conv2D(self.channels_a[i], self.glob_filter_size, activation='relu')
for i in range(1, self.layer_size):
      self.conv_layers['a'].append(Conv2D(self.channels_a[i], self.glob_filter_size, activation='relu')
self.upsample = UpSampling2D(data_format=self.data_format)
self.pool = MaxPooling2D(data_format=self.data_format)
```

Parameters

Inputs = [3x128x160] (3 Channels RGB)

channels in Prediction Layer = [3, 48, 96, 192]

channels in Representation Layer = [3, 48, 96, 192]

Global_filter_size = 3

Loss function = mean_absolute_error

Learning rate = 0.001

number_of_epochs = 1

batch_size = 4

samples_per_epoch = 4

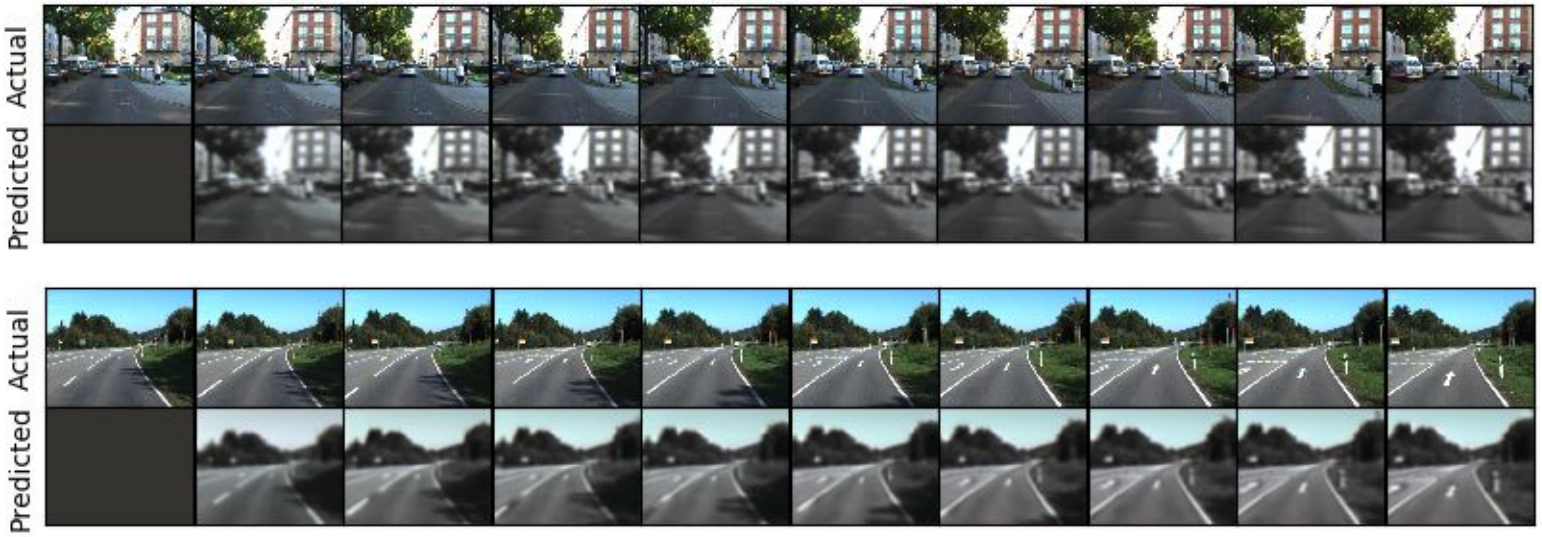
optimiser = adam

Model MSE: 0.016224
Previous Frame MSE: 0.021246

Results - PredNet

Model MSE: 0.018290
Previous Frame MSE: 0.021246

Results - Inception Inspired LSTM Network



Possible extension?

The produced results are blurry since the loss function used to optimise the results was ‘mean_absolute_error’

As discussed in the course, GAN loss results in more sharper images.

Implement GAN architecture, by using the existing architecture as a Generator and implement a simple discriminator with some Convolution LSTM layers and experiment with multiple GAN losses to see if that improves any result.

Things learnt.

- Predictive Coding Concepts
- Convolution LSTMs
- Inception inspired architectures
- Video data Analysis
- Implementing a model in Keras with forward function
- Understanding the bottlenecks in training models
- Fetching computing resources on SCC (That was hard!)

Questions?