

Forecasting Morphogenesis in Latent and Image Space

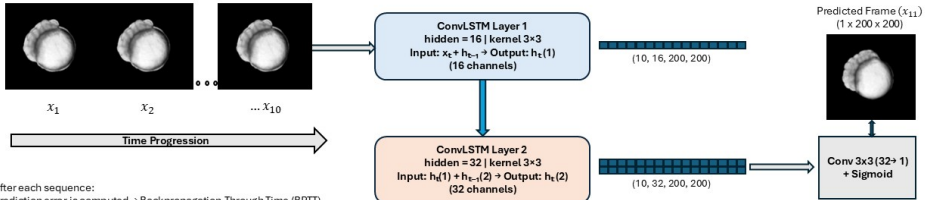
Alejandro Cobo
Angel David Rodriguez
Prof: Andres Morales-Navarrete, PhD

Yachay Tech University

November 6, 2025

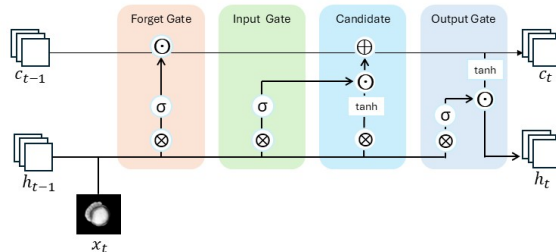
Convolutional LSTM

Training Flow: 450 frames \rightarrow 440 overlapping sequences (10 input frames \rightarrow 1 predicted frame)
Each sequence is processed through the ConvLSTM to forecast the next frame.



After each sequence:
Prediction error is computed \rightarrow Backpropagation Through Time (BPTT)
 \rightarrow Model parameters are updated \rightarrow Process repeats for all 440 sequences.

ConvLSTM Layer 1:



$\sigma \rightarrow$ outputs $\in (0, 1) \rightarrow$ probability gate
 $\tanh(x) \rightarrow$ outputs $\in (-1, 1) \rightarrow$ signed information

x_t : (1, 200, 200)
 h_{t-1} : (16, 200, 200)
 \downarrow Concatenate \rightarrow (17, 200, 200)

Conv2D 3 \times 3
 \rightarrow Output: (64, 200, 200) # 4 \times 16 feature maps
 \downarrow Split by gates

Forget Gate σ : (16, 200, 200)
Input Gate σ : (16, 200, 200)
Candidate \tanh : (16, 200, 200)
Output Gate σ : (16, 200, 200)

Cell State c_t : (16, 200, 200)
Hidden State h_t : (16, 200, 200)

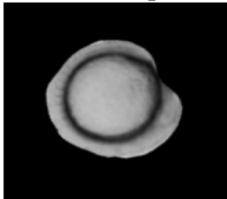
Pseudocode

```
1 # 1) Data -> Sequences
2 read_tiff_stacks()                                # each: (450, 200, 200), grayscale
3 normalize_to_[0,1]()
4 TARGET = "Control1"
5
6 # Sliding window: past 10 -> predict next 1
7 (X, y) <- build_sequences(stack[TARGET], N_PAST=10, N_FUTURE=1)
8 split 80/20 -> (X_train, y_train), (X_val, y_val)
9 reshape:
10 X_* -> (B, T=10, C=1, H=200, W=200)
11 y_* -> (B, 1, 1, 200, 200)
12
13 # 2) Model (Forecaster)
14 ConvLSTM layer 1: input (1+16)=17 ch -> hidden=16, kernel 3x3 (same)
15 ConvLSTM layer 2: input (16+32)=48 ch -> hidden=32, kernel 3x3 (same)
16 Final Conv 3x3: 32 -> 1, then sigmoid
17 y' = model(X) uses last hidden state h_T to predict next frame
18
19 # 3) Loss = SSIM
20 loss_fn(y', y) = 1 - SSIM(y', y; data_range=1.0)
21
22 # 4) Training (BPTT)
23 for epoch in 1..E:
24   for (Xb, yb) in train_loader:
25     yb' = model(Xb)                                # processes T=10 time steps
26     loss = 1 - SSIM(yb', yb)
27     loss.backward(); optimizer.step()
28
29   # Validation
30   model.eval()
31   val_loss = mean( 1 - SSIM(model(Xv), yv) over val_loader )
32   print(epoch, train_loss_avg, val_loss)
33
34 # 5) Plotting
35 take one validation batch: (Xb, yb)
36 predict next frames: yb' = model(Xb)
37
38 last_input = Xb[i, t=last]                        # last frame of the 10 inputs
39 ground_truth = yb[i]                              # real next frame
40 prediction = yb'[i]                               # predicted next frame
41 error_map = |prediction - ground_truth|
```

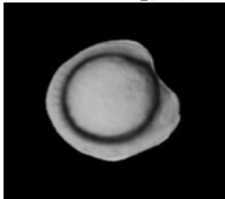
Results

Validation Sequence 2 | Input Frame #362 → Target Frame #363 (of 450)

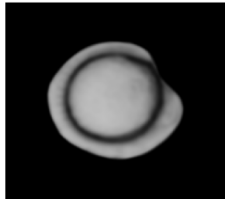
Last Input (x_t)



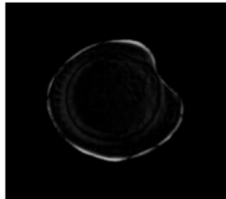
Ground Truth (x_{t+1})



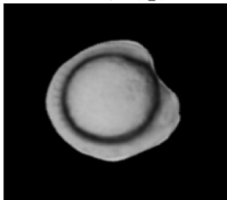
Prediction



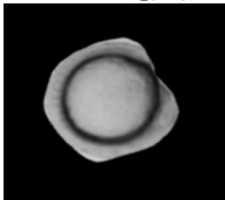
|Error|



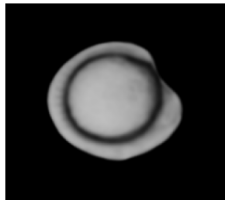
Last Input (x_t)



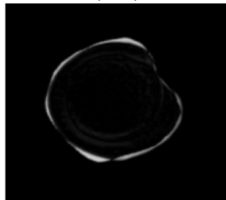
Ground Truth (x_{t+1})



Prediction



|Error|



Starting training...

Device: cuda

Epochs: 20

Epoch [01/20]	-	Time: 21.43s	-	Train Loss: 0.24090355	-	Val Loss: 0.10868752
Epoch [02/20]	-	Time: 20.99s	-	Train Loss: 0.08105249	-	Val Loss: 0.10245259
Epoch [03/20]	-	Time: 21.51s	-	Train Loss: 0.07899994	-	Val Loss: 0.10324820
Epoch [04/20]	-	Time: 21.97s	-	Train Loss: 0.07917218	-	Val Loss: 0.10193101
Epoch [05/20]	-	Time: 21.84s	-	Train Loss: 0.07819608	-	Val Loss: 0.10136267
Epoch [06/20]	-	Time: 21.90s	-	Train Loss: 0.07835551	-	Val Loss: 0.10211914
Epoch [07/20]	-	Time: 22.02s	-	Train Loss: 0.07790385	-	Val Loss: 0.10195548
Epoch [08/20]	-	Time: 21.98s	-	Train Loss: 0.07736144	-	Val Loss: 0.10197099
Epoch [09/20]	-	Time: 21.69s	-	Train Loss: 0.07743451	-	Val Loss: 0.10166430
Epoch [10/20]	-	Time: 21.74s	-	Train Loss: 0.07702232	-	Val Loss: 0.10078235
Epoch [11/20]	-	Time: 21.94s	-	Train Loss: 0.07698763	-	Val Loss: 0.10151176
Epoch [12/20]	-	Time: 21.66s	-	Train Loss: 0.07662545	-	Val Loss: 0.10044155
Epoch [13/20]	-	Time: 21.74s	-	Train Loss: 0.07643133	-	Val Loss: 0.09992271
Epoch [14/20]	-	Time: 21.33s	-	Train Loss: 0.07613668	-	Val Loss: 0.10021412
Epoch [15/20]	-	Time: 21.36s	-	Train Loss: 0.07623427	-	Val Loss: 0.10264324
Epoch [16/20]	-	Time: 21.25s	-	Train Loss: 0.07631691	-	Val Loss: 0.10134257
Epoch [17/20]	-	Time: 21.34s	-	Train Loss: 0.07572130	-	Val Loss: 0.09936752
Epoch [18/20]	-	Time: 21.27s	-	Train Loss: 0.07580627	-	Val Loss: 0.10166560
Epoch [19/20]	-	Time: 21.10s	-	Train Loss: 0.07589438	-	Val Loss: 0.10023416
Epoch [20/20]	-	Time: 21.02s	-	Train Loss: 0.07573581	-	Val Loss: 0.09998060