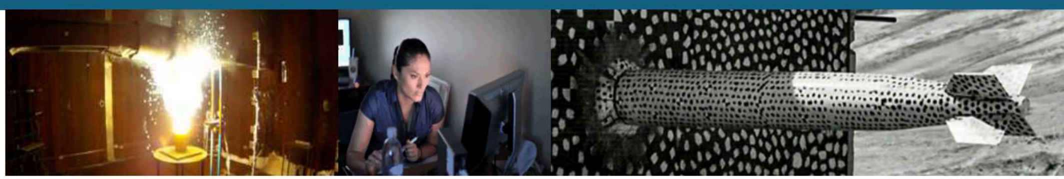
SAND2019-13677PE

Spatio-Temporal Anomaly Detection in Video



Presented by: Michael R. Smith, Joshua Rutkowski

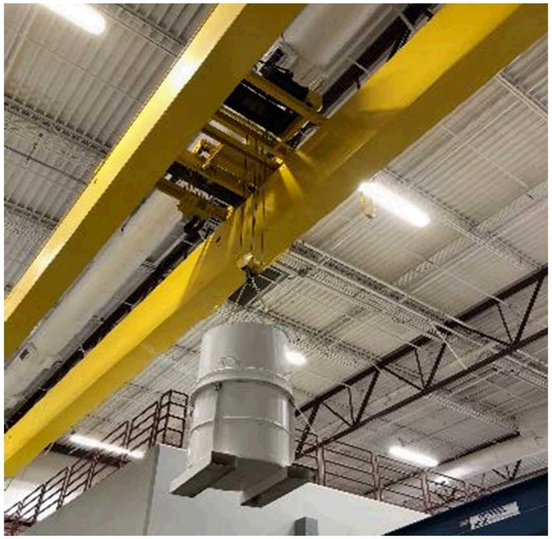
SNL team (not present): Michael Hamel,

David Hannasch and Marcellus Smith (intern)

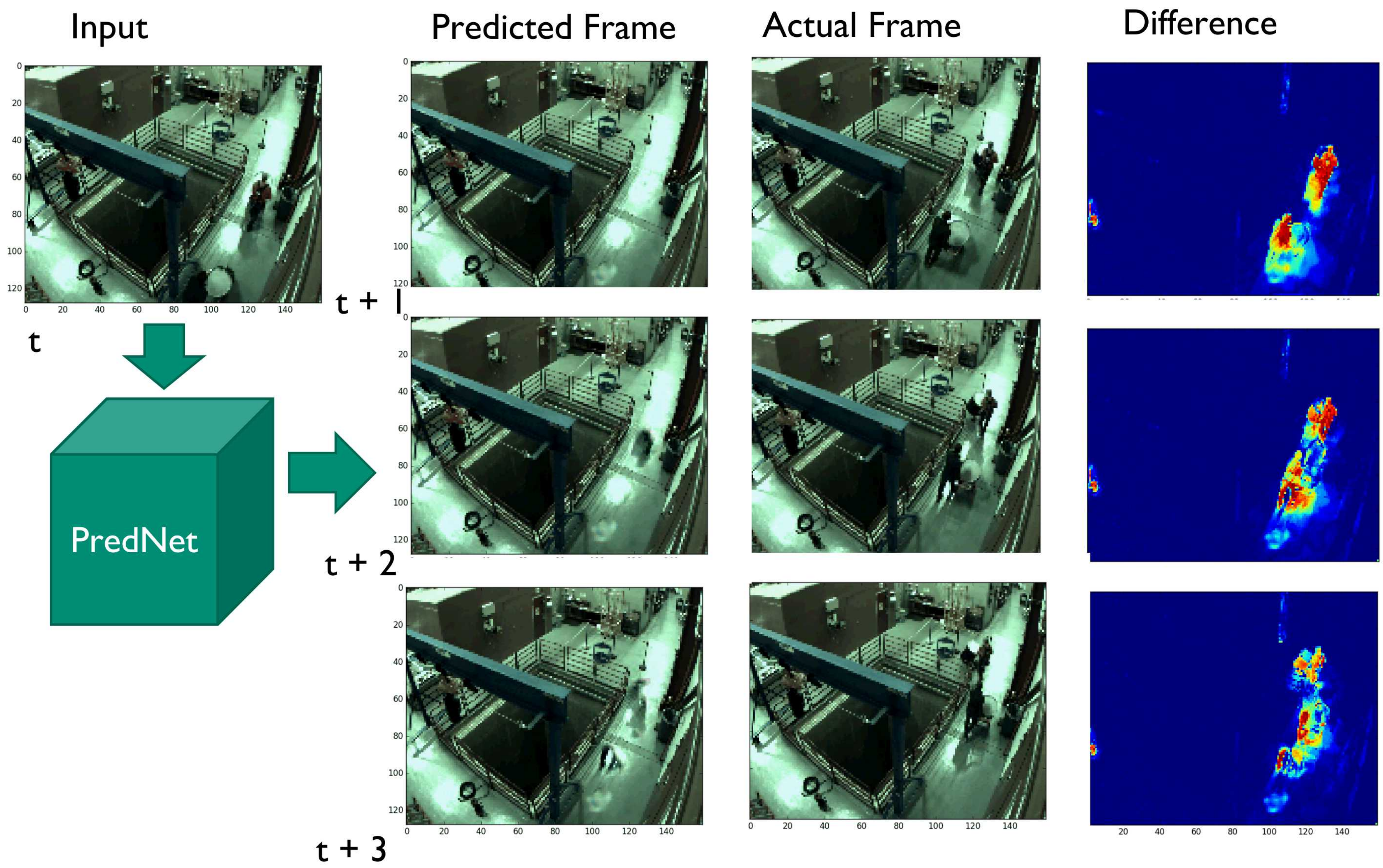
Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Problem Space and Use Case

- Review of NGSS surveillance data by IAEA inspectors is **mundane and tedious**
  - Look for anomalous activity (**unknown unknowns**)
  - **Frame by Frame**
- Common monitored activity is transfer of spent fuel to storage and transportation casks
- Assumptions:
  - No labelled training data (cannot enumerate all anomalies)
  - Data cannot leave facility
  - Non ML expert users
  - Environments and processes change significantly across facilities

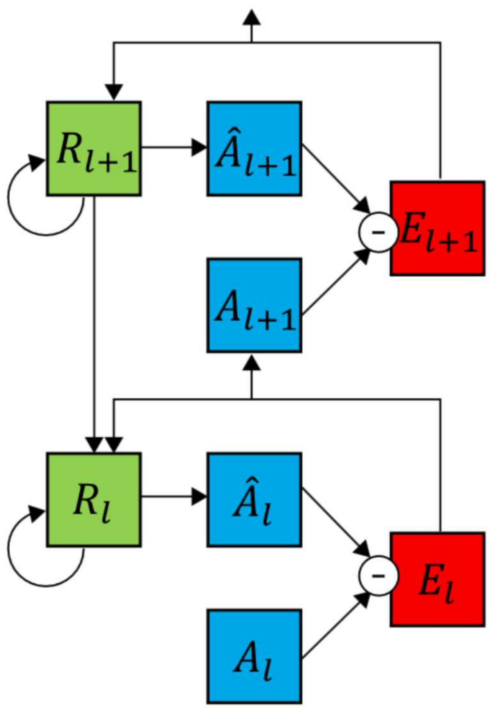


Solution: Deep Predictive Coding Networks for Video Prediction and Unsupervised Learning (PredNet)

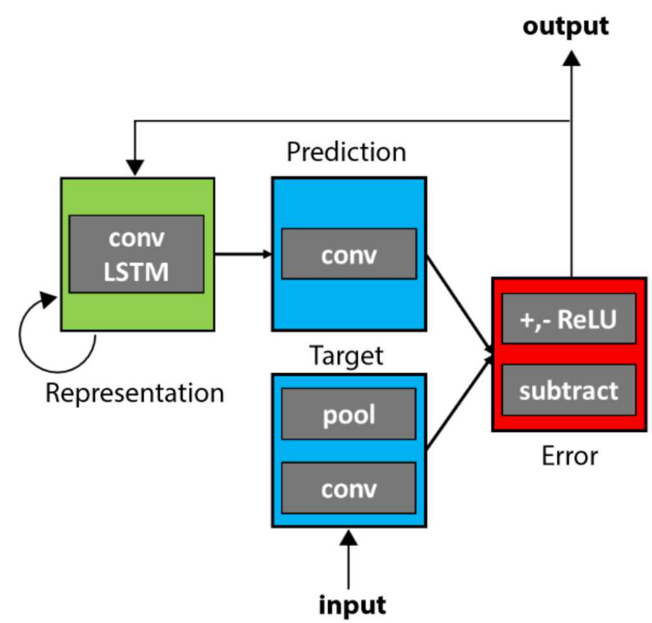


PredNet Architecture

- Each layer in PredNet consists of:
  - R_l: representation neurons
  - $\hat{A}_l$: layer-specific predictions at each time step
  - A_l: layer-specific target
  - E_l: layer-specific error term



- Information flow within 2 layers

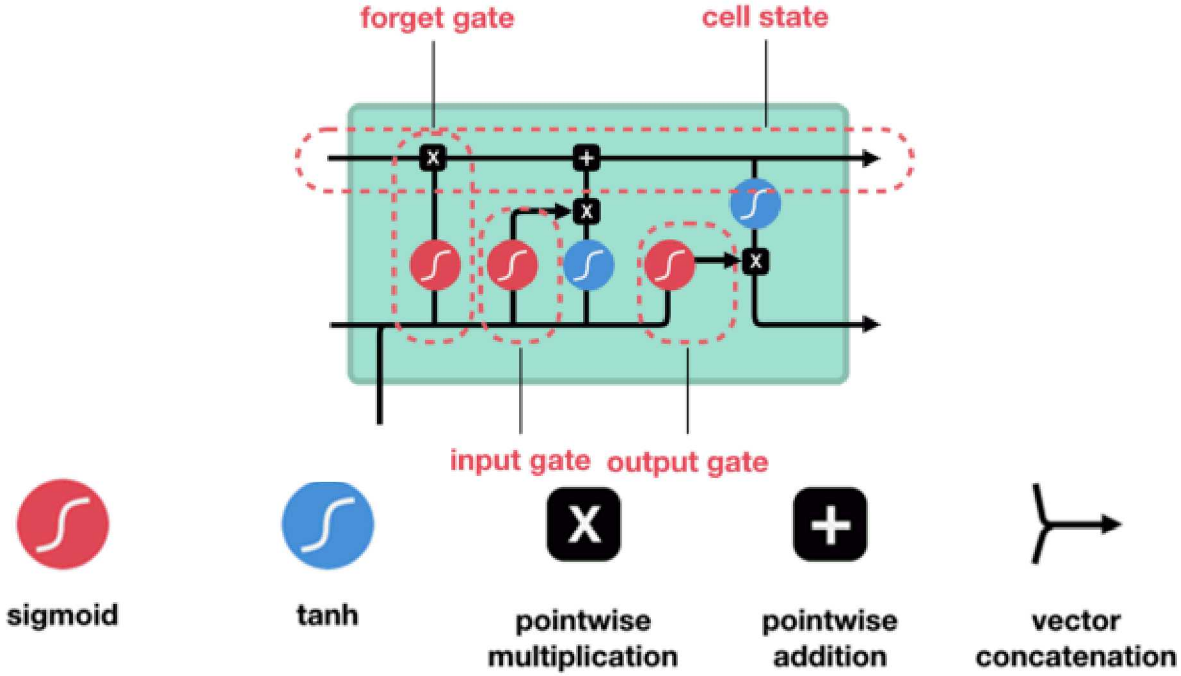


- Module operations

Long Short Term Memory (LSTM)

Hidden state from previous time step is passed in to the neuron

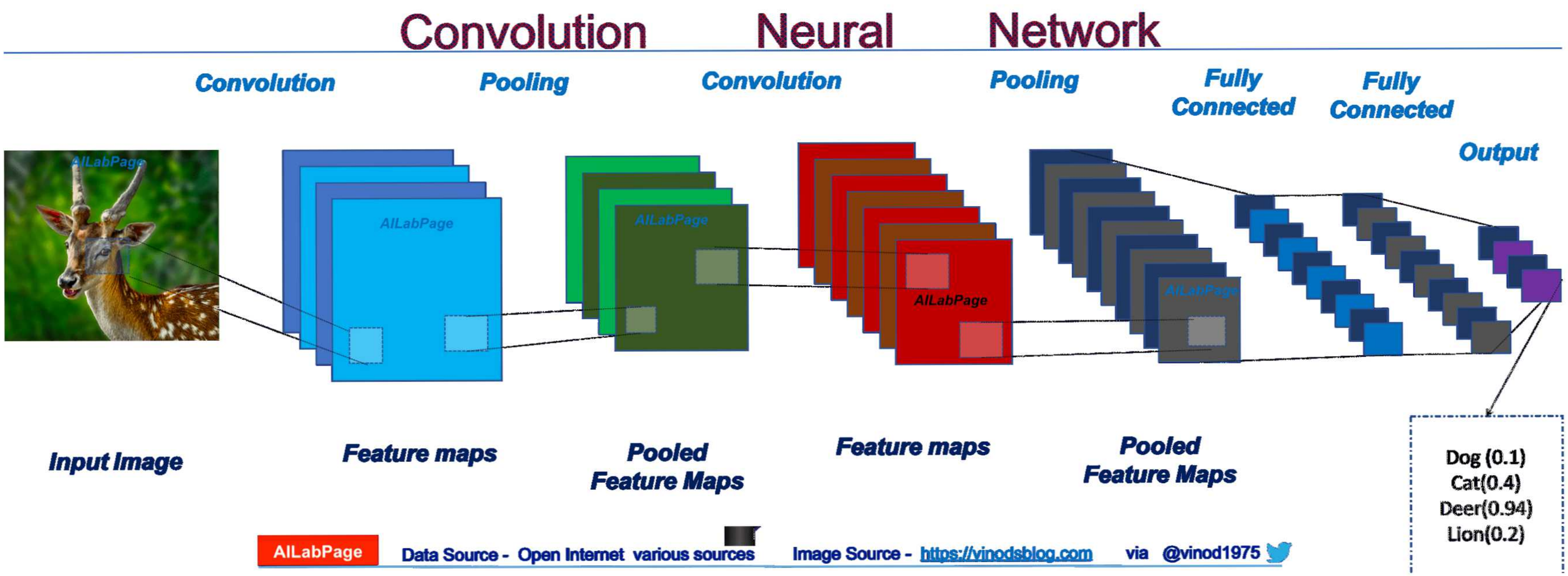
- Allows state to be built up
- The neuron can remember previous inputs
- Maintains several states/gates
  - Forget gate: What is relevant from prior steps
  - Input gate: Which inputs are relevant in the current step
  - Cell state: Combine output from input gate and forget gate to get new cell state
  - Output gate: Computes what the hidden state should be



Convolution Neural Network (CNN)

Best approach for working with images

- Each layer acts a set of filters extracting important features
- Generally, after passing through several convolutional layers, the output passed through a fully connected dense network

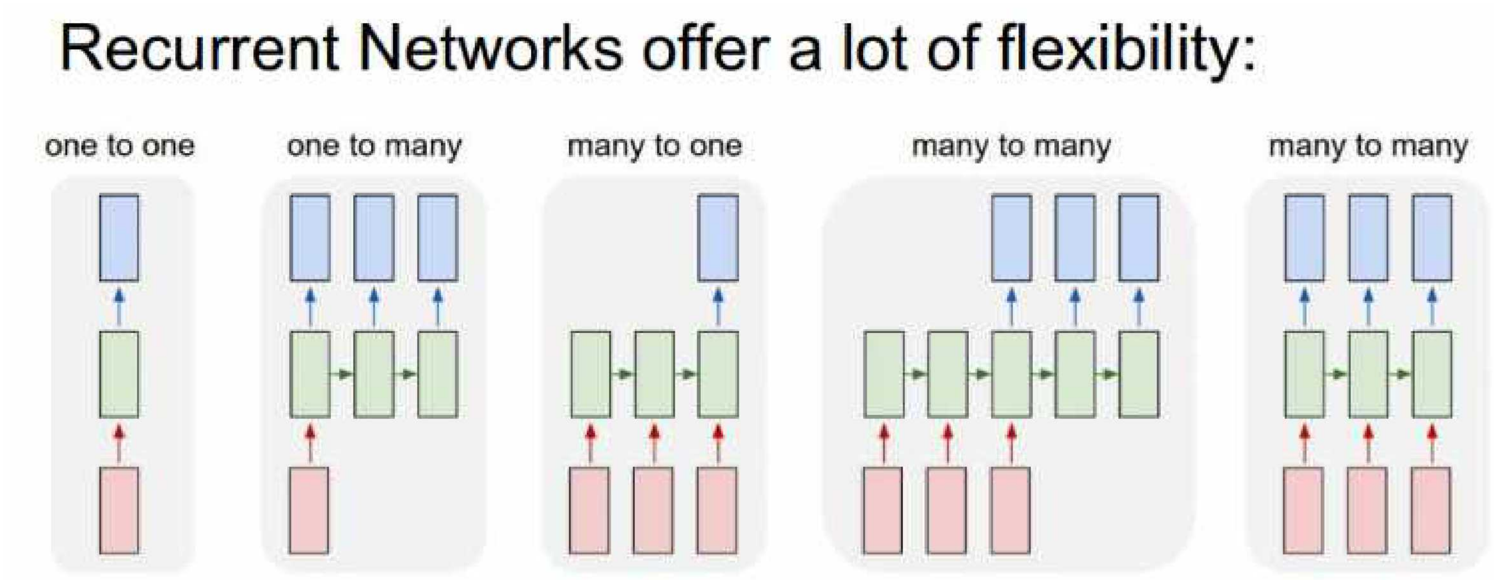


Sequence-to-sequence prediction

Many problems involving sequences and predicting sequences:
- Machine translation
- Question and answering systems

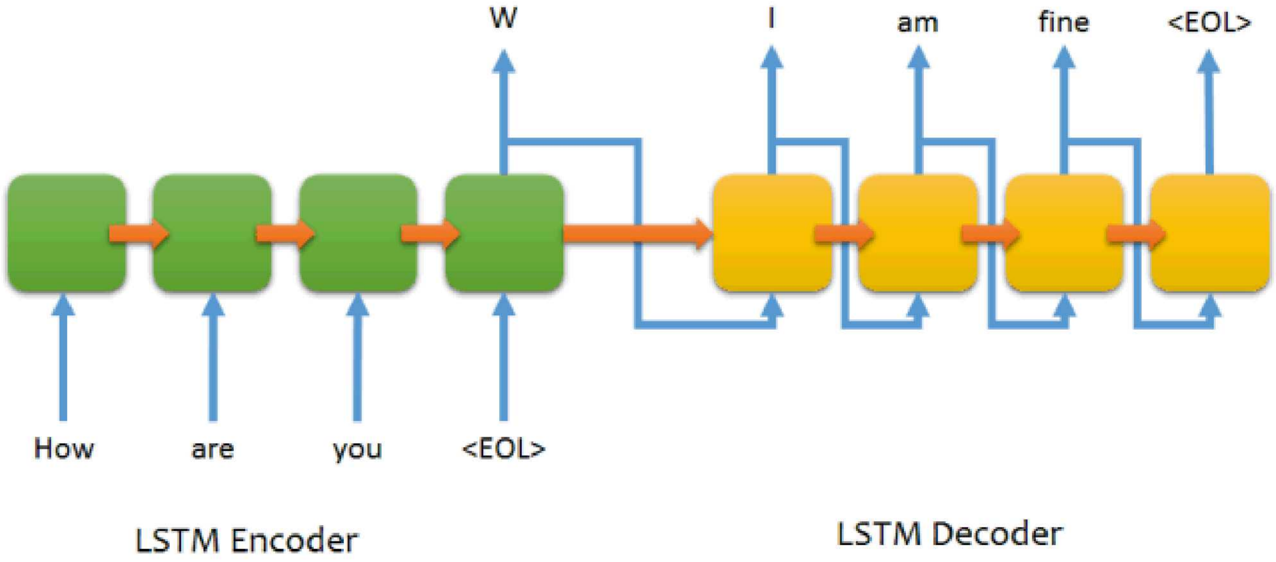
Generally use LSTMs to capture temporal dependencies

Can we cast video prediction as a sequence to sequence problem?



Sequence-to-sequence prediction

Typically involve an encoder portion and a decoder portion
- Rather than reconstruct the same input, predict the next sequence of outputs
- Encoder: Take the input sequence and learn a representation of the inputs
- Decoder: Take output from the encoder and predict next sequence of outputs

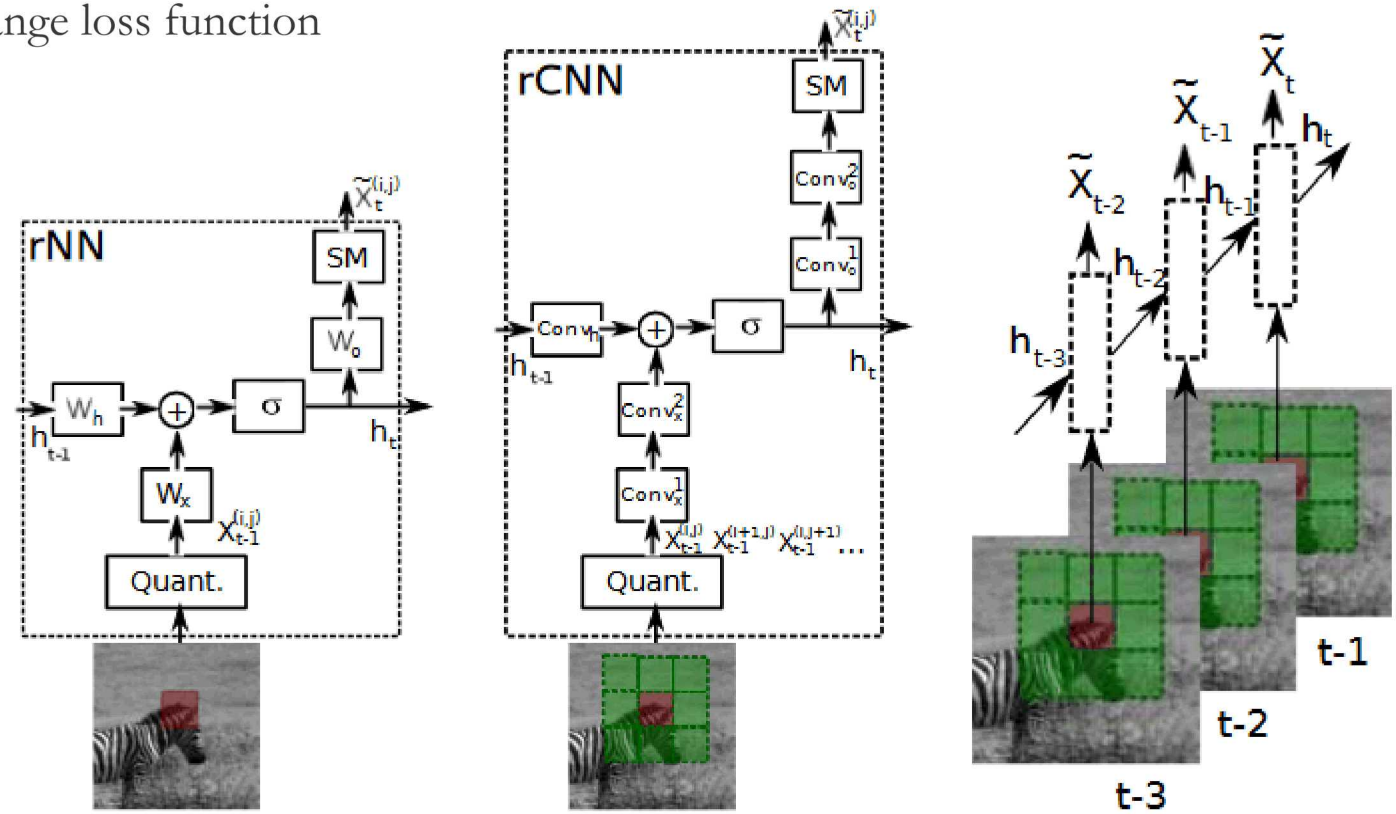


Extending sequence-to-sequence to video

Use spatial patches in images to replace words

- rNN: uses a single patch. Treats neighboring patches independently
- rCNN: also feed in the neighboring patches. Helps to with spatial correlations
- Parameters are shared over time

Change loss function



See: M. A. Ranzato, A. Szlam, J. Bruna, M. Mathieu, R. Collobert, S. Chopra **Video (language) modeling: a baseline for generative models of natural videos.** https://arxiv.org/abs/1412.6604

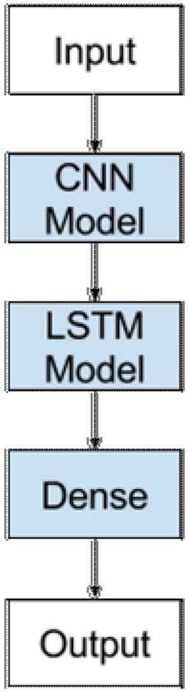
Fully connected LSTM

Combine CNN and LSTM

- Has been used for captioning:
- … it is natural to use a CNN as an image "encoder", by first pre-training it for an image classification task and using the last hidden layer as an input to the RNN decoder that generates sentences

Problems with this approach:

- Convolutions and LSTMs are modelled separately
- CNNs do not have recurrence
  - Only operate on spatial features
- LSTMs do not capture spatial features
  - N-tensor is flattened to a 1-D vector
- What about convolutional layers connected to LSTM layers?
  - The major drawback is that convolutional layers are connected to LSTMs and recurrent weights are fully connected (dense)
  - Lots of parameters and redundancy



What do we have?

LSTM: Recurrent neural networks that capture temporal relationships

CNN: State-of-the-art in computer vision for spatial relationships

Sequence-to-sequence models: use of LSTMs to process and generate sequences

CNN/LSTM network

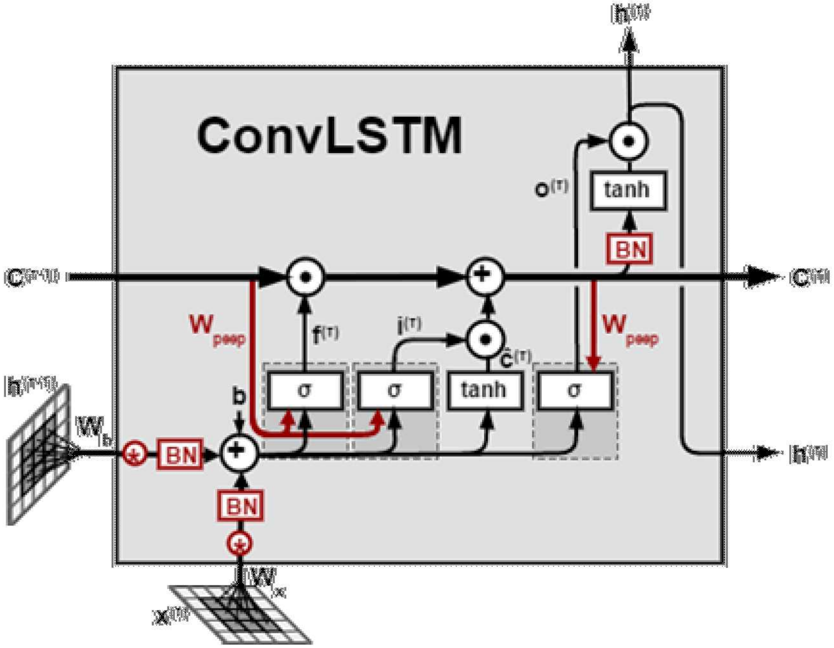
Precipitates the generation of the convolutional LSTM neuron

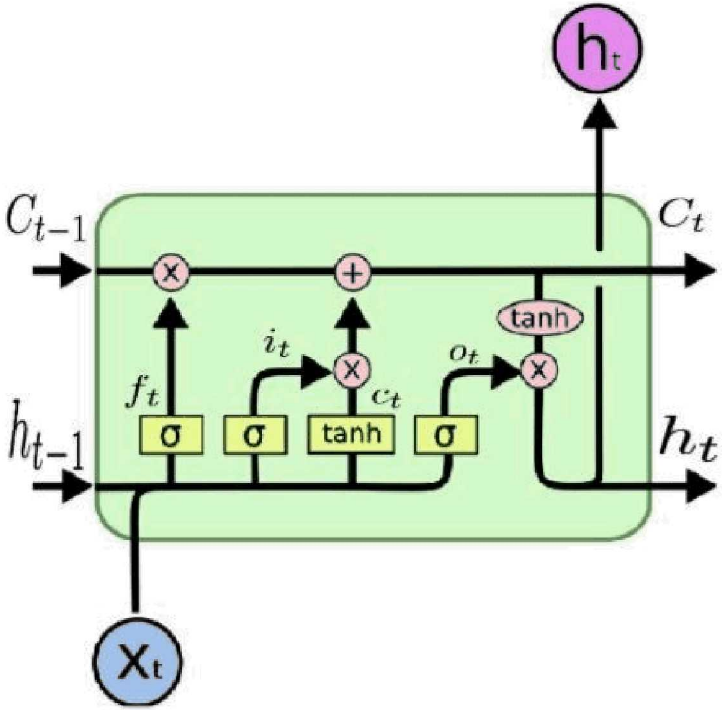
- **Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting** (https://arxiv.org/abs/1506.04214) **2015**
- Predict weather
- "Give a precise and timely prediction of rainfall intensity in a local region over a relatively short period (0-6 hours)

ConvLSTM -- Pictures

Models spatio-temporal relationships in the data

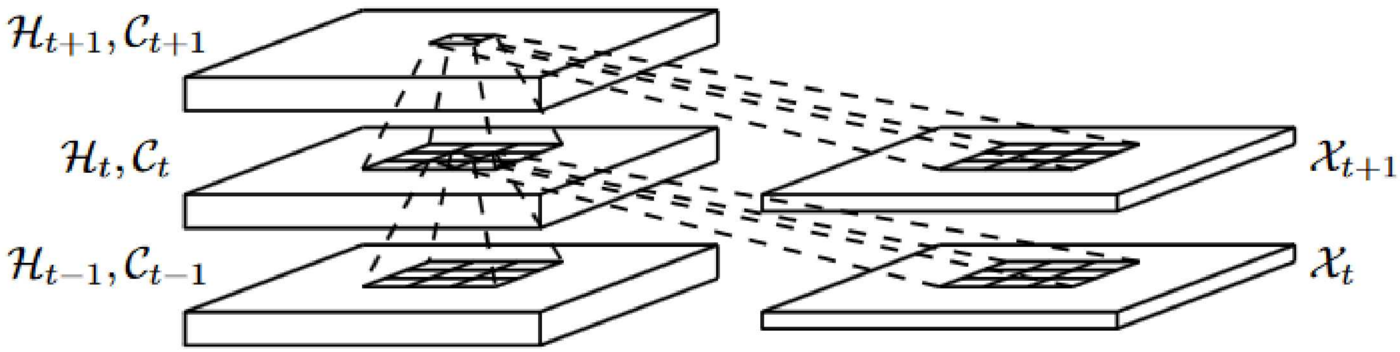
- Integration of CNN and LSTM
- Recurrent layer (like and LSTM)
- Internal standard matrix multiplications exchanged with convolution operations
- Retains multiple-dimension data (LSTM is one dimensional)



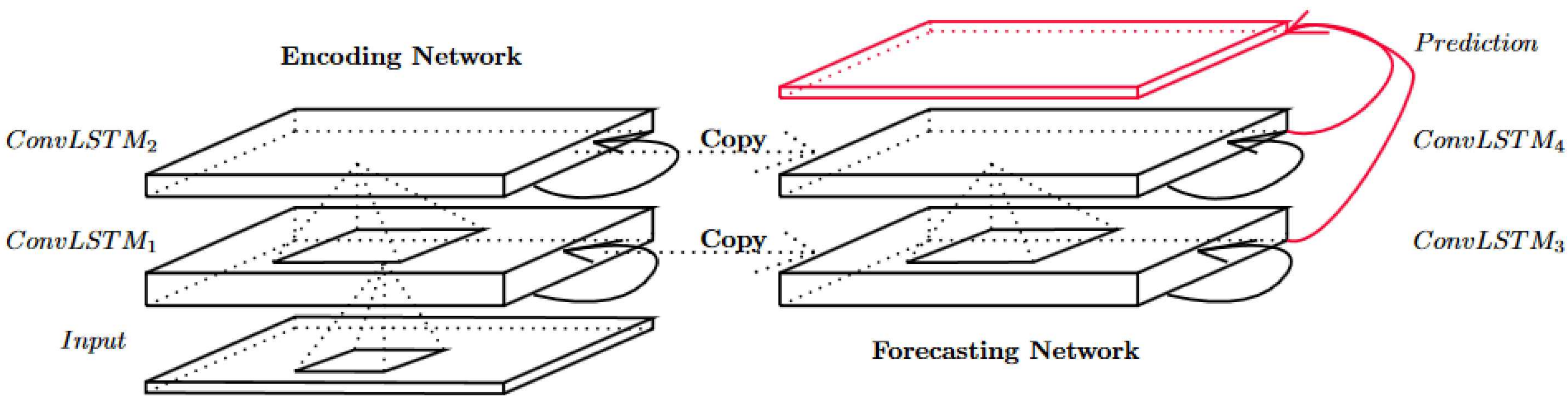


ConvLSTM – Shown another way

Everything is now stored spatially as a 3-D tensor rather than a vector



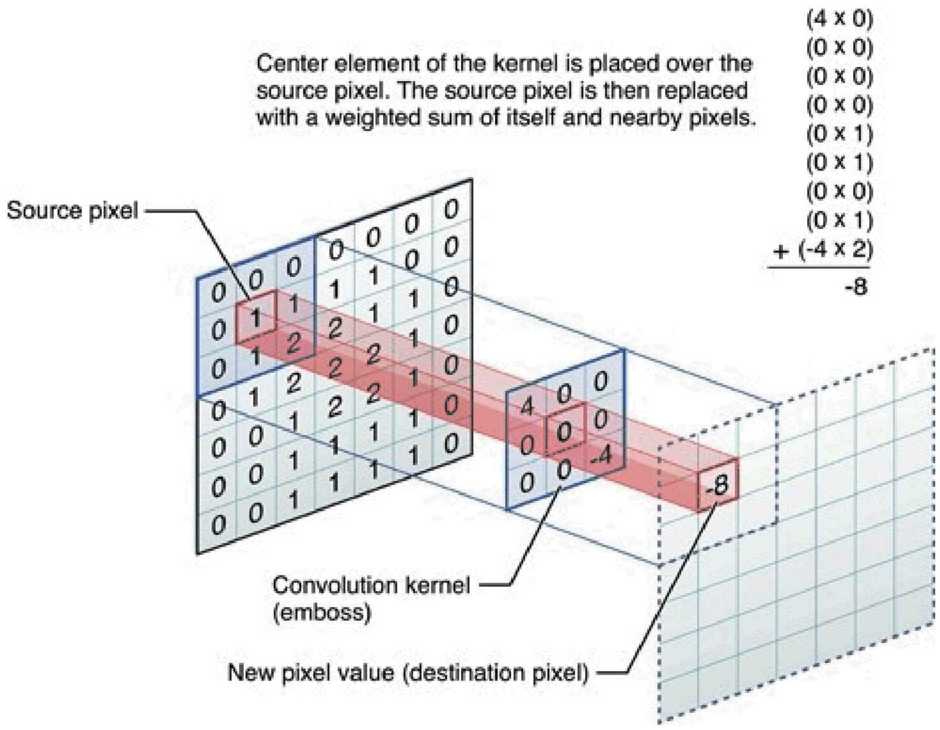
Use sequence to sequence encoder and decoder/forecasting portions



Review: Convolution

Recall: convolution is an integral that expresses the amount of overlap (or inner product) of one function g as it is **shifted** over another function f

- Blends one function with another
- Operates in multi-dimensional spaces
- Output is multi-dimensional



ConvLSTM -- MATH

LSTM

$$
\begin{aligned}
i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci} \circ c_{t-1} + b_i) \\
f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf} \circ c_{t-1} + b_f) \\
c_t &= f_t \circ c_{t-1} + i_t \circ \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\
o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co} \circ c_t + b_o) \\
h_t &= o_t \circ \tanh(c_t)
\end{aligned}
$$

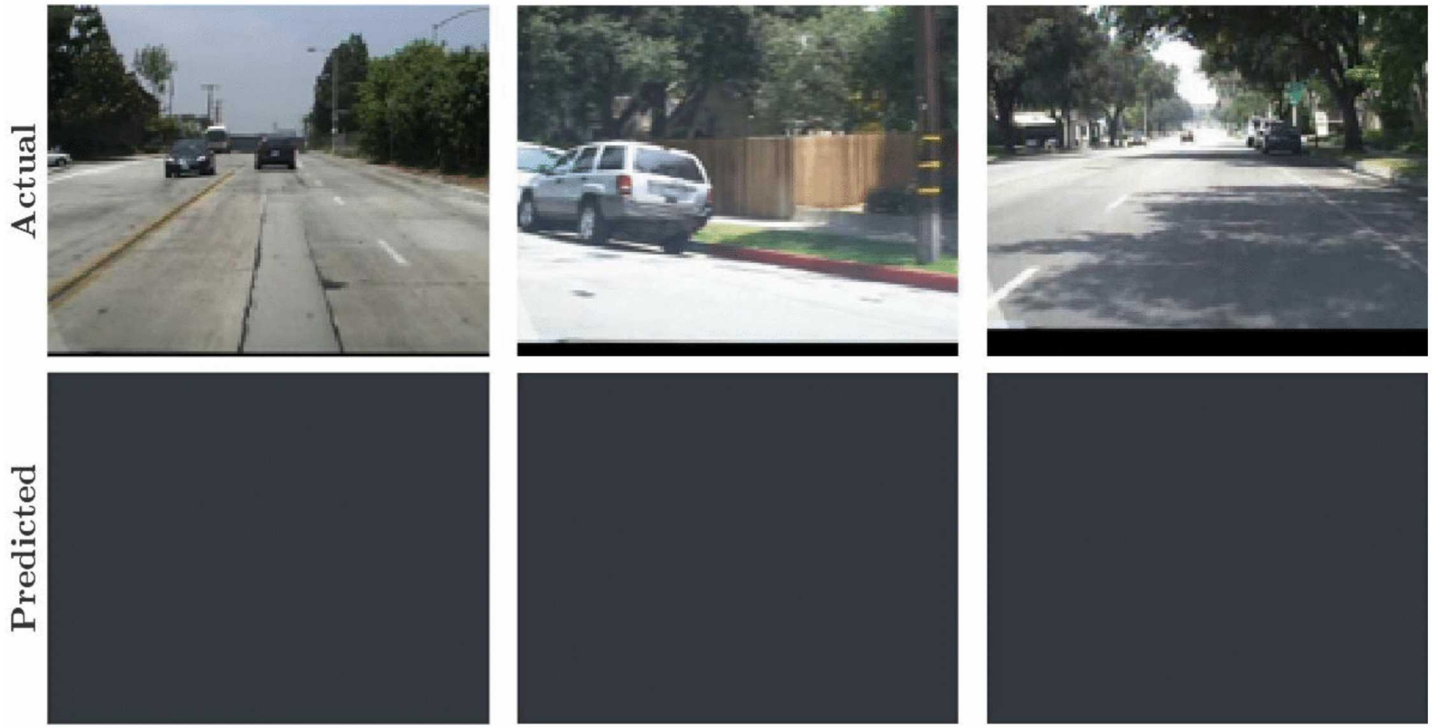
ConvLSTM

$$
\begin{aligned}
i_t &= \sigma(W_{xi} * \mathcal{X}_t + W_{hi} * \mathcal{H}_{t-1} + W_{ci} \circ \mathcal{C}_{t-1} + b_i) \\
f_t &= \sigma(W_{xf} * \mathcal{X}_t + W_{hf} * \mathcal{H}_{t-1} + W_{cf} \circ \mathcal{C}_{t-1} + b_f) \\
\mathcal{C}_t &= f_t \circ \mathcal{C}_{t-1} + i_t \circ \tanh(W_{xc} * \mathcal{X}_t + W_{hc} * \mathcal{H}_{t-1} + b_c) \\
o_t &= \sigma(W_{xo} * \mathcal{X}_t + W_{ho} * \mathcal{H}_{t-1} + W_{co} \circ \mathcal{C}_t + b_o) \\
\mathcal{H}_t &= o_t \circ \tanh(\mathcal{C}_t)
\end{aligned}
$$

$*$ represents the convolution operator

Variables are capitalized in ConvLSTM because they are 3D tensors

PredNet results



Calculating image differences

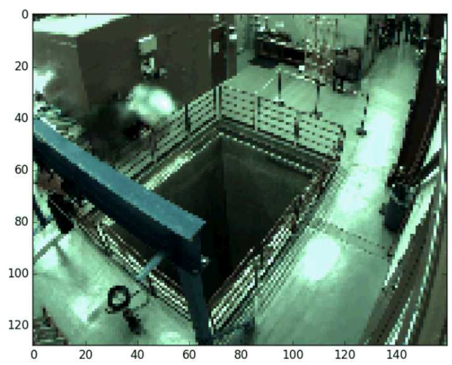
Compare **Predicted Image** to **Actual Image**

1. Convert both images to grayscale
2. Calculate Squared Error, E, for each pixel i

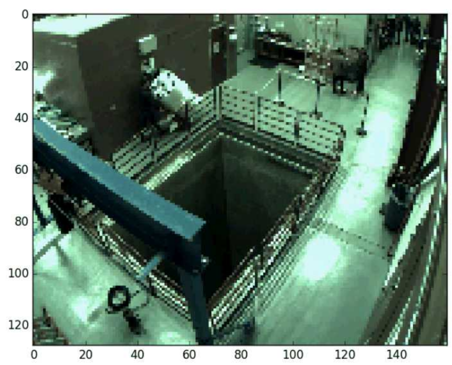
$P = Predicted\ Image \qquad A = Actual\ Image$

$$E_i = (P_i - A_i)^2$$

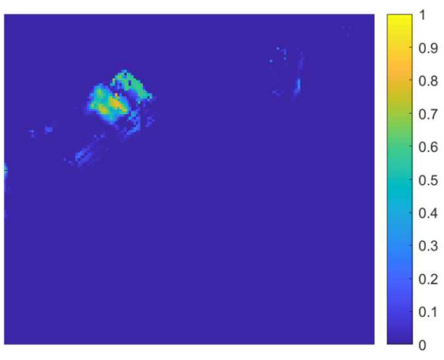
Predicted Image



Actual Image

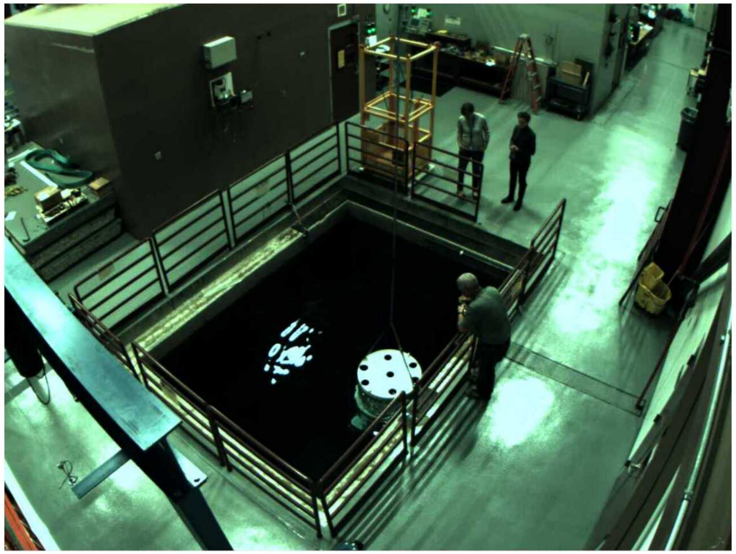


Squared Error Image



Data

- Sandia developed a proxy use-case to transfer a large (approx. 5ft. tall by 3 ft. wide) container into and out of a floor vault
- Sandia deployed two NGSS cameras in the Gamma Irradiation Facility (GIF)
  - Collected down-time data and active scripted container movements over multiple days
  - Collections include both full (water) and empty floor vault scenarios



Scenarios for Data Analysis Plan

Evaluate what the PredNet algorithm determines as "anomalous" and its relevance to safeguards

Test four categories of potentially anomalous scenarios:

1. **Unintentional Anomalies** – examine anomalies that are identified in "normal" operational scenarios
2. **Intentional Anomalies** – intentionally insert anomalous frames to determine algorithm response
3. **Operational Anomalies** – change operational activities within a facility, including types of containers present, appearance of containers, areas in which container are located
4. **Safeguards scenarios** – experiment with scenarios that are determined to be of high safeguards interest, e.g. greyscale images, longer time lapse, and play-back loops

Results

Experiment trained only on containers leaving the facility

Significantly larger irregulatity scores for containers entering the facility

Calculate Mean Squared Error for images in a series

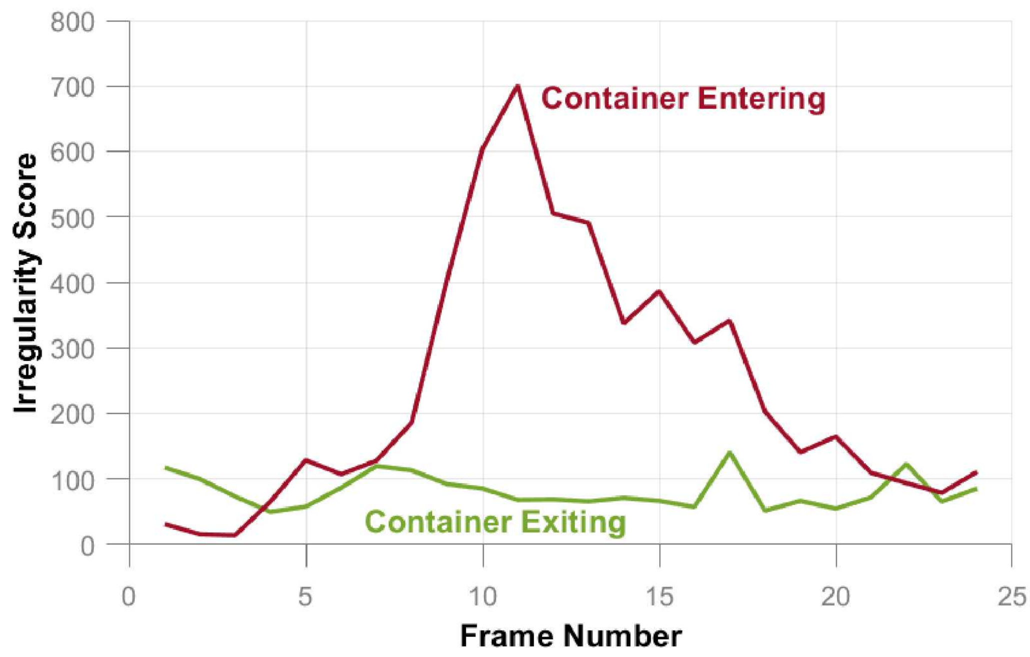
1. Convert both images to grayscale
2. Calculate Squared Error, E, for each pixel i

P = Pixel values from predicted image

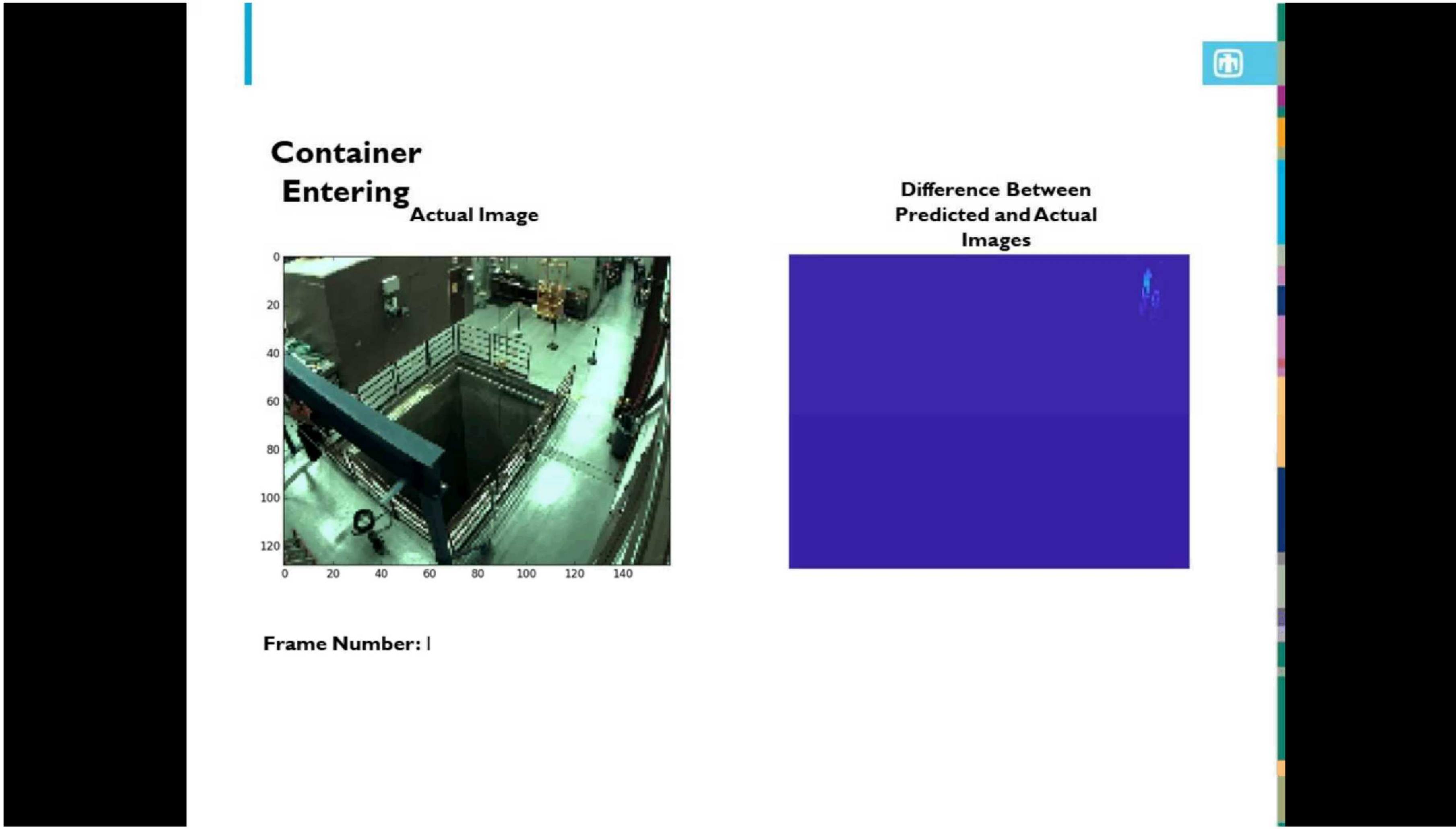
A = Pixel values from actual image

N = Number of pixels

$$\frac{1}{N}\sum_{i=1}^{N}(P_i - A_i)^2$$



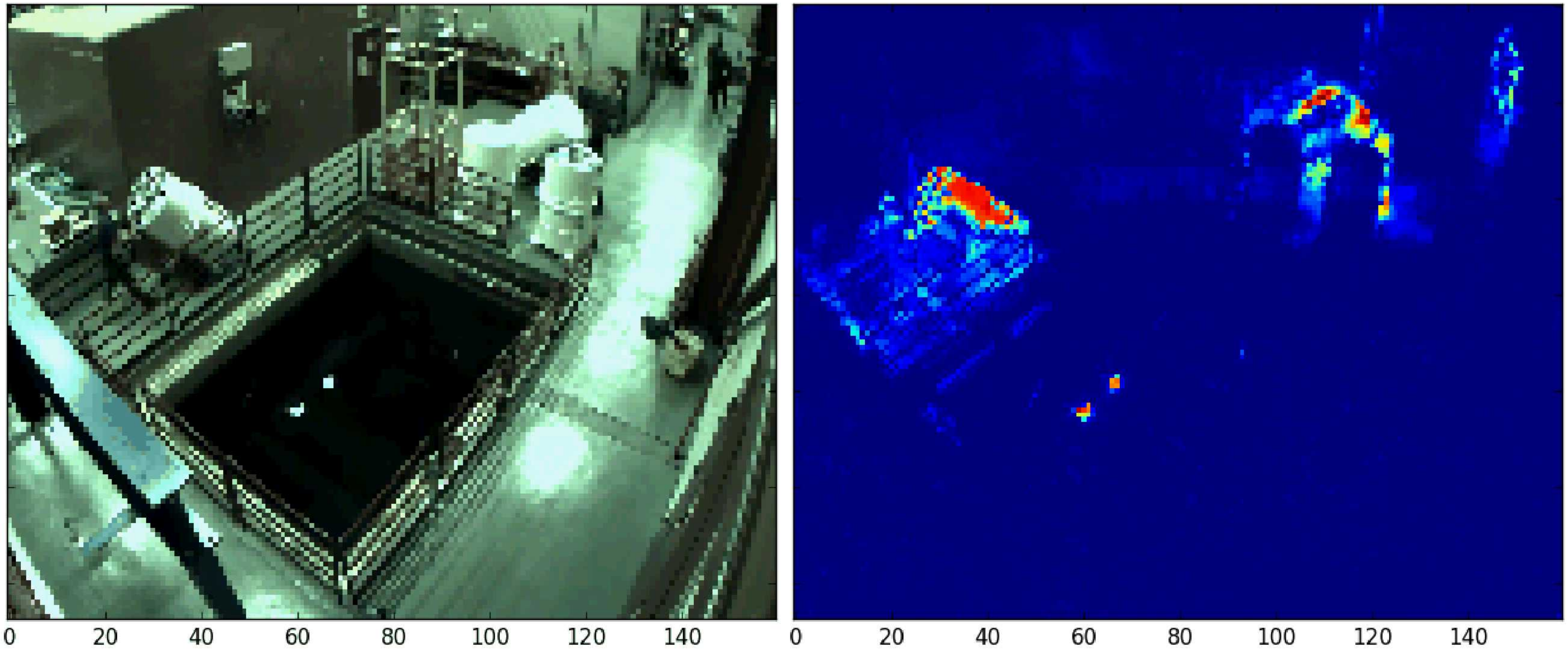
Video showing the sequence of containers entering and exiting the facility



Further results

Water behavior is hard to predict:

- Water reflection is out weighted by movement anomaly



Conclusions and Future Work

PredNet is a viable solution for detecting spatio-temporal anomalies

- Does not require labelled data (which can be time consuming and labor intensive)
- Does not require (potentially sensitive) data to leave given facilities
- Demonstration of detection of normal objects and people doing anomalous activities
- Can detect spatial anomalies (people walking in new areas)
- Can detect spatio-temporal anomalies (moving in the wrong direction)
- Hard to predict water behavior

Cons:

- Time consuming (in computational time) to train (but alleviates human burden)

Future work

- Examine PredNet on more extensive analyses
  - What does PredNet detect in day to day activities
  - Does PredNet overly detect anomalies?
- Extend to work with supervised approaches
  - Anomalous activities near objects of interest
  - Can the supervised and unsupervised share weights?