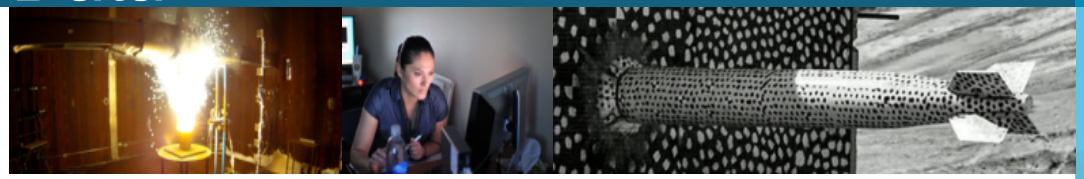
SAND2021-10007C

Sandia National Laboratories

A Deep Learning Method For Spatio-temporal Detection Of Atypical Activity In NGSS Camera Data



Presented by: Michael R. Smith[1]

David Hannasch[1], Michael Hamel[1], Maikael Thomas[2], Maria Camila Gaitan-Cardenas[1]

1 Sandia National Laboratories, Albuquerque, NM, USA

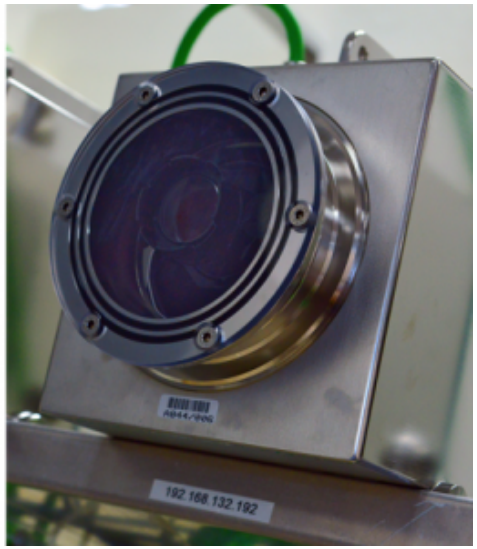
2 International Atomic Energy Agency, Vienna, Austria

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.

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.

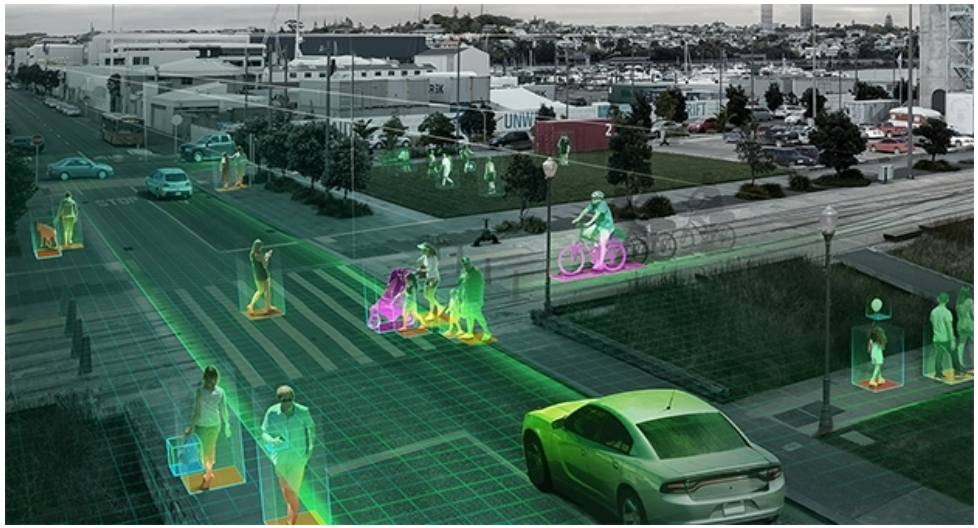
Motivation

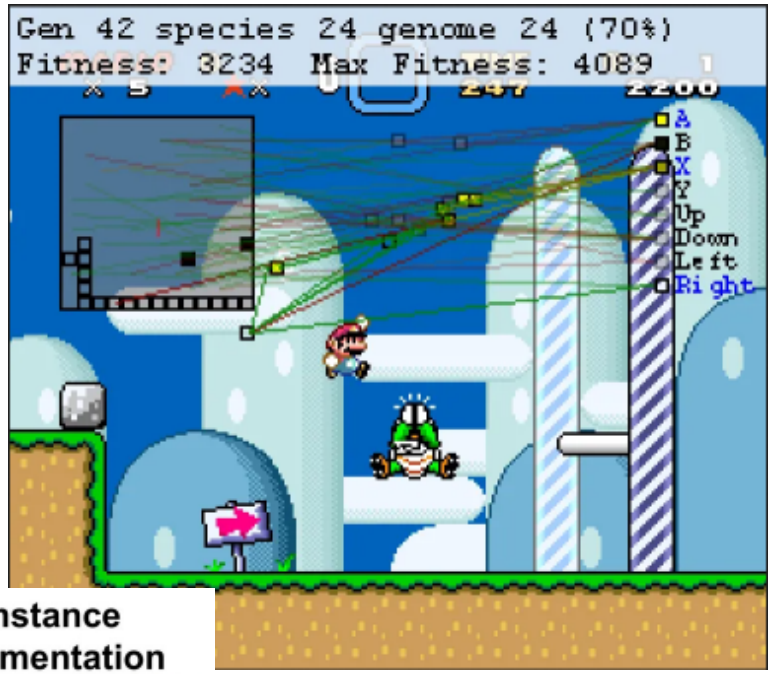
- Review of NGSS surveillance video by IAEA inspectors is **mundane and tedious**
  - Look for anomalous activity (**unknown unknowns**)
  - Low tolerance for false negatives (**Frame by Frame)**
- **Can we help improve the efficiency of an inspector with deep learning approaches?**
- Assumptions:
  - No or very few labelled data (cannot enumerate all anomalies)
  - Data cannot leave facility
  - Non ML expert users
  - Environments and processes change significantly across facilities and cameras

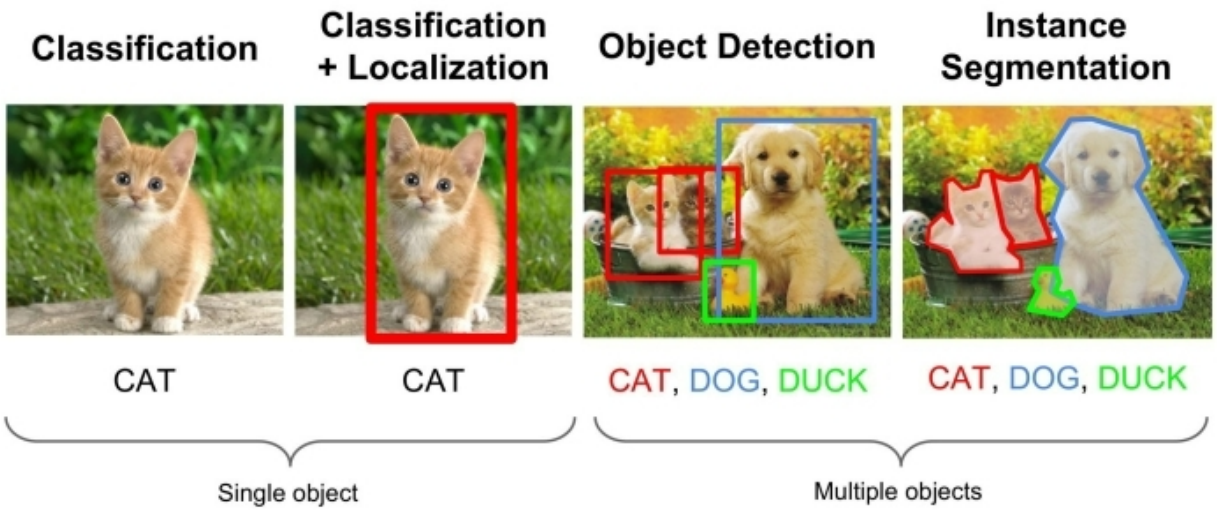


https://www.iaea.org/newscenter/news/iaea-safeguards-now-applied-in-183-states-worldwide-safeguards-statement-2019

Why Deep Learning?

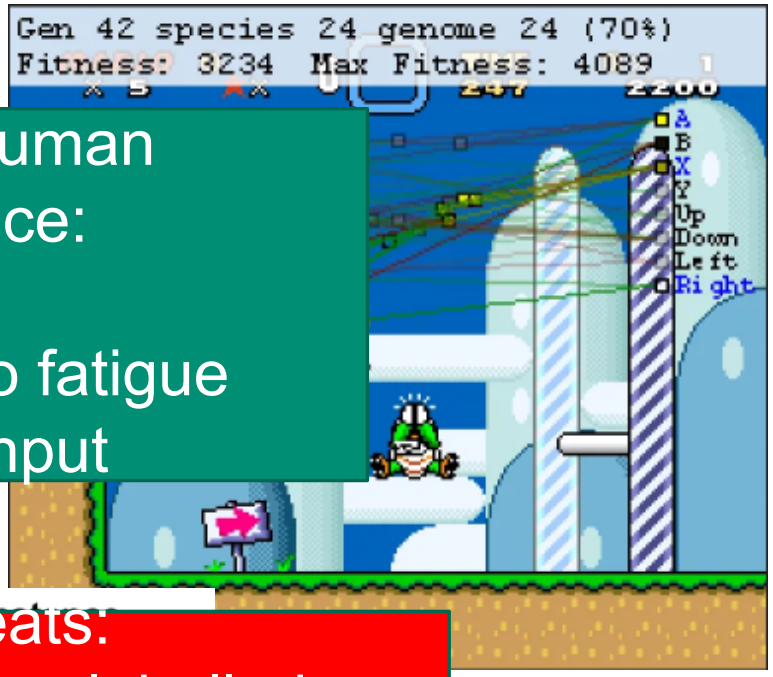






https://www.nvidia.com/en-sg/deep-learning-ai/inference-platform/intelligent-video-analytics/
https://ryanscharf.com/project/mario/
https://medium.com/zylapp/review-of-deep-learning-algorithms-for-object-detection-c1f3d437b852

Why Deep Learning?



Often beat human performance:

- Less error prone
- Not susceptible to fatigue
- High data throughput

Some caveats:

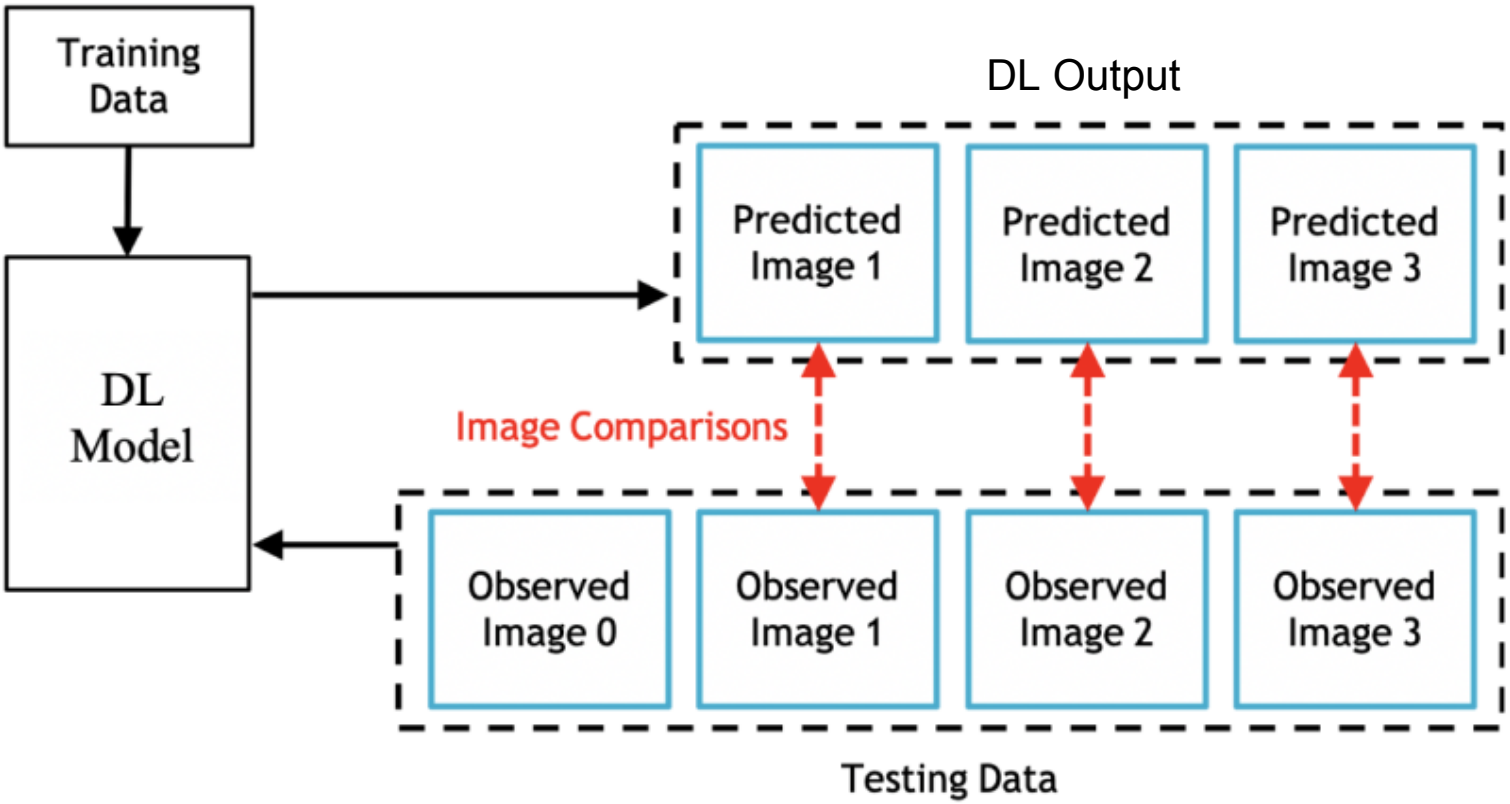
- Require lots of training labelled data
- Can be biased based on training data
- Does not consider context

https://www.nvidia.com/en-sg/deep-learning-ai/inference-platform/intelligent-video-analytics/
https://ryanscharf.com/project/mario/
https://medium.com/zylapp/review-of-deep-learning-algorithms-for-object-detection-c1f3d437b852

Self-Supervised Learning

Take advantage of large amounts of **unlabeled, structured** data

- No labeling required
- Predict the next value in a sequence
- In video, predict each pixel value in the next frame

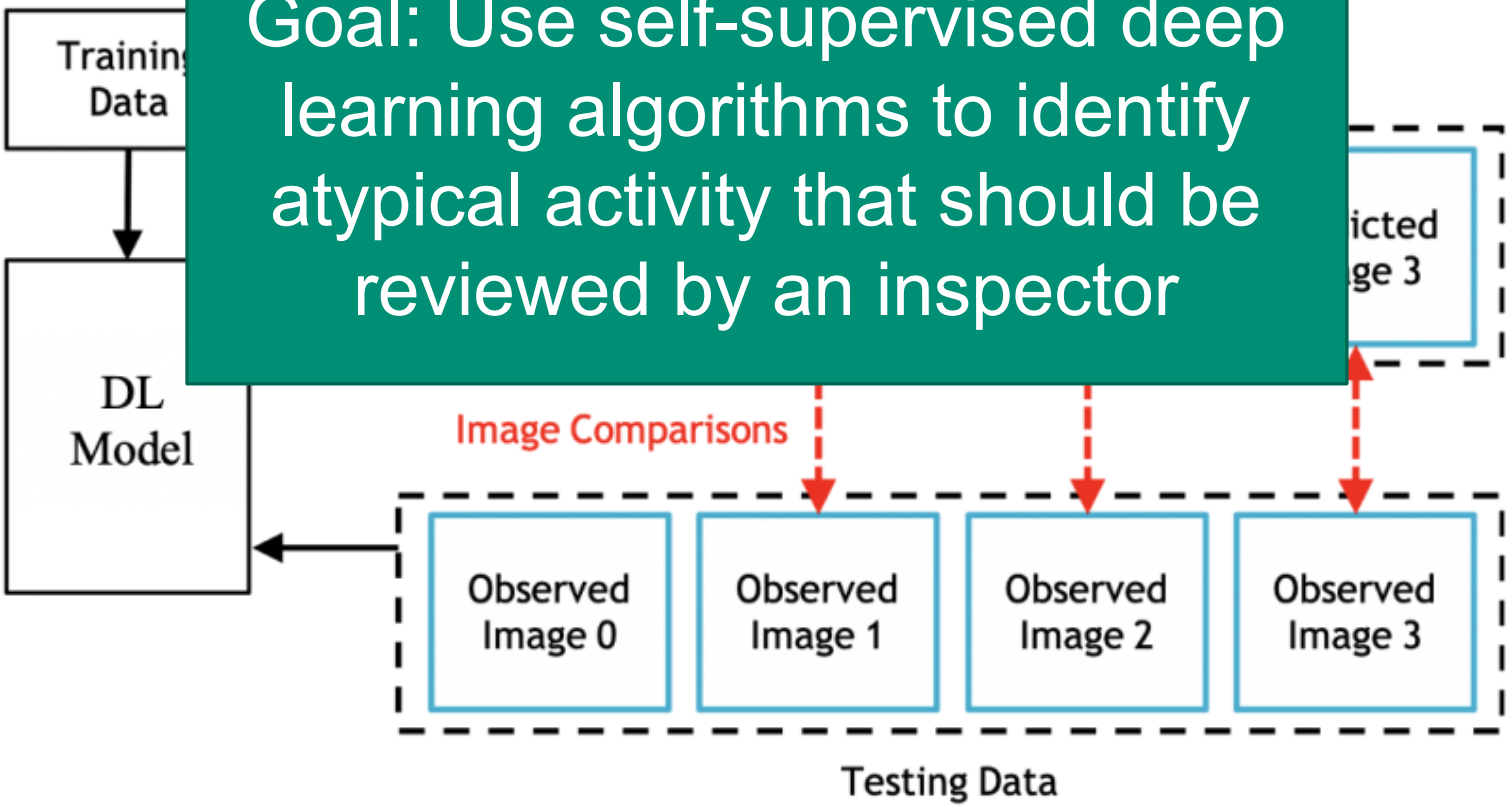


Self-Supervised Learning

Take advantage of large amounts of **unlabeled, structured** data

- No labeling required
- Predict t
- In video, predict each pixel value in the next

Goal: Use self-supervised deep learning algorithms to identify atypical activity that should be reviewed by an inspector

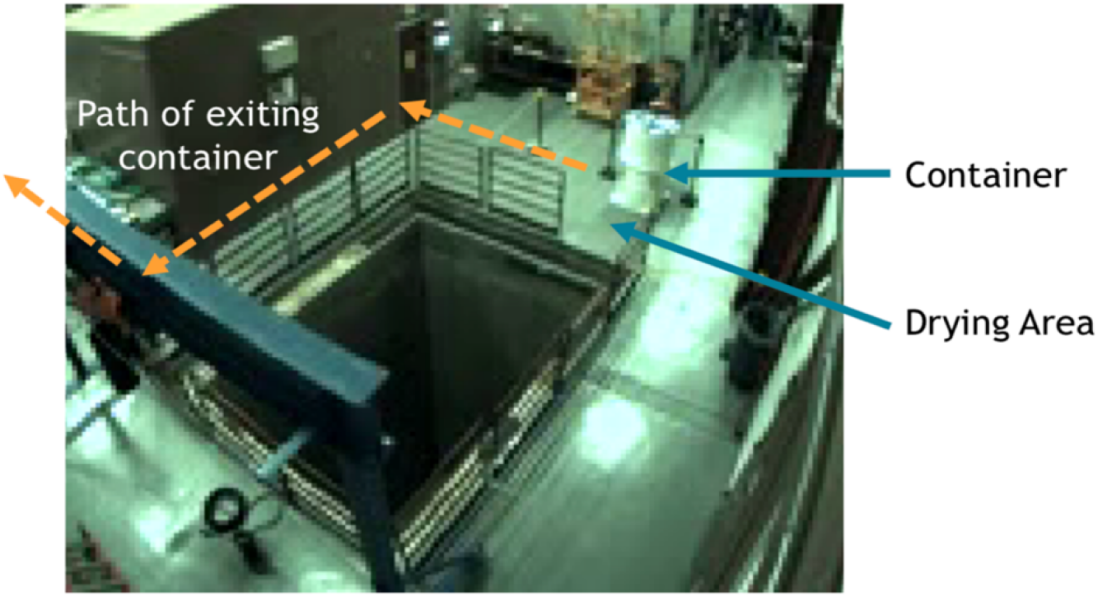


Atypical Activity Detection Workflow

1. Curate training data of typical operating activities
2. Train and tune the DL algorithm
3. Calculate atypical score
4. Present the results to an end-user

Use-case:

- Containers exiting represent typical activity
- Containers entering represent atypical activity



Curate Training Data

Performance of the algorithm depends on the quality of the training data.

- Represent standard operating activities –OR–
- Specific activities of interest

Use case:

- Examples of image series of a container being picked up at the drying area and moved along the path out of the frame
- Work required: spicing the video to capture just the activity that we are concerned with.

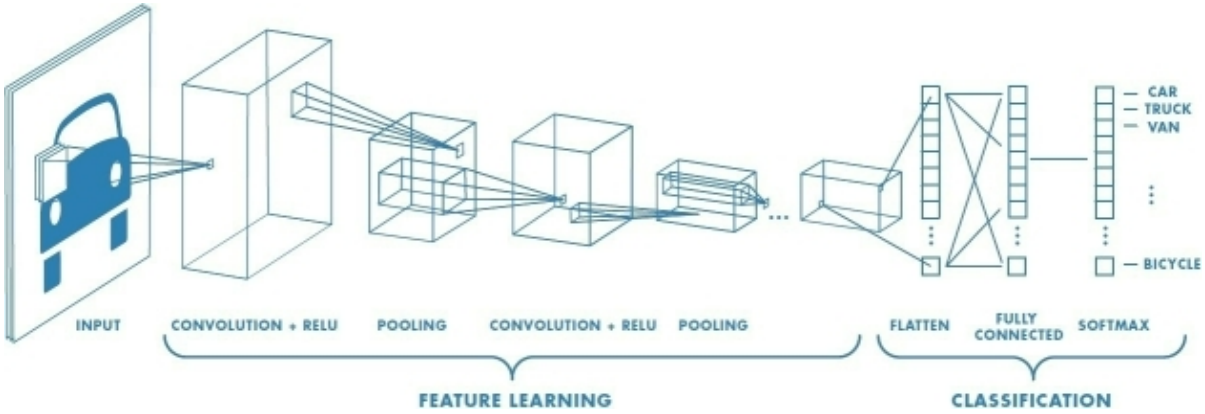
Labels are not required, but some care is needed in choosing training data

Train and Tune the Self-Supervised DL Algorithm

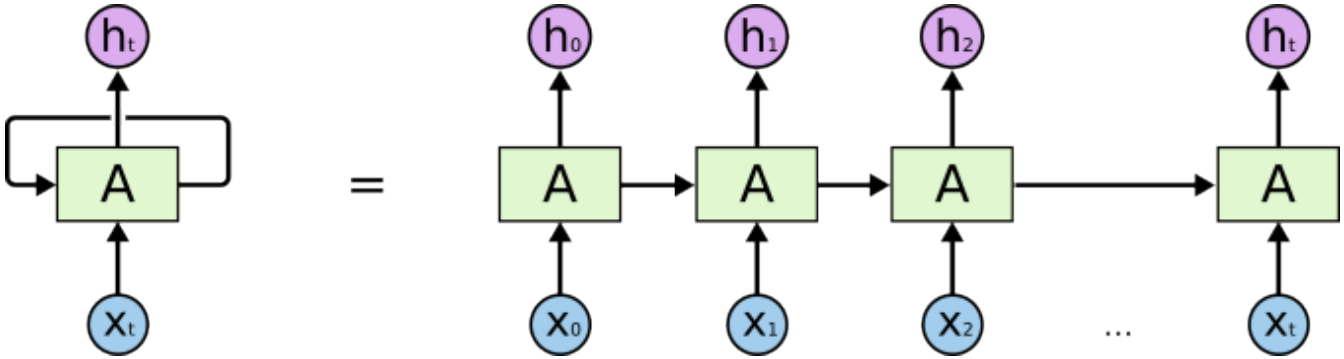
Deep Predictive Coding Networks (PredNet)

Self-supervised technique that combines spatial and temporal modeling:

- Convolution (spatial)



- Recurrence (temporal)



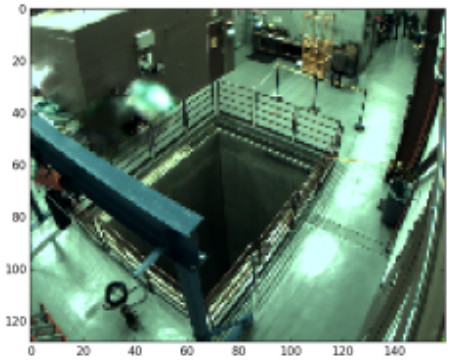
Key Point: Captures spatial and temporal patterns

https://coxlab.github.io/prednet/
https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53
http://colah.github.io/posts/2015-08-Understanding-LSTMs/

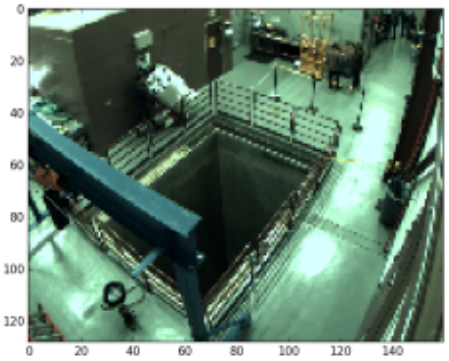
Atypical Behavior Score Calculation

Compare predicted images with observed images

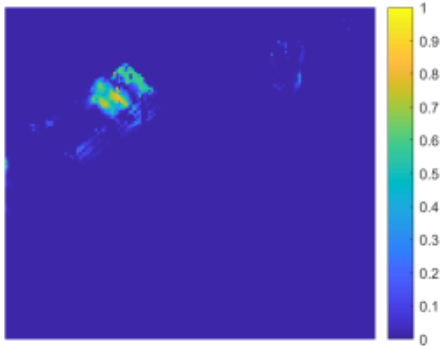
Predicted Image



Actual Image



Squared Error Image



Not obvious which method is best:

We examined:

- Mean Error: Absolute differences calculated pixel-by-pixel
- Mean Squared Error (MSE): Squared differences calculated pixel-by-pixel
- Structural Similarity (SSIM) Index: Objective measure of image quality using image properties such as luminance, contrast, and structure

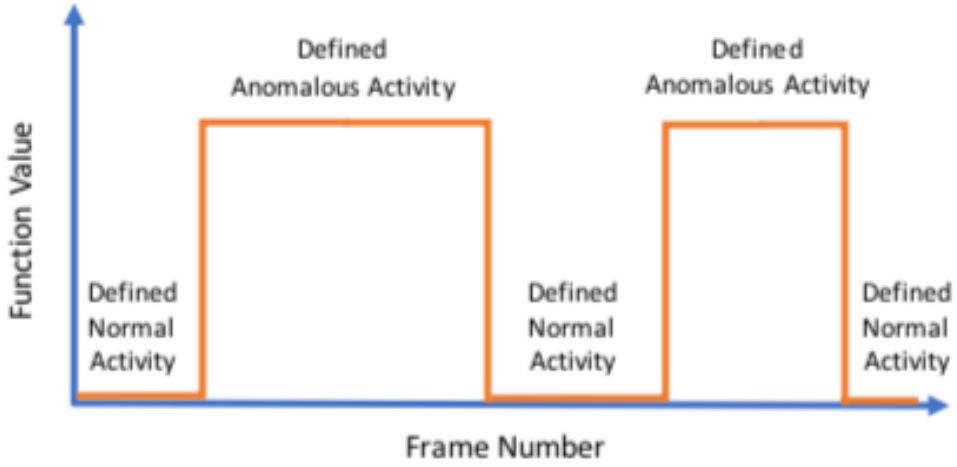
Image Preprocessing

- Images filtered with a median filter prior to calculating MSE
- Methods evaluated using grayscale and color images
- Color images used three channels: red, green, and blue

Evaluation Metric

Define an “ideal” response step function with two values

- Zero value for frames when no defined anomalous activity is occurring
- A non-zero value for frames when defined anomalous activity is occurring



Calculate how well a given method matches the “ideal” response step function

- A small metric value is desired
- Non-zero value of step-function chosen as maximum value created by image

$$M = \sum_{i=1}^{N_f} \frac{|S(i) - D(i)|}{\max(S) \cdot N_f}$$

M: Metric Score

N_f: Number of frames evaluated S: Value of step function

D: Calculated comparison value between a predicted and actual image

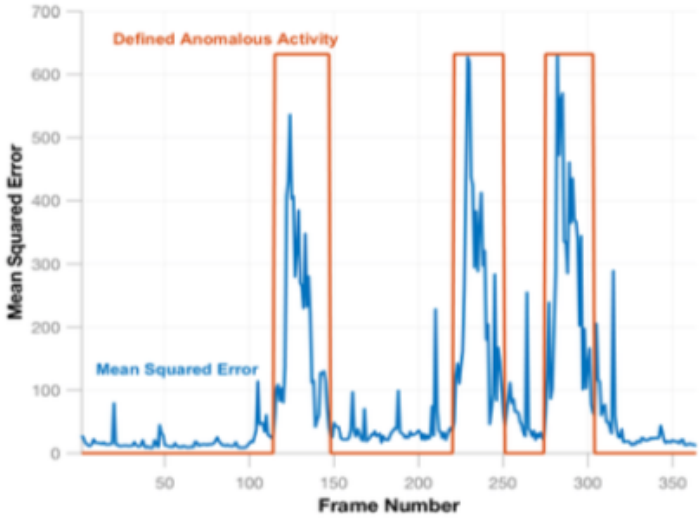
Atypical Score Results

Evaluated Video

- Some image series used in training
- Examples of image series of containers entering the frame and being moved into the drying area along the sa

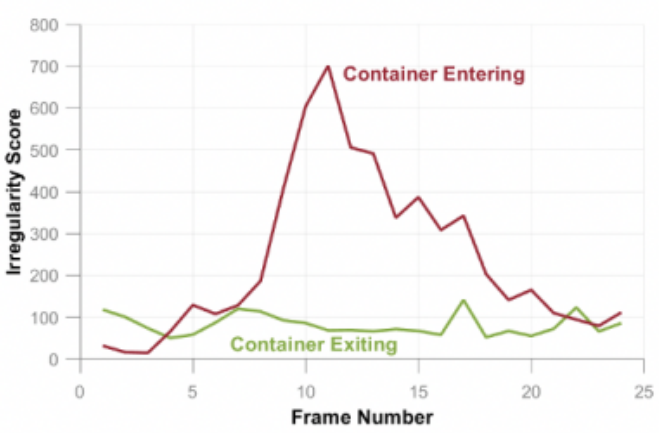
Evaluated Video Sequence

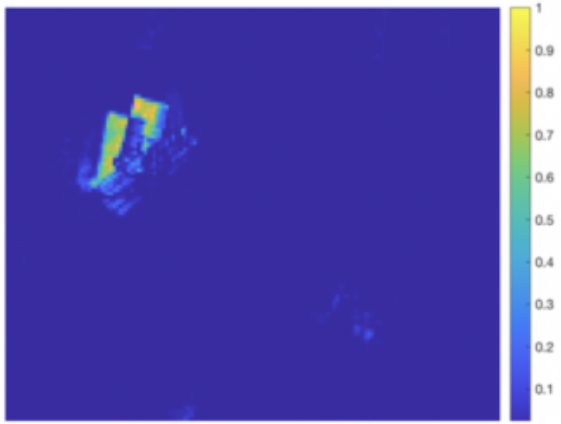
- Exit
- Exit
- Enter
- Exit
- Enter
- Enter
- Exit



Comparison Method	Grayscale or Color	Metric Value	Method Parameters
Mean Error	Grayscale	0.347	--
Mean Error	Color	0.363	--
MSE	Grayscale	0.196	--
MSE	Color	0.202	--
SSIM Index	Grayscale	0.204	Radius: 40 pixels Luminance Exponent: 0 Contrast Exponent: 0 Structure Exponent: 1
SSIM Index	Color	0.212	Radius: 40 pixels Luminance Exponent: 0 Contrast Exponent: 0 Structure Exponent: 1
MSE with Median Filter Preprocessing	Grayscale	0.193	Filter Size: 10×10 pixels
MSE with Median Filter Preprocessing	Color	0.197	Filter Size: 10×10 pixels

Visualizing Atypical Activity





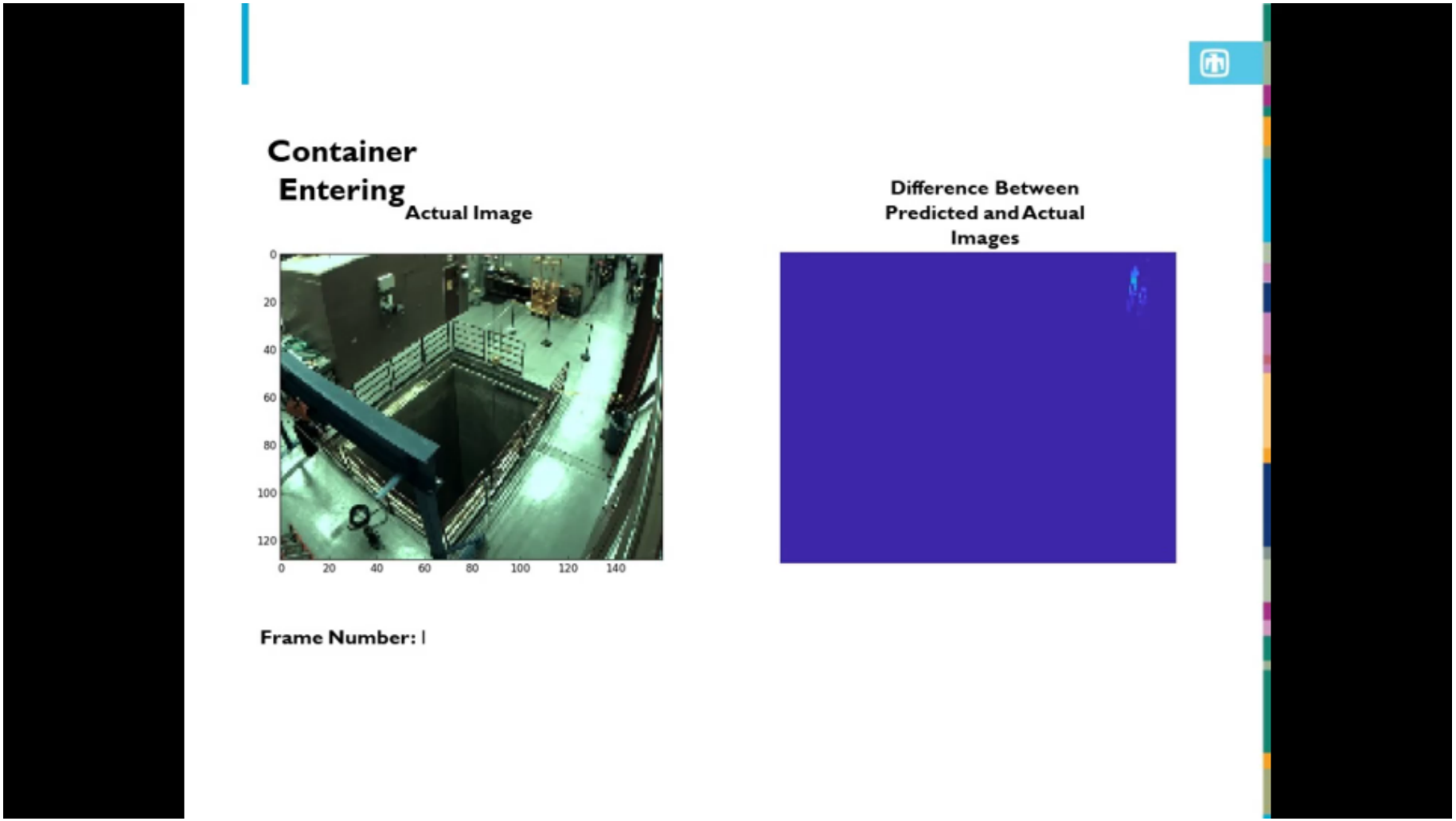
Left: Line graph over time to determine where in the video to look

Center: The actual frame at the peak of the irregularity score.

Right. Heat map of the differences clearly show where the atypical activity occurs.

Highlights spatially and temporally where atypical activity occurs

Video showing the sequence of containers entering and exiting the facility



Conclusion and Future Work

Presented a framework for identifying and visualizing atypical behavior in video

- Initial results suggest that self-supervised deep learning methods are a promising avenue for using deep learning without having to obtain large amounts of labeled training data

The entire workflow is implemented in a Docker container with all required libraries

- We are looking at alternative self-supervised algorithms that are less computationally and memory expensive

Examining specific use-cases for of interest to the IAEA

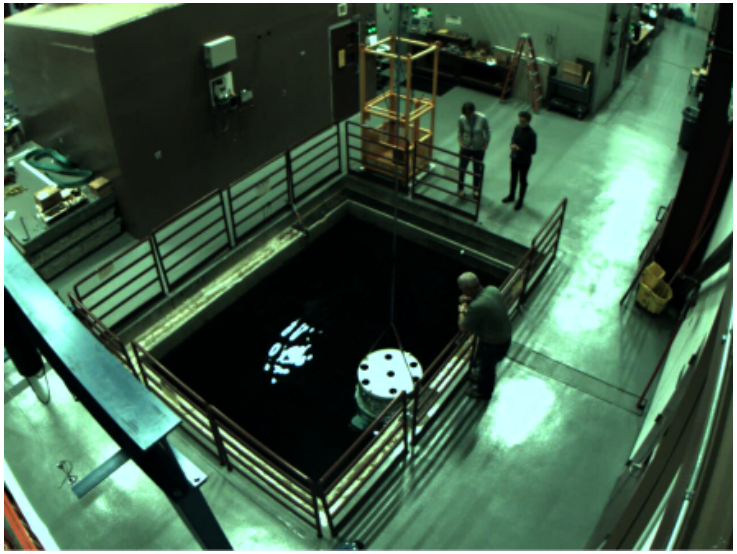
- Different frame rate captures
- Very large, moving objects vs small quick moving objects

Thank you!

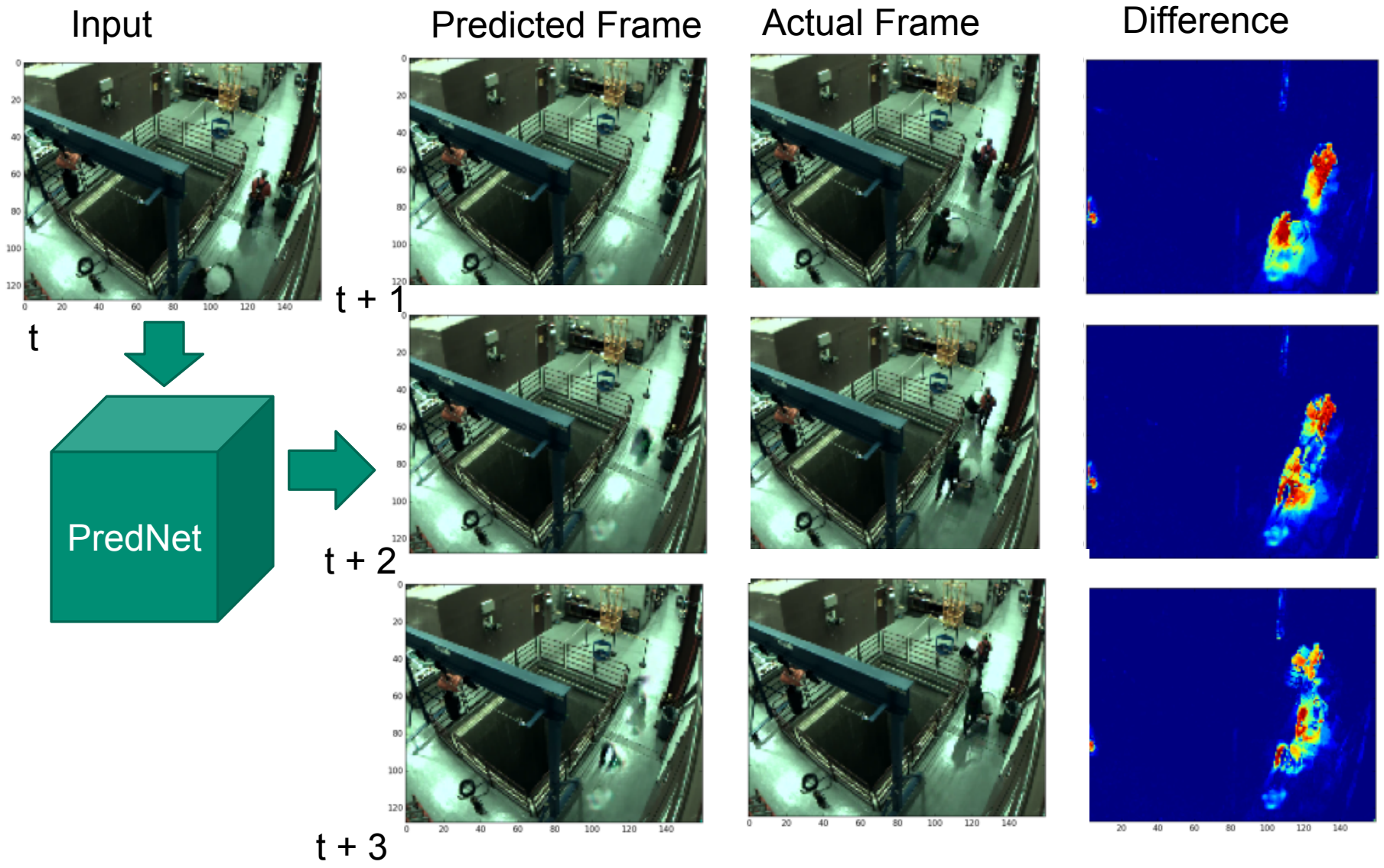
Questions: Mike Smith:
Michael.smith3@sandia.gov

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

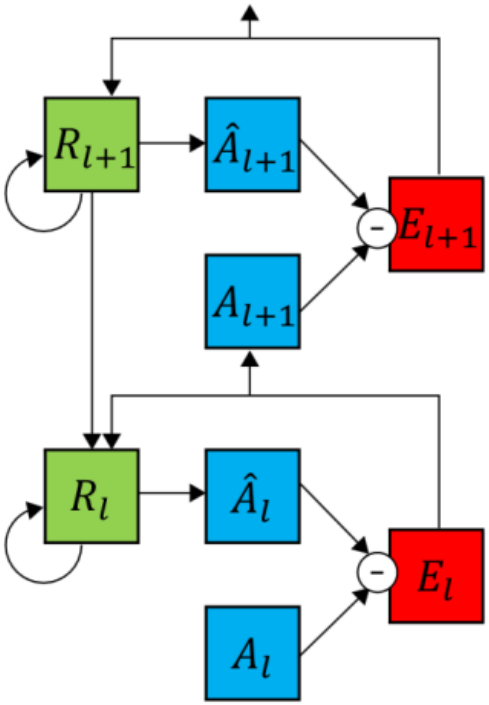


Solution: Deep Predictive Coding Networks for Video Prediction and Unsupervised Learning (PredNet)

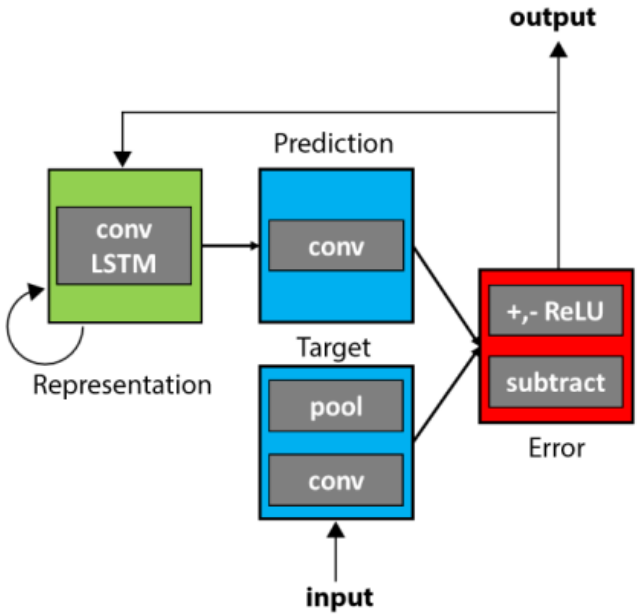


PredNet Architecture

- 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

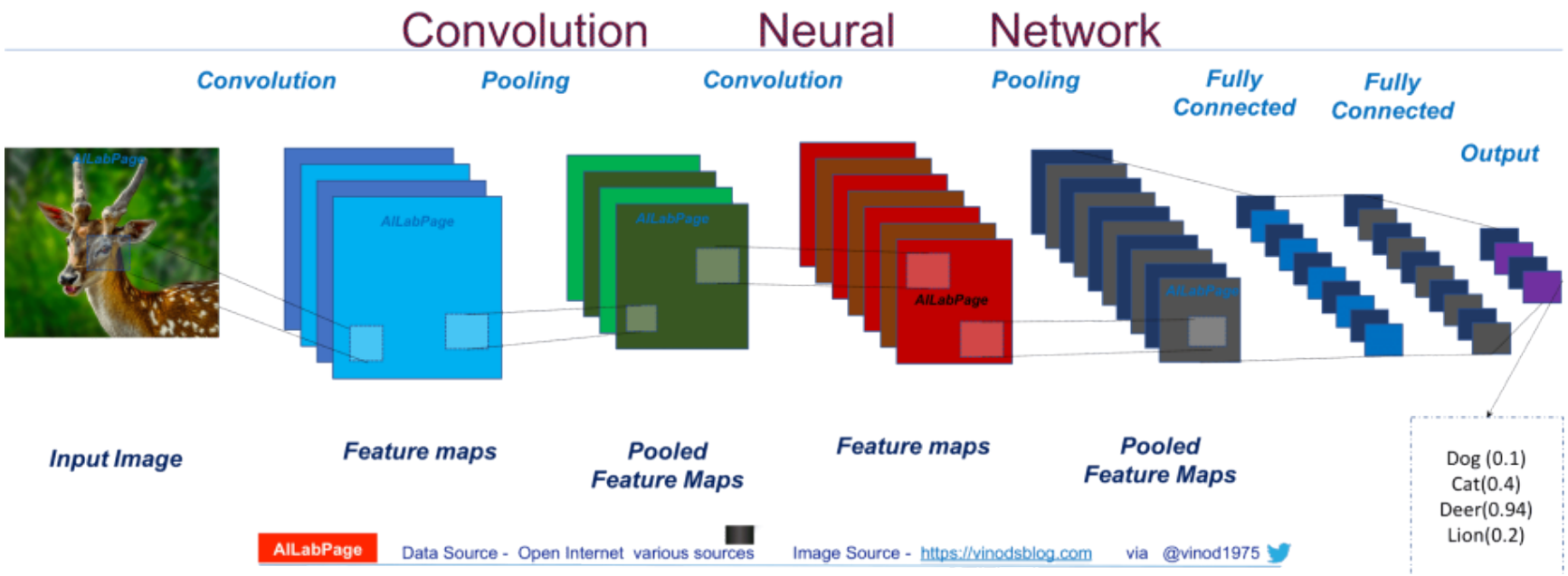


- Module operations

Convolution Neural Network (CNN)

Best approach for working with images

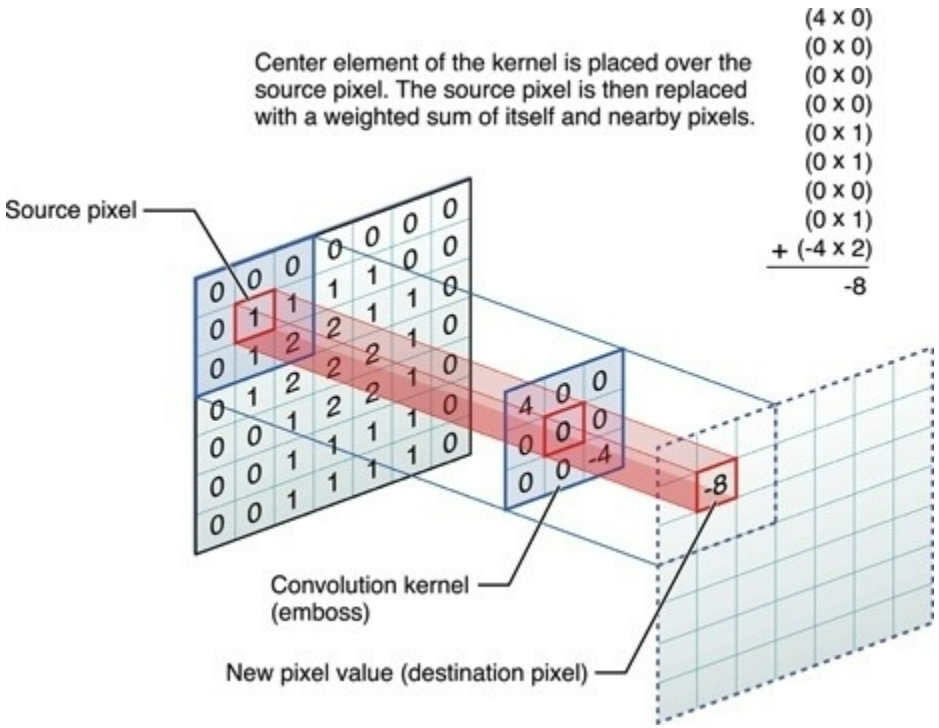
- Each layer acts a set of filters extracting important features
- Generally, after passing through several convolutional layers, the output passed through a fully connected dense network



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

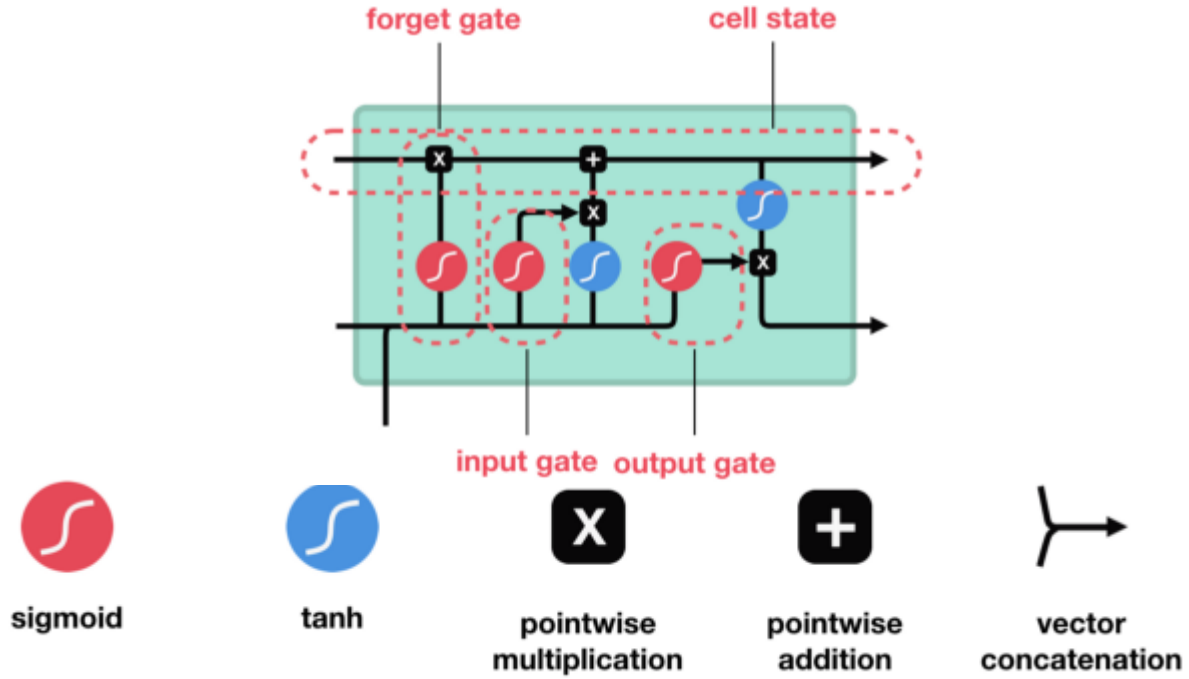
- Blends one function with another
- Operates in multi-dimensional spaces
- Output is multi-dimensional



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



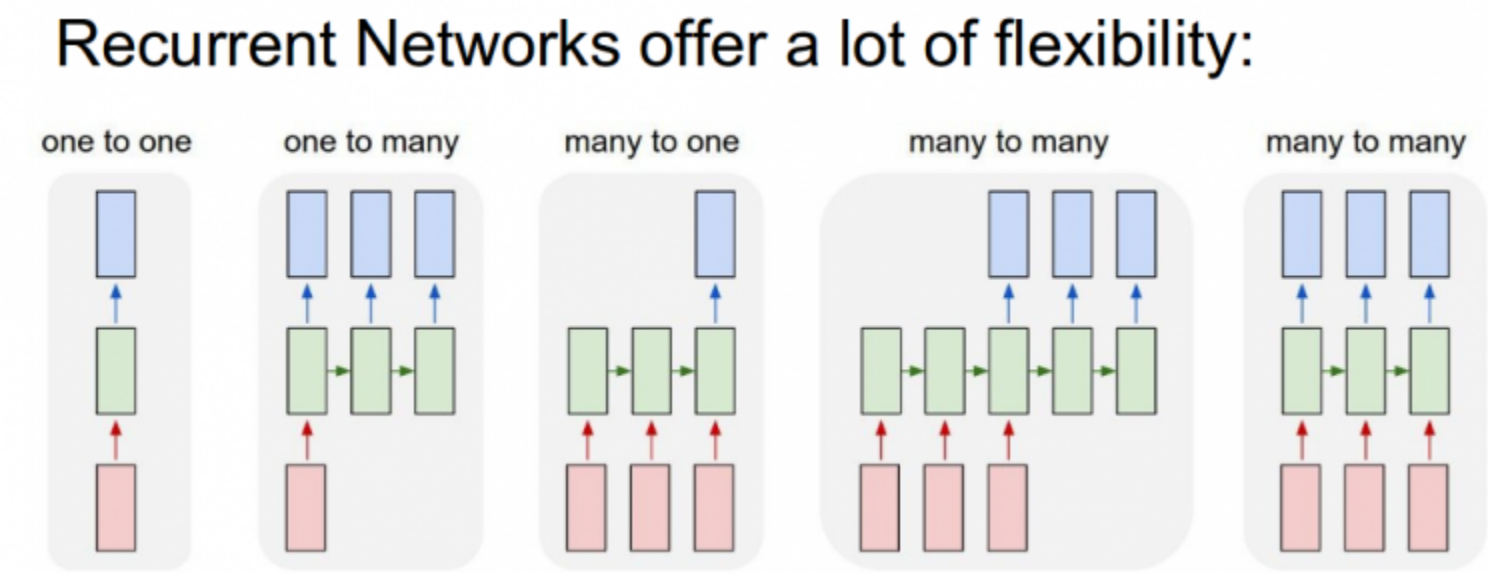
Sequence-to-sequence prediction

Many problems involving sequences and predicting sequences:

- Machine translation
- Question and answering systems

Generally use LSTMs to capture temporal dependencies

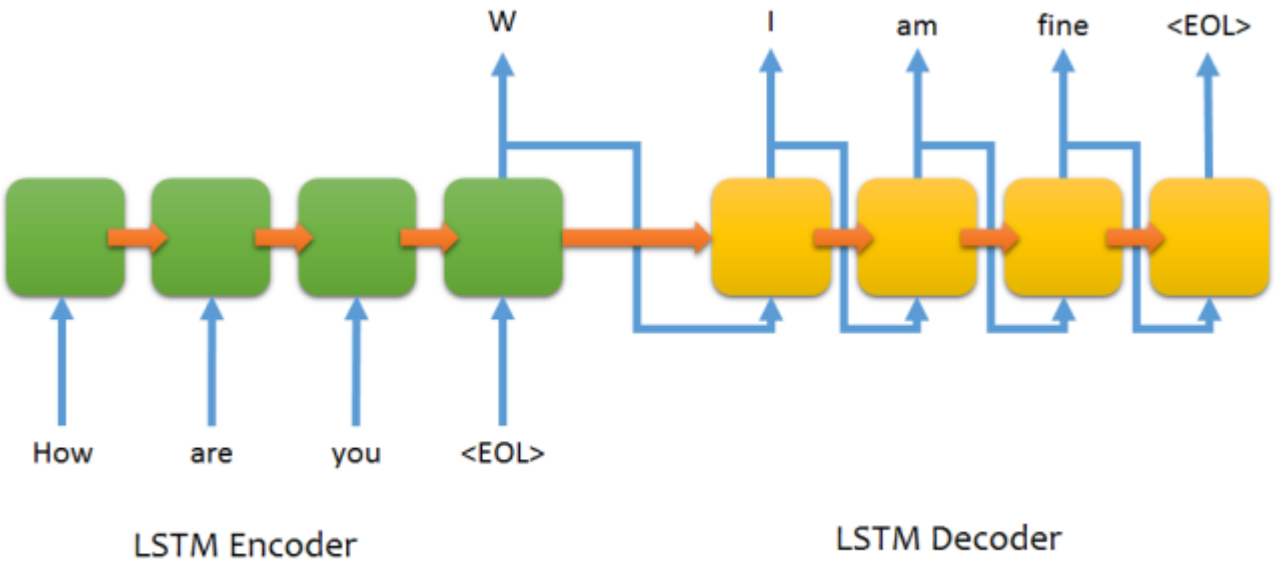
Can we cast video prediction as a sequence to sequence problem?



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

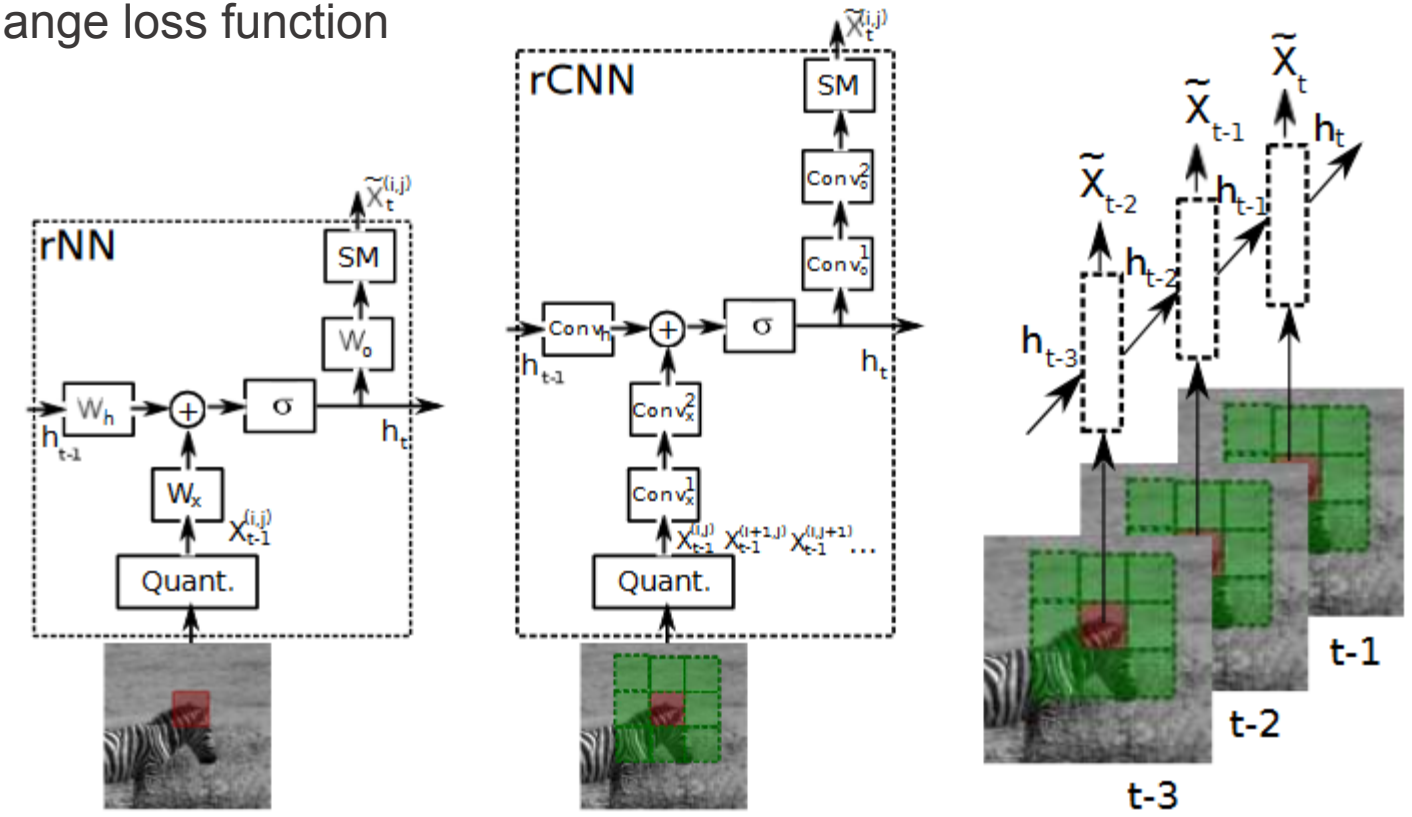


Extending sequence-to-sequence to video

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

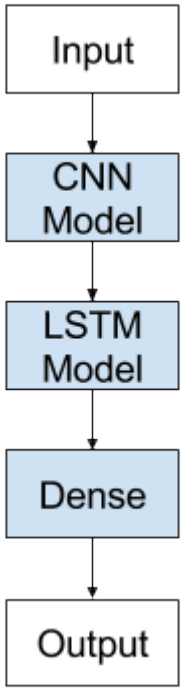


See: M. A. Ranzato, A. Szlam, J. Bruna, M. Mathieu, R. Collobert, S. Chopra **Video (language) modeling: a baseline for generative models of natural videos.** https://arxiv.org/abs/1412.6604

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

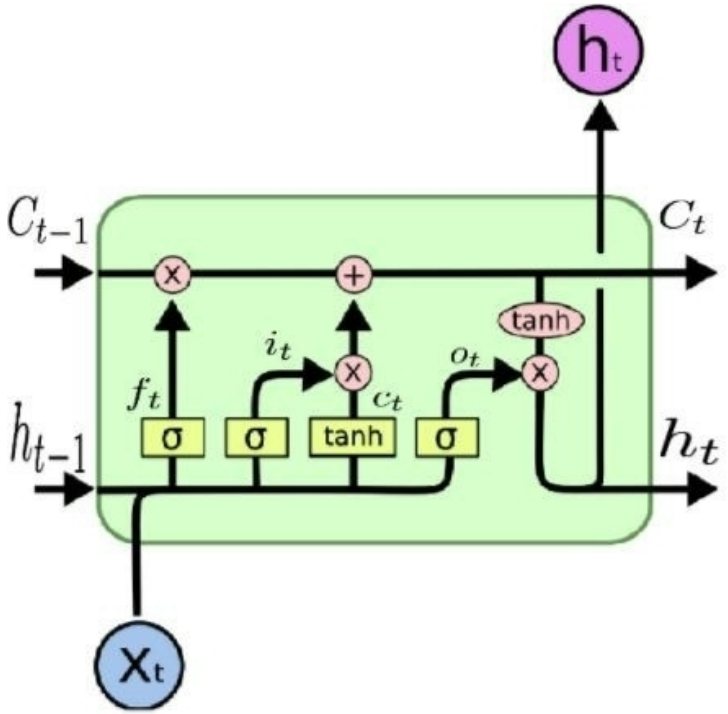
Precipitates the generation of the convolutional LSTM neuron

- **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

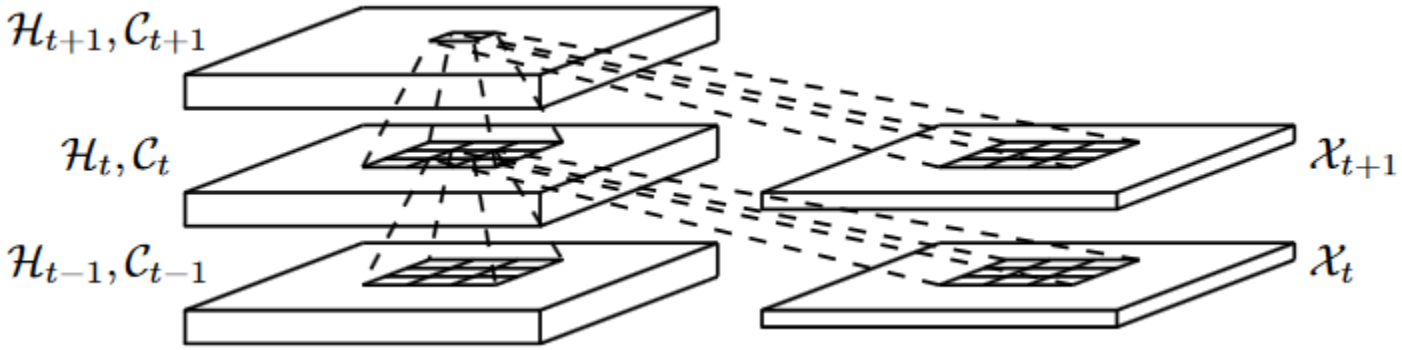
Models spatio-temporal relationships in the data

- 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)

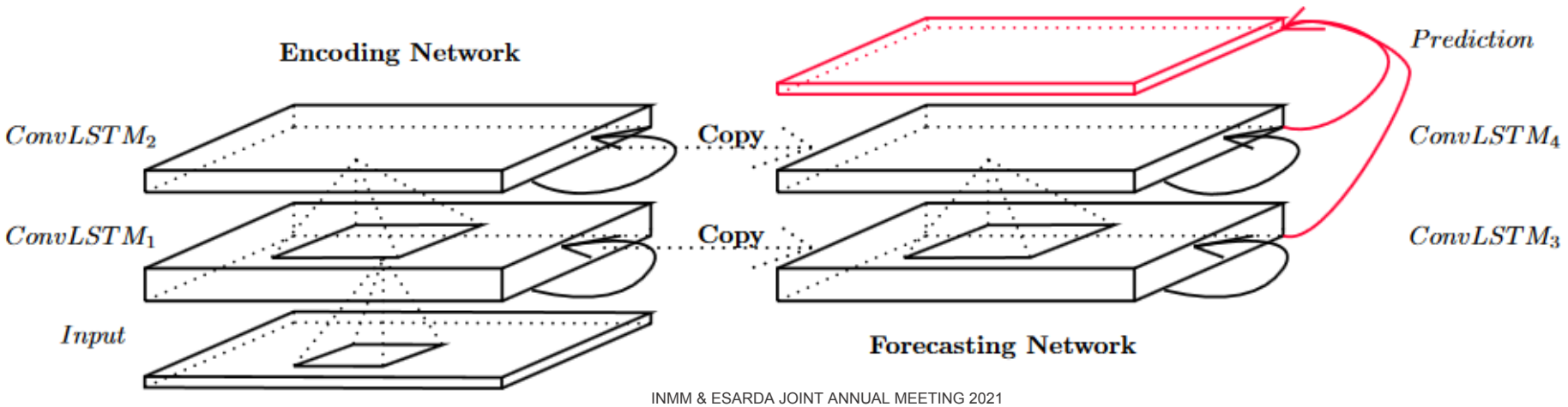


ConvLSTM – Shown another way

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



Use sequence to sequence encoder and decoder/forecasting portions



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