

# HAhRD

# DeepReconstruction

**Event Reconstruction in HGCAL sub-detector of CMS using Deep Learning**

**Abhinav Kumar**

**Mentors: G. Gilles, F. Beaudette, A. Lobanov, A. Satirana, A. Chiron**

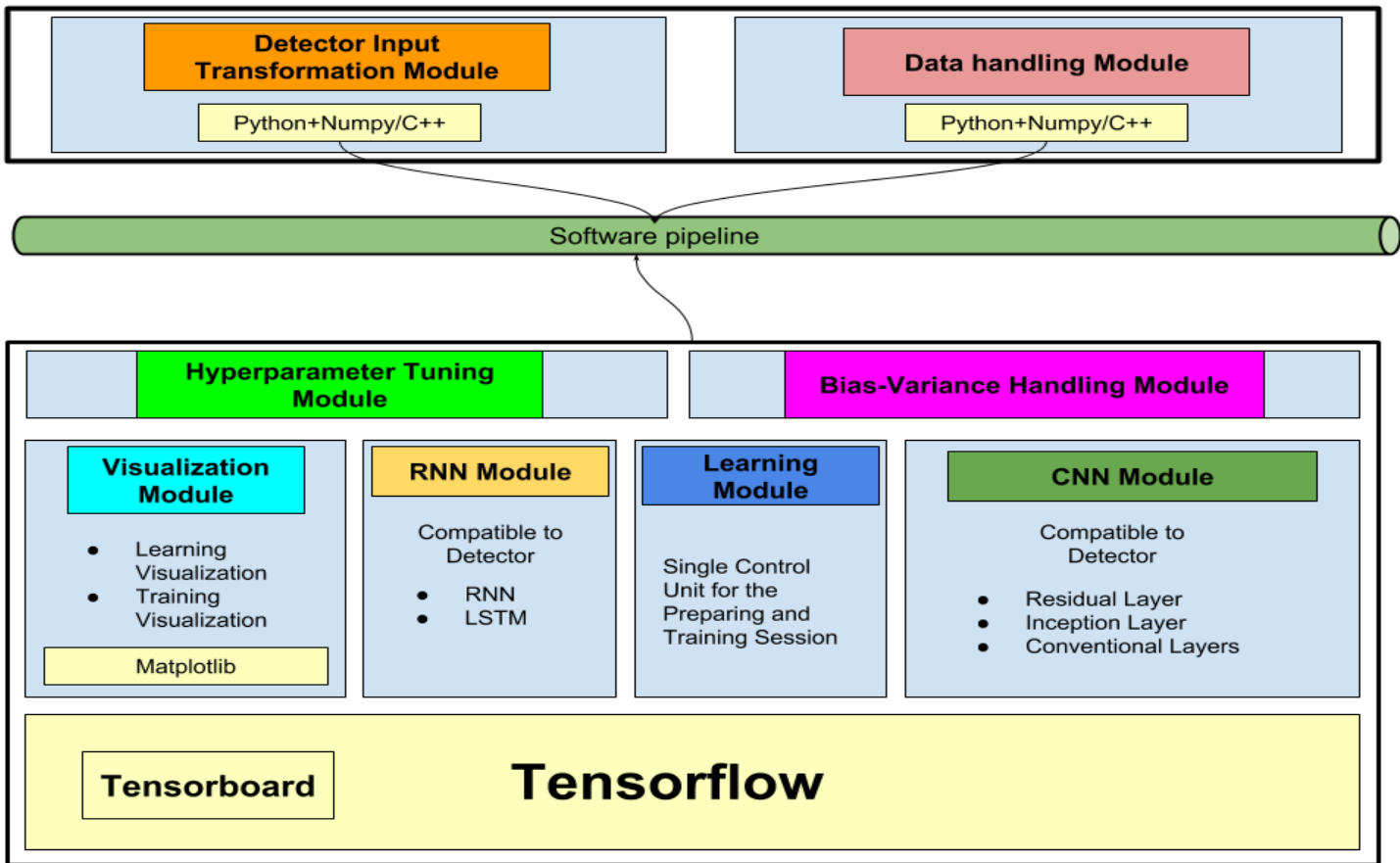
**Institute: Laboratoire Leprince-Ringuet**

**Organization: CERN-HSF (CMS collaboration)**

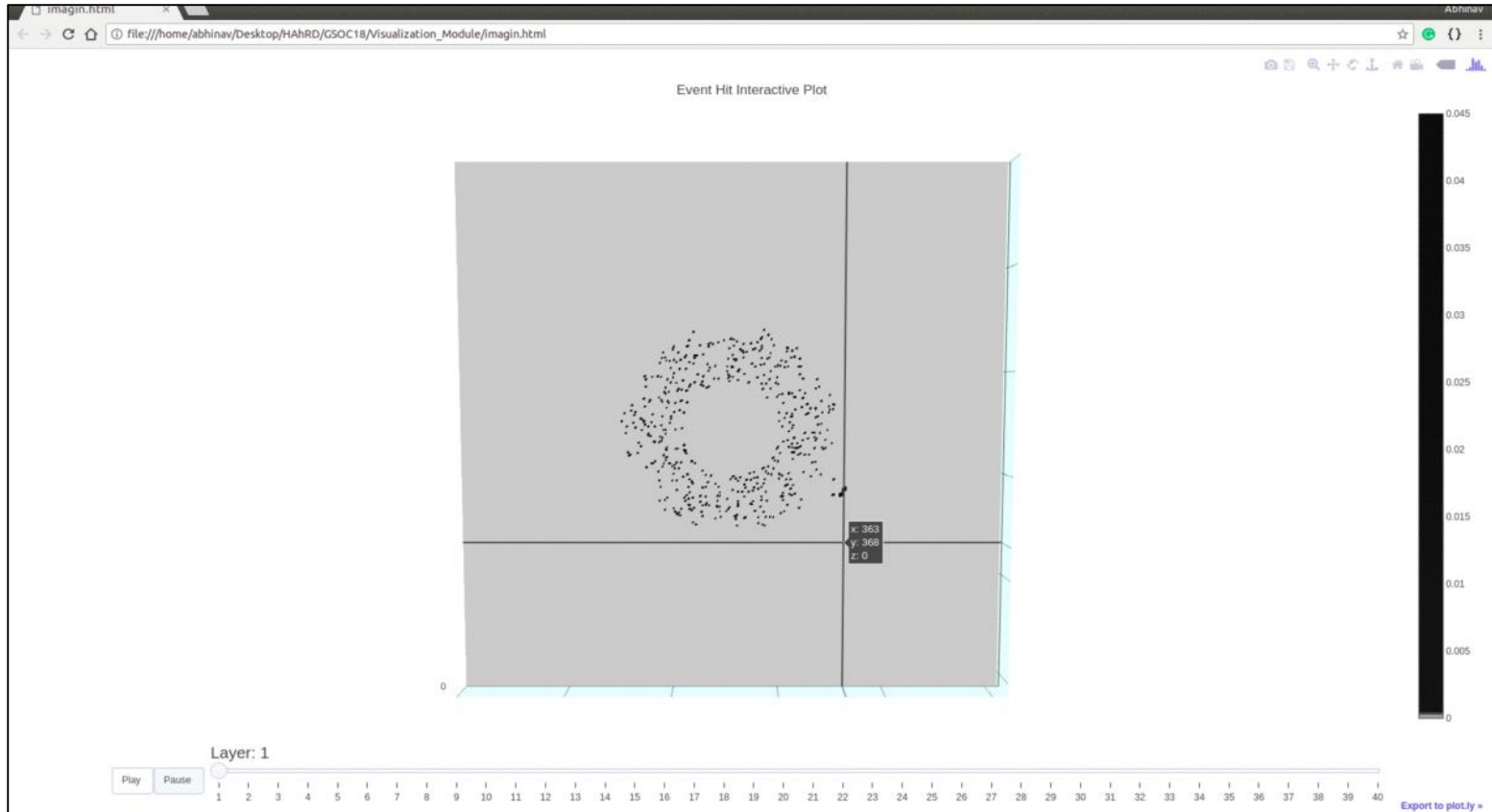
# Task Description

Aim of this GSOC-18.

1. Interpolation of “Hits” in the Detector.
  2. Development of CNN and RNN Modules.
  3. Development of Visualization Module.
  4. Multi-GPU training Strategy
  5. Making first training for verification.
-



# Interpolated 3D event-image Visualization



# First Full-Scale Training

# Training Set 3D “images”	= 380,440	
# Validation Set 3D “images”	= 20360	
# Epochs		=
9		
Time Taken for Training		= 2 days on
two		

Nvidia Tesla  
V100 GPUs

## Results

1. **Percentage Error in Energy:**  
1.86 % (T), 1.89 % (V)
2. **Absolute diff in Barycenter pos (x):**  
1.32 cm (T), 1.31 cm (V)
3. **Absolute diff in Barycenter pos (y):**  
1.28 cm (T), 1.30 cm (V)
4. **Absolute diff in Barycenter pos (z):**  
0.64 cm (T), 0.65 cm (V)

**Scale:** Each cell in detector is around 0.7 cm wide or larger and full detector is ~300 cm wide.

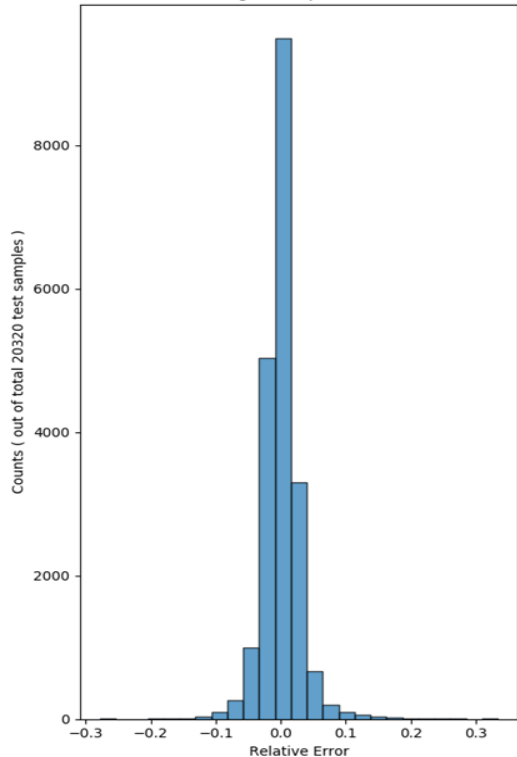
**Terminology:** T: Training Set, **Validation Set**

---

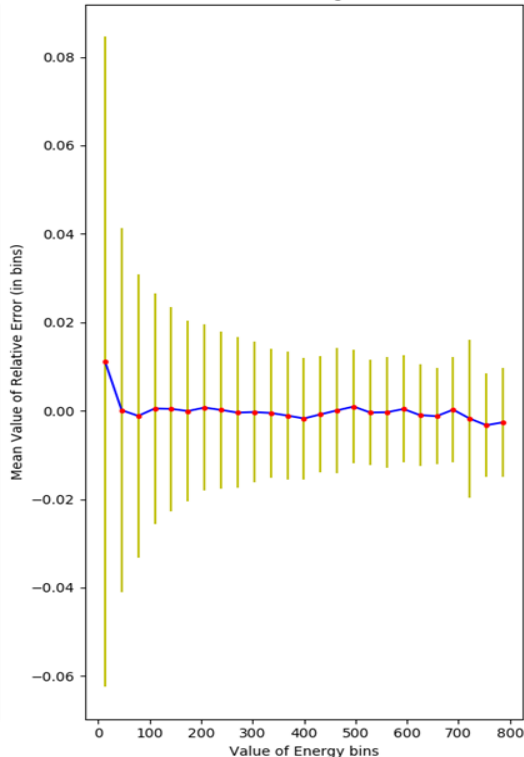
# Results 1a: Relative Error in Energy on Validation Set

Relative Error Histogram and Profile Histogram: Energy

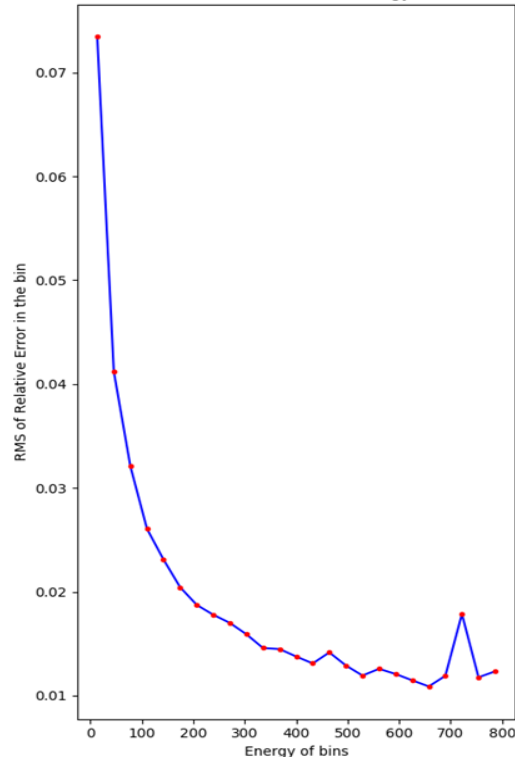
Relative Error Histogram:  $[(\text{prediction}-\text{labels})/\text{labels}]$



Profile Histogram

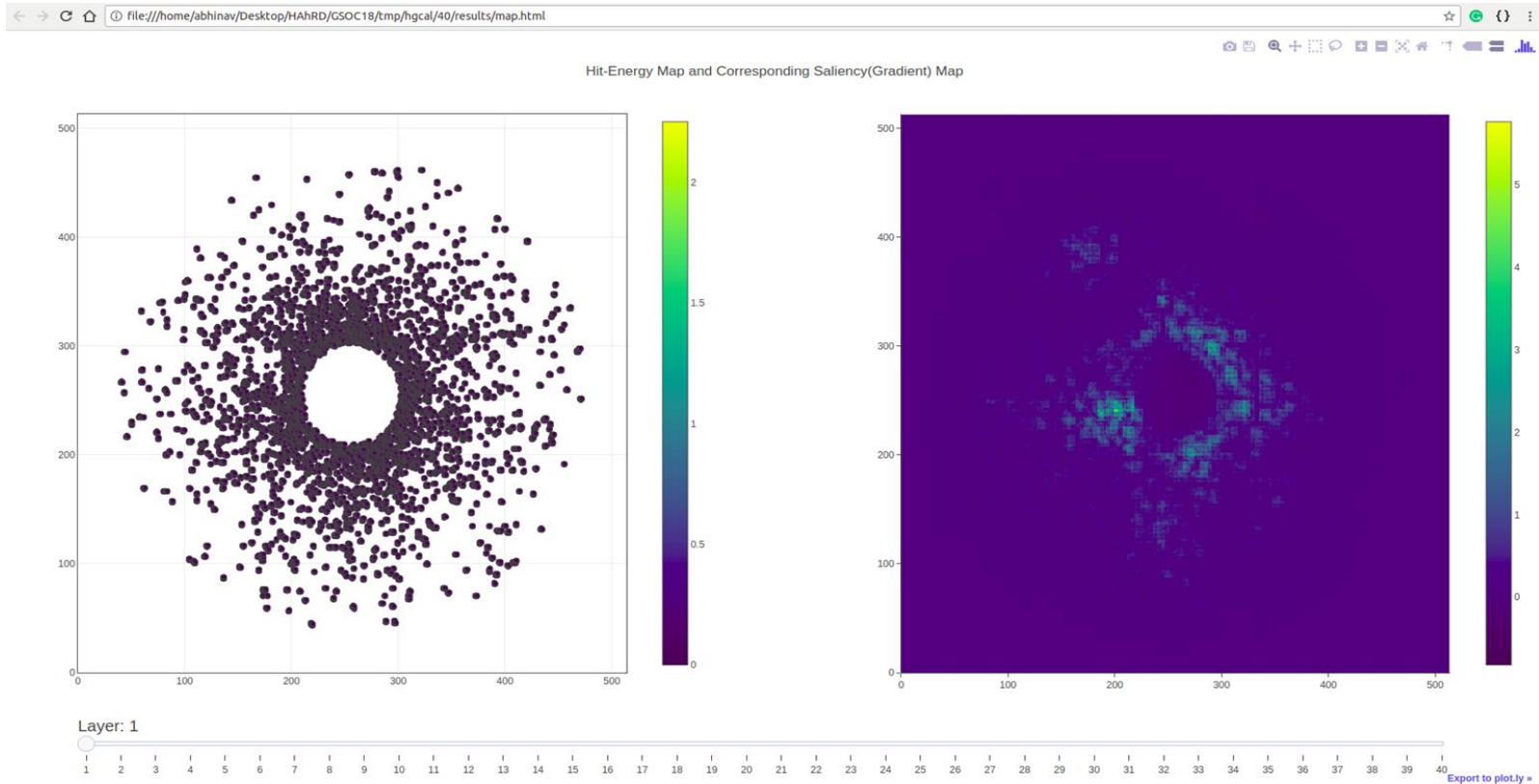


RMS of Relative Error in Energy Bins



Energy is in GeV  $\longrightarrow$

# Saliency/Gradient Map: What CNN is Learning?



\*made for Pile-up Events after training CNN on No-Pile Up Events

**THANK YOU**

---



**Backup Slides (for Q/A)**

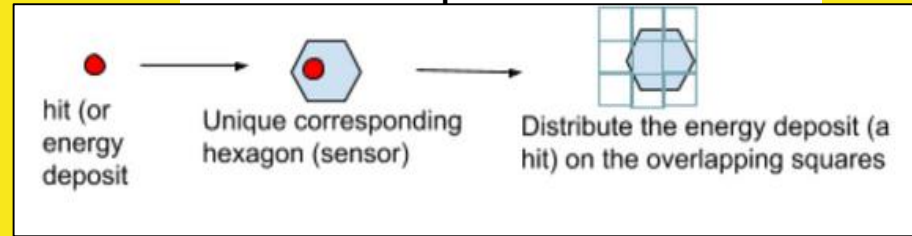
---

# Interpolation Module

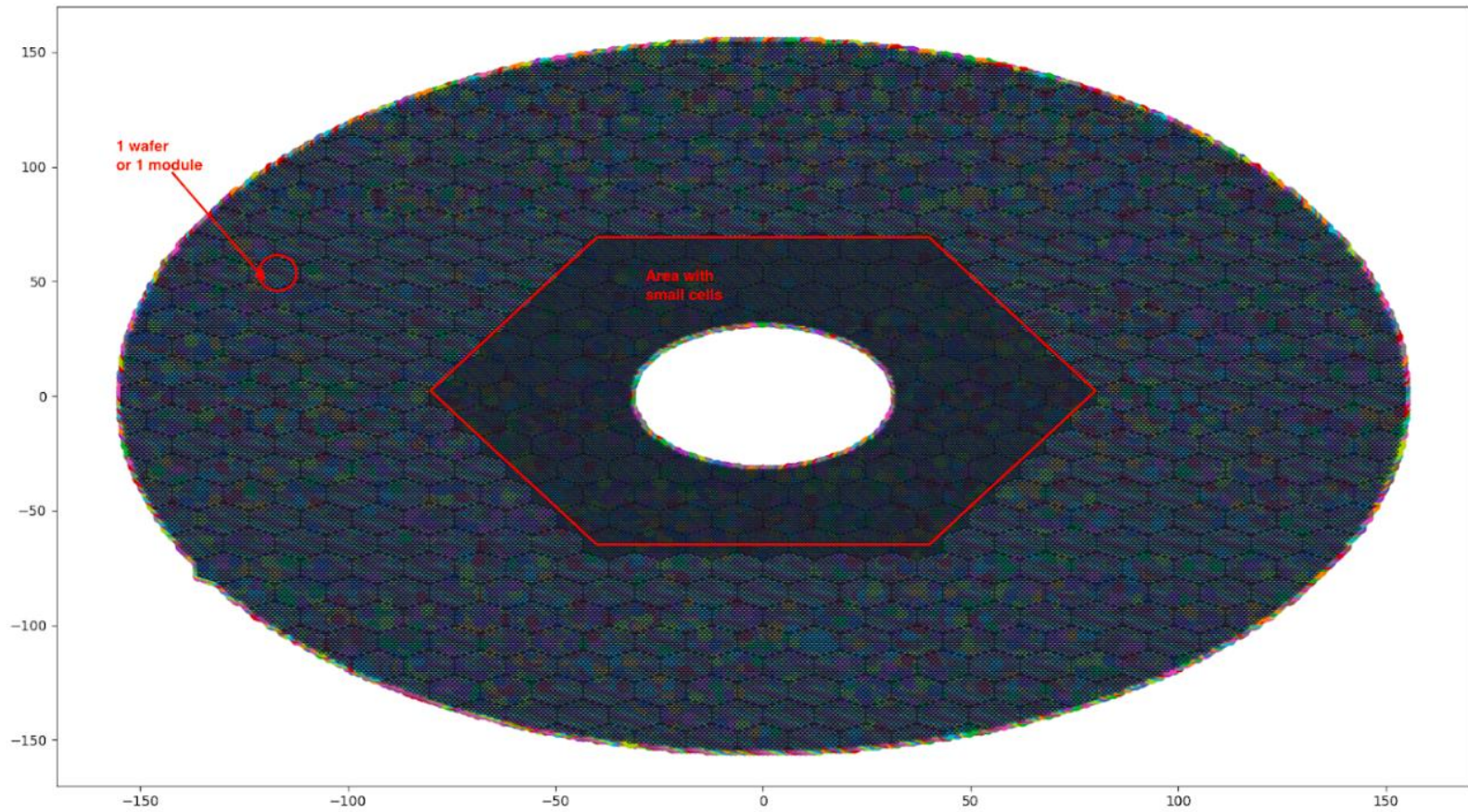
For interpolating from  
hexagonal geometry of detector  
to 3D “image”

## Task Covered

- Linear interpolation of events



- Saving image in tfrecords format for tensorflow
  - Target Creation for training the CNN models.
-



# CNN/RNN Module

Designing the important layers  
for 2D/3D convolution and  
sequence models

All the important layers for  
designing a Neural Network,  
along with the data IO pipeline  
was designed.

---

# Layers Supported in current Framework

## Convolutional Module

- **Simple Convolution** Layers
- **Max Pooling** Layers
- **Residual Layers** (2 version) based on the Residual Neural Networks
- **Inception Layers** based on the Inception Neural Networks

Every layers has their 3D counterpart. Also, regularization features like **Dropout, Weight Decay, and Batch Normalization** were infused in each layer.

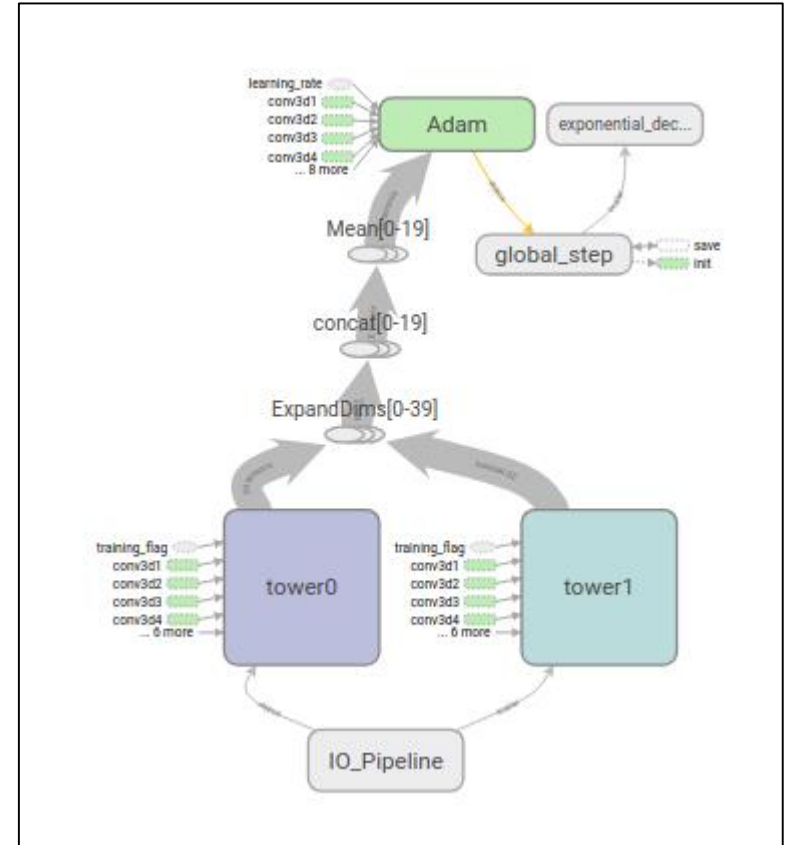
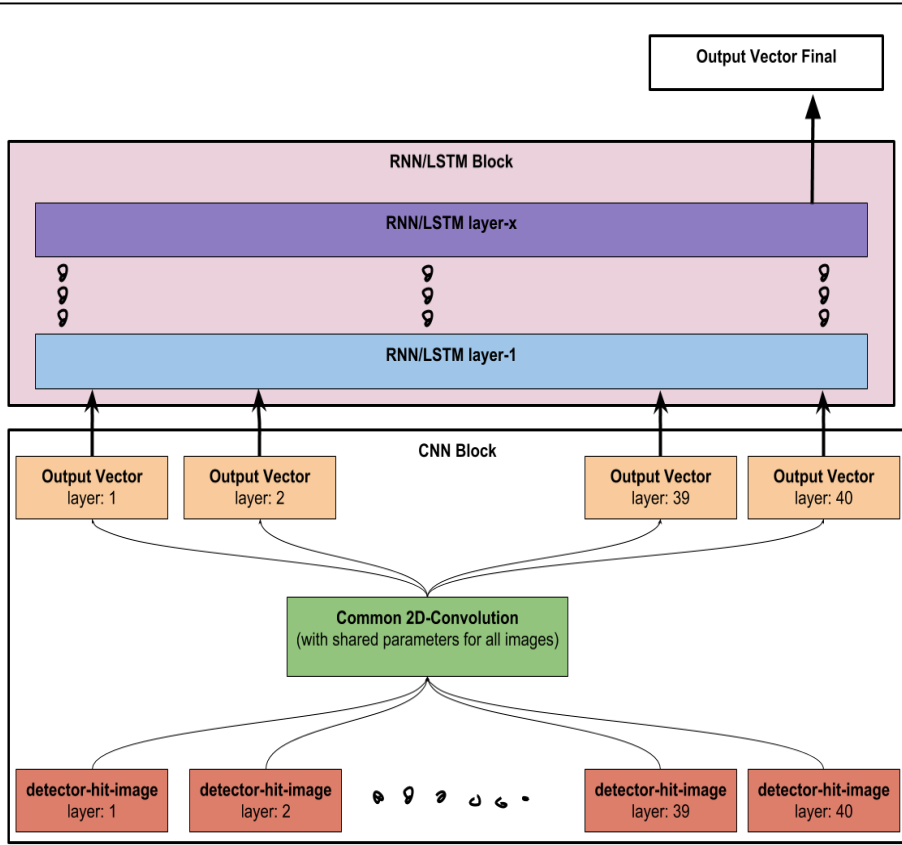
## Sequence Layer Module

- RNN layer
- LSTM layer

The particles travels through the detector (like evolution in space and time), so these layers were added for capture this.

Both the layer uses output encoding from CNN layers to propagate the information in time.

# Layers Schematic & Training Graph



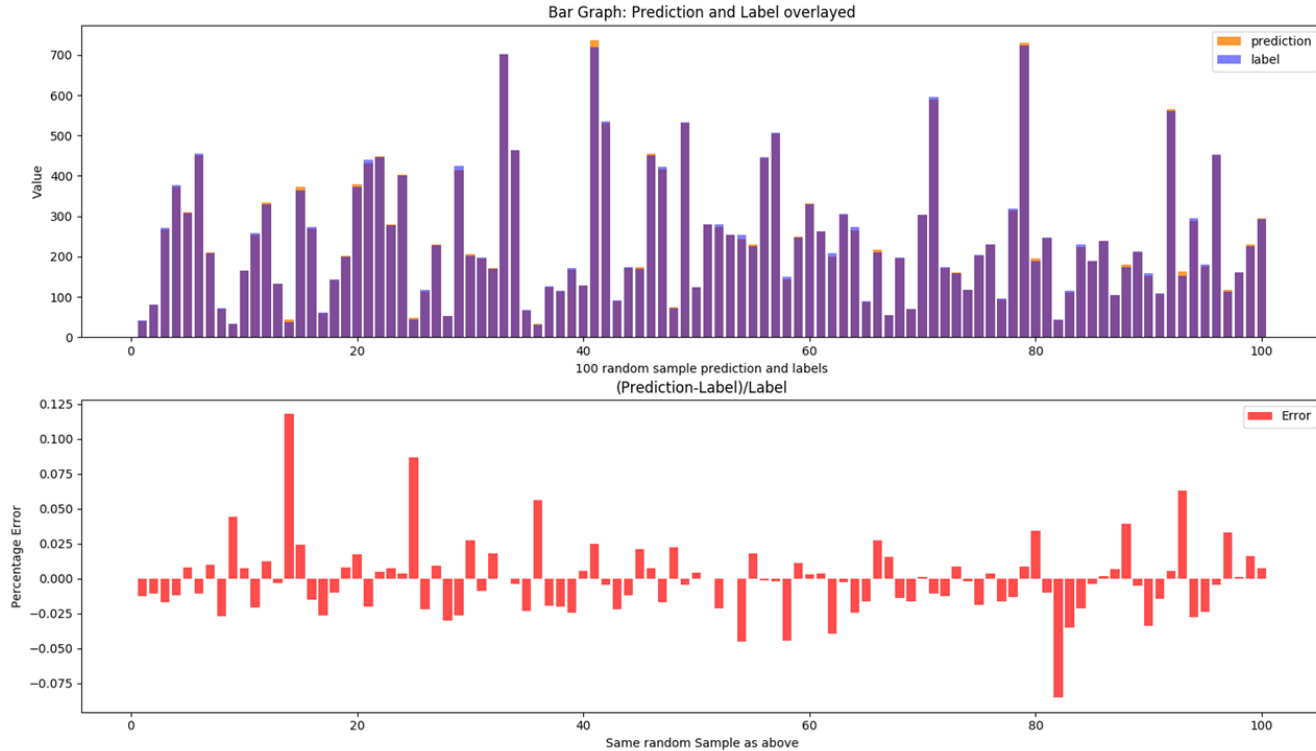
# Visualization

For visualization of Training  
and making Inference

- Energy-Hit Visualization
  - Training Accuracy and Prediction Visualization.
  - Saliency Map Visualization
  - Tensorboard integration of Training.
-

# Results 1b: Prediction Plot in Energy on 100 random

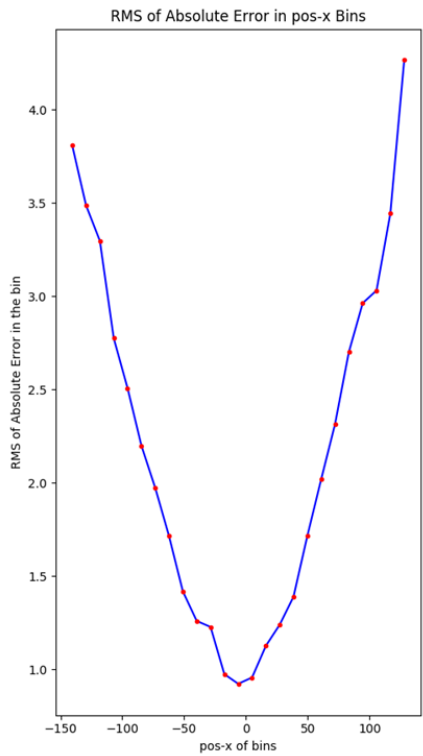
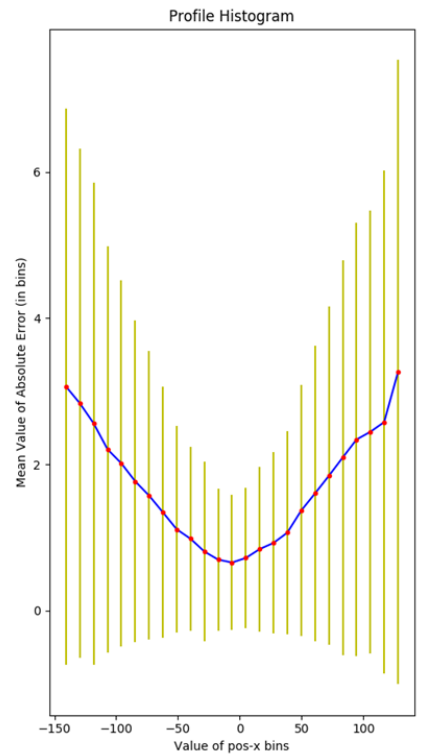
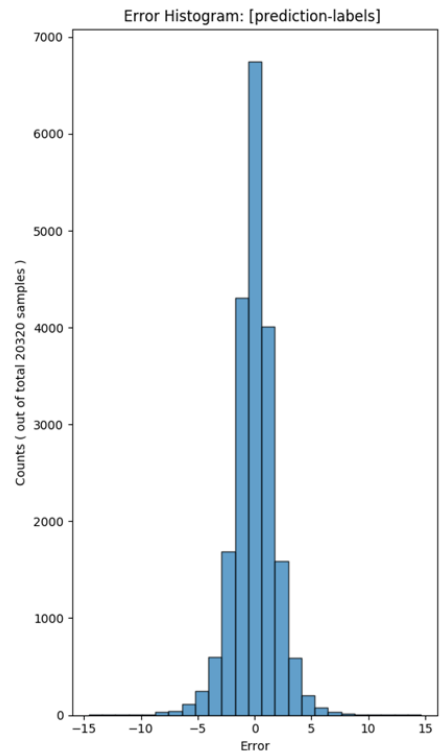
Energy Prediction and Label





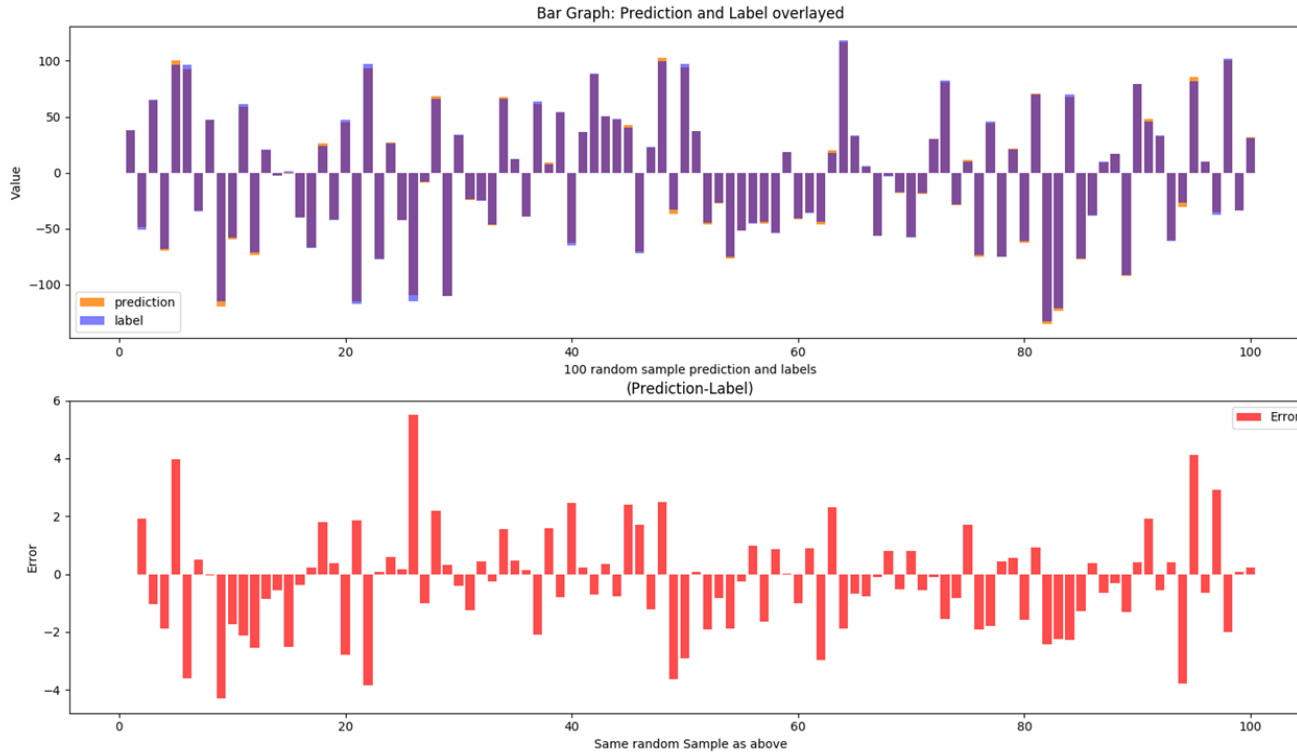
# Results 2a: Absolute Error in Barycenter pos-x

Relative Error Histogram and Profile Histogram: pos-x



# Results 2b: Prediction Plot in Barycenter pos-x

pos-x Prediction and Label



# Future Work

Learning from our mistakes  
and interpretation from Full-  
Scale Training

1. Training the model on Pile-Up events.
2. Making the IO Pipeline faster (removing the bottleneck)
3. Integration of more exotic layers like conv-RNN/LSTM layers, Visual Interaction layers
4. Better CNN interpretability using techniques other than Saliency Map.

..... and more

---