Deep Learning with JUNO

Thilo Birkenfeld

Juno DFG Meeting
Sep. 2017

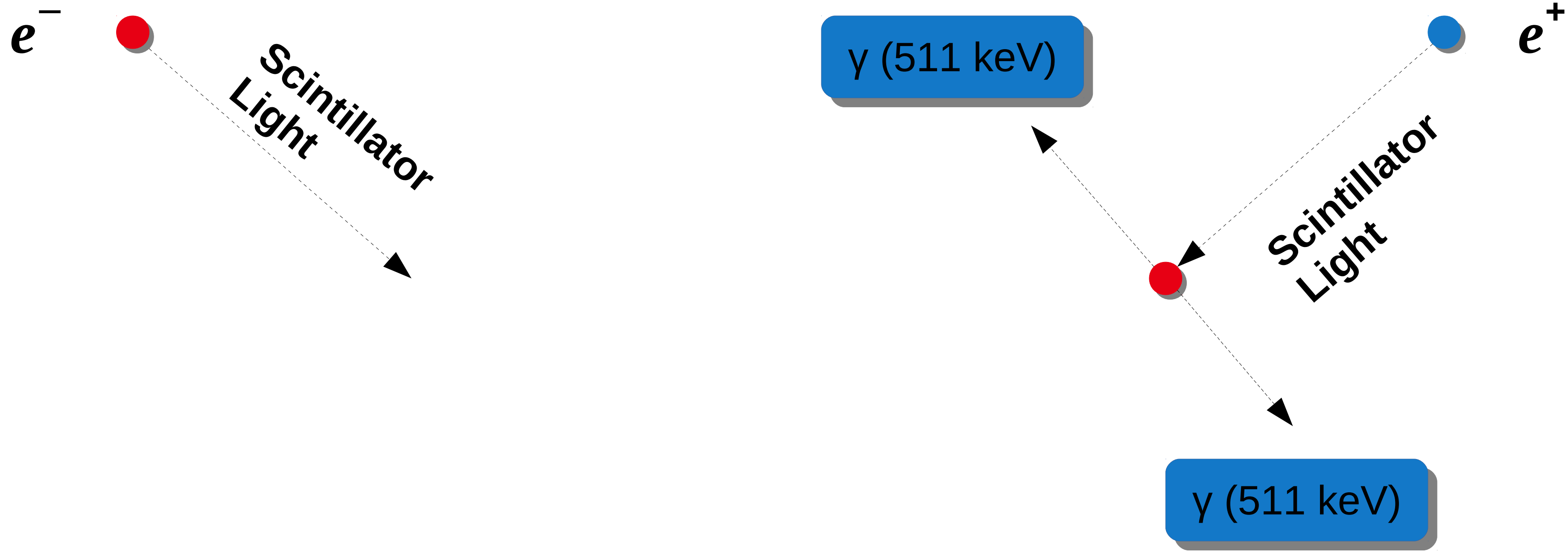
Goals

- Deep Learning can be useful in many aspects of JUNO:
  - Classification (e.g. event types)
  - Reconstruction (e.g. tracking, energy, …)
  - Many other tasks
- My studies focus on classification
  - Emphasize on e+/e- discrimination

Outline

- Electron and Positron events
- Image Generation
  - Issues
  - Implementation
- Classification
  - Current state

Electron and Positron events

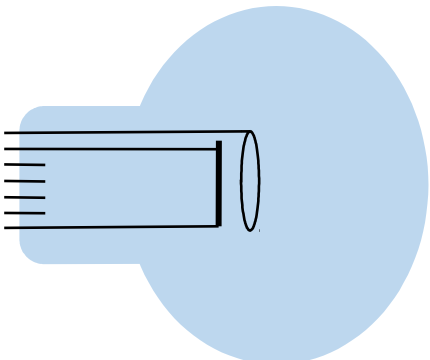


Implementation

- State of the art classifications:
  - Neural Networks (NN) with image recognition
  - Usually realized as Convolutional Neural Networks (CNN)
- Need to produce images from events with spatial information of event
  - PMT hit times
  - Mean hit time
  - Charge

PMT Data

Each PMT



Hit Time 1, Hit Time 2, ...

First hit, Mean hit, Stddev. hit

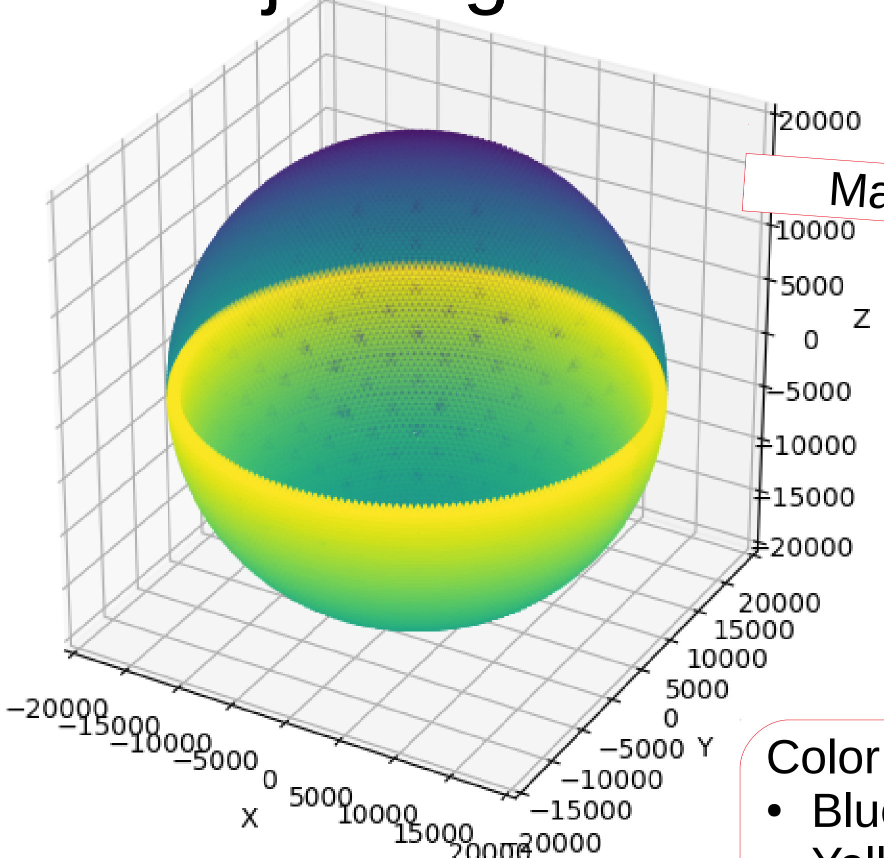
Gives us three channel with spatial information which can be interpreted as color channel for image.

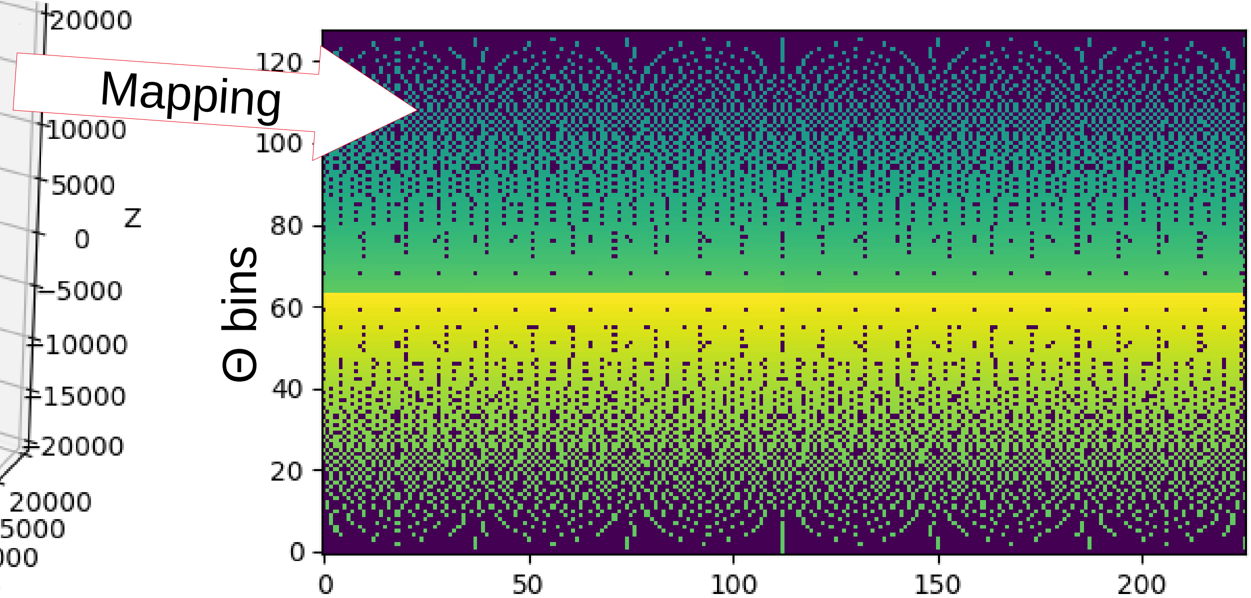
Image Generation - Issues

- NN’s work with inputs of arrays/tensors of some shape (dim1, dim2, ...)
- How to get spherical distributed PMT pixels into array like shape?

Image Generation – Mapping to Array

Projecting PMTs on (Θ, φ)-Space:





Color code:
- Blue = small ID’s
- Yellow = big ID’s
- Black = empty bins

Note: We use big PMTs only

Image Generation - Intermediate

- Simulated Electron event first hit times in (Θ, φ)-Space:

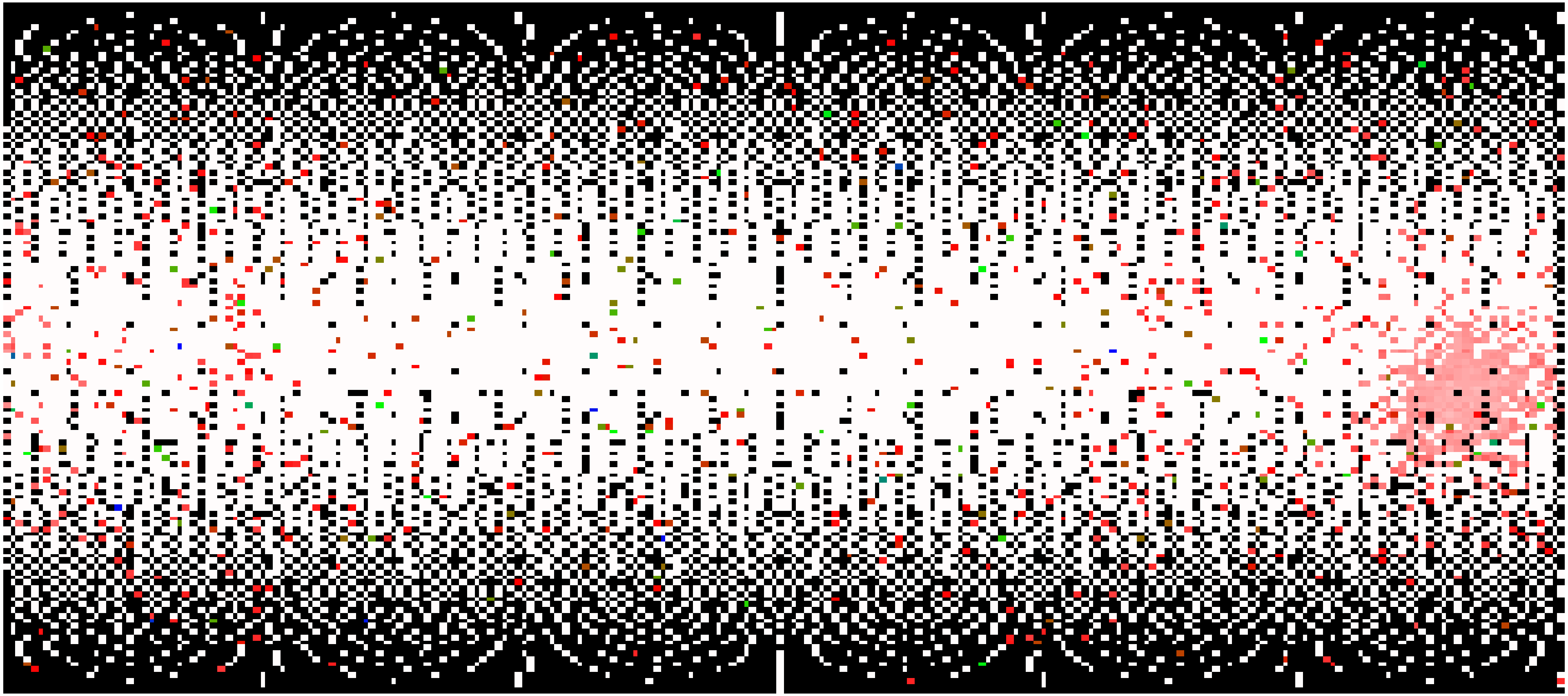


Image Generation - Intermediate

- Simulated Electron event first hit times in (Θ, φ)-Space:

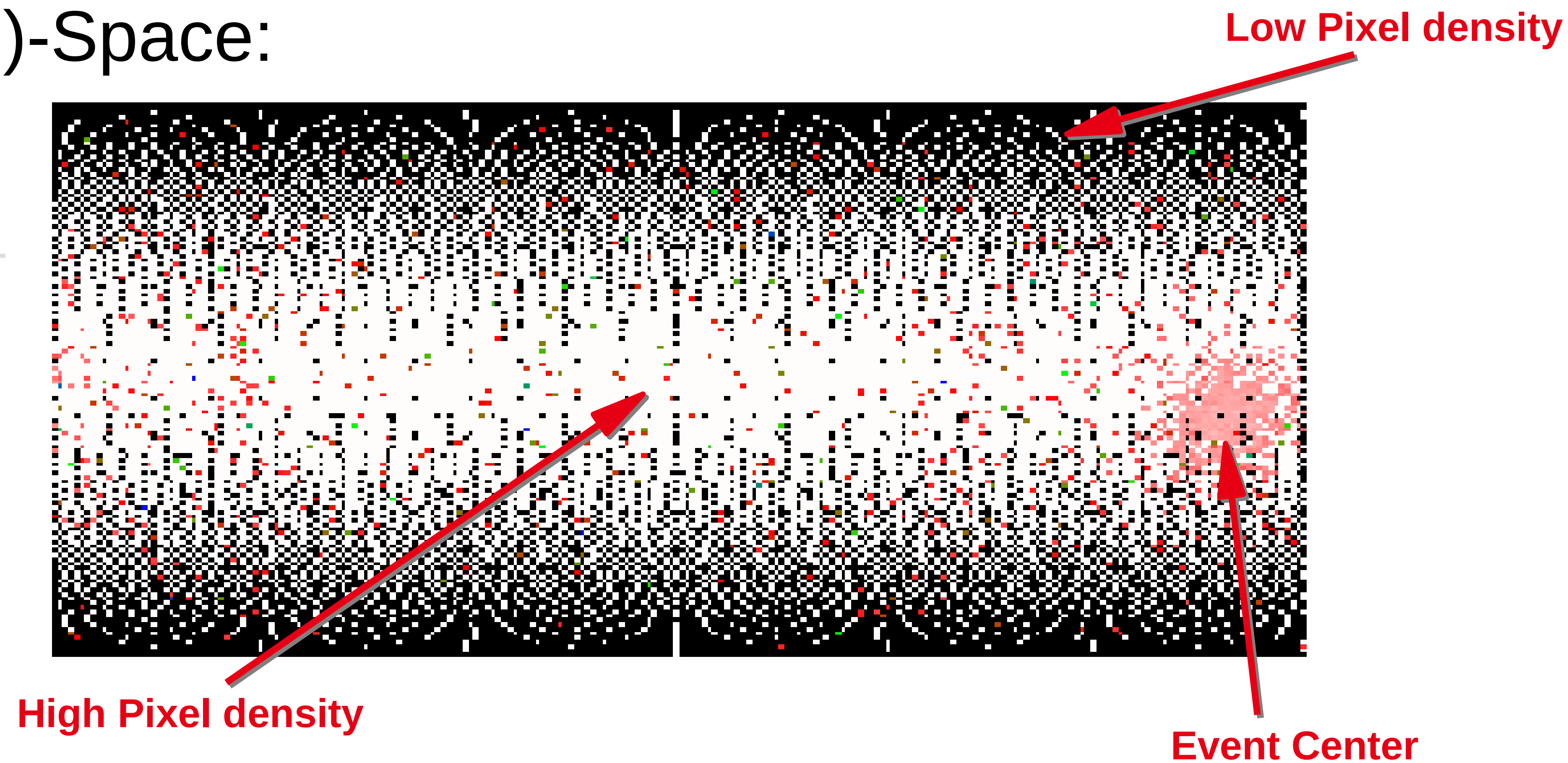


Image Generation - Improvements

- Our approach to solve issues:
  - Rotate each image to its center of mass
    - No Image splitting
    - Reduce influence of event position to images
  - Shift pixel together
    - Constant pixel density

Image Generation - Complete

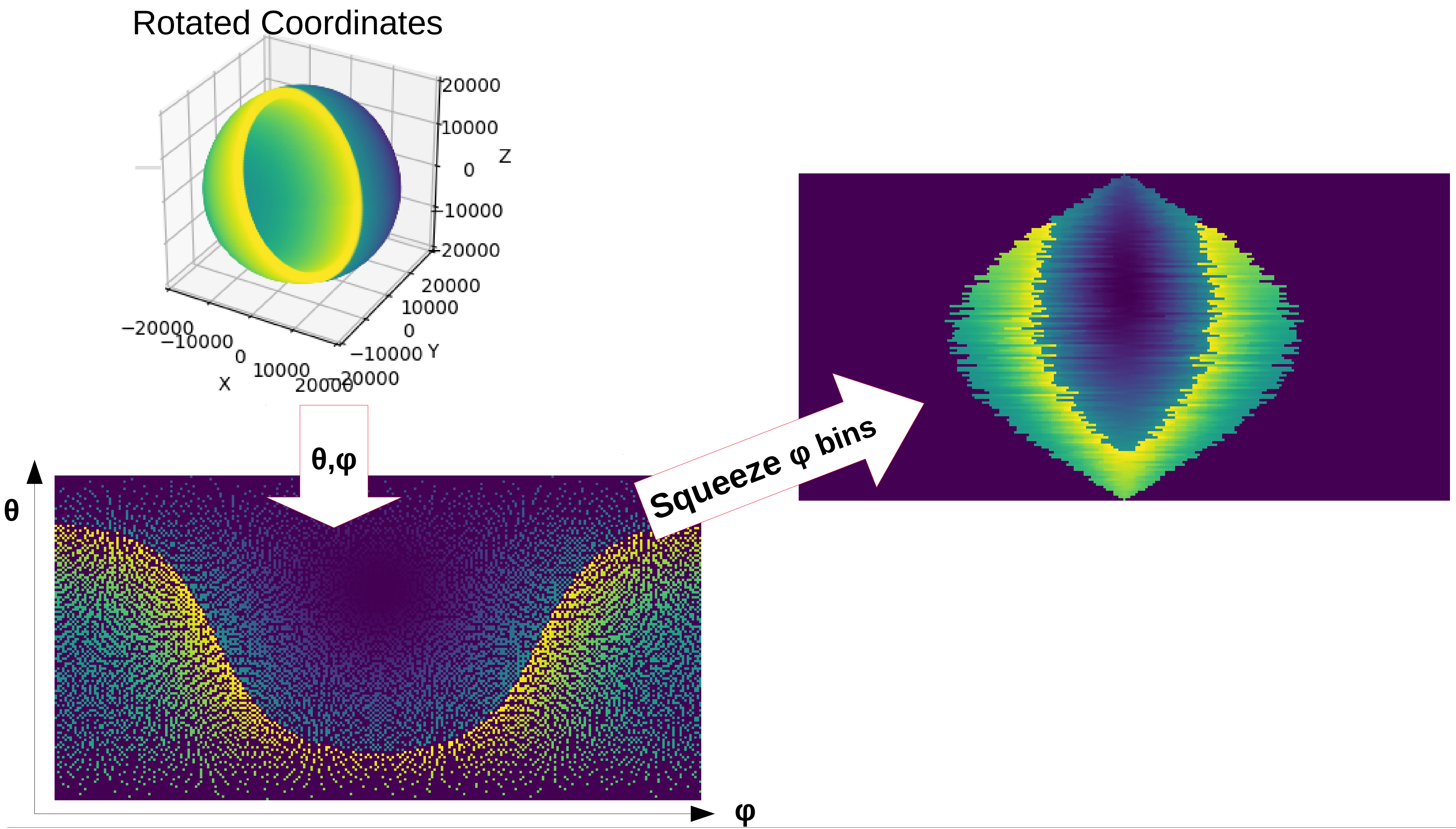
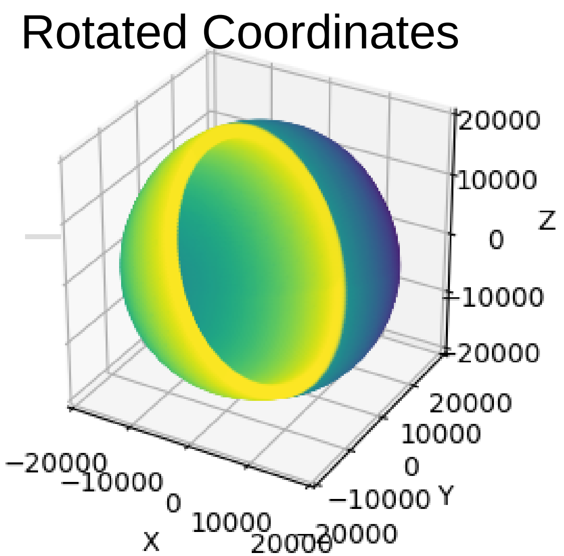
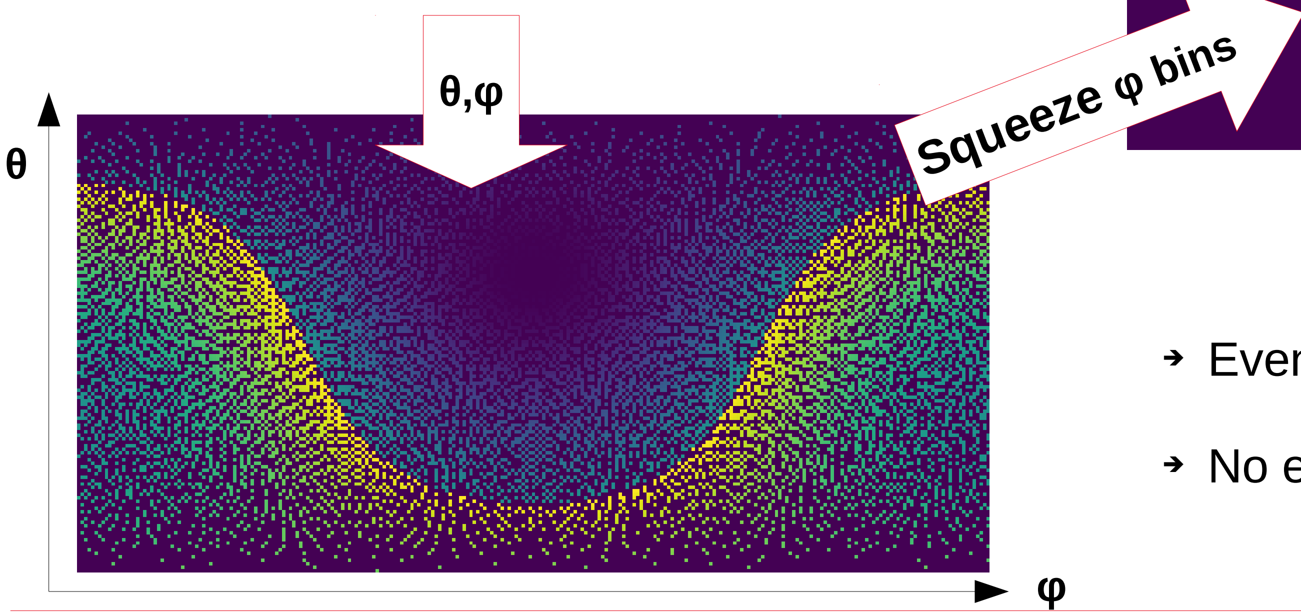
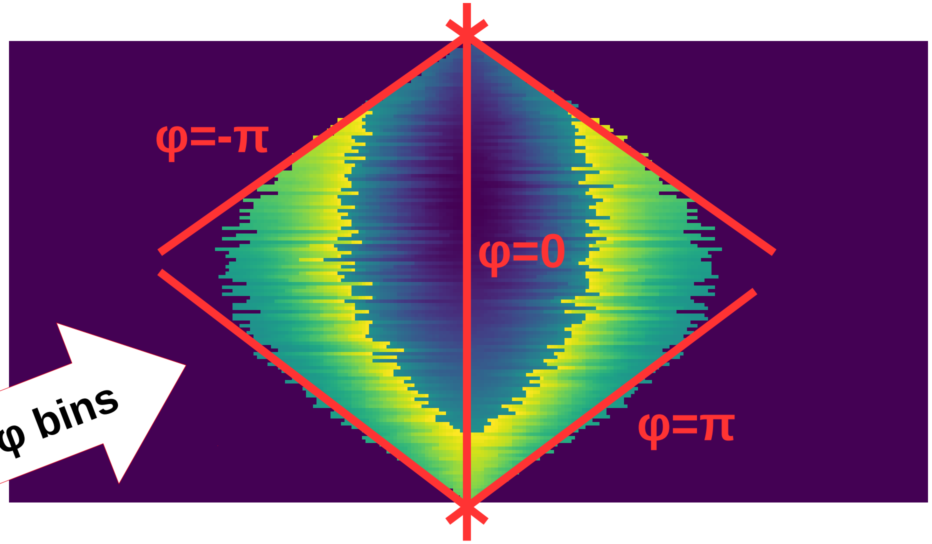


Image Generation - Complete



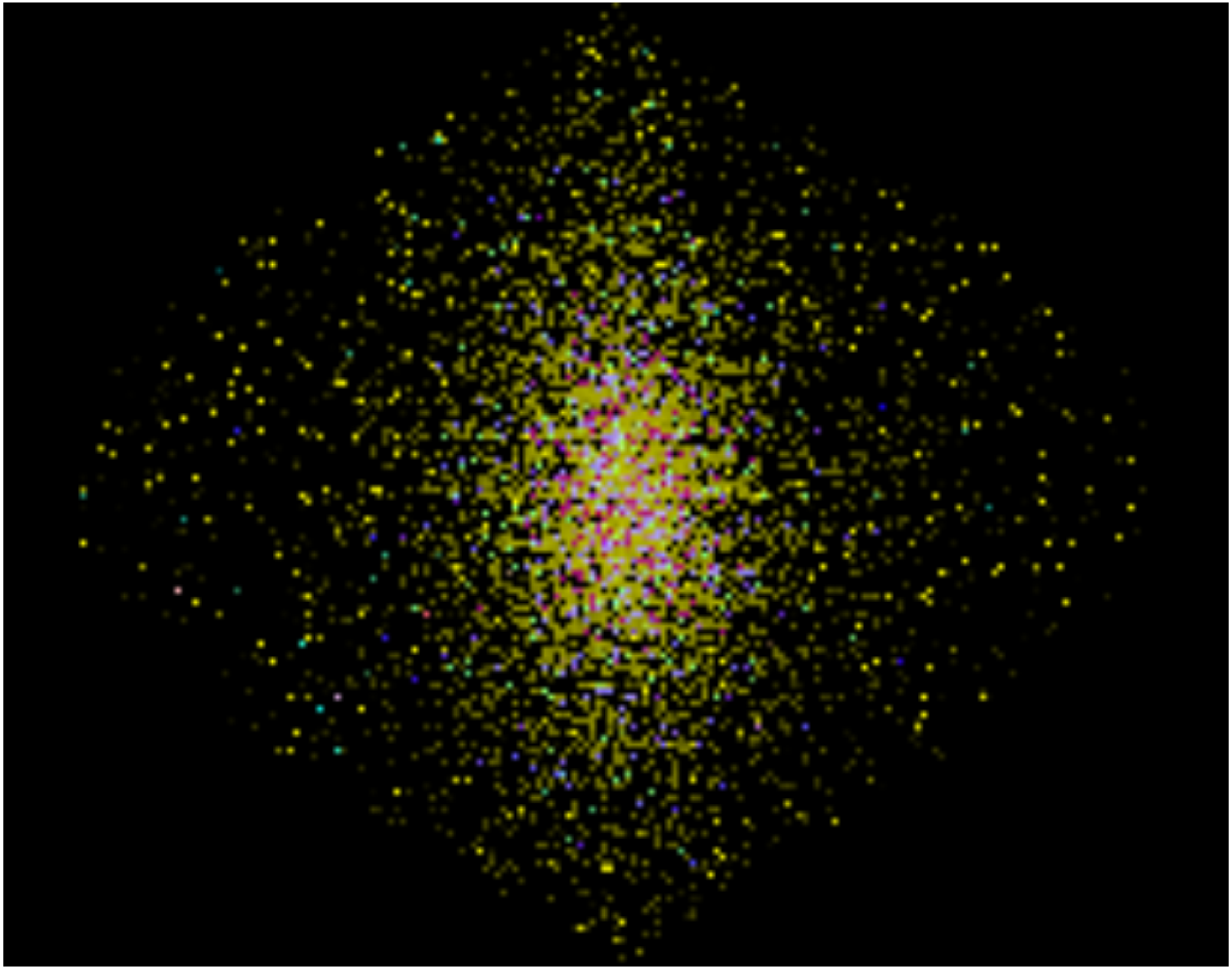




- Event always in the center of image
- No empty bins between pixel

Image Generation – Example Event

Complete processed Electron Event 5 MeV

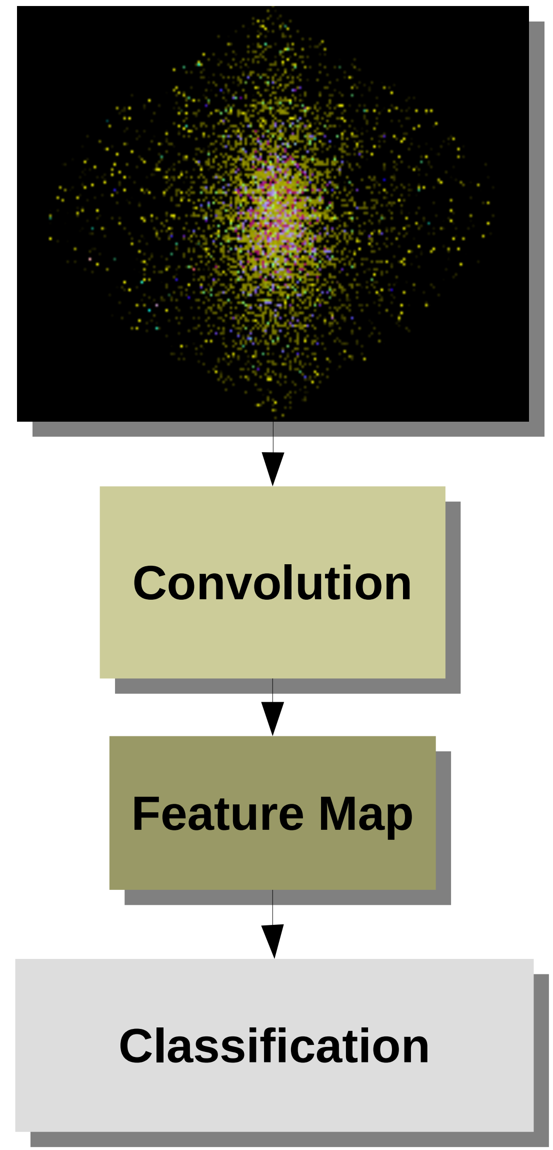


RGB Color Channel:
Red = first hit time
Green = mean hit time
Blue = stddev hit times

Classification

- Training Convolution Layers of deep networks is costly
- Especial with lots of data (3.6 million uniform distributed electron+positron events)
- Try using a high performing pretrained net as base
  - ResNet 50 for example

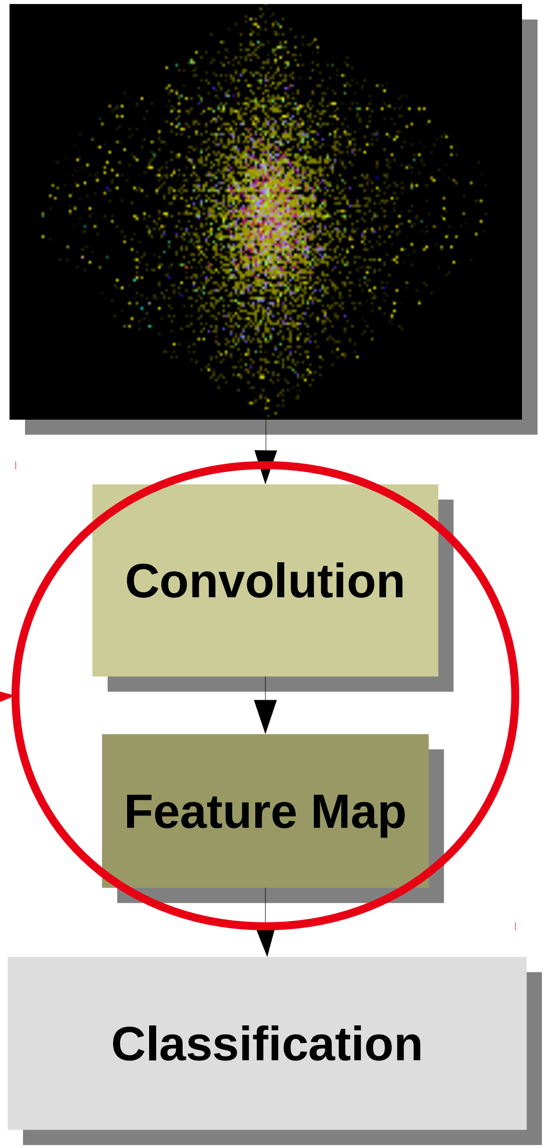
Typical CNN:



Classification

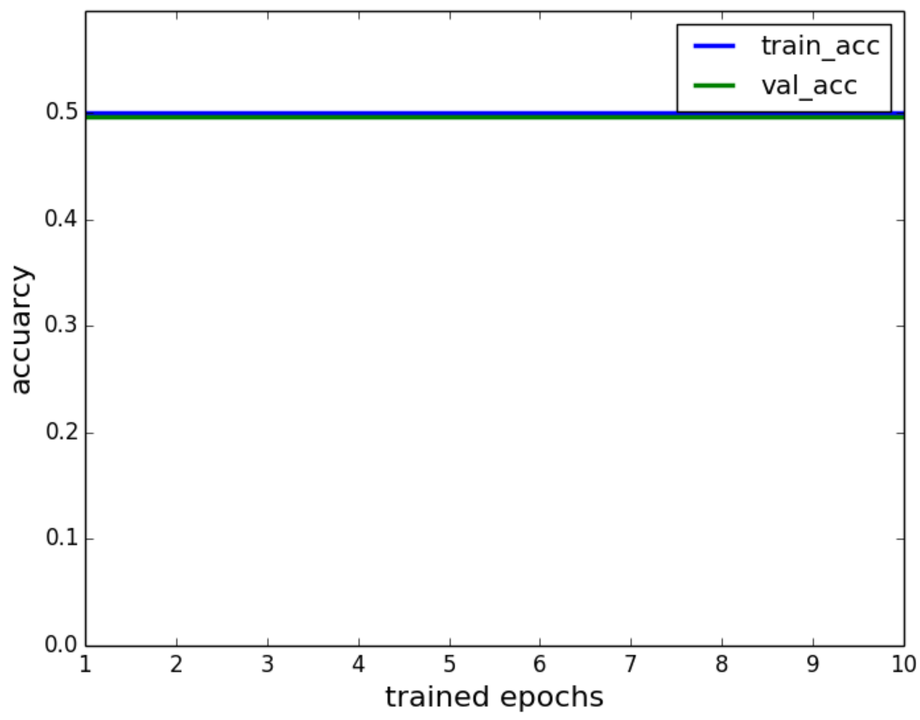
- Training Convolution Layers of deep networks is costly
- Especial with lots of data (3.6 million uniform distributed electron+positron events)
- Try using a high performing pretrained net as base
  - ResNet 50 for example

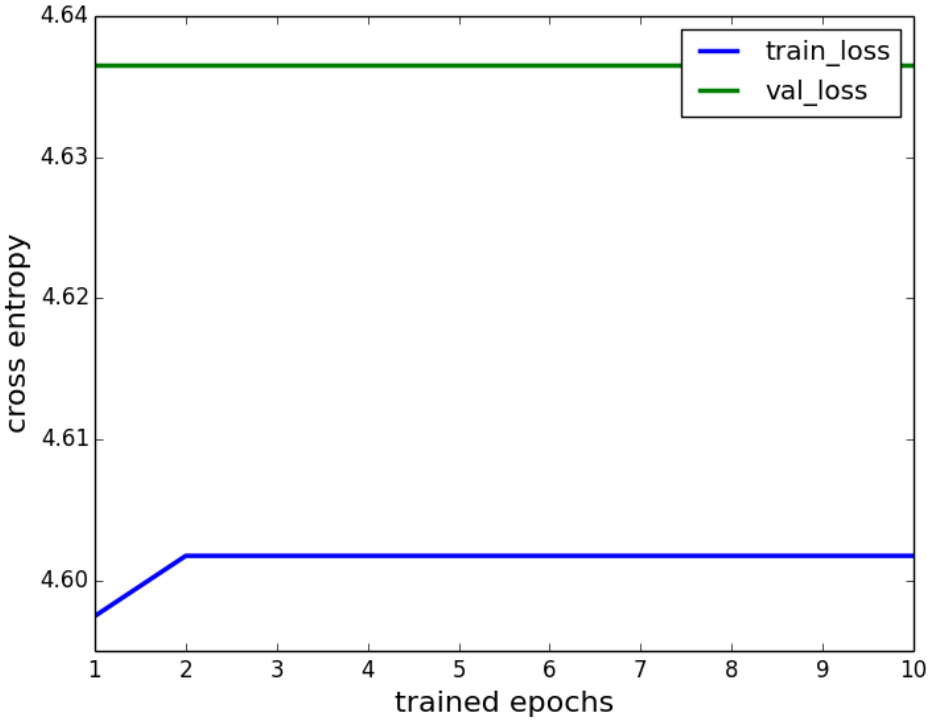
Typical CNN:



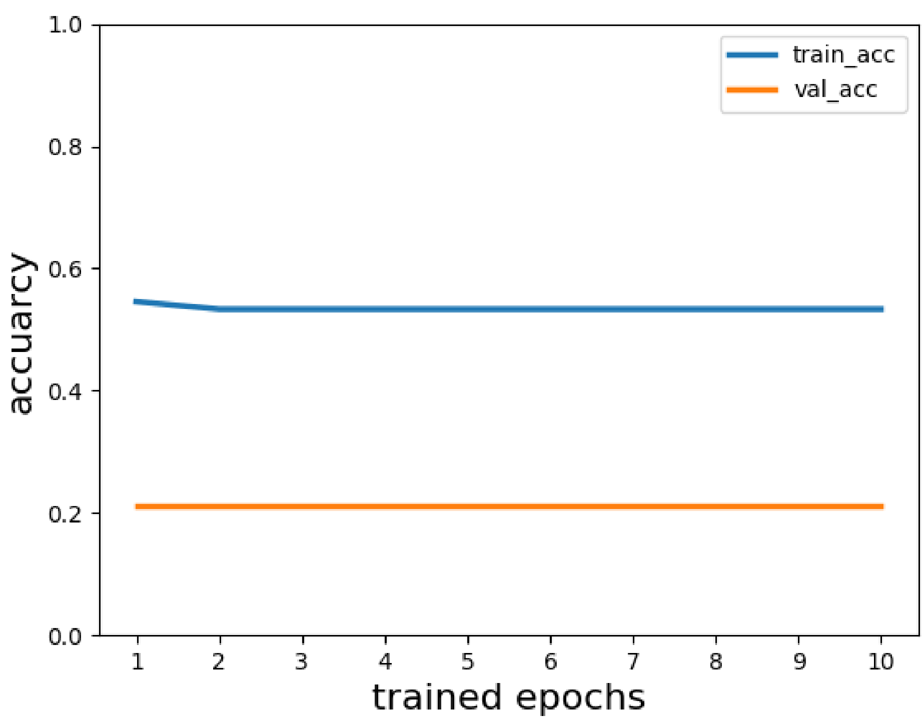
Positrons vs. Electrons

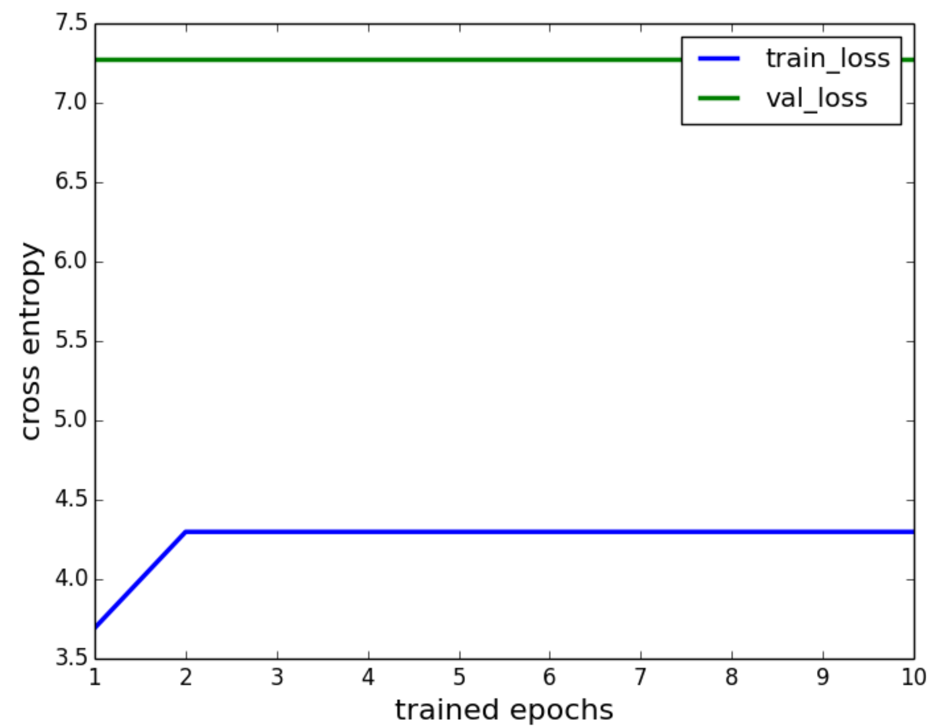
Training results with sample of 100k events





Muons vs. Electrons





- Bug in the program
- ResNet 50 not able to distinguish shape differences

Thank you for your attention!

Contact:

Thilo Birkenfeld
III. Physikalisches Institut B RWTH Aachen University

thilo.birkenfeld@rwth-aachen.de

Gefördert durch
DFG Deutsche Forschungsgemeinschaft

Categorical Cross Entropy

- Categorical Cross Entropy is minimized for training

$$J(\theta) = -\frac{1}{n} \sum_i \sum_j \mathbf{y}^i \log\left(\mathbf{y}_{\mathrm{model}}(\mathbf{x}^i|\theta)\right)$$

- $\mathbf{y}_{\mathrm{model}}(\mathbf{x}^i|\theta)$ is probability for class as predicted by NN

- $\mathbf{y}^i$ = 1 for true class, 0 else