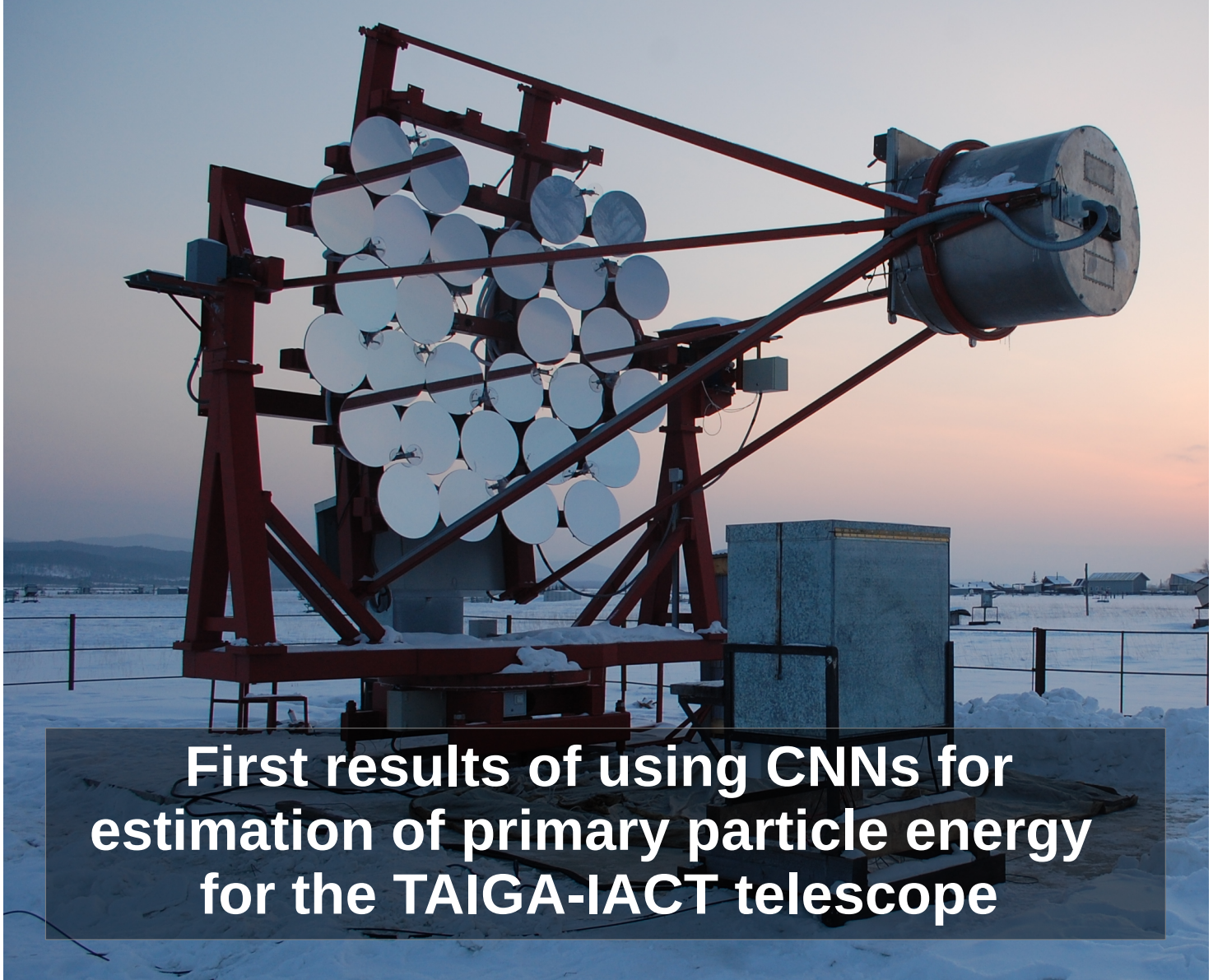
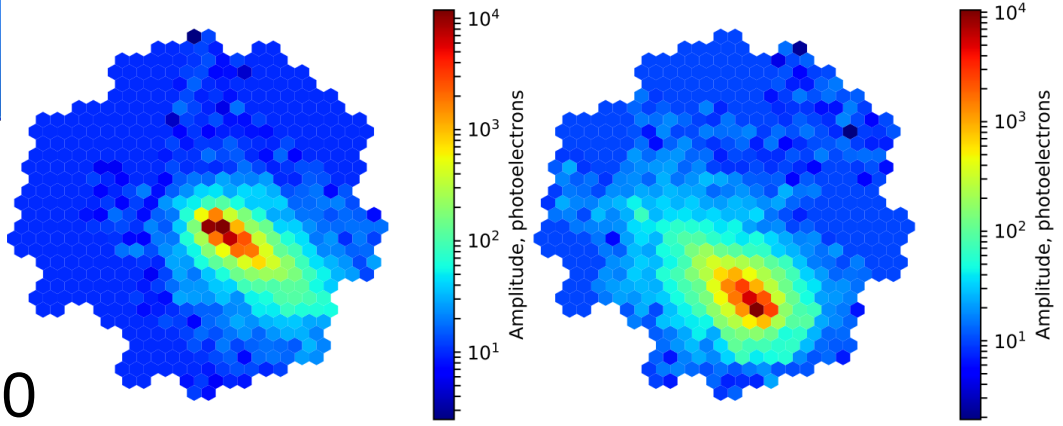


First results of using CNNs for estimation of primary particle energy for the TAIGA-IACT telescope

Introduction

- The first efforts to estimate energy of primary particles using CNN
- Based on small dataset (MC) with "soft" cleaning
- Framework: TensorFlow

The Dataset

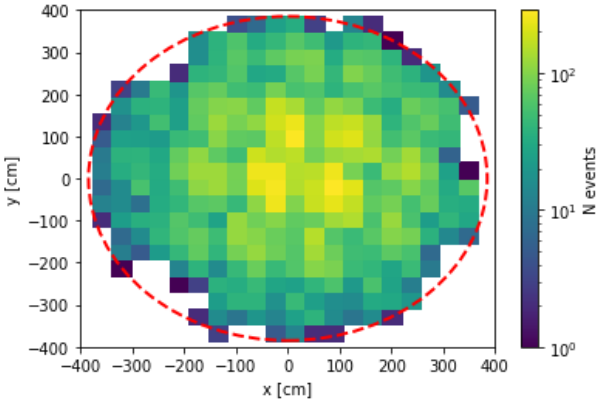


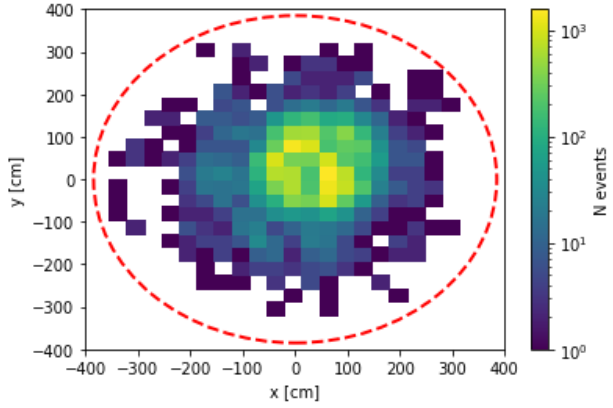
- Events: 40768
- Protons/gamma: 20238 / 20530
- Soft cleaning 6pe-3pe
- Sample divided randomly into two sets: train (**80%**) & test (**20%**)

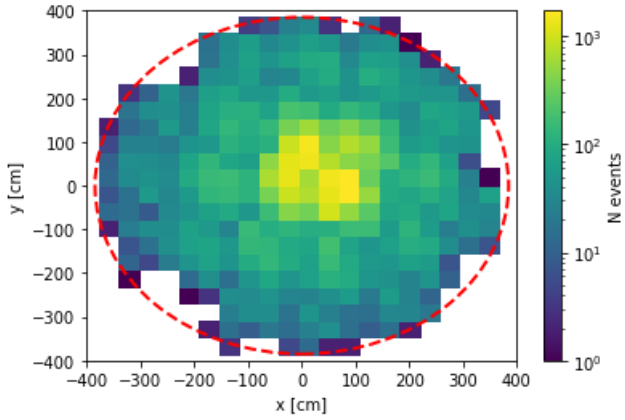
- Additional sample for "blind" check (14997 events)

Image CoG distribution

- Gamma-source is located in the camera center
- Proton images CoG dist ~ flat in camera coordinates





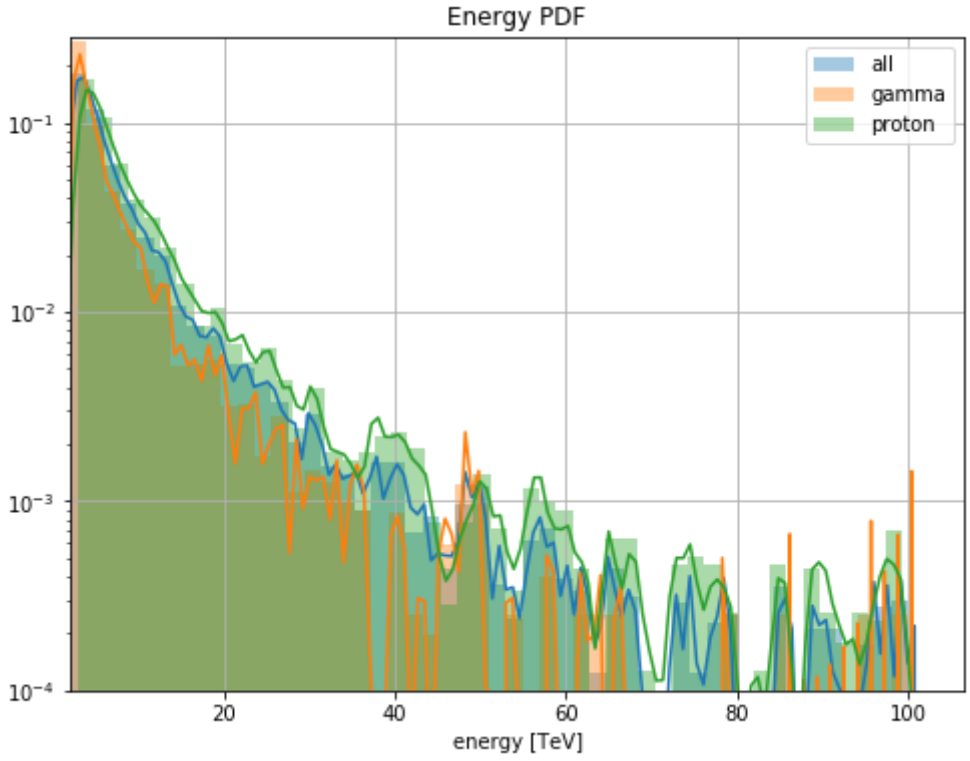


- Gamma images CoG mostly in the center of the camera

Energy distribution in sample

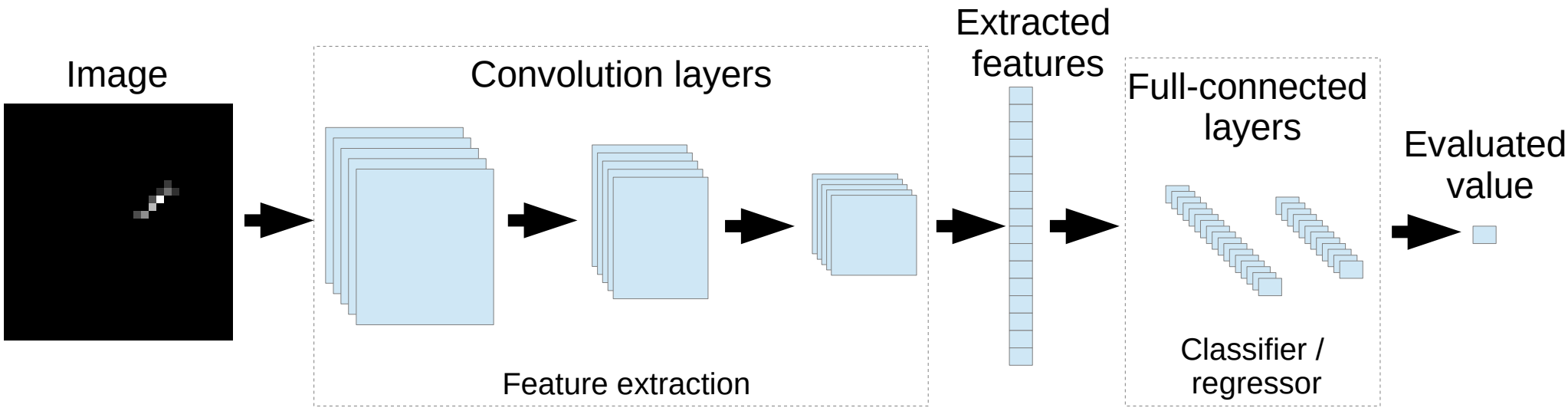
- Simulated events energy
  - Gamma 2 TeV - 100 TeV
  - Proton 3 TeV – 100 TeV

- 90 % with energy < 15 TeV



CNN architecture

- Convolution layers is used for feature extraction
- Full-connected layers is used for classification / regression



CNN for classification

- Input: Hexagonal image => oblique coordinates (31x32 pix)
- 3 convolution layers
  - 3x3 1x32 / 3x3 32x32 / 3x3 32x32
  - Averaged pooling ; Dropout 25%
  - ReLU activation
  - → 288 extracted features
- 2 full-connected layers (32x16)
  - ReLU activation ; dropout 50%
- Output layer (2 neurons)
  - Sigmoid activation

- ~30k trained parameters

Data sample extension

- As dataset is not big enough (40K events) the sample for train was extended
- The images was rotated 5 time on 60 degrees around the center of the camera
- New sample for train: ~ 200K events

CNN for regression

- Input: Hexagonal image => oblique coordinates (31x32 pix)
  - 3 convolution layers **(Extracted from CNN for classification)**
  - 3x3 1x32 / 3x3 32x32 / 3x3 32x32
  - Averaged pooling ; Dropout 25%
  - ReLU activation
  - → 288 extracted features
- 2 full-connected layers (32x16)
  - ReLU activation ; dropout 50%
- Output layer neuron
  - ReLU activation

- ~10k trained parameters

CNN for regression train

- To train the CNN all events was weighted using exponential fit of the primary energy pdf
- For the train all types of particles are used
- Mean square error is used as loss function

Results

- Currently the results of the achieved energy estimation are not good enough and are still in studying
- Events weighting method is not good enough and should be improved
- Other networks architectures and loss function should be tested

Future plans

- Continue studying of the event energy reconstruction
- More detailed studying of the particles classification (my be it’s worth to try to use Hillas parameters instead of Convolution layers)
- Tests for CNN with much bigger samples
- For the Hillas parameters extraction the **ctapipe** open python library can be used for the simulation and for real data analysis