A deep neural network method for analyzing the CMS High Granularity Calorimeter (HGCal) events

G. Grasseau[1], A. Kumar[2], A. Sartirana[1],
A. Lobanov[1] and F. Beaudette[1],

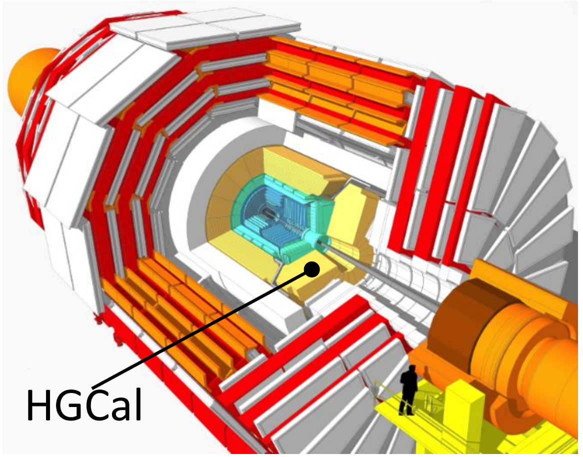
CHEP 2019 Conference, 4-8 November, Adelaide, Australia

[1] Leprince-Ringuet Laboratory (LLR), Ecole Polytechnique, Palaiseau, France
[2] Birla Institute of Technology and Science (BITS), Pilani, India

Motivations: HL@LHC

The High-Luminosity at LHC (HL-LHC) is a major evolution of the accelerator and the CMS detector (2024)

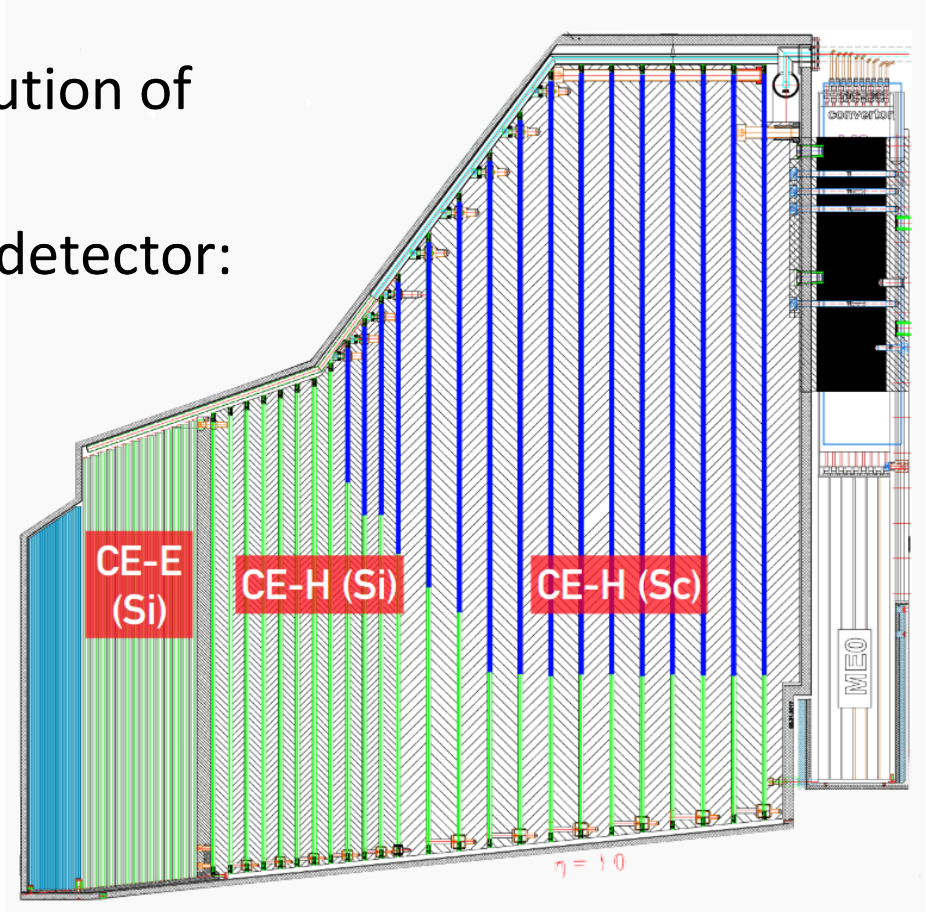
Our team is involved in the *endcaps of the* CMS sub-detector: *High Granularity Calorimeter* (HGCal)



HGCal Challenges

- Increasing pile-up (~200)
- The high granularity (> 6M channels)
- High occupancy
- High trigger rate at 40 MHz
- Time resolution : vertices spread in position and time (towards 4D analysis)

Involve drastic changes in the event reconstruction

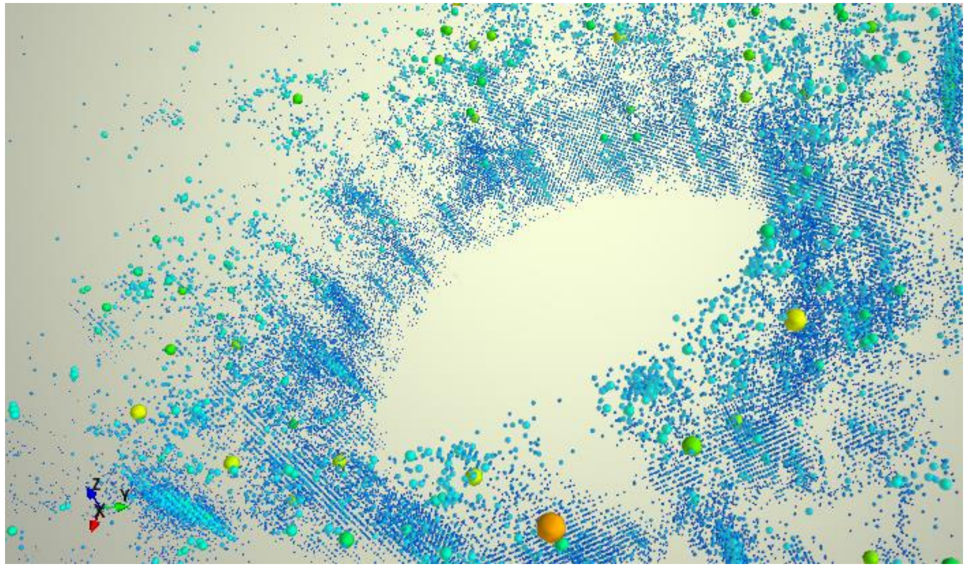


28 layers 22 layers

Motivations : HGCal event reconstruction

Current flexible approach:

- The Iterative CLustering (TICL)
- Combining clustering and pattern recognition iteratively



Event simulation in HGCAL sub-detector: energy deposits (log scale)

We propose to carry out the two steps simultaneously based on recent DL in image processing technics:

What we want:

- *Classify* in cluster categories : EM clusters (dense) or Pions showers (parse)
- *Localize* all the clusters and their footprint

In DL field our problem falls in the "Object detection" realm

The model : Mask RCNN

Benefit from the applied research, motivated by industrial challenges:

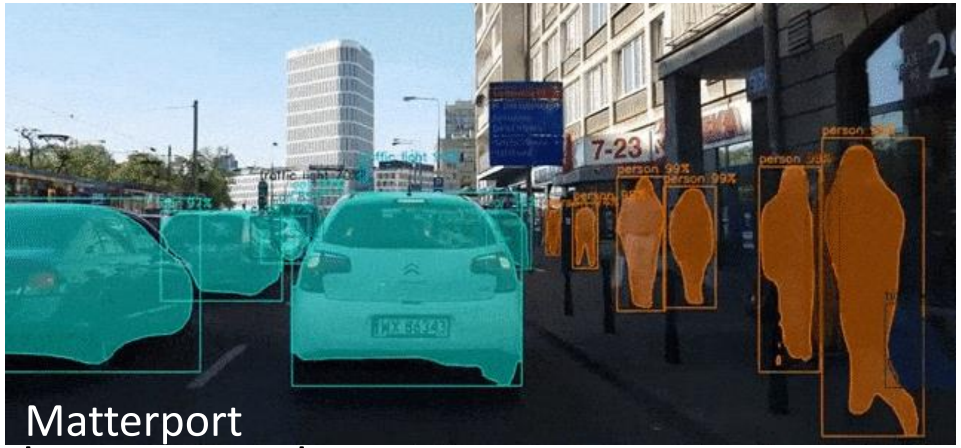
- automotive, face recognition, satellite imagery, medical, …

Object detection evolution:

- CNN with Sliding Windows
- R-CNN (2013),
- Fast RCNN (2015),
- Faster RCNN (2015) ,
- Mask-RCNN (2017-18)

Model Competition (speed & accuracy)

- Yolo - You Only Look Once (*SxS* grid)
- SSD - Single Shot Detection

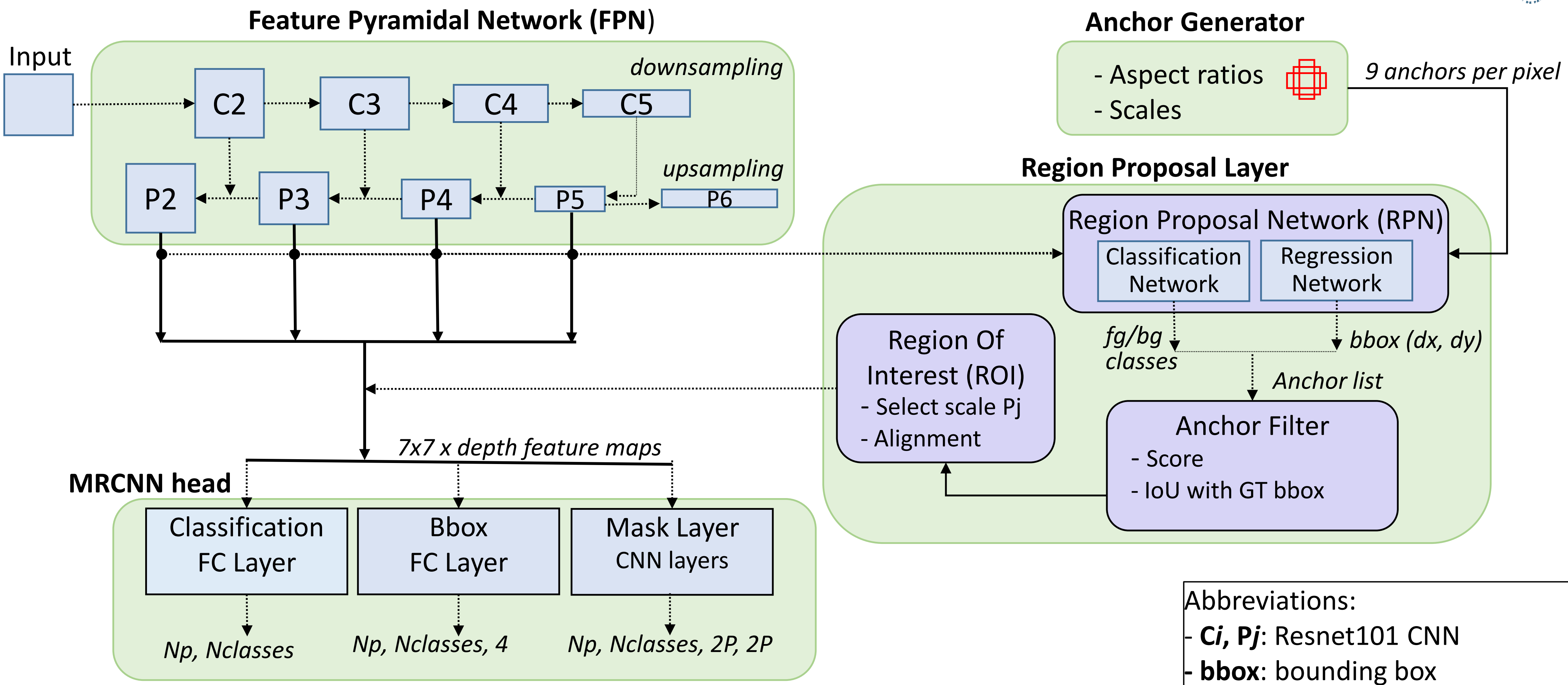


Mask R-CNN implementation

- Original Facebook Research https://github.com/facebookresearch/Detectron
- Matterport (TF, Keras) https://github.com/matterport/Mask_RCNN
- Medical Detection Toolkit (3D, PyTorch) https://github.com/pfjaeger/medicaldetectiontoolkit/tree/master/models
- Tensorflow implementation https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/instance_segmentation.md

We decided to start with, as a test bench, the 2D Matterport implementation before tackling the 3D problem

Mask R-CNN Model



Abbreviations:
- **Ci, Pj**: Resnet101 CNN
- **bbox**: bounding box
- **GT**: ground truth
- ***fg/bg***: *foreground / background*

Mask R-CNN Model

Loss computation

$$L = L^{rpn} + L^{mrcnn}$$

with

$$L^{rpn} = L^{rpn}_{class} + L^{rpn}_{bbox}$$

$$L^{mrcnn} = L^{mrcnn}_{class} + L^{mrcnn}_{bbox} + L^{mrcnn}_{mask}$$

and

- *Cross-entropy* for class

$$L^{class} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} P_{truth}(C_{y_i} = c).\log(P_{model}(C_{y_i} = c))$$

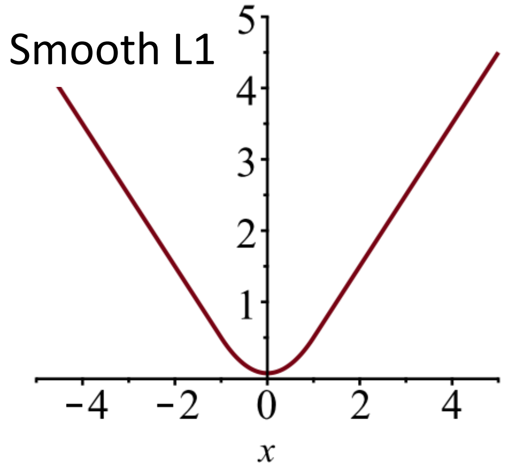
- For bounding boxes : *smooth L1 (see fig.)*

$$L^{bbox} = -\frac{1}{N}\sum_{i=1}^{N} L_1^{smooth}(y_i)$$

- Mask Loss
  Binary cross-entropy (similar to L_{class})

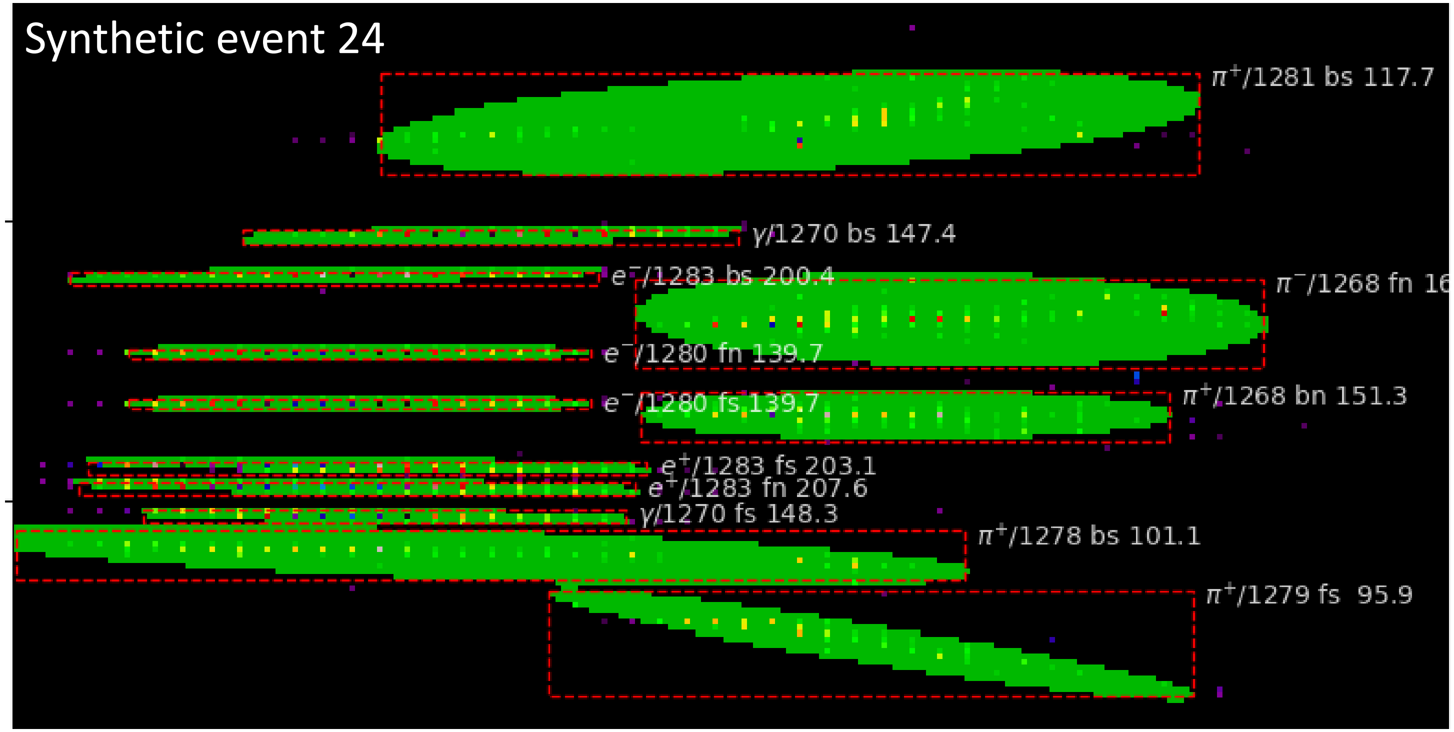
Main model characteristics

- Batch size: 8
- Optimizer:
  - Stochastic gradient descent (SGD),
  - learning rate 0.001,
  - momentum 0.9
- Penalization: L2 regularization



Building the Data-Set

Need samples with their “object” *location* (bounding box or bbox) and their *classification*.
Difficult to extract all the details of each object from a simulation with pile-up.
Choose to simulate single particles ($e^{+/-}$, γ, $\pi^{+/-}$) that are overlaid on-the-fly:
small approximation and a lot of flexibility.



Build a primary data-set with a unique object

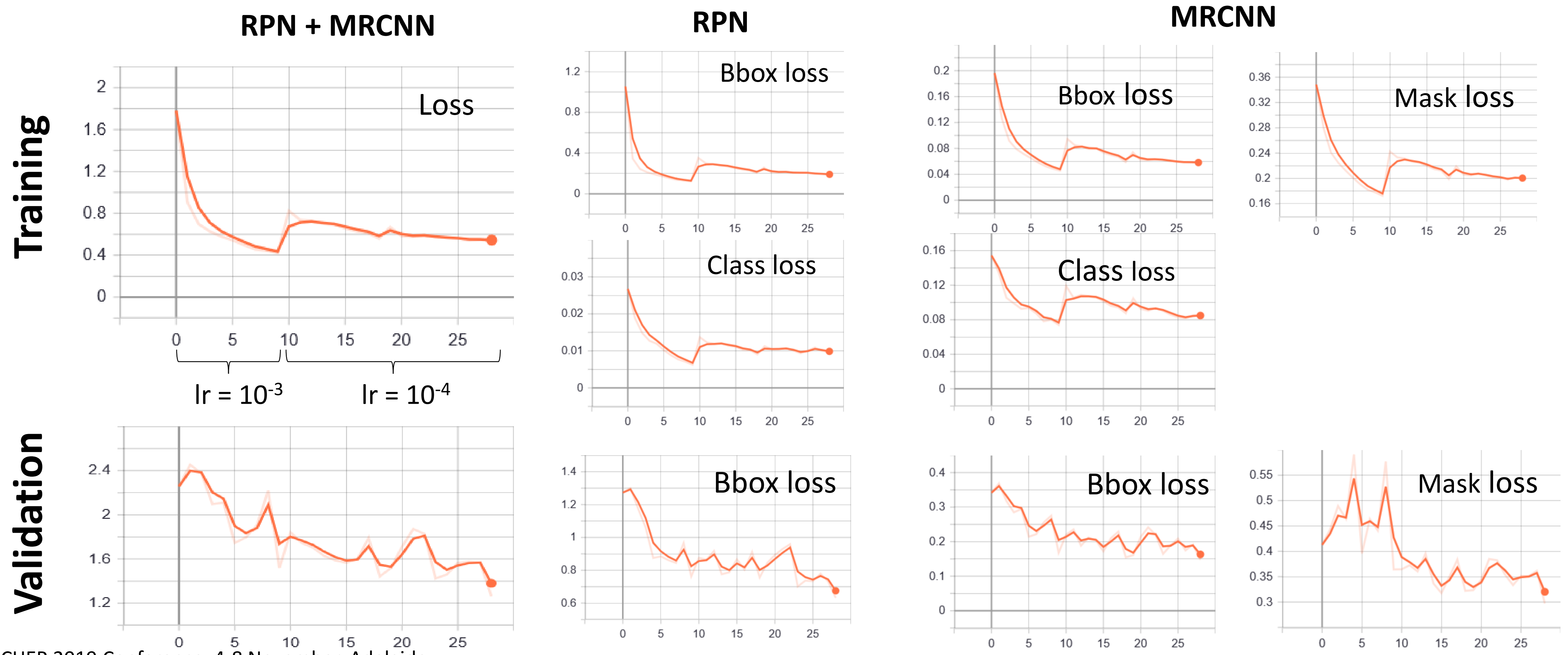
- Get events with **unique** particle with E > 20 Gev (in fwd or in bwd detector) to get the bbox
- Build a **2D histogram** (3D -> 2D image)
- The mask will be an ellipse (PCA of the cluster/shower)

Compose a training/validation data-set with “objects” in the primary data-set

- Set a number of “objects” in the image (a range)
- Select randomly them among the primary data-set
- Random operation (data augmentation): object symmetry, random shift dy = +/- 1 pixel

Results: Loss

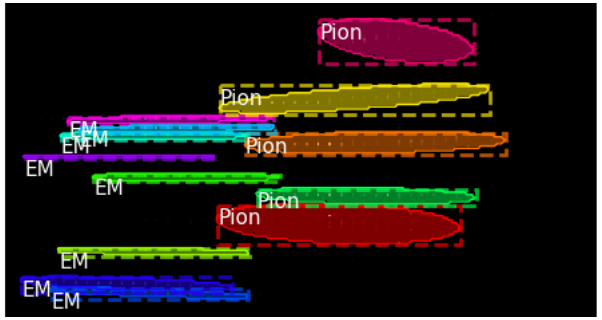
12-20 objects, Training data-set 5000 ev., Evaluation data-set 50 ev., epoch ~30

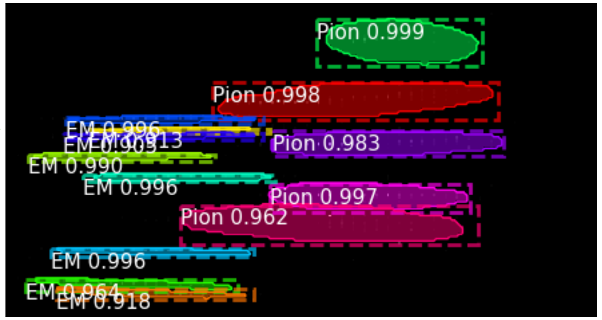


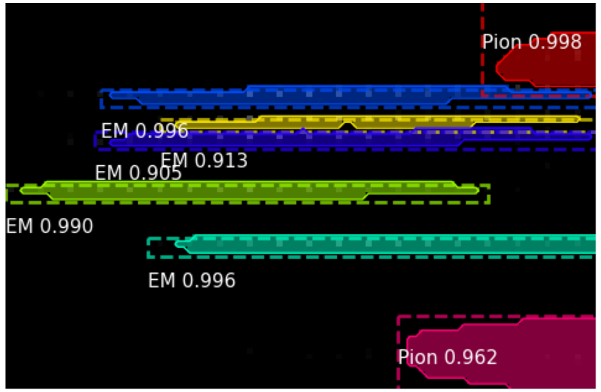
Results: nice predictions

Ground truth ev. | **Predicted ev.** | **Zoom of predicted ev.**

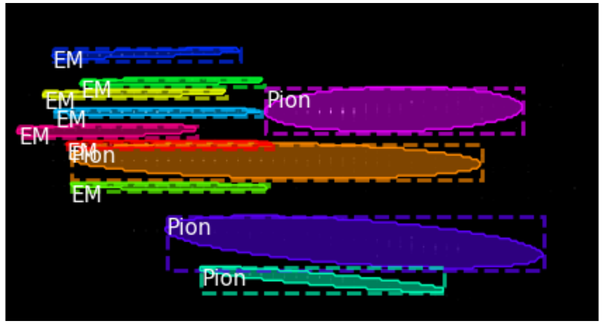
Ev 14

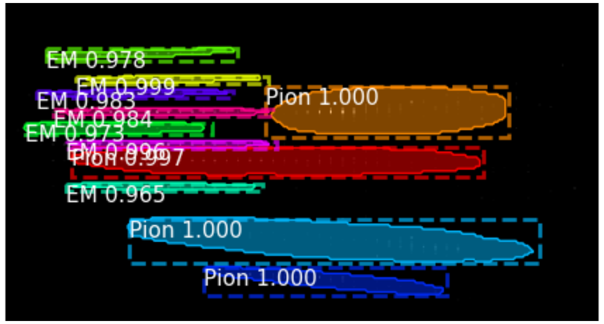


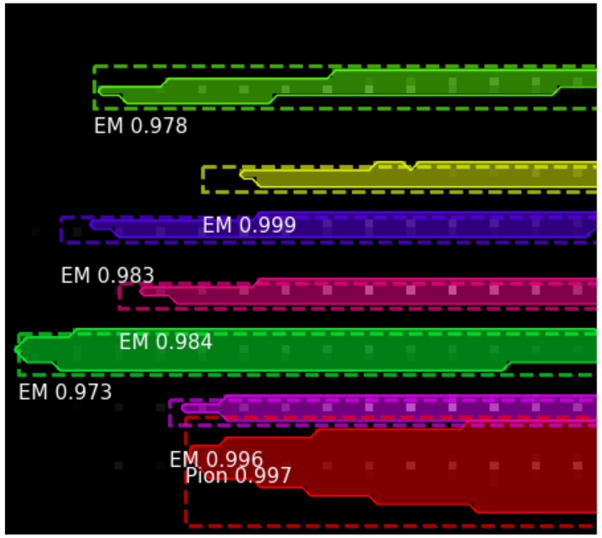




Ev 17







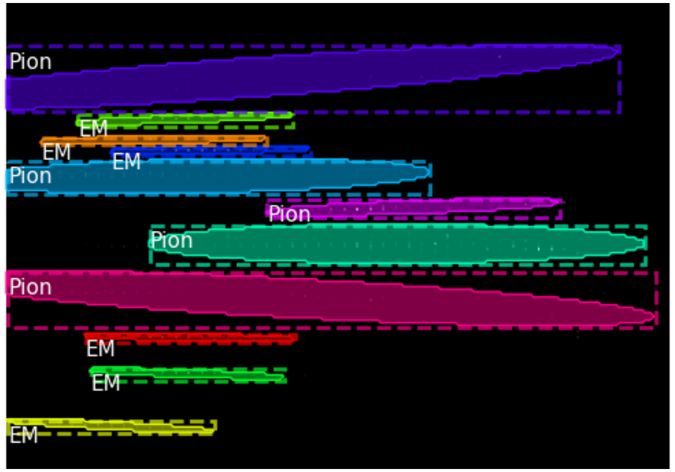
Good predictions:

- Classification, localization (bbox), mask
- Dense region of objects (green arrows)
- mAP (mean Average Precision) = 0.73

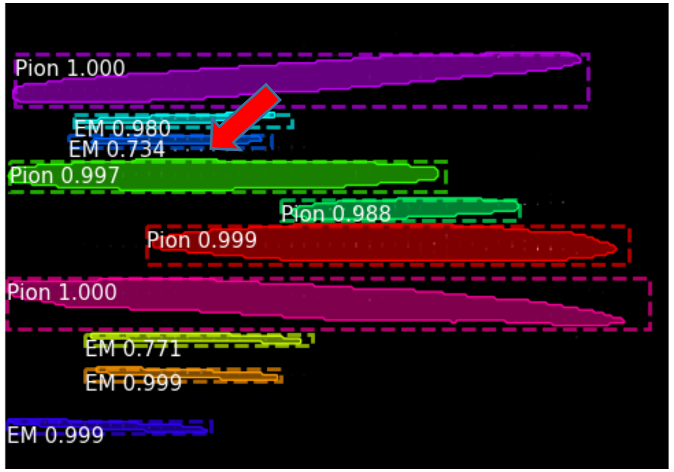
But …

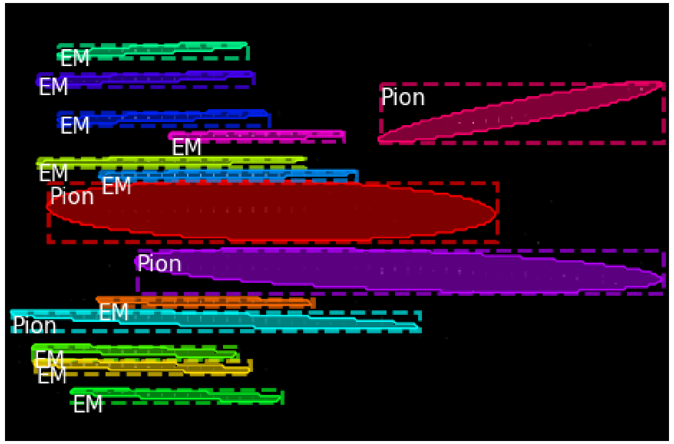
Results: … to improve

Ground truth ev. **Predicted ev.**

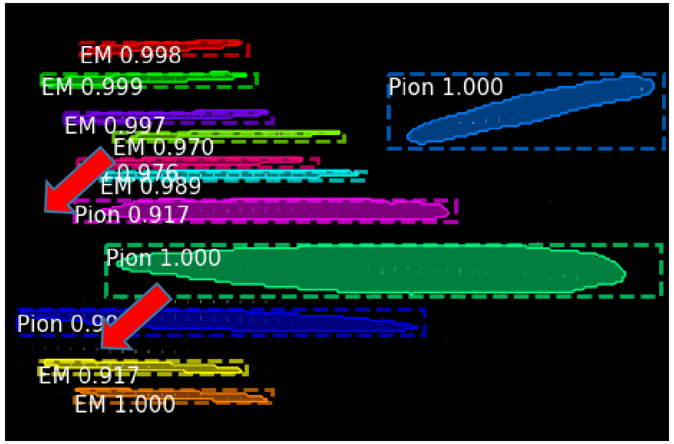


Ev 13





Ev 15



Good

- Pion showers start in EM region (green arrows)

To improve

- Missing object (red arrows)
- Small mask for pion shower (red arrows)

mAP = 0.73, ~ 15 % objects missing

Conclusion / Perspectives

HGCal 2D test bench

- Challenging conditions: small data-set, rough histograms, the layers are far from each others, int8 as input, …
- However, gives pretty good results
- Mask R-CNN captures the scattered hits coming from Pion showers

Next Steps

HGCal 2D

- Getting better conditions to train
- Modify the model in MRCNN

HGCal 3D

- Apply the lessons of HGCal 2D
- Medical Detection Toolkit (3D, PyTorch)

Acknowledgments

- Funding project P2IO (GPU platforms)
  Accelerated Computing for Physics
- Google Summer of Code 2019
  HAhRD project : DL & HGCAL
- IN2P3 project: DECALOG/Reprises
- S. de Guzman – Ecole polytechnique internship - testing 2D/3D implementations
- CHEP 2019 organizers