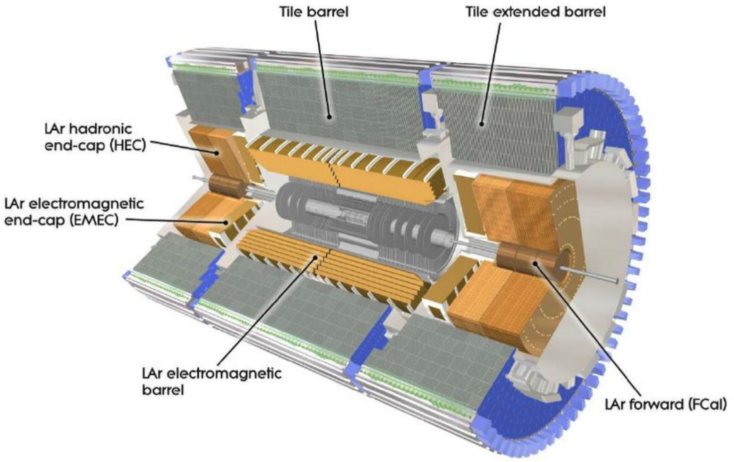
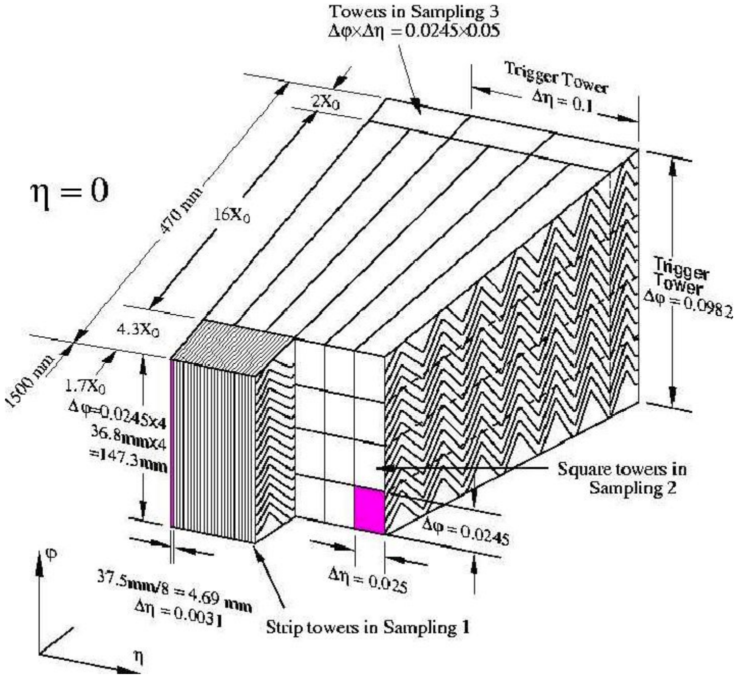
Laboratoire d'Annecy de Physique des Particules

Machine Learning @ ATLAS-LAPP

BELFKIR Mohamed
22-01-2021
LAPP-LISTIC

Introduction





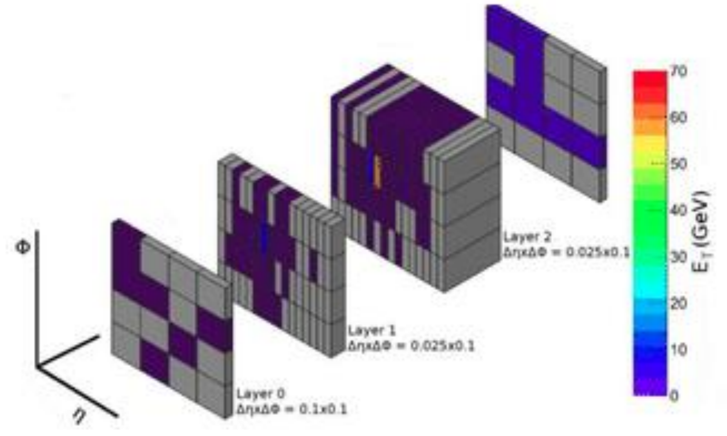
- ATLAS use calorimetry to measure energy of photons/electrons and hadronic jets.
- ATLAS Calorimeter system :
  - Electromagnetic : Photons, Electrons
  - Hadronic : Jets

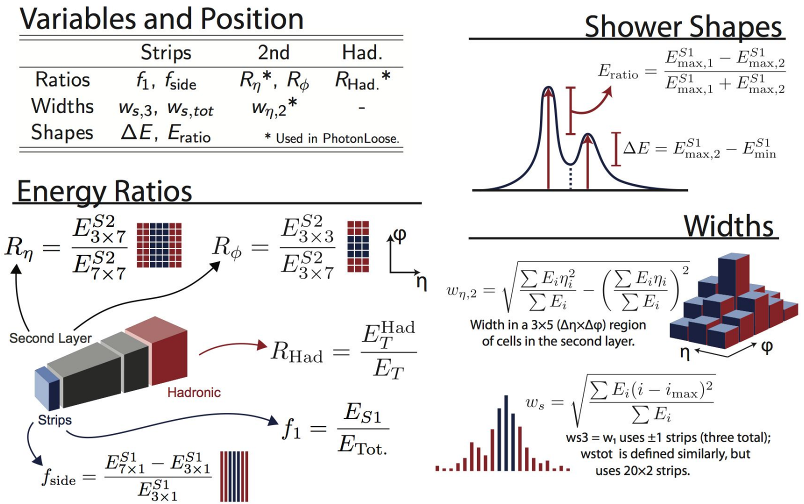
- ATLAS EM Calo is segmented in 3 samplings with different granularity
  - 3D reconstruction of the electromagnetic shower

Photons at ATLAS

- Jets and other objects have similar properties to photons.
- A powerful photon identification algorithm is needed for many physics analysis :
  - H→ yy, HH → yybb ...

- Currently, ATLAS use a cut-based to identify photons using shower shape variables (High Level):





- EM clusters are used to reconstruct photons
- Photon related variables are computed using a cells (energy, position ...) in a defined windows (e.g 7x11).

Motivation

- ❖ Cut-based doesn't explore the full EM calorimeter power.
- ❖ Explore deeper the EM calorimeter using Low Level features.
- ❖ Implement NN @ cells level : Convolutional Neural Network (CNN).
- ❖ Use Image from EM calorimeter layers to train a CNN Classifier.
- ❖ Train network on the images

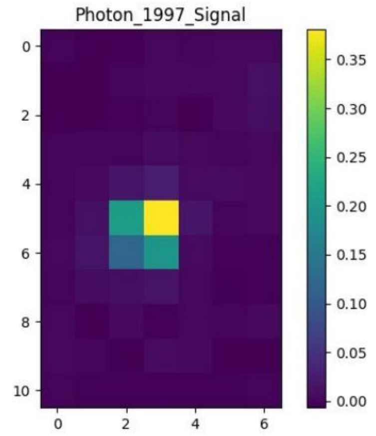


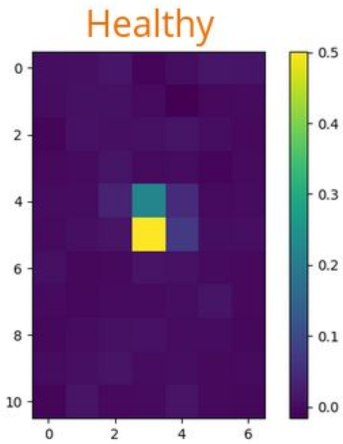
Image Preprocessing

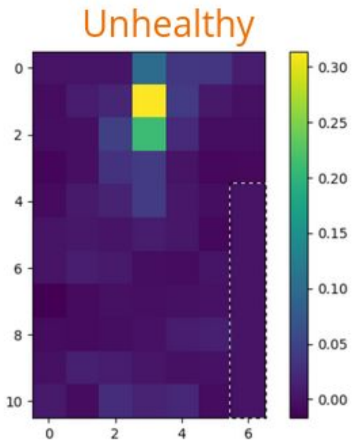
Eta	0 - 1.4	1.4 - 1.8	1.8 - 2.0	2.0 - 2.5
Sampling 1	112	112	84	56
Sampling 2	77	77	77	77
Sampling 3	44	44	44	44

Number of pixels

- Images are reconstructed from 7x11 cluster from each layer : 1, 2 and 3.
- Pixels are the cell energy normalized to the total energy the layer.
- Hardware :
  - EM Calo segmentation changes with its granularity.
    - Complete image size with zeros.
- Software :
  - Shift in EM alignment (> 50% events).
    - Complete image size with zeros.

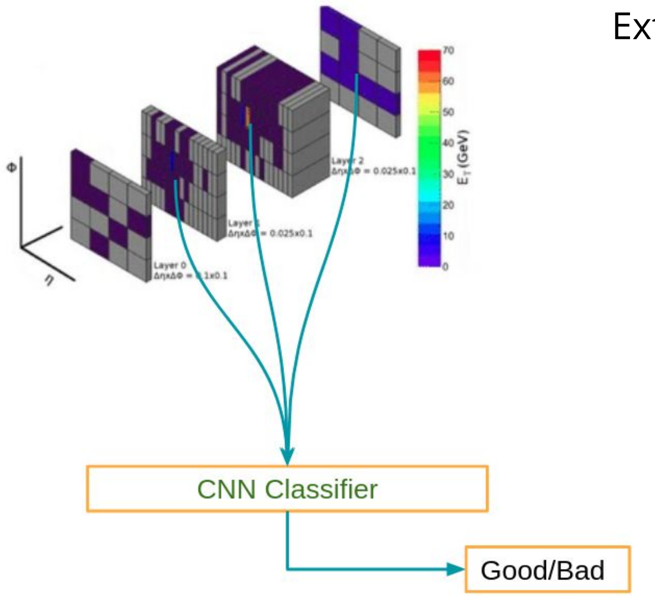
➔ Temporary solution : MTCNN (back up)



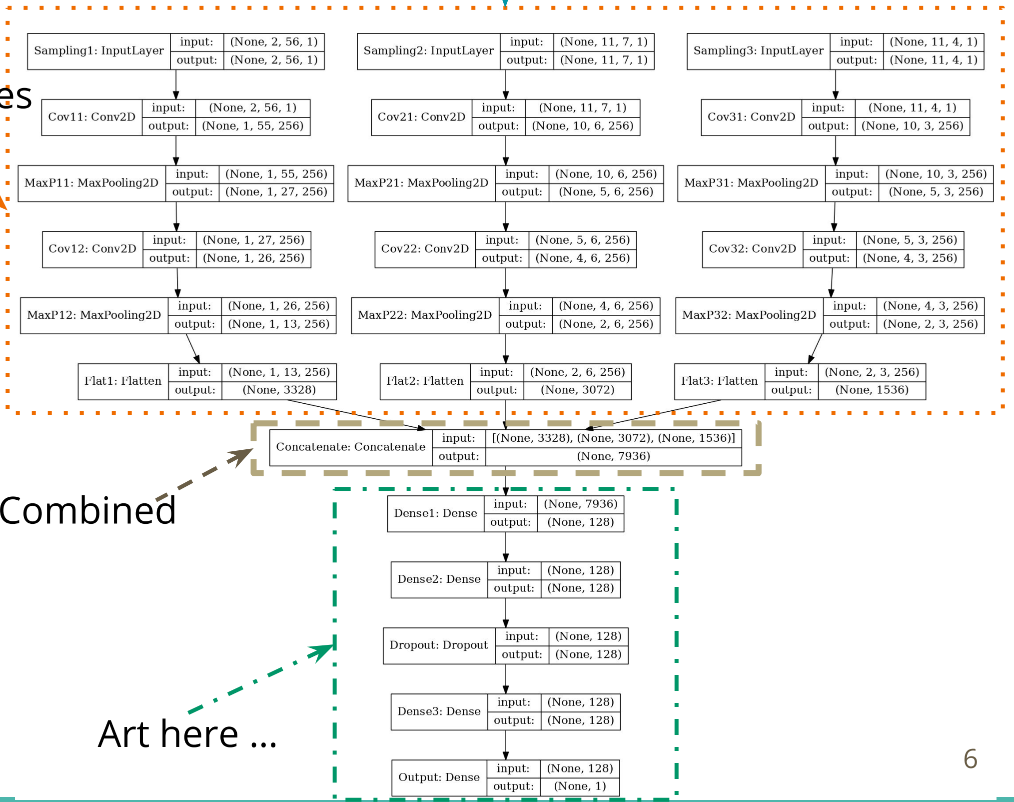


Architecture

Keras functional API (Tensorflow as backend)



Extract features



Combined

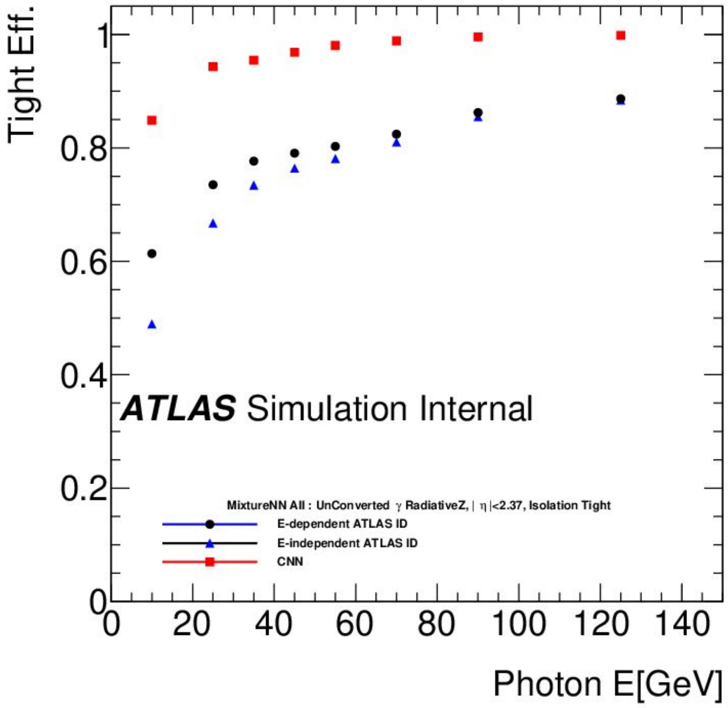
Art here ...

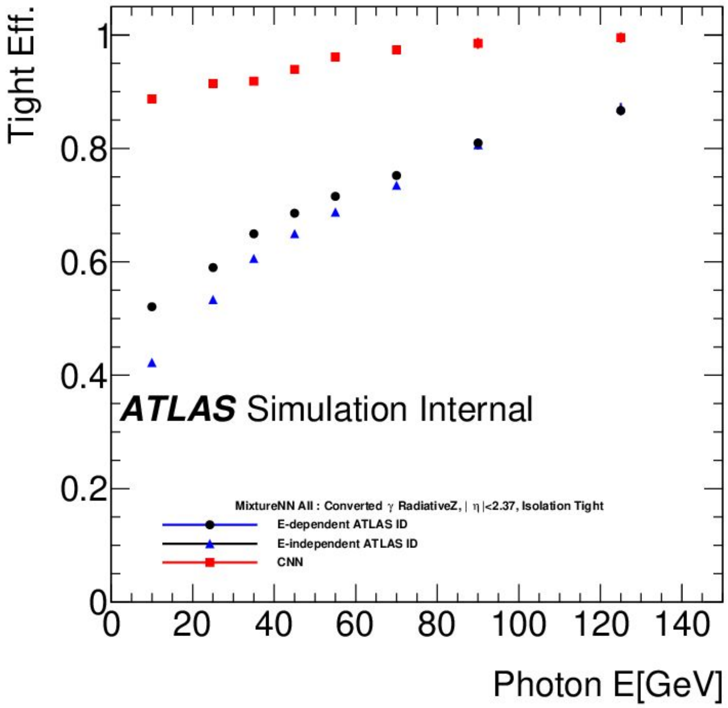
Training

- Weights initialized using He initialization
- Activation function :
  - selu for all layers.
  - sigmoid for output layer.
- Drop rate : 8%
- Unbalanced data : class weight
- Two CNN networks :
  - HealthyNN : healthy clusters.
  - MixtureNN (baseline) : mixture of healthy and unhealthy.

- Network ~1.4M parameters.
- Binary cross-entropy as loss minimized using Adam Opt.
- Training with ~5.5M Events of Inclusive photons (~11M Images):
  - 20 epochs.
  - 10^{-4} as learning rate.
  - batch size of 32 images.
- 2 CC-In2p3 Tesla K80 GPU 12 GB used for training (~15 min each epoch)

Improvement : MixtureNN (Out-of-Sample dataset)

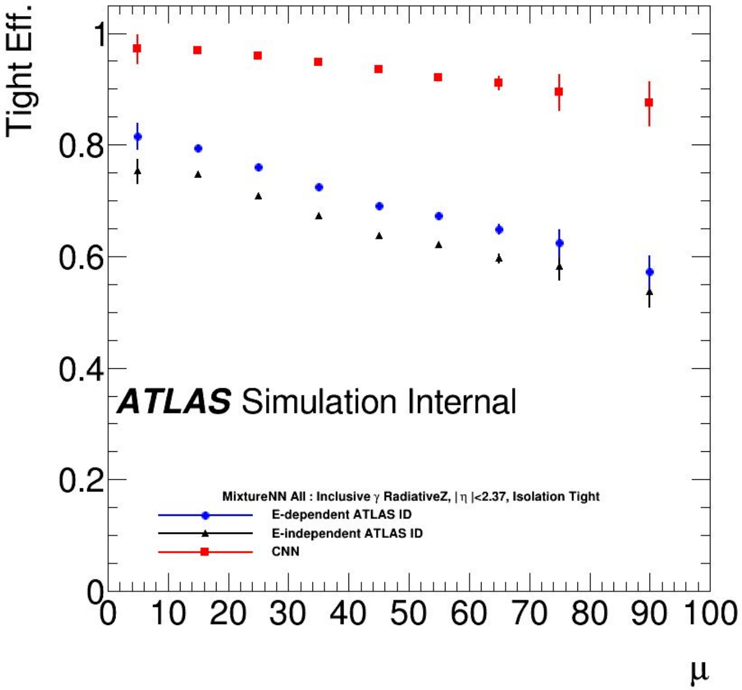




- CNN over performs the cut based algorithms.

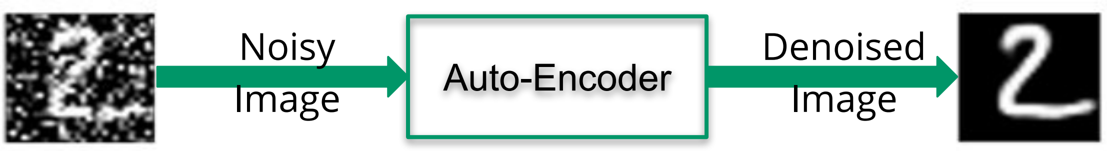
Pile Up dependency

- CNN performance affected by Number of collisions in each bunch crossing “Pile Up μ”.
- Pile Up represents an extra noise to CNN classifier.
- At HL-LHC, an average of 200 collision in each bunch crossing is expected → CNN will not support this huge pile up.



Improve pileup dependency

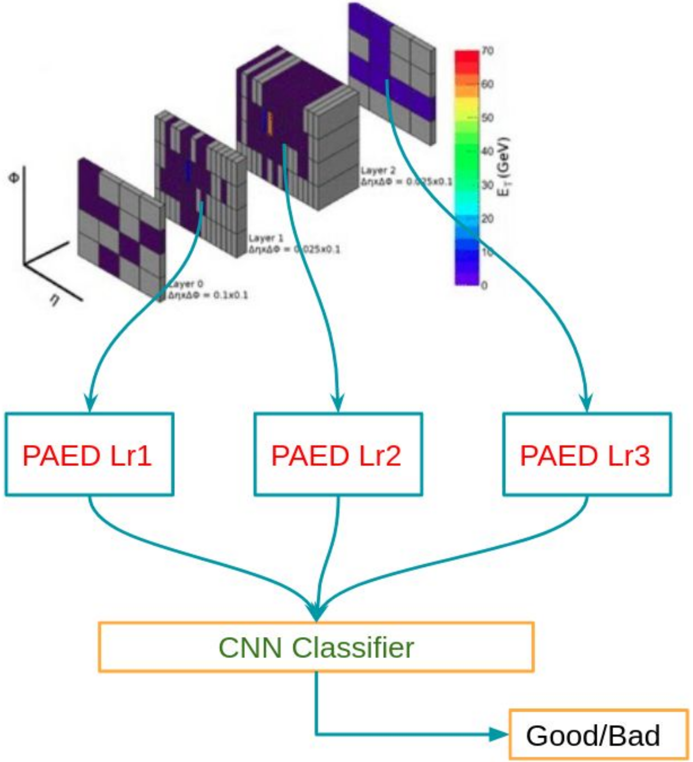
- The mean idea is to remove the pile up dependency using Auto-Encoder before CNN
- Similar to Image denoising (medicine ...); PileUp is noise



- Build a PileUp Auto-Encoder Denoising (PAED).
- Calorimeter images will pass through the PAED to extract pile up from them then pass to CNN for photon Classification.

PileUp Auto-Encoder Denoising

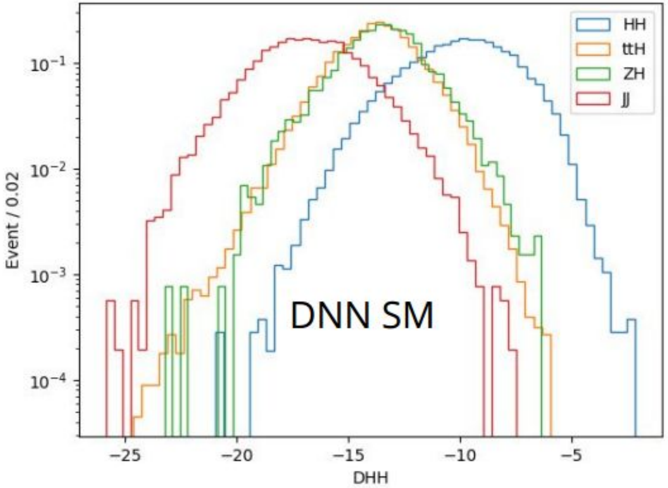
- Two MC samples (w/ and w/o PU) with same generator, same process are need for training.
- Each EM Layer will be connected to a PAED.
- PAED will be trained separately (or in group one big network) on Monte Carlo sample.
- PAED will be trained to reconstruct the non pile up latent space representation.
- Latent representation could be used for extra purposes:
    - Photon Isolation
    - Energy reconstruction



Deep Neural Network

- Enhance HH→ yybb sensitivity using multi-class DNN classification:
  - 4 Classes : Signal (HH) vs Backgrounds (ttH, ZH and jj)
  - Using Event topology (Kinematic variables of the final states)
- 4 outputs combined in one discriminante variable:

$$d_{HH}^{SM} = log(\frac{\sigma_{SM} \cdot p_{HH}}{\sigma_{ZH} \cdot p_{ZH} + \sigma_{ttH} \cdot p_{ttH} + \sigma_{jj} \cdot p_{jj}})$$



	SM
BDT	0.49
DNN	0.54

Sensitivity s/√b

Back Up

Future improvement

- Multi-Task Cascade Convolution Networks (MT-CNN)
- Used in face detection.
- Improvements :
  - Topological clusters (No fixed shower size).
  - No fixed windows size
  - Alignement (no need to have complet cluster).
- Technically complicated to be implement.