

GammaLearn: a Deep Learning framework for IACT data

**Mikaël Jacquemont^{ab}, Thomas Vuillaume^a, Alexandre Benoit^b, Gilles Maurin^a,
Patrick Lambert^b, Giovanni Lamanna^a, Ari Brill^{c*}**

^aLAPP, Univ. Grenoble Alpes, Univ. Savoie Mont Blanc, CNRS
9 Chemin de Bellevue, Annecy-le-Vieux, 74940 Annecy, France

^bLISTIC, Université Savoie Mont-Blanc Polytech Annecy-Chambéry,
5 chemin de Bellevue, Annecy-le-Vieux, 74940 Annecy, France

^cColumbia University, Department of Physics,
New York, NY, USA

E-mail: mikael.jacquemont@lapp.in2p3.fr,
thomas.vuillaume@lapp.in2p3.fr

Imaging atmospheric Cherenkov telescopes (IACT) data require an important analysis in order to reconstruct events and obtain a photon list. The state-of-the-art reconstruction is made of several steps including image analysis, features extraction and machine learning. Since the 2012 ImageNet breakthrough, deep learning advances have shown dramatic improvements in data analysis across a variety of fields. Convolutional neural networks look particularly suited to the task of analysing IACT camera images for event reconstruction as they provide a way to reconstruct photon list directly from raw images, skipping the pre-processing steps. Moreover, despite demanding important computing resources to be trained and optimised, neural networks show very good performances during execution, making them viable for real-time analysis for the future generation of IACT. Here we present GammaLearn, a python framework providing the tools and environment to easily train neural networks on IACT data. Relying on PyTorch, it allows the use of indexed convolution on images with non-cartesian pixel lattices predominant in IACT for the low-level operations and offers a simple configuration file-based workflow, producing the trained model, training estimators as well as higher level results. The proposed framework is modular and straightforward to customize by end users. It has been tested and validated on the analysis of the Cherenkov Telescope Array simulated data.

*36th International Cosmic Ray Conference -ICRC2019-
July 24th - August 1st, 2019
Madison, WI, U.S.A.*

*Speaker.

1. Deep learning for IACTs image analysis

Imaging Atmospheric Cherenkov Telescopes (IACTs) observe the Cherenkov light emitted by cosmic rays entering the atmosphere thanks to an optical system collecting and focusing the light onto an ultra-rapid camera. The produced images of the atmospheric shower must be analysed and combined to take advantage of the stereoscopic view and to reconstruct the physical parameters (mainly particle type, arrival direction and particle energy) of the primary particle. This reconstruction is based on Monte-Carlo simulations generating images based on the simulation of particles interaction in the atmosphere and simulation of the telescope response function. Several reconstruction methods have been developed through the years either based on features extraction [1, 2, 3, 4] or on templates method directly comparing images with simulated ones [5, 6, 7].

Since the Imagenet breakthrough in 2012 [8, 9], Deep Learning (DL) has established state of the art performances in various range of fields, from natural language processing [10] to computer vision [11, 12]. Taking advantage of these recent advances in image analysis, [13, 14, 15, 16] have applied Convolutional Neural Networks (CNN) to γ event reconstruction from IACT images, showing promising results.

The GammaLearn project, born from a collaboration of the Laboratory of Annecy of Particle Physics (LAPP), the Laboratoire d'Informatique, Systèmes, Traitement de l'Information et de la Connaissance (LISTIC) and an industrial partner specialized in Deep Learning solutions for manufacturers, Orobix, aims to explore DL techniques for IACT and in particular the Cherenkov Telescope Array (CTA) data analysis.

2. GammaLearn framework

Deep Learning is a highly empirical process requiring many training and testing cycles of different architectures with different hyperparameters. Moreover, the goal of the GammaLearn project is to find the best possible neural networks for gamma / cosmic rays separation and gamma parameters reconstruction, stressing the need of a tool to ensure reproducibility, traceability and easy launch for all the experiments to be run.

2.1 Description

The GammaLearn framework[17] has been designed to tackle these issues. It is a modular and plug and play Python first tool that relies on PyTorch[18] for DL capabilities, mainly tensor manipulation, automatic differentiation (which is essential for gradient descent optimization) and GPU computations. In GammaLearn the training process itself (i.e. the execution of the training, validating and testing loops) is handled by Ignite (v1.0a)[19] in order to benefit from its event management system. In the following, an *experiment* designates the whole process of training and testing a CNN with particular hyperparameters.

As described on Fig. 1, GammaLearn is composed of an engine, the `experiment_runner`, and 7 collections of tool functions and classes. The `experiment_runner` role is to load and check the experiment settings, via the `Experiment` class, load the training data, load the CNN, train, validate and test the loaded CNN and produce monitoring data and performance metrics, as defined in the experiment settings file. The tool collections provide all the functions and classes to:

- load datasets,
- pre-process data (filter, augment, transform),
- train, validate and test networks,
- monitor the training process,
- visualize training results.

Each collection of tools, serving a specific purpose, follows the same prototype for function and class definition. For example, the Handlers collection contains functions to handle the events fired by Ignite's engine, like training or validation loss logging when an epoch is completed. To add a new handler or to build one's own collection of handler compatible with GammaLearn, one needs to observe the following prototype:

```

1 def create_new_handler(experiment):
2     """
3     Function to create a handler
4     Parameters
5     -----
6     experiment (Experiment): the experiment
7     Returns
8     -----
9     A function registrable by ignite Trainer
10    """
11    # do something
12    def handle_an_event(trainer):
13        # do something
14        return handle_an_event

```

The collections allow the user to run various types of experiment on IACT data: classification, regression, single task learning, multi-task learning, mono or stereo analysis.

Beside the different collections, the plug and play quality of GammaLearn also results from its Python first nature. Each component of the framework is written in Python, even the experiment settings file, allowing the user to add his own components by calling them in the experiment settings file.

2.2 Ecosystem

To be an efficient DL framework for IACT data, GammaLearn is integrated in a wide ecosystem of tools:

- IndexedConv[20] : the IndexedConv package provides convolution and pooling operations for images with any kind of grid. Indeed, in the case of hexagonal grid images, standard 2D convolution functions implemented DL frameworks are not suitable because they assume the grid of the image they process to be Cartesian. IndexedConv relies on the list of neighbours of each pixel of interest to compute the convolution, and thus can be applied to any image grid and shape. Moreover, it comes with all the needed functions for hexagonal grid images in particular and CTA images in general (i.e. building the necessary index matrix, extracting the list of neighbours from it). IndexedConv is fully supported by GammaLearn .

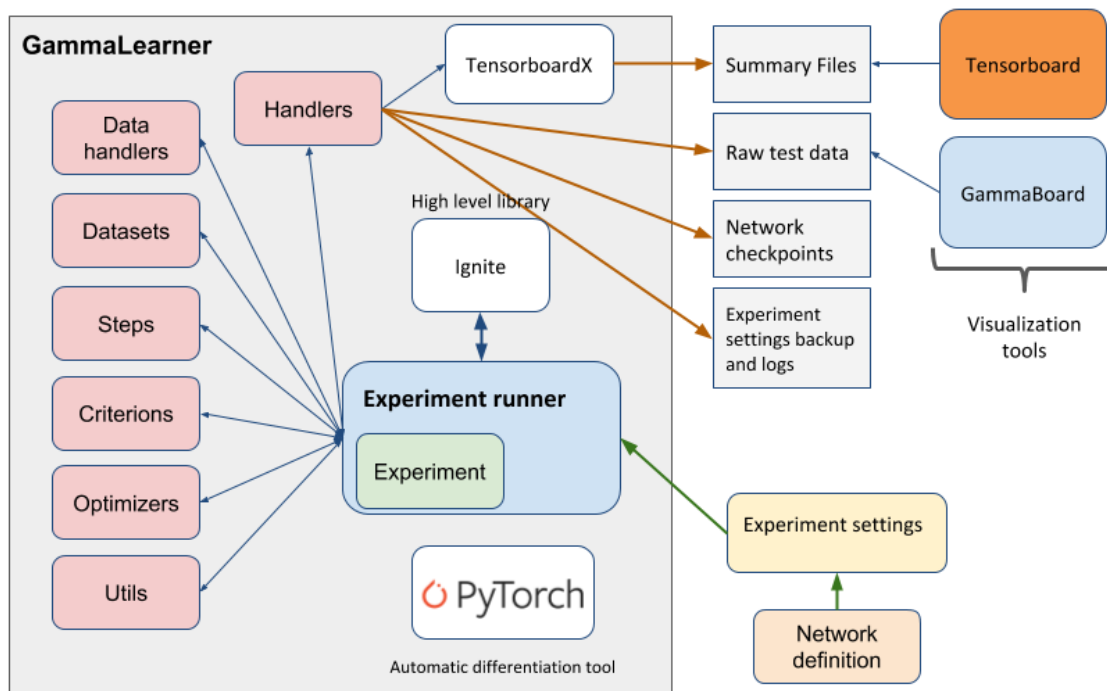


Figure 1: Description of the GammaLearn framework. GammaLearn comes with a set of function and class collections (in pink) to process IACT data. It relies on PyTorch for the Deep Learning fundamentals and on Ignite for the training routines. GammaLearn generates monitoring and performances data during the training, directly workable by Tensorboard and GammaBoard.

- **GammaBoard:** GammaBoard¹ is a dashboard build to display metrics assessing the reconstructions performances of Imaging Atmospheric Cherenkov Telescopes (IACTs). It is built upon the widespread Jupyter Notebook[21] technology. It benefits from Matplotlib[22] interactive plots and ctaplot² for IACT related metrics (resolutions curves and effective area). Thanks to its click-and-play interface, as shown on Fig. 2, it allows a quick and simple comparison of the experiments.
- **Tensorboard**
Tensorboard³ is a suite of web applications coming with Tensorflow[23]. It offers useful tools to visualize monitoring data (e.g. network weights distribution over the training or GPU memory used), training performances (e.g. loss and accuracy evolution) and to inspect neural networks. Thanks to tensorboardX⁴, a module to export data in a format readable by Tensorboard, and the Handlers collection, GammaLearn benefits from the power of Tensorboard.
- **DL1 Data Handler**

¹<https://github.com/gammaboard>

²<https://github.com/vuillaud/ctaplot>

³<https://github.com/tensorflow/tensorboard>

⁴<https://github.com/lanpa/tensorboardX>

DL1 data handler (DL1DH)⁵ is a Python library to handle calibrated images from CTA. The package has been developed to handle CTA raw data and write, read and apply image processing to calibrated images. DL1DH has been integrated to GammaLearn to ease the process of loading CTA data and provide them seamlessly to the framework. As IACTs image are often non-standard images (e.g. presenting hexagonal pixels in hexagonal lattices), image pre-processing (e.g. oversampling, rebinning or interpolation) can be applied thanks to DL1DH. A study of the effect of these pre-processing has been realised by [24].

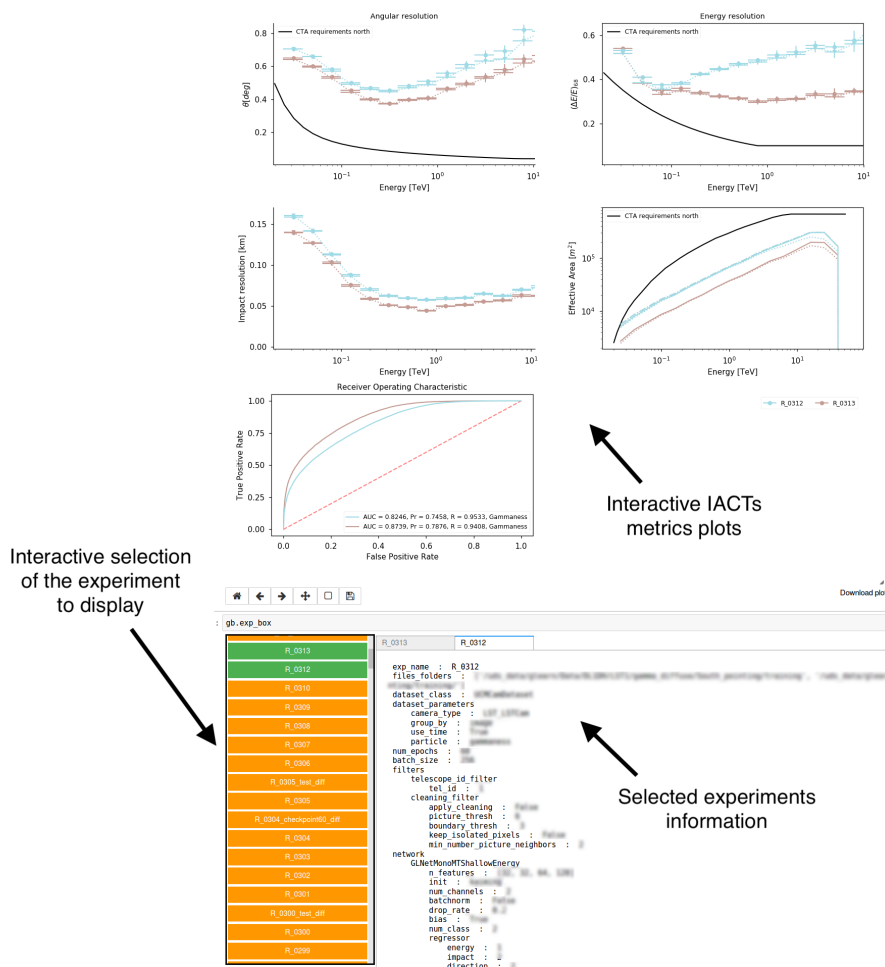


Figure 2: Gammaboard interface. It allows the user to interactively display experiments results with metrics specific to IACTs, providing a quick and meaningful comparison.

To fully benefit from this ecosystem, it is highly recommend to use Conda⁶ environment manager.

2.3 Work-flow

As shown on Fig. 1, the work-flow to train a network on IACT data with GammaLearn is

⁵<https://github.com/cta-observatory/dl1-data-handler>

⁶<https://conda.io/en/latest/>

pretty straightforward. For a typical experiment, i.e. taking advantage of the already implemented class and functions in the GammaLearn 's collections, one needs to provide the framework with an experiment settings file (in Python) and the desired network definition (with PyTorch). An example of experiment settings file can be found in the GammaLearn repository, in the folder *examples*, comprising all the mandatory and the optional setting fields handled by the framework. Then, to start the experiment, one executes the following commands in a bash terminal:

```
1 source activate <GammaLearn conda environment>
2 cd <path to GammaLearn>/gammalearner
3 python experiment_runner.py <experiment_settings_path> [--logdir] [--debug]
```

The framework trains the network and produces the monitoring and performance data as defined in the experiment settings file.

3. Conclusion

Deep learning methods based on convolutional neural networks are becoming the most powerful approach for most image analysis tasks, pushing the development of standard tools and libraries to apply them effortlessly and in a reproducible way. The study and use of CNN is likely to spread also in the gamma-ray astronomy to treat telescope and astronomical images. GammaLearn and its environment provide a complete framework and environment for their study and application to IACTs images. It solves some of the main challenges scientists are facing when applying CNN to the reconstruction of IACTs data: the application of convolutions on hexagonal images, the combination of several telescopes images, loading and treating IACTs data, displaying high-level metrics and comparing them for different experiments, reproducibility of the experiments via configuration files.

4. Acknowledgement

We gratefully acknowledge financial support from the agencies and organizations listed here: www.cta-observatory.org/consortium_acknowledgment. This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 653477. This work has been done thanks to the facilities offered by the Univ. Savoie Mont Blanc - CNRS/IN2P3 MUST computing center. We gratefully acknowledge the support of the NVIDIA Corporation with the donation of one NVIDIA P6000 GPU for this research.

References

- [1] J. Albert, E. Aliu, H. Anderhub, P. Antoranz, A. Armada, M. Asensio, C. Baixeras, J. A. Barrio, H. Bartko, and D. Bastieri, “Implementation of the Random Forest method for the Imaging Atmospheric Cherenkov Telescope MAGIC,” *Nuclear Instruments and Methods in Physics Research A*, vol. 588, pp. 424–432, Apr 2008.
- [2] S. Ohm, C. van Eldik, and K. Egberts, “ γ /hadron separation in very-high-energy γ -ray astronomy using a multivariate analysis method,” *Astroparticle Physics*, vol. 31, pp. 383–391, Jun 2009.
- [3] K. Egberts, J. Hinton, and H.E.S.S. Collaboration, “Towards measuring the cosmic ray electron spectrum from ground level,” *Advances in Space Research*, vol. 42, pp. 473–476, Aug. 2008.
- [4] R. K. Bock, A. Chilingarian, M. Gaug, F. Hakl, T. Hengstebeck, M. Jiřina, J. Klaschka, E. Kotrč, P. Savický, S. Towers, A. Vaiciulis, and W. Wittek, “Methods for multidimensional event classification: a case study using images from a Cherenkov gamma-ray telescope,” *Nuclear Instruments and Methods in Physics Research A*, vol. 516, pp. 511–528, Jan. 2004.
- [5] M. Lemoine-Goumard, B. Degrange, and M. Tluczykont, “Selection and 3D-reconstruction of gamma-ray-induced air showers with a stereoscopic system of atmospheric Cherenkov telescopes,” *Astroparticle Physics*, vol. 25, pp. 195–211, Apr. 2006.
- [6] R. D. Parsons and J. A. Hinton, “A Monte Carlo template based analysis for air-Cherenkov arrays,” *Astroparticle Physics*, vol. 56, pp. 26–34, Apr. 2014.
- [7] M. de Naurois and L. Rolland, “A high performance likelihood reconstruction of γ -rays for imaging atmospheric Cherenkov telescopes,” *Astroparticle Physics*, vol. 32, pp. 231–252, Dec. 2009.
- [8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, pp. 1097–1105, 2012.
- [9] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detectors,” pp. 1–18, 2012.
- [10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.
- [11] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.
- [12] K. He, G. Gkioxari, P. Dollar, and R. Girshick, “Mask R-CNN,” in *Proceedings of the IEEE International Conference on Computer Vision (2017)*.
- [13] D. Nieto, A. Brill, B. Kim, T. B. Humensky, Others, C. T. A. Consortium, and Others, “Exploring deep learning as an event classification method for the Cherenkov Telescope Array,” *arXiv preprint arXiv:1709.05889*, pp. 1–8, 2017.
- [14] T. L. Holch, I. Shilon, M. Büchele, T. Fischer, S. Funk, N. Groeger, D. Jankowsky, T. Lohse, U. Schwanke, and P. Wagner, “Probing convolutional neural networks for event reconstruction in gamma-ray astronomy with cherenkov telescopes,” *arXiv preprint arXiv:1711.06298*, 2017.
- [15] S. Mangano, C. Delgado, M. I. Bernardos, M. Lallena, J. J. R. Vázquez, C. Consortium, *et al.*, “Extracting gamma-ray information from images with convolutional neural network methods on simulated cherenkov telescope array data,” in *IAPR Workshop on Artificial Neural Networks in Pattern Recognition*, pp. 243–254, Springer, 2018.

- [16] I. Shilon, M. Kraus, M. Büchele, K. Egberts, T. Fischer, T. L. Holch, T. Lohse, U. Schwanke, C. Steppa, and S. Funk, “Application of deep learning methods to analysis of imaging atmospheric cherenkov telescopes data,” *Astroparticle Physics*, vol. 105, pp. 44–53, 2019.
- [17] T. Vuillaume, M. Jacquemont, L. Antiga, A. Benoit, P. Lambert, G. Maurin, G. Silvestri, and the CTA consortium, “Gammalearn - first steps to apply deep learning to the cherenkov telescope array data,” in *CHEP 2018 - 23rd International Conference on Computing in High Energy and Nuclear Physics*, 2018.
- [18] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, “Automatic differentiation in pytorch,” in *NIPS-W*, 2017.
- [19] “Ignite.” <https://github.com/pytorch/ignite>, 2017–2018.
- [20] M. Jacquemont., L. Antiga., T. Vuillaume., G. Silvestri., A. Benoit., P. Lambert., and G. Maurin., “Indexed operations for non-rectangular lattices applied to convolutional neural networks,” in *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 5: VISAPP*, pp. 362–371, INSTICC, SciTePress, 2019.
- [21] M. Ragan-Kelley, F. Perez, B. Granger, T. Kluyver, P. Ivanov, J. Frederic, and M. Bussonnier, “The jupyter/ipython architecture: a unified view of computational research, from interactive exploration to communication and publication.,” in *AGU Fall Meeting Abstracts*, 2014.
- [22] J. D. Hunter, “Matplotlib: A 2d graphics environment,” *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.
- [23] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et al.*, “Tensorflow: A system for large-scale machine learning,” in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pp. 265–283, 2016.
- [24] A. Brill, Q. Feng, M. Jacquemont, B. Kim, T. Miener, D. Nieto, and T. Vuillaume, “Studying deep convolutional neural networks with hexagonal lattices for imaging atmospheric cherenkov telescope event reconstruction,” in *ICRC 2019, these proceedings*, 2019.