

Particle identification with an electromagnetic calorimeter using a Convolutional Neural Network

Alex Rua Herrera^{1,*}, Míriam Calvo Gómez^{1,**}, and Xavier Vilasís Cardona^{1,***}

¹La Salle, Universitat Ramon Llull
Sant Joan de La Salle 42
08022 Barcelona
Spain

Abstract. The LHCb's Electromagnetic Calorimeter (ECAL) measures the energy that any particle leaves behind when it travels through its sensors. However, with the current granularity, it is not possible to exploit the shape of the shower produced by the particle when it interacts with the ECAL, which is an information that could be enough to conclude what particle is being detected. In an attempt to find out whether it would be possible to classify them in future runs of the LHC, simulated data is generated with Geant4, giving an idea of what SPACAL, an updated version of the current calorimeter with better resolution, is capable of. Convolutional Neural Networks are applied so that the algorithm can understand the shapes and energy deposits produced by each kind of particle. Results obtained demonstrate that bigger resolution in ECAL allows over 95% precision in some classifications such as photons against neutrons.

1 Introduction

The Large Hadron Collider (LHC) is the world's largest and most powerful particle accelerator [1]. It accelerates protons to collide each others at specific locations at speeds that tend to the speed of light. The Large Hadron Collider beauty (LHCb) is one of the high-energy physics experiments located at the LHC and one of its purposes is to investigate the CP violation, which could explain why the universe is made mainly of matter [2].

Convolutional Neural Networks (CNN), in deep learning, are a type of artificial neural networks which, through supervised learning, can learn to capture features in images and effectively classify them [3]. The objective of the present work is to verify whether, by improving the resolution of the current Electromagnetic Calorimeter, Convolutional Neural Networks can classify the different type of particles produced in the proton-proton collisions at LHCb.

The LHCb detector consists of several layers for gathering data to reconstruct what particle traverse in a specific moment. One of them is the Electromagnetic Calorimeter (ECAL) [4], where all kind of detectable particles (electrons, muons, photons, protons, kaons, pions, etc.) deposit some energy. While charged particles are also detected in other subsystems, ECAL is the only one that detects all visible particles, both neutral and charged.

*e-mail: alex.rua@salle.url.edu

**e-mail: miriam.calvo@salle.url.edu

***e-mail: Xavier.Vilasís.Cardona@cern.ch

This document consists of a first theoretical part where it is explained the related research that has been carried out. Afterwards, there is a section that details the data that has been used for the subsequent experimental part. Then, there is a detailed explanation about the implemented models and the results obtained. Finally, there is a brief discussion and the presentation of the conclusions.

2 State of the Art

A typical high energy physics detector has several layers, as shown in Fig. 1, where each type of particle leaves a different signature.

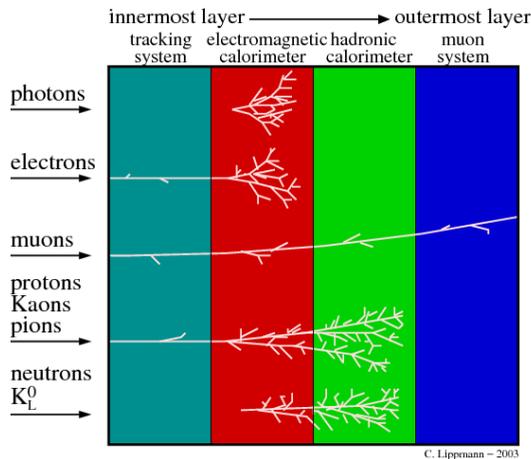


Figure 1. Particles draw patterns as they cross the detectors [5].

In the first layer, the tracking system, charged particles such as electrons, muons, protons, kaons and pions can be detected. On the other hand, in the Electromagnetic Calorimeter, which is the layer that lies immediately behind, all types of particles can be detected. However, electrons and photons are the only particles that deposit all their energy, and do not penetrate into deeper layers. Muons deposit little energy in the calorimeters, while the rest of the particles develop a shower. Hadrons, such as protons, kaons, pions and neutrons, deposit all their energy in the penultimate layer, which is the hadronic calorimeter. In the last layer, only muons can be detected.

Particle identification algorithms exploit information of several subdetectors. In the particular case of neutral particles, such as photons and neutral pions, only calorimeter information can be used for their identification. Current neutral particle identification tools in LHCb are based on shallow neutral networks [6]. As the ECAL detects every particle, which leaves different shower shapes and intensities depending on its nature, some research tried to classify the particles with information just from this layer. One of the algorithms used is Convolutional Neural Network, but the limited resolution of the current LHCb calorimeter causes these results to be very inaccurate. Having three different regions, whose square cells have lengths of 4 cm, 6 cm and 12 cm, the best resolution is not enough to apply a CNN [7].

This document explains the performance of the application of CNN for image classification with an upgraded calorimeter of LHCb for Run 5 of the LHC. In particular, the study has been done with simulation of a SPACAL module [8], a new technology that increases the image definition with smaller cells (one cell every 15 mm) and with separate front and

rear sections, being the front 4 cm long and 10 cm the rear. Note that the final design of the upgraded calorimeter is still under discussion and that the small granularity used here would be implemented only in the innermost region of ECAL, with three other regions of increasing granularity.

3 Data Sample

The datasets used to feed this project are generated in a Geant4 simulation of the SPACAL prototype, a toolkit for simulating the passage of particles through matter [9]. In total, 4,000 samples are generated for each type of charged particles and 8,000 for neutral particles. The incident angle and position of the particle with respect to the central cell is varied randomly within the typical values of particles produced in proton-proton collisions at LHCb. Only a 7x7 matrix of cells is used in the simulation. This amount of simulations allows, among others, making the balanced combinations in Table 1.

The data for each record consists of two 7x7 pixel images: each pixel captures the intensity of the energy deposited. The first image is the representation of the front sensor and the second image is the one at the rear side, which the neural network have interpreted as one single image with two channels.

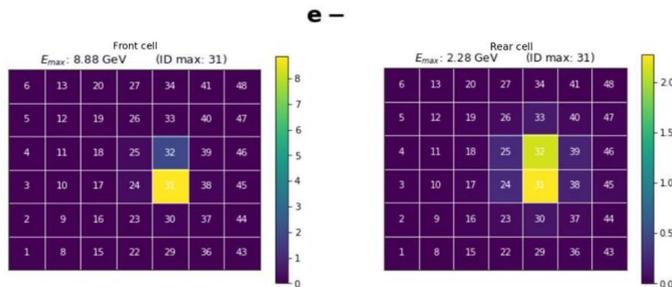


Figure 2. Example of simulation of electron detection at SPACAL's front and rear cells.

Images may have noise, so a threshold of 1 MeV was applied to the energy deposited in each cell. Apart from this, no additional cleaning have been necessary.

4 Tools

Python has been the main programming language. As a development environment, Google Colab has been used. It is an environment with a simple and user-friendly interface. Also, when running on remote servers, it allows the selection of different processors: CPU, GPU, and TPU. The main library used in Python have been PyTorch, a replacement module for Numpy optimized for the use of GPUs and the development of neural networks [10].

Additionally, MLFlow, an open source platform for the machine learning lifecycle, has been tested to save configurations and results of every model tested.

5 Development

Convolutional Neural Networks have been shown to be successful in image classification. However, it is necessary to select the appropriate configuration and hyperparameters to

Table 1. Balanced combinations of particles.

Combination number	Alias	Purpose	Particles included
Combination 1	Neutral Particles	Limit the classification to neutral particles	Neutrons: n Photons: γ
Combination 2	Three Particles	Limit the classification to three particles with different behaviours in shower generation and energy deposit	Electrons: e^- Positive Muons: μ^+ Positive Pions: π^+
Combination 3	Particles without charge differentiation	Same purpose of "combination 4", without charge differentiation	Electrons and positrons: e^-, e^+ Photons: γ Muons: μ^+, μ^- Pions: π^+, π^- Kaons: K^+, K^- Protons and antiprotons: p, \bar{p} Neutrons: n
Combination 4	All particles	Classification of 12 different particles	Electrons: e^- Positrons: e^+ Photons: γ Positive Muons: μ^+ Negative Muons: μ^- Positive Pions: π^+ Negative Pions: π^- Positive Kaons: K^+ Negative Kaons: K^- Protons: p Antiprotons: \bar{p} Neutrons: n

achieve the optimal solution, that would shorten the training time and improve the accuracy in the validation and test sets. In this research project, the following values have been tested:

I. Number of neurons:

The smallest number of neurons per layer has been 2 neurons while the maximums, which are found after flattening the convolutions, have had values up to 512 neurons.

II. Learning rates:

This parameter has ranged between 0.0001 and 0.1. However, the best results have been found with 0.01 and 0.001.

III. Number of layers and activation functions:

The number of fully connected layers has ranged from one to four. Apart from them, the following activation functions have been tested: ReLU, Leaky ReLU, Softmax, LogSoftmax, Sigmoid and LogSigmoid. The worst results have been found with Sigmoid and LogSigmoid outputs, while the best results had in common a configuration of three fully connected layers.

IV. Number of convolutions:

There has been neural nets without convolutions and just fully connected layers. However, the best results have been found in configurations with one and two convolutions (kernels of 2x2) and maxpool (different paddings and strides).

V. Optimizers:

The first optimizer used had been Adam. Afterwards, the best values were found with AdaMax, a variant of Adam which uses the infinite rule to improve stability [11]. Finally, Stochastic Gradient Descend has been tested with unsuccessfully results.

VI. Loss functions:

The functions to summarize the difference between predicted and real values have been Negative Log Likelihood Loss and Cross Entropy Loss, with similar success.

VII. Batch sizes and epochs:

The number of epochs has ranged between 10 and 3,000 epochs (Fig. 3), depending on the batch sizes. In an attempt to reduce the learning time, the preference has been the combination of little batch sizes along with bigger number of epochs [12].

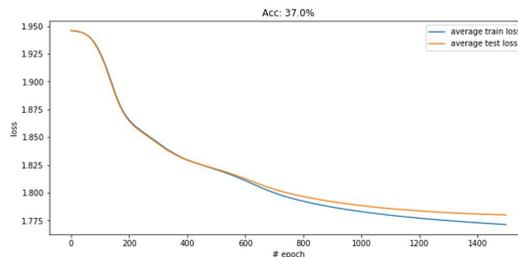


Figure 3. Loss evolution over 1,500 epochs.

Finally, 80% of the data was separated to perform a 5-fold cross-validation, while the remaining 20% of the data was used for testing.

6 Results

The following subsections show the best results achieved with different nets for each particle classification with SPACAL.

6.1 Neutral particles

The simplest classification has involved two types of particles: photons and neutrons. Photons deposit all their energy in the electromagnetic calorimeter, while neutrons deposit just a part of it. However, both of them produce a shower.

Table 2. Performance on neutral particles classification.

Particle	Precision	Recall	False Positive Rate	F1
n	0.97	0.956	0.03	0.963
γ	0.955	0.969	0.044	0.962

Table 3. Performance on three particles classification.

Particle	Precision	Recall	False Positive Rate	F1
e^-	0.942	0.963	0.031	0.953
μ^+	0.868	0.480	0.034	0.618
π^+	0.682	0.961	0.222	0.798

After having trained the model, the test set (1,600 samples) show an accuracy greater than 95% (see Table 2 and Fig. 4). In fact, cross-validation showed a mean accuracy of 95.5% with a standard deviation of 0.3%.

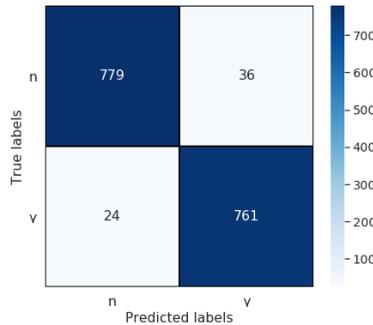


Figure 4. Confusion matrix: Neutral particles classification after testing 1,600 samples.

6.2 Three particles

This case classifies three different types of particles in which, at least, one feature is shared: electrons have in common with pions the shower generated in the ECAL, while pions have in common with muons that none of them deposits all the energy in this layer. Finally, the electrons with the muons do not share any of the characteristics: neither showers nor energy deposited.

Results showed the energy deposited in the ECAL and the shower generation in these three particles is different enough to make a very precise classification (Fig. 5). However, pions had a big False Positive Rate and Muons a big False Negative Rate (Table 3).

Cross-validation showed an average accuracy of 79.2% with a standard deviation of 1.1%.

6.3 Particles without charge differentiation

This subgroup contains all the particles without charge differentiation. Consequently, there are different particles that share behaviours in the ECAL: energy deposit and/or shower generation (Table 4 and Fig. 6). Average accuracy during cross-validation was found to be 39.1% with a standard deviation of 0.6%.

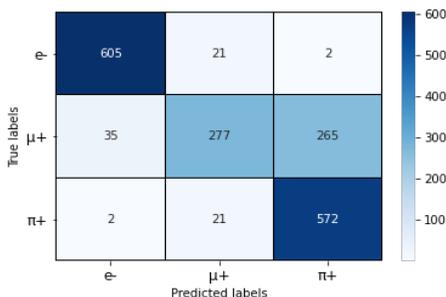


Figure 5. Confusion matrix: 3 particles classification after testing 1,800 samples.

Table 4. Performance on particles classification without charge consideration.

Particle	Precision	Recall	False Positive Rate	F1
e	0.48	0.739	0.137	0.582
γ	0.445	0.385	0.08	0.413
π	0.238	0.032	0.017	0.057
μ	0.36	0.987	0.291	0.528
K	0.247	0.171	0.086	0.202
p	0.242	0.151	0.077	0.186
n	0.342	0.128	0.041	0.187

Considering photons and electrons into a single category, because the electromagnetic showers that they produce are indistinguishable, the precision and recall was found to be 0.776 and 0.935, respectively.

6.4 All particles

The pattern that is drawn in this subsection is exactly the same as the one seen in the previous subsection: muons are an independent group, electrons and photons belong to other independent group and the other hadrons belong to a third group. In this case, it is intended to differentiate between charges, but it is unsuccessful and, physically, impossible without information from the tracking system. As a result, the overall precision decreases, but at a general level, there is a classification of three groups of particles (Table 5 and Fig. 7), as seen before:

- The first group includes e^- , e^+ and γ . Electromagnetic particles here deposit all their energy and produce wider showers.
- The second group is composed of μ^+ and μ^- . The energy deposited is lower and there is no shower.
- Hadrons (n , p , \bar{p} , π^+ , π^- , K^+ and K^-) belong to the last group. This group shows lower energy, in particular, in the front cells, so they might be confused with muons. However, hadrons usually create showers later, so they could deposit bigger amounts of energy in the rear cells.

Cross-validation showed an average accuracy of 24.1% with a standard deviation of 1.2%.

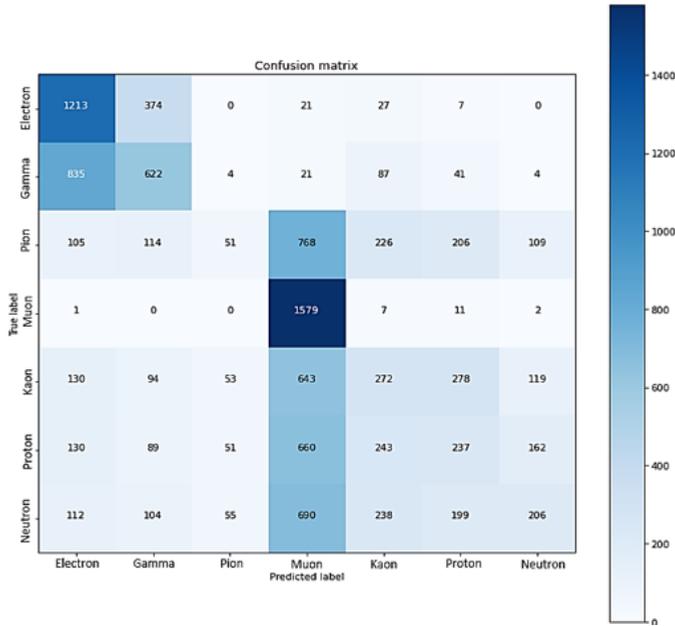


Figure 6. Confusion matrix: 11,200 samples without charge differentiation.

Table 5. Performance on particles classification.

Particle	Precision	Recall	False Positive Rate	F1
e^-	0.297	0.362	0.083	0.326
e^+	0.37	0.429	0.07	0.397
γ	0.333	0.258	0.055	0.291
π^-	0.123	0.038	0.026	0.059
π^+	0.14	0.079	0.042	0.101
μ^-	0.201	0.361	0.126	0.258
μ^+	0.207	0.555	0.174	0.302
K^-	0.129	0.056	0.037	0.078
K^+	0.204	0.139	0.051	0.166
p	0.118	0.108	0.063	0.113
n	0.497	0.448	0.039	0.471
\bar{p}	0.094	0.072	0.056	0.082

7 Algorithm comparison

CNNs are algorithms that have demonstrated a good precision in particle classification with SPACAL resolution. However, this resolution allows other algorithms to perform well, too. Table 6 shows the best results obtained by a decision tree (XGBoost [13]) after careful tuning. Parameters tested include, but are not limited to:

- Number of estimators: from tens to hundreds trees.
- Learning rates: values ranging from 0.01 to 0.1.
- Maximum depth: values from units to few tens to assure a good precision while avoiding overfitting.

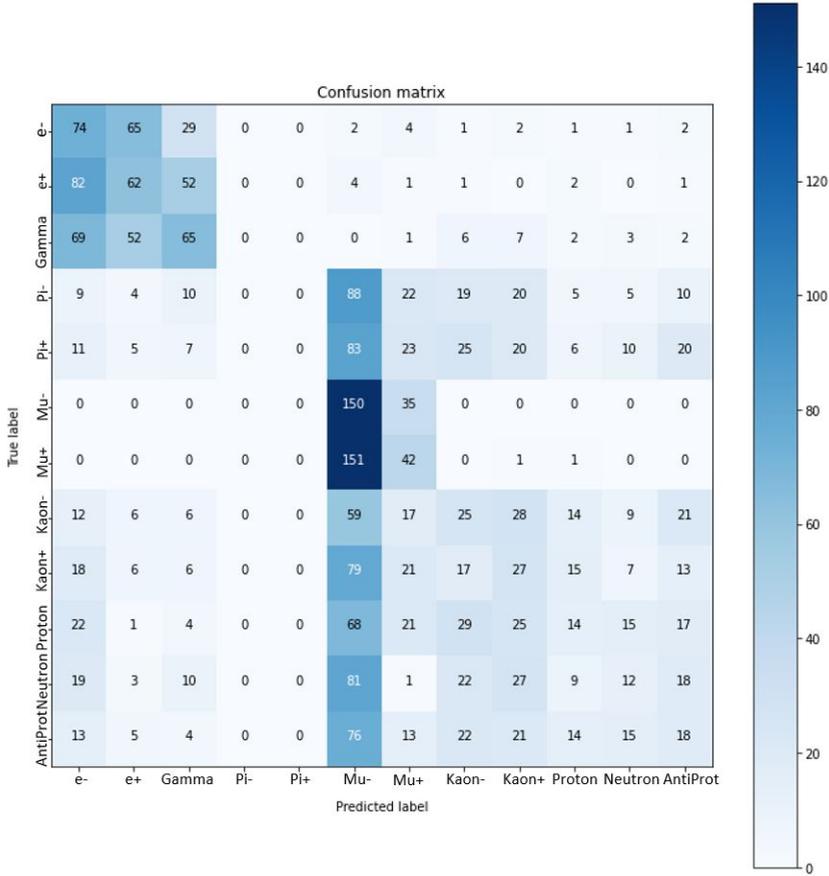


Figure 7. Confusion matrix: 12 particles after 2,400 samples.

Table 6. Accuracy comparison between CNNs and XGBoost.

Classification	CNN	XGBoost	XGBoost (Image centering)
Neutral particles	96.3%	85.4%	90.1%
Three particles	82.0%	77.2%	78.9%
Seven particles	37.0%	34.9%	38.4%
All particles	24.2%	21.9%	23.2%

- Subsample: different sampling values from 0 to 1 to prevent overfitting.

Finally, returned precision is about 10% below the CNN’s. The same decision tree was applied after image centering (ie. reducing the 7x7 matrix to 5x5 where the center contains the maximum energy) and the results improved significantly.

8 Conclusions

After having used the SPACAL simulation, it can be verified that CNNs are algorithms that perform well with the future upgrade of the ECAL (Run 5), despite they do not work for the

current one (Run 2). Specifically, it allows us to see three differentiated subgroups of particles with high accuracy:

- A first subgroup where electromagnetic particles are included: electrons, positrons and photons.
- A second subgroup consisting of muons.
- A third subgroup where hadrons are included: neutrons, pions, protons and kaons.

Acknowledgments

During the development of this work, meetings have been held with the University of Cincinnati (USA) and Higher School of Economics (Russia). These meetings have been helpful to increase knowledge of the authors of this research and for this we want to show our gratitude.

Finally, we want to thank Yanxi Zhang for the implementation of Spacal in Geant4, which allowed us to generate the necessary data.

References

- [1] Lyndon Evans and Philip Bryant. *LHC machine*. JINST 3 (2008) S08001.
- [2] LHCb collaboration, A. A. Alves Jr. et al. *The LHCb detector at the LHC*. JINST 3 (2008) S08005.
- [3] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton. *ImageNet classification with deep convolutional neural networks*. DOI: 10.1145/3065386, Communications of the ACM, Volume 60, Issue 6.
- [4] LHCb collaboration. *LHCb calorimeters: Technical Design Report*. CERN-LHCC-2000-036; LHCb-TDR-2.
- [5] Christian Lippmann *Particle Identification, Figure 1*, arXiv:1101.3276v4.
- [6] R. Aaij et al. *LHCb detector performance*. *Int. J. Mod. Phys. A30 (2015) 1530022*, arXiv:1405.7808.
- [7] D. Hilazo Aguilera. *Identificación de partículas mediante técnicas de clasificación en el calorímetro del LHCb*. Final degree project in computer engineering. La Salle, Universitat Ramon Lull. July 2019, Page 9. Accessed: November, 2020.
- [8] Loris Martinazzoli. *Crystal Fibers for the LHCb Calorimeter Upgrade*. IEEE Trans. Nucl. Sci. 67 (2020) 1003.
- [9] Geant4 collaboration, S. Agostinelli et al. *Geant4: A simulation toolkit*. Nucl. Instrum. Meth. A 506 (2003) 250.
- [10] *PyTorch* <https://github.com/pytorch/pytorch/>, Accessed: 27-May-2020.
- [11] Diederik P. Kingma, Jimmy Ba, *Adam: A Method for Stochastic Optimization*, arXiv: 1412.6980v9.
- [12] T. R. Martinez & D. R. Wilson, *The general inefficiency of batch training for gradient descent learning*, Neural Networks 16 (2003) 1429–1451.
- [13] Tianqi Chen & Carlos Guestrin, *XGBoost: A Scalable Tree Boosting System*, arXiv:1603.02754v3.