Deep Learning for Energy Estimation and Particle Identification in Gamma-ray Astronomy[⋆]

Evgeny Postnikov[1][0000−0002−3829−0379], Alexander Kryukov[1][0000−0002−1624−6131], Stanislav Polyakov[1][0000−0002−8429−8478], and Dmitry Zhurov[2,3][0000−0002−1596−8829]

[1] Lomonosov Moscow State University, Skobeltsyn Institute of Nuclear Physics (SINP MSU), 1(2), Leninskie gory, GSP-1, Moscow 119991, Russian Federation
evgeny.post@gmail.com, kryukov@theory.sinp.msu.ru, s.p.polyakov@gmail.com
http://www.sinp.msu.ru/en
[2] Institute of Applied Physics of Irkutsk State University, 20, Gagarin bulvard 664003 Irkutsk, Russian Federation
sidney28@ya.ru
http://api.isu.ru
[3] Irkutsk National Research Technical University, 83, Lermontova str. 664074 Irkutsk, Russian Federation
http://www.istu.edu/eng

Abstract. Deep learning techniques, namely convolutional neural networks (CNN), have previously been adapted to select gamma-ray events in the TAIGA experiment, having achieved a good quality of selection as compared with the conventional Hillas approach. Another important task for the TAIGA data analysis was also solved with CNN: gamma-ray energy estimation showed some improvement in comparison with the conventional method based on the Hillas analysis. Furthermore, our software was completely redeveloped for the graphics processing unit (GPU), which led to significantly faster calculations in both of these tasks. All the results have been obtained with the simulated data of TAIGA Monte Carlo software; their experimental confirmation is envisaged for the near future.

Keywords: Deep learning · CNN · Gamma-ray astronomy.

1 Introduction

1.1 Gamma-ray astronomy

Gamma-ray detection is very important for observing the Universe as gamma-rays are particles without electric charge and are unaffected by a magnetic field. Detected gamma-rays can therefore be extrapolated back to their origin. For

---

that reason, they are currently the best "messengers" of physical processes from the relativistic Universe.

With specially designed telescopes, gamma-rays can be detected on Earth (ground-based gamma-ray astronomy) at very high energies. These instruments are called Imaging Air Cherenkov Telescopes (IACTs) [1]. Gamma-rays are observed on the ground optically via the Cherenkov light emitted by extensive showers of secondary particles in the air when a very-high-energy gamma-ray strikes the atmosphere.

However, very-high-energy gamma-rays contribute only a minuscule fraction to the flux of electrically charged cosmic rays (below one per million [2]). This circumstance makes it necessary to learn to distinguish gamma-rays against charged cosmic rays, mostly protons, on the basis of the images they produce in the telescope camera.

1.2 Data Life Cycle project in Astroparticle Physics

The Russian-German Initiative of a Data Life Cycle in Astroparticle Physics (also referred to as Astroparticle.online) [3, 4] aims to develop an open science system for collecting, storing, and analyzing astroparticle physics data including gamma-ray astronomy data. Currently it works with the TAIGA [5] and KASCADE [6] experiments and invites astrophysical experiments to participate.

In this work, two important problems of gamma-ray astronomy data analysis are solved within the framework of deep learning approach (convolutional neural networks). These are the background rejection problem (removal of cosmic ray background events), and the gamma-ray energy estimation problem, in imaging air Cherenkov telescopes. The data to solve the both problems were simulated using the complete Monte Carlo software for the TAIGA-IACT installation [7].

1.3 Convolutional Neural Networks (CNNs)

CNNs are well adapted to classify images; that is why they were also chosen for all deep learning applications to the IACT technique [8–10]. Their advantage is a fully automatic algorithm, including automatic extraction of image features instead of a set of empirical parameters ('Hillas parameters' [11]). CNNs are implemented in various free software packages, including PyTorch [12] and TensorFlow [13]. In contrast to the camera with square pixels [8], a shape and arrangement of pixels of the TAIGA-IACT camera is hexagonal, and this geometrical feature has not been fully taken into account yet.

2 Data simulations

Data simulations were performed to obtain datasets with the response of a real IACT telescope for two classes of particles to be identified: gamma-rays and background particles (protons). The development of the shower of secondary particles in the atmosphere was simulated with the CORSIKA package [14].

The response of the IACT system was simulated using the OPTICA-TAIGA software developed at JINR, Dubna [7]. It describes the real TAIGA-IACT setup configuration: 29 constituent mirrors with an area of about 8.5 m^2 and a focal length of 4.75 m, and the 560-pixel camera located at the focus. Each pixel is a photomultiplier (PMT) collecting light from the mirrors.

The telescopic image was formed using a dedicated software developed at SINP MSU taking into account the night sky background fluctuations, PMT characteristics, and triggering and readout procedures of the data acquisition system.

3 Image cleaning

Image cleaning is a conventional procedure to remove images and image parts produced by the night sky background fluctuations but not by a shower of secondary particles. The conventional procedure is two-parametric: it excludes from subsequent analysis all image pixels except the core pixels, i.e. those with the amplitude above a core threshold and at least one neighbour pixel above a neighbour threshold, and the neighbour pixels themselves. If the image contains too few pixels after cleaning (for example, 2 or less), the entire image is excluded from the analysis.

Deep learning algorithms were trained on images both without and with cleaning. For the reference technique, a test sample was first subjected to the image cleaning procedure in any case. No training sample was needed for the reference technique.

4 Deep learning

4.1 Data sample

Training datasets for the CNN contained gamma-ray and proton images (Monte Carlo of TAIGA-IACT) for the task of background suppression, and only gamma-ray images for the energy estimation. Image examples are presented in Figure 1.

The dataset consisted of 2.7×10^4 simulated events after strong image cleaning (70% training + 30% test) for PyTorch, and of 5.6×10^4 events after soft image cleaning (of which 60% were used for training, 15% for validation, and 25% for testing) for TensorFlow. The images in the training dataset were rotated around the camera center by multiples of 60° thereby producing 6 times the initial sample size. Finally, the total number of events was about 2×10^5 for training (with rotations), 0.8×10^4 for validation, and 1.5×10^4 for testing.

4.2 CNN implementation

Seeing that convolutional operations are optimized for a square grid, the TAIGA-IACT hexagonal camera grid was needed to be represented in convenient form
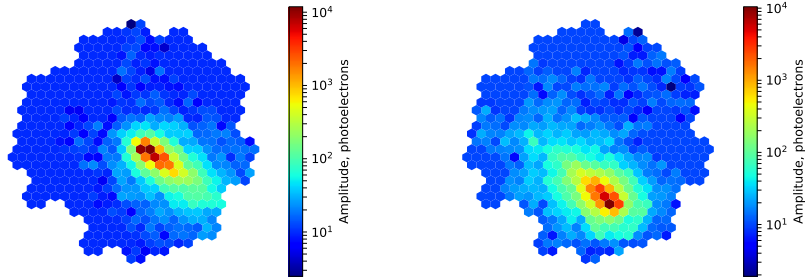
Fig. 1. Simulated image examples: generated by high-energy gamma-ray (left) and proton (right).

to fit the square one. For that purpose, a transformation to oblique coordinate system was applied to each image, so that each hexagonal image with 560 pixels was transformed to the 31x30 square grid. These square grid images were fed to the input layer of the CNN.

For the background suppression, test datasets of gamma-ray and proton images in random proportion (blind analysis) were classified by each of the packages: TensorFlow and PyTorch.

The energy was either directly predicted as a scalar parameter by the CNN, or the ratio of the energy to the total sum of the amplitudes in the image was predicted and then multiplied back by the value of the total sum to obtain the energy estimate. The reason for the second way to estimate the energy is that the above mentioned total sum of the amplitudes, referred to as 'image size', is correlated with the energy for gamma-rays incident closer than ≈100 m from the telescope [15]. Therefore, image size can be in some way directly included in the estimation algorithm to account for this strong correlation at least for nearby gamma-rays. Beyond this 'Cherenkov radius' of about 100–120 m, the Cherenkov light intensity varies rapidly with the distance from gamma-ray to the telescope, which may also lead to a substantial increase of the resulting uncertainty in the energy estimation.

Various networks with different parameters were tested to find the one maximizing background suppression and the one minimizing the relative energy error.

4.3 CNN architecture

The first part of the convolutional neural network consists of convolutional layers (Figure 2). Each layer includes convolution with 32 kernels of 3x3 pixels and the ReLU activation function, average pooling layer with 3x3 pooling size and strides of 2 pixels. To avoid overfitting during the training, a dropout layer with a dropout rate of 10% is added after each pooling layer.
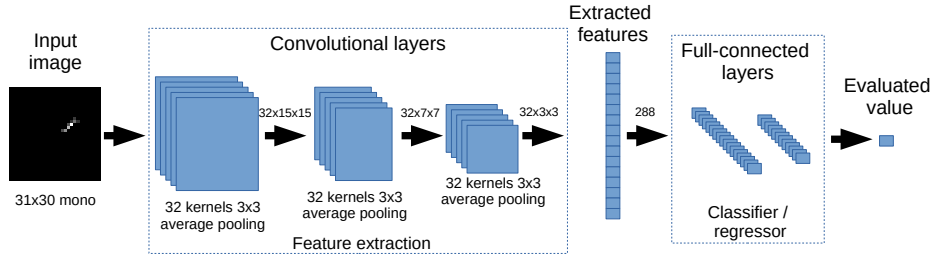
Fig. 2. Convolutional neural network for classification/regression. The network accepts square grid images in oblique coordinate system at the input of convolutional layers. Output of the convolutional layers (extracted features) is fed to the classifier/regressor (full-connected layers) that evaluate the output value.

Output of the convolutional layer is fed to the full-connected layers of classifier or regressor. The full-connected layers consist of 32 neurons with the ReLU activation function in the first layer and 16 neurons in the second one. Dropout with a 50% rate after each full-connected layer is used to avoid overfitting.

Sigmoid was set as the output neuron activation function for the classification task, whereas no activation function was set to the output neuron for the energy estimation. Adagrad optimizer with the learning rate set at 0.05 and the binary cross-entropy as the loss function were used for classification. The energy estimation was performed using Adam optimizer and the mean square error as the loss function.

The early stop criterion was set to interrupt the training procedure when the loss function for the validation dataset shows no decrease for 30 epochs. The training lasted for 144 epochs (runtime $\sim$9 minutes). The computational graph was run on NVIDIA GPU Tesla P100.

Accuracy on the training and validation sample after training was 91.29% and 90.02% respectively. ROC AUC score (an area under the receiver operating characteristic curve [16]) was 0.9712 for training and 0.9647 for validation.

5 Results

5.1 Scalar quality criterion (Q-factor)

As a quality criterion of particle identification, the selection quality factor Q was estimated. This factor indicates an improvement of a significance of the statistical hypothesis that the events do not belong to the background in comparison with the significance before selection. For Poisson distribution (that is for a large number of events), the selection quality factor is:

$$Q = \epsilon_{nuclei}/\sqrt{\epsilon_{bckgr}}, \tag{1}$$

where ϵ_{nuclei} and ϵ_{bckgr} are relative numbers of selected events and background events after selection. For our task we consider protons as a background to select gamma-rays above this background.

Quality factor (1) values obtained by the best CNN configuration among all the trained networks are assembled in Table 1 together with the quality factor for the reference technique (simplest Hillas analysis [17]).

CNN was accelerated on graphics processing unit (GPU), which led to a significantly faster calculation. Its implementation was approximately 6 times faster than an equivalent implementation on CPU and revealed no quality loss (the last column of Table 1).

Table 1. Q-factor for gamma/proton identification.

Image cleaning	Reference	CNN(PyTorch)	CNN(TensorFlow)	CNN(TensorFlow GPU)
No	1.76	1.74	1.48	
Yes	1.70	2.55	2.91	2.86

5.2 Q-factor variability

Comparison of different CNN versions for both software packages is illustrated in Figure 3.

PyTorch had more stable results in a wide range of CNN output parameter values. However, significant improvement was obtained with the TensorFlow CNN version trained by a modified training sample, which contained both original simulated images and additional ones obtained by rotating images from the initial sample by the symmetry angles of hexagonal structure. Thus the modified training sample consisted of $\sim 2 \times 10^5$ events instead of $\sim 3 \times 10^4$. Therefore, the performance of different software packages was approximately the same, indicating that the training sample size was crucial for the identification quality.

5.3 Energy error

The predicted energy RMSE is 2.56 TeV for training sample, 3.31 TeV for validation dataset, and 4.76 TeV for the TensorFlow test sample (7.82 TeV for the PyTorch test sample). The dependence of the predicted energy on the primary energy is shown in Figure 4. Absolute error distribution is presented in Figure 5.

The accuracy of the energy estimate depends on the image distance from the camera center, which corresponds to the distance of the gamma-ray induced shower to the telescope. The energy error is presented in Figure 6 for various angular distances from the image centre of gravity to the centre of the camera's field of view. This angular distance is strongly correlated with the distance from the gamma-ray-induced shower to the telescope, but is measurable in experiment unlike the unknown distance to the telescope. The angular distance of $\sim 1.5^o$ corresponds roughly to ~ 100–150 m [18].
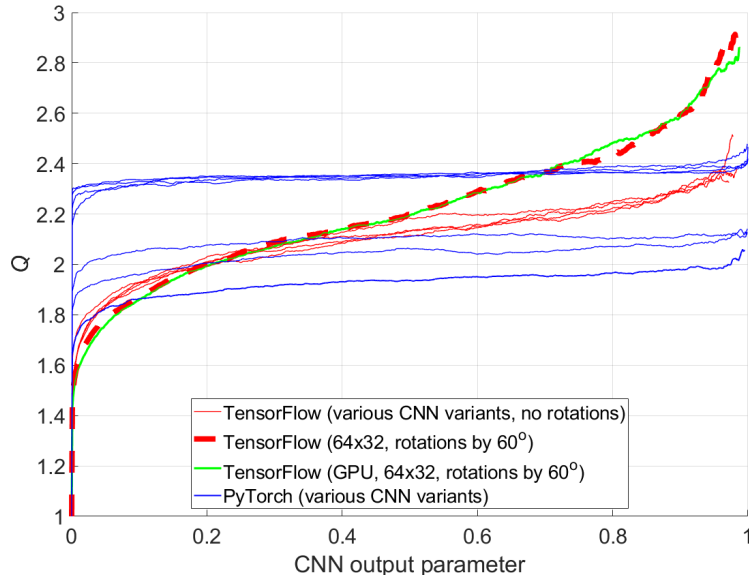
Fig. 3. Quality factor vs CNN output parameter (a scalar parameter between 0 and 1 characterizing image similarity to gamma-ray or proton).

Though for the nearest gamma-rays there is no improvement over the simplest conventional technique of a linear proportionality to the image size (section 4.2), for the distances above 1° CNN gives significantly better results, and especially does the CNN predicting the ratio of the energy to the image size instead of predicting the energy itself. However, it is not the optimal way to incorporate the image size information in the CNN, and therefore the energy estimation still contains some potential to further improve accuracy.

6   Conclusion

Convolutional neural networks were implemented to solve two important tasks of data analysis in gamma-ray astronomy: cosmic ray background suppression and gamma-ray energy estimation. Background rejection quality strongly depends on the learning sample size but in any case is substantially higher than for conventional techniques. The energy estimation achieves significantly better accuracy than conventional approach for gamma-rays incident in the area outside of a narrow circle around the telescope ($\sim$100–150 m on the ground or $\sim$1–1.5^o on the camera plane). Because of the wide acceptance of the TAIGA-IACT camera, this technique is capable of measuring energy of most gamma-rays detected by the installation.

We also note that there is still considerable potential to further improve the results by taking into account the hexagonal pixel shape and increasing training
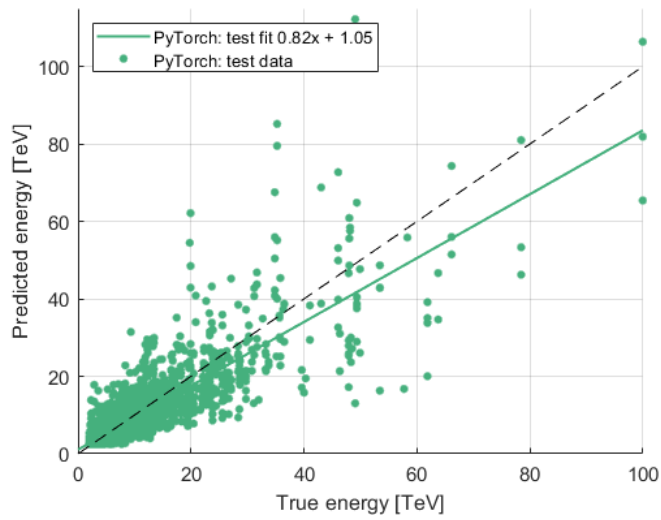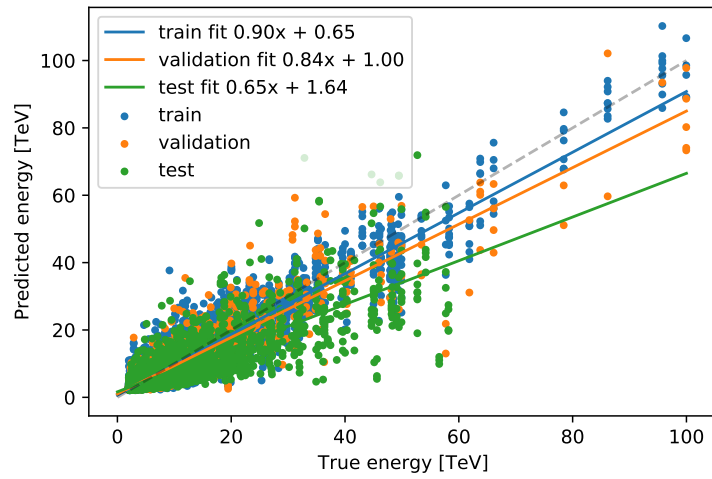
Fig. 4. Predicted energy vs true energy: top pannel TensorFlow, bottom pannel Py-Torch (dashed lines are the 'ideal case' y=x).
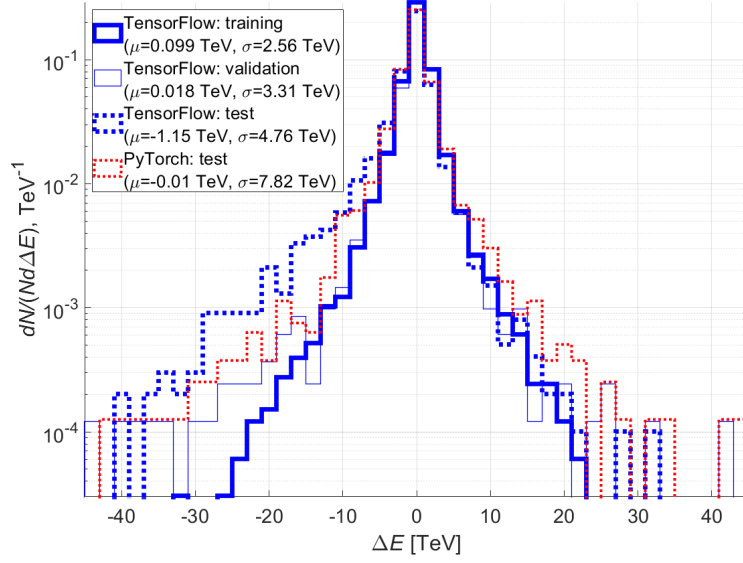
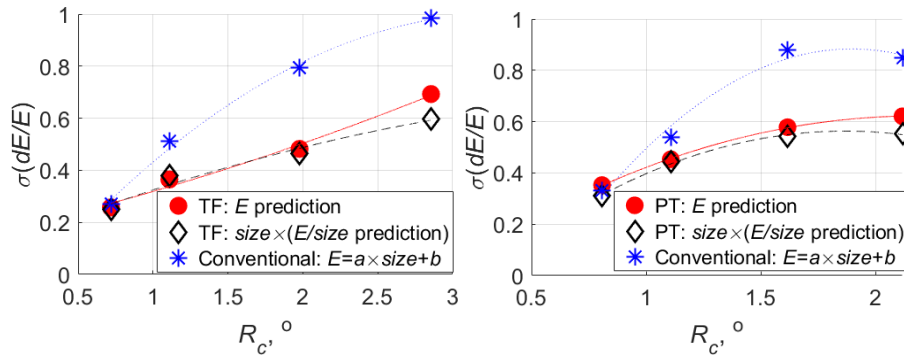**Fig. 5.** Absolute energy error distributions.

Fig. 6. Relative energy error vs angular distance of the image: TensorFlow ('TF', left), PyTorch ('PT', right).

sample size by one order of magnitude, which is a challenge for the immediate future.

References

1. Weekes, T.C. et al. [Whipple collaboration]: Observation of TeV gamma rays from the Crab Nebula using the atmospheric Cerenkov imaging technique. Astroph. J. **342**, 379 (1989)
2. Lorenz, E., Wagner, R.: Very-high energy gamma-ray astronomy. EPJ H **37**, 459 (2012)
3. Bychkov, I. et al.: RussianGerman Astroparticle Data Life Cycle Initiative. Data **3**(4), 56 (2018)
4. Astroparticle.online Homepage, urlhttps://astroparticle.online. Last accessed 15 May 2019
5. TAIGA Homepage, urlhttp://taiga-experiment.info. Last accessed 15 May 2019
6. KASCADE Homepage, urlhttp://web.ikp.kit.edu/KASCADE. Last accessed 15 May 2019
7. Postnikov, E.B., et al.: Hybrid method for identifying mass groups of primary cosmic rays in the joint operation of IACTs and wide angle Cherenkov timing arrays. J. Phys.: Conf. Series **798**, 012030 (2017)
8. Nieto, D. et al. for the CTA Consortium: Exploring deep learning as an event classification method for the Cherenkov Telescope Array. Proceedings of Science **301**, PoS(ICRC2017)809 (2017)
9. Shilon, I. et al.: Application of deep learning methods to analysis of imaging atmospheric Cherenkov telescopes data. Astroparticle Physics **105**, 44–53 (2019)
10. Feng, Q., Jarvis, J. on behalf of the VERITAS Collaboration: A citizen-science approach to muon events in imaging atmospheric Cherenkov telescope data: the Muon Hunter. Proceedings of Science **301**, PoS(ICRC2017)826 (2017)
11. Hillas, A.M.: Cerenkov light images of EAS produced by primary gamma rays and by nuclei. In: Proc. 19th Int. Cosmic Ray Conf., La Jolla, 1985, p. 445. NASA, Washington, D.C. (1985)
12. PyTorch Homepage, urlhttp://pytorch.org. Last accessed 15 May 2019
13. TensorFlow Homepage, urlhttp://www.tensorflow.org. Last accessed 15 May 2019
14. Heck, D. et al.: CORSIKA: A Monte Carlo Code to Simulate Extensive Air Showers. Report FZKA 6019. Forschungszentrum Karlsruhe (1998)
15. Hofmann, W. et al.: Improved energy resolution for VHE gamma ray astronomy with systems of Cherenkov telescopes. Astroparticle Physics **12**, 207–216 (2000)
16. Hanley, J.A., McNeil, B.J.: The meaning and use of the area under a receiver operating characteristic (ROC) curve. Radiology **143**, 29–36 (1982)
17. Postnikov, E.B., et al.: Gamma/Hadron Separation in Imaging Air Cherenkov Telescopes Using Deep Learning Libraries TensorFlow and PyTorch. J. Phys.: Conf. Series **1181**, 012048 (2019)
18. Dhar, V.K. et al.: ANN-based energy reconstruction procedure for TACTIC γ-ray telescope and its comparison with other conventional methods. Nucl. Instrum. Meth. A **606** 795–805 (2009)