PROCEEDINGS OF SCIENCE

Investigating the VHE Gamma-ray Sources Using Deep Neural Networks

V. Vodeb,[a,*] **S. Bhattacharyya,**[a] **G. Principe,**[b,c] **G. Zaharijaš,**[a] **R. Austri,**[d] **F. Stoppa,**[e] **S. Caron**[e] **and D. Malyshev**[f]

[a] *University of Nova Gorica, Center for Astrophysics and Cosmology, Vipavska 11c, Ajdovščina, Slovenia*
[b] *Istituto Nazionale di Fisica Nucleare, Sezione di Trieste, I-34127 Trieste, Italy*
[c] *Dipartimento di Fisica, Universitá di Trieste, I-34127 Trieste, Italy*
[d] *Instituto de Física Corpuscular, IFIC-UV/CSIC, Valencia, Spain*
[e] *Department of Astrophysics/IMAPP, Radboud University, PO Box 9010, 6500 GL Nijmegen, The Netherlands*
[f] *Erlangen Centre for Astroparticle Physics, Erwin-Rommel-Str. 1, Erlangen, Germany*

E-mail: veronika.vodeb@ung.si, sbhattacharyya@ung.si

The upcoming Cherenkov Telescope Array (CTA) will dramatically improve the point-source sensitivity compared to the current Imaging Atmospheric Cherenkov Telescopes (IACTs). One of the key science projects of CTA will be a survey of the whole Galactic plane (GPS) using both southern and northern observatories, specifically focusing on the inner galactic region. We extend a deep learning-based image segmentation software pipeline (autosource-id) developed on Fermi-LAT data to detect and classify extended sources for the simulated CTA GPS. Using updated instrument response functions for CTA (Prod5), we test this pipeline on simulated gamma-ray sources lying in the inner galactic region (specifically $0^\circ < l < 20^\circ$, $|b| < 3^\circ$) for energies ranging from 30 GeV to 100 TeV. Dividing the source extensions ranging from 0.03° to 1° in three different classes, we find that using a simple and light convolutional neural network it is possible to achieve a 97% global accuracy in separating the extended sources from the point-like sources. The neural net architecture including other data pre-processing codes is available online.

38th International Cosmic Ray Conference (ICRC2023)
26 July - 3 August, 2023
Nagoya, Japan

*Speaker

© Copyright owned by the author(s) under the terms of the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License (CC BY-NC-ND 4.0).

PoS(ICRC2023)599

1. Introduction

The Cherenkov Telescope Array (CTA) is an upcoming ground-based gamma-ray observatory that will comprise over 60 imaging atmospheric Cherenkov telescopes (IACTs) placed in both the northern (CTA-North) and southern (CTA-South) hemispheres [1]. Its extensive telescope array will deliver unparalleled sensitivity and imaging capabilities, providing a significant leap forward in gamma-ray observation.

The upcoming surveys conducted by CTA are anticipated to achieve two orders of magnitude deeper exposure than the currently performed surveys using IACTs [1]. This enhancement will be attributed primarily to the observatory's improved sensitivity and wider field of view (FoV). Sky surveys hold an important position within CTA's Key Science Projects, forming the fundamental scientific program of the observatory. Consequently, dedicated observation time will be allocated early on in CTA's observation schedule to ensure their successful completion. CTA will perform a Galactic plane survey (GPS) within the energy range of 20 GeV to 300 TeV, with unprecedented depth, angular resolution, and spectral coverage, using both CTA-North and CTA-South arrays.

With a target (point-source) sensitivity of the GPS at the level of a few mCrab, CTA is expected to detect a factor of 2-7 times as many astrophysical TeV sources as are in the current catalogs [2]. In the construction of high and very-high-energy gamma-ray catalogs, a standard approach to identifying new gamma-ray sources utilizes the likelihood analysis, in the form of either the classical background subtraction techniques or a modern template-fitting approach. With the advancement of state-of-the-art simulation techniques, supervised machine learning for point and extended source detection represents a new approach to cataloging the sky, potentially improving the detection sensitivity and computational effectivity achieved by likelihood-based techniques. The methods presented in this work will be part of a detection pipeline utilizing deep-learning-based algorithms to automatically detect and classify point sources from gamma-ray data [3, 4][1]. The codes related to the work presented in this proceedings paper can be accessed through a dedicated GitHub page[2].

2. Data Preparation

2.1 Simulation Using CTOOLS

To simulate the GPS observations, we utilize the methodology outlined in [2]. The planned observation schedule comprises an initial early-phase program that spans 480 hours of observation time during the first two years and a long-term program with a total of 1140 hours allocated for observations over the subsequent eight years. The planned pointing strategy for the GPS assumes a double-row pattern. The observations are scheduled specifically during the dark time, ensuring moonless nights for optimal conditions. Moreover, careful optimizations are implemented to determine the minimal zenith angles for each observation, further enhancing the effectiveness of the pointing strategy. The exposure of different regions of the Galactic plane will differ depending on the expected number of sources and objects of interest in each region, and consequently, separate regions will reach different point- and extended-source sensitivities. The Galactic plane is divided

[1] AutoSource-ID

[2] GitHub Link

PoS(ICRC2023)599

into five respective regions, with the Inner Galaxy planned to receive the most observation time and with that the deepest exposure, due to being the most densely populated with sources.

Ground-based observatories predominantly detect events originating from extensive air showers triggered by cosmic ray (CR) particles entering the Earth's atmosphere. During the event reconstruction process, some of these CR-induced events can be incorrectly identified as gamma rays, constituting the CR background component. To model this CR component accurately, extensive Monte Carlo simulations are conducted to simulate CR air showers, capture the associated Cherenkov radiation, and subsequently reconstruct the events. The IRF suite provides the rate and distribution of misidentified CR events, which vary depending on observing conditions such as the duration of observations and the zenith angle.

We have selected a specific set of Instrument Response Functions (IRFs) based on optimized background cuts derived from Monte Carlo simulations assuming 50 hours of observations and the appropriate zenith angles, as described in [2]. To simulate GPS observations, we utilize the most recent Instrument Response Functions (IRF) provided by the CTA Observatory[3], dubbed `prod5-v0.1`[4]. The IRFs correspond to the so-called "alpha" configuration of the CTA, consisting of 4 Large- and 9 Middle-Sized Telescopes (STs) in the Northern site and 14 Middle- and 37 Small-STs in the Southern site. Simulations and analyses were performed using the publicly available gamma-ray analysis software `ctools`[5].

We model the spatial component as an extended Gaussian source, with the extensions varying between 0.03° and 1.0° following a log-uniform distribution. To model the source spectra, we have used information from the GammaCat[6] catalog consisting of 162 TeV sources within the Gammapy [6] library to obtain the distribution of the parameters by selecting the sources with well-defined spectral models (75 sources) and the corresponding distributions are shown in Fig. 1. We assume a simple power-law spectral distribution $\frac{dN}{dE} = K_0 \left(\frac{E}{E_0}\right)^{-\gamma}$ and for the pivot energy (E_0), we set a fixed value of 1 TeV. For the spectral index (γ) we chose a normal distribution with the mean and the standard deviation (SD) of (2.2, 0.1), replicating the distribution of most of those TeV sources. We model the flux normalization (K_0) as 10^p TeV^{-1}cm^{-2}s^{-1} with the exponents p following a normal distribution with the mean and the SD of (-12, 0.8). For simulating the γ-ray sources using `ctools`, we choose a sky region close to the galactic plane, specifically $0° < l < 20°$ and $|b| < 3°$. The simulation of source observations is performed for energies ranging from 30 GeV to 100 TeV, binned in 60 logarithmically-spaced energy bins, and using a $0.02° \times 0.02°$ spatial binning over a $1.4° \times 1.4°$ region of interest (RoI). Before performing the analysis of classifying these sources based on their extension, we group the energy bins into four wider energy bands covering energies in the range of [0.03, 0.1] TeV, [0.1, 1.0] TeV, [1.0, 10.0] TeV, and [10.0, 100.0] TeV.

2.2 Image Data Pre-Processing

The obtained fits files contain an array of photon counts from 30 GeV to 100 TeV in 4 energy bands with dimensions $(4, 70, 70)$, with the last two dimensions corresponding to the number of spatial pixels over a selected RoI. The photon counts in each of the fits files represent the expected

[3] See http://www.cta-observatory.org/science/cta-performance/ for more details.
[4] The IRF files are publicly available at [5].
[5] http://cta.irap.omp.eu/ctools/
[6] https://gamma-cat.readthedocs.io/

PoS(ICRC2023)599

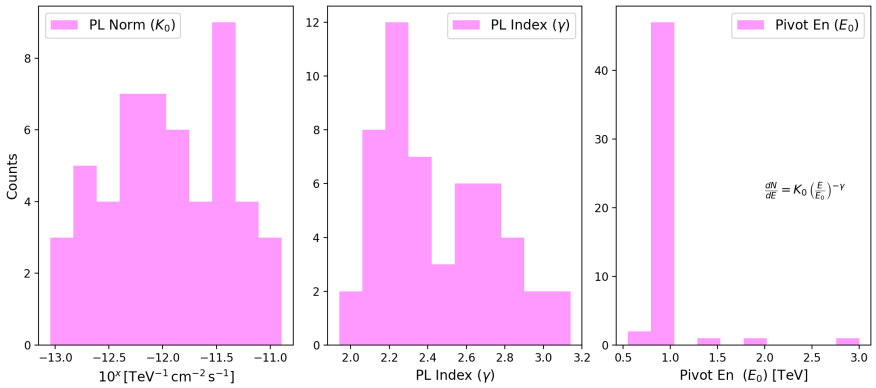


Figure 1: Distribution of parameters of the TeV gamma-ray sources with well-defined spectral models taken from GammaCat.

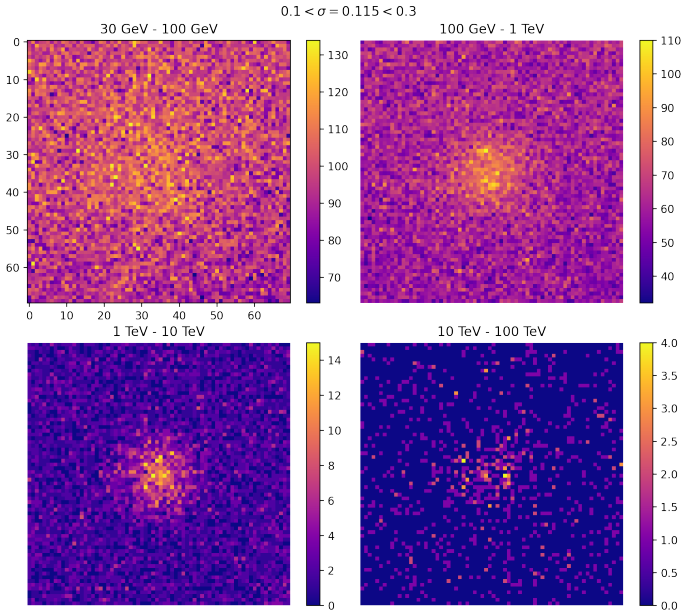


Figure 2: Example Poisson images generated using ctools at 4 different energy bands including the CR background and a source. All images are of fixed height and width with dimensions $(70, 70)$.

counts when one would average over infinitely many Poisson realizations. To account for the Poisson statistics of the real data, we *a-posteriori* apply Poisson noise to each pixel in the arrays. An example image of a γ-ray source including only instrumental background and the added Poisson noise is shown in 4 different energy bands in Fig. 2.

For segregating the source classes based on their corresponding extensions (σ), we divide our dataset into 3 classes. For point-like sources (from now on 'C0') we choose $0.03° < \sigma < 0.1°$, and for extended sources (from now on 'C2') we choose $\sigma > 0.3$. The remaining sources lay in the range between $0.1° < \sigma < 0.3°$ (from now on 'C1'). Two factors influenced this range selection: i) for both, CTA-North and CTA-South arrays, the angular resolution[3] reaches below $0.05°$ at the highest energies (> 10 TeV) whereas the resolution at lower energies (~ 100 GeV) is $\sim 0.2°$, ii)

PoS(ICRC2023)599

for our generated dataset, this selection also naturally leads to a balanced dataset in terms of the number of images belonging to each extension class 'C0', 'C1', 'C2'.

3. Neural-Net Architecture & Training

Convolutional Neural Networks (CNN) have been shown to perform well in classification tasks based on source morphology like star-galaxy classification [7, 8]. Here we propose a multi-input network (Fig. 3) for classifying the source population based on their extension. As shown in Fig. 2, the Poisson images of a source in 4 different energy bands may appear very different and the main reasons for this are, i) the CR background varies significantly from low energy regions ($\sim$ 30GeV) to the highest energies ($\sim$ 100TeV), and ii) the Point-Spread-Function (PSF) of CTA[7] decreases with increasing energy.

To capture the information encoded at different energy bands, our network independently analyzes the features from different bands. This information is then concatenated to pass through convolutional and dense layers to the final softmax layer with 3 outputs, where the network predicts the most probable extension (Fig. 3). The number of parameters in the network is just under 282k. The complete dataset consists of 1000 images of various extensions which are then split into 65% training images, 20% validation images, and 15% test images. During training time, we add augmentation by flipping the image arrays from right to left, up to down, and rotating by 90°. Although these new data points generated via augmentation are not i.i.d. (independent and identically distributed), data augmentation stabilizes the training and has been shown to carry implicit regularization that prevents overfitting [9, 10].

For training the network we used Adam Optimizer [11], where the hyperparameters are set to default values (in TensorFlow) with the learning rate set to 10^{-4}. We check several strategies to achieve better performance, prevent overfitting and achieve stable training and validation loss. To prevent overfitting, we implement early stopping, in particular, that the network stops further training once the validation loss does not improve after 20 epochs. For achieving stable training and validation loss, we navigate the loss landscape carefully by reducing the learning rate by a factor of 0.8 if the learning stagnates, i.e. the validation loss doesn't improve after 10 epochs. The complete implementation is done using TensorFlow[8][12] library. The training time, including the data reading time (consisting of $\sim$ 850 images) and augmentation, takes around 15-20 minutes using an NVIDIA GeForce GTX 1080, with 8GB GPU memory.

4. Results

After training the network, we selected and saved the best model based on the smallest loss on the validation set. We then used this model to evaluate the performance of the network on the test set. Before checking some of the common performance metrics, we visualized the activation maps from the convolutional layer (conv4 in Fig. 3) for four different randomly selected filters. An activation map corresponds to the activation of different parts of the image, which in turn helps us to understand exactly which part of the input image the layer focuses on when looking for features.

[7] CTAO Performance
[8] https://www.tensorflow.org/

PoS(ICRC2023)599

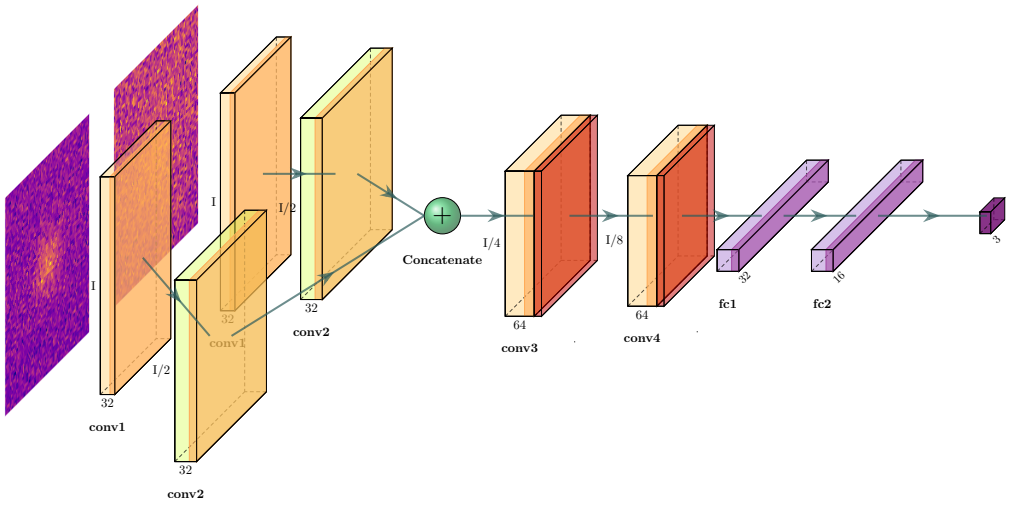


Figure 3: A simplified version of our multi-input network is shown here for only 2 energy bands. The convolutional blocks are shown in yellow (conv1, conv3, conv4), convolution with strides in light green (conv2), pooling blocks are shown in brown, and the fully connected layers are shown in purple. The final layer is the softmax layer.

We show the activation maps for two test images within the 'C0' and 'C1' classes in the top and bottom panels of Fig. 4, respectively. One can clearly see that different filters focus on different regions of the image and the zones of high and low activations are indeed in accordance with the source extensions. This gives us insight into the way the network extracts the spatial features of different source images to perform classification.

One of the most commonly used evaluation metrics for a classification algorithm is the 'Confusion Matrix' (CM) in which each row of the matrix represents the instances in an actual class while each column represents the instances in a predicted class, making it easier to visualize whether the algorithm is confusing between classes. We show the CM for a random selection of test data consisting of 52, 45, and 45 source images for the 'C0', 'C1' & 'C2' classes, respectively. In Fig. 5a, we see a clear separation between the point source class ('C0') and the extended source class ('C2'), with the misclassification between 'C1' and 'C2' sources limited to $< 10\%$.

Complementary to the CM, we employ a very common dimensionality reduction technique, T-distributed Stochastic Neighbour Embedding (TSNE) [13], to embed the high-dimensional data for 2D visualization and understand the network's performance. It converts similarities between data points to joint probabilities over pairs of high-dimensional data and minimizes the Kullback-Leibler divergence between the joint probabilities of the low-dimensional embedding and the high-dimensional data. To implement TSNE within our network, we use the saved model (based on the smallest validation loss) and use the output from the 32-dimensional dense layer (fc1 in Fig. 3). Once fully trained, a deep layer like this dense layer should be enriched with features from different source classes. This high-dimensional data is then transformed into a 2D map via TSNE and the result is shown in Fig. 5b. The TSNE visualization reveals a distinct separation between the 'C0'

PoS(ICRC2023)599

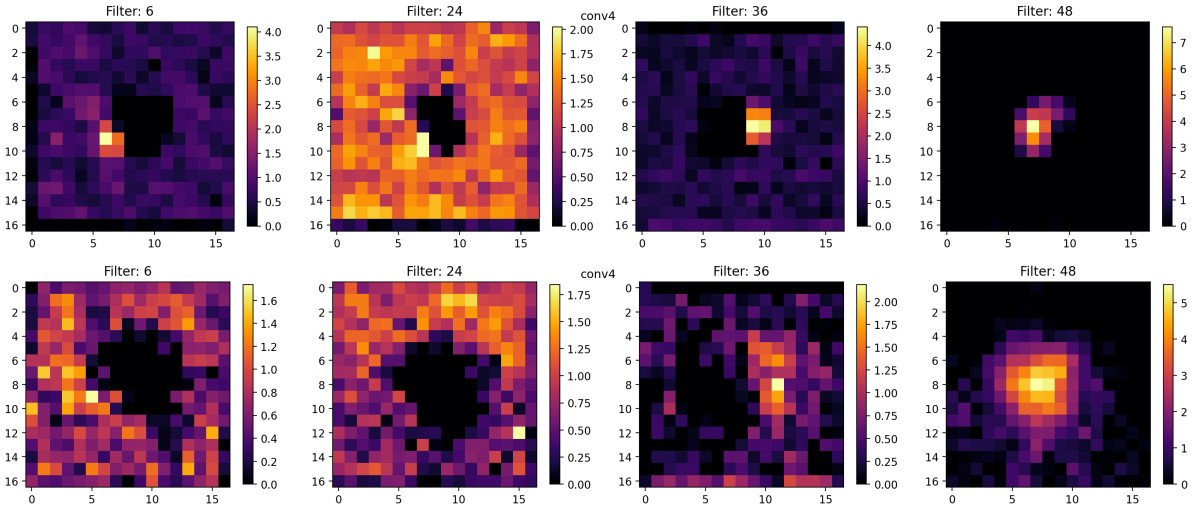


Figure 4: We show the network activations from a convolutional layer (conv4 in fig. 3) for four randomly selected filters for a 'C0' and a 'C1' source in the top and bottom panel respectively. Depending on the extensions, we see similar parts of the images in different filters are activated.

and 'C2' sources, with an overlap between 'C1' and 'C2' sources, complementing the finding that we show in the CM in Fig. 5a.

5. Conclusions

In this work, we have explored the possibility of classifying the VHE sources with various extensions, assuming observations with the full CTA, utilizing a CNN. We use source spectra similar to the spectra of the TeV sources from the GammaCat catalog, the latest CTA IRFs, and publicly available software tools `ctools` to simulate VHE source observations, including the instrumental background. We achieved $\sim$ 97% global accuracy in classifying different source extensions into 3 categories using a simple CNN with a training time of $\sim$ 20 minutes and less than 900 training images.

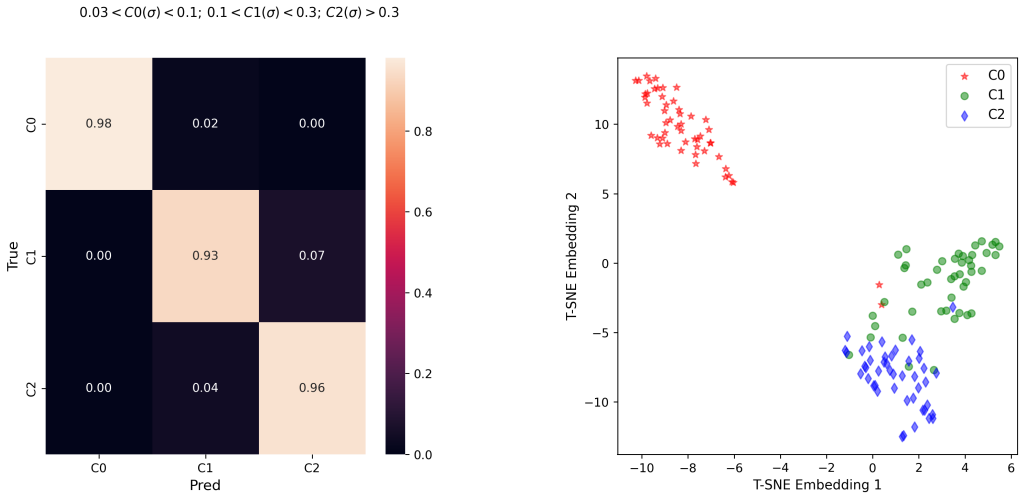
With the upcoming CTA's GPS potentially increasing the number of detected VHE sources by a factor of five, we are building our current method to be a part of a full data-analysis pipeline for VHE sources, including source detection, extension classification, and flux prediction with uncertainties. In this work, we show the prospects of using a deep CNN for classifying a source based on its spatial extension but recognize that the complexity of the task will increase with the inclusion of a simulated full Galactic source population in our analysis, which is the study we intend to pursue in the upcoming works.

References

[1] B. Acharya et al., *Science with the Cherenkov Telescope Array*. World Scientific, feb, 2018.

[2] Q. Remy, L. Tibaldo, F. Acero, M. Fiori, J. Knödlseder, B. Olmi, and P. Sharma, *Survey of the galactic plane with the cherenkov telescope array*, in *Proceedings of 37th ICRC — PoS(ICRC2021)*, Sissa Medialab, jul, 2021.

PoS(ICRC2023)599



(a) CF Matrix; Rows: True labels; Cols: Predicted labels.

(b) 2-D TSNE embeddings from fc1 layer (Fig. 3).

Figure 5: a) Shows the CM for test data points where Point ('C0') and Extended sources ('C2') are well classified with no confusion, while < 10% confusion exists between 'C1' and 'C2' sources. b) Shows the TSNE visualization of the test data points in 2D embeddings from the 32-D dense layer in Fig. 3. A clear separation between 'C0' and 'C2' sources and a modest mixture between 'C1' and 'C2' sources can be seen in the embedded space. This visualization is complementary to the results obtained from the CM.

[3] B. Panes et al., *Identification of point sources in gamma rays using u-shaped convolutional neural networks and a data challenge*, *A&A* **656** (2021) A62.

[4] C. Oetelaar et al., *Localisation and classification of gamma-ray sources using neural networks*, *PoS* **ICRC** (2021) 663.

[5] C. T. A. Observatory and C. T. A. Consortium, "CTAO Instrument Response Functions - prod5 version v0.1." https://doi.org/10.5281/zenodo.5499840, Sept., 2021.

[6] C. Deil et al., *Gammapy - a prototype for the cta science tools*, 2017.

[7] N. Kennamer et al., *ContextNet: Deep learning for star galaxy classification*, in *Proceedings of the 35th ICML*, vol. 80, pp. 2582–2590, 10–15 Jul, 2018.

[8] E. J. Kim and R. J. Brunner, *Star–galaxy classification using deep convolutional neural networks*, *MNRAS* **464** (10, 2016) 4463–4475.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, *Imagenet classification with deep convolutional neural networks*, in *Advances in Neural Information Processing Systems*, vol. 25, 2012.

[10] A. Hernández-García and P. König, *Further advantages of data augmentation on convolutional neural networks*, in *Artificial Neural Networks and Machine Learning – ICANN 2018*, pp. 95–103, 2018.

[11] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, in *ICLR 2015*, 2015.

[12] M. Abadi et al., *TensorFlow: Large-scale machine learning on heterogeneous systems*, 2015.

[13] L. Maaten and G. Hinton, *Visualizing Data using t-SNE*, *JMLR* **9** (2008) 2579–2605.

PoS(ICRC2023)599