Machine learning enhancements for Cherenkov telescope data analysis

A. Di Piano[1,2], A. Bulgarelli[2], D. Beneventano[1], N. Parmiggiani[2], G. Panebianco[2,3], V. Fioretti[2], L. Castaldini[2], R. Falco[2], E. Boella[4], M. Paladino[4], D. Gregori[4] for the CTAO Consortium[5]

[1]*Università di Modena e Reggio Emilia*
[2]*Osservatorio di Astrofisica e Scienza dello Spazio Bologna*
[3]*Università di Bologna*
[4]*E4 computer engeneering SpA*
[5]*www.ctao.org*

ABSTRACT

We developed two applications of Convolutional Neural Network (CNN) based models to enhance the real time scientific analysis of Cherenkov telescopes, in the context of the real time analysis of the Cherenkov Telescope Array Observatory (CTAO). The first model is an auto-encoder trained to denoise counts map by removing the background. The second model is a 2-dimensional regression trained to identify candidate sources in the field of view. Both models achieve results comparable to standard techniques without requiring a priori assumptions on candidate coordinates, background model or instrument response function which are extremely hard to constrain during real time. This work is part of the continuous research and development activity for improvements of future versions of the ACADA/SAG product.

Supported by: UNIMORE Università degli Studi di Modena e Reggio Emilia; E4 Computer Engineering; INAF Istituto Nazionale di Astrofisica

INTRODUCTION

One of the many challenges of real time analysis is the background estimation, especially when searching for unknown candidate sources. We developed two CNN-based deep learning models, that can be used for online inference without a priori knowledge on the background model (Fig. 1). These tools are intended to enhance standard detection techniques when real time circumstances may otherwise hinder their application. We used ctools v2.0 [1] for simulations, gammapy v1.0 [2] for standard analysis using ACADA/SAG v1.0 as reference [3,4]. Finally we used tensorflow v2.11 [5] for deep learning models.

IMAGE DENOISING

We developed a CNN auto-encoder for denoising counts maps, trained using "noisy" and "cleaned" datasets made of raw and background-subtracted normalised counts maps, respectively. Each of the training and validation datasets have 16k and 4k samples in both sets. We use a combination of convolutional 2D layers and average pooling to encode the images, and a combination of convolutional 2D layers and up-sampling for decoding. We use a ReLU activation function in all convolutional layers except the last one, where we use a Sigmoid activation function instead. Finally, we compile the model with the Adam optimiser and the binary cross-entropy loss function.

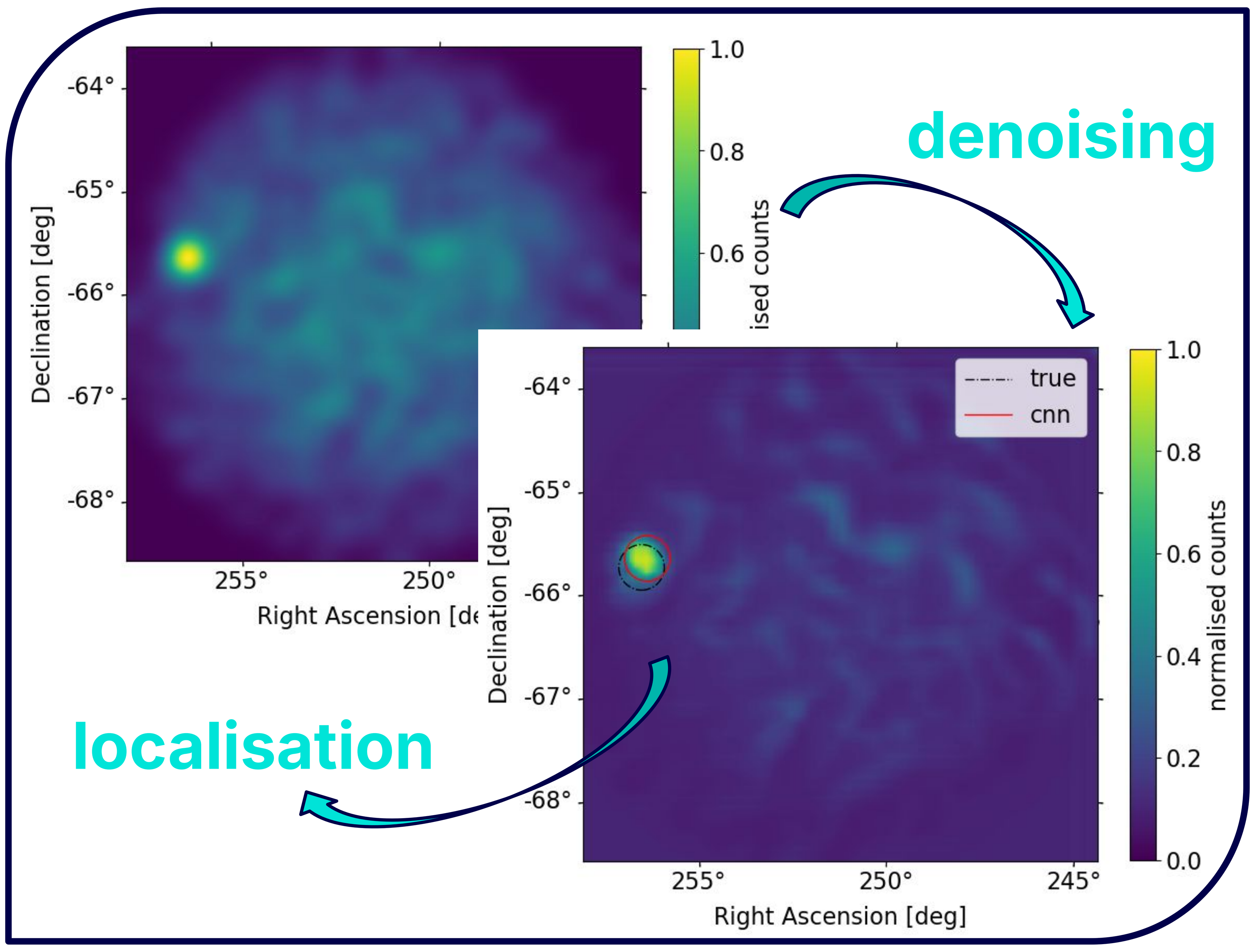

Fig. 1 – Pipeline workflow: original maps are denoised by a CNN auto-encoder that subtracts the background, then a CNN 2d-regressor localises the candidate source.

ACKNOWLEDGEMENTS

We gratefully acknowledge financial support from the agencies and organizations listed here: https://www.ctao.org/for-scientists/library/acknowledgments/

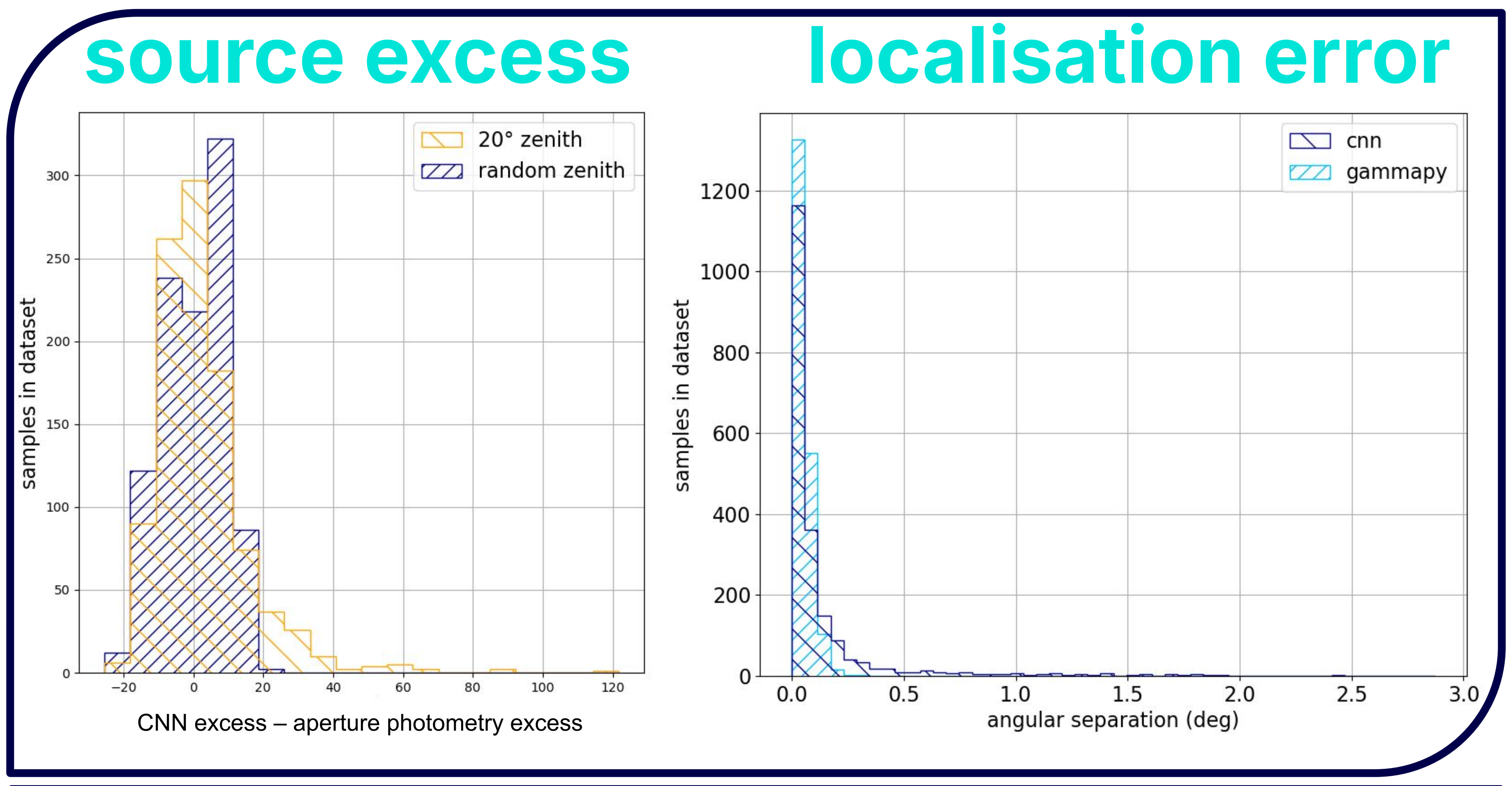

Fig. 2 – (Left) Difference between photometric excess and CNN excess for random and 20° zenith angle datasets. (Right) Localisation error comparison between gammapy's peak-search algorithm and CNN for random zenith angle datasets.

CANDIDATE LOCALISATION

The second model is a CNN 2D-regressor tasked with hotspots localisation. Its inputs are normalised and denoised counts maps with relative normalised labels that identify the simulated target source. The sizes of training and test datasets are 16k and 4k maps each. We use an initial dual layer of 2D convolution and max pooling, followed by 4 consequent 2D convolutions and another max pooling. Then we apply a dropout layer of 20% and flatten our data before a 10k dense layer. Lastly, we apply another 20% dropout layer before the final dense layer. We use a ReLU activation function for all except the final layer, where we use a Sigmoid. We compile the model with the Adam optimiser and a mean absolute error loss function.

RESULTS

We trained both models [6] using randomly selected IRFs of 4LST from the publicly available dataset (prod5-v1.0) [7] and compare with the gammapy analysis for the ACADA/SAG, as part of the continuous research and development for its future improvement.

- We compute difference between the CNN excess and photometric excess (Fig. 2, left). We obtain a mean of $\mu \approx 2$ ($\pm$ 8 at 1σ) counts for the random zenith angle dataset and $\mu \approx 0$ ($\pm$ 13 at 1σ) counts for the 20° zenith angle dataset, where if both methods were to behave perfectly identical we would have $\mu = 0$ and $\sigma \approx 0$.
- We compute the angular separation (ϑ) between the simulated and the found coordinates, comparing the gammapy and CNN methods (Fig. 2, right). On a random zenith angle dataset we obtain a 68% containment radius of about $\vartheta \approx (0.04 \pm 0.004)°$ for gammapy, and of about $\vartheta \approx (0.07 \pm 0.004)°$ for the CNN, respectively.
- We find that the two methods achieve comparable results, with the CNN having the advantage of not requiring any a priori assumptions on target, background or IRFs.

[1] J. Knoedlseder et al., 2016 [2] A. Donath et al., 2023 [3] I. Oya et al., 2019 [4] A. Bulgarelli et al., 2022,
[5] https://www.tensorflow.org/ [6] https://zenodo.org/records/11086320 [7] https://zenodo.org/records/5499840