Report Project of Computer Vision and Image Processing:

Short-term identification of gamma-ray transients

Falco Riccardo
riccardo.falco2@studio.unibo.it

Department of Computer Science and Engineering
Master degree in Artificial Intelligence

Academic year 2021-2022

Abstract

The Cherenkov Telescope Array is the next generation ground-based observatory for gamma-ray astronomy at very-high energies. Starting from FITS files a filtering method (based on a Support Vector Machine classifier) and a clustering system (able to identify and extract the centroid of the events based on techniques of Image Processing and Computer Vision) are developed. These files represent short-time gamma-ray transients (from seconds to hundreds of seconds). Data had been auto-generated by a simulation procedure named ctools, a software package developed for the scientific analysis of Cherenkov telescope data. Under certain assumptions, which refer to the specific parameters of the instrument, good results have been reached by performing a simple classification model and a distinct extraction algorithm.

1. Introduction

Cherenkov Telescope Array [2] (CTA) is an initiative to build the next generation of ground-based gamma-ray astronomy made by dozens of *Imaging Atmospheric Cherenkov Telescope* (IACT) that provide an unprecedented sensitivity to detect transient events.

The project task concerns of an identification of short-time gamma-ray transients (from seconds to hundred seconds), analysing FITS file auto-generated by the simulation procedure given in the software package *ctools*[1]. ctools comprises a set of ftools-like binary executables and Python scripts with a command-line interface that allows the user to interact and perform step-by-step data analysis.

A typical FITS file generated by the simulation procedure is the one plotted in Fig. 1. Here indeed it is shown a pixelated and smoothed AGILE image of the sky captured by the cta instrument. A small blob can be seen the center of the image. It represents the gamma-ray event that must be detected.

[1] http://cta.irap.omp.eu/ctools/

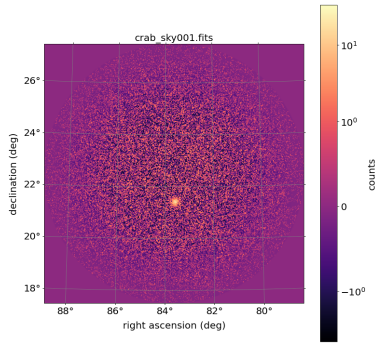


Figure 1: Example of data image. The image shows that the background surrounds the whole image and the small blob is the gamma-ray flare that should be identified.

The image is overlapped and estimated with a background to simulate instrument behavior.

In order to detect the simulated event, it must be developed a filtering and clustering system, capable of identifying and extracting the centroid of the event. It is possible to work with both real and simulated gamma-ray sky maps but for the seek of simplicity working with auto-generated data has been preferred (i.e working with ctools in python).

An appropriate preprocessing is applied for what concerns the filtering (labeling the image) and the extraction of the source coordinates to improve as much as possible the performance of the model.

2. Simulation procedure

As said before, data have been generated using the library *ctools*, a software package developed for the scientific analysis of CTA data or any other Imaging Air Cherenkov Telescope.

The two main procedures used in this project, from the set of ctools, are:

- `ctobssim`[2], which simulates event list(s) querying a pointing direction, the radius of the simulation region, a time interval, an energy interval, an instrumental response function, and an input model (an XML model), which define the property (with also the location) of the gamma source(s),
- and `ctskymap`[3], a tool that creates a sky map from either a single event list or event lists provided in an observation definition file. It will loop over all event lists that are provided and fill all events into a single sky map. Only events within an energy interval spanned by in a fixed energy range interval are considered. Optionally, the tool will subtract a background model from the sky map. The background subtraction method can be selected using the `bkgsubtract` parameter. By default, no background model is subtracted (method `NONE`). Moreover, if `IRF` is selected, the background template that are shipped with the Instrument Response Functions will be used for background subtraction. For the scope of this project no background subtraction has been set.

The XML model contains information about the event list(s) which should be provided to the first tool cited before. Every source is identified in the XML model, by a `source` child of type `PointSource`, whose sub-child, `spatialModel`, contain the coordinates of the gamma source in equatorial coordinates system, namely `RA` (right ascension) and `DEC` (declination). Moreover, the XML contains also a `source` child of type `CTABackgroundModel` which represents instrumental response function of the CTA telescope, function which should simulate the background produced by the instrument during the observation sessions. An example of the XML model for observation simulations is the following:

[2] http://cta.irap.omp.eu/ctools/users/reference_manual/ctobssim.html
[3] http://cta.irap.omp.eu/ctools/users/reference_manual/ctskymap.html

```xml
<source_library title="source library">
  <source name="Crab" type="PointSource">
    <spectrum type="PowerLaw">
      <parameter name="Prefactor" scale="1e-16" value="1" min="1e-07" max="1000.0" free="1"/>
      <parameter name="Index" scale="-1" value="2.48" min="0.0" max="+5.0" free="1"/>
      <parameter name="PivotEnergy" scale="1e6" value="0.3" min="0.01" max="1000.0" free="0"/>
    </spectrum>
    <spatialModel type="PointSource">
      <parameter name="RA" scale="1.0" value="84.92359620397251" min="-360" max="360" free="0"/>
      <parameter name="DEC" scale="1.0" value="19.560730091263924" min="-90" max="90" free="0"/>
    </spatialModel>
  </source>
  <source name="CTABackgroundModel" type="CTAIrfBackground" instrument="CTA">
    <spectrum type="PowerLaw">
      <parameter name="Prefactor" scale="1.0" value="1.0" min="1e-3" max="1e+3" free="1"/>
      <parameter name="Index" scale="1.0" value="0.0" min="-5.0" max="+5.0" free="1"/>
      <parameter name="PivotEnergy" scale="1e6" value="1.0" min="0.01" max="1000.0" free="0"/>
    </spectrum>
  </source>
</source_library>
```

Figure 2: Example of XML model used for create event lists observation. This model shows the way to create an observation with a single source in coordinates $(84.92359620397251, 19.560730091263924)$ with the presence of the simulation of the instrument noise computed with the irf `name="CTABackgroundModel" type="CTAIrfBackground" instrument="CTA"`.

3. Dataset Generation

This section describes how data have been generated for this task.

Using *ctools*, a set of FITS files (namely *skymaps*) has been created, composed by a set of 1000 images of which 500 don't contain any source (background only, *bkg_only*) and 500 contain at least one source (background with sources, *bkg_source*). The Fig. 3 shows more precisely how the dataset is composed.

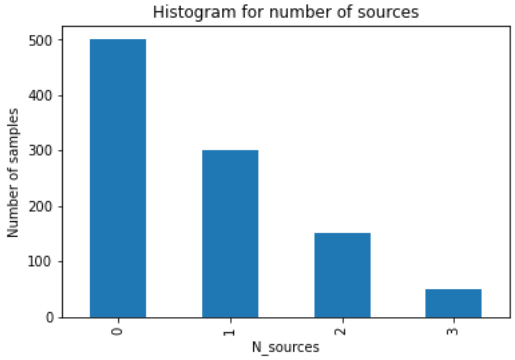


Figure 3: This histogram shows the composition of the dataset used for this task. The bars show the amount of sources event (*N_sources*) which are present in the same skymap. Over a total set of 1000 skymaps, of which 500 are images background only, 300 have 1 source, 150 have 2 sources and 50 have 3 sources.

Here there are the main parameters used for data simulations.

1. *Pointing coords* corresponds to the sky coordinate of where the telescope is pointing for the simulations.
2. *fov* is the Field of view, namely the maximum area of a sample that a camera can image.

3. The *roi* is the Region of Interest which is a region of the sky or in an image on which the scientist wants to focus because of its physics importance.

4. *time* corresponds to the start time and end time for the observation.

5. *energy*: property that all objects in the Universe have, and it can be understood as the capability to perform an activity.

6. Finally the *irf* is the Instrument Response Function (IRF), meaning that it is a group of characteristics that define the behavior of an instrument, such a Cherenkov telescope, and which must be considered during data analysis (one of them is, for example, the collection area of the instrument).

Below, the Table 1 contains the list of hypotheses for which the method set out in this report is effective. Then the dataset was divided into a training set and a test set with the following ratio 0.25.

Table 1: Parameters of the skymap simulations.

Parameter	Value
Pointing coords	(83.6331, 22.5145)
fov	5
roi	5
time	(0,100)
energy	(0.1, 50)
irf	South_z20_0.5h

4. Preprocessing

The original FITS files simulated with ctools are converted into PNG greyscale images. This process allows the data to be expressed in a fixed range of pixel values, namely [0,255]. Moreover, the techniques used for performing image processing seemed to work better in integer-range pixel values. In particular, the library OpenCV[4] has been used to achieve this goal. It is an open-source library that includes several hundreds of computer vision algorithms.

Data are then preprocessed following a double pipeline in order to achieve two different goals: (i) preprocessing for label classification of images (*bkg_only* or *bkg_source*) and (ii) a preprocessing for extracting gamma-ray centroids from the skymap labeled as *bkg_source*.

The first preprocessing (i), starts applying a Gaussian filter with σ equal to 6 and a kernel size of dimension 15. Then a binary image segmentation is used, in particular, a histogram-based segmentation. Due to the uni-modal images property, no automatic segmentation thresholds can be applied (e.g. Otsu's algorithm). For this reason a parametric threshold estimation is provided for this task.

The images have been segmented starting from the maximum value of the its histogram and then a fixed value ϵ has been subtracted in order to compute the threshold T1. Indeed for being recognizable a gamma flare should provide the highest amount of pixel intensity. Under this assumption, it is possible to extract only a set of pixels around the highest intensity value for the pixels of the gamma-ray source.

[4] https://docs.opencv.org/3.4/d1/dfb/intro.html

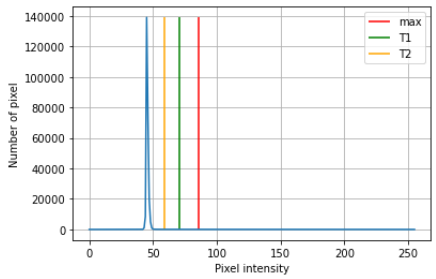


Figure 4: Example of the histogram of an image with 3 sources. The vertical colored lines show the difference between the two thresholds. The uni-modal histogram reflects the distribution of image pixels, where the peak represents the huge amount of noise of the simulated instrument. For this reason, to extract only pixels of the gamma-ray sources, only pixel intensities which fall after the "bell" shape are preferable to be selected.

After the classification model is performed, a new preprocessing step (ii) can be applied, restarting from the original PNG image. The same as before Gaussian filter is applied, and a new different threshold T2 is computed from the pixel intensity histogram, in order to extract all the pixels that belong to the gamma flares, excluding instead the instrument background pixels. The segmentation process is performed considering the threshold as the pixel value at 97% of the pixels intensity histogram for each image.

This strategy improved model performance by combining the advantages of the two approaches. On the one hand, focus only on the more intensity source to simplify the classification task and on the other hand, a more accurate threshold estimation is performed to extract as many as possible sources from the image.

It is this the reason why a double pipeline has been deployed and all the parameters have been selected by computing a Grid Search with Cross Validation with 3 folds. Table 5 shows the steps of them:

1. Label classification pipe:
    - Preprocessing for label classification, which implements threshold T1.
    - Principal Component analysis with 3 principal components.
    - Support Vector Machine for label prediction.
2. Coordinates extractor pipe:
    - Label classification best estimator: step performed after the classification model has been fitted.
    - Preprocessing for coordinates extraction, which implements threshold T2.
    - Finally an prediction algorithm is performed by using blobs moments (a better explanation of this method is presented in Section 5)

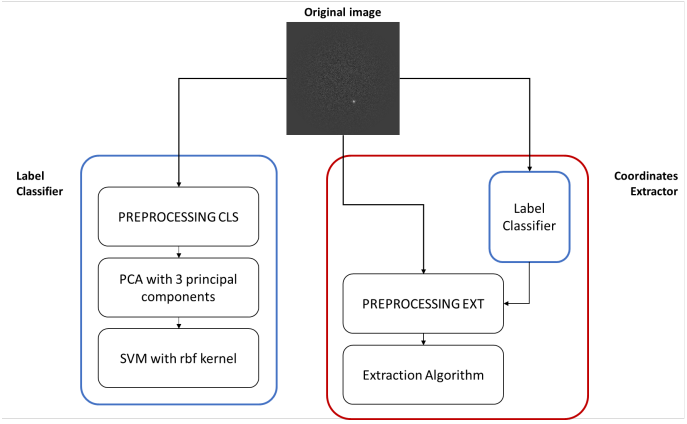


Figure 5: Two pipeline combined with a GridSearchCV for computing the best parameters in order to find the best model.

5. Extraction algorithm

This section describes the extraction algorithm.

After the classification model labels the image as *bkg_source*, second step is performed applying, as said previously, a new preprocessing phase and then the extraction algorithm. The extraction algorithm computes contours and momentums for each blob found from the new preprocessed image, to extract the centroids directly. This is implemented by performing ad-hock instructions from the library OpenCV.

`cv2.findContours` extracts all the contours c in a preprocessed image, while for each contour c centroid's coordinates $\gamma_c = (c_{\bar{x}}, c_{\bar{y}})$ are computed accordingly to the formula to find its barycenter as following, by using the instruction `cv2.moments` in order to computes contours' momentums:

$$\gamma_c = \begin{bmatrix} c_{\bar{x}} \\ c_{\bar{y}} \end{bmatrix} = \begin{bmatrix} \left\lfloor \frac{M_{10}(c)}{M_{00}(c)} \right\rfloor \\ \left\lfloor \frac{M_{01}(c)}{M_{00}(c)} \right\rfloor \end{bmatrix} = \begin{bmatrix} \left\lfloor \frac{\sum_{p \in c} i}{A_c} \right\rfloor \\ \left\lfloor \frac{\sum_{p \in c} j}{A_c} \right\rfloor \end{bmatrix}$$

6. Experimental results

The classification process performed with a simple SVM, which was combined with a Principal Component Analysis (PCA) with 3 principal component, obtained very good results. More details about metrics can be seen in Table 2.

Table 2: Accuracy and f1 score for training set and test set for the classification model, for discrimination between background only image and background with gamma flare.

Metric	Train set	Test set
Accuracy	1.0	1.0
F1 score	1.0	1.0

Fig. 6 shows the confusion matrix of the classification model, which reflects the good results seen also by the metrics in Table 2.

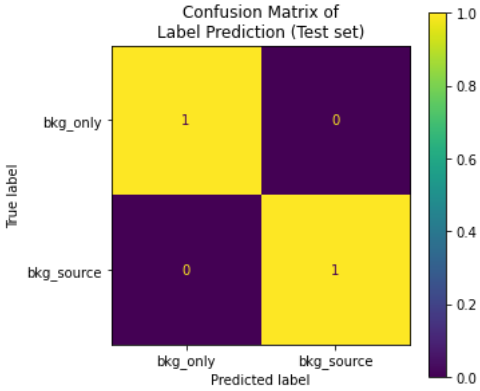
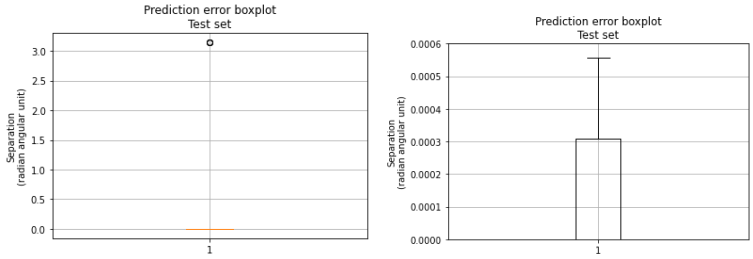


Figure 6: Confusion matrix between predicted and real labels of test set.

In order to estimate the extraction algorithm performances, results have been considered relatively to:

- the prediction error among real and predicted sources, namely the angular error which separate the closer couples between the two classes,
- the number predicted sources respect to the number of real ones.

The boxplot in Fig. 7 visualizes the angular prediction error that has already been defined.



(a) (b)

Figure 7: Prediction error. (a) In the first image a greatest outlier can be seen, which corresponds indeed to undetected sources. Indeed when there is not any match among predictions and real values, the largest value of angular error is considered (i.e. π); (b). Second image focus on the results which reaches the 75th percentile of results (i.e. the Q3). Results indeed show that the 75% of results obtained an angular error ≈ 0.0003.

The boxplot shows the prediction error between the predicted source centroid coordinates and the real ones used to simulate the skymaps, by computing the separation among the two coordinates. It is possible also notice the presence of one outlier. More details about this outliers will be covert later in the report. Table 3 shows the metric scores over the number of centroids found.

Table 3: Angular error among predicted-real centroids coordinates' details. The test set seems to show quite better results.

Metric	Train set	Test set
max	3.1416	3.1416
mean	0.0923	0.0378
dev sd	0.5301	0.3421
min	0.0000	0.0000

Then results can be also discussed in terms of the amount of coordinates well-predicted, namely the number of predicted-real matches. This is important in order to understand the impact of *false positive* (FP) and *false negative* (FN). Table 4 shows the metric scores over the number of centroids found.

Table 4: Accuracy and f1 score for test set about number of centroids extracted from each image for which there exists at least one source.

Metric	Train set	Test set
Accuracy	0.971	0.988
F1 score	0.952	0.976

The confusion matrix represented in Fig. 8 shows more in detail these mistakes. The algorithm is stable in predicting the fact that the image doesn't contain any event, due to the classifier performances mentioned before (see Table 2), and also when it is present only 1 source. While it seems quite acceptable for more then one source in the same skymap.

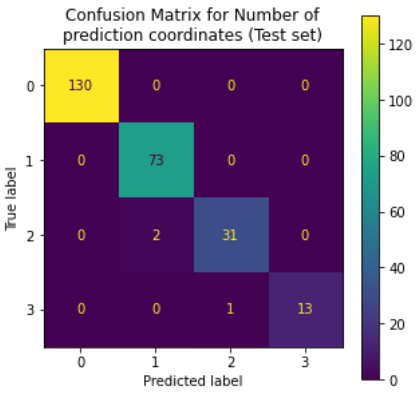
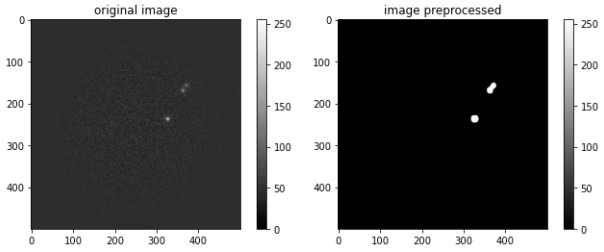


Figure 8: Confusion matrix over the number of predicted and real sources in the same image.

7. Critical issues

Looking at Fig. 7, it is possible to notice a particular value with a large angular prediction error. Indeed some critical issues can be presented when some observation shows two or more gamma-ray sources, one too much closer to the others. They will turn into a single blob which will have one single and different centroid with specific wrong coordinates.



(a) (b)

Figure 9: Original PNG image (a) Preprocessed PNG image (b). Here you can see how if two gamma-ray are one too close to the each other inside the image, they will overlap in the preprocessed image due to the Gaussian filter for noise suppression.

This problem is due to the application of the Gaussian filter σ value selected by the Grid Search Cross Validation mentioned before in 4, which tends to blur the image and the image content deals with larger size structure, that in this case overlapped.

The result of the prediction can be seen indeed in the Fig. 10, background free, which shows in correspondence of the blue cross the real coordinates of the centroid's sources and with the color red the predicted ones. The image shows that in bottom right side of the image one red cross is exactly in the middle between the blue ones, showing the fact I explained above.

8. Conclusions

During this project I explored the task of classification and extraction of centroid's sources in skymap by exploring simple techniques of machine learning, image processing and computer vision.

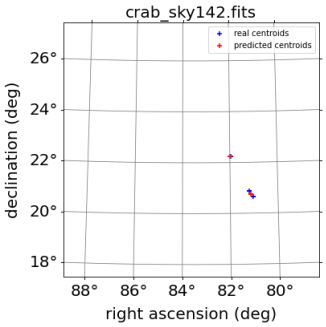


Figure 10: Centroids coordinates prediction plot.

Under the assumptions previously defined in Table 1, it is possible to reach good results by following the approach designed in this project, keeping in mind that, as already presented in Section 7, if two events appear too close in the same skymap, this will result in a wrong prediction.

Future works in this task can follow the implementation of a more complex deep neural network (e.g. U-net [1]) in a way to build a more general model, capable of avoid this issue or work in a more general case of this task. Another relevant improvement can follow the issue raised from the use of the automatic thresholding method of Otsu's algorithm. A more accurate and automatic thresholding method can be developed for this task, in order to make it useful also for more generalised version of this task (i.e. instances of the problem which not follow the assumption in the Table 1).

References

Boris Panes, Christopher Eckner, Luc Hendriks, Sascha Caron, Klaas Dijkstra, Guðlaugur Jóhannesson, Roberto Ruiz de Austri, and Gabrijela Zaharijas. Identification of point sources in gamma rays using u-shaped convolutional neural networks and a data challenge. *arXiv preprint arXiv:2103.11068*, 2021.

RM Wagner, EJ Lindfors, A Sillanpää, and S Wagner. The cta observatory. *arXiv preprint arXiv:0912.3742*, 2009.