Impact of filtering on CYGNO’s Experiment

Igor Fonseca Pains

Advisors: Rafael Antunes Nóbrega
Igor Abritta Costa

September 27, 2022
Summary

- Image filtering on CYGNO experiment
  - Motivation and goals
  - Proposal
  - Results
  - Conclusions
  - Next steps
Motivation and goals

● Motivations
  ○ Is it possible to improve SNR of CYGNO using filtering techniques?
  ○ What is their impact on detection efficiency?
  ○ What is their impact on energy estimation?

● Goals
  ○ Propose different pre-processing techniques.
  ○ Define a methodology to assess their performance.
  ○ Evaluate their impact on simulation and real data for the LIME detector.
Image Filtering

Collaboration vs Proposal

Preprocessing

(1) Pedestal subtraction
(2) Noise thresholding
(3) Filtering
(4) Rescale
(5) Noise Reductor

Subtraction of average component of noise

Input Image (2304x2304)

Preprocessing

(1) Pedestal subtraction
(2) Filtering
(3) Thresholding
(4) Rescale

Suppression of pixels below a threshold

Median Filter
Average Filter
Gaussian Filter
U-net

Suppression of pixels below std component

Pedestal subtraction
Noise thresholding
Filtering
Rescale

Image reduction to 576x576 pixels

Clustering

27/09/2022
Evaluation proposal

Filtering parameter selection flowchart

Real Image simulation

(1) Simulation Image

(2) Noise generator

\[ I(x, y) \rightarrow I(x, y) + \eta(x, y) \]

Insert noise to simulation image

Preprocessing

(1) Pedestal subtraction

(2) Filtering

(3) Thresholding

Pixel classifier

Performs pixel classification for each filter parameter and threshold

Evaluation

\[ (X, Y) > t \]

\[ (X, Y) > 0 \]

Compare to truth

27/09/2022
Image Simulation

6 keV signal simulation

Noise simulation based on the ECDF of a noise run

Simulated image with signal and noise background
Two approaches will be used:

- **Spatial filtering:**
  - Linear (Average and Gaussian filters);
  - Non-linear (Median filter);
    - Parameter is the used mask (window size and filter type).

- **Deep learning based:**
  - U-Net;
    - Parameters are obtained after training process, by optimizing the layer’s weights;
Proposed reconstruction algorithm

Evaluation

- Threshold is applied to define if a pixel is signal or background.
- The result is then compared to the truth image to access the precision, recall and f1 score through the confusion matrix.

<table>
<thead>
<tr>
<th>Predicted</th>
<th>Truth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Negativo</td>
<td></td>
</tr>
<tr>
<td>True Negative (TN)</td>
<td></td>
</tr>
<tr>
<td>False Positive (FN)</td>
<td></td>
</tr>
<tr>
<td>Negative</td>
<td></td>
</tr>
<tr>
<td>Positive</td>
<td></td>
</tr>
<tr>
<td>False Negative (FN)</td>
<td></td>
</tr>
<tr>
<td>True Positive (TP)</td>
<td></td>
</tr>
</tbody>
</table>

\[
\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}
\]

\[
\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}
\]

\[
F1 = 2 \cdot \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}
\]
The optimization of the filter parameters (window size and threshold) was done through a scan aiming the maximum F1 score on the training dataset (200 images).

- Average Filter: Completed
- Gaussian Filter: Completed
- Median Filter: Completed
- U-net: Using the network trained with LEMON data. (For now)
The range of possible thresholds was chosen by looking at the ADC counts distribution of signal pixels after filtering.
Image Filtering

F1 score evaluation - Average Filter

The ideal window size seems to be around 13 whereas the threshold 2.

27/09/2022
Optimization

- The following parameters were chosen for each filter ([Window size, threshold]):
  - Average Filter: [13, 2]
  - Gaussian Filter: [13, 2]
  - Median Filter: [15, 2]
  - U-net: Using the network trained with LEMON data. (For now)

- These parameters were used on a modified version of the reconstruction algorithm in order to create reco files from the test dataset (100 images) for comparison.
## Results

<table>
<thead>
<tr>
<th>Method</th>
<th>Number of clusters</th>
<th>Precision</th>
<th>Recall</th>
<th>F1 score</th>
<th>Energy estimation accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CYGNO</td>
<td>295</td>
<td>0.8021 ± 0.0051</td>
<td>0.8172 ± 0.0053</td>
<td>0.8088 ± 0.0046</td>
<td>-1.26 ± 0.17</td>
</tr>
<tr>
<td>CYGNO (1.4)</td>
<td>106</td>
<td>0.8344 ± 0.0050</td>
<td>0.7848 ± 0.0052</td>
<td>0.8081 ± 0.0044</td>
<td>-1.92 ± 0.17</td>
</tr>
<tr>
<td>Average Filter</td>
<td>100</td>
<td>0.7991 ± 0.0045</td>
<td>0.8511 ± 0.0055</td>
<td>0.8237 ± 0.0046</td>
<td>-0.97 ± 0.16</td>
</tr>
<tr>
<td>Gaussian Filter</td>
<td>100</td>
<td>0.7991 ± 0.0045</td>
<td>0.8509 ± 0.0056</td>
<td>0.8238 ± 0.0045</td>
<td>-0.92 ± 0.16</td>
</tr>
<tr>
<td>Median Filter</td>
<td>100</td>
<td><strong>0.8619 ± 0.0051</strong></td>
<td>0.7726 ± 0.0048</td>
<td>0.8143 ± 0.0045</td>
<td>-2.42 ± 0.16</td>
</tr>
<tr>
<td>U-net</td>
<td>108</td>
<td>0.8063 ± 0.0045</td>
<td>0.8454 ± 0.0051</td>
<td><strong>0.8250 ± 0.0044</strong></td>
<td>-1.11 ± 0.16</td>
</tr>
</tbody>
</table>
All the fake clusters are located in the very low-energy region, thus a simple energy cut would be enough to remove them.
All the fake clusters are located in the very low-energy region, thus a simple energy cut would be enough to remove them.
Both histograms seem pretty close, showing a trend to lose energy. Also, an increase in sigma means a bigger loss in energy reconstruction.
The same trend can be seen here. However, the median filter has a higher loss in energy reconstruction.
The two filters that resulted in the closest estimation of the signal energy.
Conclusions

- The gaussian and average filters alongside the U-net showed the best results concerning the F1 score.

- The median filter had the best precision, but its smaller recall reduced the F1 score.

- A larger dataset containing more diverse images should be used to justify this proposal.
Next steps

- Use a dataset containing more diverse images (more signals with different energies per image).

- Train the U-net using LIME data.