Trigger proposal

Igor Pains
with Rafael Nóbrega

20/02/2025
Analysis & reconstruction meeting

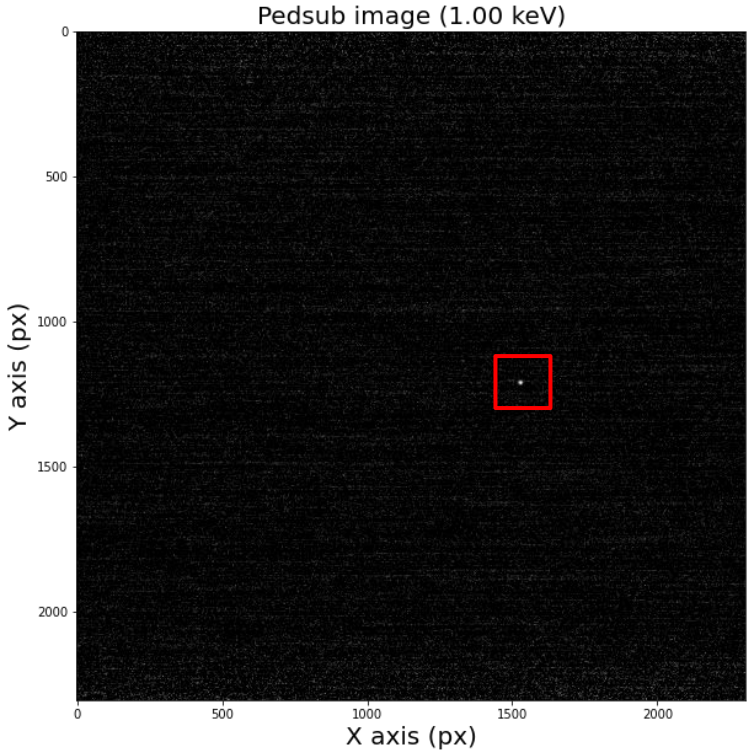
Previous presentation

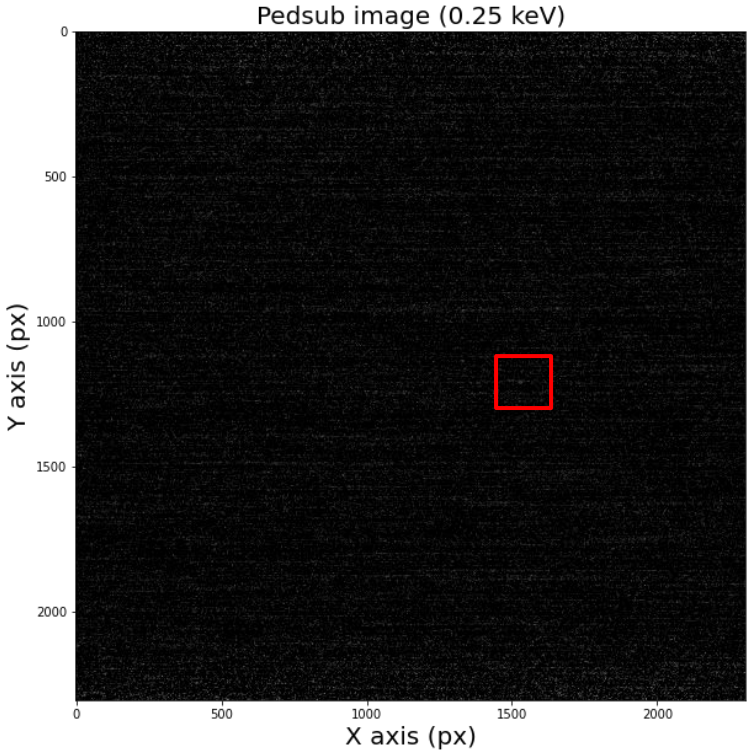
Previous presentation

- A **pixel level trigger** algorithm was proposed:
  - Detect **centroids** through **filtering (high correlation points).**
  - Save **pixels** in a **square region** around **centroids.**

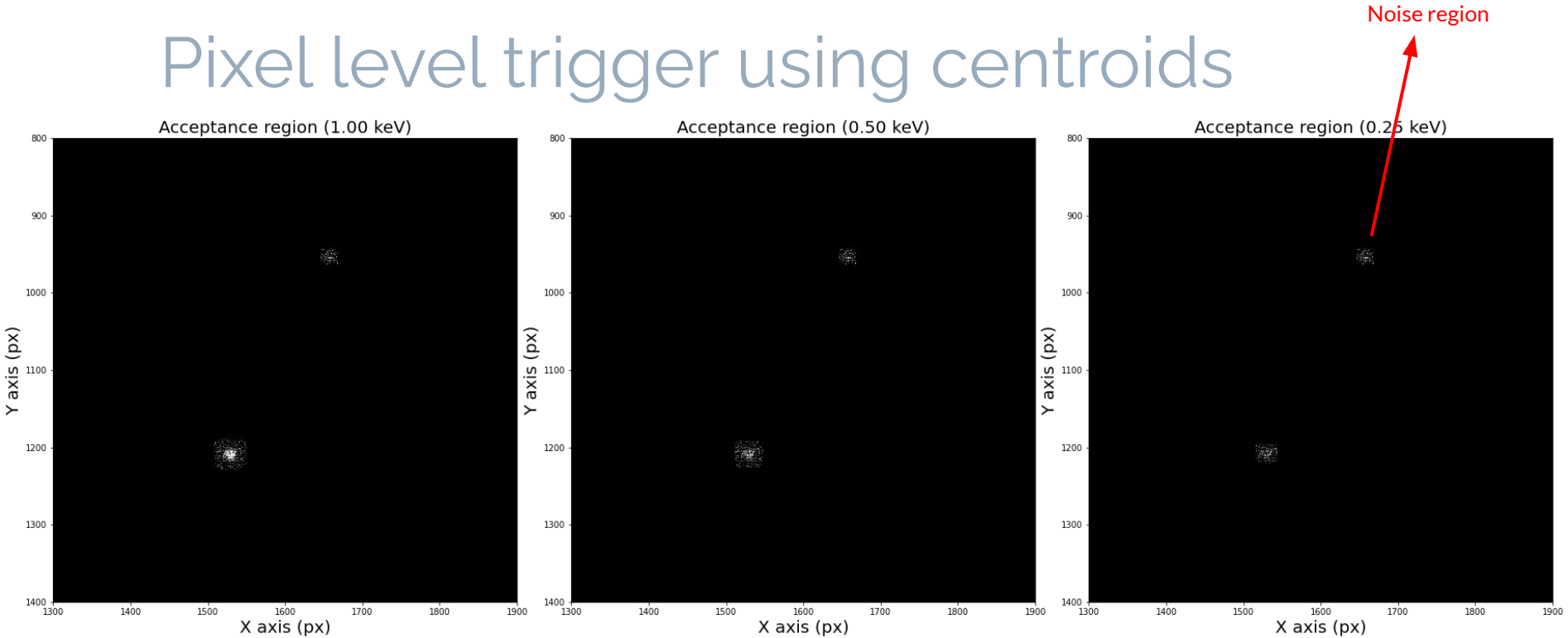
- Tests performed on images showed **promising results.**
  - **Hyperparameters** should be selected: **threshold** for **centroids** and **radius** of the **square region.**

Pixel level trigger using centroids





Pixel level trigger using centroids



- 1 keV: : **360** centroids -> **2060** pixels on the acceptance region **(100% of the signal energy)**.
- 0.5 keV: **195** centroids -> **1675** pixels on the acceptance region **(99.93% of the signal energy)**.
- 0.25 keV: **21** centroids -> **1121** pixels on the acceptance region **(99.15% of the signal energy)**.

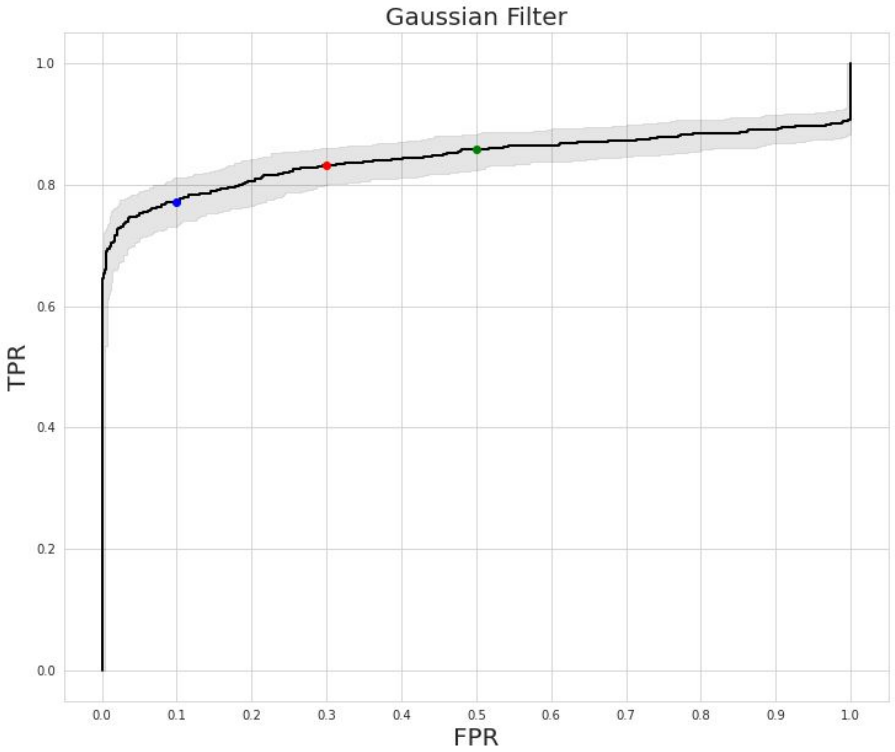
1.
Image trigger

Datasets

- **Datasets**
  - **Training:**
    - Noise dataset: 600 images from pedestal runs (Run 4 underground).
    - ER and NR signal simulation: 600 images each containing 0.25-1 keV signals added to pedestal runs (different from noise dataset).
  - **Validation:**
    - Noise dataset: 200 images from pedestal runs.
    - ER and NR signal simulation: 200 images each containing 0.25-1 keV signals.
  - **Test:**
    - Same configuration as validation.

Training ROC - Filtering

- The **ROC curve** shows the performance of the **image trigger** based on **filtering:** Link of presentation.
  - Choosing a **threshold** based on the **FPR** measure gives an estimate of **noise images discarded** by the **trigger.**
  - Three thresholds were selected to be used on the test dataset: **10%**; **30%**; **50%** of **false alarm.**



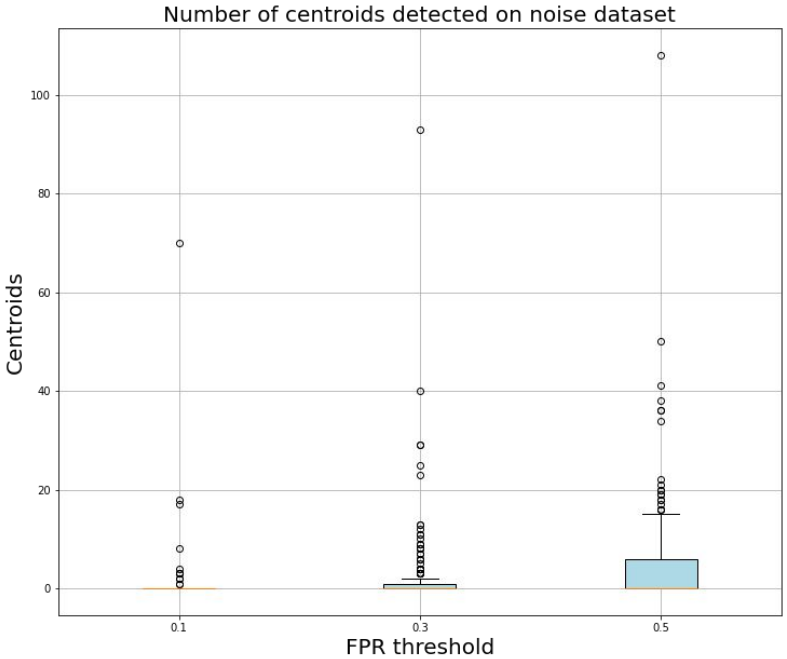
2.

Pixel level trigger

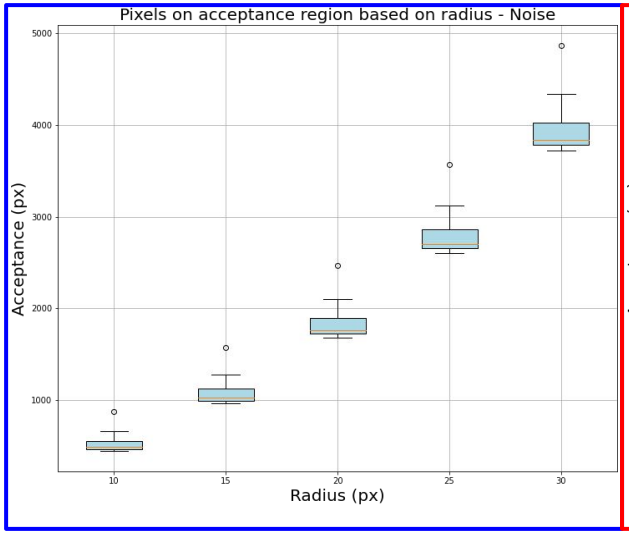
Centroids - Noise

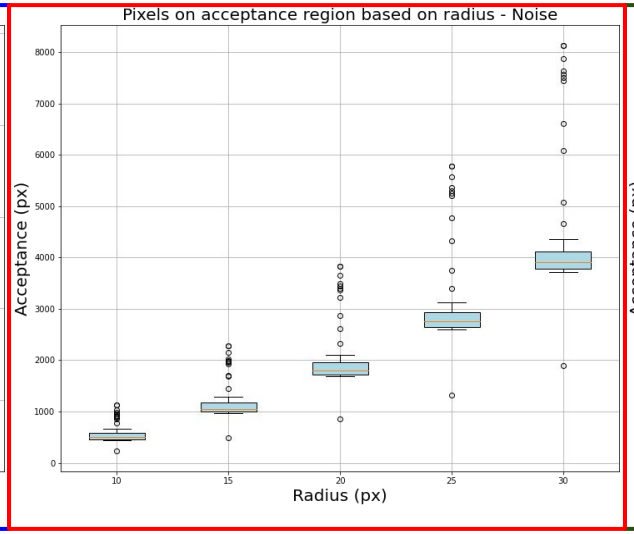
- The thresholds detected at least one centroid on **12 (6%)**, **55 (28%)** and **93 (46%)** images.

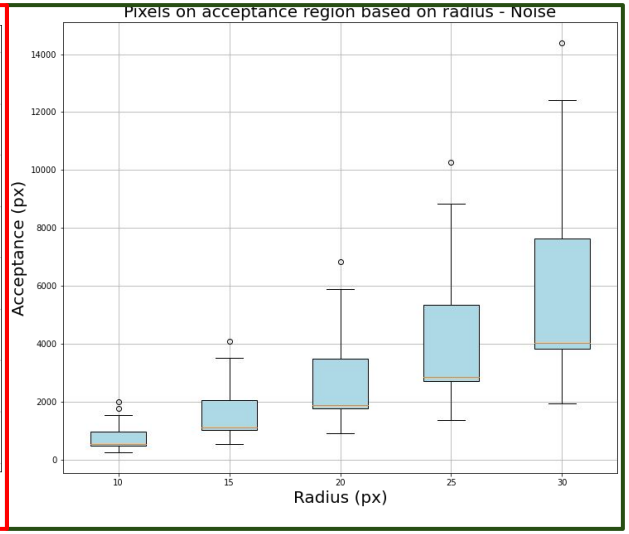
- As expected, a looser threshold (50% fpr) generates more centroids than a stricter threshold (10% fpr).



Pixel level trigger - Noise







- Average pixels on acceptance region (**10%**): 534.33; 1096.00; 1857.67; 2819.33; 3981.00.
- Average pixels on acceptance region (**30%**): 586.80; 1217.64; 2074.91; 3159.45; 4471.27.
- Average pixels on acceptance region (**50%**): 718.00; 1476.37; 2501.16; 3792.27; 5345.98.

Pixel level trigger - Noise

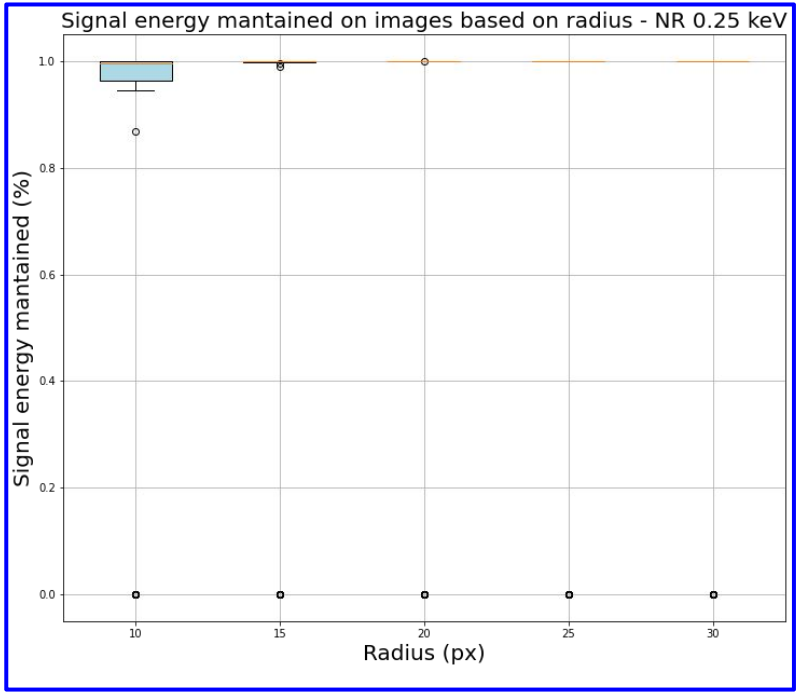
- The **memory** needed to save the **remaining pixels** is proportional to the **number of pixels** and **data type** (i.e int32 needs **4 bytes** per pixel).
  - Estimated memory **(10%)**: ~25; ~51; ~87; ~132; ~187 KB.
  - Estimated memory **(30%)**: ~126; ~261; ~445; ~678; ~960 KB.
  - Estimated memory **(50%)**: ~261; ~536; ~908; ~1378; ~1942 KB.
  - Full dataset memory: **~963 MB** (compressed ROOT).

- In practice, more memory is required because the spatial information of the image is also important.
  - On the other hand, compression methods like ROOT can save images with zero-intensity values in irrelevant positions (noise) using much less memory than if they were fully filled.

Pixel level trigger - 10% fpr th

- The **stricter threshold** can detect at least a centroid on:
  - **0.25 keV - 155 images (77%).**
  - **0.5 keV - 199 images (99%).**

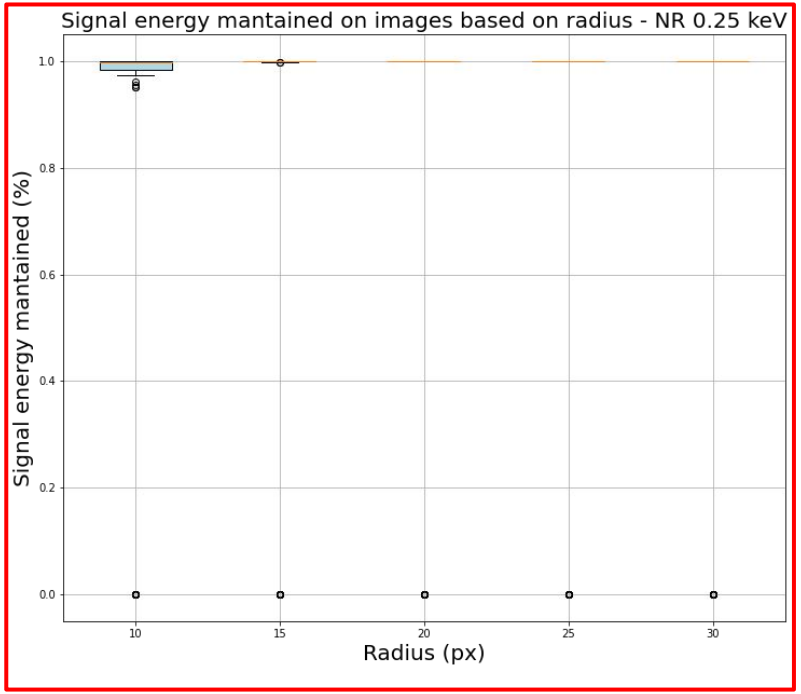
- A radius of 20 pixels is enough to save all the energy from the signals detected.



Pixel level trigger - 30% fpr th

- The **middle threshold** can detect at least a centroid on:
  - **0.25 keV - 163 images (82%).**
  - **0.5 keV - 200 images (100%).**

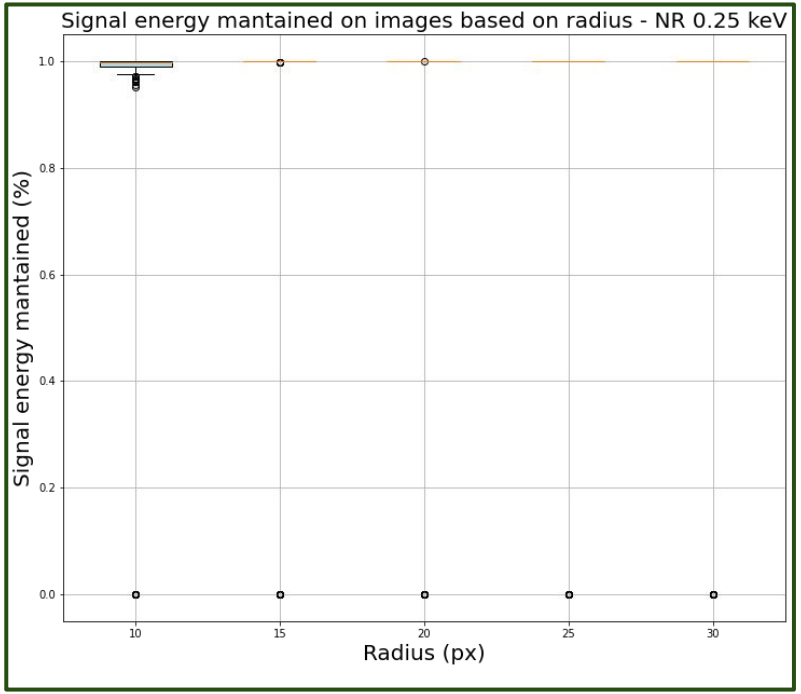
- A radius of 20 pixels is enough to save all the energy from the signals detected.



Pixel level trigger - 50% fpr th

- The **looser threshold** can detect at least a centroid on:
  - **0.25 keV - 169 images (85%).**
  - **0.5 keV - 200 images (100%).**

- A radius of 20 pixels is enough to save all the energy from the signals detected.



Pixel level trigger

- This algorithm can be applied on the datasets to check the actual memory needed to store the images on a ROOT file.

- The radius equal to 20 and each one of the three thresholds were used.

- Comparison with the truth information that had to be stored.

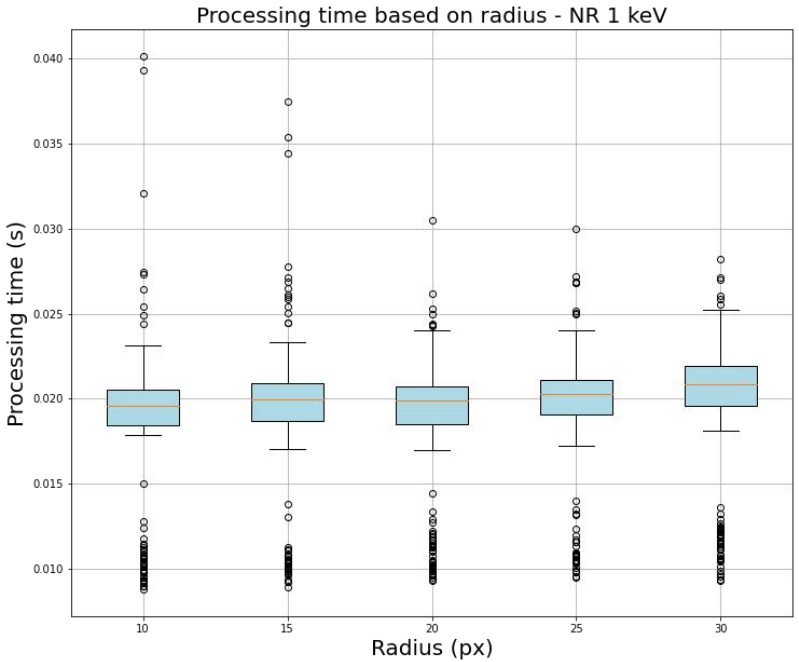
Memory used

	Truth	Full images	10 % fpr	30% fpr	50 % fpr
Noise	0 MB (0 images)	963.3 MB (200 images)	1.1 MB (12 images)	4.9 MB (55 images)	8.3 MB (93 images)
NR 0.25 keV	17.9 MB (200 images)	963.3 MB (200 images)	14.2 MB (159 images)	15.5 MB (174 images)	16.4 MB (184 images)
NR 0.5 keV	17.9 MB (200 images)	963.3 MB (200 images)	17.8 MB (199 images)	17.9 MB (200 images)	17.9 MB (200 images)

- Over **100x reduction factor** on worst case for noise.
- **50x reduction factor** on **0.5 keV NR** maintaining **all the signal information.**
- The **overall reduction factor** depends on the **signal rate** on images.

Processing time

- The processing time for this algorithm depends mostly on the number of centroids present in the image.
- On the worst case (more centroids on images) it needed ~20 ms per image.



3.
Conclusion

Conclusion

- This algorithm could achieve from **100x to 1000x** reduction factor on **memory used** on **noise data.**

- The **threshold** to be used should be selected based on the **energy desired** for **detection** (i.e **0.25 keV** needs a **lower threshold** with **higher false alarm**).

- This algorithm could be used on the **reconstruction** as **preprocessing** (maybe replacing zero suppression/median filter).

Thanks!