Image based trigger algorithms

Igor Pains
with Rafael Nóbrega

Reconstruction & analysis meeting - 23/10/2025

Contents

1. Introduction

Motivation

- One major **challenge** for the **CYGNO experiment** in the long term will be to **store and analyse all the data produced by the detector.**
  - Each run containing **400 images** needs **~1.36 Gb** to be stored (Fusion, compressed .mid).
  - A **single day of acquisition** may produce **~266 Gb** of data (Run5 on 26th september).

- The motivation of this work was to study algorithms capable of **distinguishing images or regions** containing a **signal of interest** and **background events.**

- An algorithm capable of doing this task was called **image based trigger algorithm.**

What was done

- Two approaches were proposed:
  - **Image level trigger** using **filtering** and **CNNs.**
  - **Pixel level trigger** using **filtering.**

- A performance analysis was done using simulated and real data from Fusion (focused on **low energy** events):
  - Trigger **detection performance**.
  - Reconstruction comparison.
  - **Processing time.**

Last results

- The **Gaussian filter** may **detect ~80%** of the **0.25 keV NR** and **ER** events with a **~10% false alarm (~0.25** and **0.02 seconds** on **CPU** and **GPU).**

- The **CNN** may **detect ~80%** of the **0.25 keV NR** and **ER** events with a **~0.5% false alarm (~0.55** and **0.2 seconds** on **CPU** and **GPU).**

- Both methods **outperform** the **reconstruction** in **detecting 0.25 keV** events.

- **All methods** can easily **detect** energies **above 0.5 keV.**

Last results

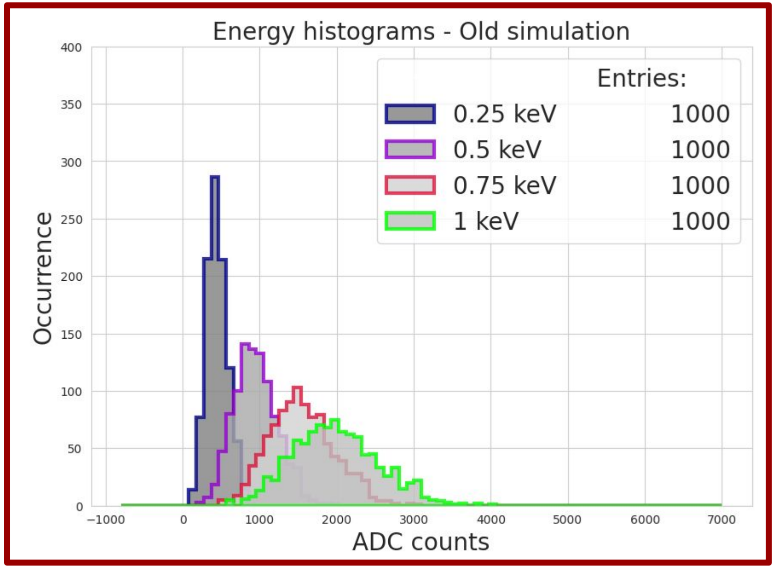
- The **idea** of the **pixel level trigger** is to apply the **Gaussian filter** looking for **high correlation** points and **save** the **region around** them. [1] [2]

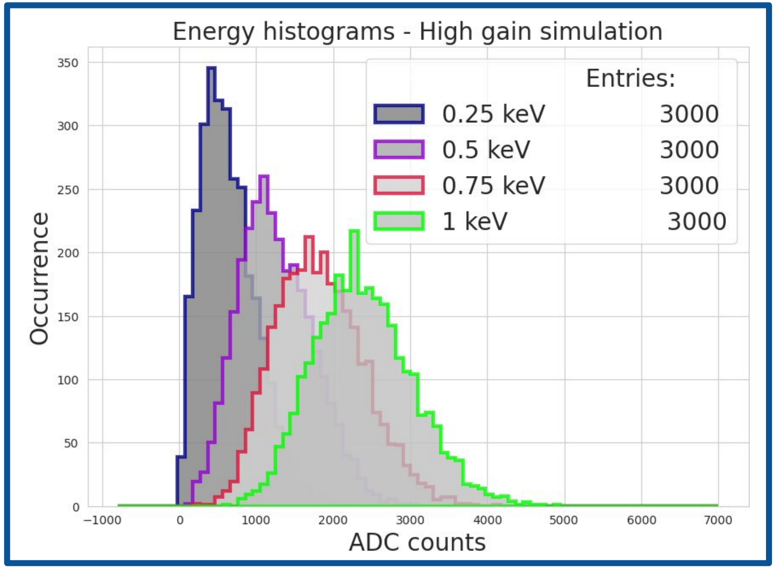
- It **preserved 100%** of **0.5 keV** and **85%** of **0.25 keV** simulated events, while reducing **~50x memory** used to **store** the images (**~20 ms** on **CPU**).

- **These results** were achieved **without** a **proper methodology** to determine an **ideal threshold** to **detect high correlation** points (this will be **explored** with the **new simulation**).

2. Fusion simulations

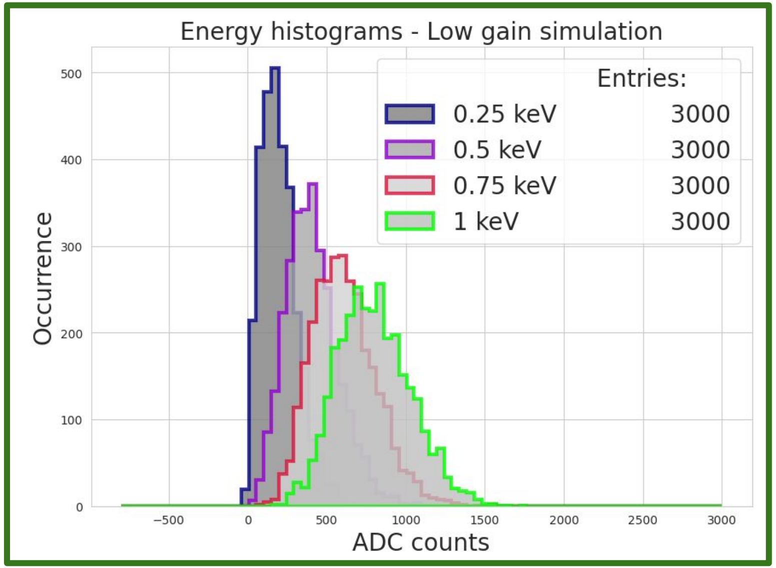
High gain simulation

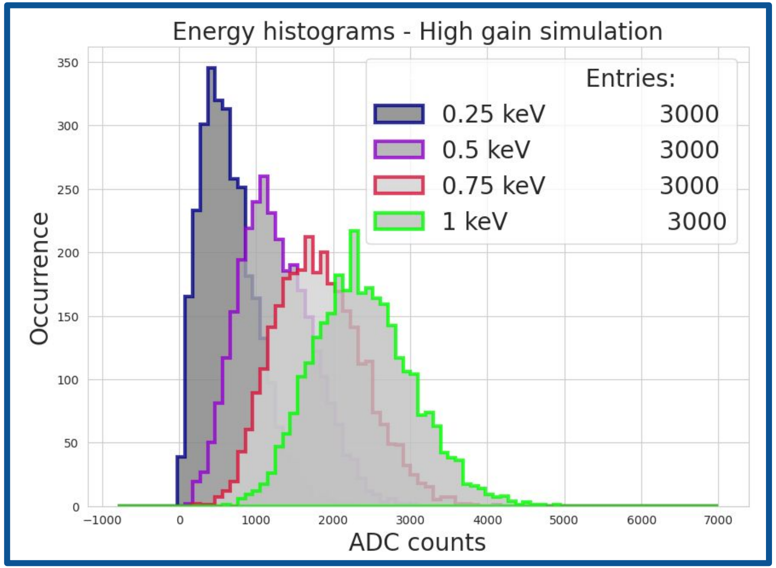




- The new high gain simulation energy in ADC counts is slightly higher and wider.
  - **0.25 keV: [446.17 ± 138.45]** vs **[650.76 ± 367.88]**
  - **0.5 keV: [946.93 ± 278.46]** vs **[1256.11 ± 479.44]**
  - **0.75 keV: [1546.18 ± 418.33]** vs **[1824.22 ± 570.07]**
  - **1 keV: [1992.36 ± 556.59]** vs **[2386.86 ± 639.05]**

Low gain simulation





- The **low gain** simulation energy in **ADC counts** is **~3 times smaller** than **high gain**.
  - **0.25 keV: [200.89 ± 116.91]** vs **[650.76 ± 367.88]**
  - **0.5 keV: [408.99 ± 166.39]** vs **[1256.11 ± 479.44]**
  - **0.75 keV: [609.78 ± 201.59]** vs **[1824.22 ± 570.07]**
  - **1 keV: [801.70 ± 230.01]** vs **[2386.86 ± 639.05]**

- It is **expected** to be **harder** to **detect** these **signals**.

3. Results

Image level trigger

- The **same training procedure** for the **filtering** methods was used with the **new data.**
  - **Gaussian** and **matched filter** still provide the **best results.**
  - **Kernel sizes** of **17 (HG)** and **21 (LG)** were selected.

- The **CNN models trained** with the **old simulation** were used as a **start point.**
  - **New training** will be done in the **future.**

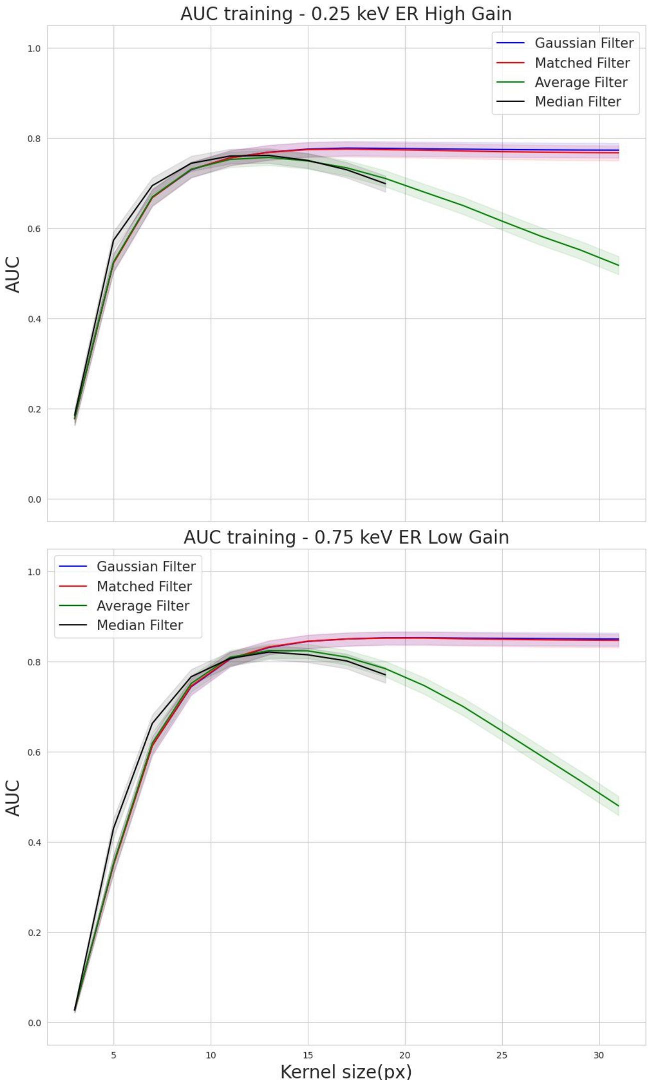
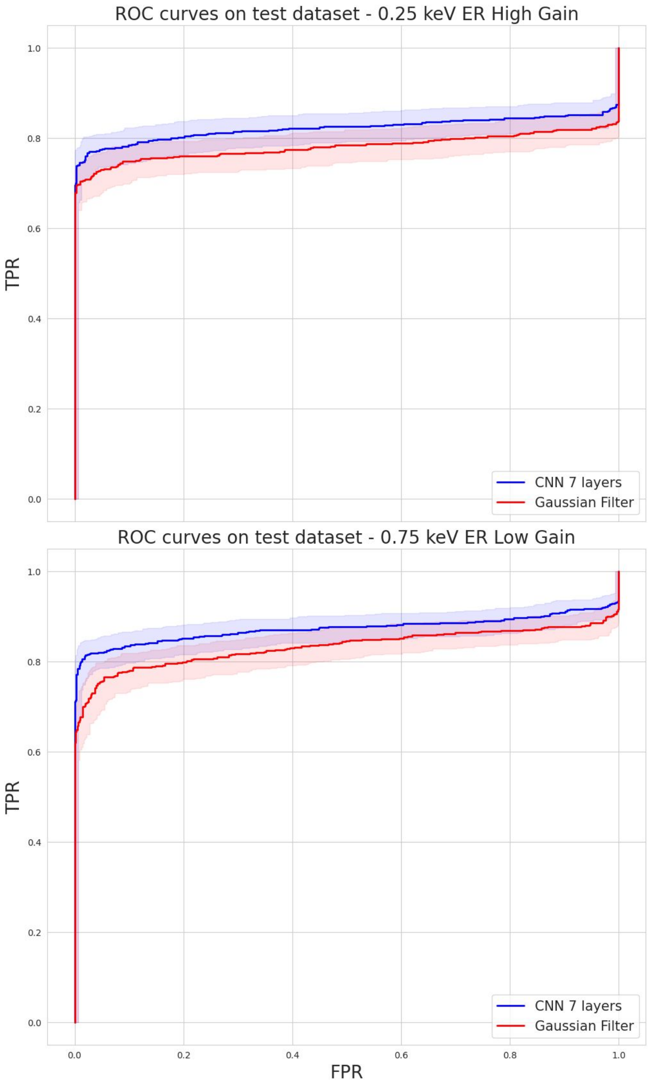


Image level trigger

- The **new high gain simulation** showed **slightly worse results** compared to the **old** one.
    - It has more **low ADC counts events** due to its higher **std**.

- The **low gain 0.75 keV ER** shows **similar** results compared to the **0.25 keV ER** from the old simulation.

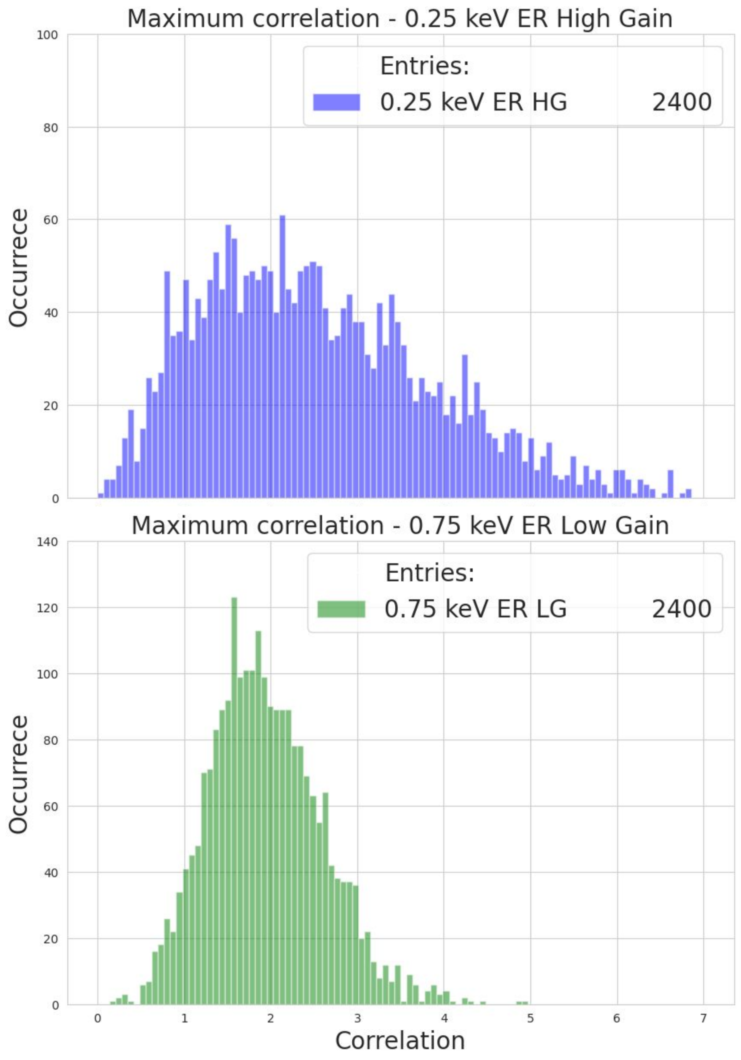
- Further **results** considering **higher energies** will be shown in the **future.**



Pixel level trigger

- The **pixel level trigger** needs **two parameters**: **threshold** (centroid detection) and **radius** (save region).
  - The **threshold** should **detect** as much as possible of the **signal** while **discarding noise.**

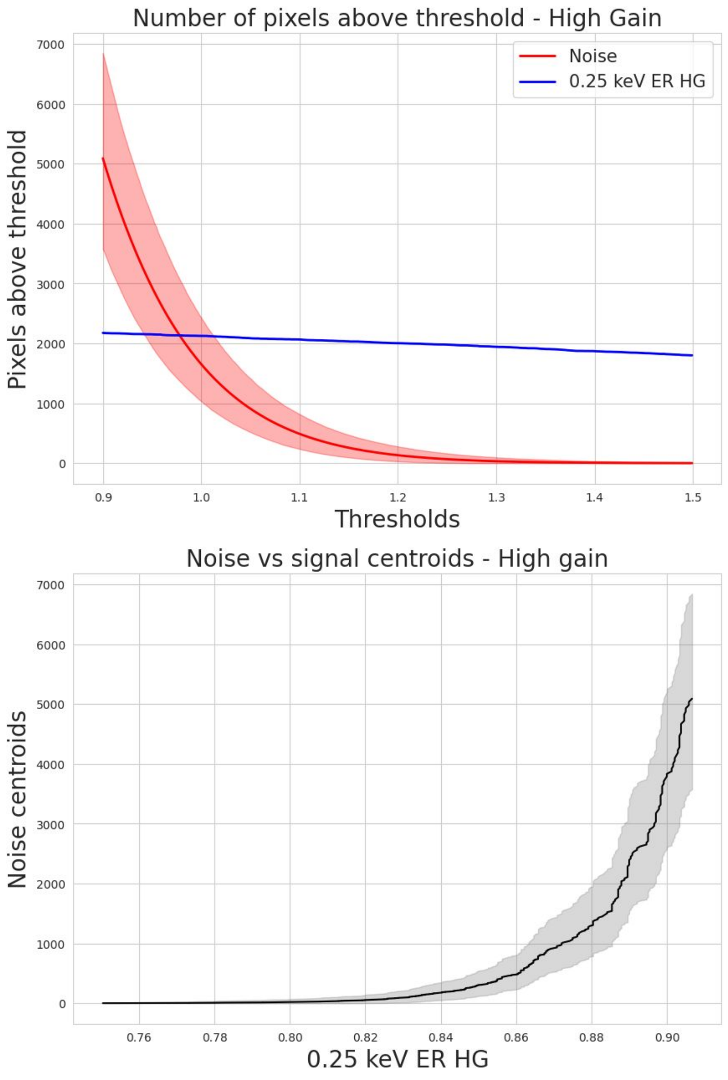
- **Possible ranges** of **thresholds** were selected to see the number of **noise centroids** detected on **noise dataset.**
  - **High gain:** [0.9, 1.5]
  - **Low gain:** [0.8, 1.2]



Pixel level trigger

- The **number** of **noise centroids** detected **start high** and **rapidly decreases** as the **threshold increases**.
    - **5000 centroids** could **retain 2 million pixels** in the image with a **radius** of **20** for example.

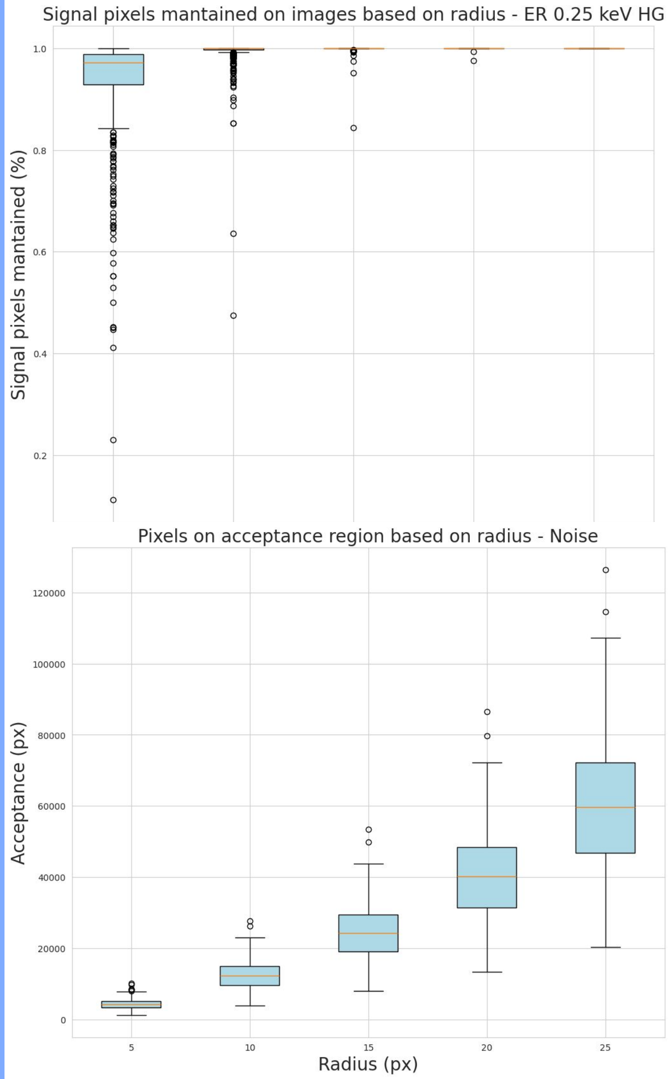
- The **lower plot** gives a better look at the possible i**deal threshold**: the **knee**.
    - **High gain threshold: 1.123**
    - **Low gain threshold: 0.988**



Pixel level trigger HG

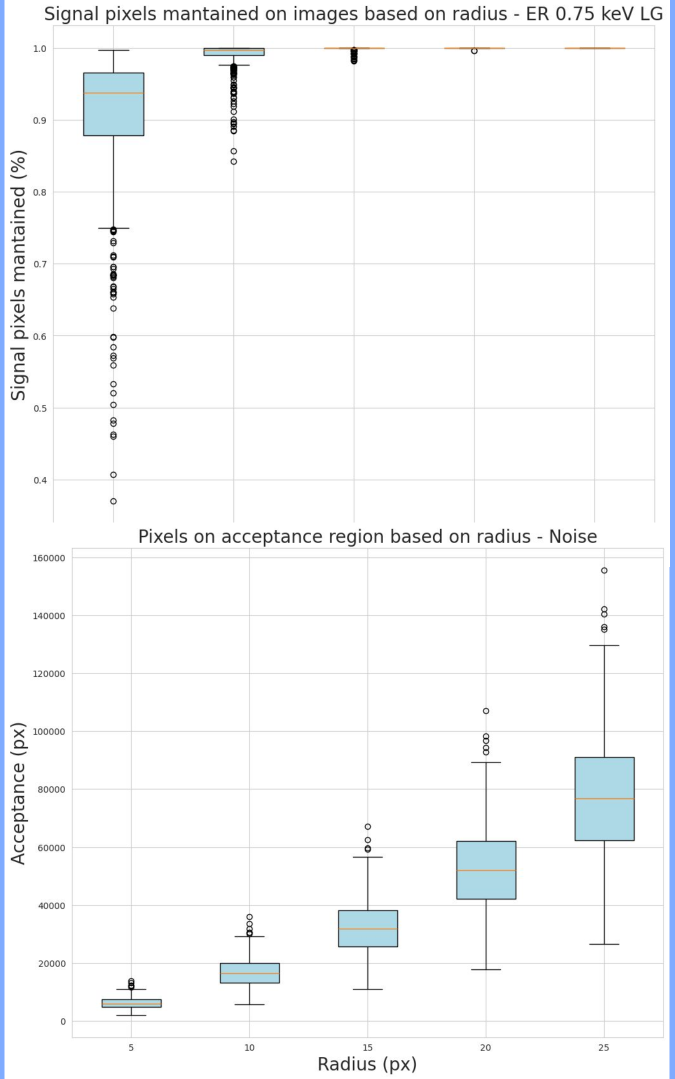
- The algorithm was applied on the **HG** data to select the **radius**.
    - A **radius** of **20** is **enough** to **detect almost all** the **pixels** of the **signal detected.**
    - **~86%** of the **0.25 keV ER HG** were **detected.**

- The test dataset was saved after the algorithm was applied.
    - Original full images: 7.1 GB
    - Images as root histograms: 229 MB
    - Images as sparse arrays (scipy): 185.7 MB
    - Images as sparse arras (root): 188.5 MB



Pixel level trigger LG

- The algorithm was applied on the **LG** data to select the **radius**.
    - A **radius** of **20** is **enough** to **detect almost all** the **pixels** of the **signal detected.**
    - **~93%** of the **0.75 keV ER LG** were **detected.**



3. Conlusions

Conclusions and next steps

- The analysis using the **new simulation** showed **consistent results** and **proves** that the methods are **resilient**.
- The **pixel level trigger** has a **good potential** to **efficiently save memory** on the **data acquisition**.
  - **High energy results** will be shown in the **next presentation.**
- **CNN** will be **retrained** to **try** to **improve performance.**
- **Quest data** will be the **focus** of **analysis** from now on.

Thank you

Credits

This is where you give credit to the ones who are part of this project.

Did you like the resources on this template? Get them for **free** at our other websites.

Presentation template by Slidesgo

Icons by Flaticon

Images & infographics by Freepik

Author introduction slide photo created by **katemangostar** - Freepik.com

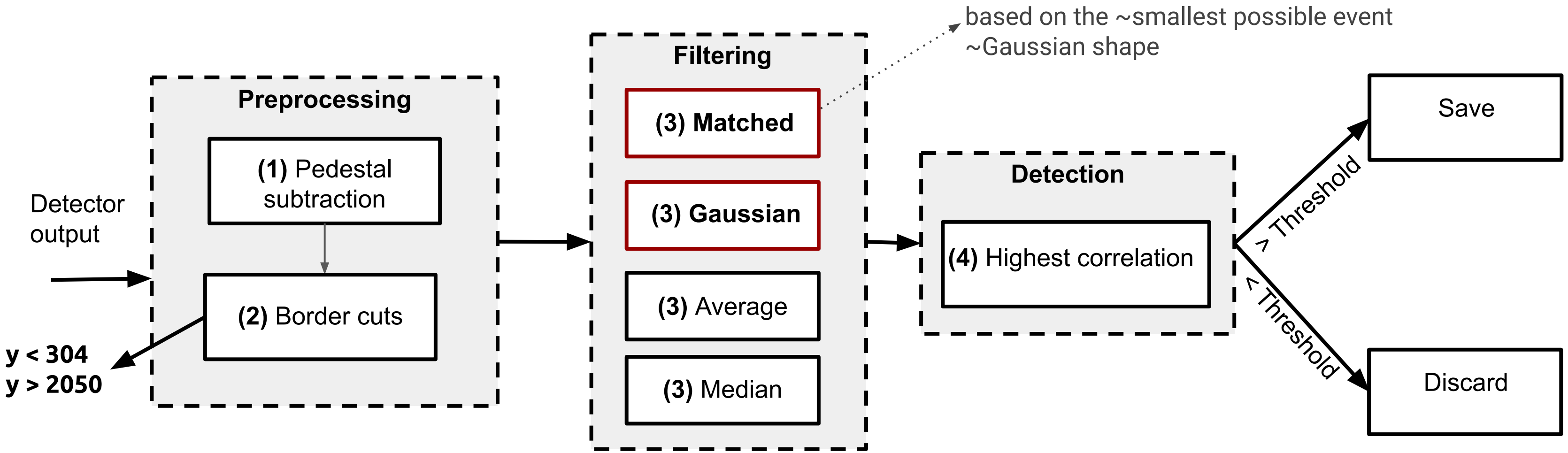
Big image slide photo created by **jcomp** - Freepik.com

Text & Image slide photo created by **rawpixel.com** - Freepik.com

Text & Image slide photo created by **Freepik**

Backup

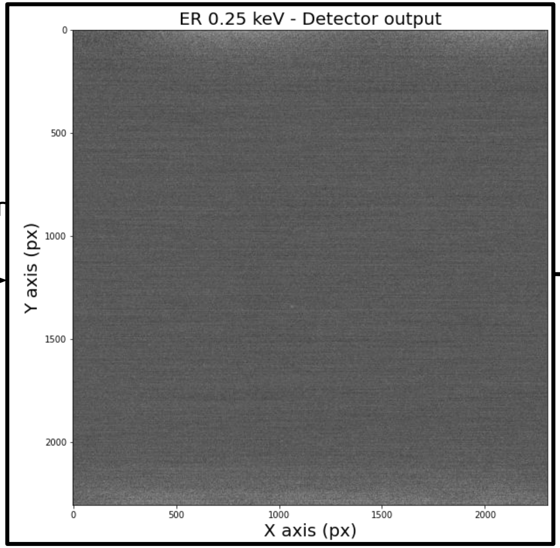
Filtering based trigger

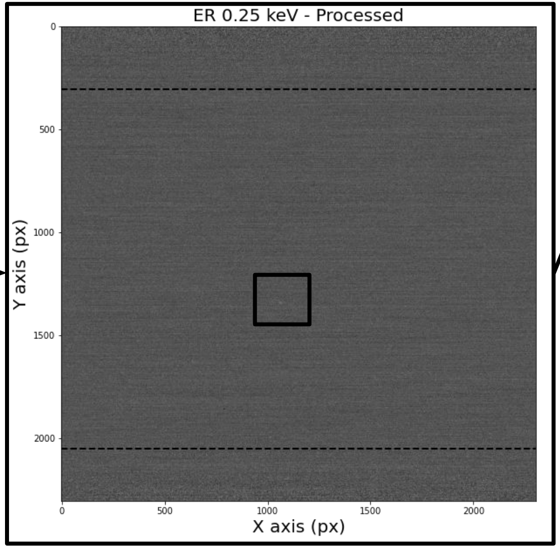


Filter parameters and detection threshold selected based on training data

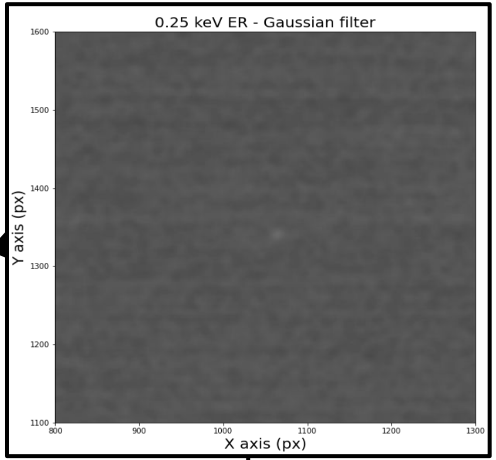
Filtering based trigger

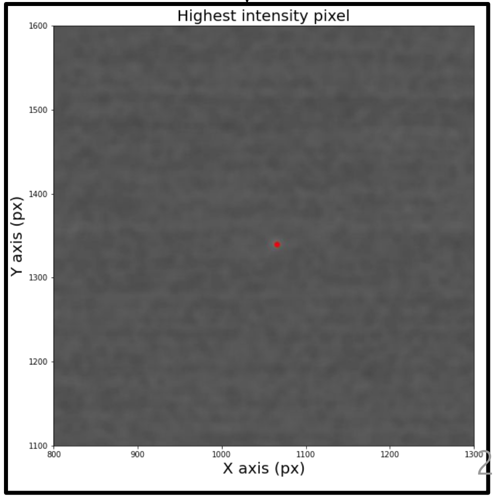
Detector output



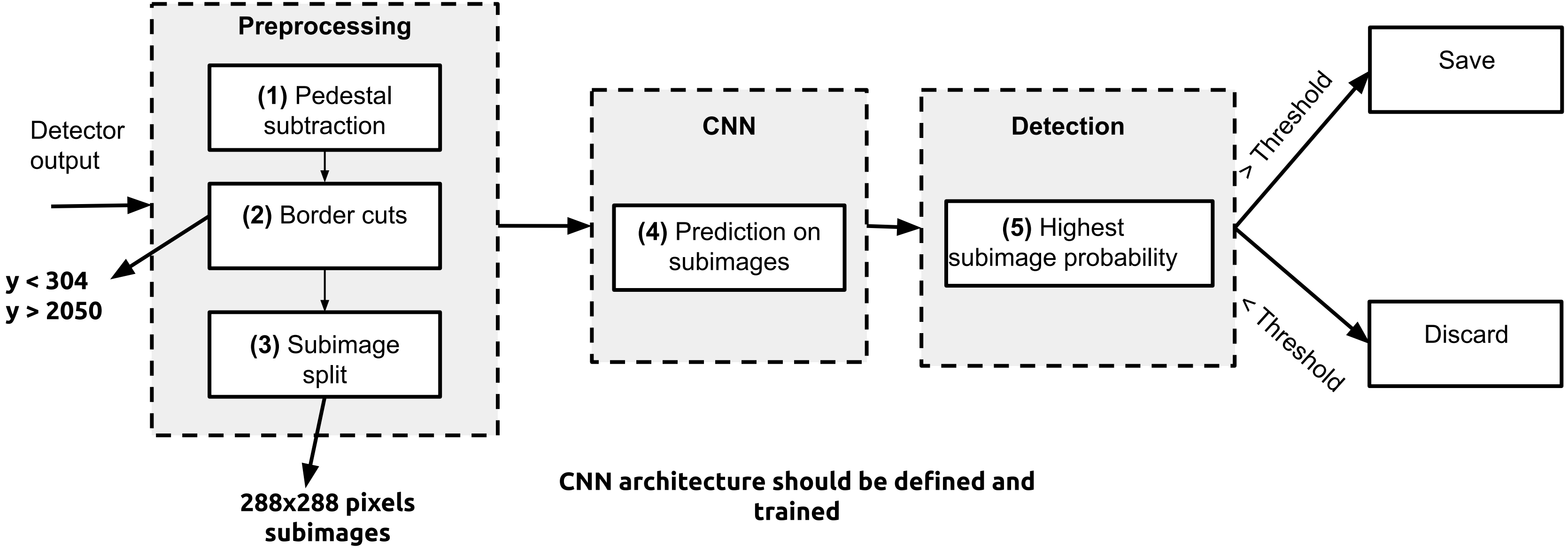


Pedsub + Border cuts





CNN based trigger

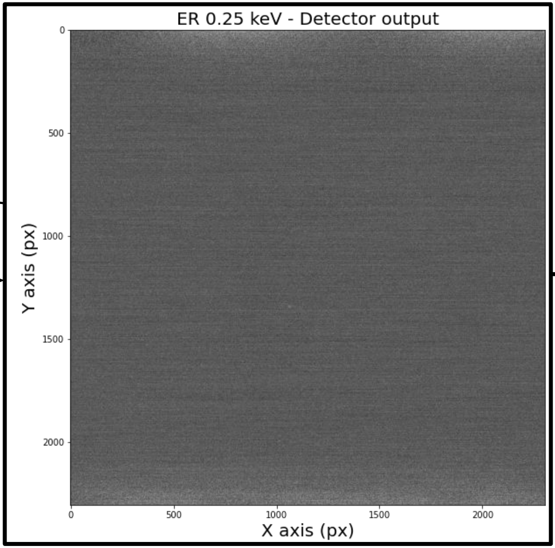


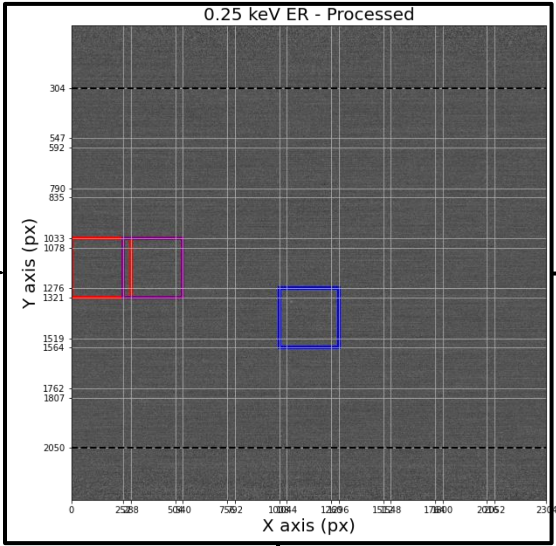
CNN architecture should be defined and trained

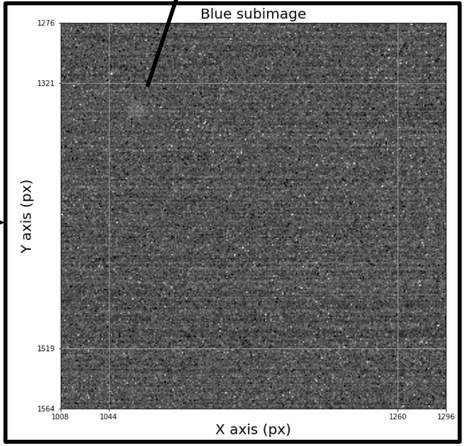
CNN based trigger

Signal Detection

Detector output

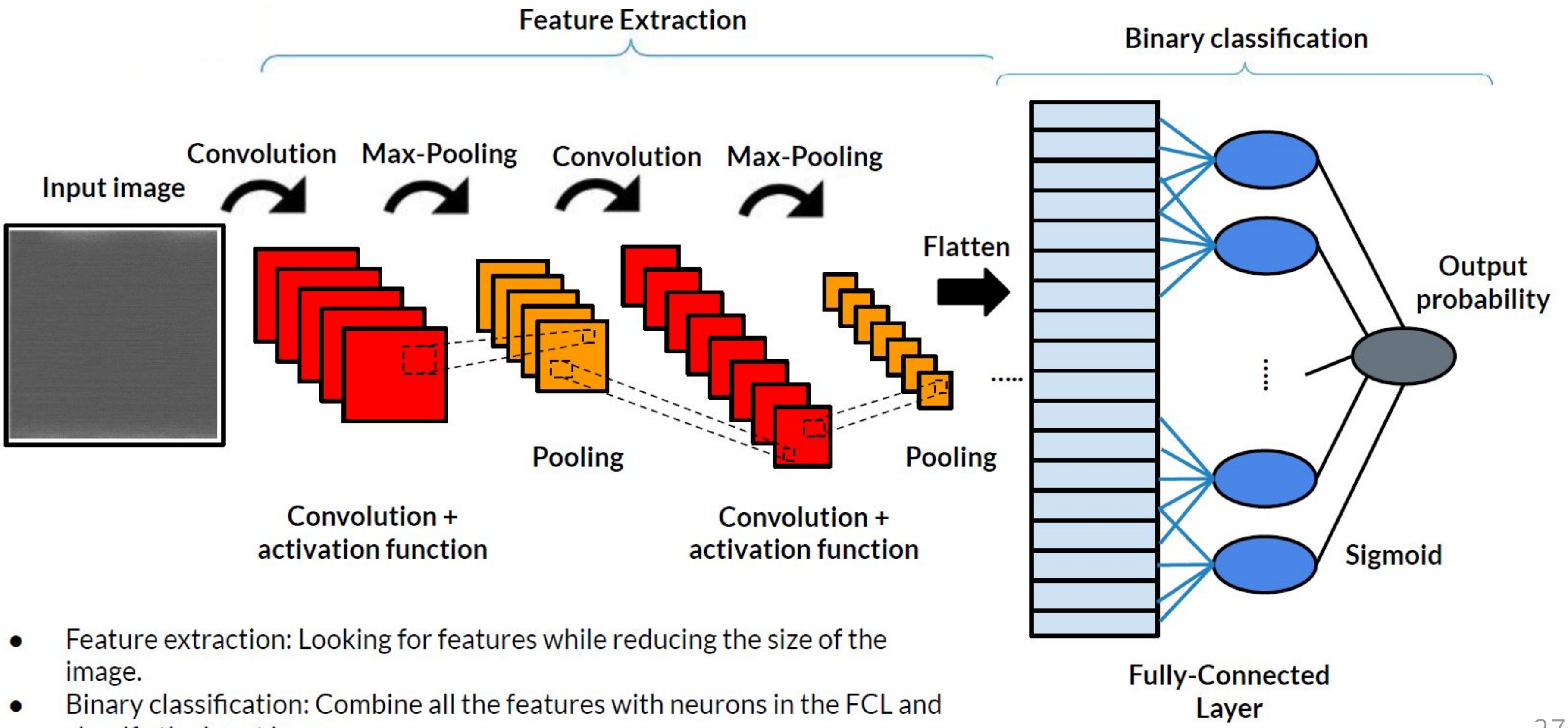






Pedsub + Border cuts + subimage divisions

CNN architecture

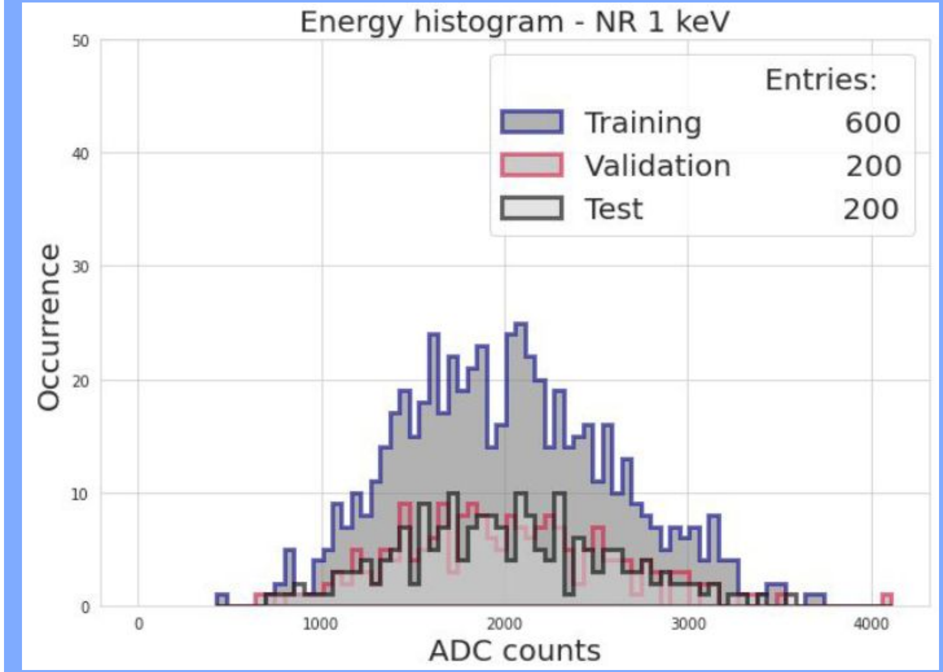


- Feature extraction: Looking for features while reducing the size of the image.
- Binary classification: Combine all the features with neurons in the FCL and classify the input image.

Datasets

- The signal simulation was **divided** considering the **balance in ADC counts** across the three datasets.

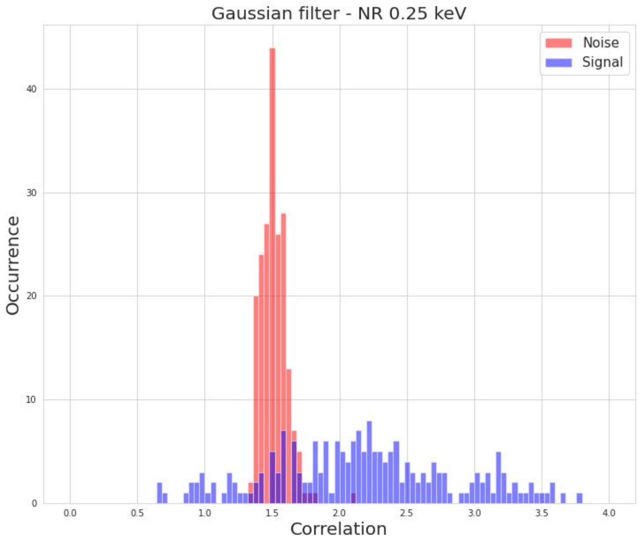
- This prevents the **data split** from **influencing** the results.

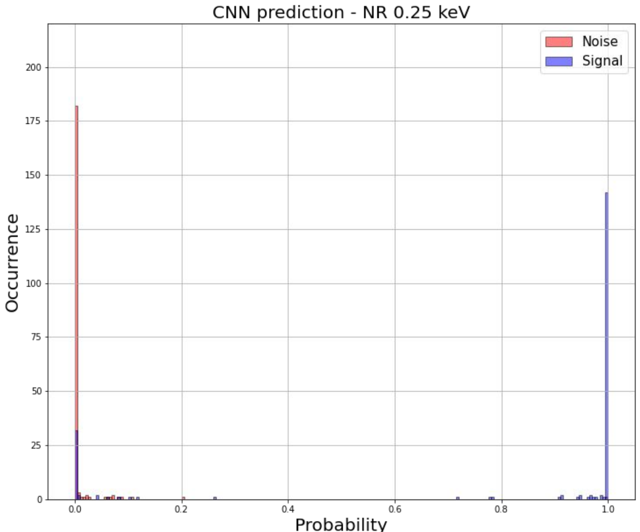


Detection performance

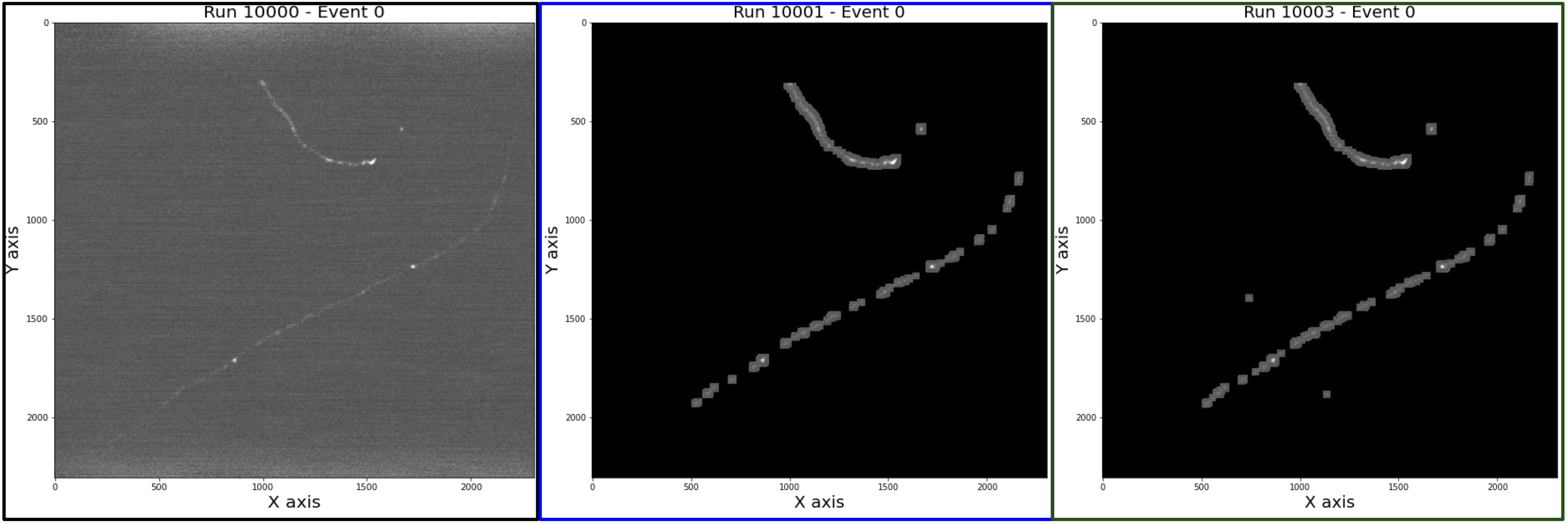
- Applying the trigger algorithms on the test dataset results in **two distributions.**
  - The **Gaussian filter** method **output** is a **correlation.**
  - The **CNN output** is a **probability** (more interpretable)

- These distributions may be used on **ROC curves** to evaluate the results.
  - All possible thresholds are used to measure the true positive rate **(TPR)** and false positive rate **(FPR).**
  - **TPR** is analogue to **signal detection.**
  - **FPR** is analogue to **false alarm.**



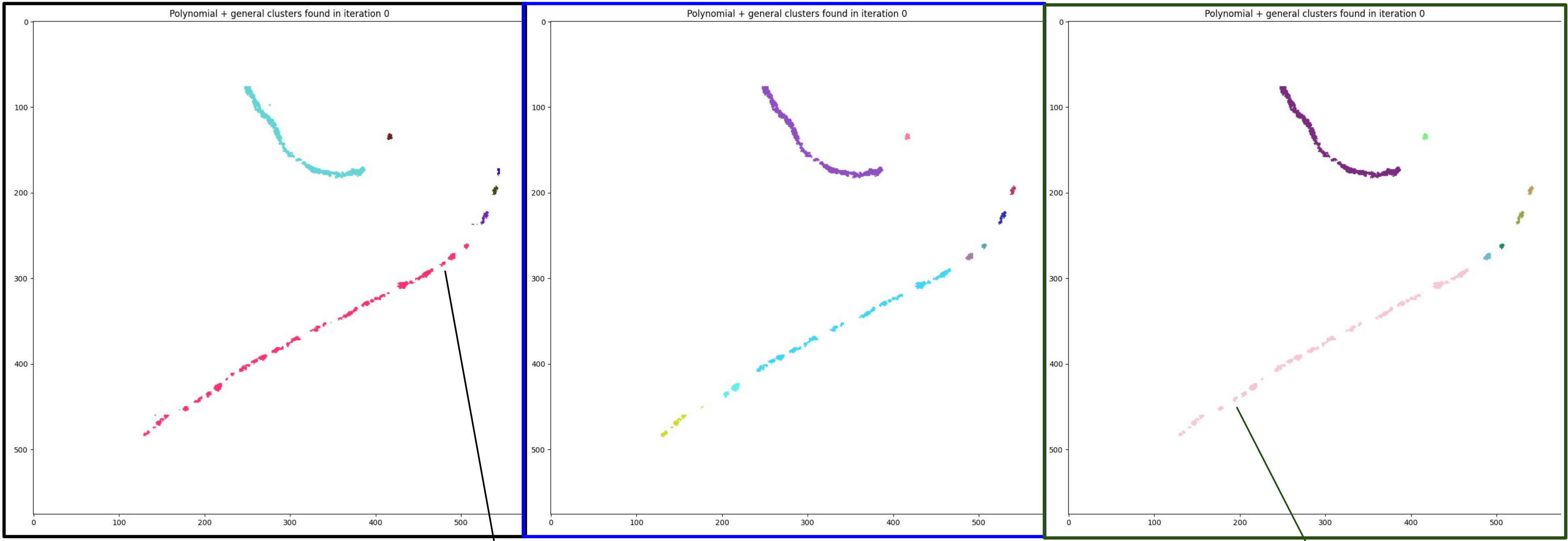


Visual example pixel level trigger



- The **pixel level trigger removes** most **noise pixels** from the images.
- The **extra regions** saved by using a **looser threshold** may keep **noise clusters** or **help** the **iDDBSCAN** to detect the **long track** as a single cluster.

Visual example pixel level trigger



Missing part on images after trigger process.

Small part worked as a bridge on the iDDBSCAN's directional search.

- The iDDBSCAN parameters may be adjusted to work on images containing less noise.