Image based trigger algorithms

Igor Pains
with Rafael Nóbrega

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.**

2. Fusion results

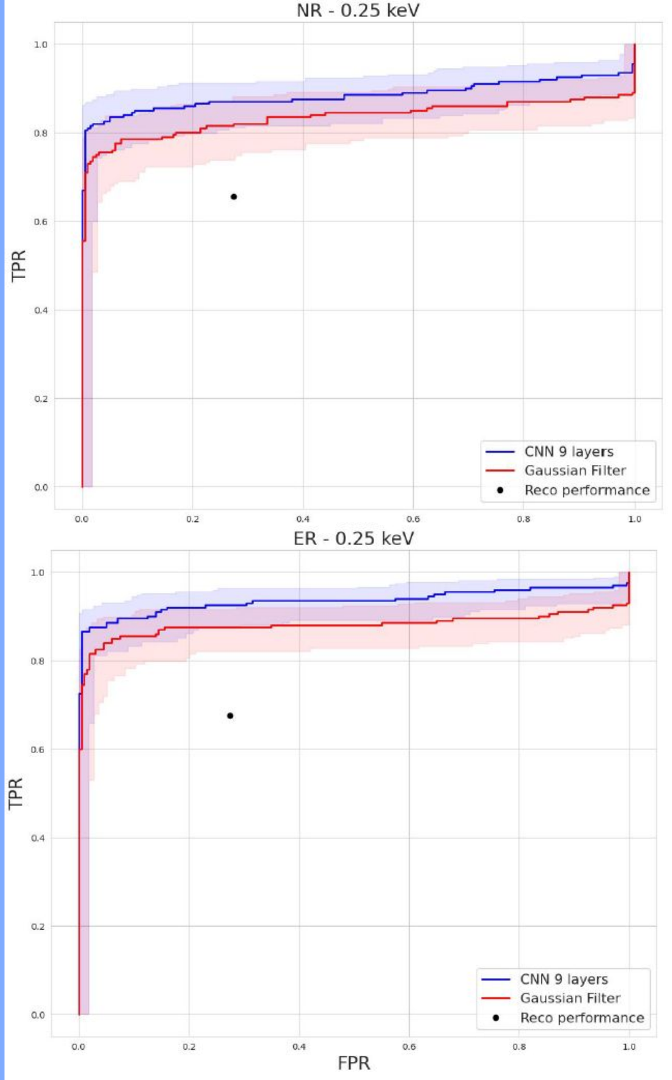
Image level trigger

- 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.**

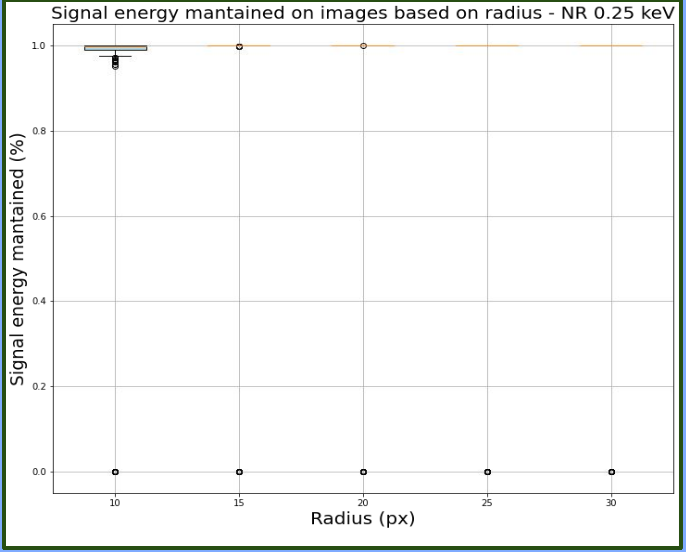


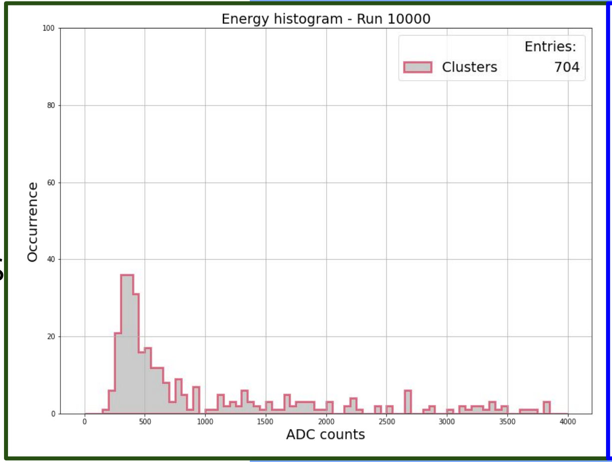
Pixel level trigger

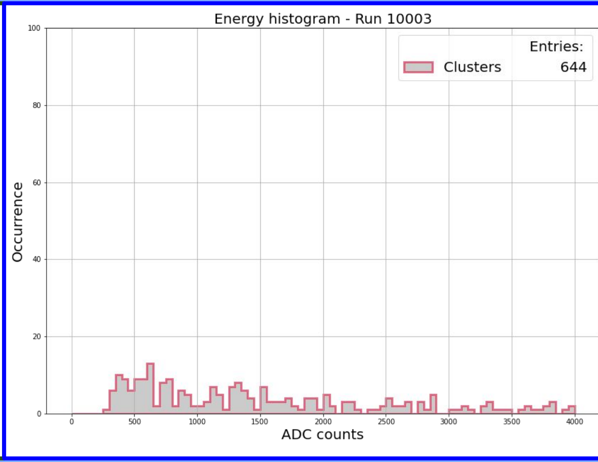
- The **idea** of the **pixel level trigger** is to apply the **Gaussian filter** looking for **high correlation** points and **save** the **region around** them. [1] [2]

- 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**).

- The reco output on a **NRAD** run before **(left)** and after **(right)** the **algorithm reduction** shows **less clusters** on the low energy region.





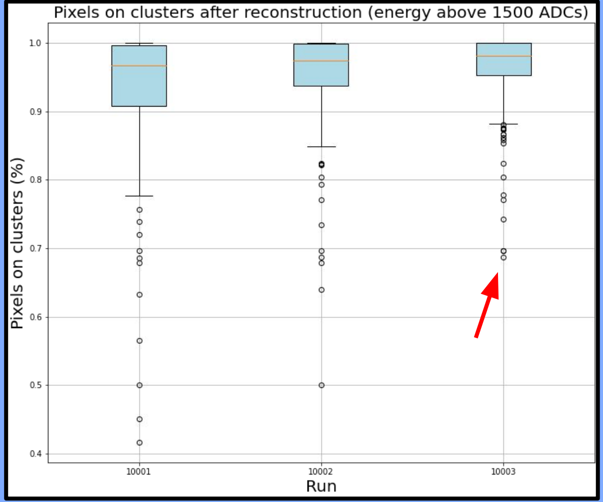


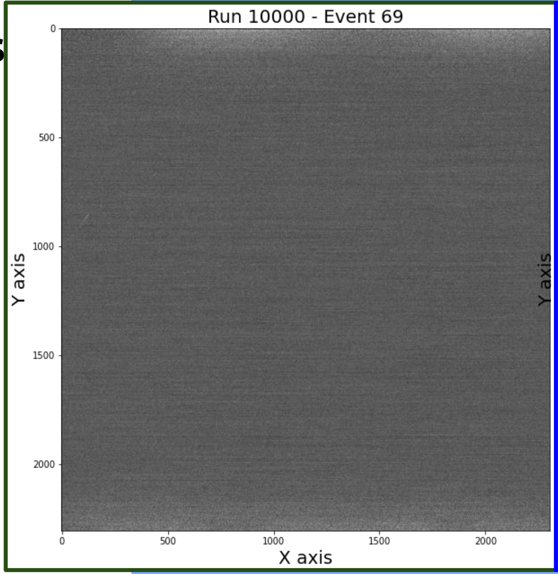
Pixel level trigger

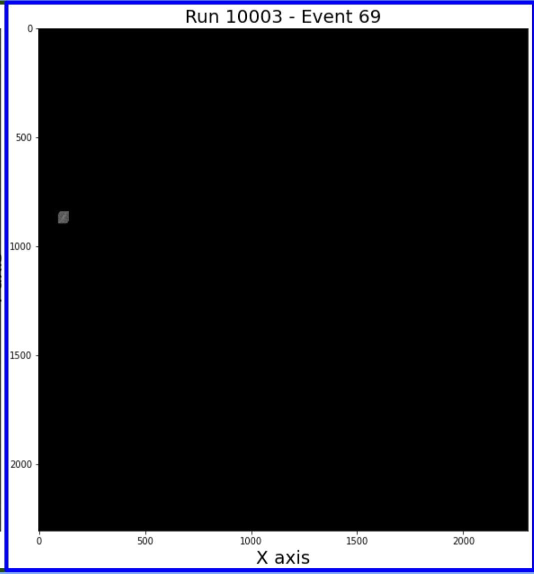
- **Pixel detection** performance on **NRAD** can be **estimated** by using the **reconstructed pixels** as **target** and **checking** the **percentage** of **maintained pixels (underestimates** the **algorithm).**

- It **preserved** more than **90%** of **pixels** from **clusters** with **> 1500 ADCs** and **reduced ~35x** the **memory used** to **store** the **images.**

- A **visual example** of the **worst case** shows that some **noisy pixels** were **possibly considered** as **target** by this **approach.**







3. Quest

Expectations

- There are some **differences** between the **noise** from **Fusion** and **Quest**:
  - The pedestal levels are **(100.87±1.08)** and **(199.71±1.32)** respectively.
  - The deviation of pedestal are **(3.60±1.51)** and **(2.28±0.94)** respectively.
  - **Quest** has some problems with **sparks** (at **random** and **fixed** positions).

- **Lower GEM gain** will **alter** the **signal detection efficiency.**
  - **Signal ADC counts** reduced by **~2x** with this change in **Fusion** (comparing 1 keV simulation)**.**
  - **^{55}Fe ADC level** on **Quest** seems to be **slightly higher** than on **Fusion.**

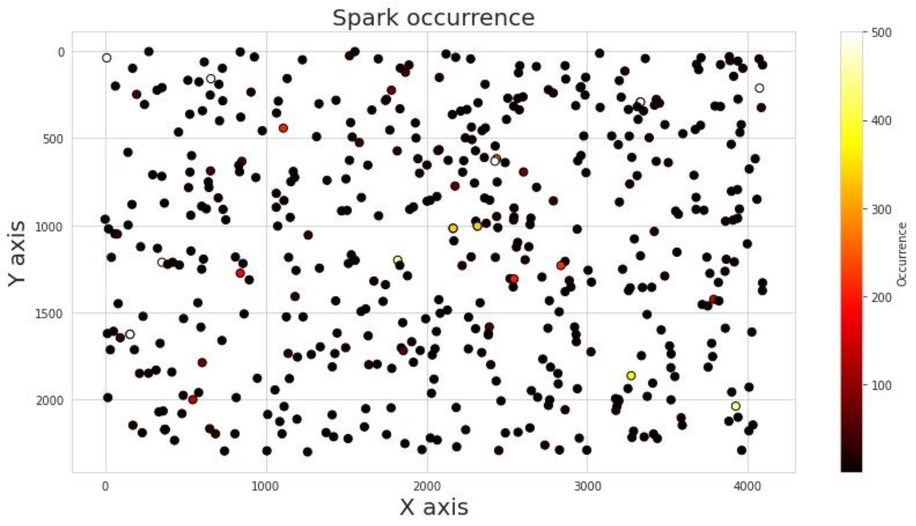
- The **expectation** is that **0.5 keV** on **Quest** should have slightly **higher detection efficiency** compared to **0.25 keV** on **Fusion** (**440 GEM gain**).
  - Need simulation to confirm this result and train the trigger algorithms.

Quest noise

- A **spark mapping** was **performed** on **500 pedestal images** from **Quest.**
  - **472 pixels** had an **intensity** above **300 ADCs** on **at least one image** (**28** of them on **more** than **50 images**).

- The **spark** generally **appears** on a **single pixel** and **don't affect** its **neighbors**.
  - It may **affect** the **CNN** and **filtering** as this **higher intensity** is **spread** to the **neighbors** by the **convolution**.

- A **possible solution** is to apply **derivative filter** and **discard pixels** above a threshold (i.e **400** would **require** a **step** of **50 ADCs** in each **direction**).

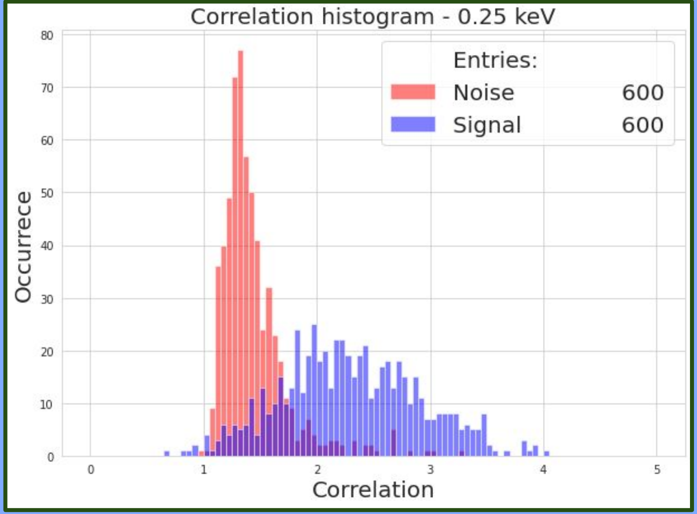


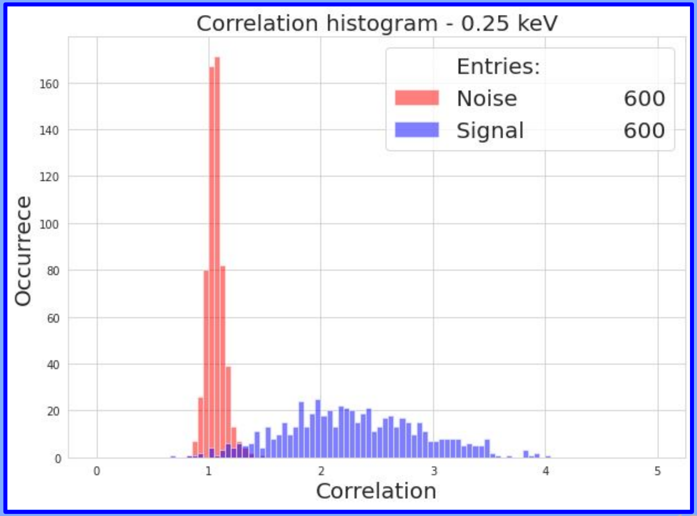
Filtering test

- The **Fusion simulation** was used to **test** how this **preprocessing** could **affect** the **noise** and **signal**.
  - **Fusion simulation** was combined with **Quest noise**.

- The **noise distribution** after filtering was **shrinked** whereas the **signal's** was **not affected**.
  - **Gaussian filter** was **used** on the images **without** and **with** the **preprocessing**.

- The **signal detection performance** will be **measured** with the **Quest simulation** data.





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

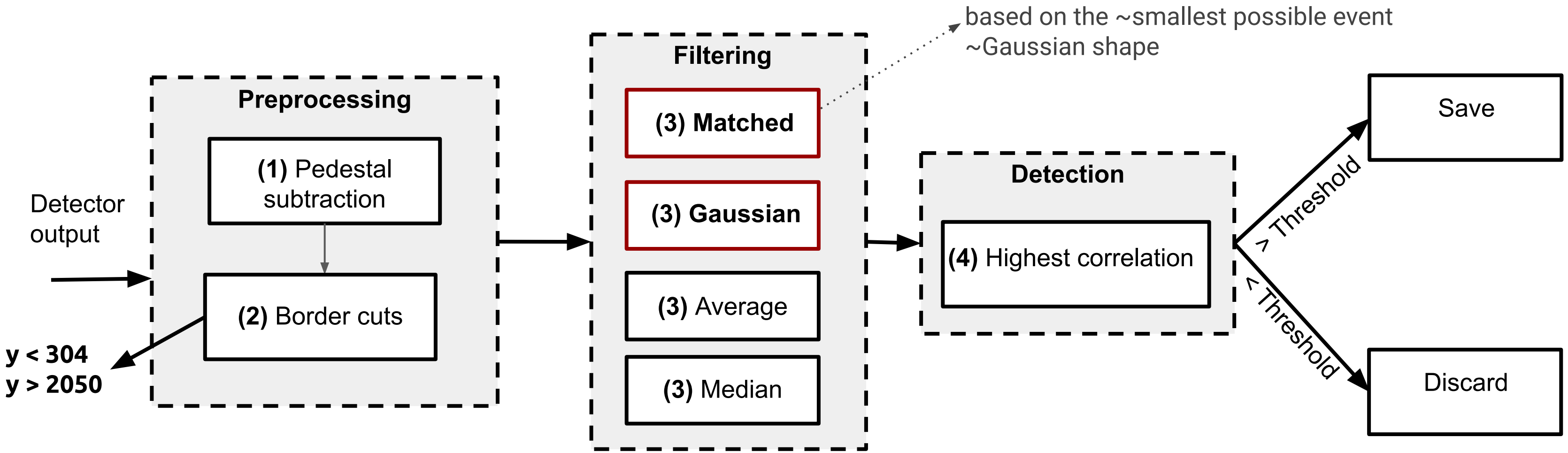
Big image slide photo created by **jcomp** - Freepik.com

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

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

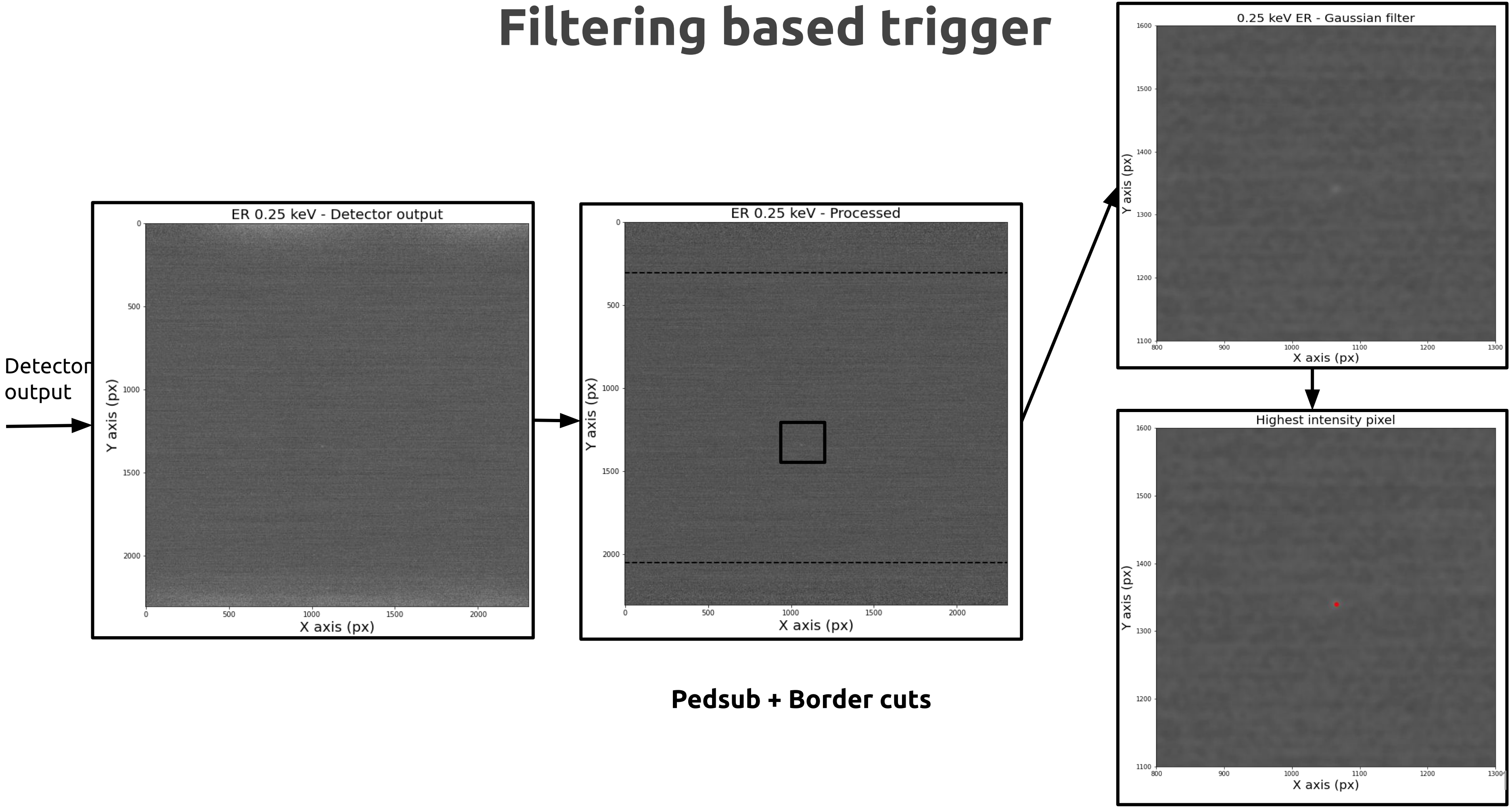
Backup

Filtering based trigger



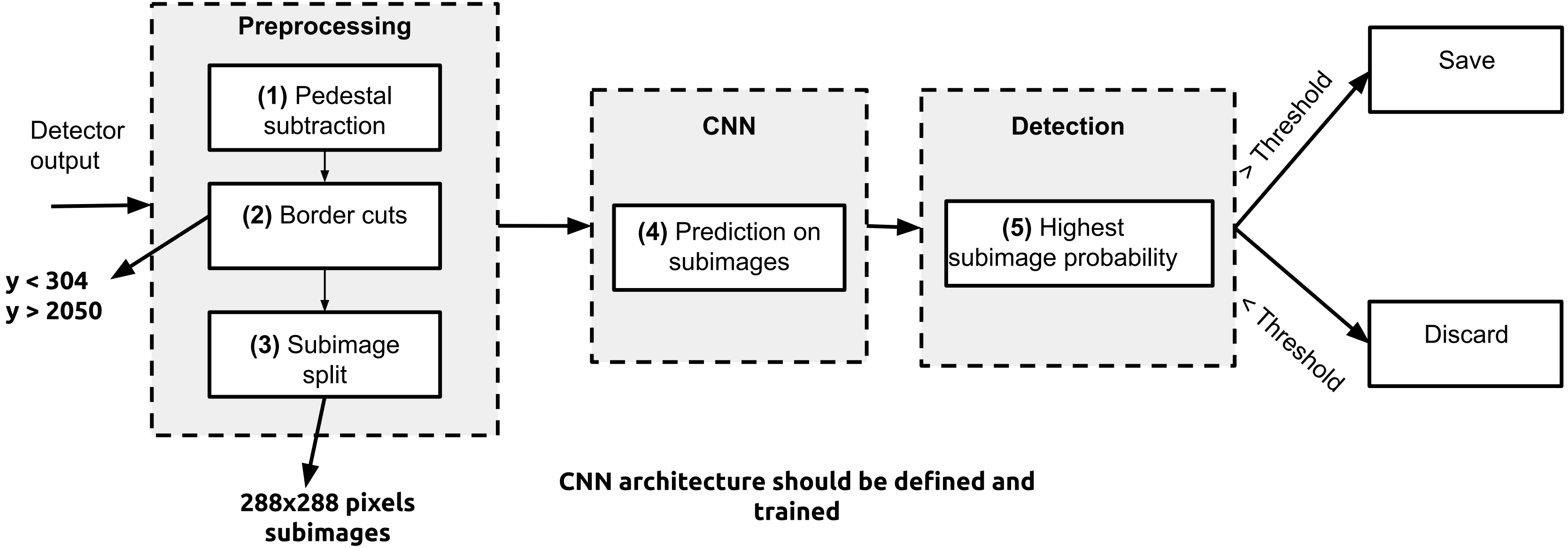
Filter parameters and detection threshold selected based on training data

Filtering based trigger



Pedsub + Border cuts

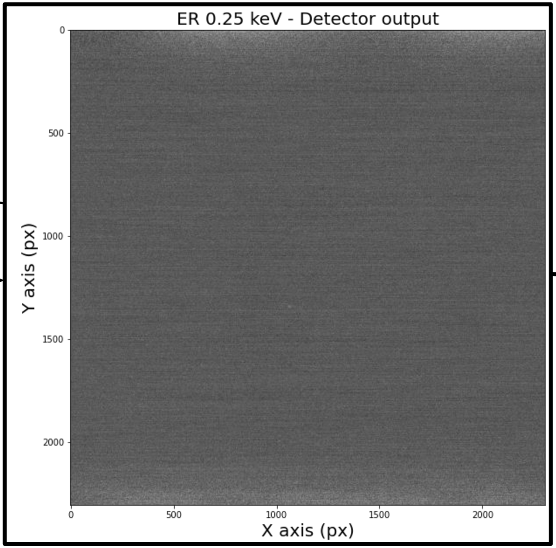
CNN based trigger

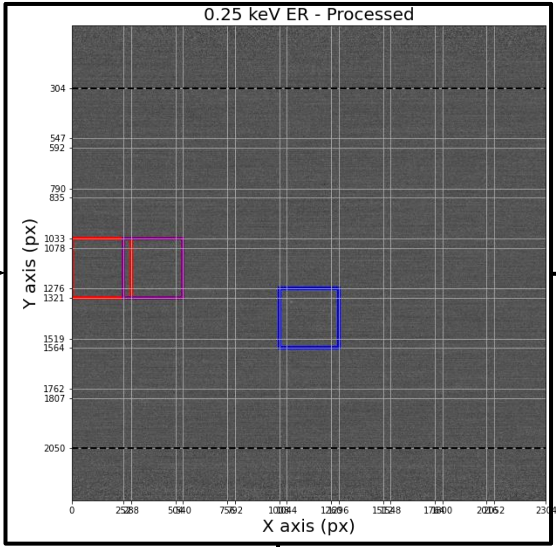


CNN architecture should be defined and trained

CNN based trigger

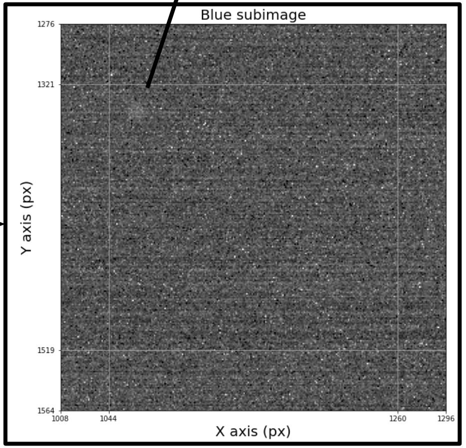
Detector output



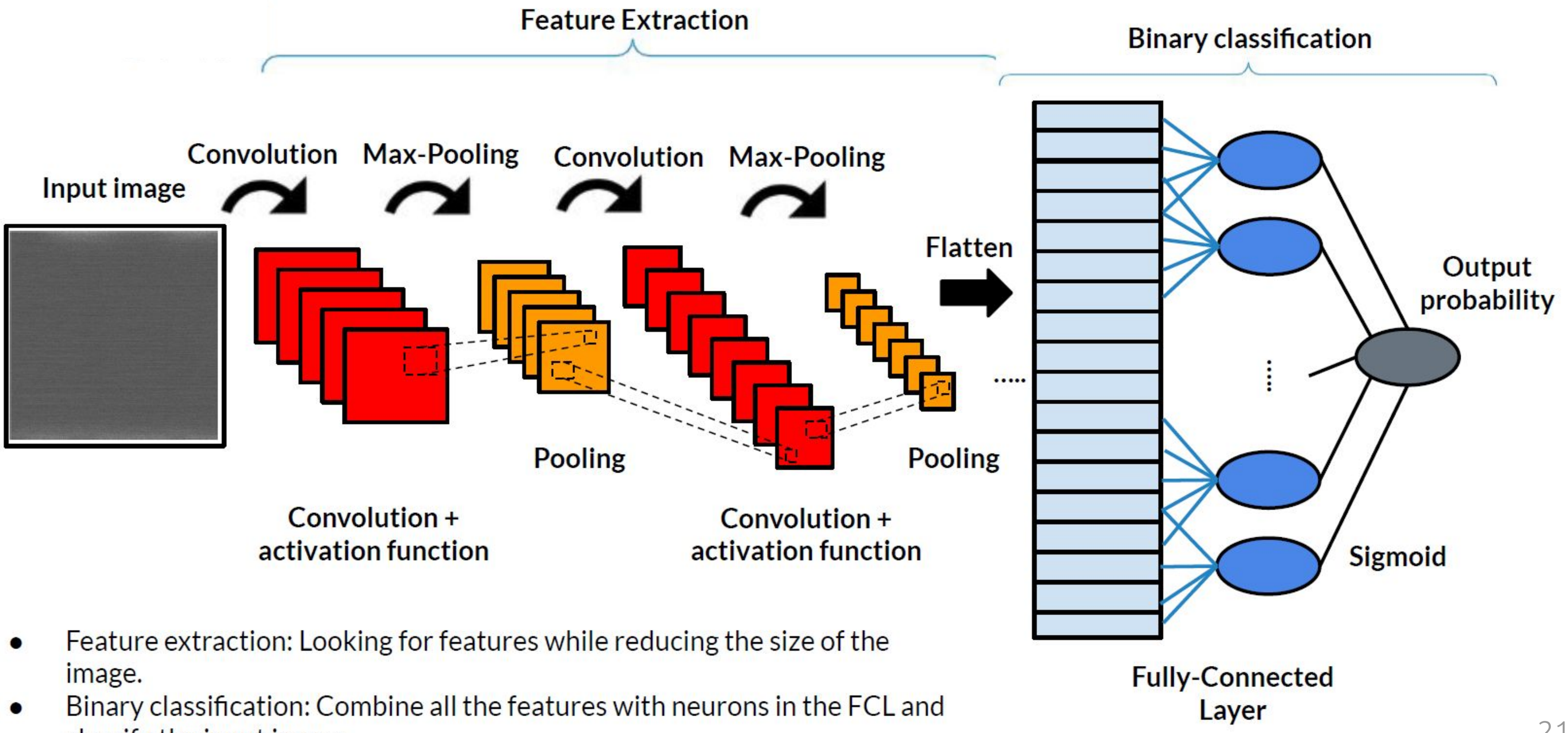


Pedsub + Border cuts + subimage divisions

Signal Detection



CNN architecture

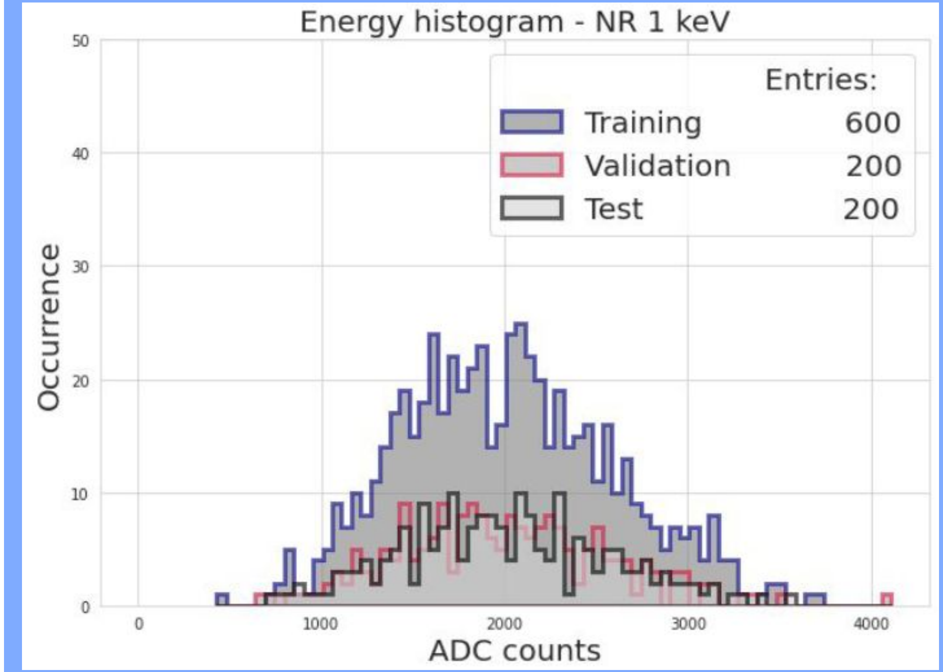


- Feature extraction: Looking for features while reducing the size of the image.
- Binary classification: Combine all the features with neurons in the FCL and classify the input image.

Datasets

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

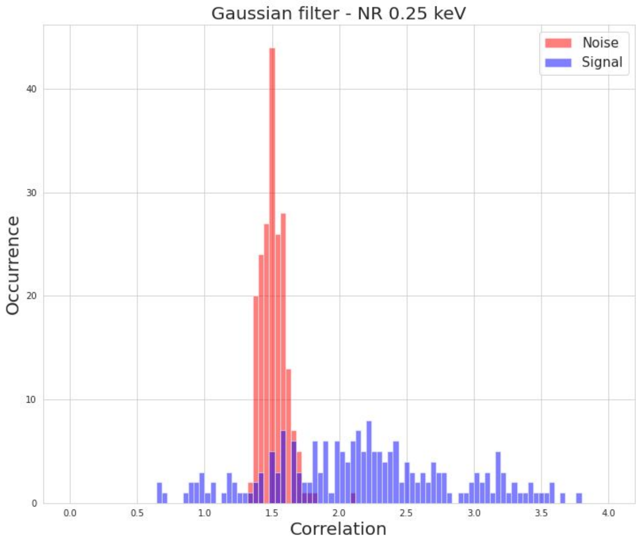
- This prevents the **data split** from **influencing** the results.

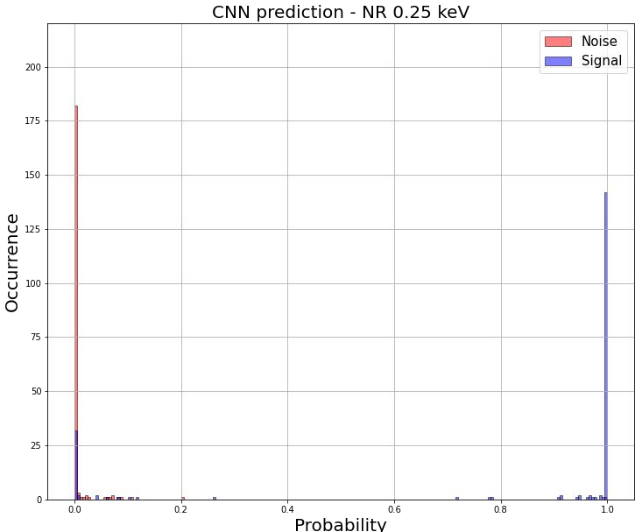


Detection performance

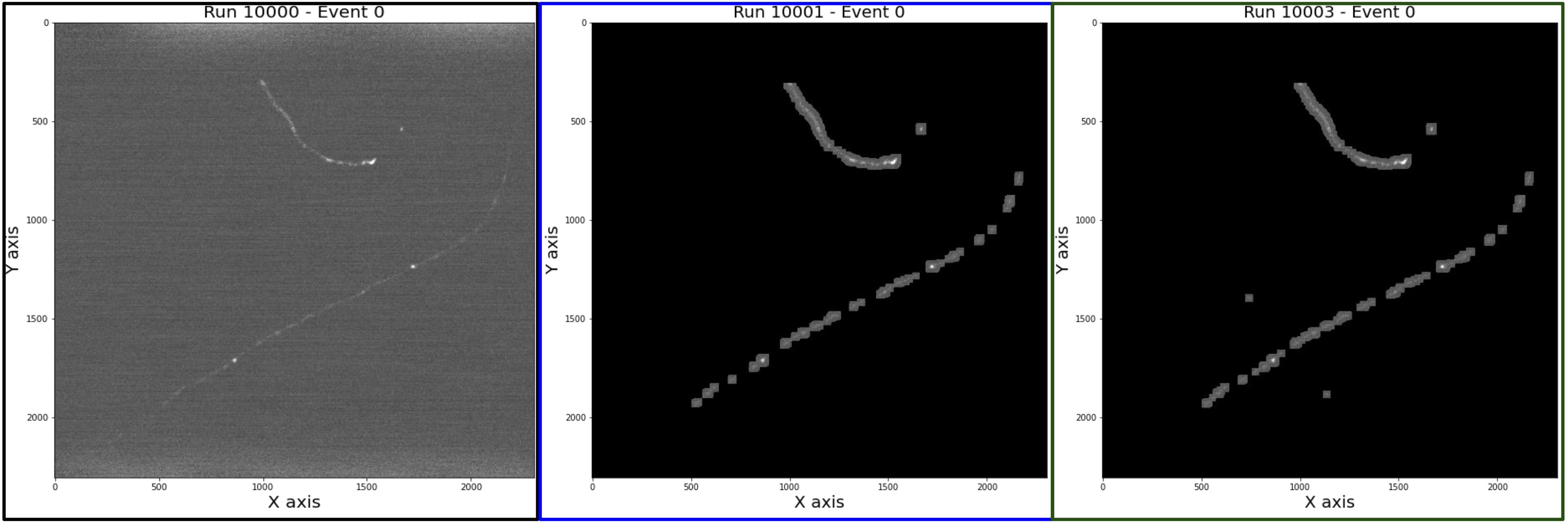
- Applying the trigger algorithms on the test dataset results in **two distributions.**
  - The **Gaussian filter** method **output** is a **correlation.**
  - The **CNN output** is a **probability** (more interpretable)

- 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.**





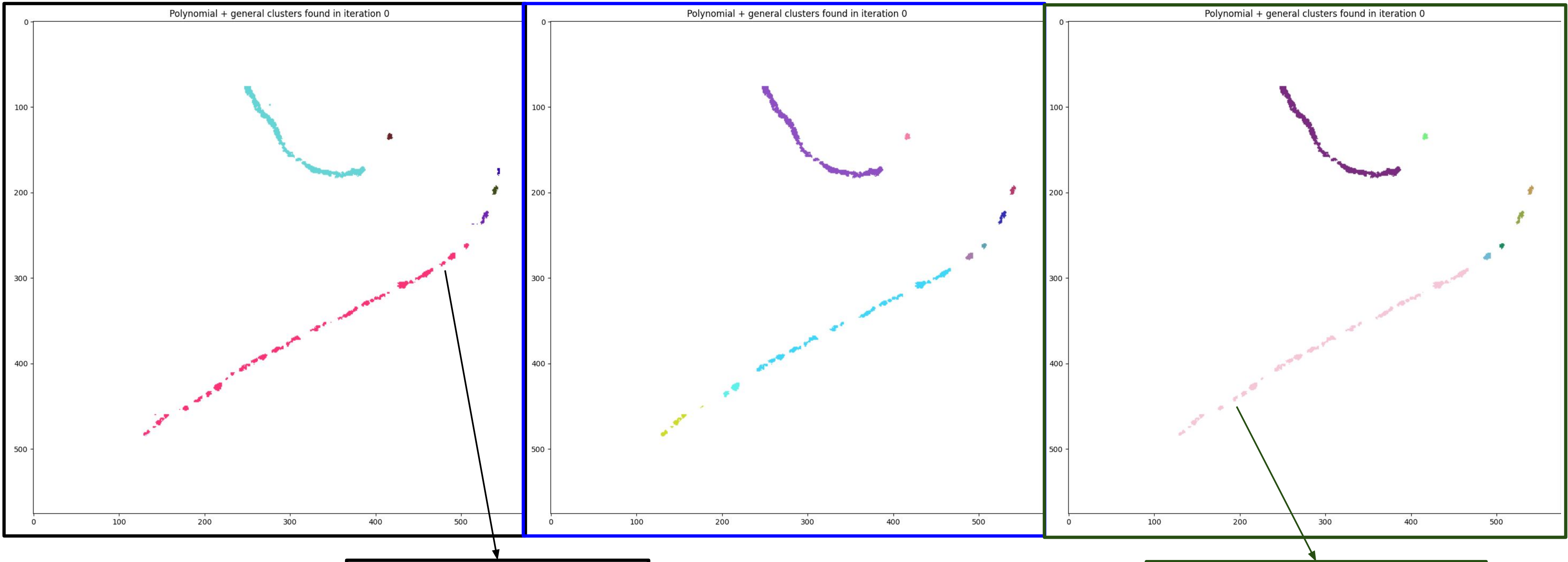
Visual example pixel level trigger



- 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.