

CMOS Cosmic Ray Detector (CCRD): A desktop muon detection software for citizen science

Noemi Zabari¹, **Katsiaryna Katrankova**¹, Magdalena Dobrowolska¹, Azam Zabihi², Piotr Kalaczyński^{2,4}, Piotr Gawron^{3,4}

1 - Muotech, Rzeszow, Poland

2 - Astrocent, Nicolaus Copernicus Astronomical Center of the Polish Academy of Sciences, Warsaw, Poland

3 - Nicolaus Copernicus Astronomical Center of the Polish Academy of Sciences, Warsaw, Poland

4 - Center of Excellence in Artificial Intelligence, AGH University of Krakow, Krakow, Poland



Budapest, June 2026

Muography for all

Project goal

Lightweight, cross-platform app
for real-time detection

Hardware strategy

Utilizing existing
consumer-grade webcams

Citizen science

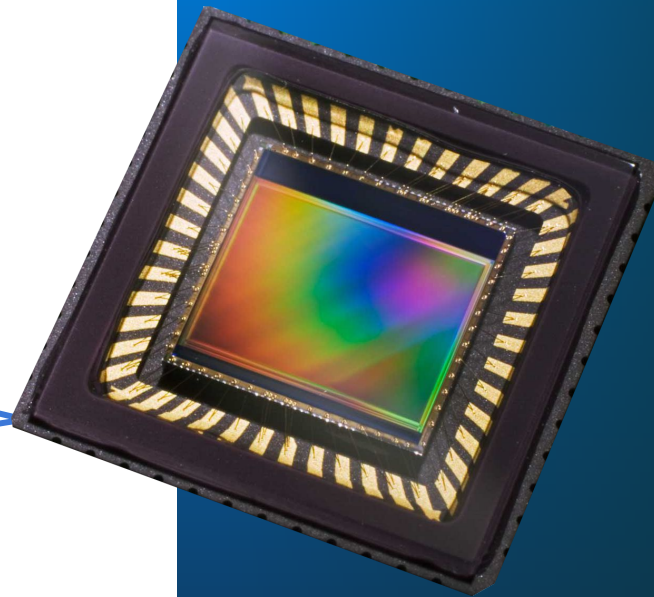
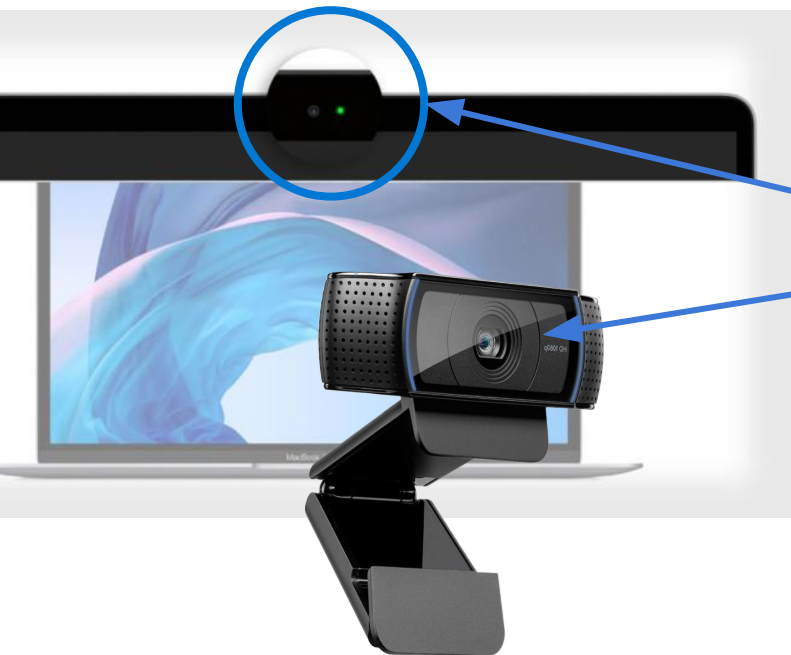
Complementary to global
networks

Accessibility

No specialized detectors required



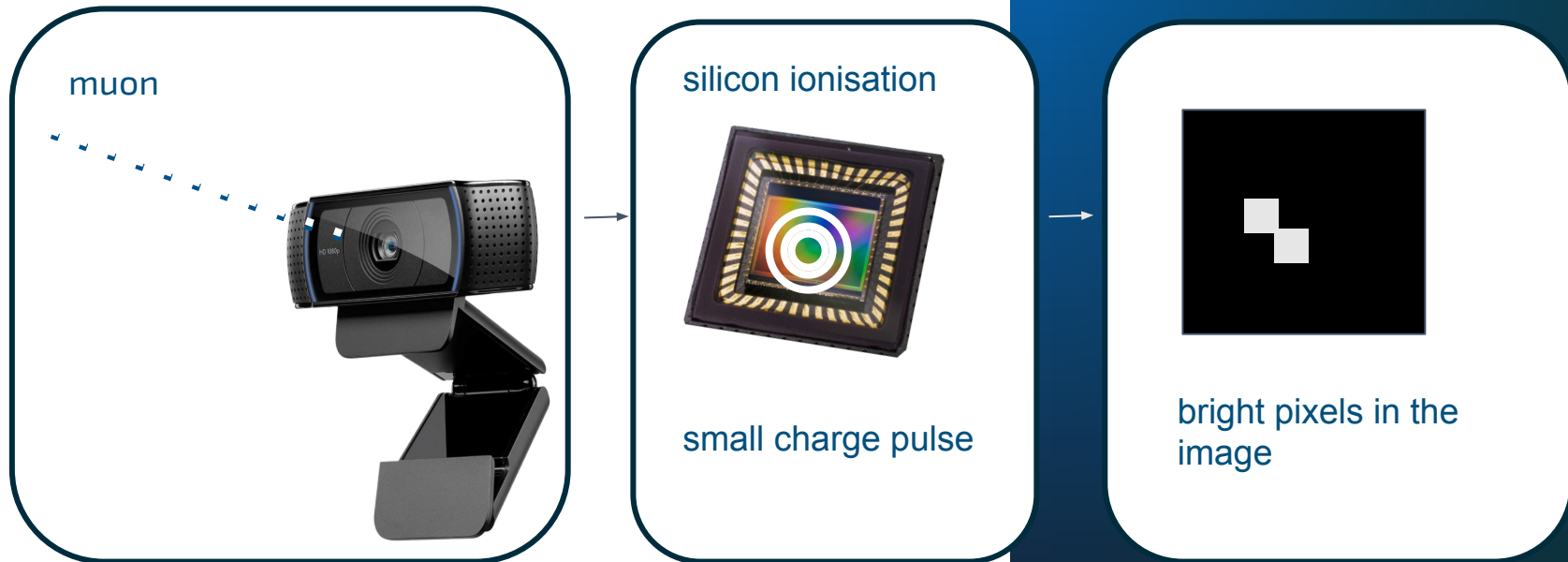
Detection principle



CMOS sensor

Detection principle

Main challenge: to separate true particle events from noise



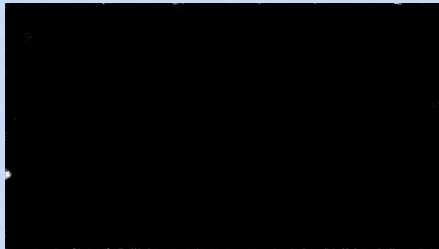
Detection principle

Solution

1

Noise calibration

50 dark frames



median noise map

2

Track classification

Detection principle

2 Track classification

$L > 0.8 \rightarrow$ MUON

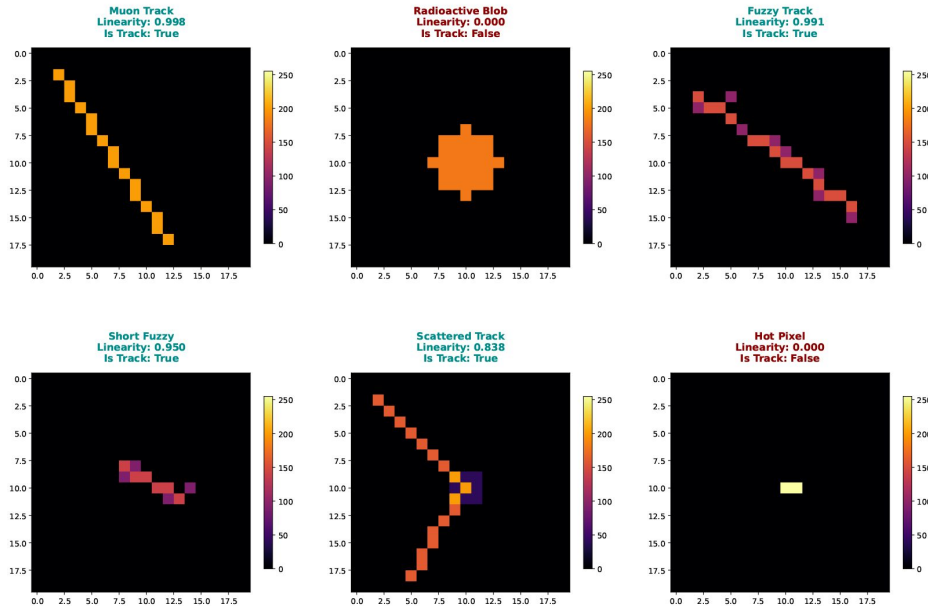
$L < 0.8 \rightarrow$ NOT A MUON

Shape-based method

6 test cases on synthetic data:

Score:0.998

Score:0.000

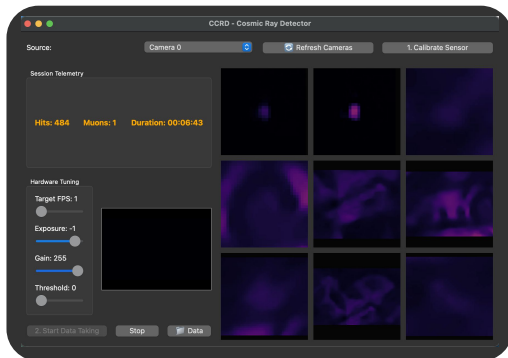


A track is classified as a muon if a certain linearity score **threshold** is exceeded (default: $L > 0.8$), indicating a highly elongated (linear) shape, whereas blobs (noise/alphas) result in $L \approx 0$.

Software Architecture

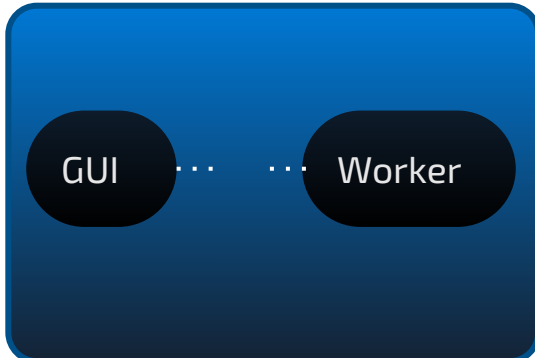
Frontend

PyQt6 for OS native
visualisation



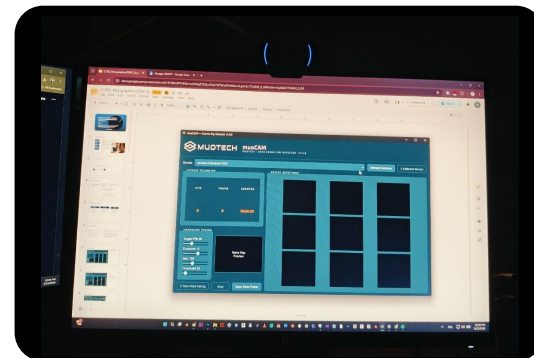
Worker thread

**Decoupled
multi-threaded logic**
*detect without UI lag

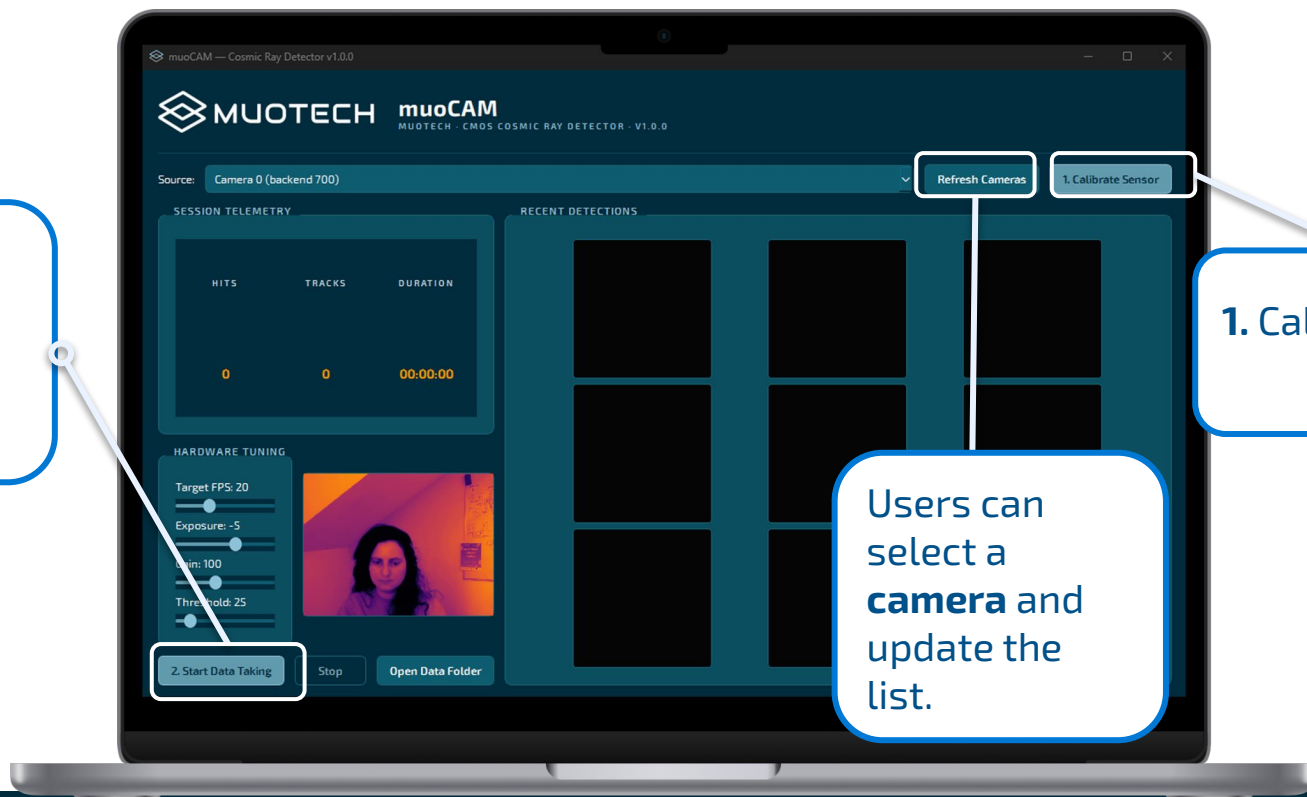


Backend

OpenCV for cross-
platform **hardware
abstraction**



User Interface



2. Start data taking

1. Calibration

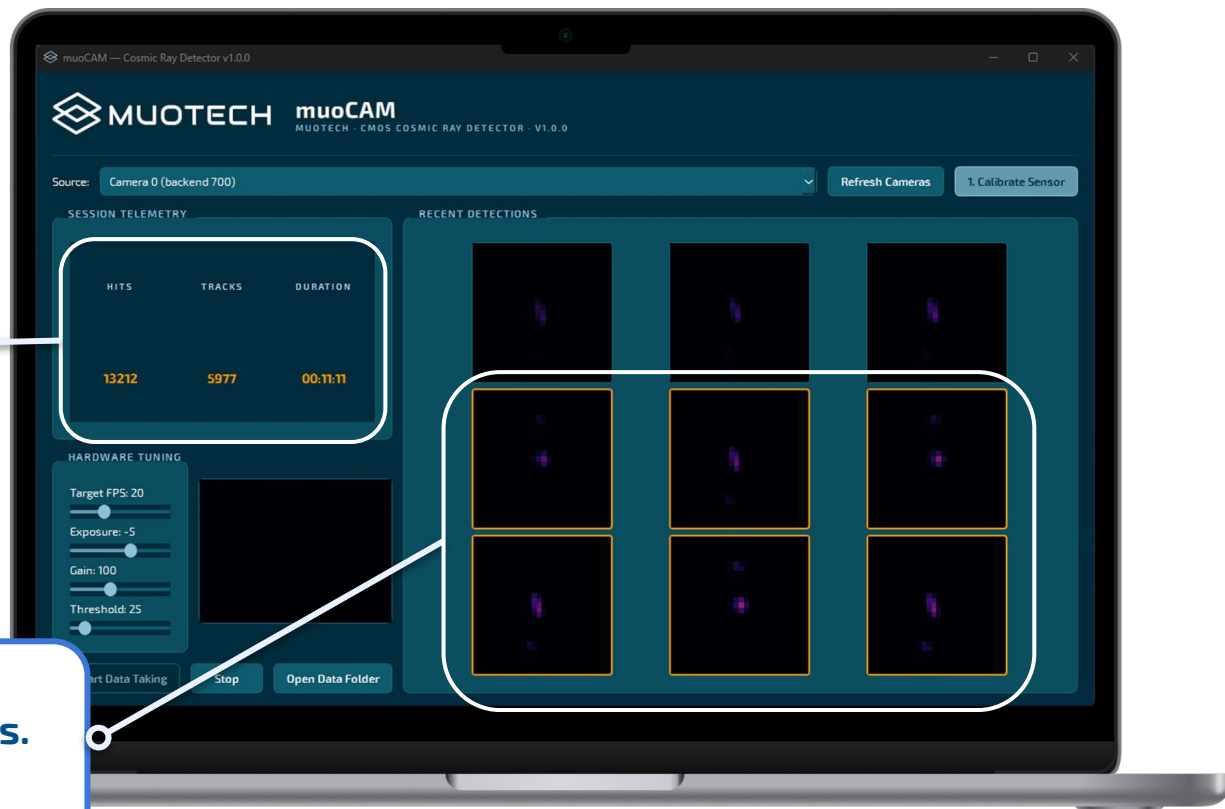
Users can select a **camera** and update the list.

User Interface

Live session telemetry

The app shows
hits, muons, duration

Recent event images.

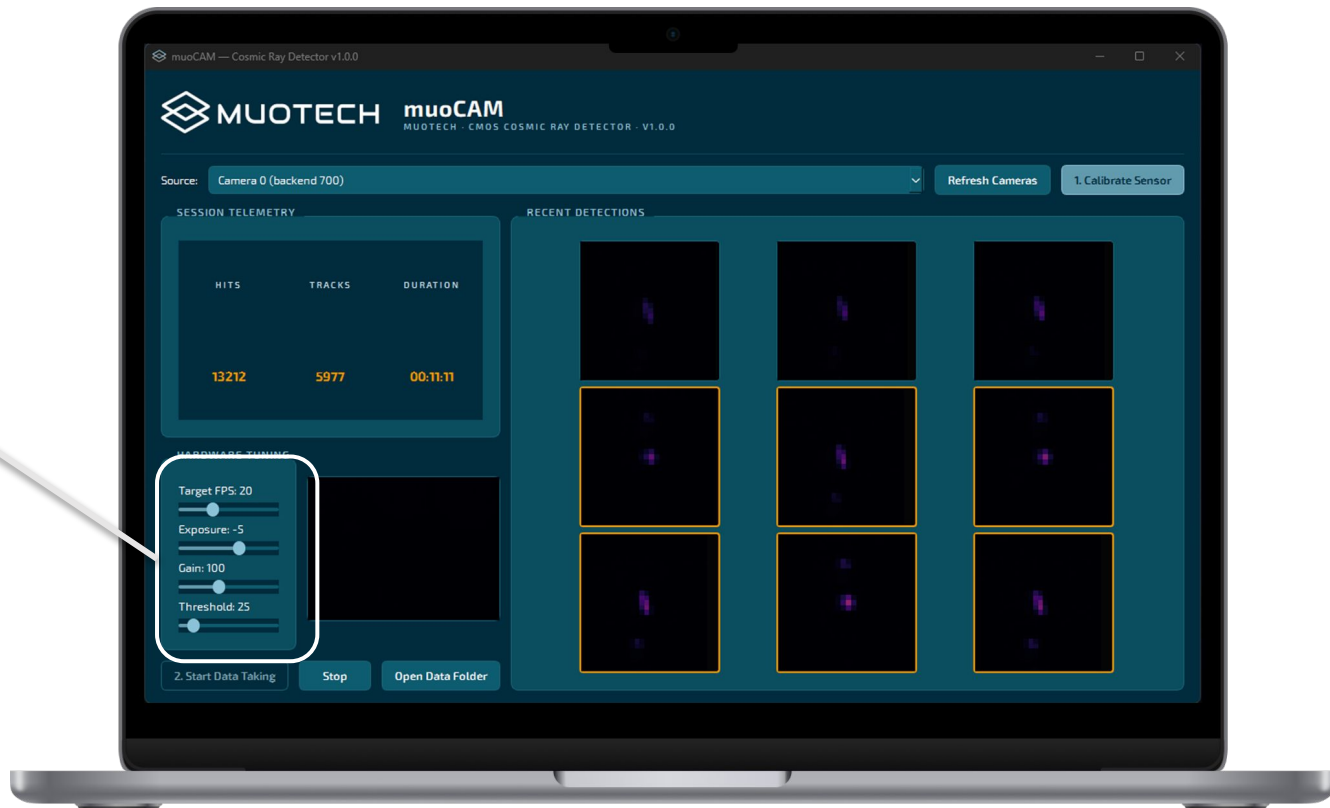


User Interface

Live sliders

Live sliders adjust

- **FPS** (Frames Per Second)
- **exposure**
- **gain**
- **threshold**



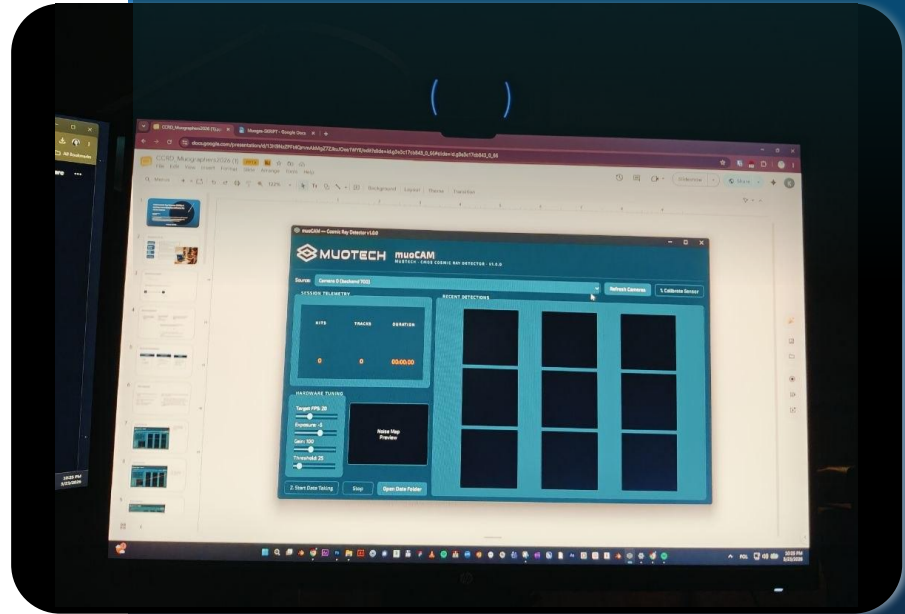
Data taking recommendation

Optical Isolation

Block all light to prevent optical noise

Thermal stability

Keep sensor away from heat sources



Performance & Compliance

Real-time processing performance



~2.2ms

**Average processing
latency per frame**

Target limit -50 ms

350 MB

Footprint

0%

Frame Loss

CPU usage is 30–50%
depending on platform

Deployment

PyInstaller creates self-contained Windows and macOS builds

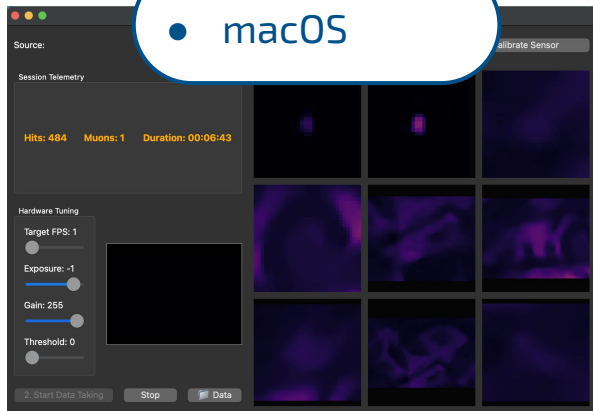
The pipeline supports **.exe** and **.dmg** packaging



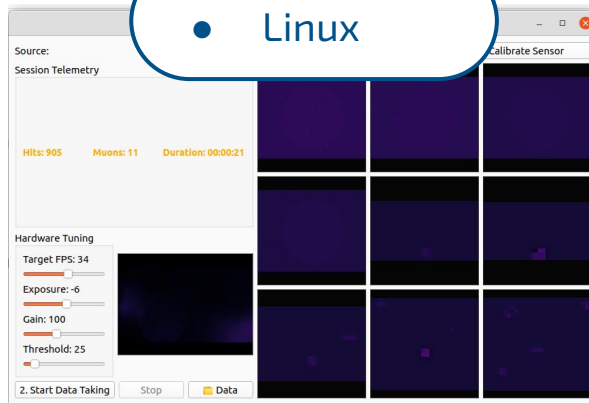
```
2023-08-08 10:11:18.000 | notacja pominięta! Call 'lsblk' w katalogu <...>
2023-08-08 10:11:18.000 | notacja pominięta! Call 'lsblk' w katalogu <...>
2023-08-08 10:11:18.000 | notacja pominięta! Call 'lsblk' w katalogu <...>
2023-08-08 10:11:18.000 | notacja pominięta! Call 'lsblk' w katalogu <...>
2023-08-08 10:11:18.000 | notacja pominięta! Call 'lsblk' w katalogu <...>
2023-08-08 10:11:18.000 | notacja pominięta! Call 'lsblk' w katalogu <...>
2023-08-08 10:11:18.000 | notacja pominięta! Call 'lsblk' w katalogu <...>
2023-08-08 10:11:18.000 | notacja pominięta! Call 'lsblk' w katalogu <...>
2023-08-08 10:11:18.000 | notacja pominięta! Call 'lsblk' w katalogu <...>
2023-08-08 10:11:18.000 | notacja pominięta! Call 'lsblk' w katalogu <...>
```

Cross-platform

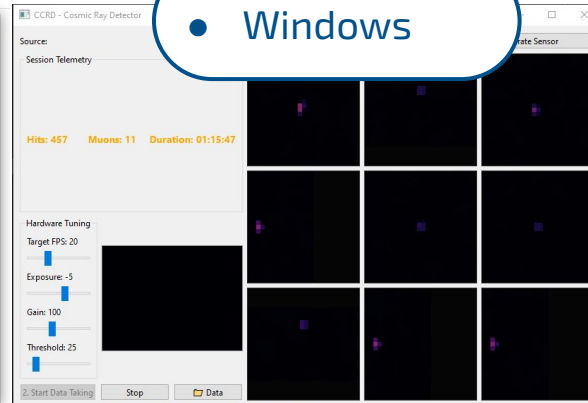
• macOS



• Linux



• Windows



macOS: Significant hurdle due to pre-processed image streams (Denoising).

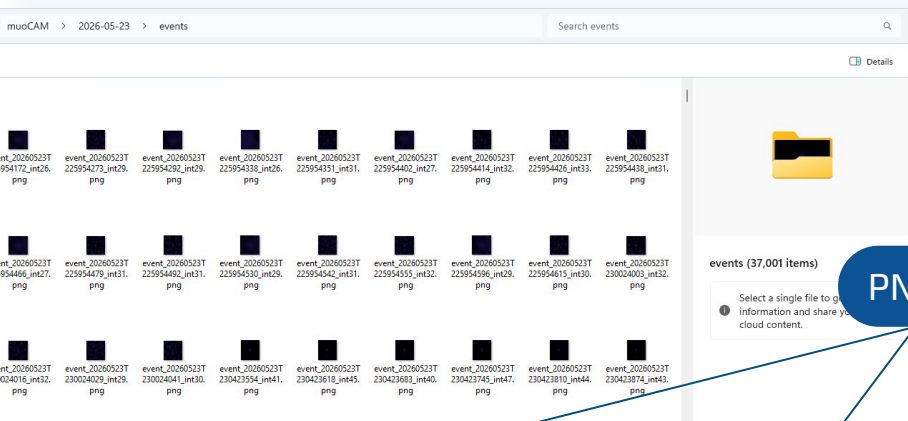
Ubuntu/Linux: Best low-level control via framework **V4L2**

Windows: Stable capture via multimedia framework **DirectShow**; easy deployment

Recommendation: Use external USB cameras to minimize OS-level interference.

Data storage

Saved locally



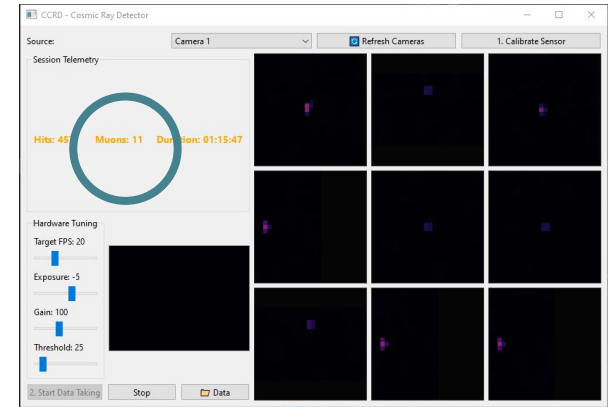
cropped images stored as PNG with:



Experimental Results

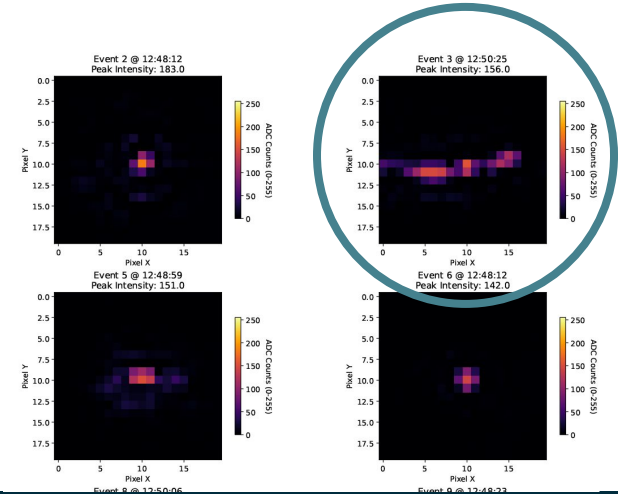
1. Atmospheric Muon Detection

Confirmed muon tracks at **~7 events/hour** under optimized settings (low exposure, high gain, lens covered).



2. Radiation Detection (^{18}F -FDG)

Increased trigger rate near the patient administered ^{18}F -FDG. Interactions detected via a MacBook Air placed next to an FDG-administered patient. Signal bypassed hardware denoising algorithm.

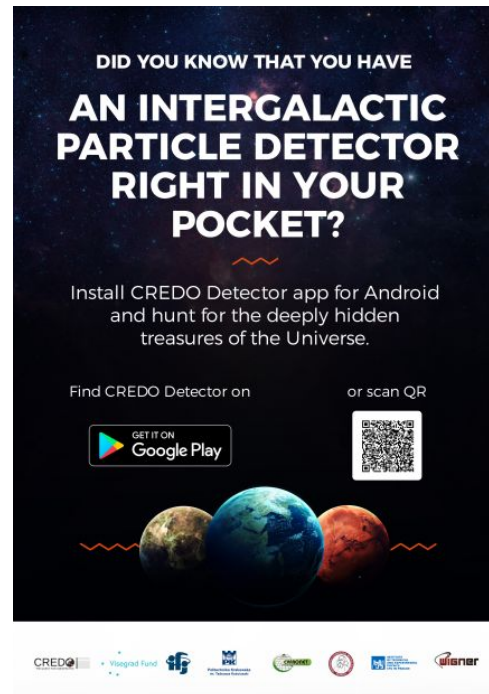


CREDO Collaboration





Start
CREDO-EDU
citizen-science
PROGRAM
for schools in Poland


<https://credo.science/>



DID YOU KNOW THAT YOU HAVE
**AN INTERGALACTIC
PARTICLE DETECTOR
RIGHT IN YOUR
POCKET?**

Install CREDO Detector app for Android
and hunt for the deeply hidden
treasures of the Universe.

Find CREDO Detector on  or scan QR 



CREDO
THE QUEST FOR THE UNEXPECTED

Muographix cooperation



— Outlook & Future Work

Stacked Sensor Coincidence

Two CMOS cameras vertically stacked → temporal coincidence → directional reconstruction & drastically lower false-positive rate.

C++ / Julia Backend

Migrate core image-processing to compiled language → target $\leq 10\%$ CPU, enabling Raspberry Pi deployment.

CNN Classification

Lightweight convolutional network trained on CCRD hits to identify complex topologies — showers, curved tracks — beyond the linearity score.

CREDO Network Integration

Opt-in metadata upload (timestamp + GPS) to CREDO servers → planetary-scale Cosmic Ray Ensemble searches involving desktop detectors.

Summary

Consumer CMOS sensors can detect cosmic-ray muons — no specialized hardware required

Eigenvalue linearity scoring ($L > 0.8$) reliably distinguishes tracks from thermal noise and blobs.

Successfully detected atmospheric muons (~7/hr) and ^{18}F -FDG induced radiation events

Real-time performance: 2.2 ms latency, cross-platform on Windows / macOS / Linux.



Contact



noemi@muotech.io
kate@muotech.io



+48 790377752



muotech.io



Stanisława Moniuszki 11, Rzeszów,
Białoprądnicka 15B, Kraków
Poland

Detection Principle

Noise Subtraction

$$I_{\text{diff}} = I_{\text{frame}} - I_{\text{noise_map}}$$

Linearity Score (Eigenvalue Method)

$$L = (\lambda_1 - \lambda_2) / \lambda_1$$

λ_1, λ_2 = eigenvalues of pixel coordinate covariance matrix, weighted by intensity

Muon classified if $L > 0.8$