LEAPS Integrated Platform Workshop

11-12 May 2021
Zoom
Europe/Zurich timezone

ALBA

Real-time beam detection and tracking from pinhole imaging system based on computer vision with TensorFlow.

Andriy Nosych (ALBA-CELLS, Spain)

Target: automated beam diagnostics routine for pinhole operation

Rockets are machines, but so far they went to space not because of Machine Learning.

Rockets go to space thanks to precise models.

Precise models beat good statistics!

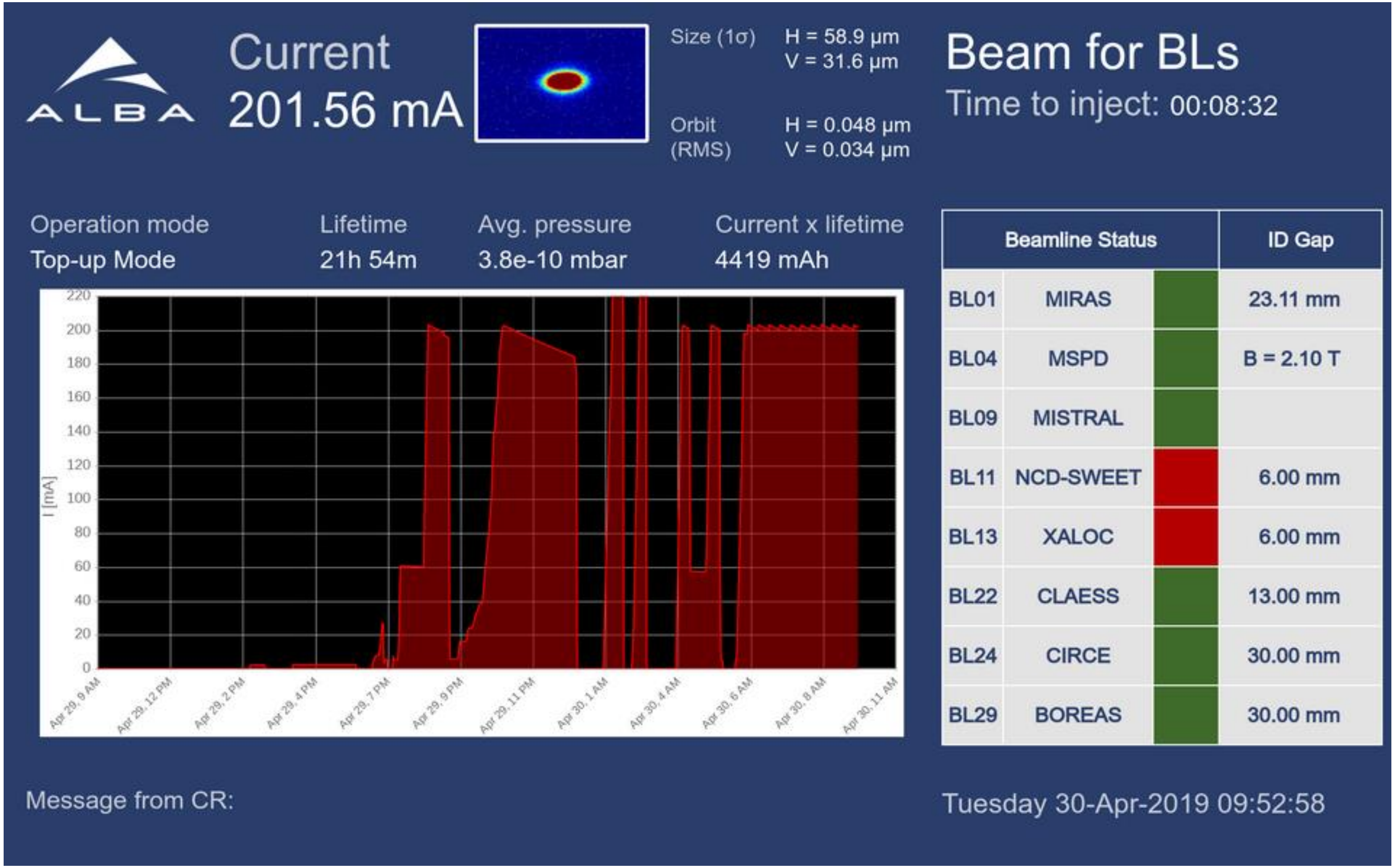
Particle accelerators are also precise machines.
ML is not very common here yet, more auxiliary.

Common ML applications in accelerators:

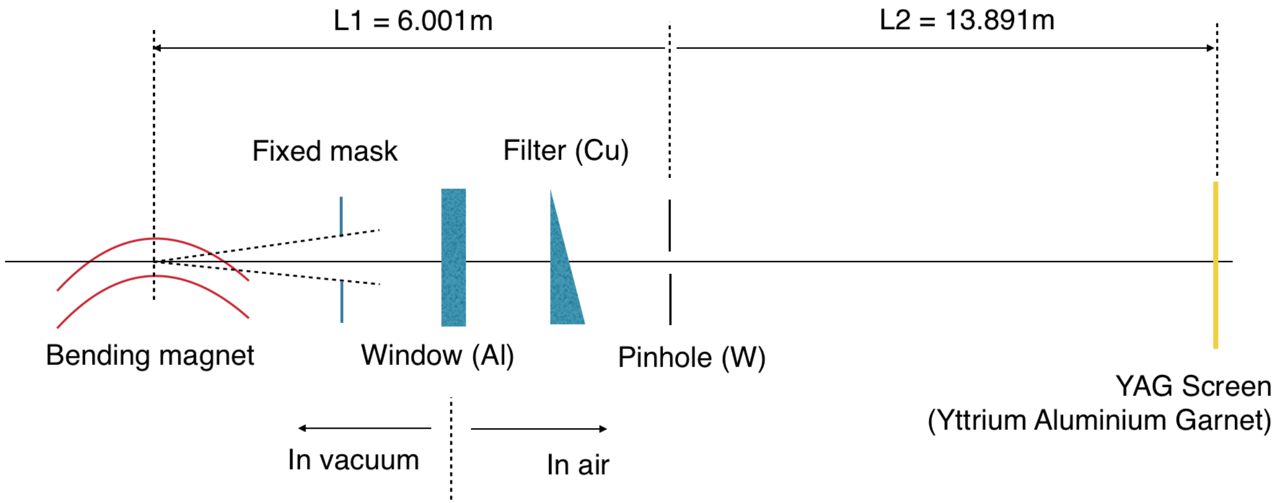
- Design + performance optimization
- Anomaly detection
- Fault prediction
- Models from data

What about accelerator diagnostics and beam dynamics?
Not easy to find an application!

Alba machine status screen: you can always see beam image from a pinhole camera



Pinhole camera system at Alba Synchrotron



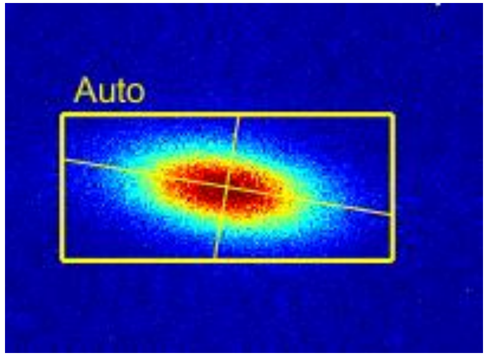
X-ray pinhole camera provides transverse size measurements of the electron beam:

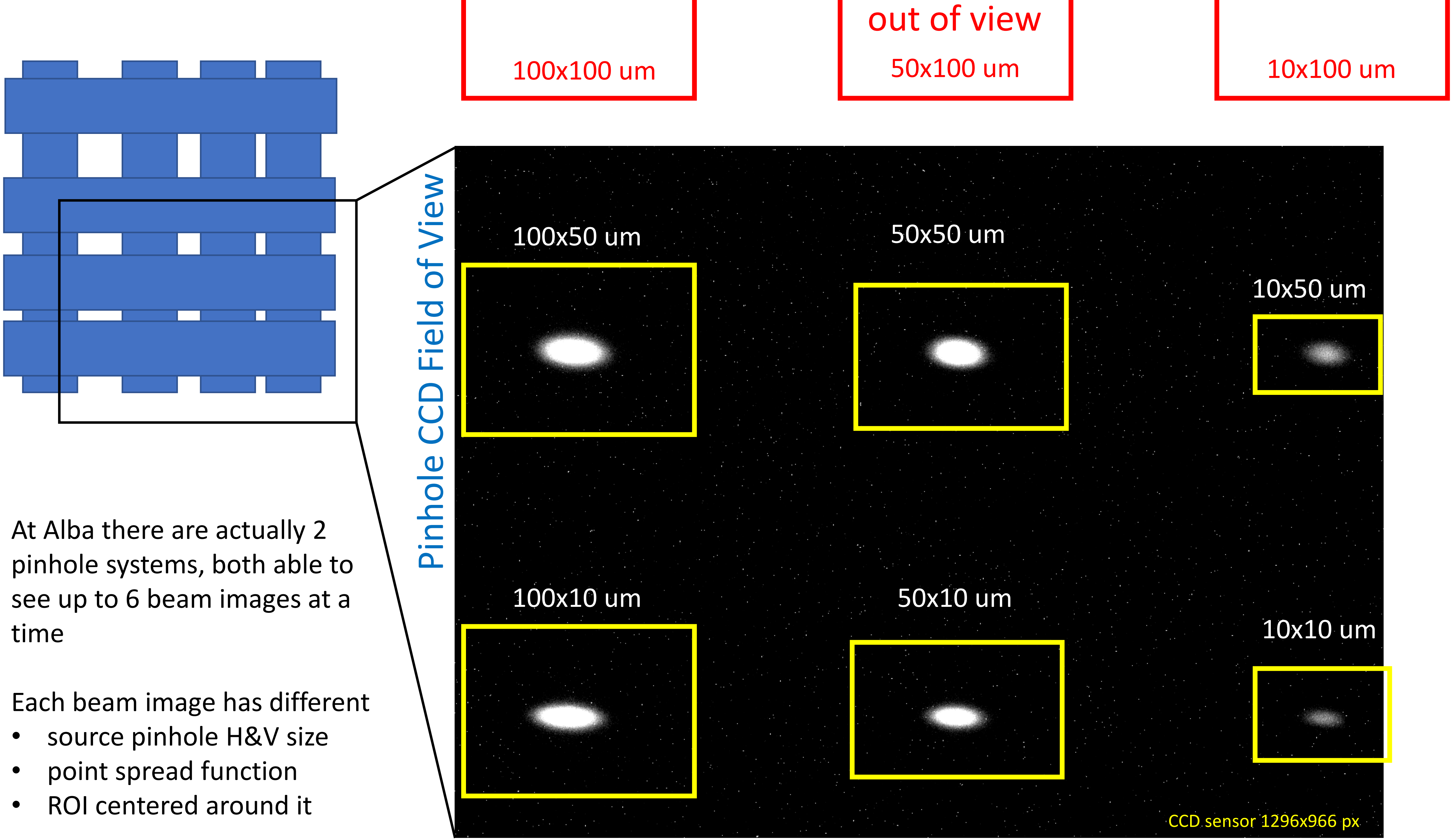
$$\sigma_{YAG=}=(L_2/L_1)\ \sigma_{source}$$

$$\sigma^2_{\mathrm{YAG}} = (X\sigma_{\mathrm{b}})^2 + (\sigma_{\mathrm{PSF}})^2$$

$$\sigma_{\mathrm{PSF}} = \sqrt{\sigma^2_{\mathrm{blur}} + \sigma^2_{\mathrm{DIFF}} + \sigma^2_{\mathrm{screen}}}$$

$$\sigma_{\mathrm{diff}} = \frac{\sqrt{12}}{4\pi}\frac{\lambda L_2}{w} \qquad \sigma_{\mathrm{blur}} = \frac{w(L_1+L_2)}{\sqrt{12}L_1}$$





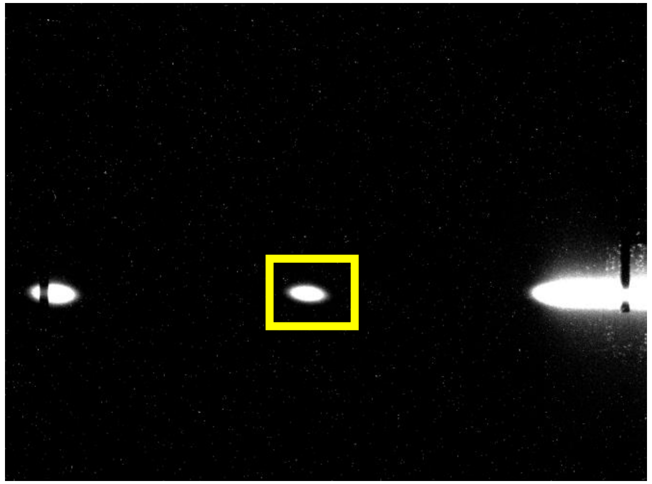
At Alba there are actually 2 pinhole systems, both able to see up to 6 beam images at a time

Each beam image has different

- source pinhole H&V size
- point spread function
- ROI centered around it

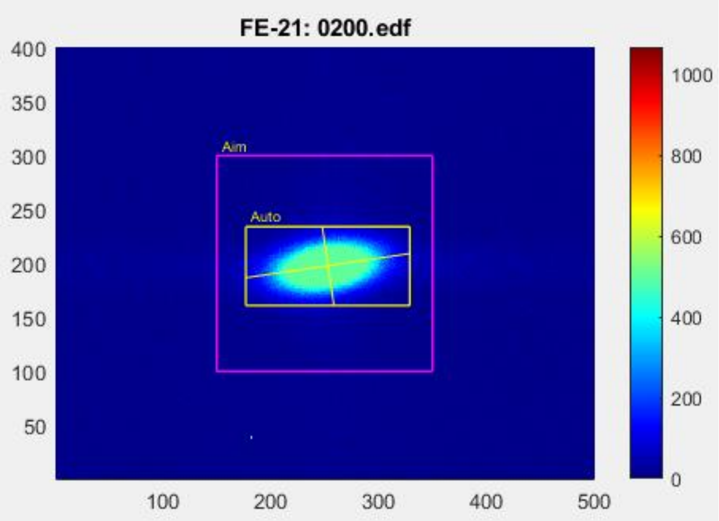
Beam spot parameters are manually fixed for fitting

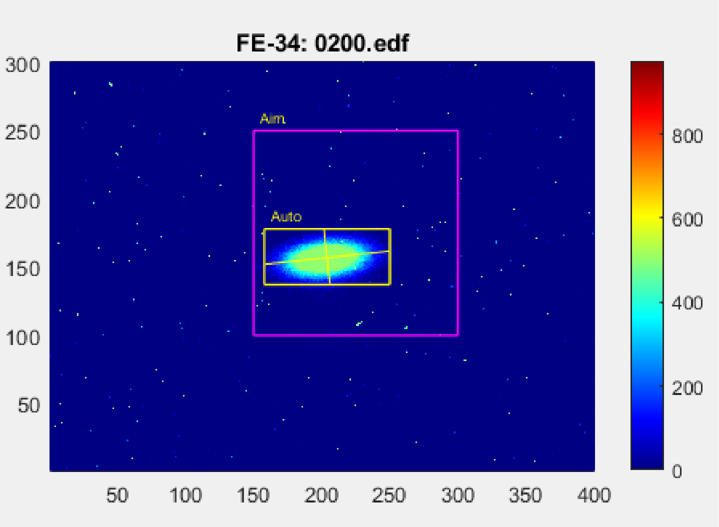
We have to tell the system where to look (ROI), and hope the conditions won't change (e.g. beam moves out or pinhole motors are moved for experiments)



3Hz refresh rate to control system

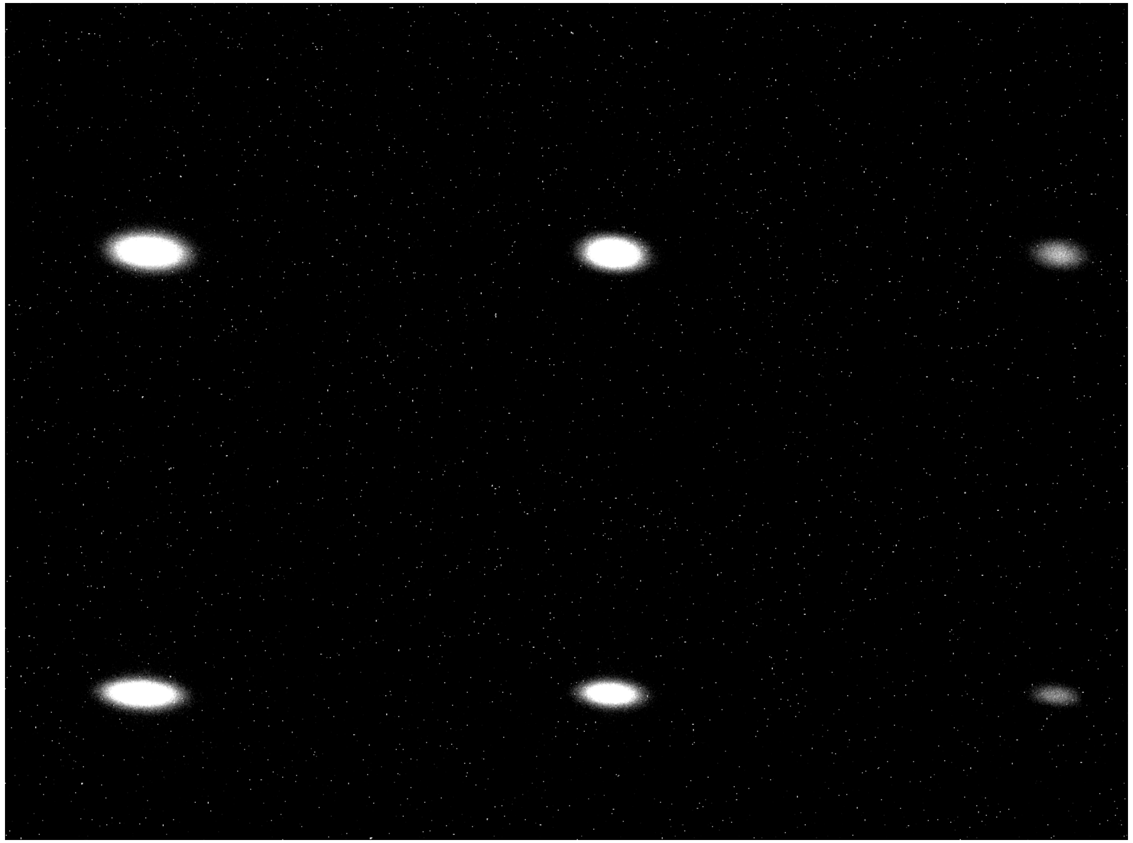
Math analysis within ROIs of both pinholes





==== FE-21 ====	==== FE-34 ====
filename: 0086.edf	filename: 0086.edf
Amp = 1074.42	Amp = 970.28
Bkg = 22.0649	Bkg = 0.0000
X_0 = 254.14 [px]	X_0 = 203.57 [px]
Y_0 = 198.29 [px]	Y_0 = 156.72 [px]
X_0 = 466.38 [um]	X_0 = 594.48 [um]
Y_0 = 323.34 [um]	Y_0 = 460.78 [um]
σ_X (2D) = 56.23 [um]	σ_X (2D) = 55.21 [um]
σ_Y (2D) = 22.33 [um]	σ_Y (2D) = 23.10 [um]
σ_X (2D) = 30.72 [px]	σ_X (2D) = 19.04 [px]
σ_Y (2D) = 13.92 [px]	σ_Y (2D) = 8.16 [px]
σ_X (1D) = 55.61 [um]	σ_X (1D) = 52.98 [um]
σ_Y (1D) = 23.61 [um]	σ_Y (1D) = 23.03 [um]
tilt = 0.147 [rad]	tilt = 0.109 [rad]
tilt = 8.4 [deg]	tilt = 6.2 [deg]
ϵ_H = 5.2985	ϵ_H = 3.6421
ϵ_V = 0.0219	ϵ_V = 0.0197
K = 0.41 [%]	K = 0.54 [%]
Int_{px} = 25.38	Int_{px} = 9.17
Err_H 1D = 197.85	Err_H 1D = 595.05
Err_V 1D = 1182.07	Err_V 1D = 568.26
Err 2D = 4.51e+06	Err 2D = 6.08e+06

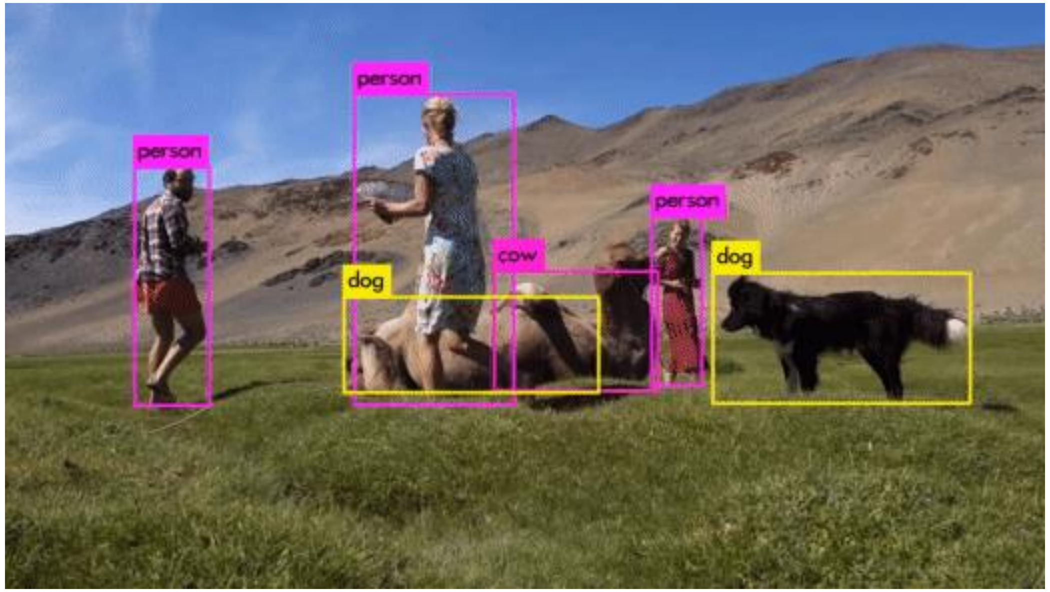
What if there was an automatic system that looks at this image, knows what is what, and what fit parameters to use?



Selecting a ML technique

Structure: Computer vision

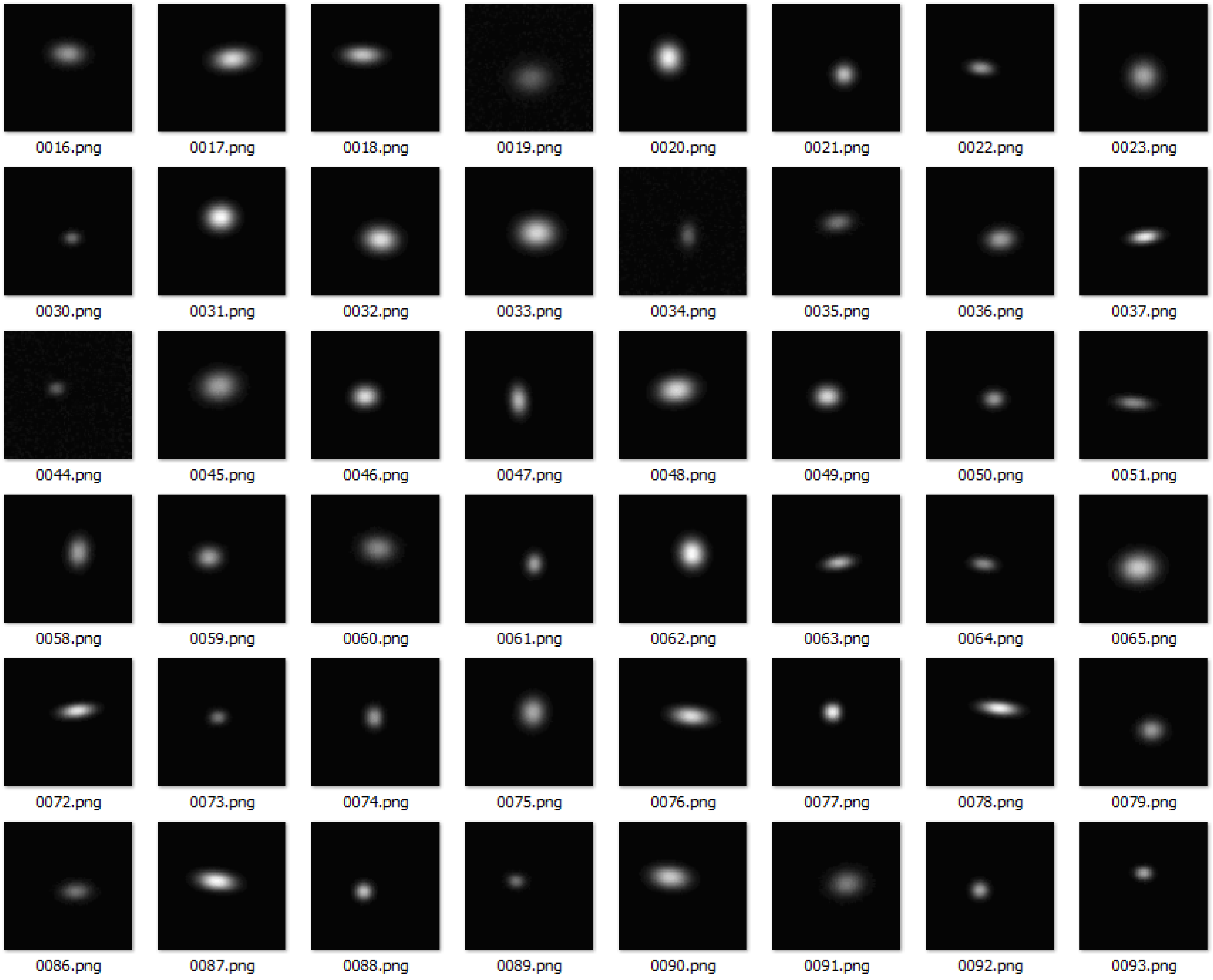
- Learning: **Supervised**
  - Task: **Classification**
    - Architecture: **ImageAI with Tensorflow2.4 backend** [https://imageai.readthedocs.io]
      - Algorithm: **YOLOv3** [https://pjreddie.com/darknet/yolo/]



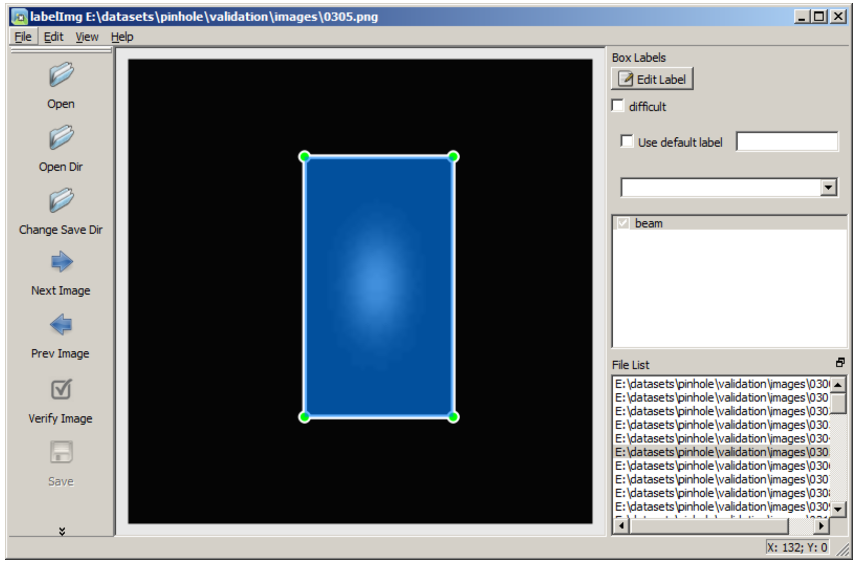
YOLO: You Only Look Once

A real-time object detection system, based on convolutional NN which looks at the whole image and predicts bounding boxes with classifiers

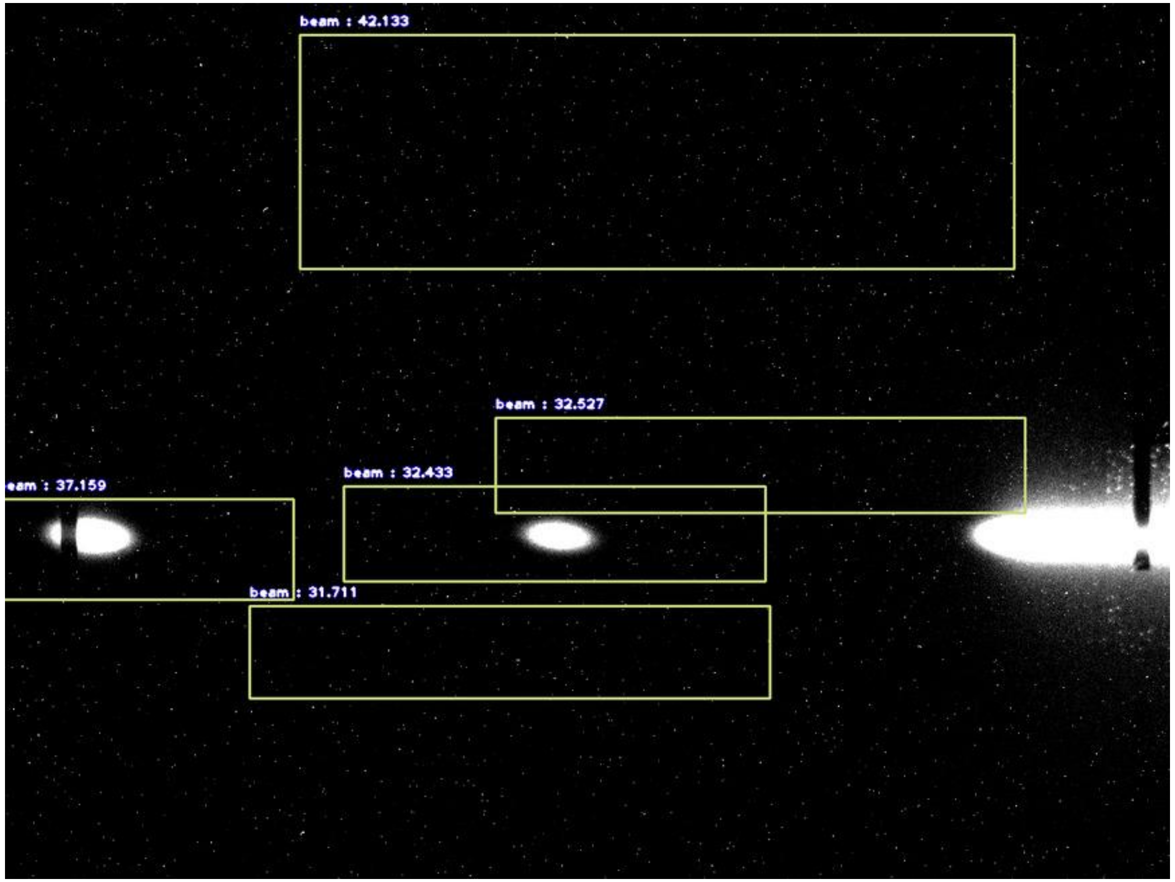
Models trained on 200-1000 randomly generated Gaussians with annotations



Annotations automatically generated and verified with labelImg.py

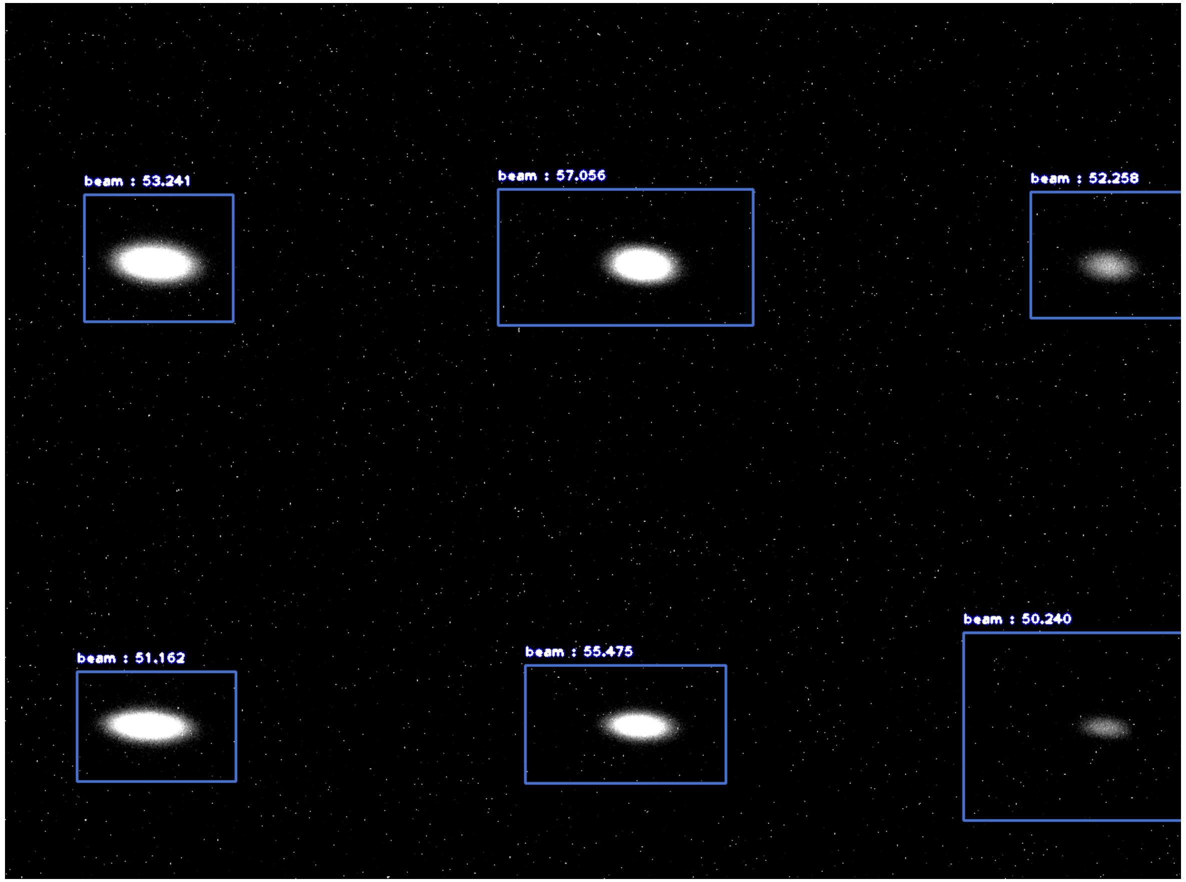


When learning goes wrong



8 epochs
1000 training images 72x72 px
Time of each epoch: 6h (3GHz CPU)
Wrong hyperparameters?
Not enough training epochs?
Insufficient/wrong learning data?

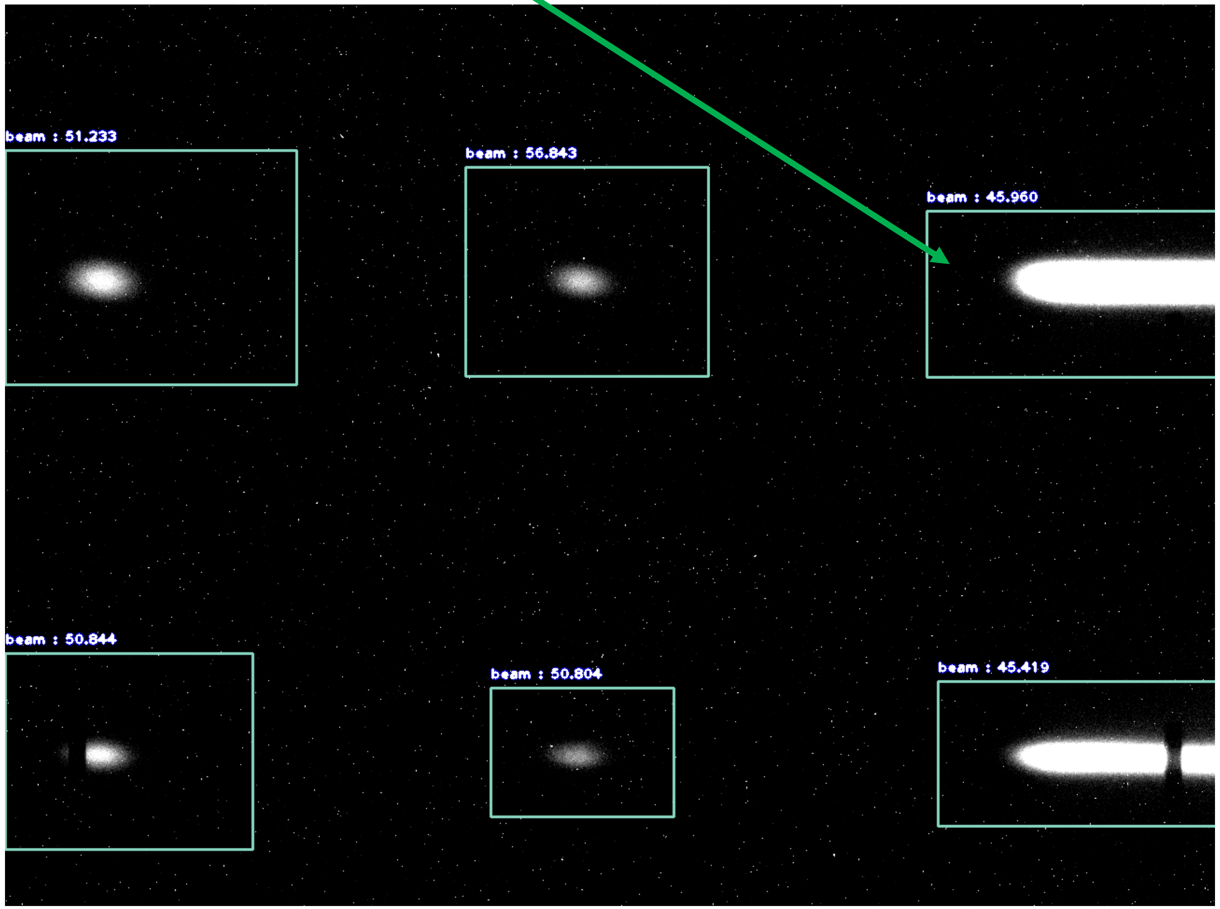
Exam passed

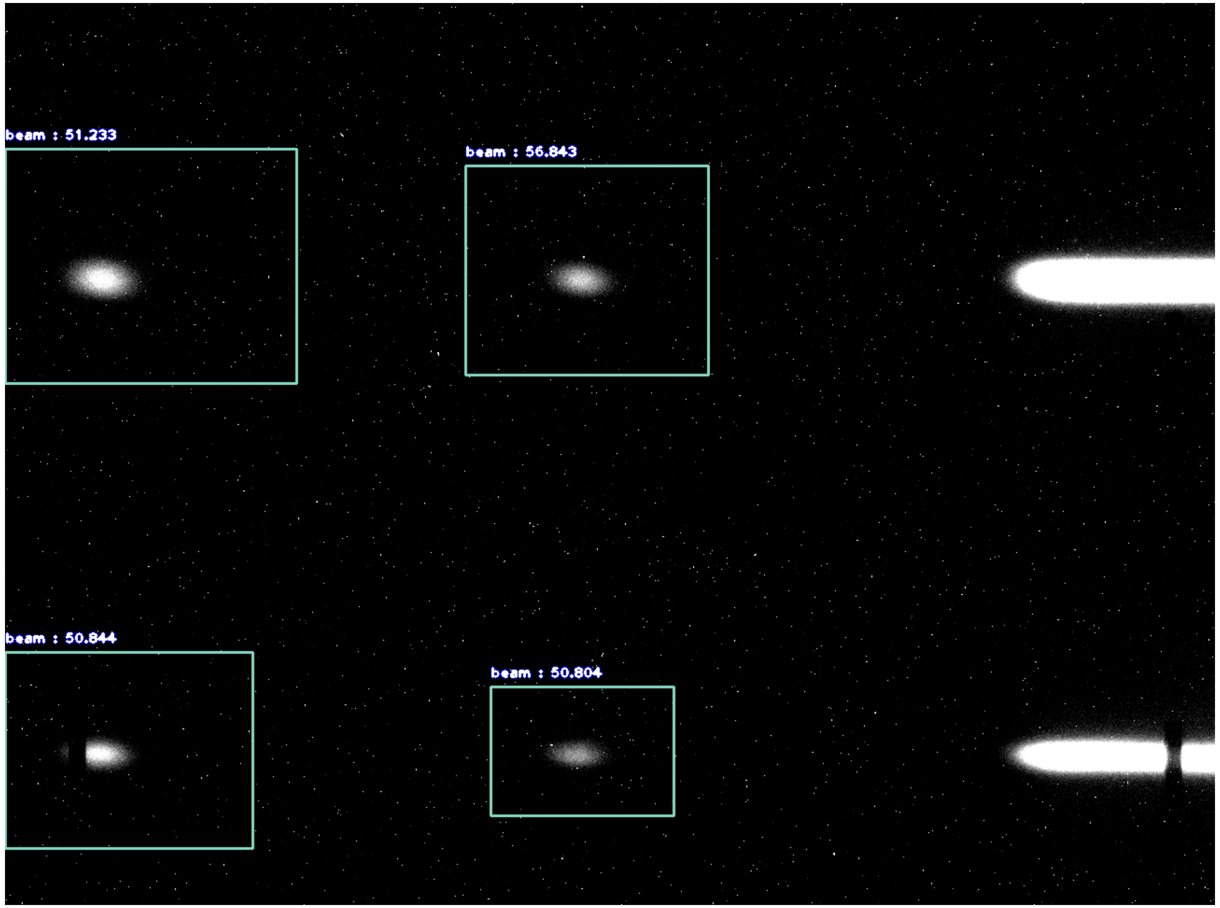


23 epochs
300 training images 100x100 px
Time of each epoch: 2h (3GHz CPU)
Model size: 250 Mb

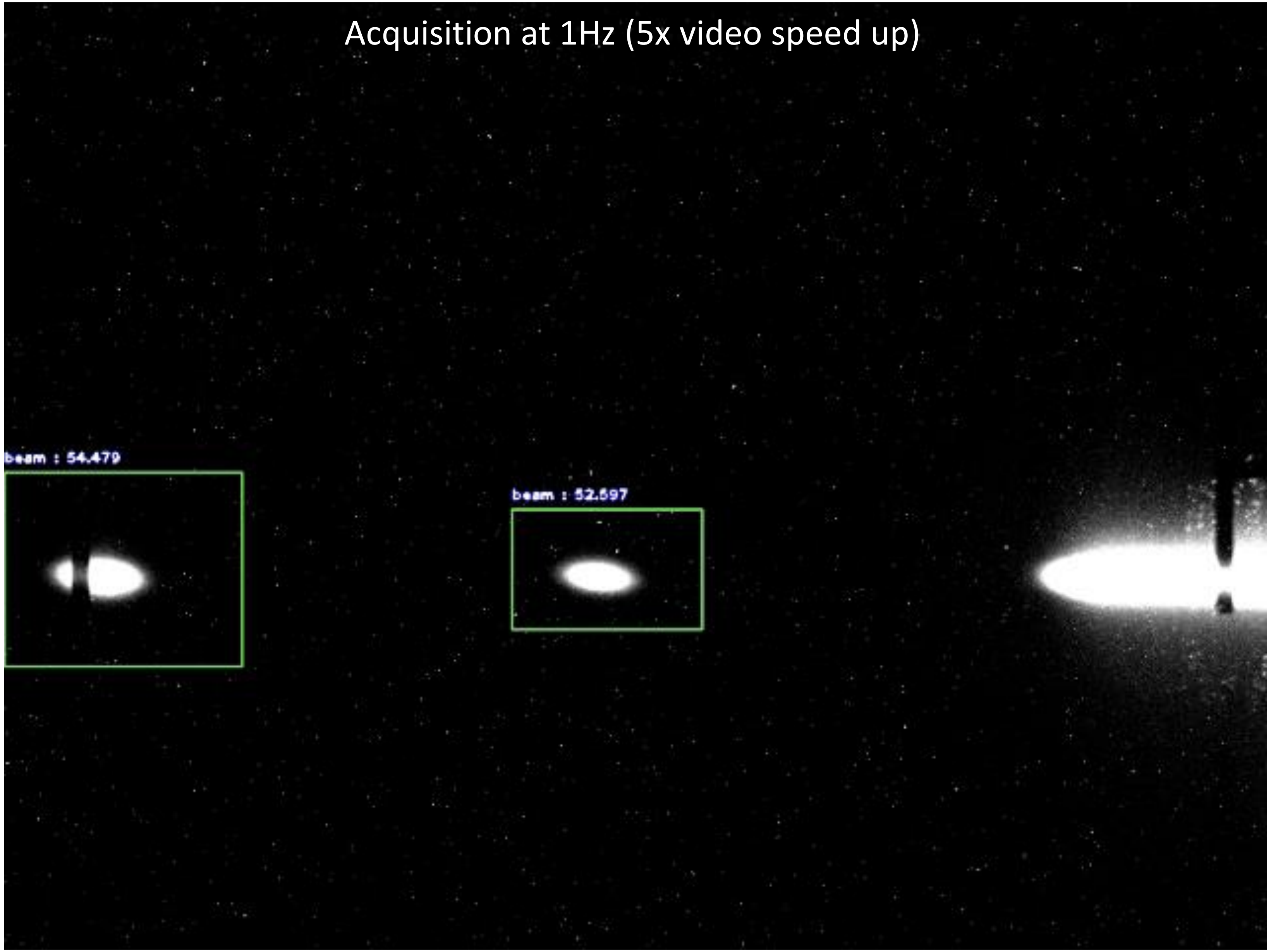
A small detail

Residual beam fan: unusable. What to do with it?





1) Ignore using Confidence score threshold
2) Train NN on another object class “Fan” (not tested)



Current AAN performance:
1.6 FPS detection speed on a laptop
250 Mb model size
Trained on single object ("beam")

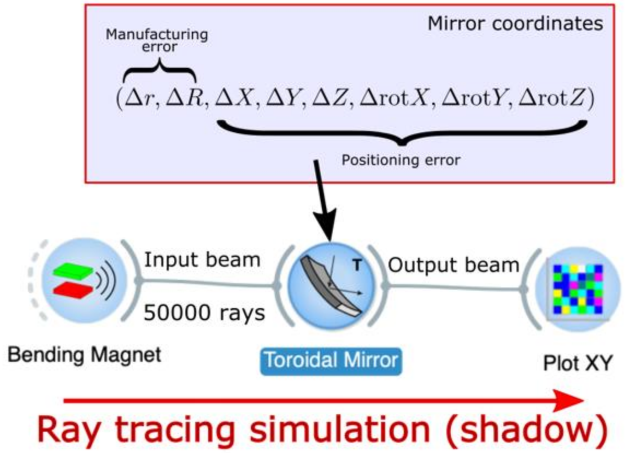
Questions to ANN
- How many Epochs do you really need?
- How to further optimize the confidence score?
- Better to train on "fan" object as well?
- ROI sometimes off, why?
- Can you detect faster?
- Can you have a smaller model size?

Other ML projects done at Alba Synchrotron

MINERVA Beamline (Dominique Heinis)

Machine learning to model the behavior of a mirror subjected to manufacturing errors and misalignments (scikit-learn)

- learning using linear model (with polynomial features) to take into account optical aberrations and nonlinearities
- model calculates quickly the output beam (based on tensor product instead of ray tracing)
- model can be used to retrieve an analytical formula
- model allows numerical optimization (can be easily integrated in a steepest gradient like method)
- for the moment just one optical element but there is no theoretical constraints to extent it to a complete beamline



Beam Physics (Zeus Marti, Emilio Morales)

ANN to solve the measured orbit response matrix fit to avoid using the model (TF, Keras).

It does not work, it has too many elements (88x120x4) and knobs (112) and is too non linear, the training data is huge.
Instead it was possible to fit the inverse orbit response matrix, since it has much less elements (432) for the same number of knobs and it is much more linear.
It turned out to be so linear that a simple linear fit behaves as well as an ANN for the present precision of our measurements.

Conclusions

If NN works - it’s a huge advance.

Advantages to date:
All tools , tutorials, guides, libraries, algorithms are free.

Bottlenecks to date:
Large time investment to tune and train the model.
Large hardware investment.
Quantity matters (CPU->GPU->HPC).

Machine training is long
Machine thinking is hard
(Great opportunity for student projects)