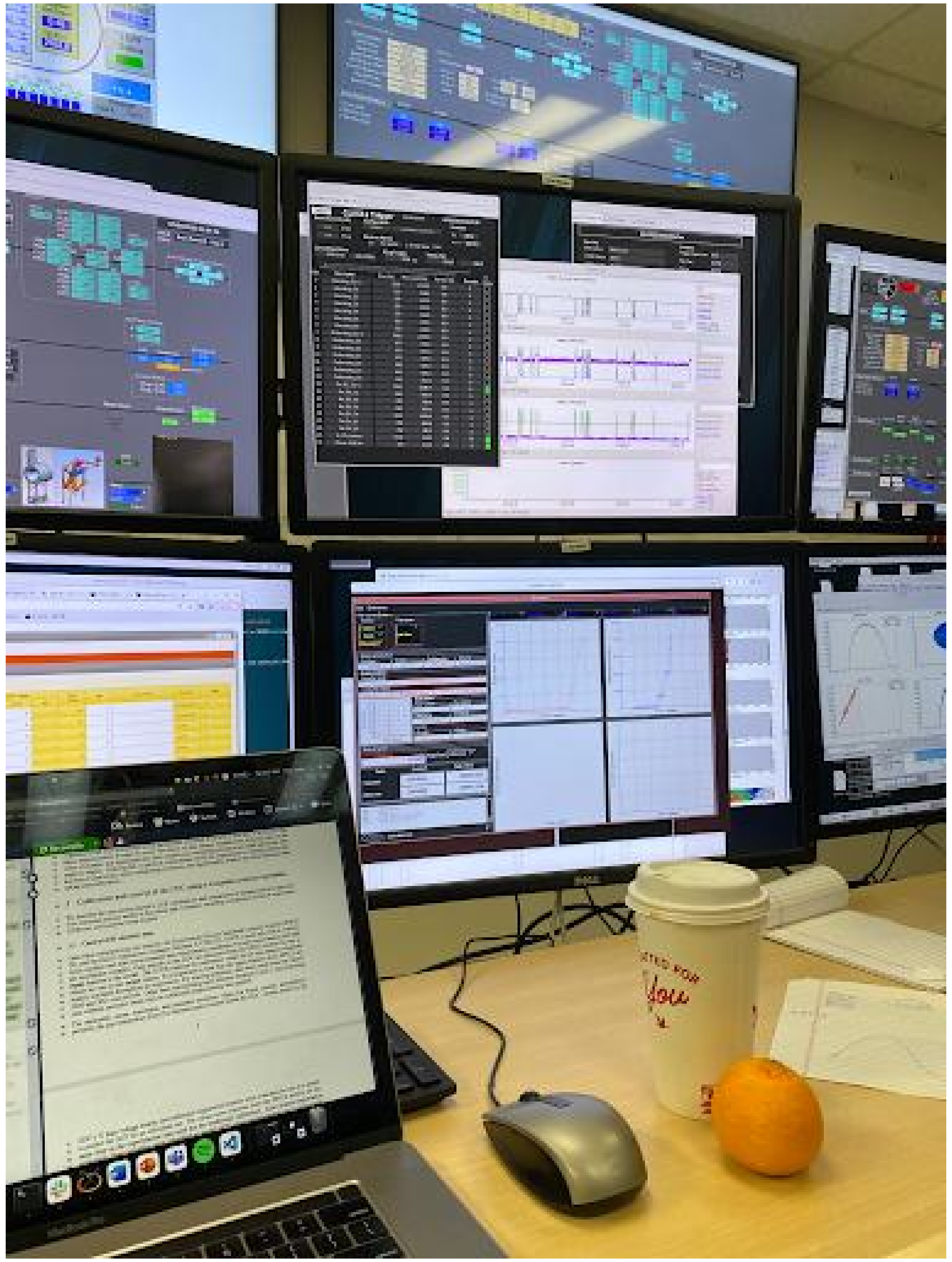
Jefferson Lab

HYDRA

Computer vision for data quality monitoring

hydrateam@jlab.org



Hydra: 10,000 ft view

Monitoring Workflow

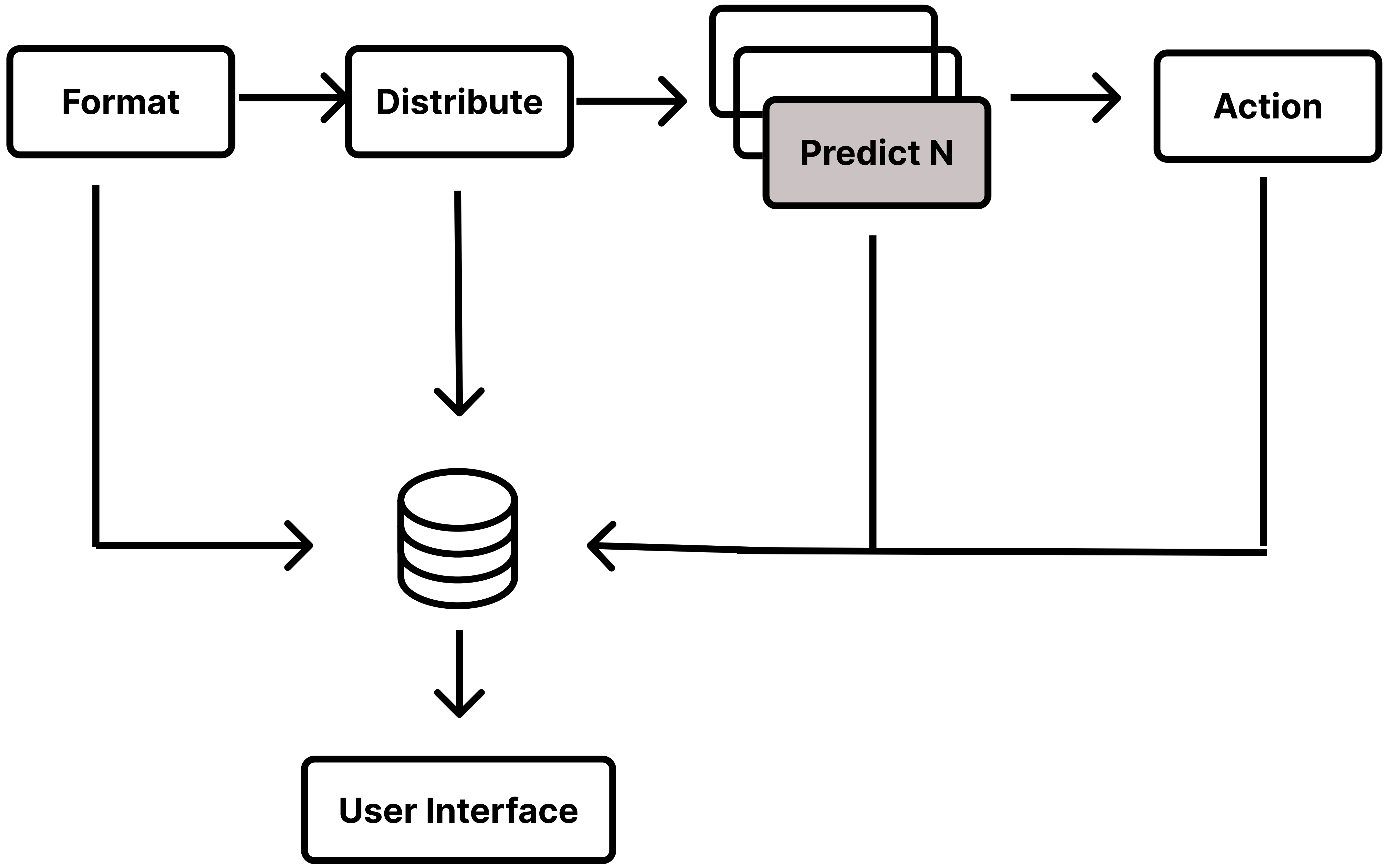
Python based modules responsible for managing images, classification, and performing actions (alarming, logging, etc)

Back End

Comprehensive MySQL database serves as communication and storage between system and front end

Front End

Web pages provide users with actionable insight *from anywhere* without requiring technical knowledge of the system or AI



Summary Statistics

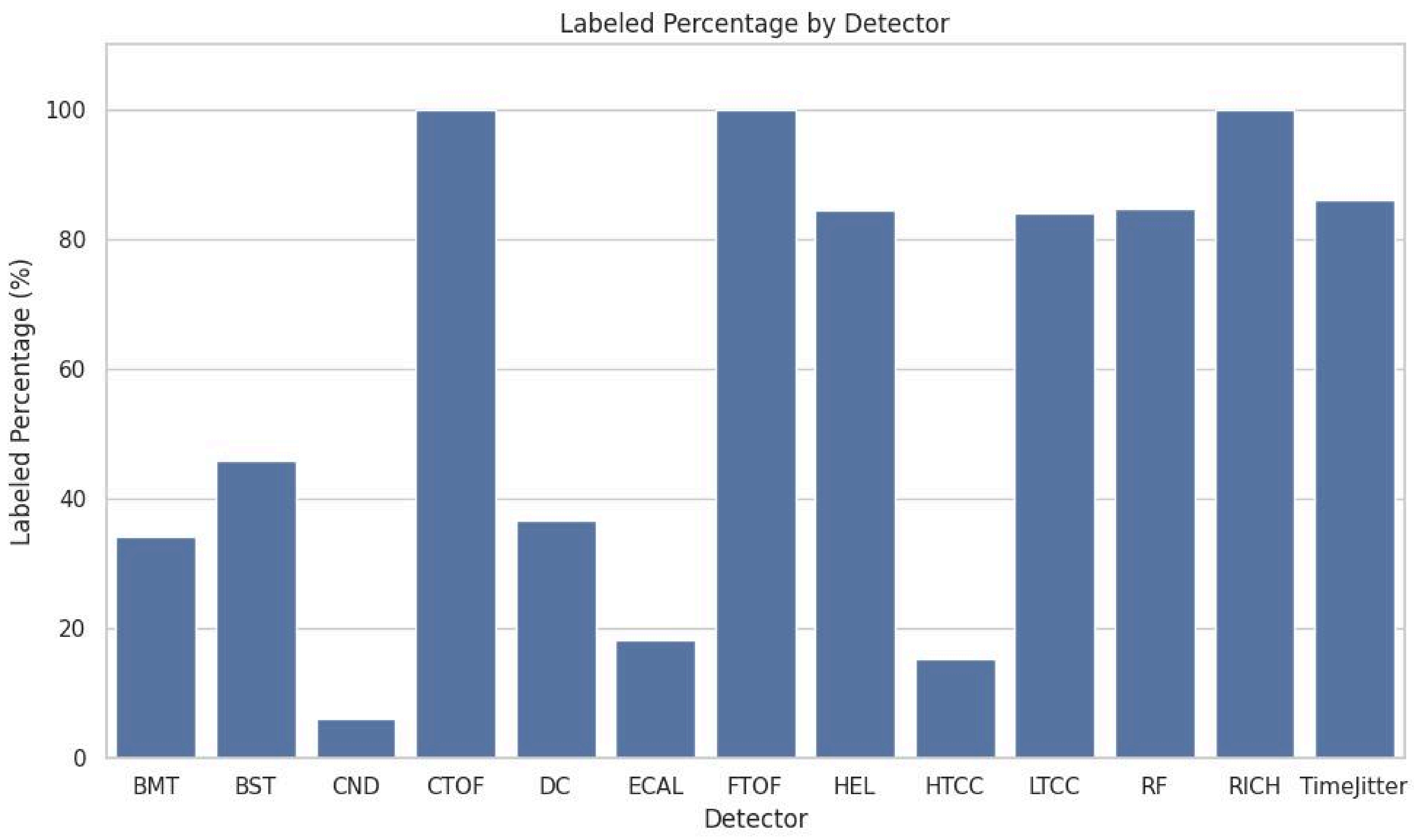
	online				*offline*		
	Hall A	**Hall B**	Hall C	Hall D	mini-Hydra (GlueX)	mini-Hydra (Hall C)	ePIC
Labeled Images	5,440	**237,825**	362	140,517	350,683	144,392**	340,644**
Plot Types	42	**73**	16	15	46	39	568
Active Models	0	**41**	0	6	-	-	-
Batch / frequency	-	**73 / 3-5min**	-	15 / min	-	-	-
Runs with 1 bad image	-	**1,019**	-	1,035	-	-	-
Total analyzed	-	**1,199,771**	-	1,101,160*	-	-	-

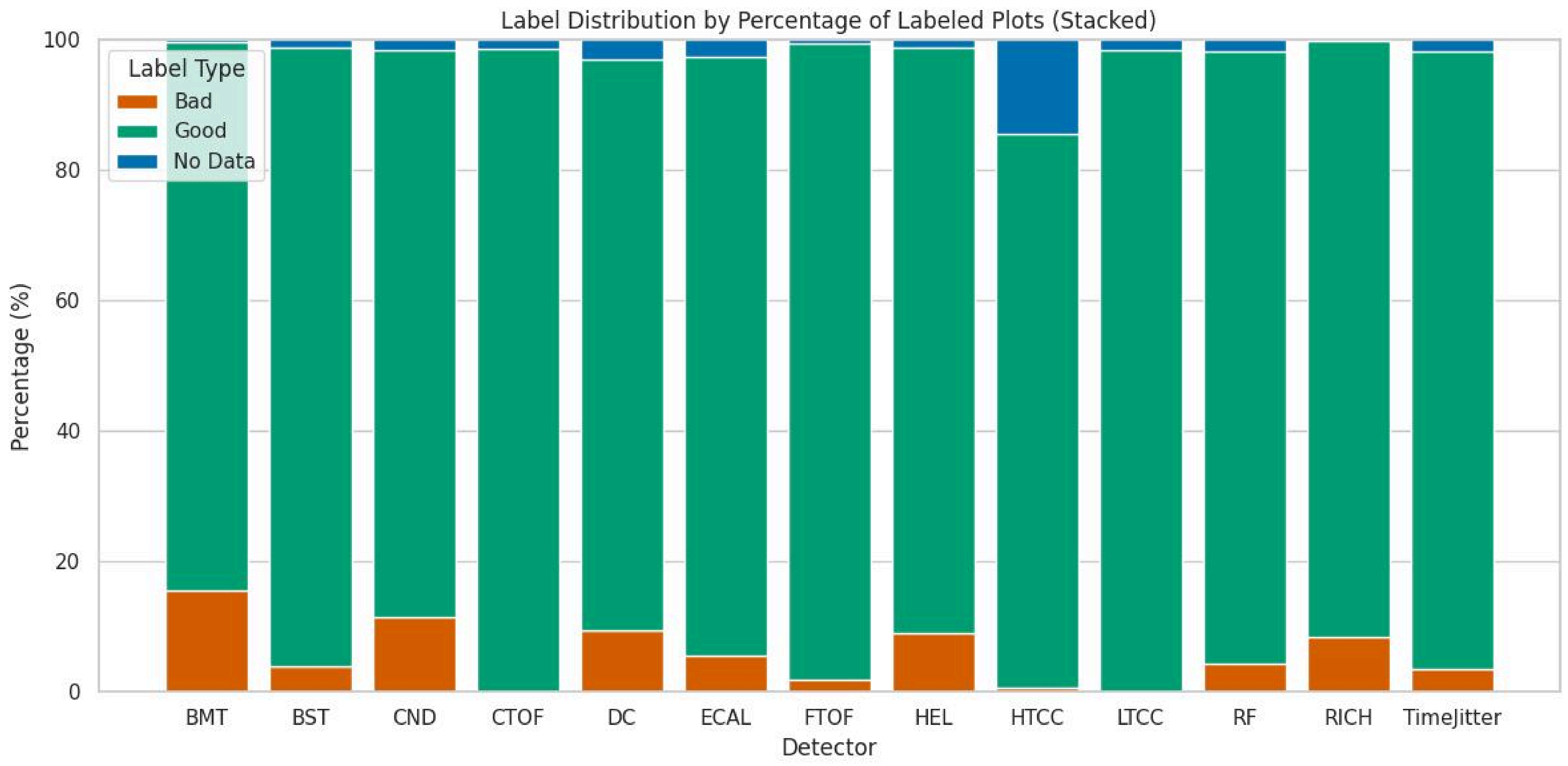
*since record keeping began

** Hydra is just tracking these images

Labeling Statistics

237,825 total labeled images





% labeled *chunked* images per plot type

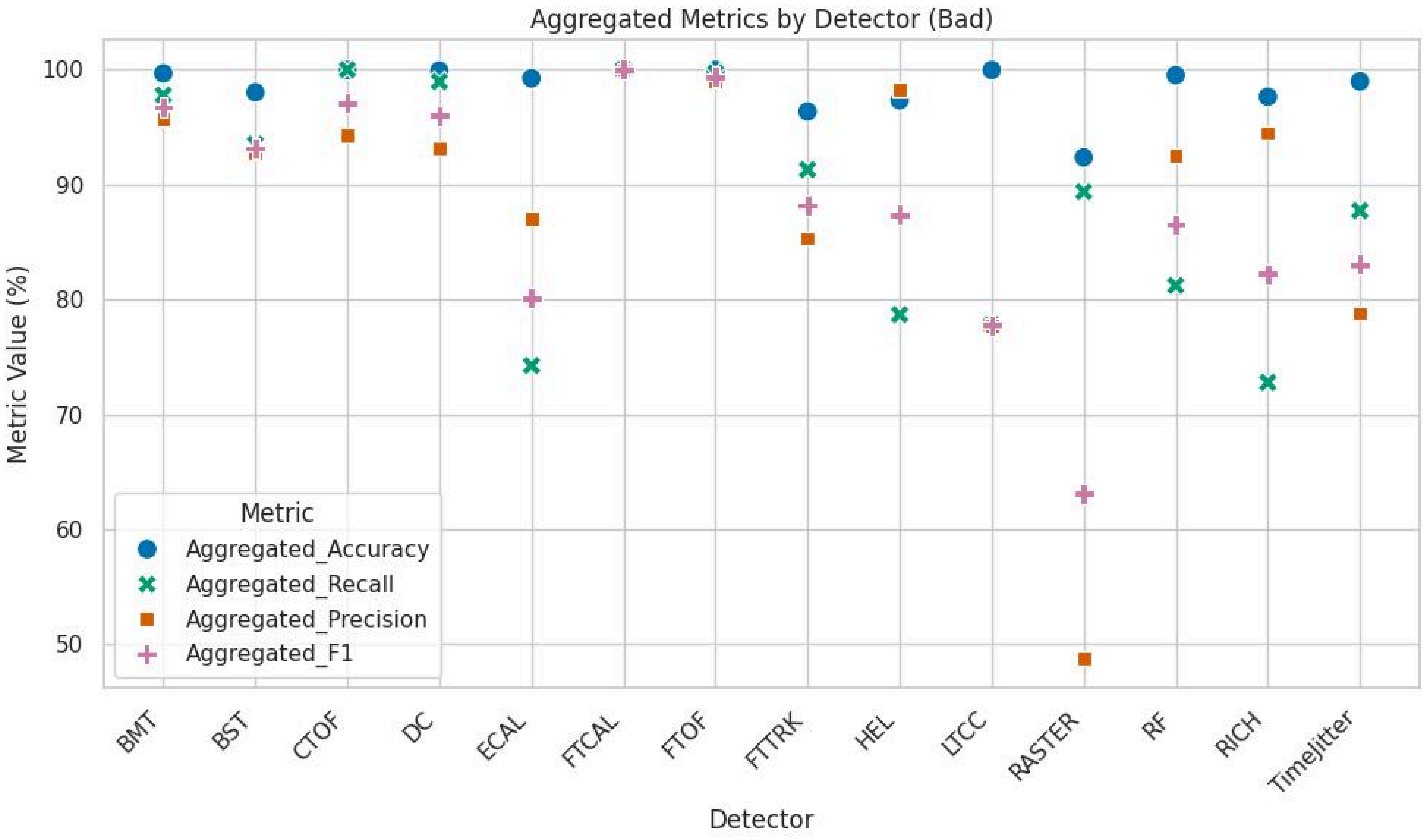
Hydra has a separate instance of mon12 that produces images every 3-5 minutes.

Label distribution by plot type

We typically have a large class imbalance. We mitigate this by difference sampling methods used in training models.

Performance Metrics

Aggregated metrics for each subsystem ***for currently labeled Bad images***



Accuracy = (TP + FN) / (TP + FP + FN + TN)

Recall = TP / (TP + FN)

Precision = TP / (TP + FP)

F1 Score = 2 x (Precision x Recall) / (Precision + Recall)

Why do we want all the labels all the time?

Performance metrics

We want to measure and track the accuracy of models over time

How accurate are these models? I cannot calculate accuracy on new images without your labels.

Interpretability

Heat maps are usually easier to interpret when models are well trained

Localized heat maps can help us determine what the model is learning.

Continual Learning

With frequent labeling, we can track a running accuracy and trigger on retraining or other corrective actions.

This is valuable especially when our monitoring images change with new experiments and/or configurations.

How are images sent to the labeler?

Randomly sample

A very small sample of images are randomly selected for labeling.

All bad

All bad images are sent to the labeler for human confirmation.

All unconfirmed

If Hydra is not sure (output weight below threshold) about an image, it will be sent for labeling.

Better models send less images to the labeler!

Performance improvement with frequent labeling

In general, training set size increases with increasing Model ID

Accuracy = (TP + FN) / (TP + FP + FN + TN)

Recall = TP / (TP + FN)

Precision = TP / (TP + FP)

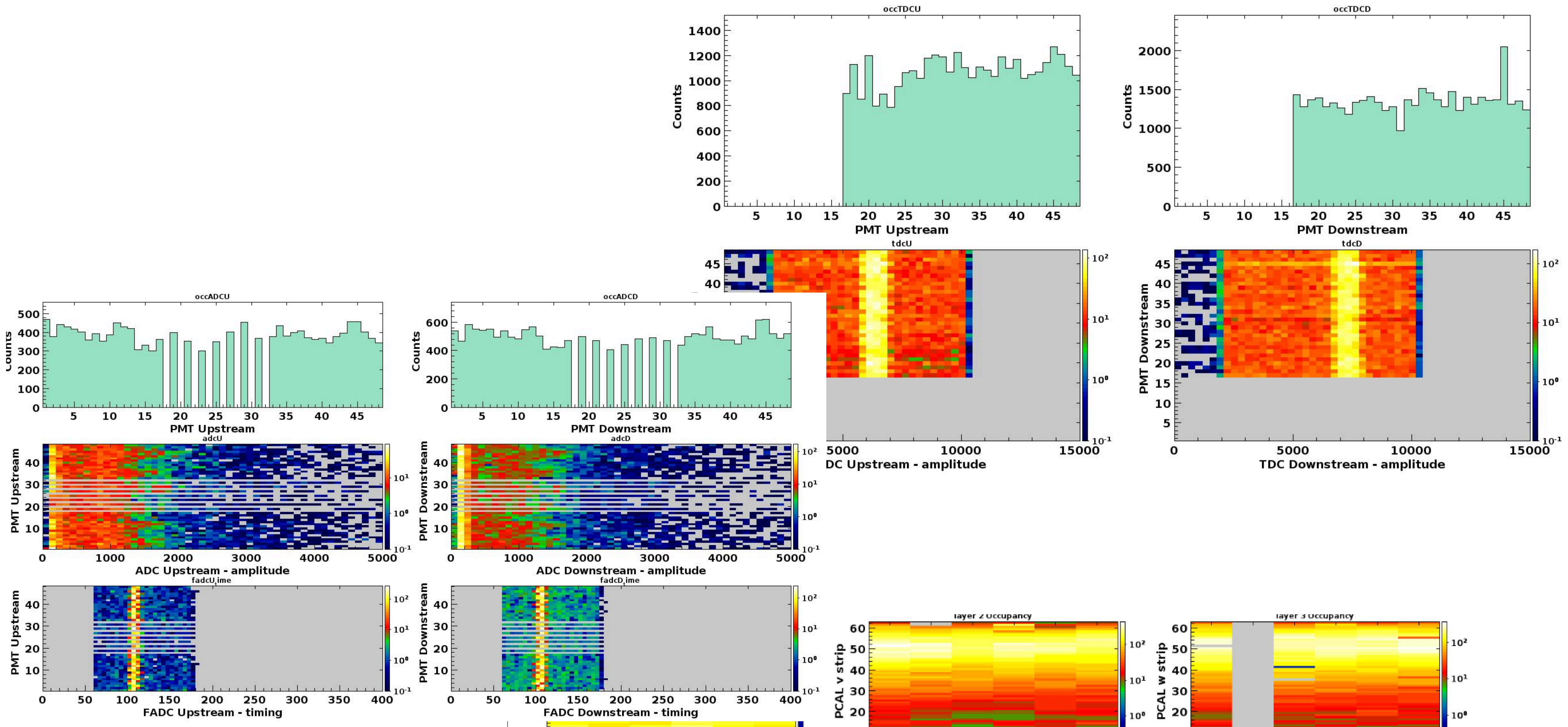
F1 Score = 2 x (Precision x Recall) / (Precision + Recall)

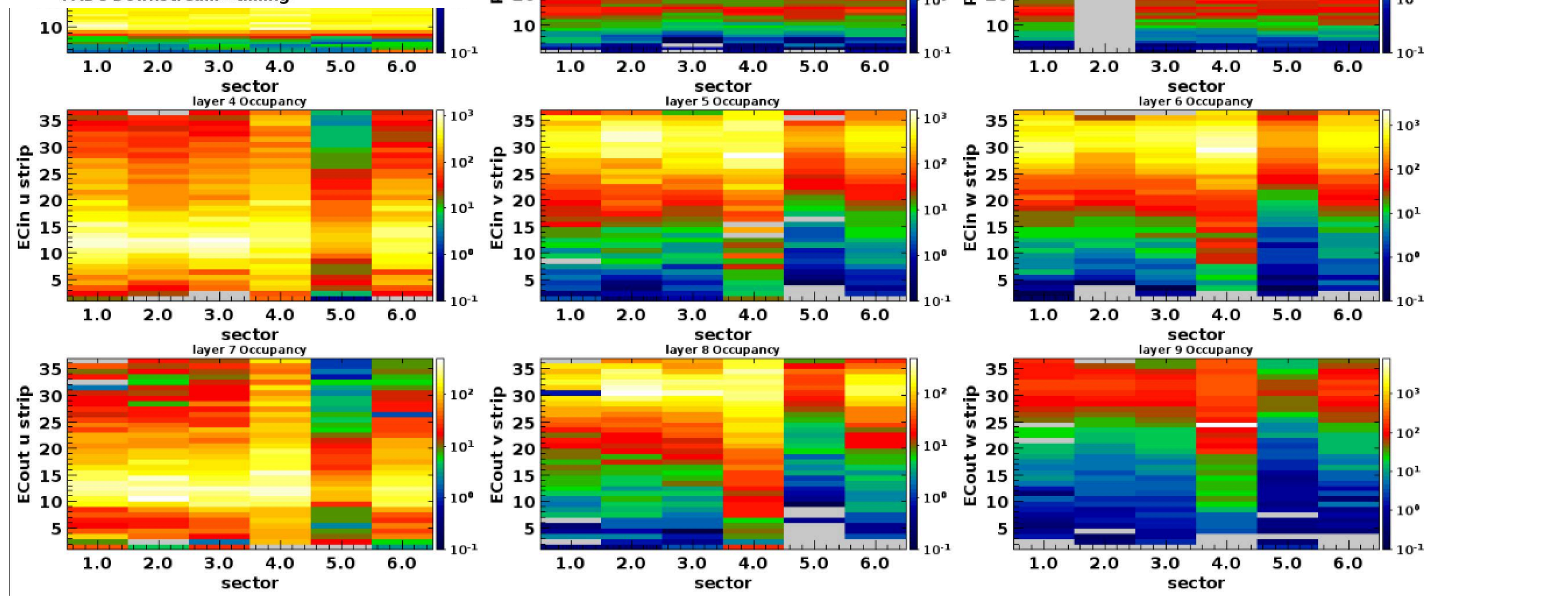
Images with no Bad labels

If there are no examples of a “Bad” image, we drop that classification in training. This means the specific model will not be able to say an image is bad when we are running.

Affected Plot Types:
CTOF_tdc,
FTOF_adcEnergy s1, s2, s5
FTOF_adcTime s1, s2, s5
FTOF tdc s1, s2
ECAL_tdc_s2,
HTCC_adcEnergy

Darren Upton





We can simulate bad plots to use in training.

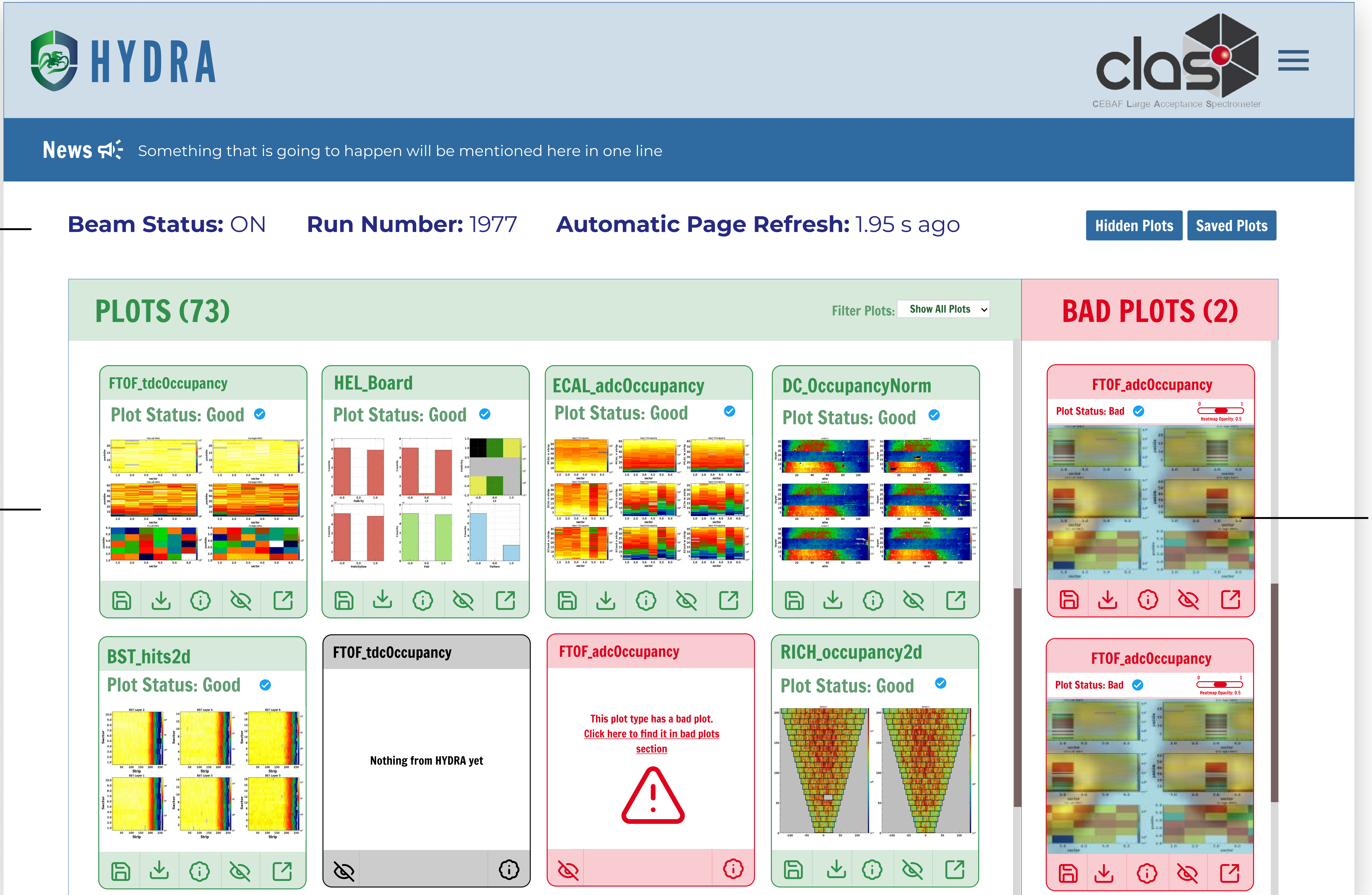
We redesigned the web interface.

All details about the user interface and experience would be a complete separate talk, so this is just a highlight.

Run

status indicators

Main image gallery



reserved for Bad plots

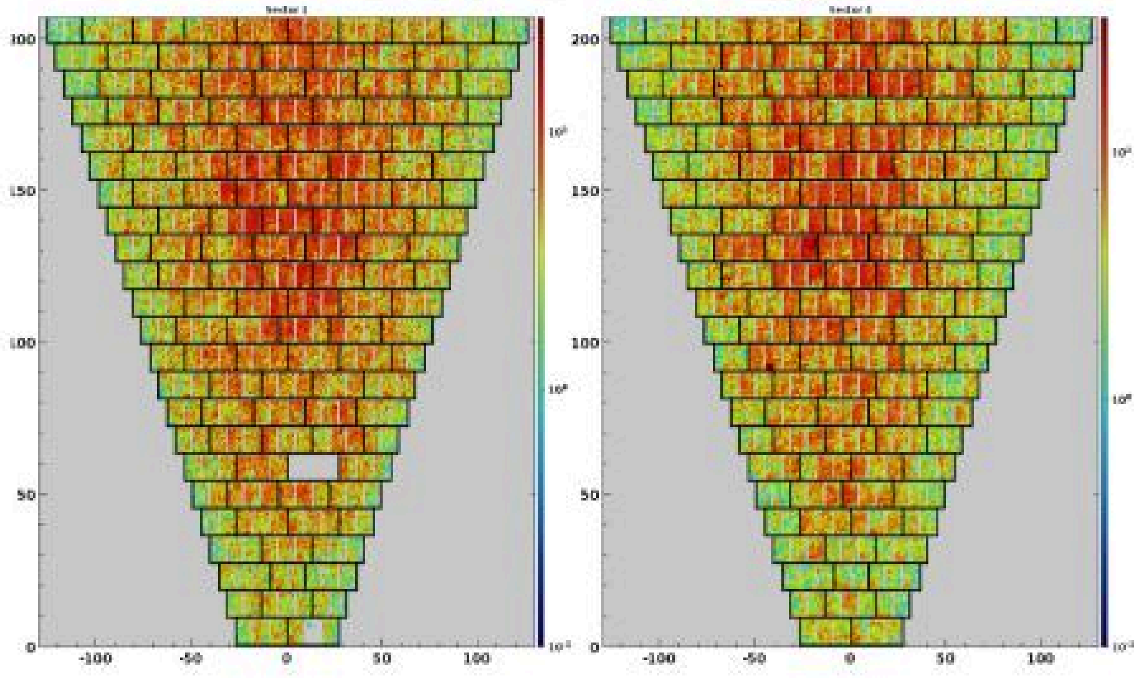
GradCAM Heat Maps

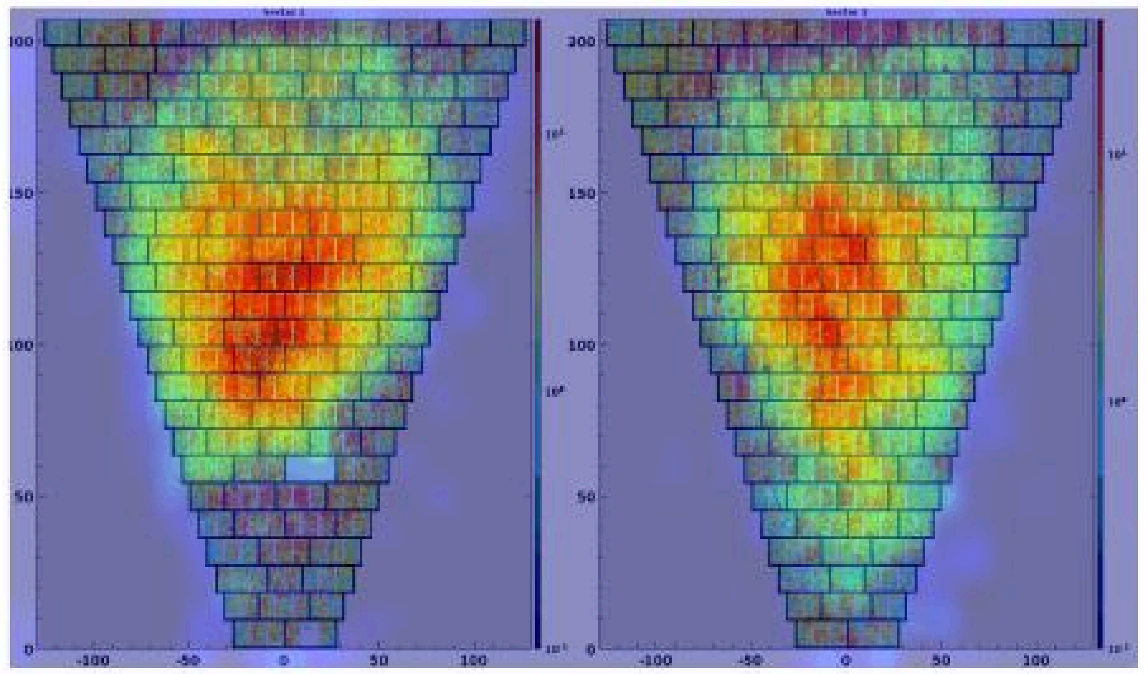
“Why did Hydra say this was bad?”

Bright spots indicate important regions of the image for the given classification.

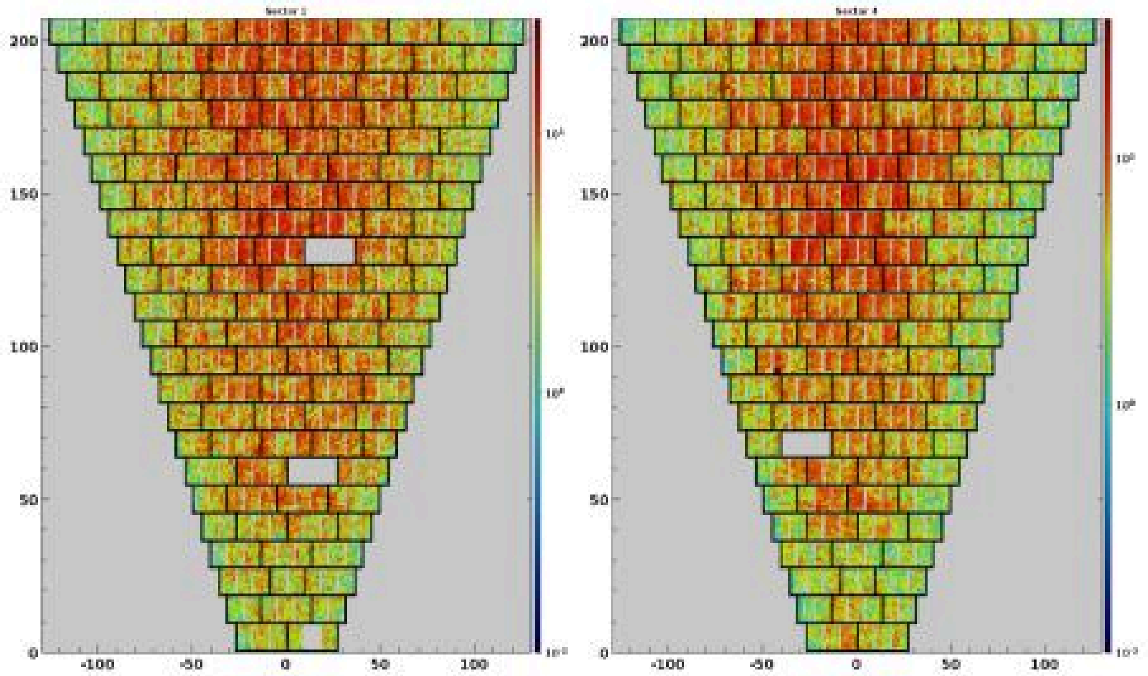
These are very sensitive to how well the model is trained.

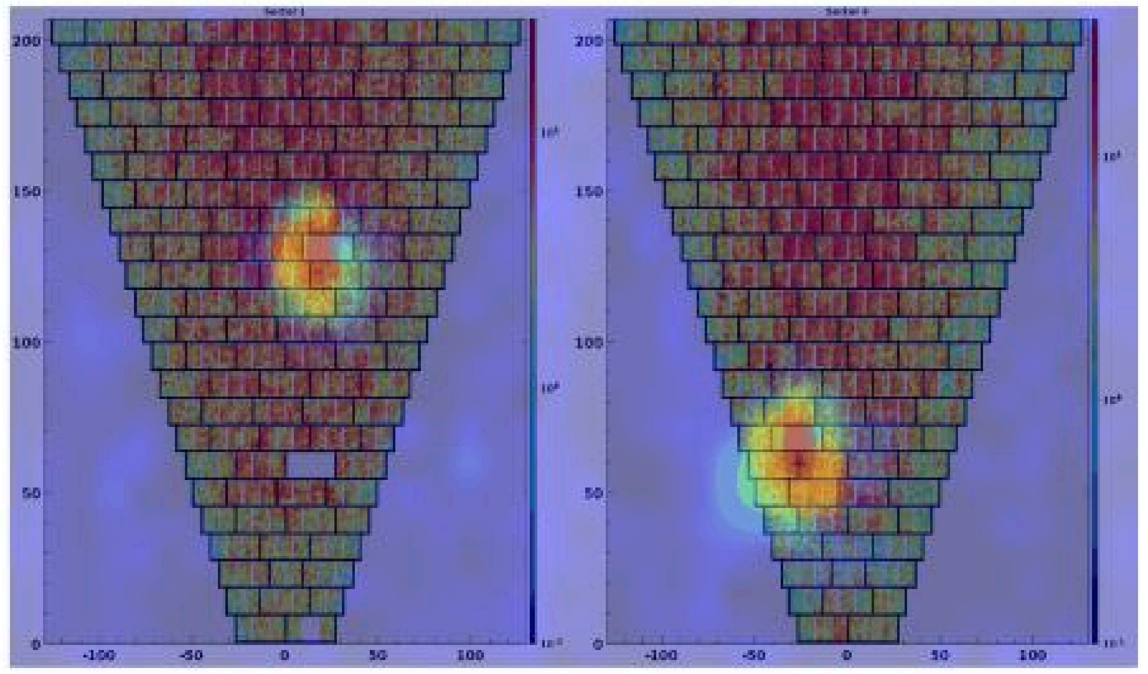
this is a normal image





this is a bad image





Heat maps are produced from mixed layers in InceptionV3+CBAM

Image Labeler

Plot Type Selector

Images to label

Front-end

design ongoing

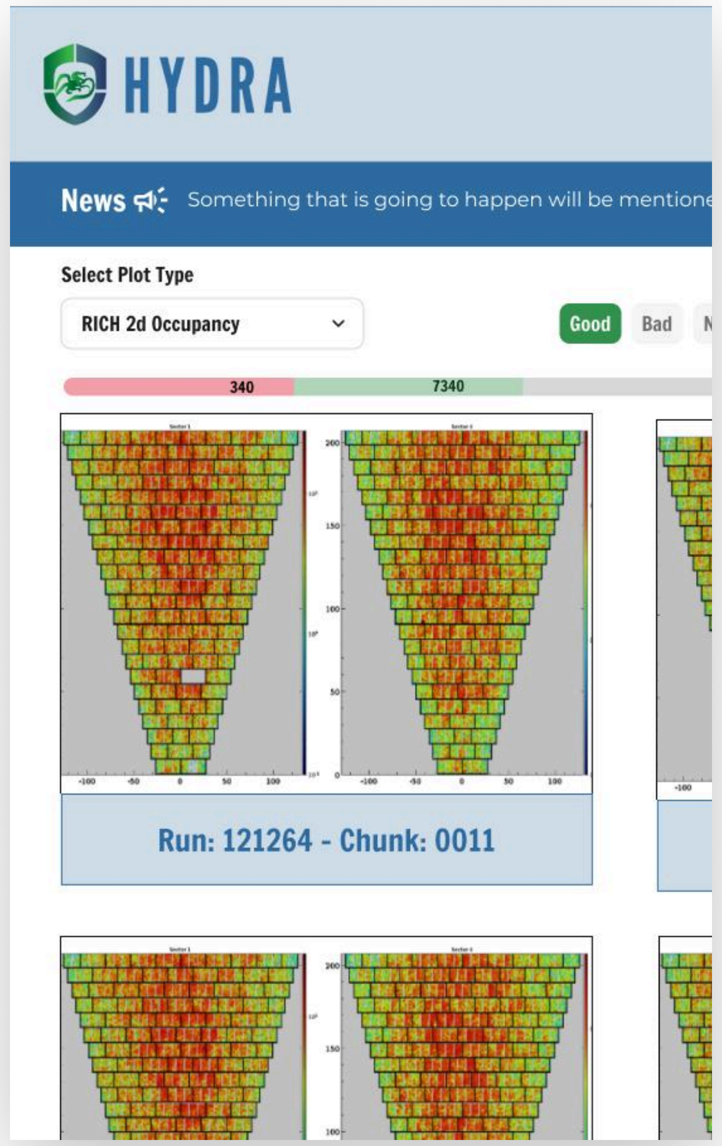
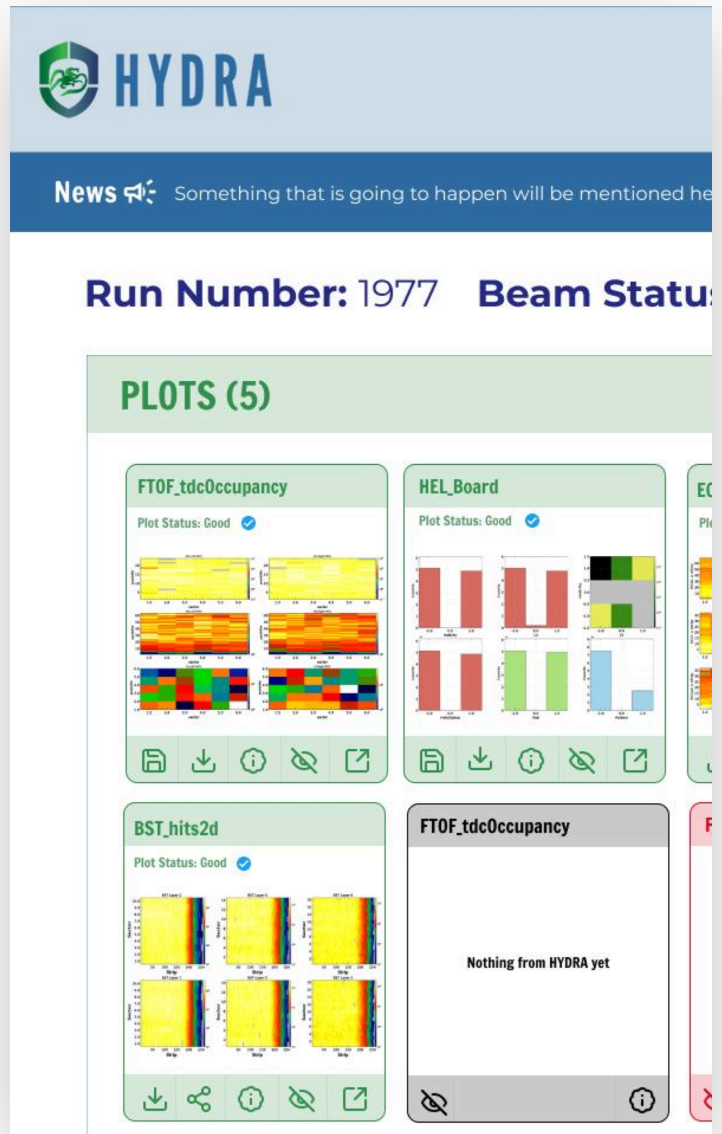


Image Labeler

Efficiently label thousands of images used for training a model.

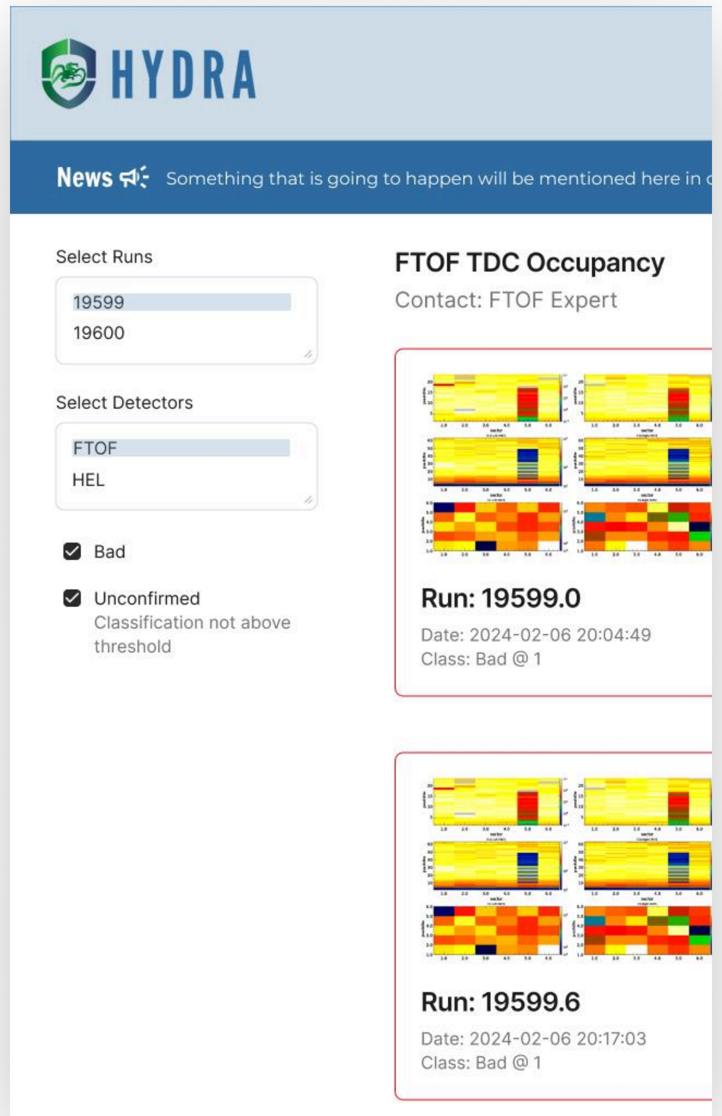


Run

See predictions in near real-time. This page continuously updates with new images during an experiment.

Grafana

Displays all predictions over time. Trend analysis on predictions can indicate when it is time to retrain a model.

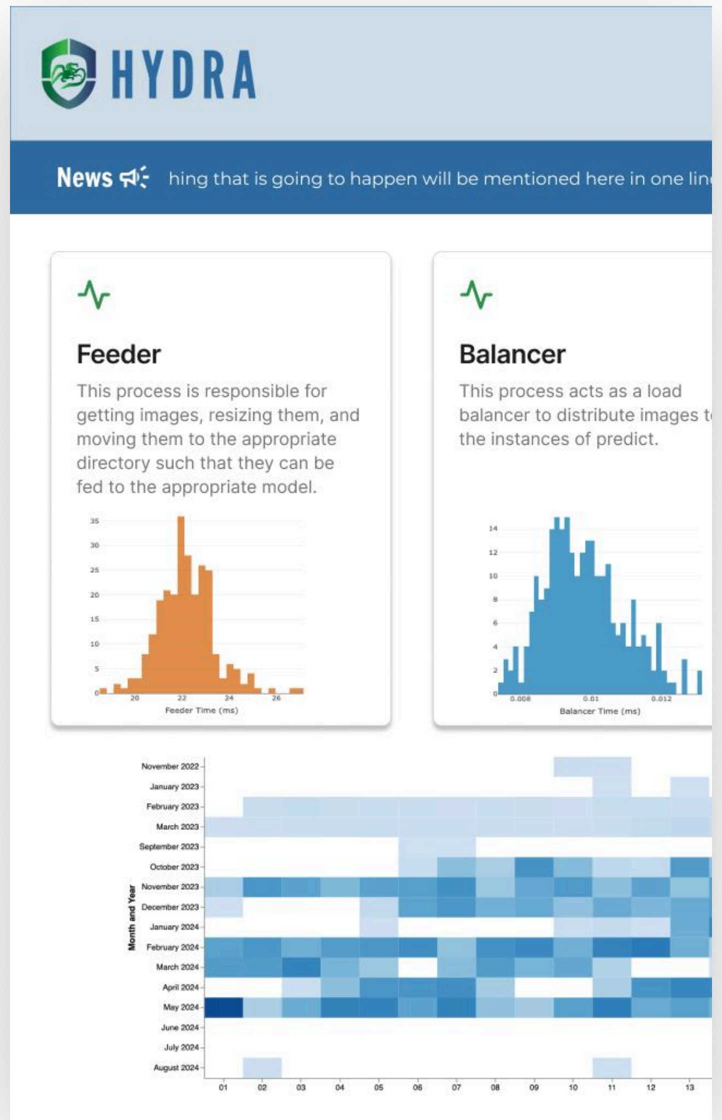


Log

Displays problematic and potentially problematic images from a trailing 24 hour window.

Library

Contains information useful to evaluate a given model's training and performance.



Status

Primarily used by administrators to monitor system performance.

Conclusions

Please label!

Use shift + click to label multiple images at once
If you have questions on how to use the labeler, we can schedule a demo for you.

Models will be retrained before the run starts.

Training reports will be sent out, performance will be monitored.

New UI will be released soon.

If you would like to test it out, give feedback, criticize, whatever, please let us know!

extras

Hydra team

David Lawrence

EPSCI
Physics, CS, AI/ML

Thomas Britton

EPSCI
Physics, CS, AI/ML

Torri Jeske

EPSCI
Physics, CS, AI/ML

Nataliia Matsiuk

Information + Records
Containerization, QA

Raiqa Rasool

EPSCI
CS, Full Stack Development

Darren Upton

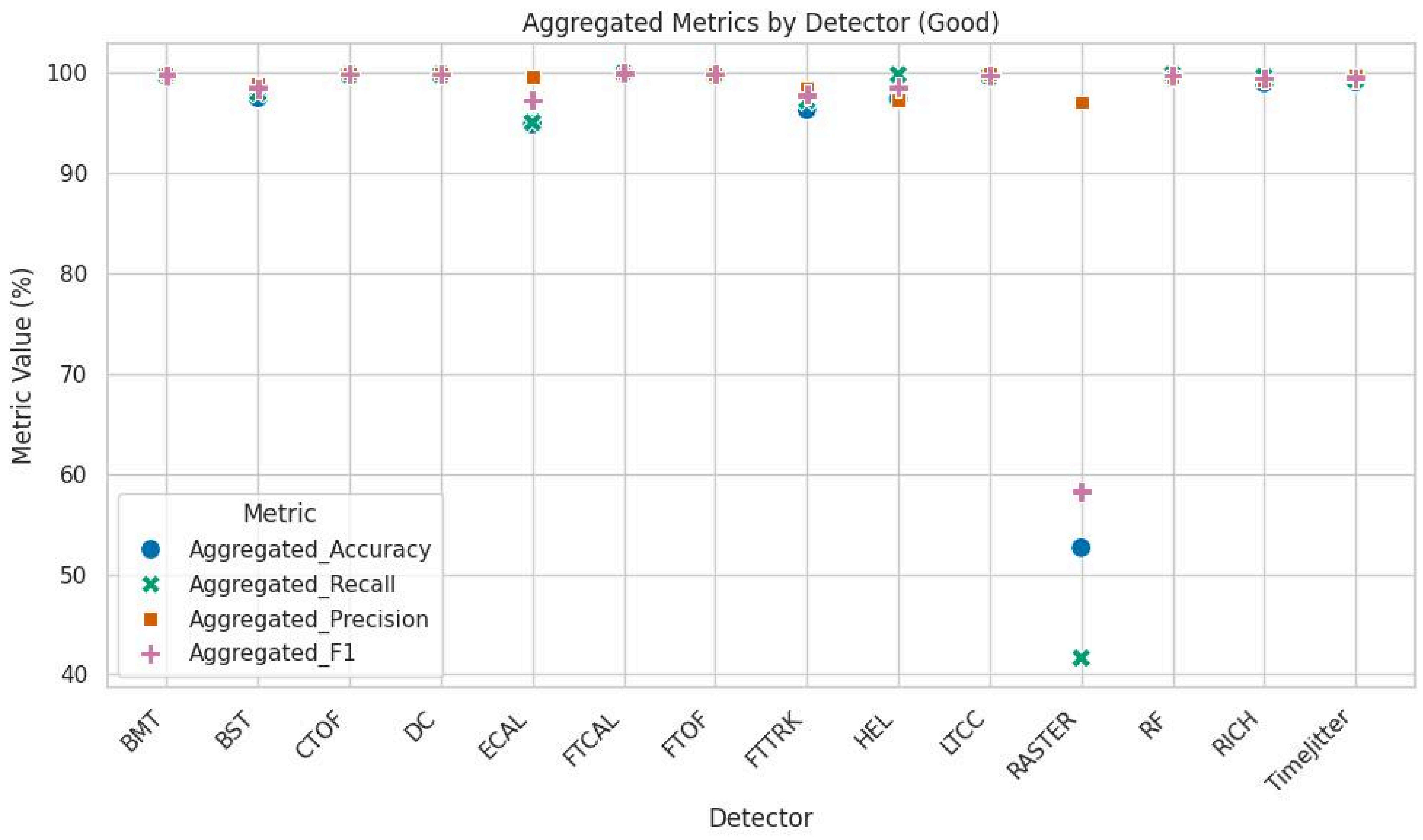
ODU Physics Grad Student
Hall B Service Work

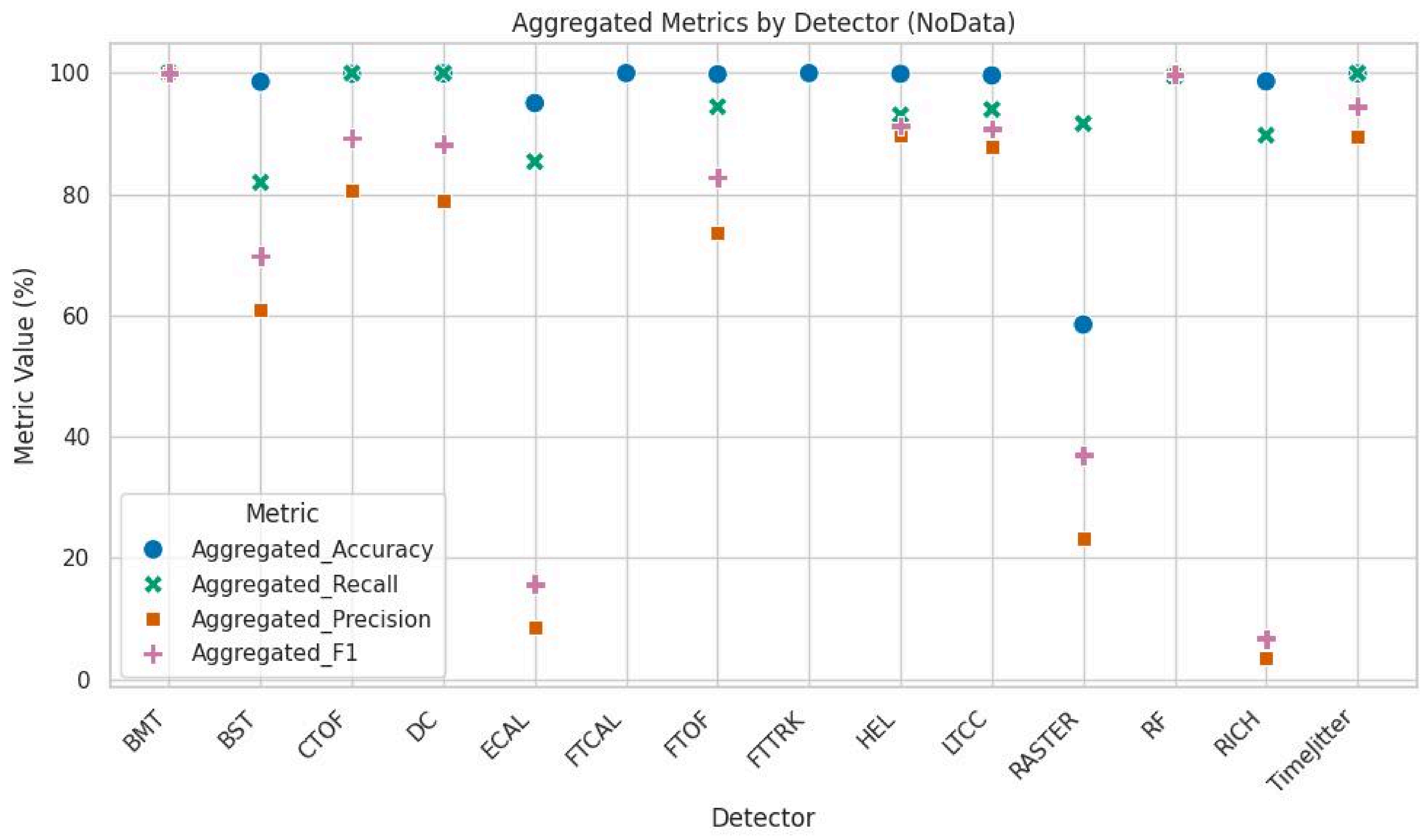
Jordan O'Kronley

UT Physics Graduate Student
Just started

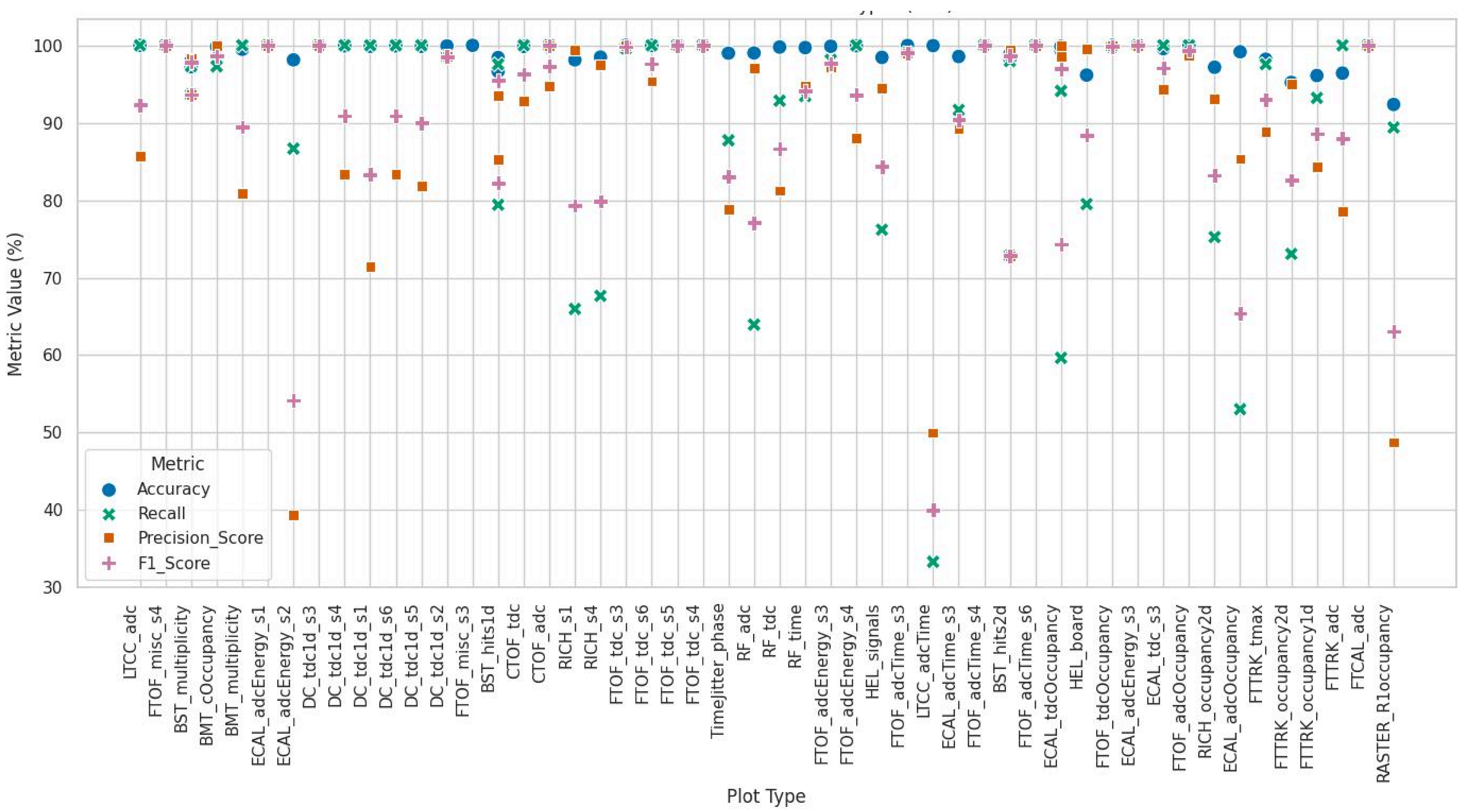
Aggregated metrics by Class (Good and NoData)

These are obtained from the set of labeled images for a given detector system.

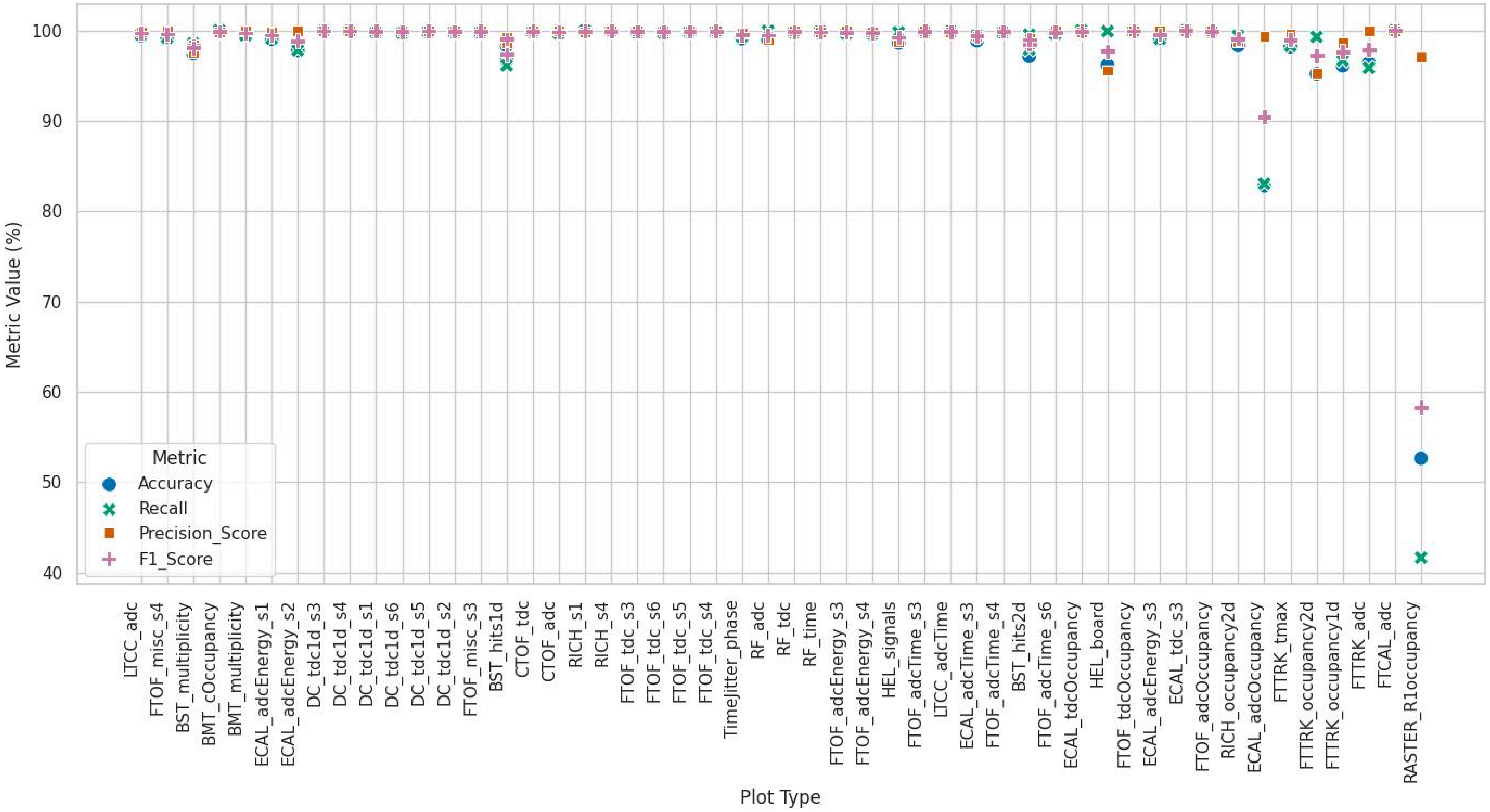




Performance metrics by Plot Type for Bad Class



Performance metrics by Plot Type for Good Class



Performance metrics by Plot Type for NoData Class