Hydra: Computer Vision for Online Data Quality Monitoring

Torri Jeske[1,*], *Thomas* Britton[1], *David* Lawrence[1], and *Kishansingh* Rajput[1]

[1]Thomas Jefferson National Accelerator Facility

Abstract. Hydra is a system utilizing computer vision for near real-time data quality monitoring. Currently operational across all of Jefferson Lab's experimental halls, it reduces the workload of shift takers by autonomously monitoring diagnostic plots during experiments. Hydra uses "off-the-shelf" supervised learning technologies and is supported by a comprehensive MySQL database. To simplify access, web apps have been developed to facilitate both labeling and monitoring of Hydra's inferences. Hydra can connect with the alarm system and incorporates complete historical tracking, enabling it to identify issues that shift takers could miss. When issues are detected, a natural first question is: "Why does Hydra think there is a problem?" To answer, Hydra employs Gradient-weighted Class Activation Maps (GradCAM) to identify regions of the image that are important for the specific classification. This interpretive layer enhances transparency and trustworthiness, which is essential for integration with experiment workflows and operation. The Hydra system, results, and sociological considerations for deployment will be discussed.

1 Traditional Data Quality Monitoring

The Thomas Jefferson National Accelerator Facility in Newport News, Virginia is home to four experimental halls, each having unique detector systems in place to carry out research in support of the Lab's scientific mission. During experiments, shift crews are responsible for monitoring the quality of the data, responding to phone calls, alarms, configuration changes, unplanned experiment down events, emergencies, and other events during the normal course of data taking. The counting houses are staffed 24 hours a day with a 2 person shift crew, at minimum. During the COVID-19 pandemic, this was reduced to a single person in the counting house and one person performing tasks remotely. The shift crew is staffed from a pool of graduate students, post-doctoral researchers, and staff scientists for either 6 or 8 hour shifts, depending on the specific hall.

Online monitoring is tedious. There are discrepancies inherent with human-based monitoring as no two shift crews monitor the data with the same diligence and frequency, even with established guidelines. In addition, there are typically thousands of plots to look at on a daily basis, as shown in Fig. 1. Given the challenges associated with human-based online monitoring, potential issues are prone to going unnoticed; this is especially true of intermittent problems that occur on small time scales. The volume of images to review along with the other responsibilities of the shift crew can contribute to even obvious issues being missed. To mitigate the inconsistencies with human based monitoring, Hydra has been developed and deployed in all four experimental halls.

*e-mail: roark@jlab.org

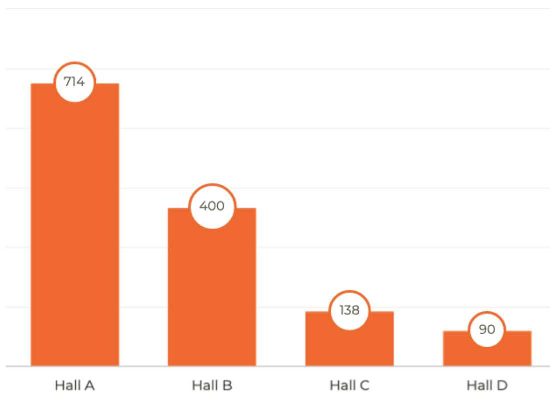

Figure 1. The approximate number of individual histograms, per experiment per run, monitored by the shift crew.

2 Hydra: Data Quality Monitoring with Computer Vision

Hydra is a system designed for real-time data quality monitoring, encompassing both the training and managing of AI models as well as the user interface to them. A comprehensive MySQL database serves as the backend, containing a wealth of information related to various monitoring histograms, model training parameters, classifications, user permissions, available labels, and storage locations for model files and images. The front end of Hydra is comprised of web applications, a deliberate choice to accommodate users who may lack experience in working directly with databases or artificial intelligence. The web applications are accessible from anywhere, facilitating remote monitoring of detector performance, a particularly valuable feature during the COVID-19 pandemic when remote shifts were required.

By default, Hydra primarily utilizes InceptionV3 [1], a deep learning network developed by Google for image classification. InceptionV3 contains 48 layers, combining convolutional, pooling, and dense layers with extensive use of batch normalization applied to activation inputs. The loss function is computed using Softmax. When evaluated on the ImageNet data set, InceptionV3 is able to attain greater than 78% accuracy. The model is available in Keras [2] and TensorFlow [3]. Hydra does not impose a restriction on the use of InceptionV3; users can develop and train their own models for image classification if they wish.

At present, each Hall has its own software to display images for the shift crew to compare to references and record in the logbook. As a result of this logging, we have a tremendous amount of monitoring images available for use in training image classifiers. This is important, as deep neural networks tend to require large data sets for training. Hydra utilizes a "push-button" training script to train a model using these images. The pool of monitoring images employed for training typically exhibits a large class imbalance, stemming from the fact that our detectors function properly nearly all of the time. Consequently, we can strategically under sample the "Good" images as they are often very similar, without sacrificing model performance. Once training has completed, inference is performed across the complete set of labeled images with metrics such as precision, recall, accuracy, and the F1 score being automatically computed. A report is generated enabling users to re-evaluate instances where Hydra's classification and the expert's label do not agree. This stage proves instrumental in identifying and correcting errors in labeling or classification.

Currently, models are trained for a single monitoring image, with the selection of images for Hydra's monitoring being the responsibility of the detector experts. Once the model is deemed suitable for use in production, both the image and its associated classifications will appear on the Hydra Run page. As shown in Fig. 2, this is displayed in the counting houses, typically alongside the traditional monitoring software. Here, a "bird's eye view" of the monitoring histograms is displayed. For documentation purposes, the run number, date and time, plot name, and Hydra's classification with confidence is shown. The Hydra Run page has a configurable display based on user preferences. If a plot has an associated model but is not currently suitable for monitoring, users can temporarily hide the plot from the page. Users can choose to superimpose heat maps generated using Gradient weighted Class Activation Maps (GradCAM) [4], which serves to highlight regions of focus that Hydra considers when generating its classification. This feature can assist the shift crew in promptly identifying problematic images and comprehending the underlying reasons, without necessitating the presence of a detector expert. When Hydra is confident that an image is "Bad", the image will be highlighted in red, alerting the shift crew of a potential problem. A green border indicates an image is classified as "Good" or "Acceptable" and that the confidence is above the designated threshold. A black border is reserved for miscellaneous classifications like "LED", "cosmics", and other labels that the individual detector experts requested for their plot types.

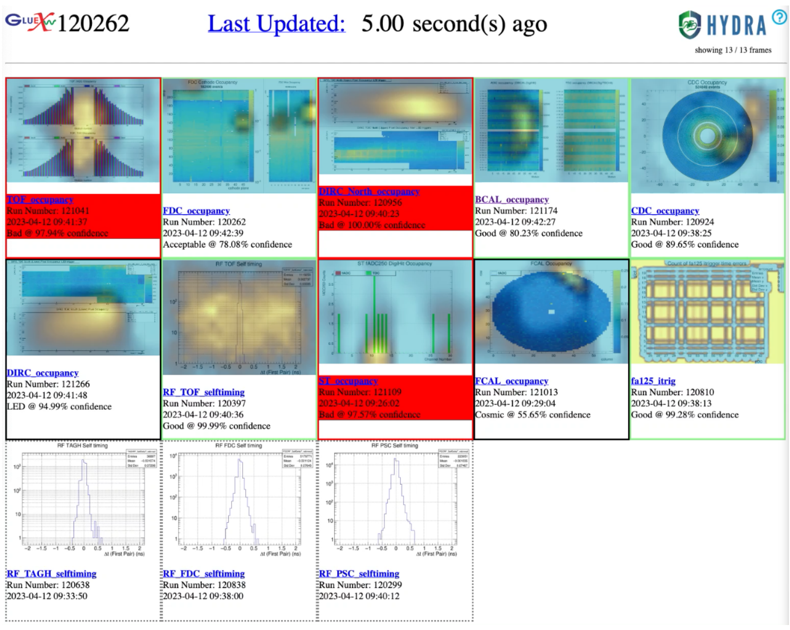

Figure 2. The Hydra Run page. This page is visible in the counting house during experiments. For each image, it displays the name, date and time, associated Run Number, and Hydra's classification and confidence. Optionally, the GradCAM heat maps can be overlayed. Images that are classified as "Bad" with high confidence are highlighted in red.

Note, the Hydra Run page will still display monitoring plots that do not have a model suitable for production because the shift crew should, in principle, still be monitoring those. They appear at the bottom of the page with a dashed border and without Hydra's classification and confidence.

3 Hydra User Experience

Web applications have been developed to facilitate interaction between the Hydra system and its users. Ensuring that the front end system is user-friendly, responsive, and dependable is a top priority. To this end, the front end of Hydra consists of multiple pages that serve different purposes. The initial point of interaction for users, aside from the Run page, is the data labeler, shown in Fig. 3. Here, detector experts and users (with permission) can efficiently label numerous images concurrently. When the labels are submitted, the associated database is updated. A continual stream of images is directed to the labeler throughout the experiments to allow for performance monitoring and further model training.

Figure 3. The image "labeler" tool. This page allows users to assign labels to multiple images efficiently. Users can select images from a range of run numbers and can choose how many columns of images to display.

Once a model is trained and inference is run, users can visit the Library page, shown in Fig. 4, to view an "enhanced" confusion matrix. For each entry in the confusion matrix, users can see a distribution of the maximum Softmax probability (normalized to 1) for each inference. This enables users to determine if Hydra is *confidently* mis-classifying images. Moreover, the option to apply a threshold to the distribution, using either the F1 score as a default or through manual input, enables the rate of false positive alarms to be fine tuned. The active models, those used in production, are denoted by a star icon. The page facilitates comparison of all trained models for each plot type.

The remaining web applications are dedicated to displaying time series data of Hydra's classifications. The Grafana dashboard, shown in Fig. 5, visually represents prediction over time for all detector systems. Users have the flexibility to selectively view a subset of predictions and/or detector systems. Grafana solely displays the predictions and confidence values, excluding the corresponding images. For a comprehensive view that includes both predictions and images, Hydra Log was developed. This page enables users to inspect "Bad" and "Unconfirmed" images from the preceding 24 hours. An example is shown in Fig. 6. Similarly to the Grafana dashboard, users can selectively view a subset of run numbers and/or detector systems.

4 Performance

Hydra has proven its efficacy on multiple occasions by identifying issues both in advance of and alongside the shift crew. Two such examples are shown below. The first occurred

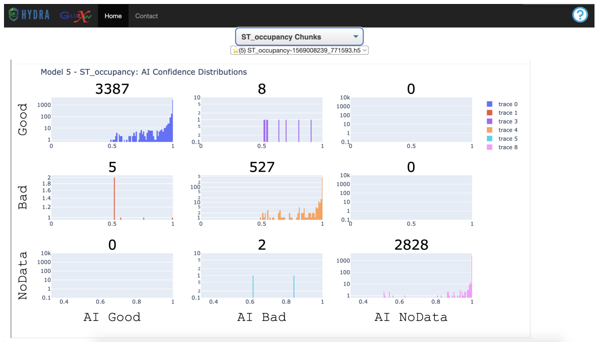

Figure 4. The Library page displays an enhanced confusion matrix. In each cell, the confidence distribution is shown in addition to the counts belonging in that cell.

Figure 5. The Grafana dashboard allows users to see all of Hydra's classifications over time.

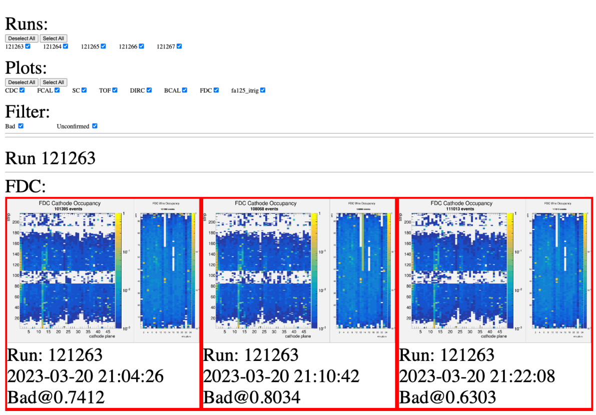

Figure 6. The Hydra Log page displays all "Bad" and "Unconfirmed" images for the previous 24 hours. Users can select all available runs and detector systems or just a subset for viewing.

in Hall B on 02/20/2023 at 08:35 EST, Hydra detected an anomaly with the electromagnetic calorimeter (ECAL). As shown in Fig. 7, sector 2 of the ECAL is empty. The heat map shows that Hydra correctly identified the problematic regions of the histograms. This was rectified immediately during a planned downtime by replacing an electronics board.

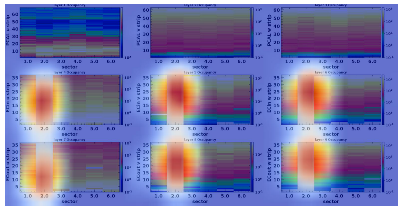

Figure 7. A monitoring image obtained from Hall B showing the occupancy per sector for the CLAS12 Electromagnetic Calorimeter. The GradCAM heat map is superimposed to indicate where Hydra was looking when it classified this image as "Bad".

The second example, occurs infrequently in Hall D, is critical to identify and respond to immediately. The histogram displays the count of trigger time errors, which should always be zero. In the event Hydra detects color in the histogram (indicating a non zero count), the shift crew is alerted via the audible alarm system. It is imperative the shift crew stops taking data and contacts the DAQ experts as data taken during this state is not usable for analysis. It was observed that the shift crew would frequently miss this plot displaying color, even though the monitoring histograms are cycled through. Since applying an audible alarm to this plot, the shift crew's responsiveness has improved significantly drastically reducing the amount of unusable data collected.

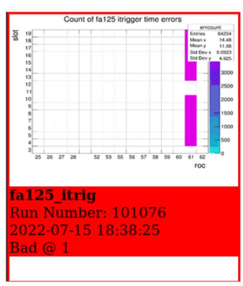

Figure 8. A monitoring image obtained from Hall D showing the count of trigger time errors. This plot should always remain empty. In the event the count is non-zero, Hydra flags the image as "Bad" and an audible alarm is sounded to alert the shift crew.

5 Conclusion

Variations in monitoring, diligence among the shift crews, and the overwhelming number of plots to review collectively present challenges to all nuclear physics research facilities in promptly identifying potential issues which can impact data quality. By incorporating computer vision techniques that consistently and rapidly analyze images, Hydra successfully augments the shift crew and provides a more frequent and standardized approach to online data quality monitoring. Through its user-friendly web applications, users can interact with Hydra without having experience with databases or artificial intelligence. With a low barrier to entry, users actively participate in model training, evaluation, and deployment. Hydra is under active development with dedicated efforts to enhance its core capabilities, especially in regards to interpretability and trustworthiness. Continuous updates to the front end interface, driven by user feedback, underscores the commitment to providing a system that can seamlessly integrate with humans and optimize the quality of the data taken at Jefferson Lab.

6 Acknowledgements

Jefferson Science Associates, LLC operated Thomas Jefferson National Accelerator Facility for the United States Department of Energy under U.S. DOE Contract No. DE-AC05-06OR23177.

References

[1] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, *Rethinking the inception architecture for computer vision* (2015), `1512.00567`

[2] F. Chollet, *keras*, https://github.com/fchollet/keras (2015)

[3] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin et al., *TensorFlow: Large-scale machine learning on heterogeneous systems* (2015), software available from tensorflow.org, `https://www.tensorflow.org/`

[4] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, International Journal of Computer Vision **128**, 336 (2019)