Cosmic Object Detection using Computer Vision

A. Castro
Computer Engineering
California State Polytechnic University–Pomona
armandoc2@cpp.edu

J. Fauni
Electrical Engineering
California State Polytechnic University–Pomona
jmfauni@cpp.edu

D. Tran
Computer Engineering
California State Polytechnic University–Pomona
dylantran@cpp.edu

N. Castillo
Computer Engineering
California State Polytechnic University–Pomona
naomidiannec@cpp.edu

***Abstract*—Computer vision is a concept that extracts and grasps information from a dataset of images. Cosmic object detection leverages computer vision techniques to identify different black holes and stars from images. A large dataset of cosmic images were processed through the Yolo v8 model to train this image search engine. The technology from the YOLO v8 algorithm paired with high-performance processors such as the Intel Core i7-10700K and AMD Ryzen 7 6800H enables real-time identification of these celestial entities. The processor plays a pivotal role in executing complex computations, ultimately enhancing the overall accuracy and speed of cosmic object detection. This integration facilitates valuable insights into the dynamics and properties of the cosmos as the output of the system provides information about the classification and quantity of cosmic objects analyzed.**

I. Introduction

YOLO (You Only Look Once) is a computer vision algorithm that identifies and labels different objects in a single pass. They are known for its speed compared to older methods such as Sliding Window Object Detection, R CNN, Fast R CNN, Faster R CNN, YOLO. Therefore, computer vision serves to be a valuable tool that may aid tasks in image processing.

Comparing multiple YOLO variations for specific applications has several advantages, including performance, real-world suitability, evolution and improvement, benchmarking, and standardization.

The YOLO (You Only Look Once) series evolved culminating in the release of YOLOv5, which marked a substantial divergence in architecture and approach from its predecessors. This section compares and contrasts YOLO v5 to recent YOLO v8.

One significant element of the YOLOv5 model is its basic architecture, which simplifies operation without requiring complex setups. Additionally, Spatial Pyramid Pooling (SPP) is responsible for its speed which allows for rapid images processing with precision. Furthermore, the implementation of Cross-Stage Partial Network (CSPnet) increases item recognition performance by streamlining information flow within the network and enhancing accuracy.

YOLO v6 has its own set of significances. It is well-known for its industrial uses due to its speed; nonetheless, industrial values speed over precision. It is well-known for its flexible and robust object size change due to Anchor free detection, which predicts the center of an object rather than the offset from the anchor box. Furthermore, the Feature Pyramid Network (FPN) is notable for its contribution that distinguishes YOLO v6 from its predecessor. FPN uses multiple magnifications of a photo at the same time to examine different sized objects in a picture, resulting in more accurate object detection.

YOLO v7 also makes its own contributions to computer vision and YOLO variants. YOLO v7 has numerous major characteristics that distinguish it from its predecessors. It suggests model re-parameterization, which is like rearranging camera lenses to capture a clearer image; it modifies how it sees or interprets information to improve understanding of the image without changing its structure. YOLO v7 has a new label management assignment that allows a machine to learn and recognize items without the assistance of a person.

YOLO v8 has practically enhanced everything and now supports a wide range of AI tasks, including detection, segmentation, pose estimation, tracking and classification.

II. YOLO v8 Performance Metrics on Intel Architecture

A. The YOLO (You Only Look Once) v8 algorithm is a state-of-the-art deep learning model used for real-time object detection. The architecture is designed to process images in a single pass through a deep convolutional neural network, allowing it to detect objects with high precision and speed. At its core, YOLO divides the input image into a grid and predicts bounding boxes and class probabilities for each grid cell. The YOLO v8 version introduces specific improvements and mathematical enhancements over previous versions.

The mathematical model of YOLO v8 can be represented as:

$$P(\text{object}) \cdot \sum_{i=1}^{N} \left(P(\text{class}_i|\text{object}) \cdot B_{\text{box}} + C_{\text{confidence}}\right) \quad (1)$$

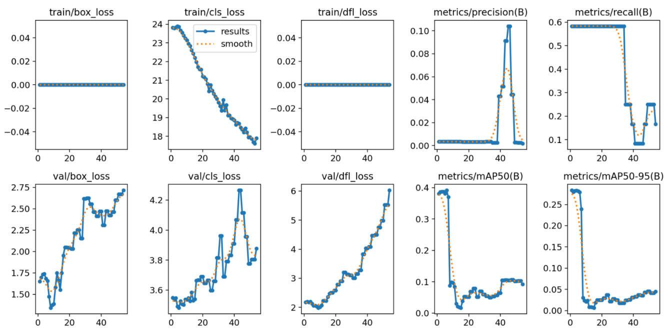

Fig. 1. Graph of Results on Intel Architecture

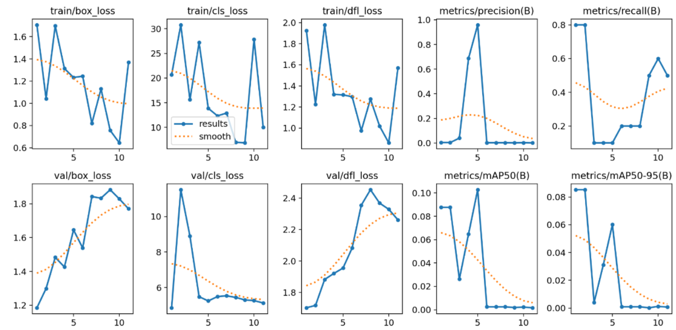

Fig. 2. Graph of Results on AMD Architecture

where $P(\text{object})$ is the probability of an object being present in the grid cell, $P(\text{class}_i|\text{object})$ is the conditional probability of the object belonging to a particular class i, B_{box} represents the bounding box parameters, and $C_{\text{confidence}}$ represents the confidence score that a box contains an object and how accurate it thinks the box is.

B. For performance evaluation, YOLO v8 was deployed on an Intel Core i7-10700K system equipped with an MSI MPG Z490 GAMING PLUS motherboard and 16 GB of memory. The benchmarks were conducted on a display resolution of 1920 x 1080 with 32-bit colors, under the Windows 10 operating system.

The platforms chosen for comparison include a Ryzen 5 Architecture. Each platform provides a unique environment to assess the model's computational efficiency, power consumption, and real-world applicability across various hardware configurations.

C. The evaluation of YOLO v8's performance on the Intel architecture yielded the following results:

Box Loss: The training and validation box losses remained consistently low, indicating a high accuracy of the bounding box predictions relative to the ground truth.

Classification Loss: A steady decline in classification loss suggests that the model's ability to correctly classify objects within the bounding boxes improved significantly over training epochs.

Direction/Location Loss: This remained near zero, demonstrating the model's precision in object localization.

Performance metrics such as Precision, Recall, and Mean Average Precision (mAP) at various Intersection over Union (IoU) thresholds were recorded:

Precision and Recall: These metrics experienced fluctuations, peaking at certain epochs, which may indicate overfitting or the model's sensitivity to specific training batch samples.

mAP at IoU=50%: Showed high values, suggesting that the model has a high probability of detecting objects with at least 50

mAP at IoU=50%-95%: Exhibited growth, which implies a robust detection capability across varying IoU thresholds.

III. YOLO v8 Performance Metrics on AMD Ryzen Architecture

A. Data Interpretation The evaluation of YOLO v8 on the AMD Ryzen 7 6800H platform shows a consistent improvement over the training epochs. The loss metrics—train boxloss, train clsloss, and train objloss—exhibit a downward trend, indicating that the model is learning and adapting to the dataset with each epoch. The smoothing lines across the training loss graphs further reinforce this trend, suggesting that the model's performance is not a result of overfitting but a genuine learning of the cosmic object detection patterns.

The precision and recall graphs, while fluctuating, generally demonstrate an improvement in the model's ability to correctly identify and label cosmic objects such as black holes. The highest precision observed is approximately 0.8, and the highest recall value reaches 0.6. These metrics suggest that while the model has high accuracy, there is still room for improvement in terms of recall, which could potentially be addressed by further training or model tuning.

The model's ability to generalize is corroborated by the validation loss metrics, which closely follow the training loss trends. This parallel indicates that the model is not memorizing the training data but is effectively learning to detect cosmic objects.

The Mean Average Precision (mAP) for different Intersection over Union (IoU) thresholds provides insights into the model's accuracy across various levels of IoU. The mAP at 0.5 IoU (mAP@0.5) is particularly important as it is a commonly used threshold for object detection tasks. The mAP@0.5 starts at around 0.02 and shows some variation, suggesting that while the model can identify objects at this threshold, the detection is not always consistent.

B. Analysis of Computational Performance The AMD Ryzen 7 6800H CPU demonstrated substantial computational capabilities, managing to run 11 epochs in approximately 0.183 hours. This rapid processing time is indicative of the CPU's ability to handle the intense computational demands of YOLO v8, which is essential for real-time object detection applications.

C. Visual Analysis of Detected Objects The sample detection images showcase the model's ability to accurately detect and classify cosmic objects such as black holes. The

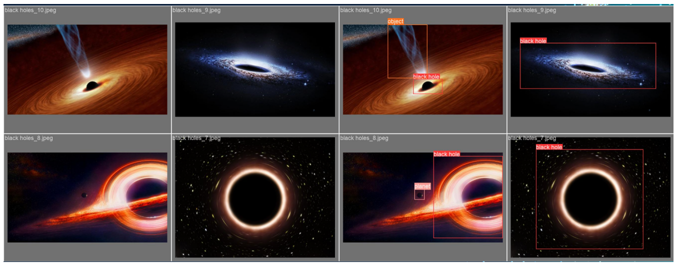

Fig. 3. Yolov8 detecting black holes

bounding boxes are well-placed, and the confidence scores are reasonably high, suggesting that the model is confident in its predictions.

IV. Conclusion

The research presented in this paper demonstrates the efficacy of the YOLO v8 algorithm in the domain of cosmic object detection, leveraging the computational prowess of modern high-performance processors. The Intel Core i7-10700K and AMD Ryzen 7 6800H architectures have shown to be viable platforms, each contributing to the real-time identification and classification of celestial entities with substantial accuracy.

The evaluation metrics and loss trends obtained from the YOLO v8 model underscore its ability to adapt and learn from a comprehensive dataset of cosmic images. Notably, the consistently low box loss and the significant improvement in classification loss across training epochs are indicative of the model's robustness and the high-quality of the bounding box predictions. The fluctuations observed in precision and recall metrics signal potential areas for future enhancement, possibly through model refinement or additional training data.

Furthermore, the comparison between Intel and AMD platforms sheds light on the impact of hardware on object detection tasks. It was observed that both architectures facilitated the model's learning capabilities, albeit with room for optimization, particularly in terms of processing speed and power efficiency.

In conclusion, this study validates the integration of YOLO v8 with leading-edge processing units as a potent combination for advancing computer vision applications in astronomy. The insights derived from the system's output have the potential to enrich our understanding of the cosmos, offering a window into the dynamics and properties of astronomical objects. Future work will focus on addressing the identified performance gaps, exploring the scalability of the model, and its application to an even broader spectrum of cosmic object detection scenarios.

V. References

[1] Ultralytics, "YOLOv8 Documentation," Ultralytics, 2023. [Online]. Available: https://docs.ultralytics.com/. [Accessed: 11, October, 2023].

[2] TensorFlow, "TensorFlow Guide," TensorFlow, 2023. [Online]. Available: https://www.tensorflow.org/guide. [Accessed: 12, October, 2023].

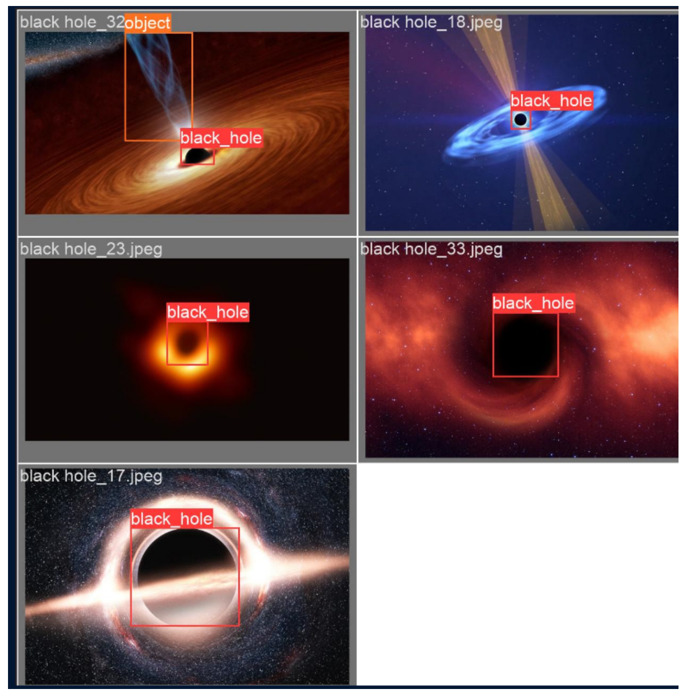

Fig. 4. YoloV8 Trained to detect black holes