1. The selection of the beyblade detection model that you choose, and the data that your model is trained on (if it's a machine learning model).

Consideration of choosing YOLO V8 model :

1. State-of-the-Art Performance: YOLOv8 represents the latest iteration of the YOLO series, known for its high accuracy and efficiency in real-time object detection tasks. Leveraging advanced algorithms and architectures, YOLOv8 can accurately detect Beyblades even in dynamic and cluttered video scenes.
2. Speed and Efficiency: One of YOLO's hallmark features is its ability to deliver high-speed detections without compromising on accuracy. YOLOv8 continues this tradition, making it ideal for video processing where real-time or near-real-time performance is crucial. Additionally i choose this because im developing on pc without gpu
3. Ease of Use and Integration: YOLOv8 can be easily integrated with OpenCV. This flexibility simplifies the integration of the model into different systems and applications, facilitating seamless deployment and scaling.
4. Community and Documentation: YOLOv8 benefits from an active community and extensive documentation, providing developer with ample resources for troubleshooting and optimization. This support network ensures that i can quickly address any issues and continuously improve the model.

YOLO V8 combination of cutting-edge performance, speed, ease of use, and strong community support makes it a highly suitable choice for my beyblade detection project, ensuring i can deliver a robust and efficient solution is short time period.

2. The accuracy of the beyblade detection model.

Performance Metrics

Dataset	Precision	Recall	mAP50	mAP50-95
train	0.985	0.987	0.993	0.768
val	0.986	0.979	0.992	0.777
test	0.99	0.977	0.994	0.748

* the metric is in the range of 0-1 higher indicating better performance.

Precision is akin to the model's sharpness, signifying how often it gets things right. In simpler terms, when our model spots a beyblade, it's accurate approximately [Precision]% of the time. This reliability suggests that if we deploy it on new, unseen videos, it's likely to perform well in detecting beyblades. *

Recall measures the model's thoroughness, reflecting its ability to catch all beyblades present in images. A recall score of [Recall]% implies our model successfully spots nearly all beyblades, ensuring comprehensive coverage in our inspections. **

mAP50 tells us how good our model is at finding these objects accurately. A higher mAP50 score means our model is really good at pointing out where the beyblades are in the pictures. This is super important for checking the quality of our products.

mAP50 95 is similar to mAP50. Now, let's say we want our model to be even better at finding those beyblades. mAP50 95 looks at how well our model does this job, but it also considers different levels of difficulty. It's like testing our model's accuracy in different situations. So, even if things get a bit tricky (like if the beyblade is only partly visible), our model still does a good job overall.

In summary, our object detection model demonstrates exceptional accuracy and reliability in identifying beyblades within images, as reflected in its strong performance across multiple evaluation metrics. These results affirm the potential of our model to be utilized in production

3. Logic behind the additional data that your program generates.

The additional feature i extracted from the analysis is collision and the cause of battle ends. The number of collision before battle end may be a good factor in determining the strength and resiliency of the beyblade. If the beyblad are able to withstand multiple collision, its a sign of its toughness. Another feature is the cause of the battle ends. I think its a very important feature because we can derive information wheter the beyblade is weak or not stable so its easily thrown outside the arena, or if its not balanced and symmetrical so its faster to stop spinning.