# Beyblade Battle Analyzer

**Project: Bakuten Shoot Corp. Beyblade Battle Analysis**

**Author**: Efrad Galio
**Date**: 02 June 2025
**GitHub Repo**: https://github.com/Efradgalio/Beyblade-Battle-Video-Analysis-System

## Objective

The goal is to create a program that processes top-down view videos of real-life 1v1 Beyblade battles and outputs structured data useful for analysis. The system must:

- Detect and track both Beyblades
- Determine when the battle starts and ends
- Identify the winner
- Log battle events (e.g. collision, broken parts)
- Save all outputs into an analyzable CSV format

## Model Selection

### Model Used: `YOLOv11n` + `BoT-SORT`

- **YOLOv11n** (You Only Look Once, version 11 nano) is used for object detection due to:
    - State of the Art
    - Lightweight and Fast (Real-Time Inference)
    - High Detection Accuracy even using a small dataset
    - Easy to Use and Fine-Tune
    - Open Source

- **BoT-SORT** (Simple Online and Realtime Tracking with appearance embedding):
    - Accurate ID Tracking (with Re-ID Embedding)
    - Easy Integration with YOLO Models
    - Actively Maintained and Easy to Customize

### Model Training

- **Dataset**: The detection model is fine-tuned on a custom dataset with:
    - Top-down views of battles
    - 2 classes: Beyblade, Broken Part
    - Annotated using Roboflow, trained using Ultralytics
    - *Dataset Example: Can check the github inside test, train or valid folder*

- **Augmentations**: No Augmentations from Roboflow

### Model Evaluation

- **Metrics**:
  - mAP@0.5: 0.956<br>Precision: 0.962
  - Recall: 0.927

- **Inference Speed**: ~73 FPS on GPU (RTX 3050) with 2GB Dedicated GPU Memory and 6.9GB Shared GPU Memory

## System Design & Logic

### ID Stabilization

- Since BoT-SORT IDs can switch on collision or occlusion, a **stable ID assignment logic** is added.
- If new track IDs appear, they are assigned the closest `stable_id` based on Euclidean distance.

### Event Detection Logic

| Event | Logic |
|---|---|
| **Launch Detection** | First appearance of a Beyblade (stable_id 1), second launch when both detected |
| **Battle Start** | When both stable_ids are visible |
| **Collision** | If distance < 0.6 × bbox width or IoU > 0.05 |
| **Battle End** | If more than 2 objects (assumed broken parts) are detected |
| **Winner** | Based on which Beyblade keeps moving after breakage (motion threshold < 2 pixels for 30+ frames) |

### Optional Feature Logic

- **Remaining Spin Duration**: Currently not implemented, but can be tracked using time until movement stops.
- **Out-of-Arena**: Not implemented yet; possible via boundary constraints.
- **Arena & Beyblade Characteristic:** Not implemented yet.
- **Draw Condition:** If the two beyblades stop at the same time.

## Output Data

Data is exported into `battle_summary.csv` and contains:

| Features | Description |
|---|---|
| `First Beyblade Launch Time (s)` | Timestamp when the first Beyblade appears |
| `Second Beyblade Launch Time (s)` | Timestamp when the second Beyblade appears |
| `Battle Start Time (s)` | When both are present and motion begins |
| `First Collision Time (s)` | Time of first detected contact |
| `Total Collision` | Number of detected collisions |
| `Battle End Time (s)` | When a breakage occurs |
| `Winner` | Beyblade 1 or 2 |
| `Type of Finish` | `"broken beyblade"` (more can be added later) |
| `Battle Duration (s)` | End Time – Start Time |

## Testing

- **Test Videos**: 3 different battles, but with the same beyblades, arena, angle, and lightning.
- **Stable ID Accuracy**: Not that really stable, there few moments that the IDs are switching.
- **Winner Detection Accuracy**: Still can’t detect who's the winner, probably because the logic or threshold haven’t tuned or explored more yet.
- **Other Features Accuracy (*except Winner):** Quite accurate most of the time.

## Limitations

- Does **not yet** detect:
  - Beyblade stopping due to spin-out
  - Beyblade leaving the arena
  - Tie scenarios
- Still **can’t detect** the winner.
- The IDs must be very stable otherwise some of the logic feature extraction might not work.
- **Can’t run** in the CLI, need to use a notebook to run it.

## Future Improvements

- Fixing the winner detection
    - Fixing spinning velocity tracking per frame
    - Broken Beyblade Detection
    - And add Tie scenarios
- Integrate arena boundary logic to detect out-of-bound exits
- Deploy as web app for batch analysis
- Store logs in SQLite/PostgreSQL for advanced analytics
- Add visualization tools with graphs of distance vs. time
- Use Yolov11m or even higher with a lot of various dataset
- Use other Model like Gemini 2.5 Pro or others that can be directly use without training at all or small fine tune

## Insights for Data Analyst

With more battles recorded, a data analyst can extract:

- **Win Rate by Launch Order**
    - Does launching first or second give an advantage?
    - Useful for training strategy or mechanism optimization.
- **Win Rate by Collision Count**
    - Does high or low collision frequency correlate with wins?
    - Could hint at more aggressive (or evasive) strategies being better.
- **Average Time to First Collision**
    - Do faster collisions lead to shorter or longer battles?
    - Could indicate attack-type vs defense-type behavior.
- **Correlation Between Launch Gap and Outcome**
    - If the second launch is significantly delayed, does it decrease chances of winning?
- **Finish Type vs. Win Pattern**
    - Once more finish types are added (e.g. ring out, burst, spin finish), you can analyze:
        - Which Beyblade excels at which finish?
        - Is a Beyblade consistently winning via the same method?