Stadnik A.V.[1], Streltsova O.I.1[2,3], Bezhanyan T.[2,3], Belov O.V.[4]

1 NN Sputnik

2 Meshcheryakov Laboratory of Information Technologies JINR, Dubna, Russia

3 Dubna State University, Dubna, Russia

4 Veksler and Baldin Laboratory of High Energy Physics, Joint Institute for Nuclear Research

FULLY AUTOMATED DELINEATION AND ANALYSIS OF THE OPEN FIELD EXPERIMENTAL ARENA USING A NEURAL-NETWORK-BASED APPROACH

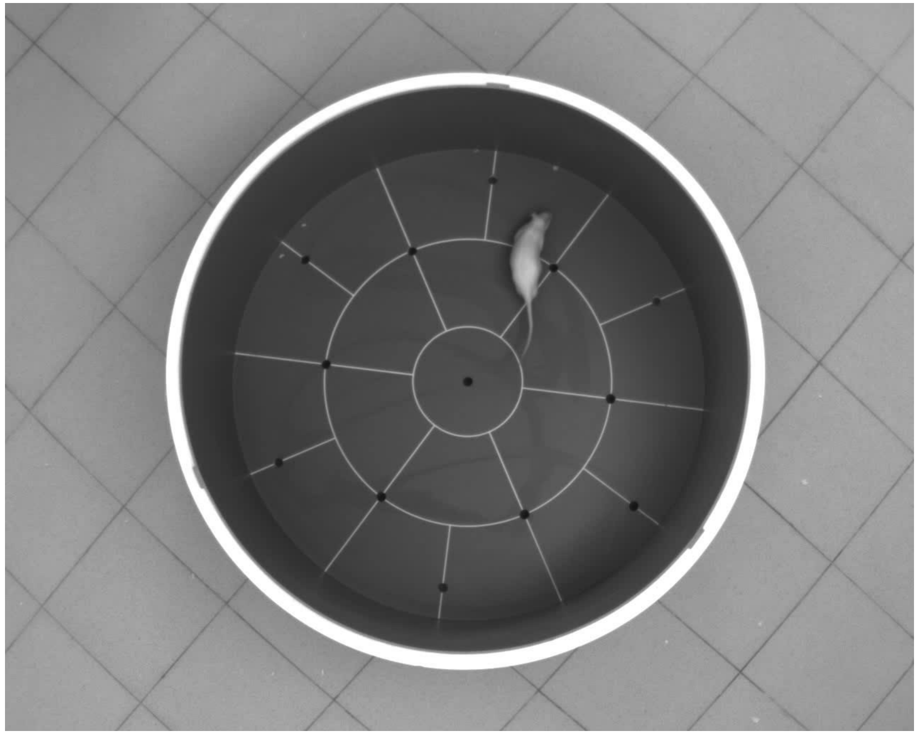
PROBLEM STATEMENT

Goal: Fully automate behavioral arena analysis

Pipeline:

- Arena annotation (defining ROI)
- Animal detection & tracking
- Matching detection results to arena layout + action detection:

Goes to → complete automation

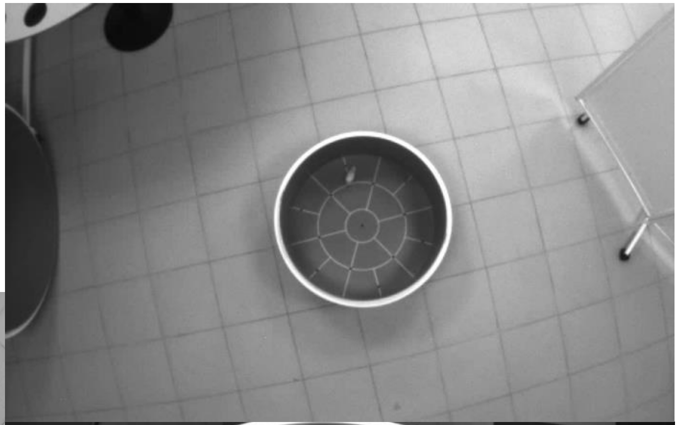


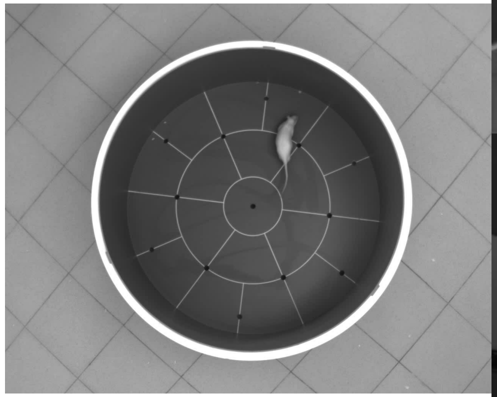
OPEN FIELD ARENA: TODO

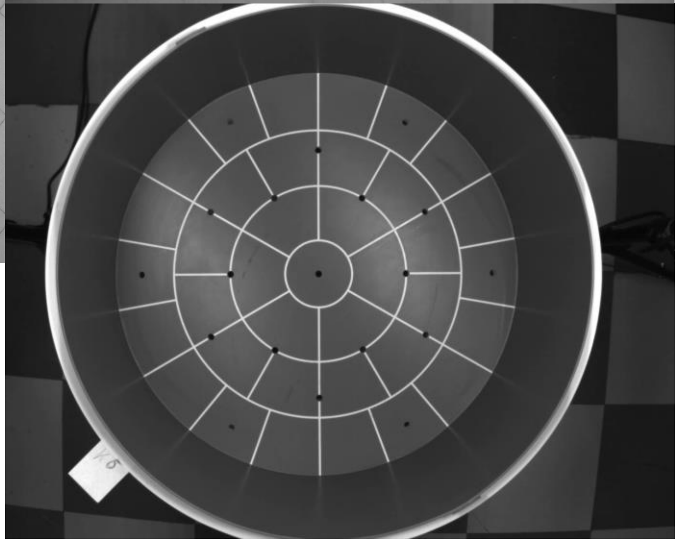
Objective: Fully automate the segmentation of the "Open Field" test arena

Benefit: Streamlines and enhances accuracy of subsequent animal-behavior tracking

- Leverages the SOLD^2 model from the Kornia library
- Delivers precise detection of both the arena's straight boundaries and its interior sectors





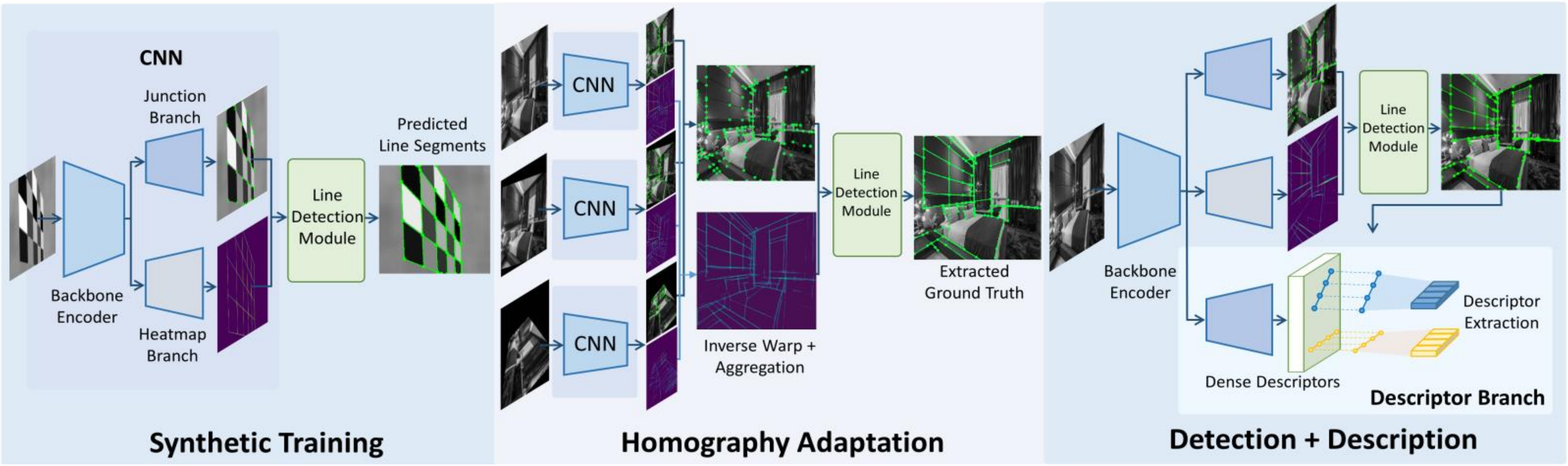


LINE DETECTION MODULE (KORNIA)

Uses SOLD² (Self-supervised Occlusion-aware Line Description and Detection) [https://arxiv.org/abs/2104.03362]

Unified deep network that simultaneously finds and describes line segments

Trained without manual labels on synthetic data, generalizes to real-world images



HOW SOLD² WORKS/ VS TRADITIONAL METHODS

Feature Maps:

- Junction heatmap: highlights potential line intersections
- Line heatmap: highlights pixels along lines

Segment Extraction:

- Search hot regions via dynamic programming
- Link junctions into straight segments

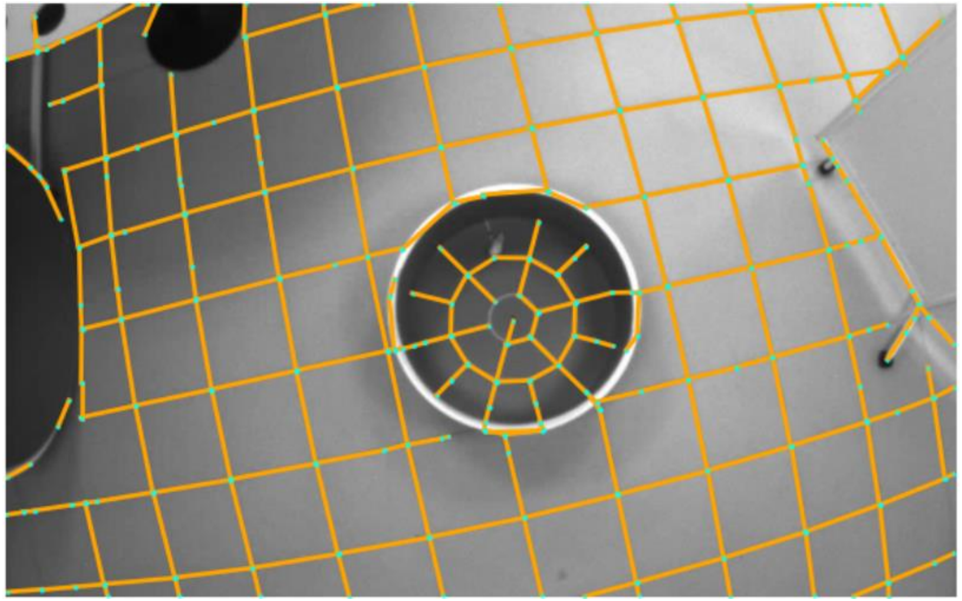
Descriptor Computation: robust to occlusions and viewpoint changes

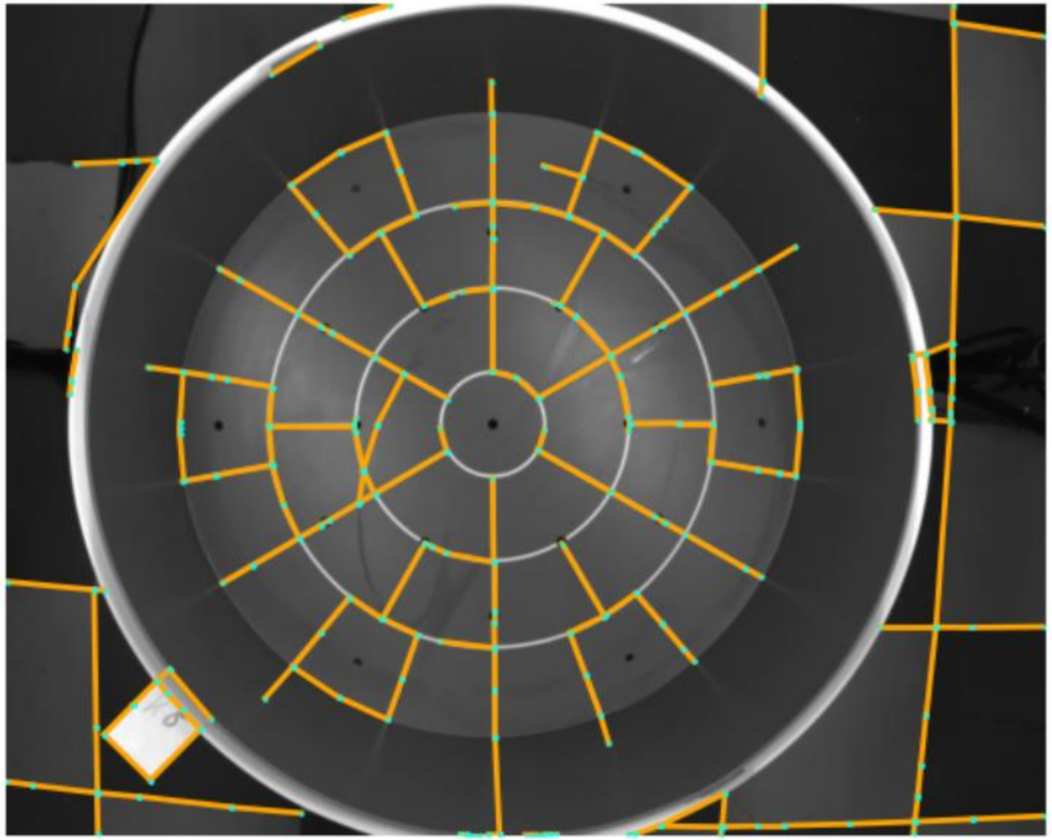
More precise endpoint localization

Higher repeatability across lighting and angle variations

SOLD² : LINE DETECTION – HOW IT LOOKS LIKE?

Segment filtering: discard segments with low confidence (score < 0.5). Enforce a minimum segment length to remove noisy short fragments.





FINDING THE ARENA CENTER

Intersection Computation for each pair of segments

Filter intersections outside image bounds

Accumulate intersections into a pixel-wise heatmap

Apply Gaussian blur for noise resilience

Locate max-value pixel via cv2.minMaxLoc

IQR (INTERQUARTILE RANGE)

Definition: IQR = Q3 – Q1 (25th to 75th percentile)

Robust to extreme values and outliers

Used to flag outliers via Tukey's rule: [Q1 – 1.5·IQR, Q3 + 1.5·IQR]

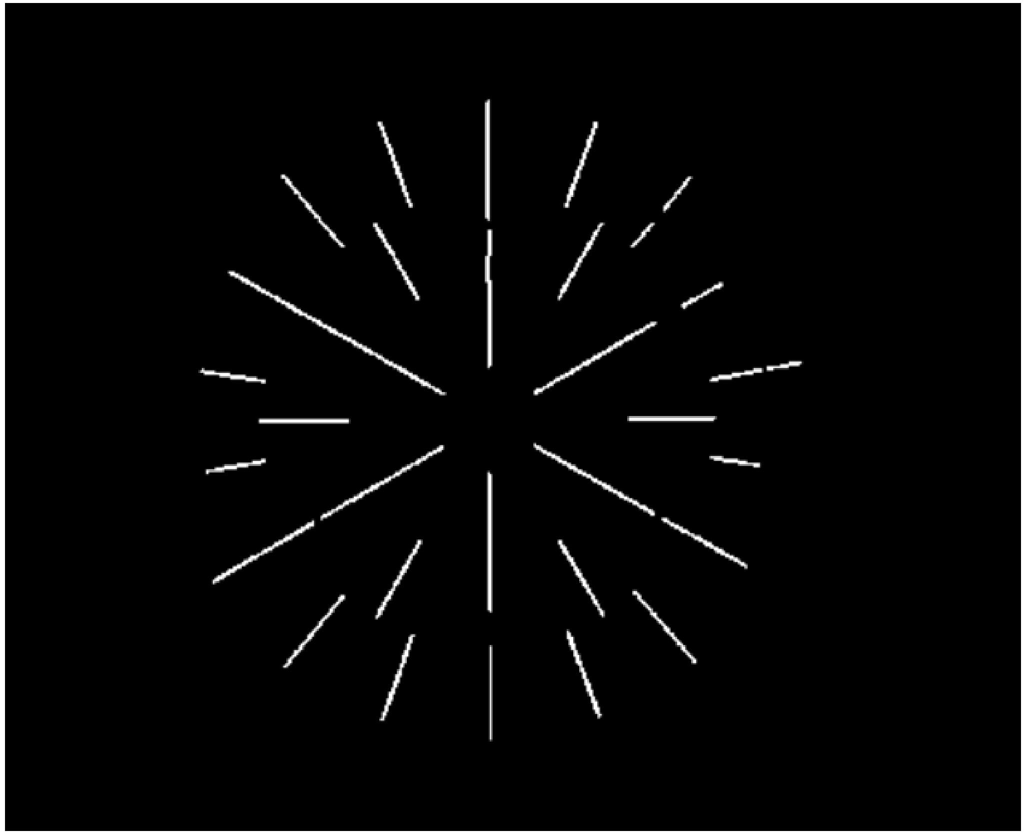
Sometime k != 1.5

BACKWARD STEP: FILTERING LINE SEGMENTS

Function remove_outliers(distances) computes IQR and discards points outside Tukey's bounds

point_to_line_distance: computes perpendicular distance to a segment

Keeps only segments within 3 px of the computed center

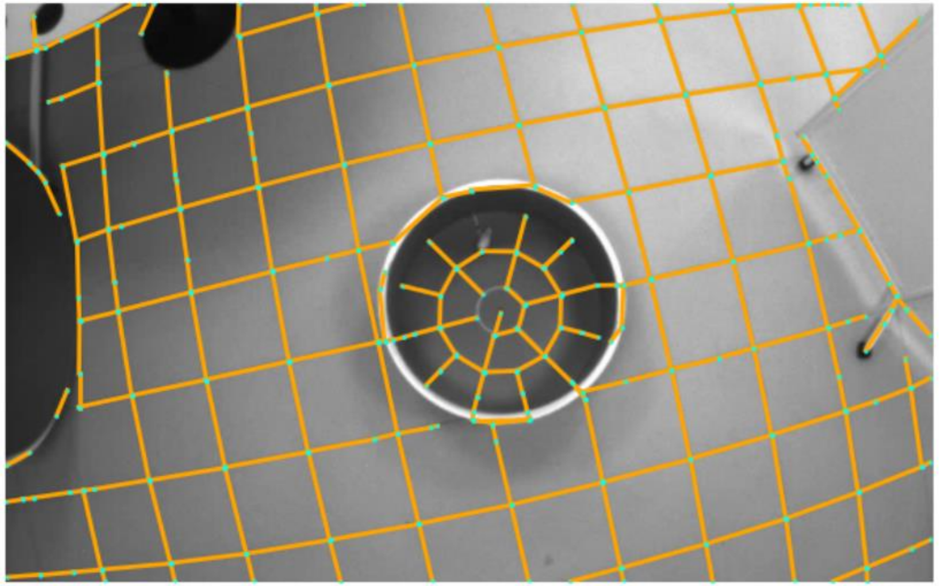


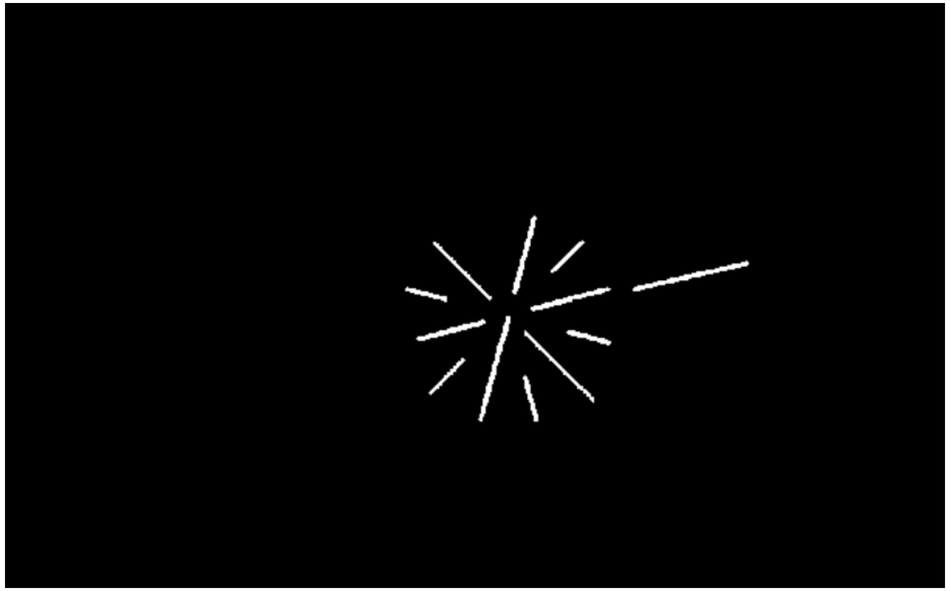
REGULARIZATION: RADIUS RANGE FILTERING

Filter segments by their radial distances

Compute min/max radii and discard outliers using IQR

Retains segments within robust radius bounds



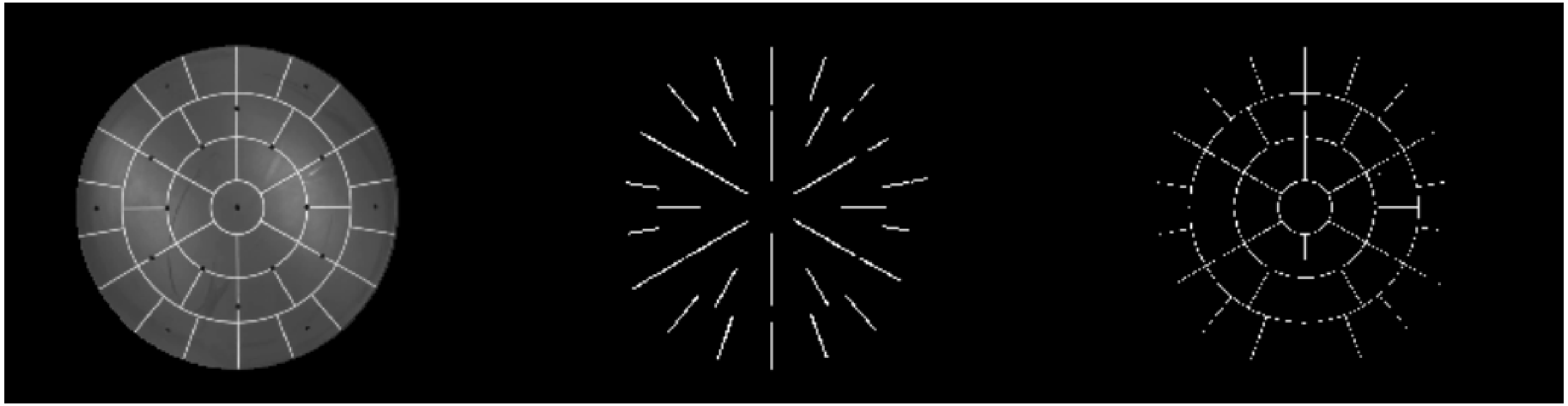


RADIUS ESTIMATION WORKFLOW

Objective: Identify likely radii around (x_0, y_0)

Mask Preparation: background subtraction, line masking,

IQR filtering, thresholding. Build a refined radius mask



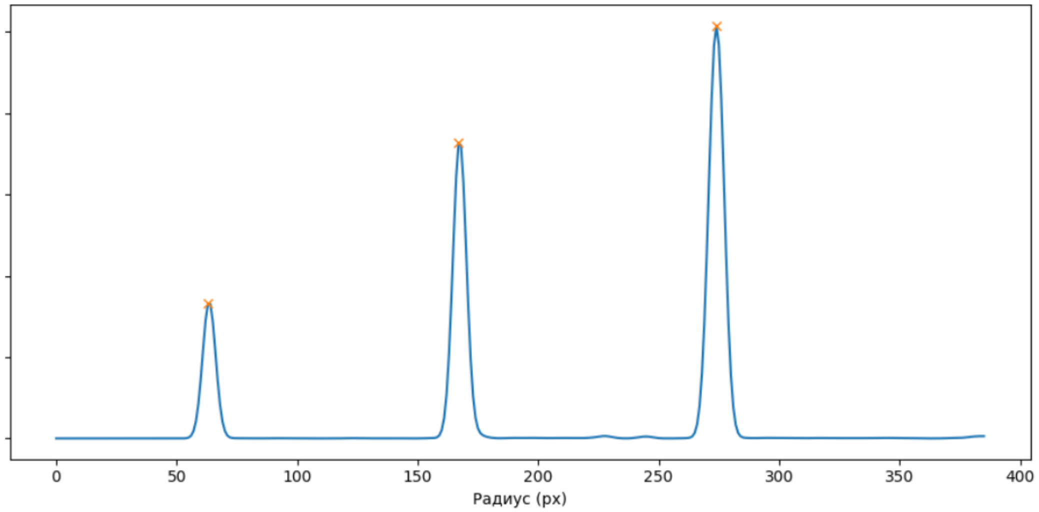
RADIUS DETECTION METHOD

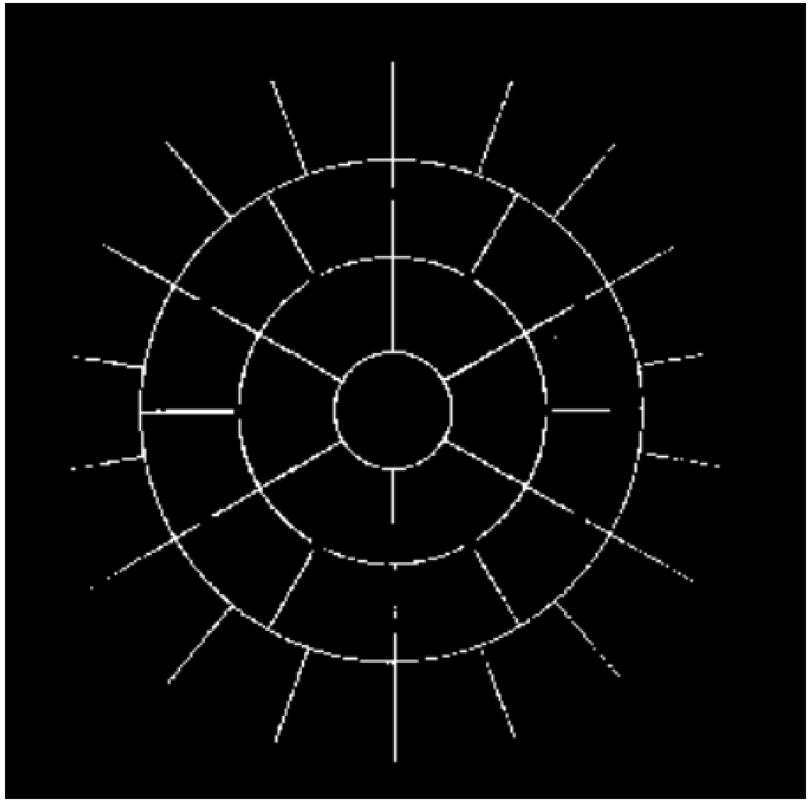
Select edge pixels with radial-gradient alignment ($|\cos(\theta)| > 0.8$)

Compute radii $r = \sqrt{(x-x_0)^2+(y-y_0)^2}$

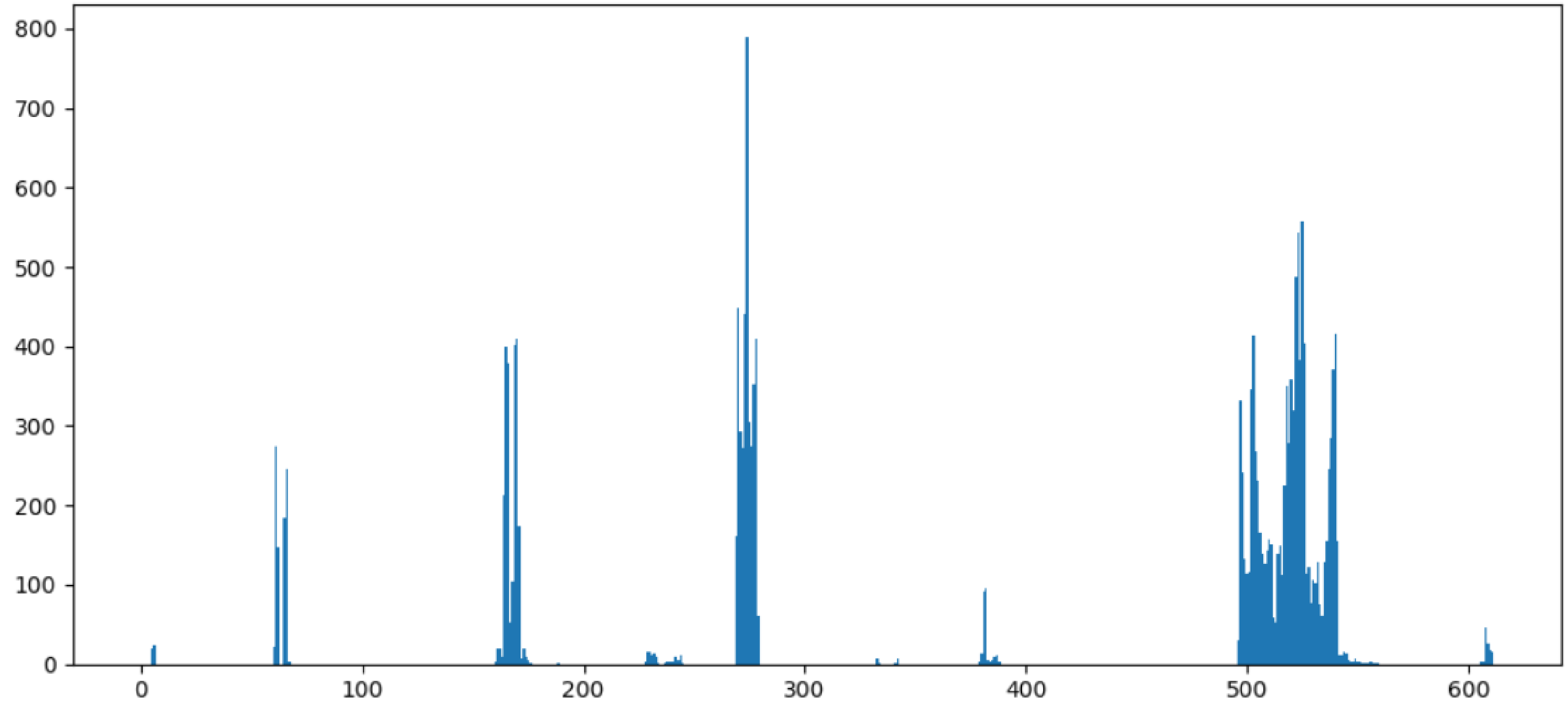
Histogram radii with 1 px bins

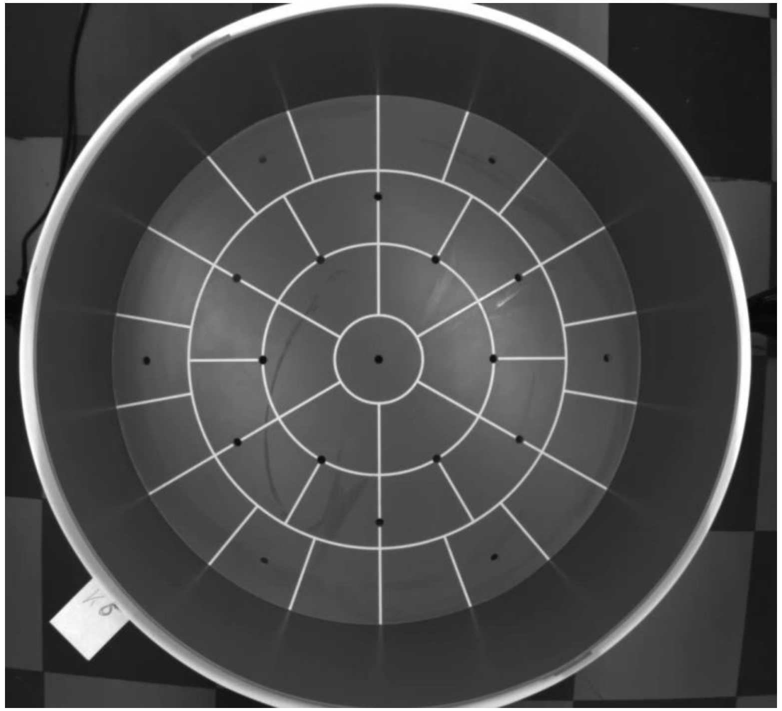
Gaussian-smooth histogram and detect peaks



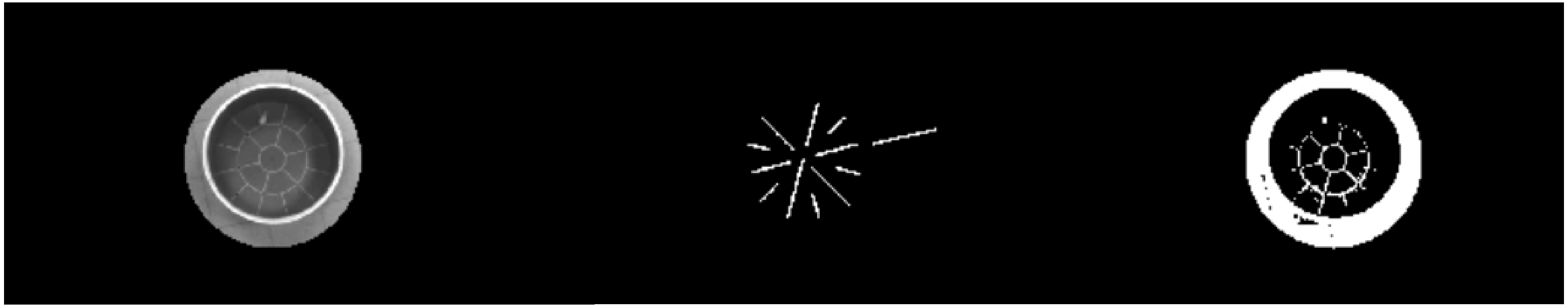


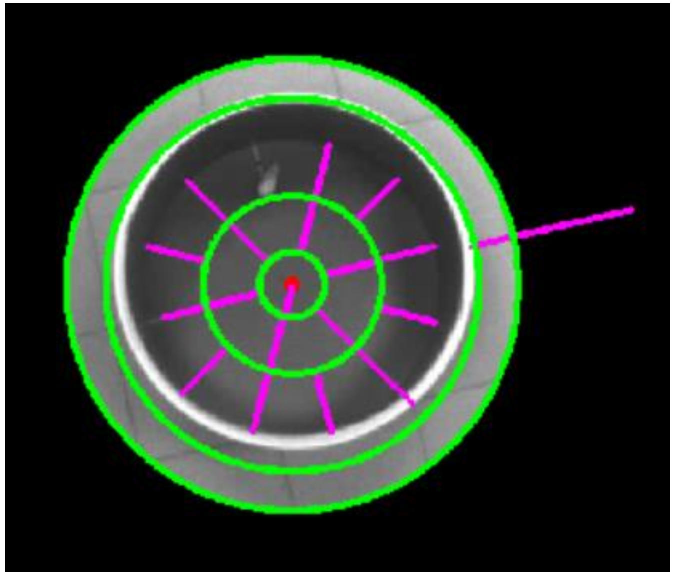
WHY DO WE NEED BINARY MASKS?

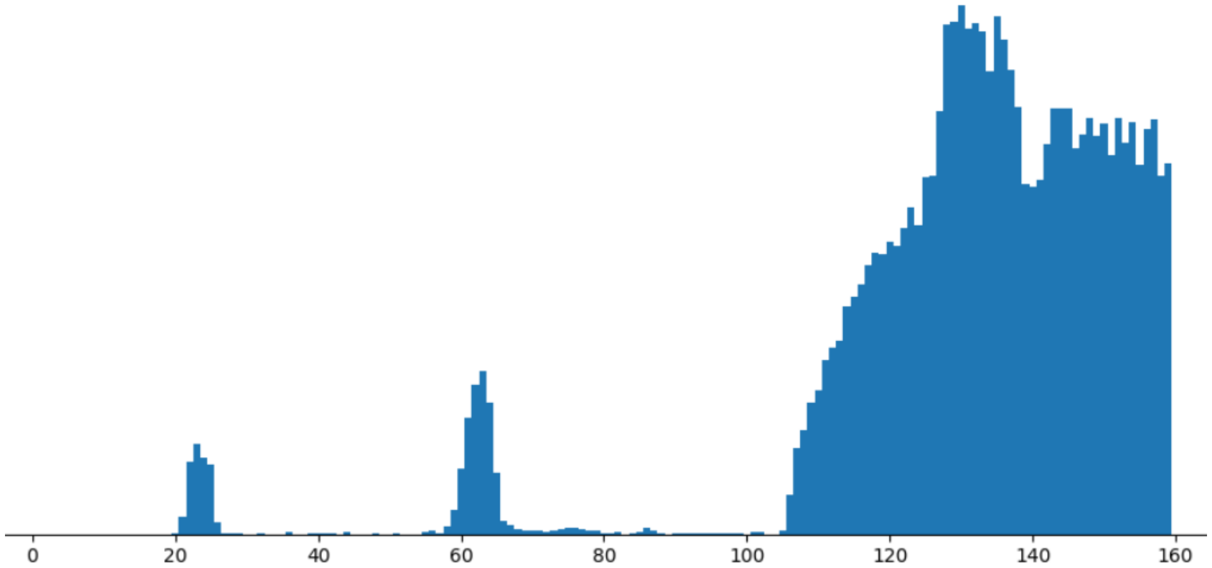




LOOKING BACK: WHY WE NEED IQR STEPS?







SECTOR DETECTION

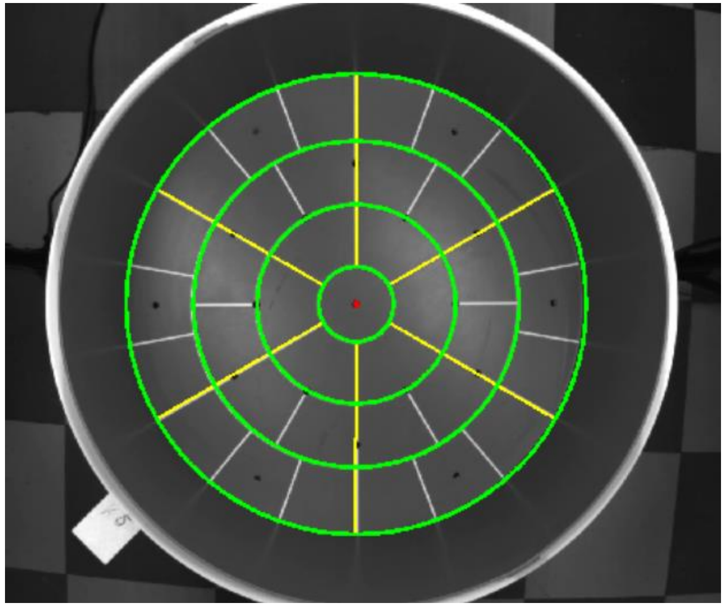
Goal: Find top 3 dominant directions of active pixels

Compute angles $\theta = \text{atan2}(y0-y, x-x0)$ in $[0,360)$

Convert to $\theta\_mod \in [0,180)$

Histogram with 4° bins and select top 3 bins

Add opposite angles (+180°) for directional coverage

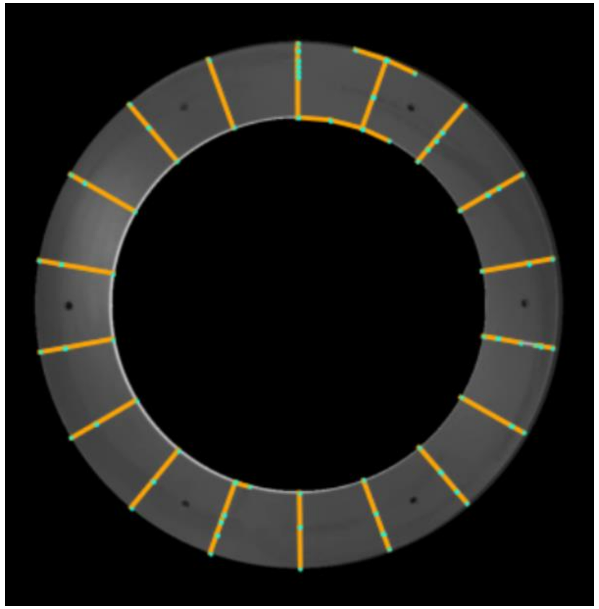


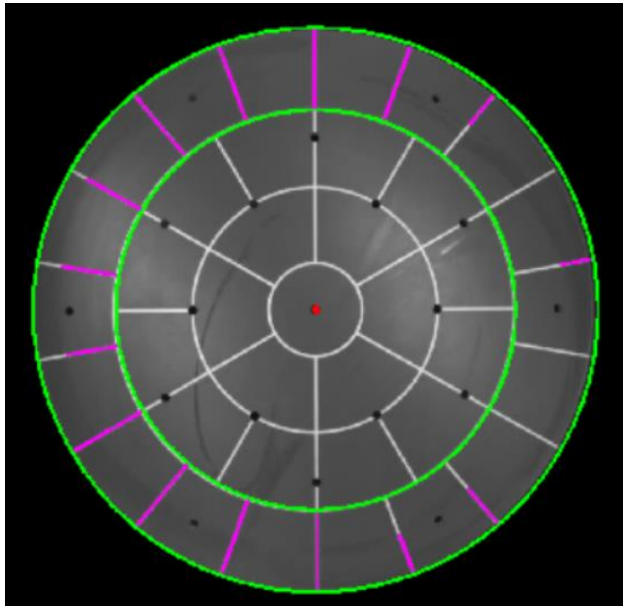
WHY TOP-3 DIRECTIONS?

Captures the main structural alignments

Robust against missing or spurious segments

Ensures detection of key arena axes and repeatable patterns

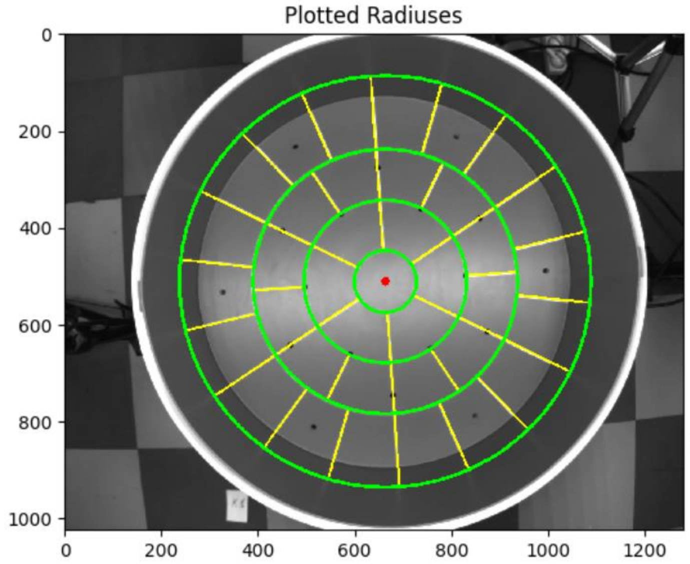


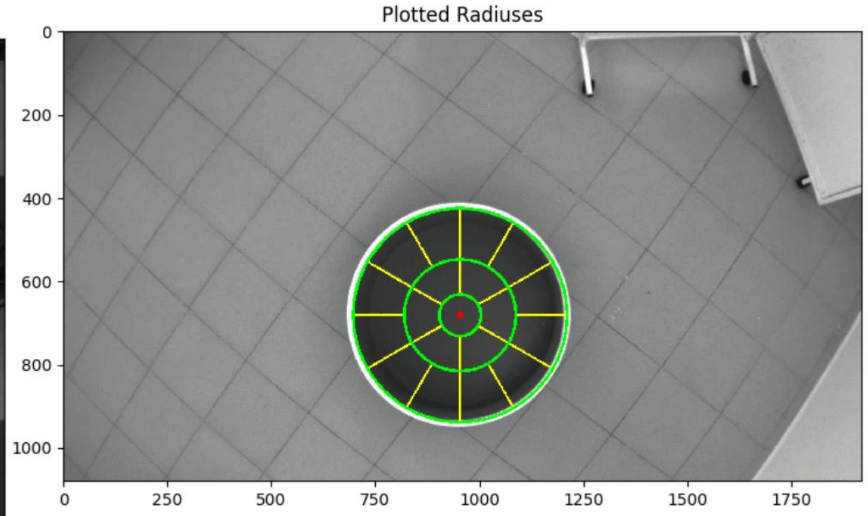


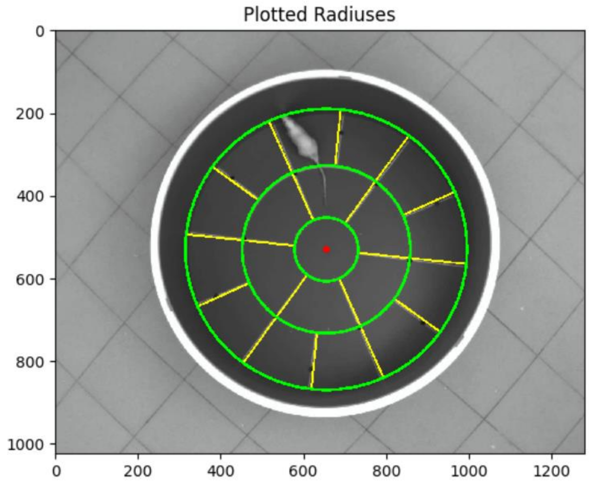
FINAL MARKUP STEP: RESULTS

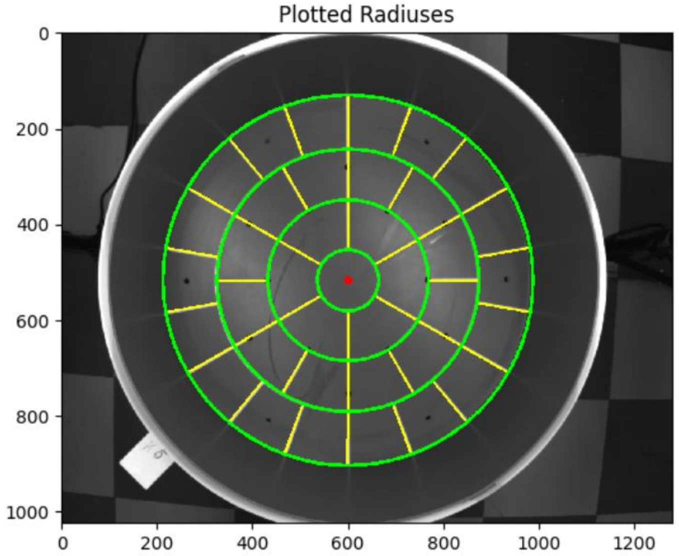
Save markup
to JSON


```
{
    "center": [
        599.0,
        517.0
    ],
    "radii": [
        63.0,
        167.0,
        274.0,
        387.6681823730469
    ],
    "rays": [
        30.0,
        90.0,
        150.0,
        210.0,
        270.0,
        330.0
    ],
    "markup": [
        {
            "index": 0,
            "r_inner": 0,
            "r_outer": 63.0,
            "angles": []
        },
        {
            "index": 1,
```

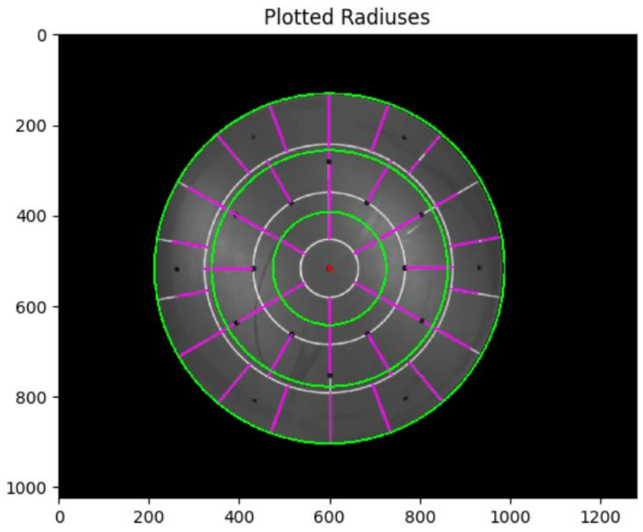


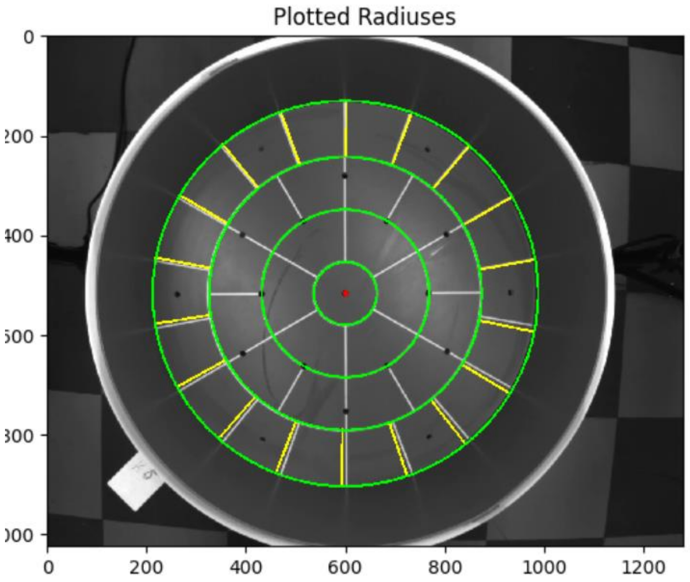




WHAT COULD GO WRONG?

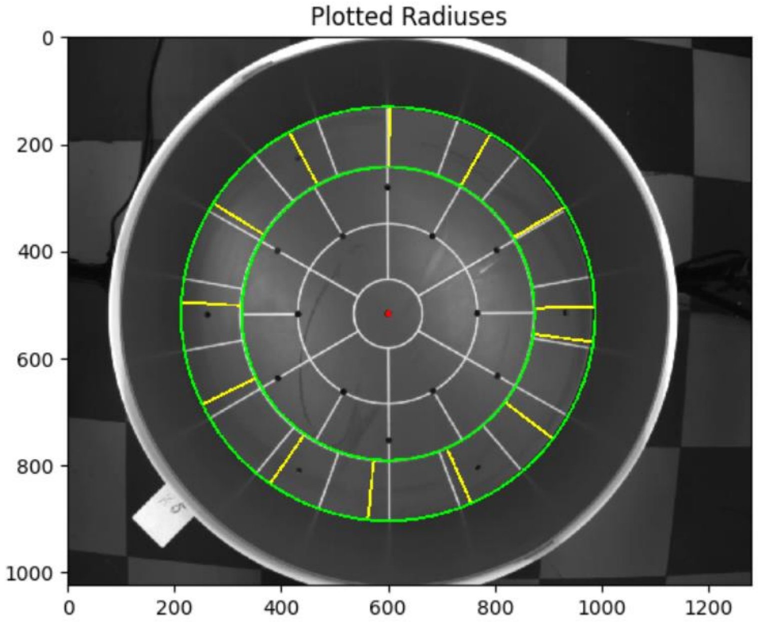
Errors in determining the concentric radii and their count





Precision of determining angles

Missed sectors and rings



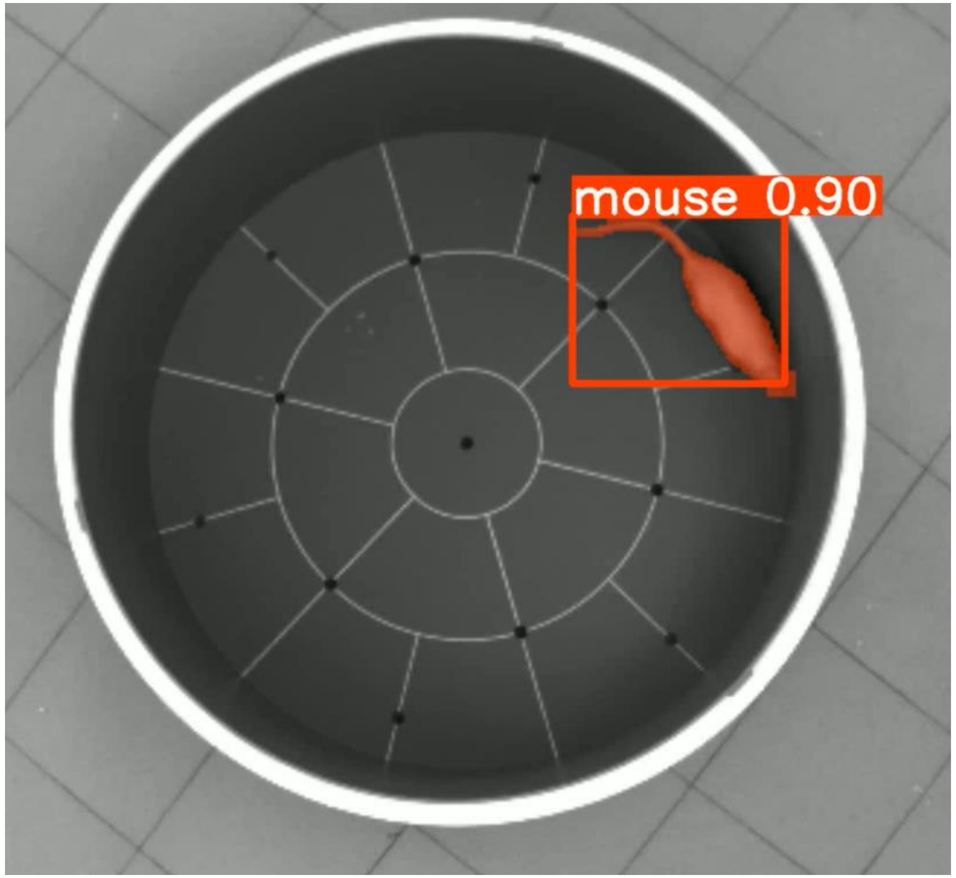
DETECTION? TRACKING?

Segmentation instead Detection.

Tracking by detection.

Ultralytics YOLOv11-seg [https://docs.ultralytics.com/ru/tasks/segment/]

~ 300 marked images

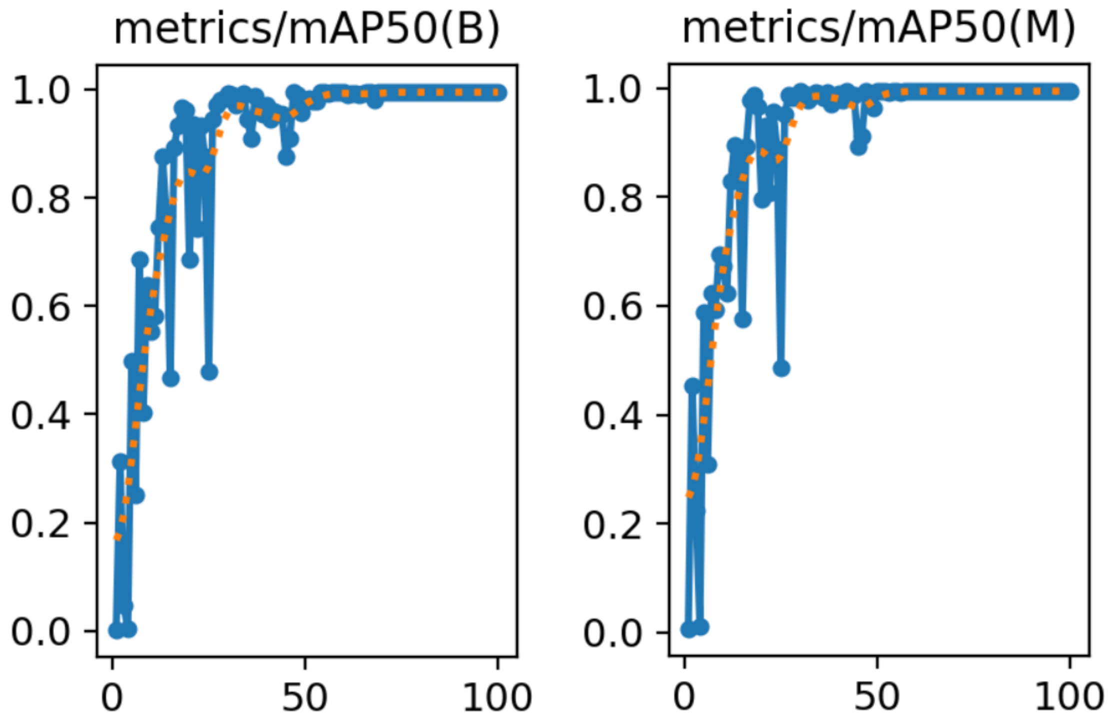


SEGMENTATION! TRACKING-BY-DETECTION.

mAP50 (Mean Average Precision@IoU = 0.50) – the average area under the Precision–Recall curve at an IoU threshold of 0.5

mAP50(B) – computed over bounding boxes; evaluates how accurately the model localizes objects with rectangles (reflects the effectiveness of object detection and localization)

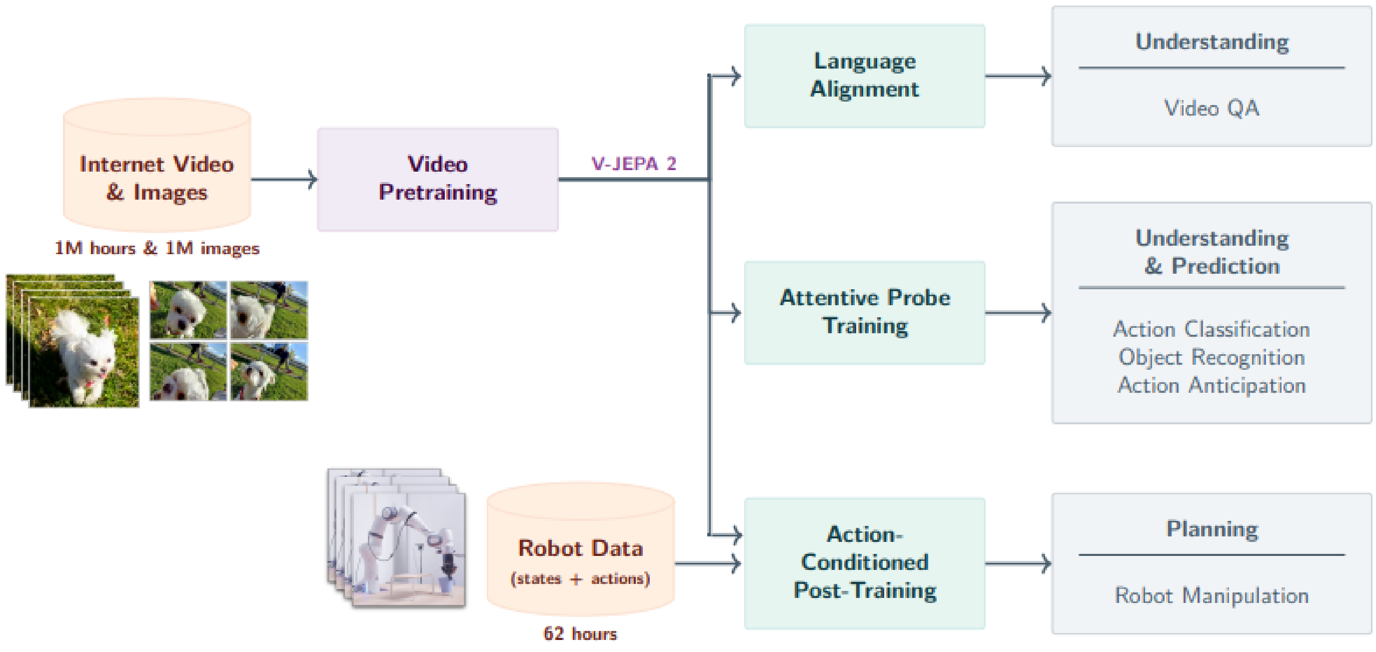
mAP50(M) – computed over segmentation masks; measures the pixel-level correspondence quality between predicted and ground-truth regions (reflects the precision of their contour segmentation)



TODO: ACTION DETECTION?

V-JEPA 2: Self-Supervised Video Models Enable Understanding, Prediction and Planning [https://arxiv.org/pdf/2506.09985] (11/06/25)

* Zero-shot Robot Control
* Strong video-based embeddings



THANK YOU FOR YOUR ATTENTION!