Final Project Initial Report

`Deep Learning Course ,TAU, 0510-7255, Spring 2025`

Real-Time Lizard Tracking for Screen-Proximal Behavior

Students: Dror Hazan `301263513` , Bareket Damari `315394874`

Project summary: A real-time computer-vision system that tracks lizards in a rectangular arena, estimating position, distance to the screen, and head orientation. The system will publish these signals at low latency for use in experiments.

Milestones:

- **Data ready & baseline training** → *Initial report* :
    - obtain data
    - label data
    - initial model
- **Latency + accuracy pass** → *Mid-project checkpoint* (internal).
- **Live demo + final report/code** → *Presentation Day / Final submission*.

Data ready & baseline training

1) obtain data

Data been obtained from multiple arenas with a top/front view, captured by two types of cameras*, at 60 fps, producing mono videos with varied illumination and animal posture.

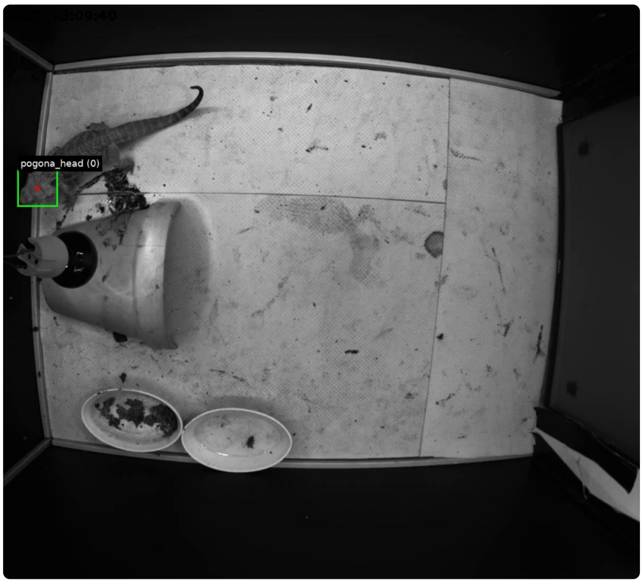
***Cameras**: Firefly FFY-U3-16S2M-DL, FLIR Blackfly BFS-U3-16S2C-CS with fixed lenses (6 mm)

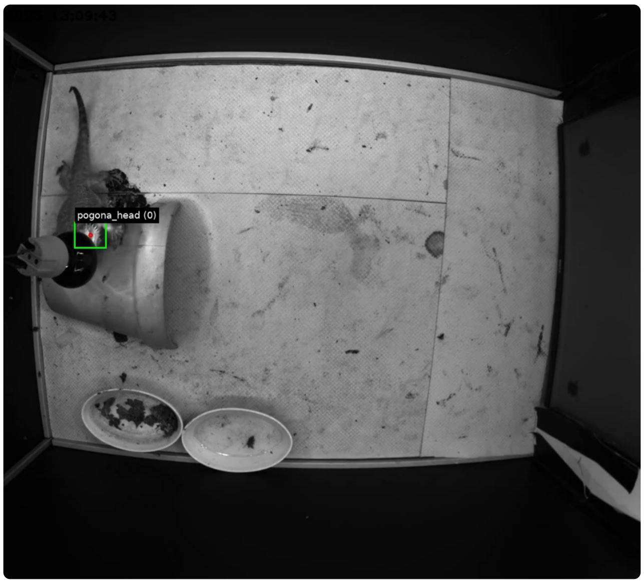
2) label data

Tool: LabelImg (free, offline). Labels are in **Ultralytics YOLO** format.

- For each image, a `.txt` with one line per object:
  `class x_center y_center width height` where all values are **0–1** (relative to image size).
- The box marks the center (x, y point) of the lizard head - class 0: pogona_head

samples:





Organize data:

folder layout - data split

```
dataset/
  images/
    train/  ...JPG/PNG
    val/
  labels/
    train/  ...YOLO .txt (one per image)
    val/
```

Preparing an Ultralytics-style dataset config (`config.yaml`):

```
path: /pose_estimation/pogona_head/dataset
train: images/train
val: images/val
names:
  0: pogona_head
```

Split sizes: train: **2700** files; val: **700** files.

3) Baseline model & runnable training script

Model: YOLOv11 (detector). Starting with a small backbone for speed, then we will scale up.

Notes:

- Current labels are **head boxes** only. For **heading angle**, we will either (a) add **snout + head-center** key-points (pose model), or (b) estimate a coarse proxy from short tracklets (direction of motion) as an interim measure.
- Distance-to-screen is computed in **cm** using the arena calibration (homography) file.

4) Next steps

- Evaluate the baseline & scale up and freeze.
- Decide on **pose** scope (add snout + head-center).
- Estimate a **heading proxy** from the **direction of motion**.
- Integrate the **publisher** and **real-time saving** for experiments.
- Full **evaluation** on val/test (det/pose; distance RMSE; heading MAE)
- Prepare **demo** video, slide, final report; repo cleanup

• • •