Automating Insect Detection in Videos with Metacentrum

Jan Filip

Insect Communities Research Group
Charles University, Prague

Principal Investigator: Robert Tropek

Design and Development: Petr Chlup

Deployment and Testing: Jan Filip, Dominik Anýž

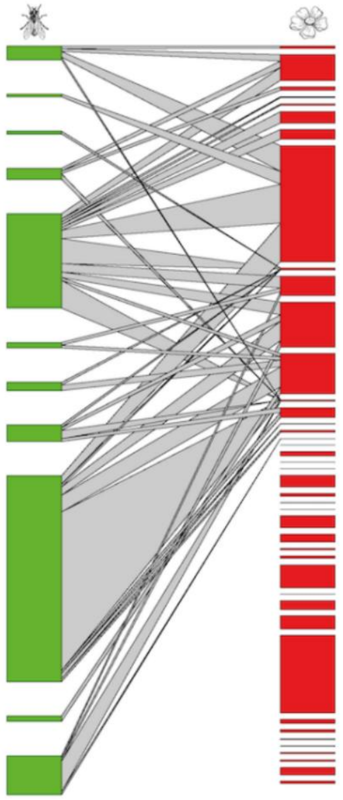
cesnet metacentrum

INSECT COMMUNITIES
INSECT COMMUNITY ECOLOGY GROUP

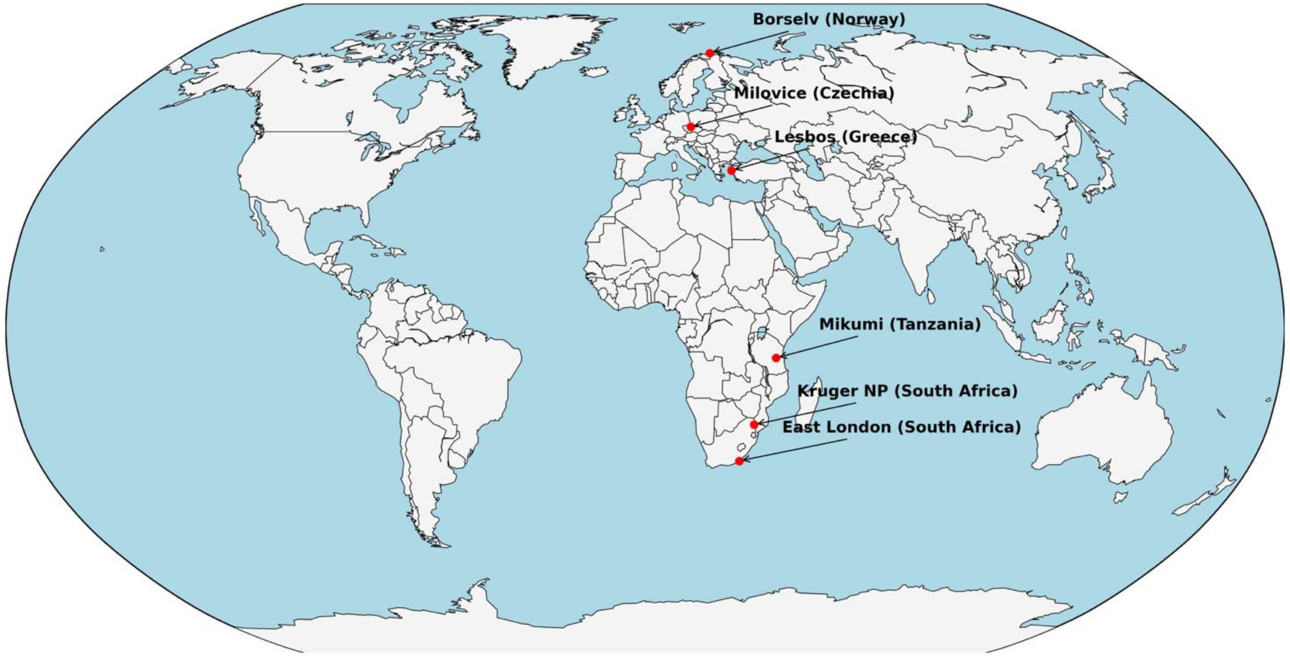
UNIVERZITA KARLOVA

DetectFlow
Object Detection Pipeline

Scientific Motivation



- We study **pollination networks** (who pollinates whom, how often)
  - help quantify ecosystem interactions and specialization
- Current project: **latitudinal gradient** (Junior Star GAČR, 2021-2025)



Advertisement
a) sensory biases
b) reward motivation

Pollen Transfer
a) anthers -> pollinator
b) pollinator -> stigma

The Data Challenge

- Method: **CCTV cameras on flowers**
  - recording full 24h cycles across many flowering plant species

Advantages: Captures **diurnal** and **nocturnal** activity (infrared at night)

Ensures **statistical power** (huge recording times)

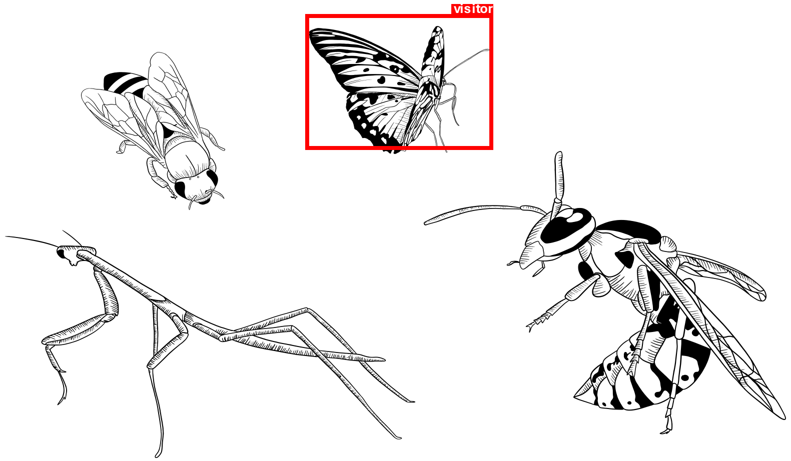
Deployable at **ecosystem scale** (30-50 flowering species per site)

Drawbacks: **10 years** of accumulated recording time

~3,500 full-day recordings = 350,000 video files = **~8 billion frames**

Video **quality varies** (day vs night, movement of flowers, small insects)

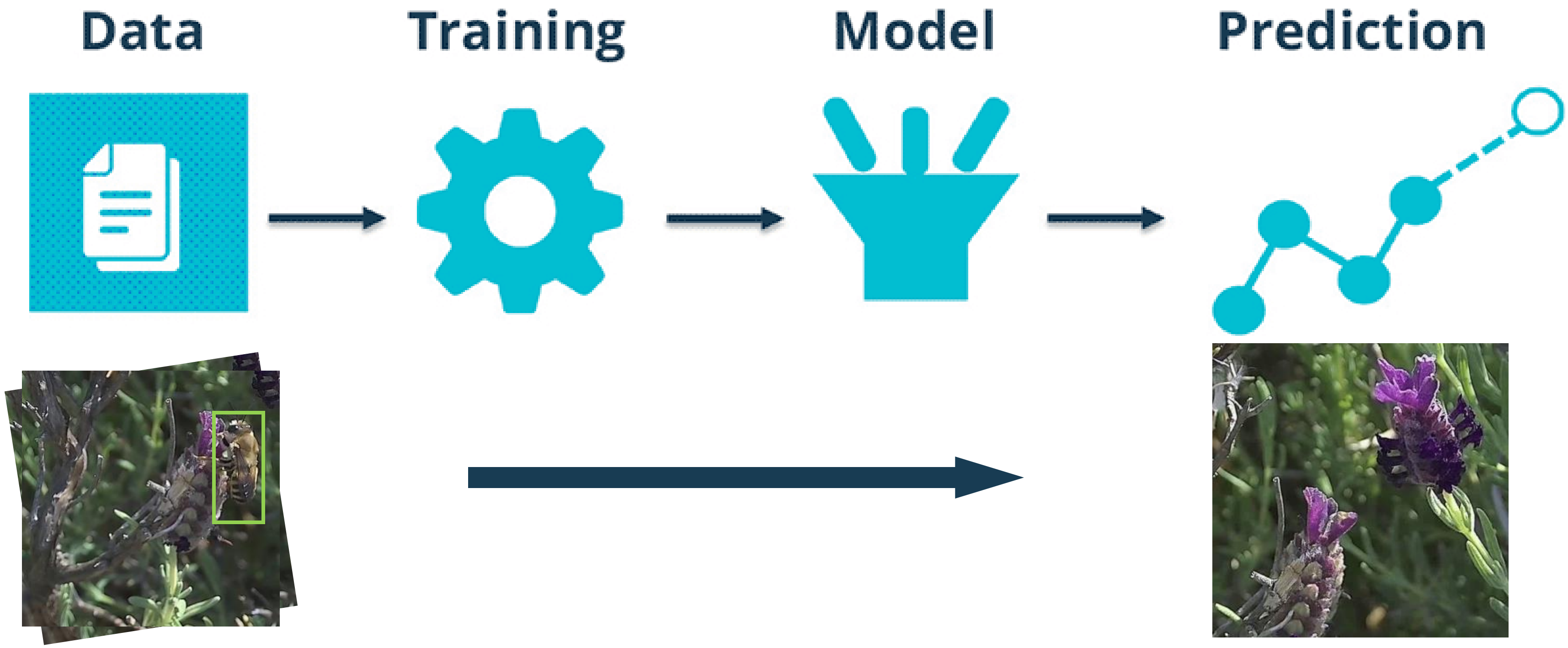
Data Processing Goals



- **Raw videos** -> detect insect visits

  locate insects relative to flowers

  extract **visit-level records** (arrival/departure)



Data → Training → Model → Prediction

Model Training

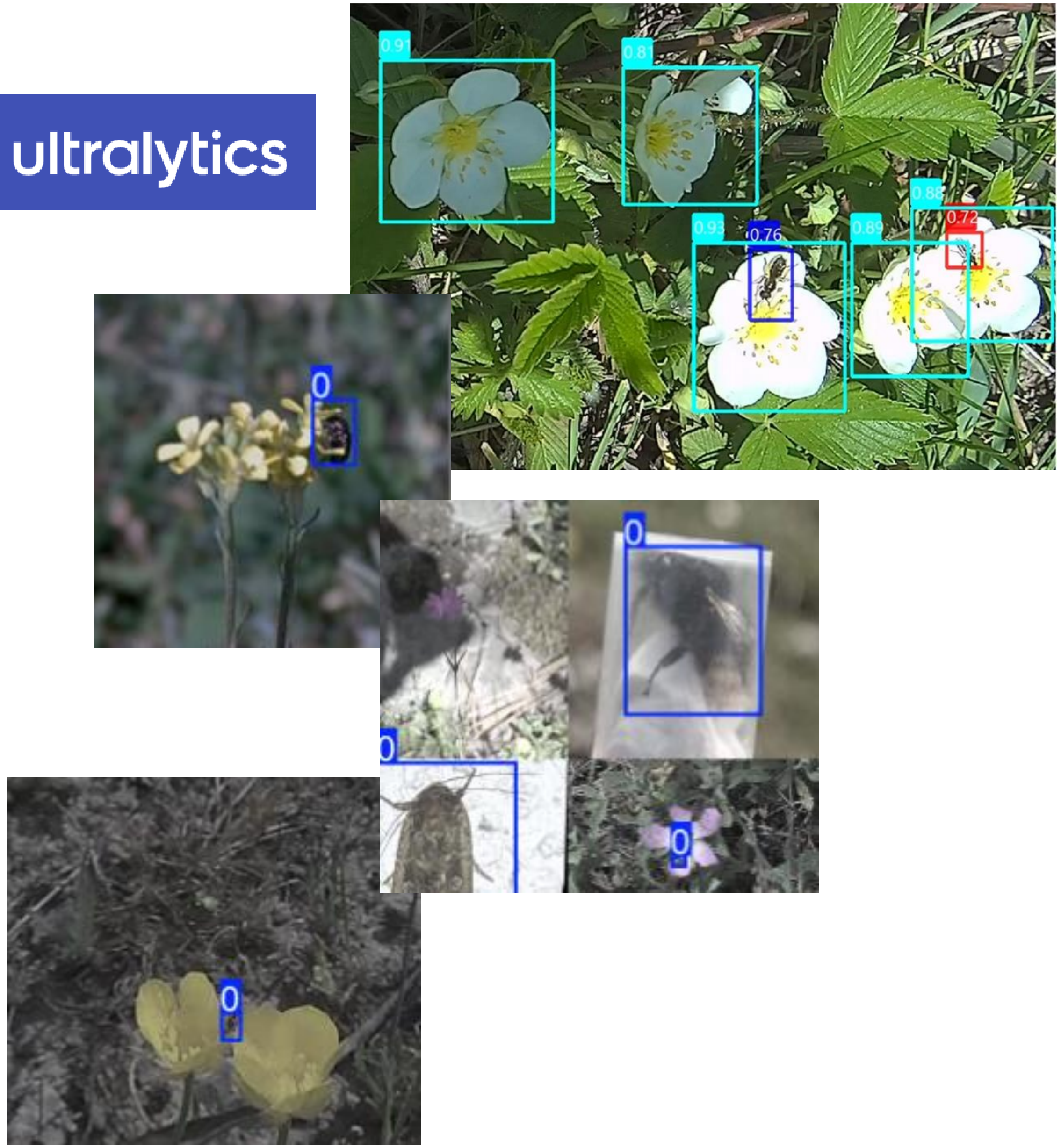
- Chosen algorithm: **YOLO** (Ultralytics)

- **Training data**:
  - Manually annotated frames
  - Semi-automated labeling
  - Enriched with internet insect images

- Strategy: pretrain „**insect expert**“

  -> fine-tune on flower videos

- **Flower** x **insect** model



Why MetaCentrum?

- Needed **HPC resources** for:

  Large training runs (GPUs)

  Huge inference workload (CPUs)

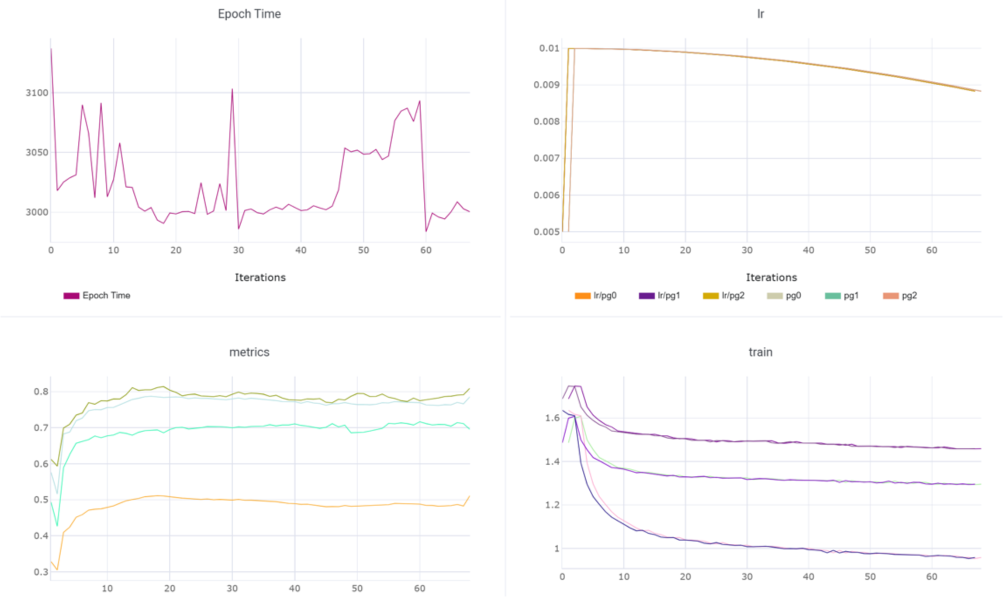
  Long-term S3 storage of TBs of video

  Jupyter Notebooks with processing resources

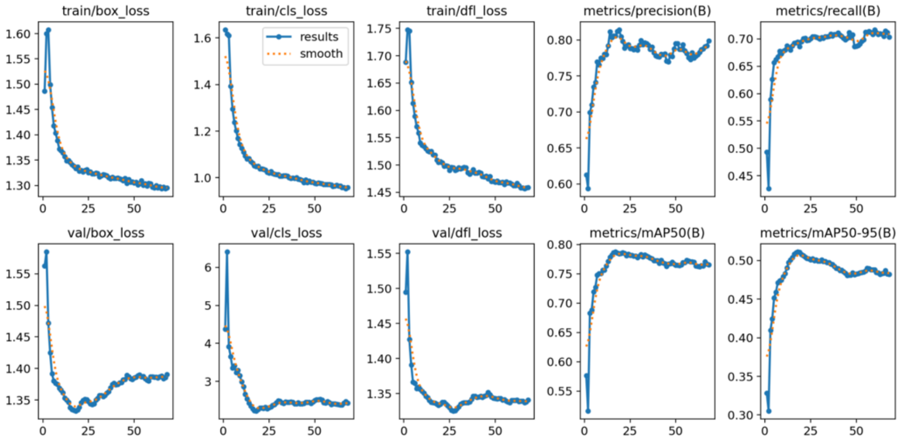
- MetaCentrum **enabled** scaling to billions of frames

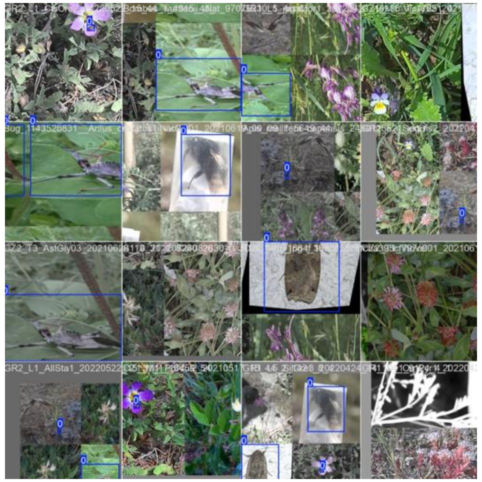
Jupyter -> OnDemand -> PBS jobs -> GPUs, CPUs, Singularity

Training on GPUs



- Used GPUs nodes with high **VRAM (~64 GB)** (**adan**, **galdor** – gpu_long)
- Avoided AI-specific cluster (queues too long)
- Built and ran models in Singularity **containers**
- Solved CUDA/package version issues by pinning dependencies

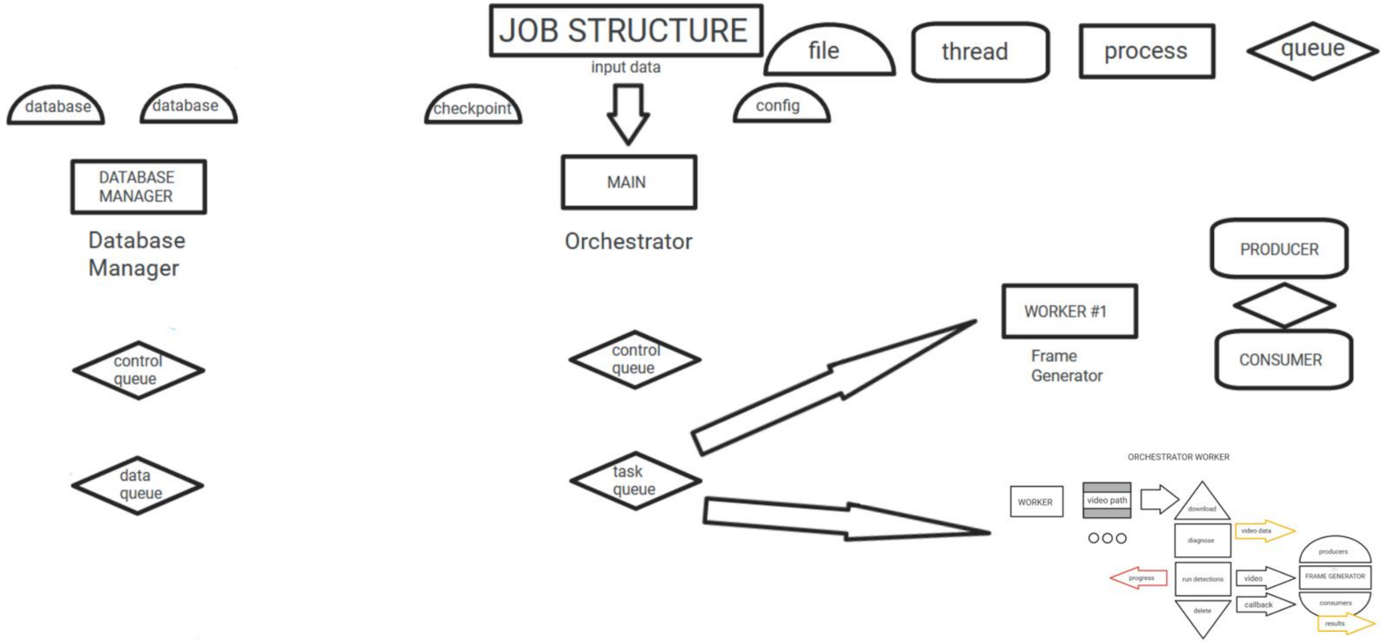


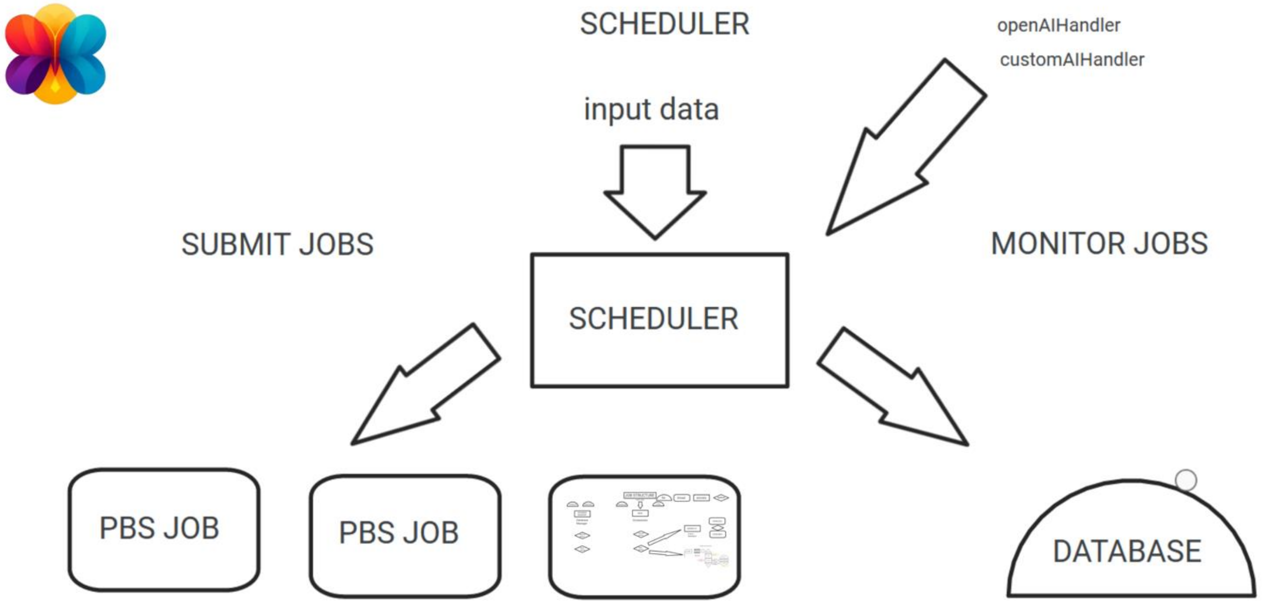


Inference on CPUs

- **PBS** job design:
  - Each job = batch of tens of vic
  - Dozens of jobs in parallel -> h

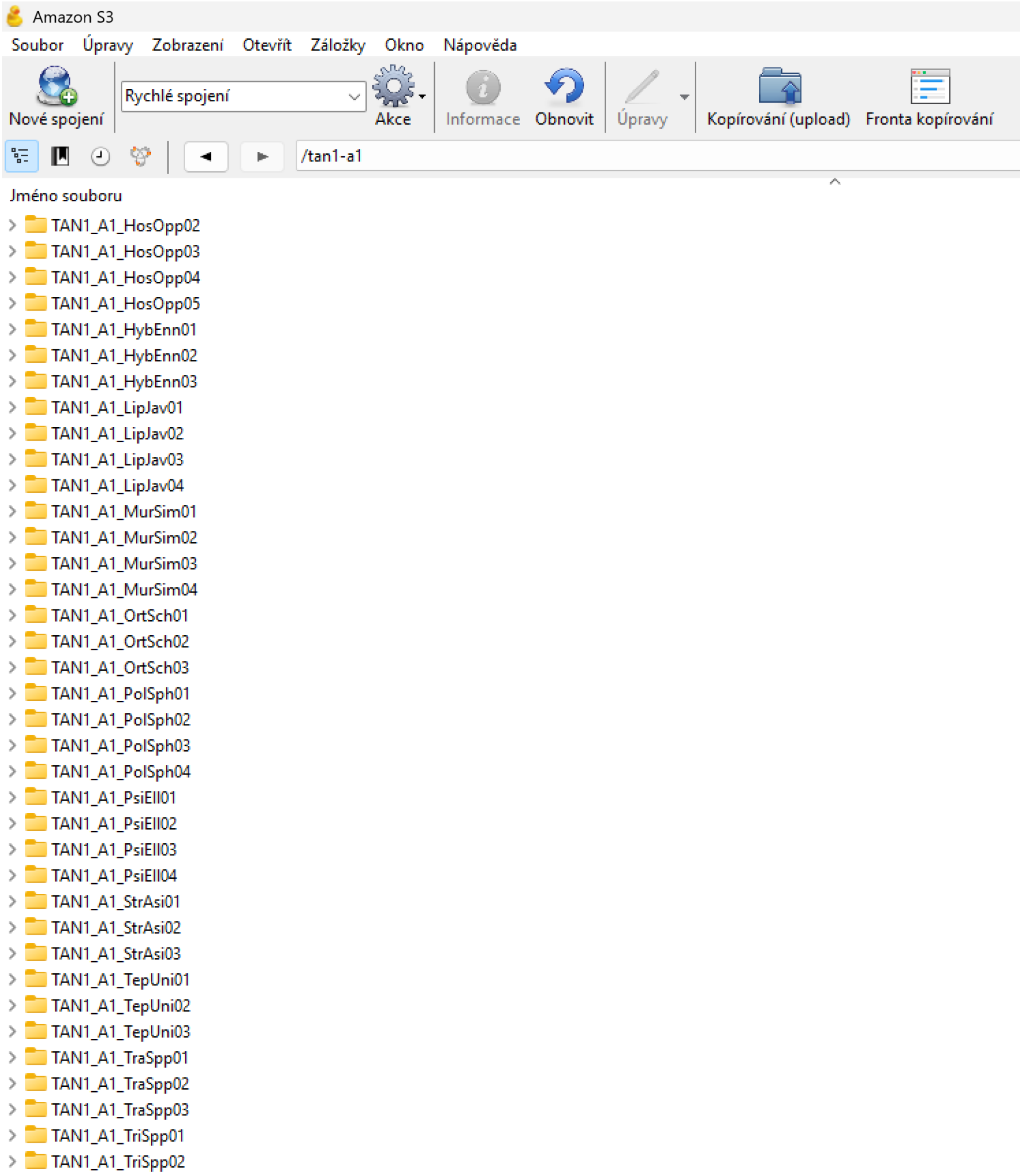
- Jobs restartable to handle failure
- All large-scale inference ran on (**only**:
  - GPU queues too slow
  - CPU jobs easy to parallelize





Storage & Data Handling

- **S3 storage**: long-term archive, easy upload
- Local cluster storage only for temporary files
- Essential for handling terabytes efficiently
- Perfect protection through **credentials**
- **Shareable** in virtual organizations, work groups

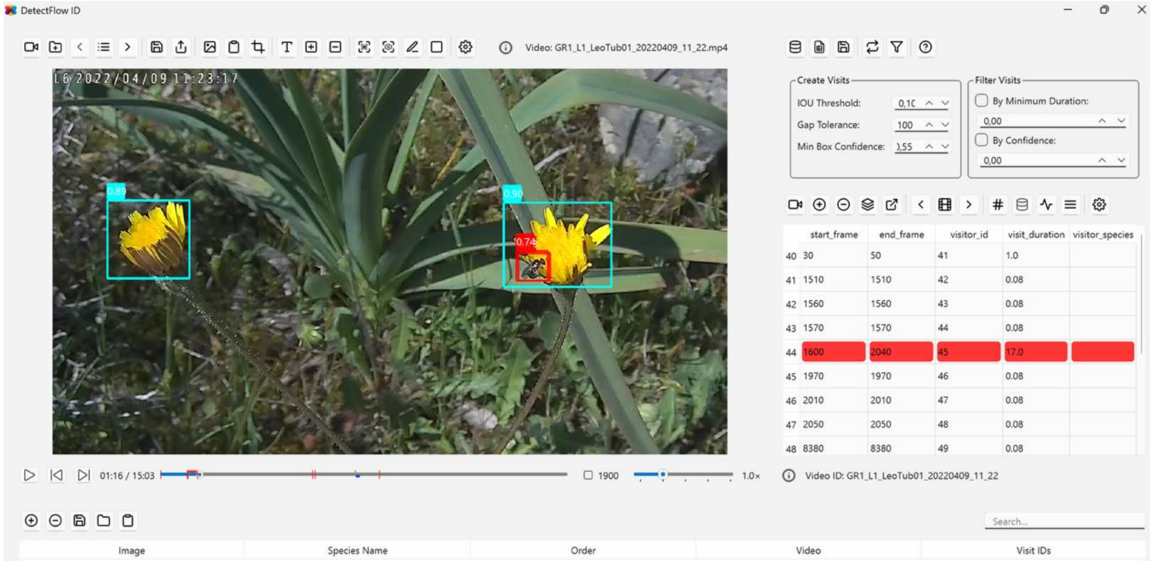


/tan1-a1

```
TAN1_A1_HosOpp02
TAN1_A1_HosOpp03
TAN1_A1_HosOpp04
TAN1_A1_HosOpp05
TAN1_A1_HybEnn01
TAN1_A1_HybEnn02
TAN1_A1_HybEnn03
TAN1_A1_LipJav01
TAN1_A1_LipJav02
TAN1_A1_LipJav03
TAN1_A1_LipJav04
TAN1_A1_MurSim01
TAN1_A1_MurSim02
TAN1_A1_MurSim03
TAN1_A1_MurSim04
TAN1_A1_OrtSch01
TAN1_A1_OrtSch02
TAN1_A1_OrtSch03
TAN1_A1_PolSph01
TAN1_A1_PolSph02
TAN1_A1_PolSph03
TAN1_A1_PolSph04
TAN1_A1_PsiEll01
TAN1_A1_PsiEll02
TAN1_A1_PsiEll03
TAN1_A1_PsiEll04
TAN1_A1_StrAsi01
TAN1_A1_StrAsi02
TAN1_A1_StrAsi03
TAN1_A1_TepUni01
TAN1_A1_TepUni02
TAN1_A1_TepUni03
TAN1_A1_TraSpp01
TAN1_A1_TraSpp02
TAN1_A1_TraSpp03
TAN1_A1_TriSpp01
TAN1_A1_TriSpp02
```

Outcomes

- HPC saved years of manual work -> project otherwise impossible
- Processing of **~8 billion frames** feasible
- Produced **visit-level dataset** from raw video
- Custom GUI app allows refining detections into final data



Video: GR1_L1_LeoTub01_20220409_11_22.mp4

Create Visits

IOU Threshold: 0,1(

Gap Tolerance: 100

Min Box Confidence:),55

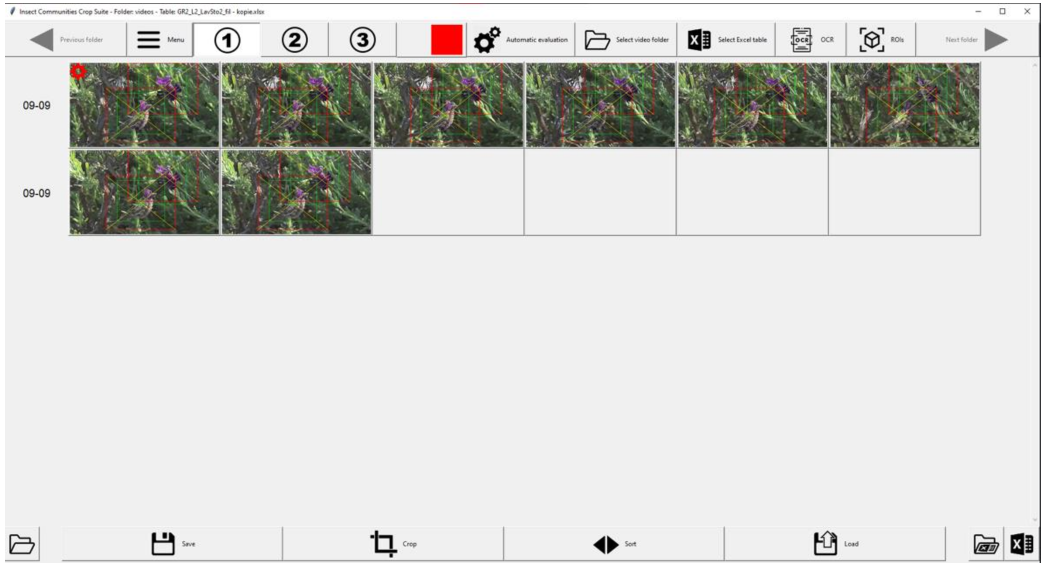
Filter Visits

□ By Minimum Duration: 0,00

□ By Confidence: 0,00

	start_frame	end_frame	visitor_id	visit_duration	visitor_species
40	30	50	41	1.0	
41	1510	1510	42	0.08	
42	1560	1560	43	0.08	
43	1570	1570	44	0.08	
44	1600	2040	45	17.0	
45	1970	1970	46	0.08	
46	2010	2010	47	0.08	
47	2050	2050	48	0.08	
48	8380	8380	49	0.08	

Video ID: GR1_L1_LeoTub01_20220409_11_22



Lessons Learned

- **Containerization** critical for reproducibility
- **S3 storage** indispensable for large video workflows
- **PBS job parallelization** allowed scaling
- **Standardised** sampling crucial for easier AI learning

Acknowledgements

INSECT COMMUNITIES
INSECT COMMUNITY ECOLOGY GROUP

- Many thanks metacentrum cesnet for infrastructure and help!
- Thanks to the whole team and worlwide collaborators!
- Contact: **jan.filip55@gmail.com**

Open to work ☺