Efficient Computer Vision Models for Silkworm Feeding Prediction and Habitat Analysis

Leonardo Mariut - 1986191
Mohamed Zakaria Benjelloun Tuimy - 2190452
Computer Vision
Sapienza University of Rome

Table of contents

1 The problem & State of The Art approaches

2 Proposed Methods

3 The Dataset

4 Experimental Setup

5 Models Evaluation

6 Conclusion

Automating Silkworm Feeding and Habitat Analysis

Binary Classification

The goal is to test and fine-tune lightweight models for classifying whether feeding is needed, based on images of the rearing beds. This enables automated monitoring of silkworm activity and feeding status, reducing the need for manual inspection.

Segmentation

Apply unsupervised segmentation techniques to differentiate silkworms, mulberry leaves, and background within rearing bed images. The objective is to analyze habitat composition without ground truth labels, enabling scene understanding through semantic or panoptic separation.

State of the Art: Current Research Approaches

Efficient Model Architectures

- *MobileViT*: Combines convolutional neural networks with vision transformers to process local and global features efficiently, suitable for mobile vision tasks.
- *EfficientNetV2*: Optimizes training speed and parameter efficiency through neural architecture search and progressive image scaling, achieving faster training and reduced model size
- *RepNeXt*: Integrates multi-scale feature representations with serial and parallel structural reparameterization to enhance network depth and width without compromising inference speed.

Unsupervised Segmentation Techniques

- *U2Seg*: Employs self-supervised learning and clustering to generate pseudo-labels for semantic, instance, and panoptic segmentation tasks, eliminating the need for manual annotations.
- *CutLER:* Unsupervised object discovery: clusters self-supervised features (e.g., Masked DINO) across scales to cut out foreground objects without labels.
- *STEGO:* Learns pixel-wise embeddings from self-supervised features and groups them with contrastive losses to form coherent semantic regions.
- *Rossetti et al. (2023)*: Utilizes local-global patch matching and area balancing to predict categories and shapes in semantic segmentation without supervision.

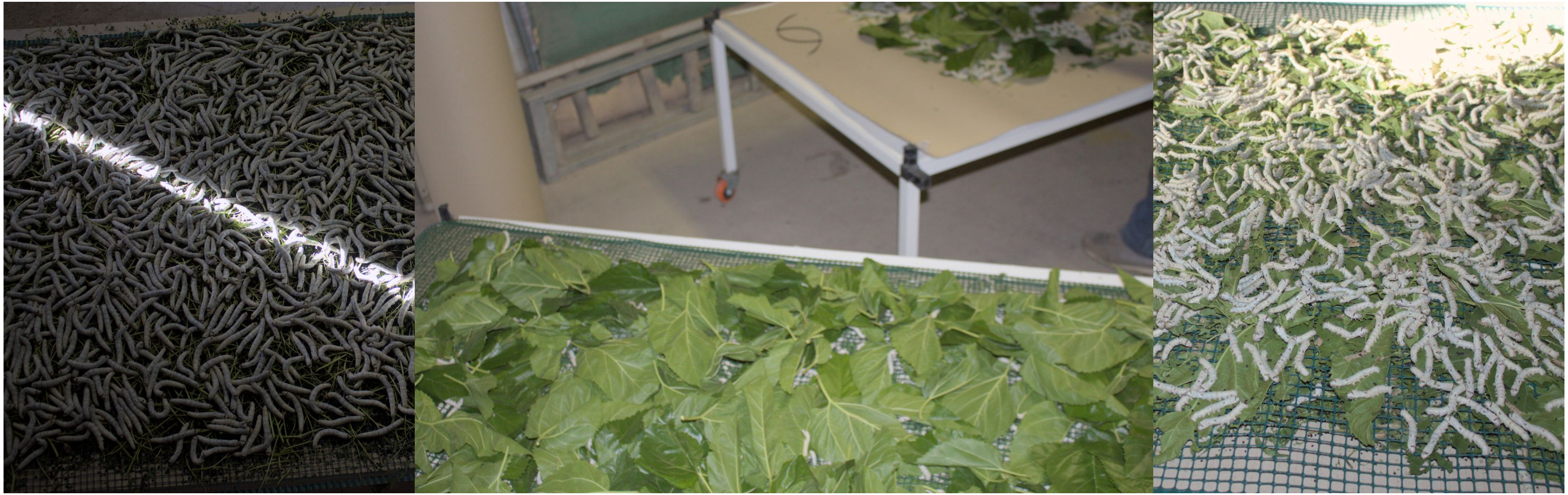
Proposed Methods

- We began with a simple **DINOv2**-based notebook: patch features **(37x37)** were clustered spectrally into three groups, cluster 0 was hard-coded as “silkworms”, and worms were split with a distance-transform + watershed step.
- We then removed that hard-coding with a heuristic upgrade: the same **DINO** features and clustering were kept, but clusters were auto-labeled via *HSV statistics* (lowest saturation → worms, hue≈60 → leaves), giving a fully label-free semantic map.
- Finally, we unified instance and semantic cues : one branch over-clustered patches (~150) to get class-agnostic instances, another produced 3 semantic clusters; each instance was assigned to a class by majority overlap, yielding panoptic-style masks. After all of this the results were not the best, and all unsupervised methods were unsatisfactory.
- In response to these limitations, we developed a heuristic self-supervised semantic segmentation pipeline based on custom image preprocessing and color thresholding.
- For classification, all three efficient models (**EfficientNetV2**, **RepNeXt**, **MobileViT**) performed sufficiently well out of the box and required minimal tuning.
- An additional simple **linear regression** on the generated masks has been developed for the binary classification task, *achieving comparable performance with the more complex models*.
- Finally, we have *distilled the segmentation knowledge from the heuristic approach* into **SegFormer**, obtaining overall decent performance in terms of class separation and visual coherence.

The Dataset: Rearing bed image collection

- ↪ The dataset consists of **1,351 images** of *silkworm rearing beds*, acquired under *varied lighting conditions, different camera angles, and random orientations*. The images present *substantial visual clutter* due to overlapping silkworms, leaf fragments, and irregular backgrounds.
- ↪ Image-level labels for feeding status provided in **0_data.csv** (binary).
- ↪ This variability introduces *significant challenges for segmentation,* particularly for unsupervised methods, which struggle to consistently separate silkworms, leaves, and background without annotated guidance.
- ↪ Generated masks from heuristic and self-supervised pipelines were used to support tasks such as fine-tuning in models like SegFormer.

The Dataset: difficult images



Experimental Setup: Tools, Environment, and Configuration

General configuration:

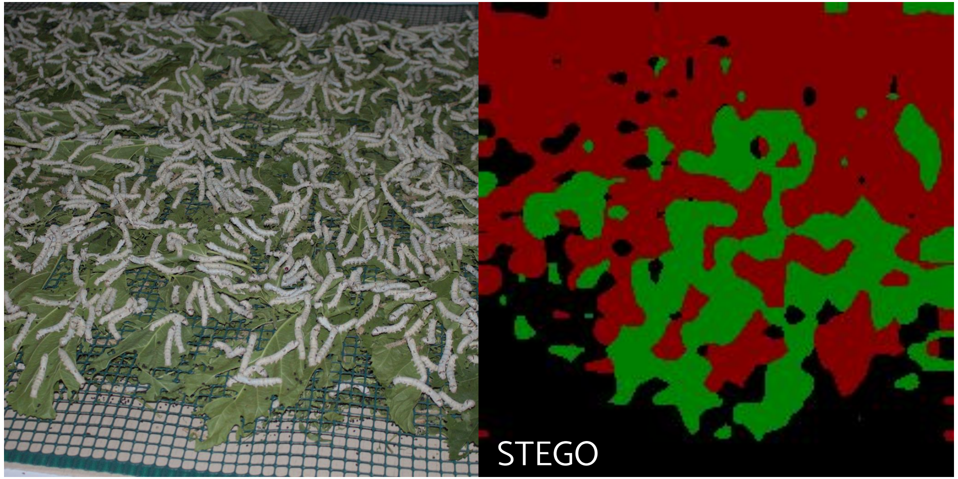
- ↪ The project is organized into **Python notebooks**, *executed locally via VSCode* using a custom Conda based Jupyter kernel.
- ↪ Image data is stored and accessed locally following the structure documented in the GitHub repository.
- ↪ Models with complex dependencies (**CutLER**, **STEGO**, **U2Seg**) were run in separate **Anaconda** environments from terminal, using the provided scripts, to ensure compatibility.
- ↪ All remaining components, including segmentation heuristics and classification models, were run in shared notebooks locally.
- ↪ **Hardware 1**: 32 GB RAM, Nvidia GTX 1660 Ti Mobile (6 GB).
- ↪ **Hardware 2**: 32GB RAM, Nvidia RTX 3000 Ada Mobile (8GB).

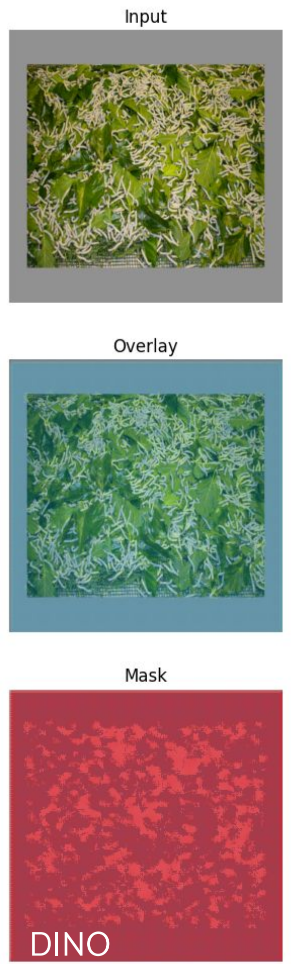
Experimental Setup: Tools, Environment, and Configuration

Models configuration:

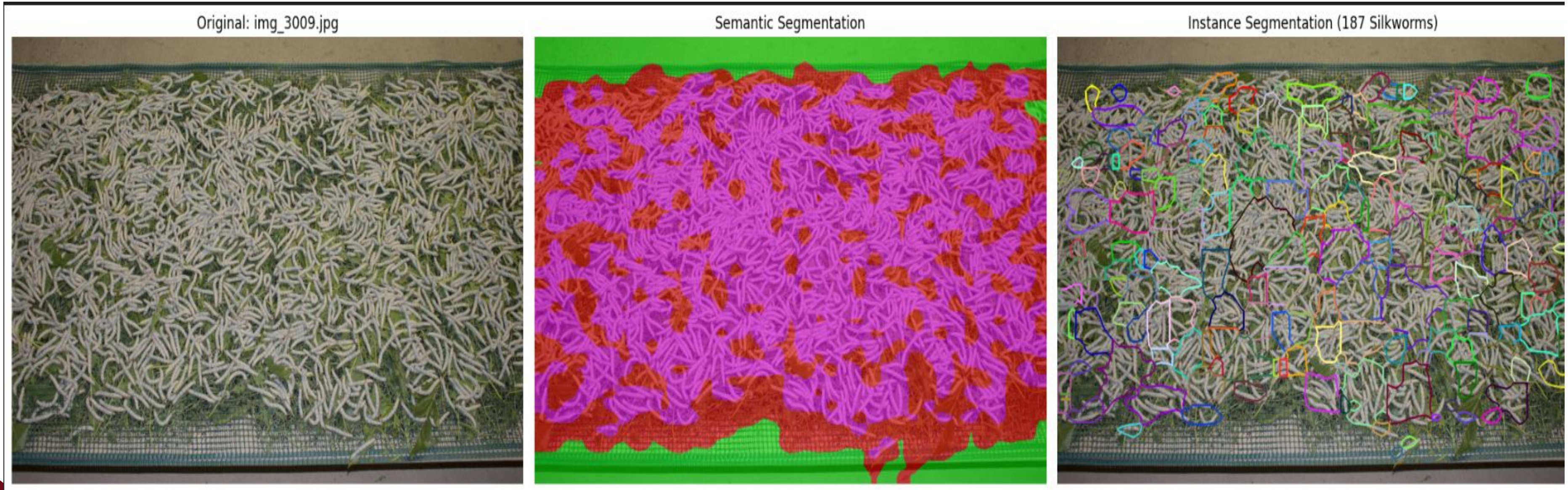
- ↪ **Heuristic DINO** pipeline and **U2SegUnified** ran fully unsupervised (spectral clustering + watershed; $\tau \approx 0.05/0.12$, $k \approx 150$ for instances, $k=3$ for semantics).
- ↪ **SegFormer B0** was trained at 512×512 resolution, selected as a tradeoff between performance and spatial detail.
- ↪ **EfficientNetV2**, **RepNeXt**, and **MobileViT** were used with **pretrained backbones**; only a lightweight classification head was fine-tuned using a cross-entropy loss.
- ↪ **CutLER** and **STEGO** were tested with default or recommended hyperparameters
- ↪ **SegFormer** was fine-tuned using *heuristically generated pseudo-masks*.
- ↪ Due to the absence of ground truth, segmentation outputs were assessed qualitatively based on visual separation and consistency.

Segmentation Results



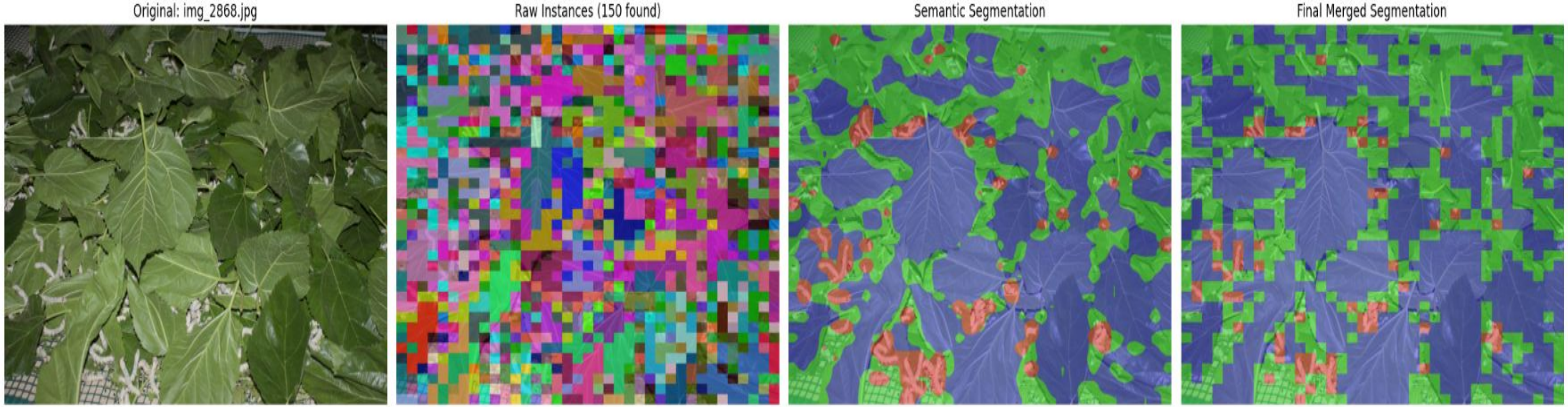


Segmentation Results 2: *Dino_segmentation_basic*



Params: τ=0.12, k=3, min_size=150, min_dist=12

Segmentation Results 3: *Dino_Segmentation_U2seg*



Original: img_2868.jpg

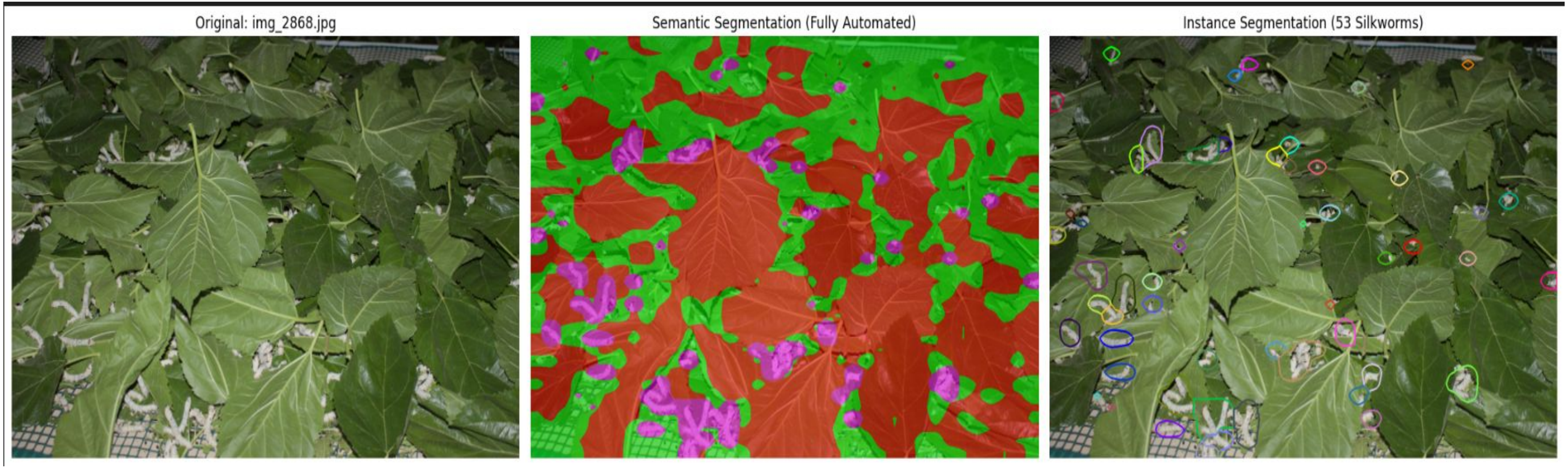
Raw Instances (150 found)

Semantic Segmentation

Final Merged Segmentation

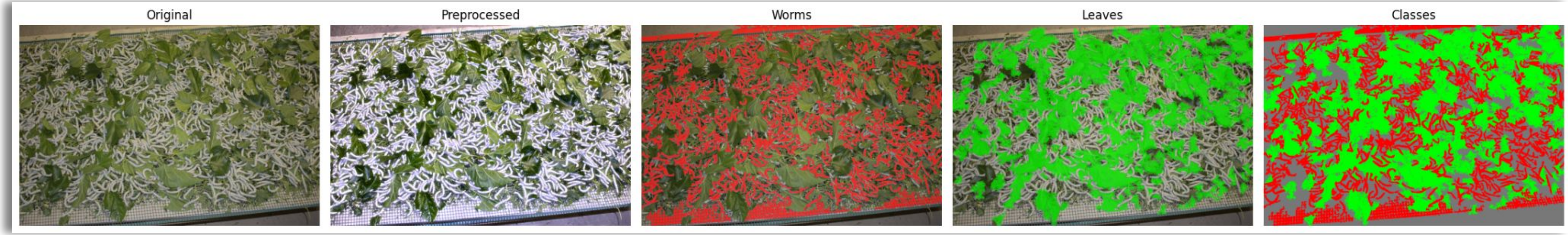
$\tau_{inst}=0.05$, $k_{inst}\approx150$ | $\tau_{sem}=0.12$, $k_{sem}=3$

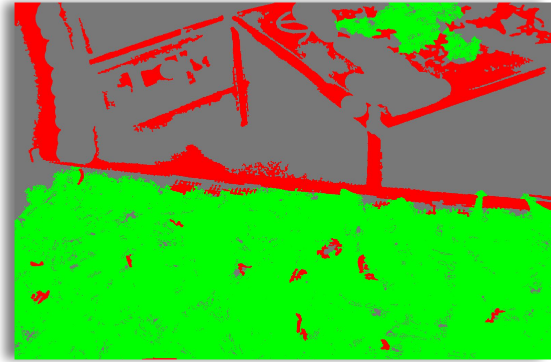
Segmentation Results 4: *Dino_Segmentation_Heuristic*

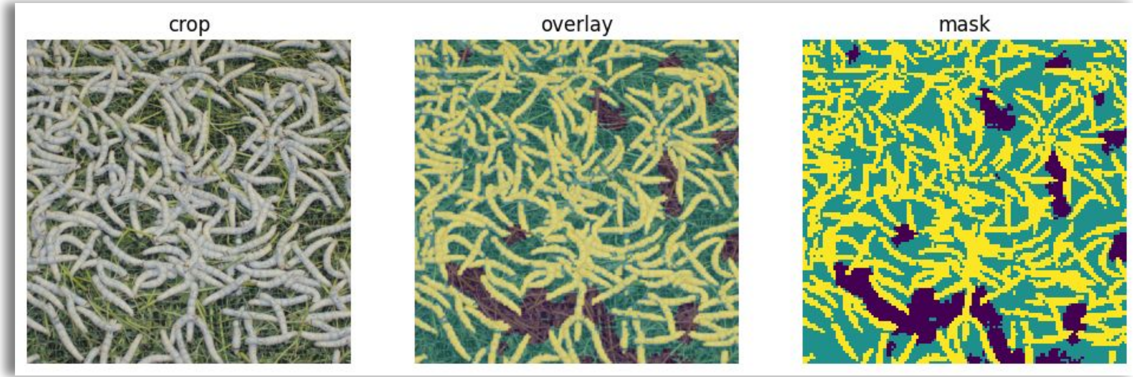

Original: img_2868.jpg
Semantic Segmentation (Fully Automated)
Instance Segmentation (53 Silkworms)

No hard-coded IDs | Same watershed split

Final segmentation results







Binary classification results

Model name	Accuracy	Precision	Recall	F1 Score	Specificity	False Positive Rate
EfficientNetV2	**0.9556**	0.9351	**0.9863**	**0.9600**	0.9194	0.0806
RepNeXt	**0.9556**	0.9653	0.9521	0.9586	0.9597	0.0403
MobileViT	0.9074	**0.9690**	0.8562	0.9091	**0.9677**	**0.0323**
Linear Regr.	0.9360	0.9280	0.9540	0.9410	0.9160	0.0840
L. Reg. + Conv	0.9480	0.9450	0.9580	0.9520	0.9370	0.0630

Binary classification results (STEGO masks):

Classification	Precision	Recall	F1-Score	Specificity	FPR
Class 0 (not_feed)	**0.9697**	0.8348	0.8972	**0.9806**	0.0194
Class 1 (feed)	0.8889	**0.9806**	**0.9325**	0.8348	**0.1652**

Two-stage Model
(instances first, then raw images)
Overall Accuracy: **0.9185**

Classification	Precision	Recall	F1-Score	Specificity	FPR
Class 0 (not_feed)	**0.7608**	0.6434	0.6972	**0.8452**	0.1548
Class 1 (feed)	0.7558	**0.8452**	**0.798**	0.6434	**0.3566**

Single-stage Model
(instances only)
Overall Accuracy: **0.7577**

Model Evaluation: Classification & Segmentation

Classification (Feeding Prediction)

- Evaluated models using accuracy on the binary feeding task.
- Tested **EfficientNetV2**, **RepNeXt**, and **MobileViT** with frozen backbones and trained classification heads.
- Observed consistent performance across architectures.
- **ROC analysis** enabled threshold tuning beyond naive 0.5 cutoff for the **regressor model**, improving binary decision robustness.
- *No signs of overfitting* due to the small model heads, augmentation, and inconsistent dataset.

Segmentation (Habitat Analysis)

- No ground truth masks available, so evaluation was qualitative only.
- Compared visual output of unsupervised methods (**CutLER**, **STEGO**, **U2Seg**, **DINO**) against expected class separation.
- Focused on **visual coherence, region consistency, and semantic separation** of silkworms, leaves, and background.
- **Heuristic DINO** pipeline gave the most stable masks; U2SegUnified added panoptic-style instances
- Heuristic segmentation provided strongest baseline: SegFormer distilled this knowledge with promising visual output.
- Overall subjective

Conclusions: Final considerations and future work

What we achieved:
We built a lightweight pipeline for silkworm monitoring that combines unsupervised segmentation (DINO heuristic + U2SegUnified; others) with efficient binary classifiers. The best instance-only models (**EfficientNetV2** / **RepNeXt**) reached ~0.96 F1, and a two-stage MobileNetV3 fine-tuned on raw images achieved 0.9185 accuracy, while simple models like a linear regressor over masks also achieved 0.9480 accuracy . SegFormer, fine-tuned on our self-supervised pseudo-masks, produced visually coherent results without human annotations.

What didn’t work / limits:
Methods like **CutLER**/**STEGO** struggled on our data, mainly due to *low resolution (224x224), noise, and heavy clutter*. HSV rules can still mislabel edge cases (e.g., pale leaves/refections vs worms), and lack of pixel ground truth forced a qualitative segmentation evaluation. Simple heuristics overall (qualitatively) performed better.

Future work:
Use higher-resolution inputs and stronger foundation segmenters.
Add a small labeled subset of ground truths to calibrate/validate masks and learn cluster-to-class mapping instead of HSV/color heuristics.

References

- **Papers:**
- Zhao et al. (2024) – RepNeXt: A Fast Multi-Scale CNN using Structural Reparameterization
- Tan & Le (2021) – EfficientNetV2: Smaller Models and Faster Training
- Mehta & Rastegari (2022) – MobileViT: Light-weight, General-purpose, and Mobile-friendly Vision Transformer
- Xie et al. (2023) – SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers
- Caron et al. (2021) – Emerging Properties in Self-Supervised Vision Transformers (DINO)
- Wang et al. (2022) – CutLER: Cut and Learn for Unsupervised Object Detection and Instance Segmentation
- Hamilton et al. (2022) – STEGO: Unsupervised Semantic Segmentation by Distilling Feature Correspondences
- Rossetti et al. (2023) – Removing Supervision in Semantic Segmentation with Local-Global Matching and Area Balancing
- Niu et al. (2023) – U2Seg: Unsupervised Universal Image Segmentation
- **Slides:**
- https://github.com/pietro-nardelli/sapienza-ppt-template
  [Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)]

Thank you for the attention!