ETH zürich

Improving Volume Prediction of Wheat based on multi-view Images

Yann Bourdé
Electrical Engineering and Information Technology – Signal Processing and Machine Learning

18.11.2025

Presentation Outline

1. Problem and Motivation
2. Background
3. Detection Experiments
4. Segmentation Experiments
5. Discussion
6. Conclusion
7. Future work
8. Q&A

Problem and Motivation

- Predict volume in mm^3 of wheat spikes/heads. Use cases:
  - Crop resilience
  - Fruiting efficiency/yield
- Old methods are slow and labor intensive.
- Design a system/pipeline which can automatically predict volume based on images.

- FIP data:
  - 13 images top down with different angles.
  - Scaned labeled spikes for accurate volume ground truth.

Figure 1: FIP camera (left), field image (right).

Background – Previous work

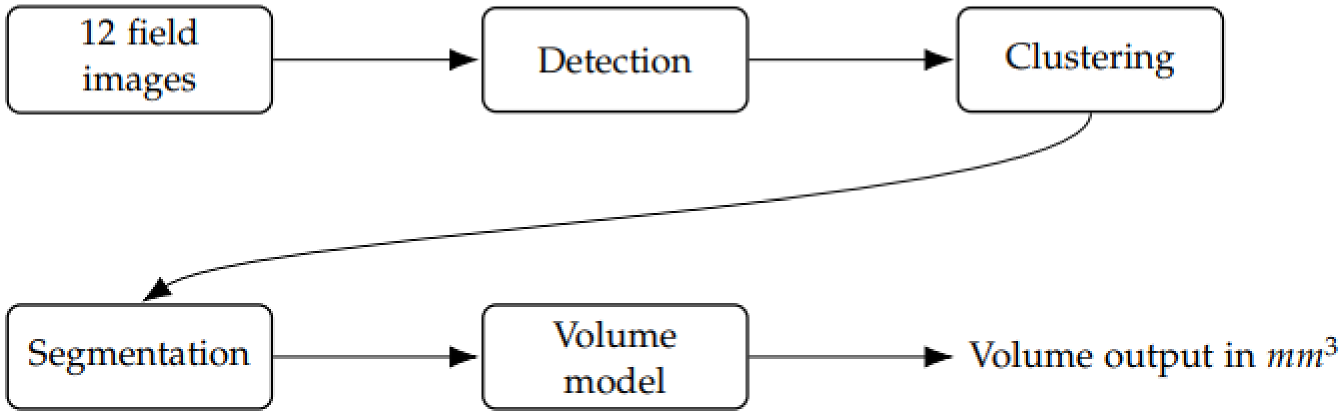

Figure 2: Process developed in the thesis «Multi-View Deep Learning for 3D Wheat Spike Volume Estimation in the Field" [1]".

- Four main parts:
  - Detection
  - Clustering
  - Segmentation
  - Volume prediction

- Shortcomings:
  - Detection speed and accuracy.
  - Clustering view retention.
  - Segmentation speed and accuracy.

Background – Datasets

- FIP:
  - 8190 images total
  - 1100 labeled spikes
  - 13 images with different angles for each plot and date
  - ~4000x3000 image size
- GWHD:
  - 6512 images
  - 1024x1024 image size
  - Over 300k bounding boxes total
- FIP2 manually labeled:
  - 7 plots with 36 images and one labeled image each
  - Same properties as FIP

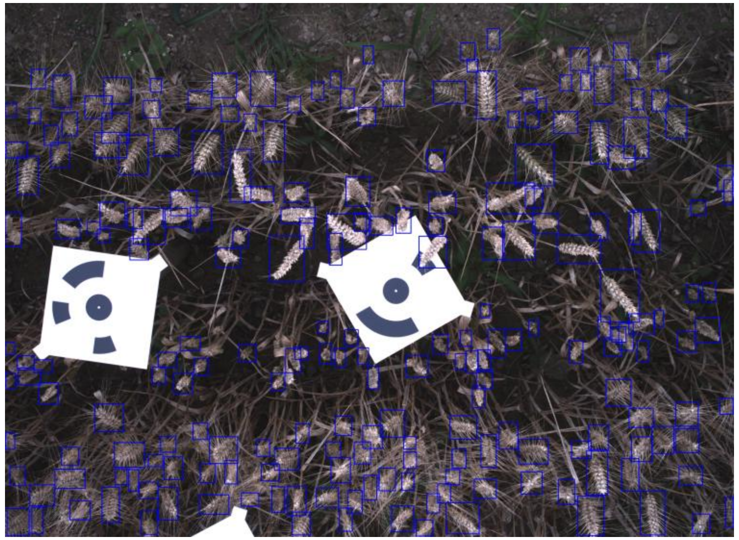

Figure 3: FIP image (top left), GHWD image (top right), FIP2 image (bottom left).

Detection Experiments - Accuracy

Models	Macro				Micro					
	Precision	Recall	F1	AP50	Precision	Recall	F1	TP	FP	FN
yolo5_all_20ep	0.73	0.91	0.81	0.68	0.73	0.91	0.81	1262	458	124
yolo5_all_40ep	0.73	0.90	0.81	0.68	0.73	0.90	0.81	1251	455	135
yolo5_train_20ep	0.67	0.89	0.77	0.62	0.68	0.89	0.77	1235	587	151
yolo11_all_20ep	0.76	0.93	0.84	0.72	0.77	0.93	0.84	1285	393	101
yolo11_all_40ep	**0.77**	**0.94**	**0.84**	**0.74**	**0.77**	**0.94**	**0.84**	**1300**	**393**	**86**
yolo11_train_val_20ep	0.68	0.85	0.76	0.59	0.69	0.85	0.76	1181	544	205
yolo12_all_20ep	0.75	0.91	0.82	0.70	0.75	0.91	0.82	1258	419	128
yolo12_all_40ep	0.75	0.93	0.83	0.71	0.76	0.92	0.83	1280	409	106
yolo12_train_20ep	0.67	0.88	0.76	0.60	0.68	0.88	0.76	1218	583	168

Table 1: Detection performance of different detection models variants

- Run detection on FIP2 with different models and record metrics.
- Observations:
  - Strong performance across all models.
  - Models trained on all data perform better → not enough data to max performance.
  - Newer/larger model not necessarily better.

Detection Experiments - Speed

Models	plot_461	plot_462	plot_463	plot_464	plot_465	plot_466	plot_467	Avg (ms)
yolo5_all_20ep	1127	1121	1131	1143	1123	1133	1106	**1126**
yolo5_all_40ep	1124	1250	1180	1143	1076	1080	1106	**1137**
yolo5_train_20ep	1119	1120	1123	1141	1116	1134	1098	**1122**
yolo11_all_20ep	1348	1270	1348	1264	1263	1277	1260	**1290**
yolo11_all_40ep	1352	1357	1370	1263	1253	1272	1263	**1304**
yolo11_train_val_20ep	1976	1169	1169	1158	1167	1168	1936	**1392**
yolo12_all_20ep	3261	3315	3328	3307	3313	3321	3316	**3309**
yolo12_all_40ep	3313	3315	3316	3316	3321	3327	3316	**3318**
yolo12_train_20ep	3302	3285	3286	3302	3306	3302	3300	**3298**

Table 2: Average inference time of different detection model on the FIP2 dataset.

- Record speed metric when running inference.
- Observations:
  - YOLOv5 and YOLOv11 are quite fast with a slight advantage to YOLOv5.
  - YOLOv12 is very slow and does offer better accuracy.

Detection Experiments – Clusters and Volume

Models	gt_views	cluster_views	missing_in_gt	present_in_gt	Avg. vol. diff. (mm^3)
baseline – yolo11_train_val_20ep	1969	1994	79	1915	1340
yolo11_all_40ep	1969	2043	109	1934	1335
yolo5_all_40ep	1969	2020	90	1930	1317

Table 3: Clustering[1] performance with different detection models.

- Idea: see if different detection model increase number of missing views and if extra missing views improve volume prediction.
- Observations:
  - Increase in missing views using better models.
  - No increase in volume prediction.

1. Clustering algorithm based on the works of Takuma Doi et al. [3].

Segmentation Experiments – Exploring

- Segmentation is closer to final output
  → might have larger impact on final volume.
- No large labeled segmentation dataset for this task.
- Use SAM2.1 to create rough masks.

- Idea – re-train on cropped images:
  - Closer to inference in the pipeline.
  - Reduces patchy masks.

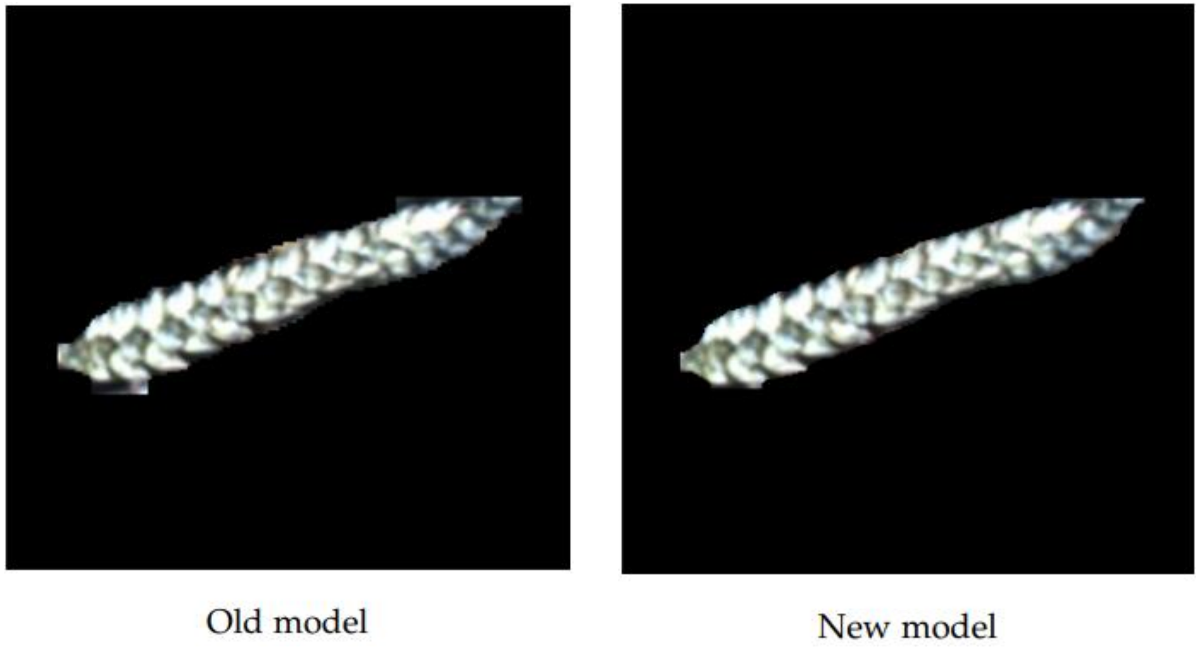

Figure 4: Visual comparison of old and new segmentation models.

Segmentation Experiments – Volume

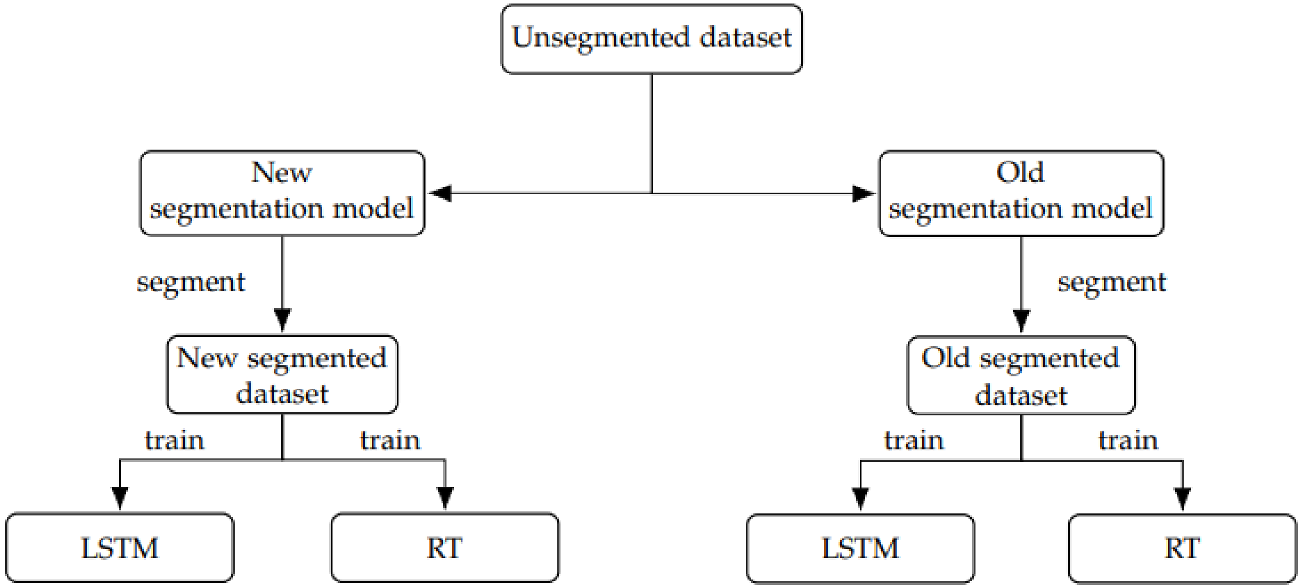

Figure 5: Overview of the volume experiment process for segmentation models.

- Can the segmentation model impact volume prediction?

- Experiment process:
  1. Train a new segmentation model on cropped data.
  2. Re-segment the unsegmented FIP dataset (has GT volumes).
  3. Train LSTMs and RTs on segmented datasets.
  4. Compare volume prediction outputs.

Segmentation Experiments – Volume – Results

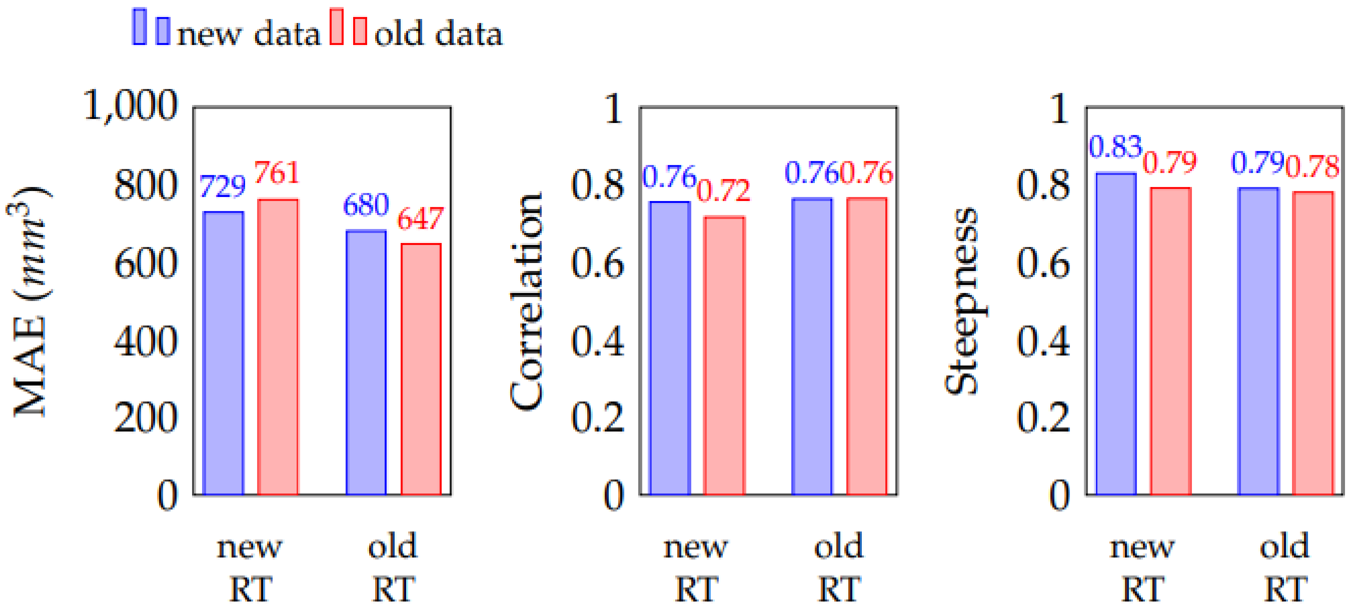

Figure 5: Regulated Transformer models performance using own mapping with distance normalization.

- Observation across all experiments:
  - MAE is usually slightly lower with old model, regardless of data.
  - Correlation is slightly higher with old model, regardless of data.
  - Steepness is slightly higher for new model and even higher on new data.
  - Differences are usually small.
  - Tendency for the new model to struggle on old data but not the other way around.

Segmentation Experiments – Speed

Model	Size (MB)	Latency (ms)	FPS	mAP@50	mAP@50–95	Precision	Recall
XLarge	119	23	43	0.94	0.78	0.94	0.87
Large	53	15	66	0.94	0.77	0.91	0.88
Medium	43	12	85	0.93	0.76	0.94	0.85
Small	20	9	110	0.93	0.73	0.92	0.85
Nano	6	8	122	0.90	0.68	0.91	0.81

Table 4: Size, latency, and performance with different segmentation model sizes

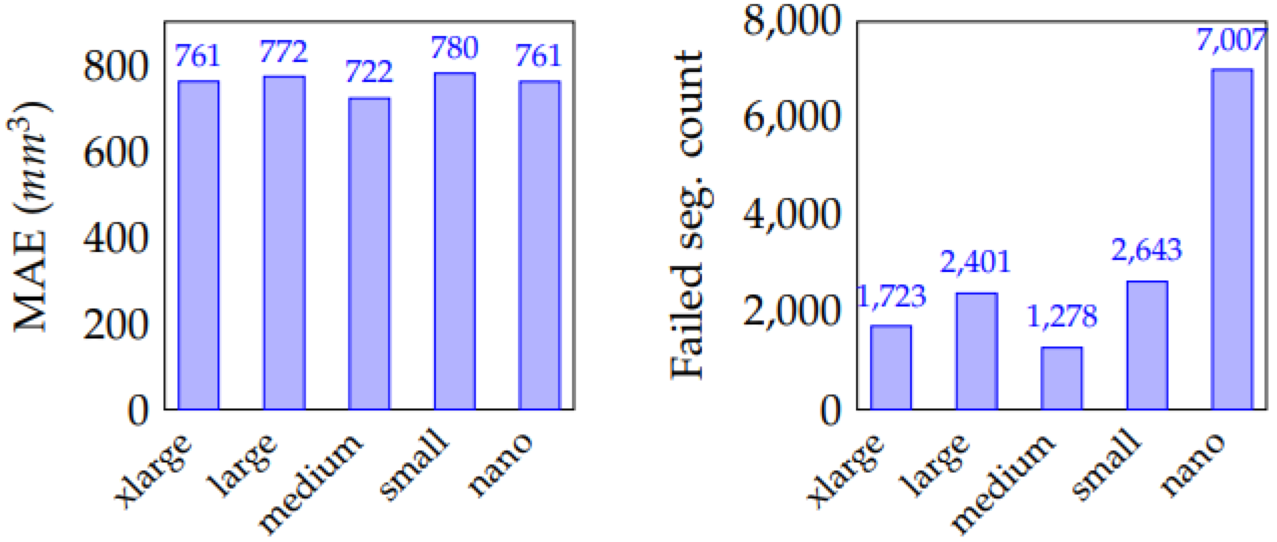

Figure 6: LSTM performance and failed segmentation count with different segmentation model sizes.

- Train models on the same data and evaluate performance on validation set.
- Observations:
  - Predictably, increased size → increases accuracy and reduces speed on validation set.
  - Accuracy drop is small across models.
- What about on final volume?
  - Medium is best performing.
  - Failed segmentations increase massively for smaller models and slightly for larger models.

Conclusion

Detection:

- Lack of data to max out performance of these models.
- Accuracy is good enough → speed can be prioritized.
- Detection has no significant impact on final output volume in the current pipeline (admit there are enough views)
  - Does the volume model really leverage “3D“ data?

Segmentation:

- Segmentation can impact final output volume and consistency of predictions.
- Training on cropped images may lead to overfitting.
- A minimum model size is required to avoid large numbers of segmentation failure.

Future Work

- Train segmentation models on mixed data cropped and uncropped → best of both worlds?

- Explore improvements for the third aspect of the pipeline which is the volume prediction model.
  - Is DINOv2 the best solution for this task?

- Explore the GWFSS dataset for self-supervised training of segmentation models.
  - Contains a lot of unlabeled data that could be leverageable.

ETH zürich

Thank you!

Q&A

Bibliography

- [1] Jannis Widmer. “Multi-View Deep Learning for 3D Wheat Spike Volume Estimation in the Field”. Master’s thesis. Zurich, Switzerland: ETH Zurich, Apr. 2025.
- [2] N. Kirchgessner et al. “The ETH field phenotyping platform FIP: a cable-suspended multi-sensor system”. In: Functional Plant Biology 44.1 (2016), pp. 154–168. doi: 10.1071/FP16165.
- [3] Takuma Doi et al. Descriptor-Free Multi-View Region Matching for InstanceWise 3D Reconstruction. 2020. arXiv: 2011.13649 [cs.CV]. url: https: //arxiv.org/abs/2011.13649.
- [4] Maxime Oquab et al. DINOv2: Learning Robust Visual Features without Supervision. 2024. arXiv: 2304.07193 [cs.CV]. url: https://arxiv. org/abs/2304.07193.

Additional

Models	# Param
yolo5_all_20ep	87,198,694
yolo5_all_40ep	87,198,694
yolo5_train_20ep	87,198,694
yolo11_all_20ep	25,311,251
yolo11_all_40ep	25,311,251
yolo11_train_val_20ep	20,053,779
yolo12_all_20ep	26,389,875
yolo12_all_40ep	26,389,875
yolo12_train_20ep	26,389,875

Table 5: Detection model parameter counts.

- YOLOv5 has a lot more parameter but its architecture is very different so it is faster.