A Supervised Learning Solution for Autonomous Row Following Tasks in Horticulture

Tuan Le, Vignesh Raja Ponnambalam, Jon Glenn Omholt Gjevestad and Pål Johan From

***Abstract*— Precision agriculture is the key to sustainable farming. The usage of autonomous robotics systems in agriculture is rising. Similar to other mature areas of applied robots, agricultural robots must be able to robustly navigate in their working places (polytunnel, crop fields, etc.,). In horticulture, row following is one of the key tasks that autonomous agricultural robots must perform. Several studies had been done to address this problem. However, existing methods are tailored to their specific environments. This work aims to provide a CNN approach to row following tasks that can be used for both indoor (polytunnel-liked) and outdoor (orchard-liked) environments.**

I. Introduction and Motivation

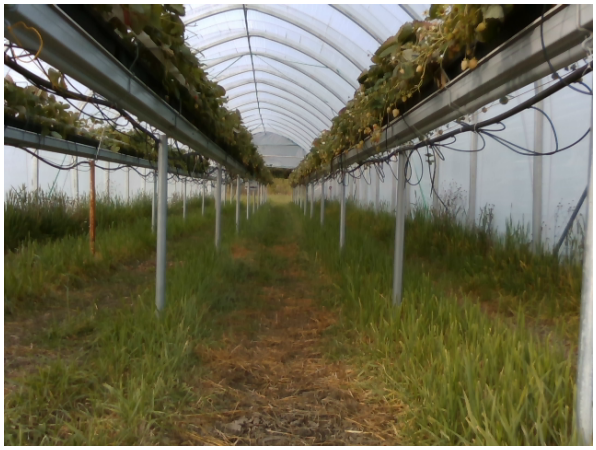
A common practice for growing vegetation in horticulture is to form row-like structures. For outdoor environment, orchards mostly use row-liked structures for growing. For fruits such as apples and oranges, the most common row structure is a tree wall, e.g a row is formed by placing trees on both sides of a path. However, for fruits such as grapes, pears and kiwi, a pergola structure is more common. In a pergola, rows are formed by trees and supporting poles. For indoor environment such as polytunnel, rows are formed either by lines of table-trays placing on poles or being hung from the roof. We show three examples of polytunnel, open orchard and pergola in Fig. 1, respectively.

For open fields like orchards, classical navigation methods relying on external position sensors such as GNSS were fully developed [2]. For greenhouses or polytunnels, existing navigation methods from the robotics community using a 2D laser scanner can be directly applied [3]. Obviously, these classical methods may suffer in some specific conditions: blockage of GNSS signals (in pergolas where dense canopies usually exist), uneven ground floor distorts 2D laser scanner, or in case of missing trees in a row (Fig. 1c) might also confuse the laser scanner reading. Several works have been done to address these problems, which specifically avoid using any external position sensor or assuming a flat terrain. Zhang *et al.* in [4] used a rotating 2D laser scanner for augmenting 3D scans to detect tree trunk and traverse along tree rows in an orchard. Bell *et al.* in [5] propose a navigation approach using a 3D LiDAR sensor to navigate inside a kiwifruit pergola, where GNSS signals are blocked by dense canopies.

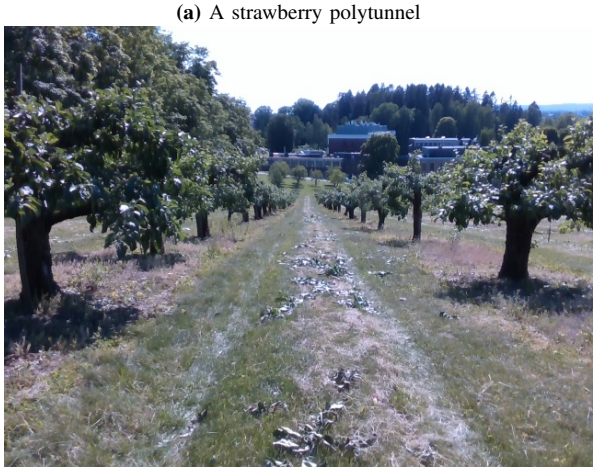
We are motivated by the structural variations that we have in our test fields at NMBU. We have a strawberry polytunnel, in which three rows of tabletop trays are placed on poles.

All authors are with Faculty of Science and Technology, Norwegian University of Life Sciences

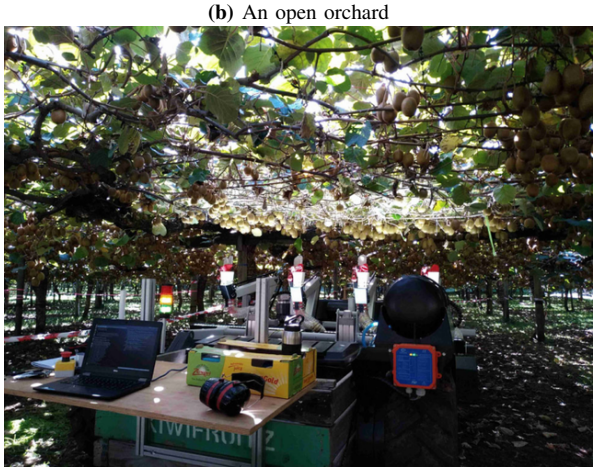
*Corresponding author: tuan.dung.le@nmbu.no



(a) A strawberry polytunnel



(b) An open orchard



(c) A kiwifruit orchard with pergola structure. Image courtesy of [1]

Fig. 1: Different types of horticultural environments.

(a) Trees with supporting poles **(b)** Plant bushes with supporting poles **(c)** A row with missing trees

Fig. 2: Different types of orchard environments at NMBU.

The row width is 1.5 meters (Fig. 1a). On the other hand, we have an open orchard where three different types of structure are utilized: a) standard rows, where trees are roughly spaced 2 meters apart, as shown in Fig. 1b, b) trees with supporting poles, which are roughly 2 meters apart, as shown in Fig. 2a, c) small trees with large supporting poles, where poles are roughly 2.3 meters apart, as shown in Fig. 2b. On some rows, one tree or several trees might be missed as shown in Fig 2c. The row width in our orchard is much wider than the one in our polytunnel. Moreover, different types of row following tasks may be performed on these environments. For example, UV light treatment in polytunnel or tree watering on orchards are classified as *centerline following* tasks, meaning a robot needs to maintain equidistant to both sides. An example of centerline following in UV light treatment in a polytunnel is shown in Fig. 3.

Fig. 3: A design of Thorvald robot for UV line treatment inside a polytunnel. The robot is required to perform centerline following.

For orchard with a wide row in harvesting season, a robot may need to stay close to one side of a row while moving along that row for fruit harvesting. This is classified as *sideline following* task.

We are inspired by the work of Bell *et al.* in [6], in which the authors trained a fully convolutional neural network (FCN) for segmenting drivable areas for row following in a kiwifruit pergola. Drivable area means the area a robot can translate to from its current position without collision. We believe this approach is more generic and applicable than existing methods relying on external position sensor (high cost for RTK-GNSS devices), artificial landmarks (burden on infrastructure for placing and maintaining) or laser scanner sensor (being confused in the presence of missing/additional objects). More over, it uses a low-cost camera sensor, which keeps the whole robotic system cost-efficient.

We argue that our work is different from the one in [6] by a magnitude of generalization. The authors in [6] were only concerned about centerline following for harvesting tasks in a specific kiwifruit pergola. We train our network for segmenting traversable ground on an inclusive dataset containing both indoor (a strawberry polytunnel) and outdoor (orchards with three different types of row structure) environment. We also cross-validate our network performance on different network architectures, including ResNet [7], Darknet [8], MobileNet [9] and ERFNet [10]. Hence, we can evaluate how our network performs in different types of environments with different network architectures. In addition, for outdoor environment, we also have three different types of structure. Hence, our network is suitable for many types of environment, which makes it more generic.

II. Description of Dataset and Training Process

For data collection, we use a popular Intel Realsense Camera D435i. We mount the Realsense camera on our ground robot [11] as shown in Fig. 4. We manually joystick the robot along rows in our strawberry polytunnel and our open orchard. We made sure to capture as many different scenarios as possible: a) our strawberry polytunnel recordings contain our robot moving along rows with in-row rotations that are not considered dangerous b) for our open orchard, our robot undergoes different moving directions while traversing rows - straight line, rotating, diagonally c) data is being recorded under various light conditions.

We select 500 images of size 640x480 pixels for training and 57 held out images of the same size for testing. For labeling images, we manually label each pixel either traversable or non-traversable. The human expert who controlled the robot during data collection decides which pixel areas can be

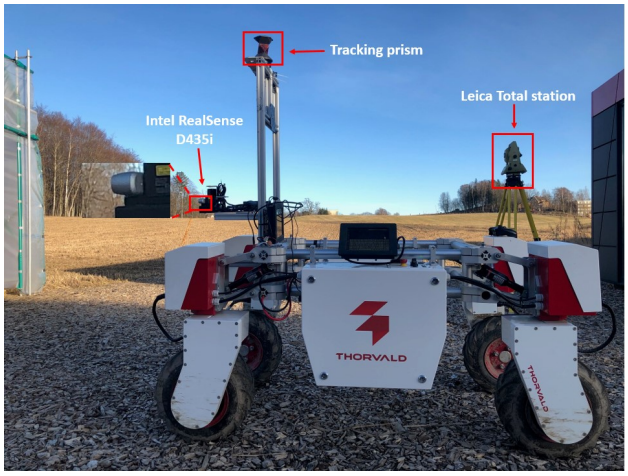


Fig. 4: Robot setup for data collection.

considered traversable. The human expert follows a similar definition of "traversable" as in [6], in which traversable area is defined as a space that the robot might get to from its current position by following a straight line and without collisions. This definition means that in cases, where the robot can observe several rows from its current position, the network should not classify neighbor row areas as traversable. We train our network using the training tool in [12] with a Zotac Mini Gaming PC equipped with an Nvidia Geforce GTX 1070K, 16GB memory, and a quad-core Intel i5-7500T CPU. A sample architecture based on ERFNet, which we use, is shown in Fig. 5.

We also show examples of annotated data that we use for training in Fig. 6.

III. Experimental Results and Discussions

A. Results

We report our training results, including types of network architecture, the average accuracy (mAcc), mean Jaccard index (mIoU), and the mean Jaccard index of "traversable" class (mIoU of class 1) in Table I. Note that for each network, we average the results of the best three trained models and report those values. As illustrated in Fig. 7, our trained network is able to segment traversable areas, which is the part of a row our robot is currently in and can safely translates to without collisions. Some test results including corner cases are presented in Fig. 7, where the network correctly ignores "traversable" areas of neighbor rows. Note that in case of a row with missing trees as in Fig. 7d, we explicitly do not want our robot to make a cross movement to a neighbor row, even it is safe to do so in this case. Obviously, for indoor environment, we do not want our robot to make any cross movement from one row to another. Our trained network was able to correctly identify the traversable areas inside our polytunnel (Fig. 7j-k).

We also report the average inference time (Infer. time) per image in milliseconds by each architecture when interfacing in ROS in Table I. From experiments, we see that ERFNet gives us the fastest inference time at roughly 48 ms, which is approximately 20Hz. The slowest FPS is reported at approximately 5Hz using Darknet 53. Since our robot moves at a relatively low speed of 0.7 m/s, this inference rate is sufficient for row following performances. We do not observe significant differences in segmentation accuracy between different network architectures. Hence, it is up to an end-user to select a specific network architecture.

Architecture	mAcc	mIoU	mIoU of class 1	Infer. time
ResNet 18	0.986	0.941	0.899	$\sim$ 54ms
ResNet 50	0.987	0.946	0.906	$\sim$ 142ms
ResNet 152	0.985	0.939	0.895	$\sim$ 190ms
Darknet 21	0.985	0.938	0.892	$\sim$ 118ms
Darknet 53	0.987	0.947	0.908	$\sim$ 206ms
ERFNet	0.986	0.941	0.898	$\sim$ 48ms
MobileNet V2	0.984	0.935	0.888	$\sim$ 55ms

TABLE I: Training results

B. Discussions

Currently, we have two main drawbacks in our work:

- We only consider traversable areas for in-row movements. We observe that headland areas are much different from in-row areas. Incorporating headland into our current network actually worsens its performance. Hence, we leave between-rows transition as a separate problem to solve.
- Ground truth determination is our bottleneck. Relying on a human expert for ground truth labeling is time-consuming and error-prone. However, to our knowledge, there are not any publicly available data sets that we can use for training or compare with. We envision a good ground truth that must come from professional terrain surveying services, for which we plan to do in the future. Nonetheless, we want to stress at the current state, our network can accurately segment traversable areas on par with a human expert.

IV. Conclusions

In this work, we propose a supervised learning solution for row following tasks in horticulture. Using a low cost camera, our solution is suitable for a wide range of agricultural robots. We present our approach to collect and train a convolutional neural network for segmenting traversable areas, which can be subsequently used for motion planning. We show that our trained networks (based on different network architectures) are well generalized to different environments than existing methods. We also show that the inference time of our network is sufficiently fast for motion planning tasks. For future work, we plan to achieve a professional ground truth data for labeling traversable area using terrain surveying services and release our data set to our agricultural robotics community.

V. Acknowledgement

We would like to thank Antonio Candea Leite for helping with data collection on the orchard data set.

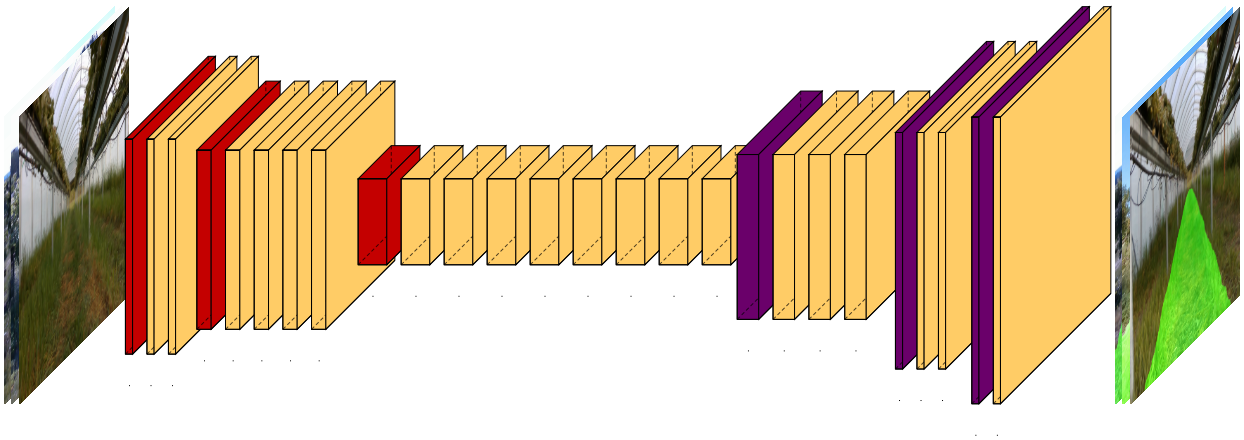
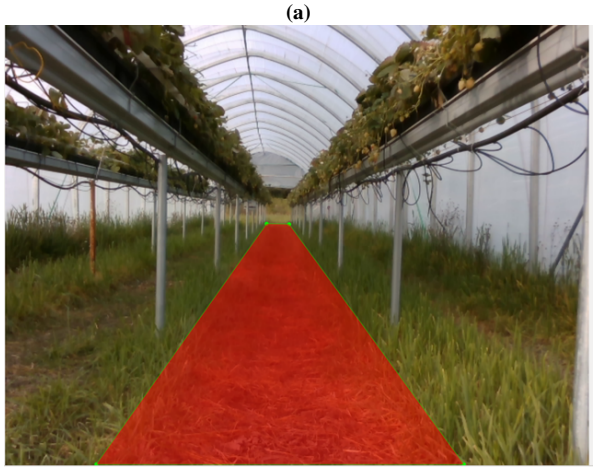


Fig. 5: An illustration of a network architect that we use. This is similar to the structure of ERFNet in [10]. Red layers - downsample module, Yellow layers - variable receptive field, Purple layers - upsample module.

(a)



(b)

Fig. 6: Screenshots of annotated images for training, where red areas depict traversable areas.

REFERENCES

[1] H. Williams, C. Ting, M. Nejati, M. H. Jones, N. Penhall, J. Lim, M. Seabright, J. Bell, H. S. Ahn, A. Scarfe *et al.*, "Improvements to and large-scale evaluation of a robotic kiwifruit harvester," *Journal of Field Robotics*, vol. 37, no. 2, pp. 187–201, 2020.

[2] P. Biber, U. Weiss, M. Dorna, and A. Albert, "Navigation system of the autonomous agricultural robot bonirob," in *Workshop on Agricultural Robotics: Enabling Safe, Efficient, and Affordable Robots for Food Production (Collocated with IROS 2012), Vilamoura, Portugal*, 2012.

[3] L. Grimstad, R. Zakaria, T. D. Le, and P. J. From, "A novel autonomous robot for greenhouse applications," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1–9.

[4] J. Zhang, A. Chambers, S. Maeta, M. Bergerman, and S. Singh, "3d perception for accurate row following: Methodology and results," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 5306–5313.

[5] J. Bell, B. A. MacDonald, and H. S. Ahn, "Row following in pergola structured orchards," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 640–645.

[6] ——, "Row following in pergola structured orchards by a monocular camera using a fully convolutional neural network," in *Australasian conference on robotics and automation (ACRA)*, 2017, pp. 133–140.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[8] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[9] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[10] E. Romera, J. M. Alvarez, L. M. Bergasa, and R. Arroyo, "Erfnet: Efficient residual factorized convnet for real-time semantic segmentation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 1, pp. 263–272, 2017.

[11] L. Grimstad and P. J. From, "The thorvald ii agricultural robotic system," *Robotics*, vol. 6, no. 4, p. 24, 2017.

[12] A. Milioto, L. Mandtler, and C. Stachniss, "Fast Instance and Semantic Segmentation Exploiting Local Connectivity, Metric Learning, and One-Shot Detection for Robotics ," in *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2019.

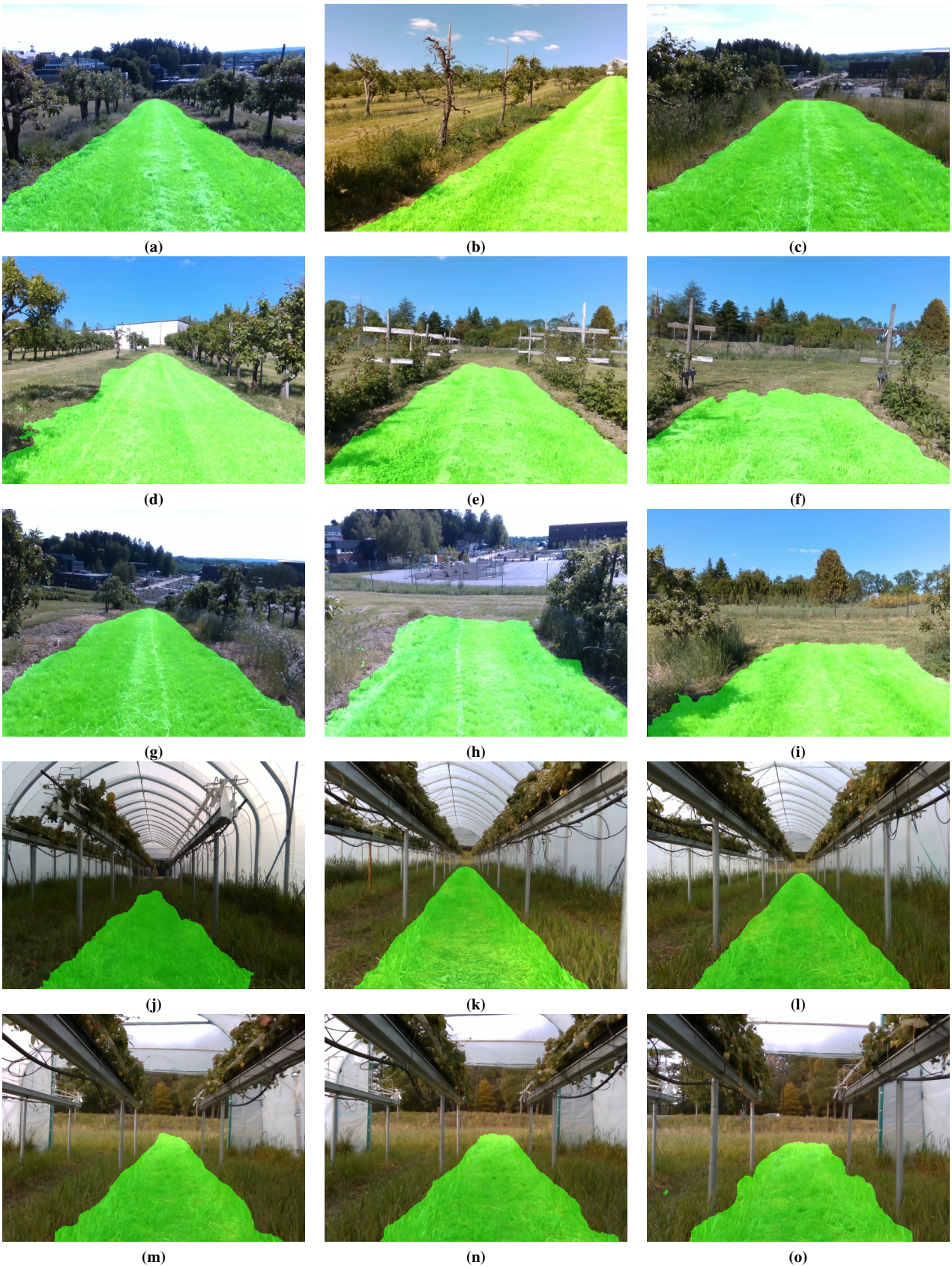


Fig. 7: Segmentation test results. Best viewed in color.