Row Segmentation and Local Path Planning in Fruit Plantations using Lightweight Neural Networks

Pedro Felipe Jaquetti, Heitor Silverio Lopes
Computational Intelligence Laboratory – LABIC/CPGEI
Federal University of Technology – Paraná (UTFPR)
Emails:pedrojaquetti1@outlook.com
hslopes@utfpr.edu.br

***Abstract*—Drive assist in agriculture has emerged as a promising solution for increasing resource efficiency and reducing operating costs. This paper details the development of essential components for an agricultural drive-assist system: a lightweight semantic segmentation model for crop row perception and a local path planning algorithm that extracts a navigation trajectory from the model's output. The segmentation model employs a U-Net-style decoder with a MobileNetV3 encoder, optimized for computational efficiency. The system was rigorously evaluated on a custom dataset from orange groves and vineyards using 5-fold cross-validation. To confirm its suitability for real-world deployment, the segmentation model's performance was benchmarked on an RK3588 System-on-Chip (SoC). Our approach achieved a mean Intersection over Union (IoU) of 94.76% (±0.24%), outperforming a DeepLabV3+ baseline in both accuracy and stability. On the RK3588, the model demonstrated an inference speed of 46.4 FPS. These results validate that the proposed components provide a robust, stable, and efficient foundation for vision-based drive-assisted on resource-constrained agricultural hardware.**

***Index Terms*—Computer vision; Deep learning; Drive assisted; Vision-based autopilot**

I. INTRODUCTION

The Food and Agriculture Organization of the United Nations (FAO) projects an urgent challenge: by 2050, food production must increase by 70% to meet the needs of an estimated population of 9.1 billion [1]. This outlook not only encourages modern agriculture to seek innovative and sustainable solutions to optimize production.

In this context, artificial intelligence (AI) emerges as a useful tool, helping to address particular challenges and enhance the efficiency of agricultural resources [2]. Agricultural machines, often operating for long hours in vast areas, can benefit significantly from autonomous navigation, ensuring precision, operational safety, and cost reduction [3].

Drive-assist systems in agriculture offer significant advantages by minimizing crop damage, enhancing operational efficiency, and enabling operators to cover larger areas daily. These capabilities directly address key challenges in modern agriculture. However, the premier method for this navigation, Real-Time Kinematic GPS (RTK-GPS), is hindered by several limitations. These include susceptibility to signal interference, coverage gaps, and, most significantly, a prohibitive cost that limits its adoption in developing nations or on smaller farms. Furthermore, trajectories based solely on RTK-GPS data often fail to adequately account for the dynamic and unpredictable nature of the agricultural environment [4].

Fig. 1: Sample of typical image collected.

Given these challenges, the use of cameras mounted on agricultural machinery emerges as a promising alternative. With a lower cost compared to RTK-GPS, these cameras capture information in real time and, when combined with adequate processing, demonstrate greater adaptability to the dynamics of the environment. Furthermore, unlike RTK-GPS, cameras are not affected by signal fluctuations or lack of coverage. This approach represents a more flexible and robust solution for autonomous navigation in constantly changing agricultural environments. Overall, such a technological innovation has the potential to increase competitiveness in agriculture.

In navigation in fruit-growing areas, the study conducted by [5] uses the conventional Otsu segmentation method [6] to distinguish between sky and ground in the image. From the generated segmentation mask, the white pixels in each column are counted. The column with the highest number of white pixels is then identified as the guide that the robot should follow.

A more effective and robust approach is presented by [7], where RTK-GPS is used to create an automatic annotation system for the optimal path in vineyard rows. The model receives depth and RGB images from an RGB-D camera as input and, as output, returns a heat map of the possible path that the robot should follow. After processing, this heat map is transformed into a guide that the robot uses as a reference.

This paper presents the development of essential components for agricultural drive-assist systems: a lightweight segmentation model and a local path planning algorithm. These components are designed to work in tandem to guide an agricultural vehicle and maintain its trajectory within crop rows, specifically in environments like orange groves and vineyards. To validate its efficiency for on-vehicle use, the segmentation model's performance is rigorously evaluated on an RK3588 System-on-Chip (SoC), a hardware platform with dedicated acceleration for deep learning workloads.

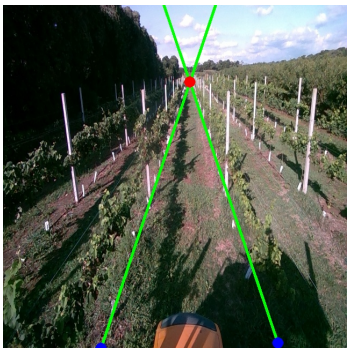
II. Methodology

This section describes in detail the steps involved in developing the proposed solution for segmenting rows in fruit plantations. It is important to mention that the central problem approached in this work is semantic segmentation of agricultural images. It is a computer vision technique different from object detection since it goes a step further, by outlining the exact boundaries of each object and classifying every single pixel that belongs to it. First, we present the process of collecting and annotating the data used to train and evaluate the model. Next, we discuss the neural network architecture adopted, based on a U-Net with MobileNetV3 encoder, detailing its motivations and internal structure. Finally, we address the model training procedures, including the hyperparameters used, the loss functions chosen, and the strategies applied to optimize performance during learning. The implementation of the proposed U-Net model with a MobileNetV3 encoder, along with the local path planning algorithm, is publicly available in our open-source repository [8].

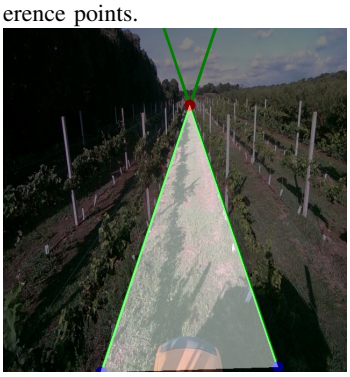
A. Dataset

Data collection for this study was carried out using a camera mounted on the roof of two different tractors, enabling images to be captured in vineyards and orange groves. For both cases, images were collected on the same rural property, always seeking to record images on land with different characteristics, such as variations in relief and plant layout. In addition, diversity in lighting conditions was taken into account, including images collected at dusk and on cloudy days, in order to increase the robustness of the dataset due to environmental variations. It is important to emphasize that the dataset was entirely created for the purposes of this study, tailored to represent realistic navigation scenarios in fruit-growing areas. An example of one of these collected images is shown in Figure 1.

In the image annotation procedure, a technique was adopted to streamline the process, consisting of creating two lines that outline the possible free path for the machine. Next, three reference points were identified, being the intersection point between the two lines and the point closest to the base of the image of each of the lines. Figure 2a illustrates this procedure, where the intersection point is represented by the red dot, and the points on the lines closest to the base are shown in blue. In the next step, the **fillPoly** function of OpenCV was used to create a triangle-shaped segmentation mask, as shown in Figure 2b.



(a) Image annotated with the 3 reference points.



(b) Mask generated from the 3 reference points.

Fig. 2: Transformation of annotated lines to segmentation mask.

This method for annotating and processing images was essential to streamline the work and ensure accuracy in identifying areas of interest. A total of 1,246 images were annotated, with 692 corresponding to orange groves and 554 to vineyards. This dataset forms the basis for our model training and evaluation, which employs the cross-validation methodology detailed in Section II-E.

B. Data Augmentation

Data augmentation consists of applying random transformations to training images in order to increase the diversity of the dataset and make the model more robust to variations found in real-world scenarios. In this study, this technique was applied dynamically during training—that is, the transformations were performed on the fly, at each new iteration, avoiding the need to expand the dataset beforehand. The transformations used were implemented using Python's Albumentations library[1]. Among the transformations applied, the following stand out:

- **Horizontal Flip** (`p=0.5`): horizontally flips the image with a 50% probability, simulating different orientations.
- **ShiftScaleRotate** (`scale_limit=0.5, rotate_limit=0, shift_limit=0.1, p=1`):

[1]https://pypi.org/project/albumentations/

applies random shift and scale without rotation, with a limit of 50% for scale and 10% for shift, preserving the content.

- **PadIfNeeded** and **RandomCrop** (`fixsize=640`): ensure that the image has a minimum size, followed by random cropping to maintain the input resolution.
- **GaussNoise** (`p=0.2`): adds Gaussian noise to simulate sensor variations and imperfections.
- **Perspective** (`p=0.5`): applies perspective transformations to simulate different viewing angles.
- **OneOf (CLAHE, RandomBrightnessContrast, RandomGamma)** (`p=0.5`): applies one of these transformations to improve contrast and brightness, simulating different lighting conditions.
- **OneOf (Sharpen, Blur, MotionBlur)** (`p=0.4`): applies a sharpening or blur filter to simulate variable camera focus.
- **OneOf (RandomBrightnessContrast, HueSaturationValue)** (`p=0.3`): alters brightness, contrast, and hue to increase color variety.

This set of techniques allows the model to learn how to segment in varied scenarios, increasing its robustness in real images of the plantation.

C. Model

For the task of identifying streets in fruit-growing areas, we chose to explore approaches consolidated in the literature previously used on urban road segmentation. Studies such as those by [9] and [10] demonstrate that U-Net-based architectures perform robustly even under adverse lighting and weather conditions. Motivated by these results, the model adopted in this work consists of a convolutional neural network for semantic segmentation, based on the U-Net architecture, using the pre-trained MobileNetV3-Large as an encoder to ensure efficient extraction of visually relevant features with low computational demand. Figure 3 shows a schematic diagram of the architecture used.

The encoder extracts hierarchical representations at multiple levels, whose intermediate outputs are used in skip connections, facilitating the recovery of spatial details lost during the downsampling process.

The decoder reconstructs the segmentation map through bilinear *upsampling* blocks, followed by separable convolutions in depth. These convolutions consist of a *depthwise* step, which applies filters separately per channel, followed by a *pointwise* (1×1) convolution, which combines the extracted information. This structure significantly reduces the computational cost compared to traditional convolutions while maintaining good representation capacity.

The final output of the model is obtained by a 1×1 convolution, which generates the segmentation probability map for a single class.

During training and inference, the images were resized to 320×320 pixels. This resolution was chosen to strike a balance between achieving a fast inference time and preserving essential visual details. This ensures that the identification of the guide line—responsible for keeping the machine centered on the plantation road—loses as little quality as possible, even after resizing.

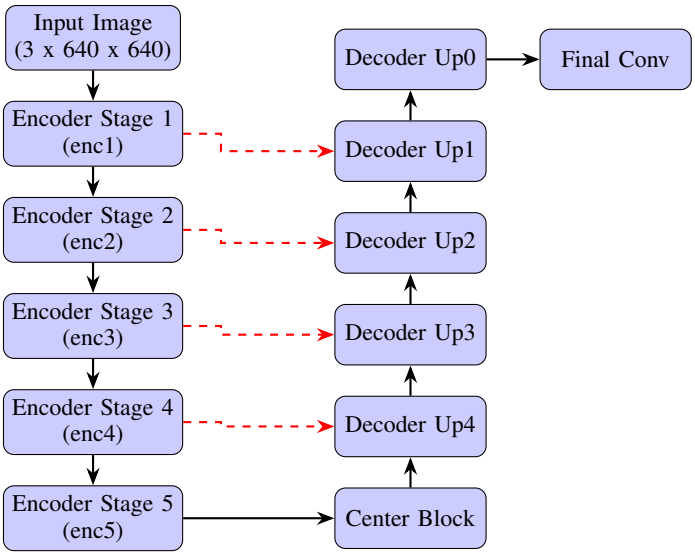


Fig. 3: Architecture of the U-Net model with MobileNetV3 encoder and decoder with skip connections.

This architecture is particularly well suited for segmenting rows in fruit plantations, as it combines computational efficiency with accuracy in preserving relevant spatial details.

D. Training

The models were trained using the hyperparameters summarized in Table I. These parameters were optimized through empirical experiments to maximize the mean Intersection over Union (IoU). All input images were resized to a resolution of 320×320 pixels. This smaller resolution was chosen as a strategic trade-off to significantly reduce the computational load for faster processing, while still retaining sufficient visual detail for the segmentation task.

1) Loss Function: The total loss function combines Dice Loss and Focal Loss to leverage the strengths of both. This hybrid approach balances the model's ability to handle class imbalance while maximizing the overlap between the predicted and ground-truth masks. The total loss is defined as:

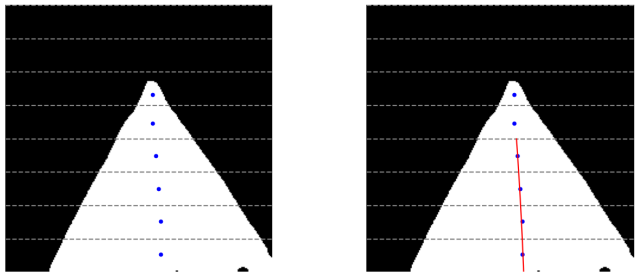
$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{Dice}} + 0.25 \cdot \mathcal{L}_{\text{Focal}} \tag{1}$$

where $\mathcal{L}_{\text{Dice}}$ measures region similarity and $\mathcal{L}_{\text{Focal}}$ focuses training on difficult examples.

2) Optimization and Scheduling: We used the Adam optimizer with a weight decay of 1e-4 for regularization. A two-part learning rate schedule was employed, starting with a 20-epoch linear warmup for stable initial convergence. Following the warmup, a 'ReduceLROnPlateau' scheduler managed the learning rate, reducing it when the validation loss plateaued. Training was configured to stop if the learning rate fell below 1×10^{-5} or if the validation loss did not improve for 10 consecutive epochs ('EarlyStopping').

TABLE I: Training hyperparameters.

Parameter	Value
Input Size	320 × 320 × 3
Optimizer	Adam
Initial Learning Rate	1e-3
Weight Decay	1e-4
Batch Size	16
Epochs	200
Warm Scheduler	Linear Warmup (20 epochs)
Main Scheduler	ReduceLROnPlateau (patience=10, factor=0.1)
Early Stopping	Patience=10, min_delta=0



(a) Slicing and centroid calculation

(b) Application of polynomial regression on centroids

Fig. 4: Segmentation mask processing for ideal case

E. Model Evaluation Method

The primary performance metric used for evaluation is the IoU (*Intersection over Union*), widely used in semantic segmentation tasks [11]. This metric quantifies the overlap between the segmentation predicted by the model and the ground truth segmentation, and is defined by the following equation:

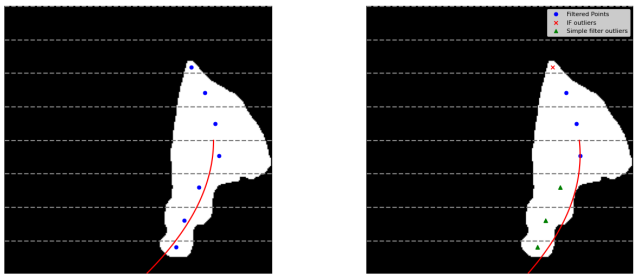
$$IoU = \frac{\text{Intersection Area}}{\text{Union Area}} \tag{2}$$

To robustly assess the performance of our proposed architecture, we employed the K-Fold Cross-Validation method. The entire dataset of 1,246 images was partitioned into 5 non-overlapping folds. The training and evaluation process was iterated 5 times, and in each iteration, a different fold was held out as the test set while the remaining four folds were used for training. The final performance is reported as the mean and standard deviation of the IoU scores obtained across the 5 test folds, providing a reliable and unbiased estimate of the model's generalization capability.

Furthermore, to provide a clear benchmark for our results, we implemented and trained a DeepLabV3+ model with a MobileNetV2 encoder as a baseline for comparison. This baseline model was subjected to the exact same 5-fold cross-validation protocol, ensuring a fair and direct comparison of performance.

F. Processing the Segmentation Mask to Obtain the Navigation Guide

The output of the convolutional neural network is a segmentation mask, typically with an approximately triangular shape.



(a) Example of a non-ideal mask

(b) Application of filters for outlier removal

Fig. 5: Segmentation mask processing for non-ideal case

To transform this segmented region into a usable guidance path, it is necessary to efficiently process the mask's data and extract a reference trajectory.

The initial step involves dividing the mask into several horizontal slices, with the number of slices ranging from 1 to 320, depending on the desired resolution. For each slice, the centroid is computed, resulting in a sequence of representative points across the segmented area. To model the trajectory, a second-degree polynomial regression is fitted to these centroids, enabling the approximation of both straight and curved rows.

A second-degree polynomial was chosen over more complex alternatives like splines due to its computational efficiency and its sufficiency for this application. The crop rows in the dataset primarily feature smooth curves, which are well-approximated by a quadratic function without the overhead of more flexible models. Furthermore, robust estimators like RANSAC were not required, as our pipeline already includes a dedicated filtering stage to remove spurious centroids prior to the regression.

Figure 4a shows the results for $N = 8$, where the horizontal slices are shown in gray and their centroids in blue. Figure 4b shows the polynomial curve fitted to these points, which constitutes the navigation guide.

This example represents an ideal case in which the segmentation mask is well-defined and free from noise. However, in real-world scenarios, the mask may contain distortions, noise, or outliers, which can negatively impact the quality of the extracted trajectory. Figure 5a presents such a scenario, where spurious points near the bottom of the mask cause a deviation in the estimated path.

To mitigate these issues, a sequence of filtering steps is applied to detect and remove outliers:

- **Geometric Filter:** This step leverages the expected geometric behavior of the triangular-shaped mask. Let $C = \{c_1, c_2, \ldots, c_n\}$ be the set of centroids and $D = \{d_1, d_2, \ldots, d_n\}$ be the set of average horizontal distances from each centroid to the lateral borders of the image. In an ideal triangle, these distances should increase from top to bottom. Thus, for all $i < j$, we expect $d_i < d_j$. Any point c_j for which $d_i > d_j$ is considered an outlier and

TABLE II: Comparative performance of the models. The IoU is reported as Mean ± Std Dev. Inference time was measured on the RK3588 platform (FP16 precision).

Model	IoU (%)	Speed (FPS)
Our Model	**94.76% ± 0.24%**	**46.4**
DeepLabV3+ (MobileNetV2)	93.34% ± 1.0%	39.2

removed:

$$C_{\text{del}} = \{c_j \mid d_i > d_j,\ i < j\}$$

The filtered set is then:

$$C_{\text{filtered}} = C \setminus C_{\text{del}}$$

- **Isolation Forest Filter:** To further refine the centroid set, an anomaly detection algorithm (Isolation Forest) is applied to C_{filtered}. This method identifies outliers based on deviations from the spatial distribution of the remaining points. The final set of centroids is:

$$C_{\text{final}} = \text{IsolationForest}(C_{\text{filtered}})$$

Figure 5b illustrates the result of this process. Triangles mark outliers removed by the geometric filter, red "x"s denote those discarded by the Isolation Forest, and blue dots represent the final centroids used to compute the navigation guide.

Finally, a second-degree polynomial is re-fitted to the cleaned set C_{final}, ensuring a more reliable and accurate trajectory for guiding autonomous navigation in agricultural rows.

III. Results

This section presents the comparative performance of our proposed model against the DeepLabV3+ baseline, evaluated using the 5-fold cross-validation protocol. The primary quantitative results are summarized in Table II.

The IoU learning curves for each model across the five folds are illustrated in Figure 6 and Figure 7.

The converted model sizes for RK3588 deployment were 13MB for the proposed model and 17MB for DeepLabV3+. Average inference speeds over 5000 runs were 46.4 FPS and 39.2 FPS, respectively.

Finally, Figure 8 presents several qualitative examples of the model's segmentation output on sample images from the dataset.

IV. Discussion

The experimental results confirm that our proposed model outperforms the DeepLabV3+ baseline in the key areas of accuracy, stability, and computational efficiency.

The superiority in accuracy is demonstrated by the higher mean IoU score (94.76% vs. 93.34%). However, the most compelling finding is the model's stability. Our architecture achieved a standard deviation of only 0.24%, which is four times lower than the baseline's 1.0%, indicating that our model's performance is highly consistent and reliable across different data subsets. Furthermore, the lightweight nature of

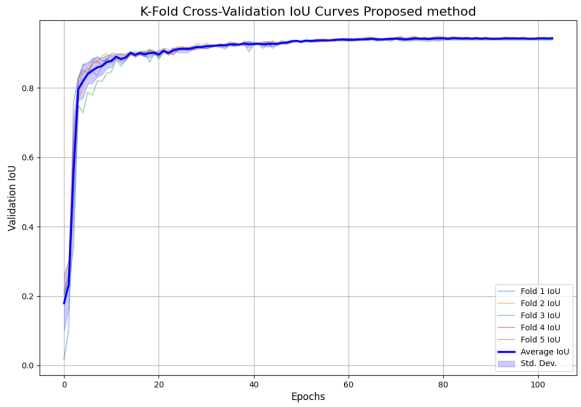


Fig. 6: IoU convergence curves for our proposed model across the 5 cross-validation folds. The bold line represents the mean performance.

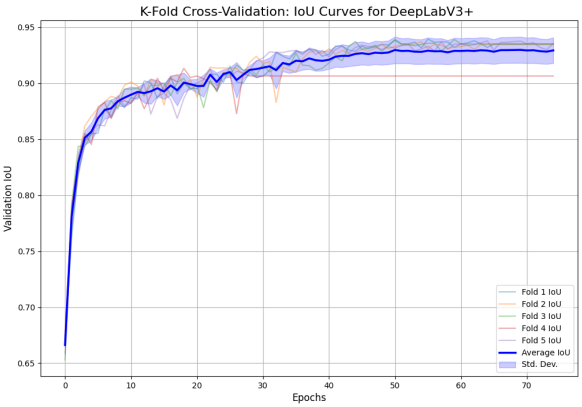


Fig. 7: IoU convergence curves for the DeepLabV3+ baseline model across the 5 cross-validation folds.

the model is evidenced by its compact file size of 13 MB, compared to the 17 MB of the DeepLabV3+ baseline.

In terms of computational performance, our model's faster inference speed (46.4 FPS vs. 39.2 FPS) further validates its efficient design. This processing speed is essential for deployment on resource-constrained embedded systems, such as the RK3588, where high-throughput processing is a prerequisite for practical on-vehicle perception tasks.

The learning curves in Figure 6 visually support the model's stability, showing smooth convergence across all folds. The qualitative examples in Figure 8 confirm that the high IoU scores translate to visually precise segmentation masks in challenging real-world scenes. These examples also demonstrate that the local path planning algorithm effectively generates smooth and appropriate trajectories from the segmentation output, even for curved rows.

Despite these positive results, we acknowledge certain limitations that open avenues for future work. The dataset, while

Fig. 8: Qualitative examples of the proposed method's output, showing the predicted segmentation mask overlaid on the original images.

diverse in terms of lighting and terrain, was collected from a single property; evaluating the model on a wider range of plantations is necessary to further assess its generalization. Furthermore, the evaluation of the local path planning component was qualitative. A quantitative assessment, for instance by comparing the generated path against an expert-driven trajectory, should be developed. Finally, extending the model's capabilities to identify obstacles on the path, such as people or potholes, would be a valuable next step toward developing a more complete and safe perception system.

V. Conclusion

In this work, we successfully developed and validated essential components for an agricultural drive-assist system: a lightweight semantic segmentation model for crop row perception and a local path planning algorithm that generates a guidance path. Through rigorous 5-fold cross-validation, our proposed model, which pairs a MobileNetV3 encoder with a U-Net style decoder, proved its superiority over a DeepLabV3+ baseline. It achieved a mean Intersection over Union (IoU) of 94.76% with an impressively low standard deviation of ±0.24%, highlighting its accuracy and stability.

The practical viability of these components was confirmed by their computational performance. When benchmarked on an RK3588 System-on-Chip (SoC), the model demonstrated a high-speed inference rate of 46.4 FPS. These results collectively validate our approach as a robust, stable, and efficient foundation for vision-based guidance on resource-constrained agricultural hardware. The system presents a powerful and accessible alternative.

As future work, we propose extending the model to segment additional classes, such as obstacles—for instance, potholes or people—to further enhance system safety. Developing a framework for the quantitative evaluation of the planned guidance paths will also be a key step in creating a comprehensive and fully reliable guidance solution.

Acknowledgment

The authors would like to thank AGRES Sistemas Eletrônicos for providing the equipment and resources used in the development of this study.

References

[1] Food and Agriculture Organization, "How to feed the world in 2050," Available online: https://www.fao.org/fileadmin/templates/wsfs/docs/expert_paper/How_to_Feed_the_World_in_2050.pdf, 2009.

[2] R. Sharma, "Artificial intelligence in agriculture: A review," in *2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)*, 2021, pp. 937–942.

[3] E. Vrochidou, D. Oustadakis, A. Kefalas, and G. Papakostas, "Computer vision in self-steering tractors," *Machines*, vol. 10, p. 129, 02 2022.

[4] S. K. Panda, Y. K. Lee, and M. K. Jawed, "Agronav: Autonomous navigation framework for agricultural robots and vehicles using semantic segmentation and semantic line detection," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2023, pp. 6272–6281. [Online]. Available: https://api.semanticscholar.org/CorpusID:258049183

[5] E. Mendez, J. Piña Camacho, J. Escobedo Cabello, and A. Gómez-Espinosa, "Autonomous navigation and crop row detection in vineyards using machine vision with 2d camera," *Automation*, vol. 4, no. 4, pp. 309–326, 2023.

[6] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[7] E. Liu, J. Monica, K. Gold, L. Cadle-Davidson, D. Combs, and Y. Jiang, "Vision-based vineyard navigation solution with automatic annotation," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 4234–4241.

[8] P. Jaquetti, "Row segmentation and local path planning in fruit plantations using lightweight neural networks," Zenodo, jul 2025. [Online]. Available: https://doi.org/10.5281/zenodo.16421899

[9] D.-V. Giurgi, T. Josso-Laurain, M. Devanne, and J.-P. Lauffenburger, "Real-time road detection implementation of unet architecture for autonomous driving," in *2022 IEEE 14th Image, Video, and Multidimensional Signal Processing Workshop (IVMSP)*, 2022, pp. 1–5.

[10] Y. Hou, Z. Liu, T. Zhang, and Y. Li, "C-unet: Complement unet for remote sensing road extraction," *Sensors*, vol. 21, no. 6, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/6/2153

[11] A. de Souza Inácio and H. Lopes, "Epynet: Efficient pyramidal network for clothing segmentation," *IEEE Access*, vol. 8, pp. 187 882–187 892, 10 2020.