Segmentation of Foreground Row Trees in Apple Orchard Images Collected by Ground Vehicles

Marina A. Merzliakova[1], Boris M. Shurygin[2], Alexei E. Solovchenko[2], Andrey S. Krylov[1], Dmitry V. Sorokin[1]

[1] Laboratory of Mathematical Methods of Image Processing, Faculty of Computational Mathematics and Cybernetics, Lomonosov Moscow State University, Moscow, Russia - s02200283@gse.cs.msu.ru, (kryl, dsorokin)@cs.msu.ru
[2] Faculty of Biology, Lomonosov Moscow State University, Moscow, Russia - shu_b@mail.ru, solovchenkoae@my.msu.ru

Keywords: Object Segmentation, Apple Tree Images, Convolutional Neural Networks, Trees Segmentation, Precision Farming.

Abstract

This paper proposes a fully automatic method for the segmentation of foreground row trees in industrial apple orchard images. The segmentation is based on analyzing a combination of a depth map constructed by the Marigold diffusion model and a model depth map created using automatically detected vanishing lines. The output of the method is a binary mask of the selected foreground trees. These masks can be used in subsequent stages of the image processing pipeline to discard false detections in the fruit counting module. The proposed method was evaluated as a preprocessing step for an apple detection method using the OrchardAppleDet-MSU dataset. Experiments showed that the proposed method can improve the quality of apple detection by 1-3%.

1. Introduction

In modern agriculture, many tasks that are currently performed manually by people can be automated. The use of digital technology to automate agricultural processes leads to increased efficiency and improved product quality. In particular, this automated approach helps simplify apple harvest assessment. One way to automatically analyze images of apple trees is by using neural network methods. This significantly improves the accuracy of crop size assessment (Wang et al., 2013) and reduces production costs.

With the development of information technology, special systems have been created to automate precision agriculture (Zhao et al., 2016). These systems include robots programmed to perform labor-intensive yet necessary tasks such as transplanting, spraying, pruning, harvesting, and crop assessment. Performing any of these tasks requires extracting information from vision sensors. Accurate calculation of crop volume is essential, as farmers base critical economic and industrial decisions on this data. Therefore, one of the primary objectives in computerizing farm and garden operations is the detection and segmentation of fruits for quantitative analysis. To address this, numerous solutions have been proposed, including both neural network-based methods and classical mathematical approaches.

For example, in (Slaughter and Harrell, 1987), the authors developed a method based on determining the intensity threshold to generate a binary image. In the resulting binary mask, large segmented areas are recognized as fruits. This method is easy to implement but highly dependent on varying light conditions. In (Changyi et al., 2015), the authors propose using the Hough transform to obtain binary images with extracted contours of objects. This approach works well on a simple background but is less applicable in complex, structured environments, such as dense orchards. Another idea is to compare fruits by shape and texture (Zhao et al., 2005). However, this method is also highly sensitive to light conditions and tree overlaps.

A neural network solution to the problem of fruit detection was proposed in (Tian et al., 2019). The authors use an improved YOLO-V3 model (Jiang et al., 2022) to detect apples in orchards at different stages of growth. The advantage of the model is that YOLO transforms the detection problem into a regression problem. The network's output generates the bounding box coordinates and probabilities of each class directly using regression. This significantly increases the detection speed.

Another neural network solution to the problem was presented in (Nesterov et al., 2023). Apple detection is carried out using the Mask R-CNN neural network (He et al., 2017). As a two-stage detector, Mask R-CNN differs from single-stage detectors (for example, YOLO) by providing higher prediction accuracy.

The above methods address the problem of detecting all available fruits in the image. However, to correctly estimate the volume of the harvest, it is necessary to avoid counting some apples twice and to exclude those that are wasted. To achieve this, it is important to discard apple detections from background trees and the ground. Accordingly, it is essential to solve the problem of segmenting trees in the front row and selecting trees located far from the camera. To tackle this problem, several approaches can be used: the first is based on semantic segmentation methods, and the second is based on constructing a depth map.

In (Chen et al., 2021), the authors proposed using three different neural networks independently for semantic segmentation. The first, Pix2P, is a modified generative adversarial network (Creswell et al., 2018) that works well for pixel matching. The second, U-Net (Ronneberger et al., 2015), is a convolutional neural network designed for accurate semantic segmentation. The third network, DeepLabV3 (Chen et al., 2017), is a convolutional neural network that employs a set of spatial pyramids (ASPP - Atrous Spatial Pyramid Pooling) in combination with an encoder-decoder methodology. As a result, the DeepLabV3 network demonstrated the best accuracy, although its accuracy drops significantly on trees that overlap with others. A method for semantic segmentation of trunks and branches using the Kinect V2 sensor and the SegNet segmentation network was proposed in (Majeed et al., 2018). The method showed good results, but it requires depth information and is tailored

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-X-2-W1-2024-25-2024 | © Author(s) 2024. CC BY 4.0 License.

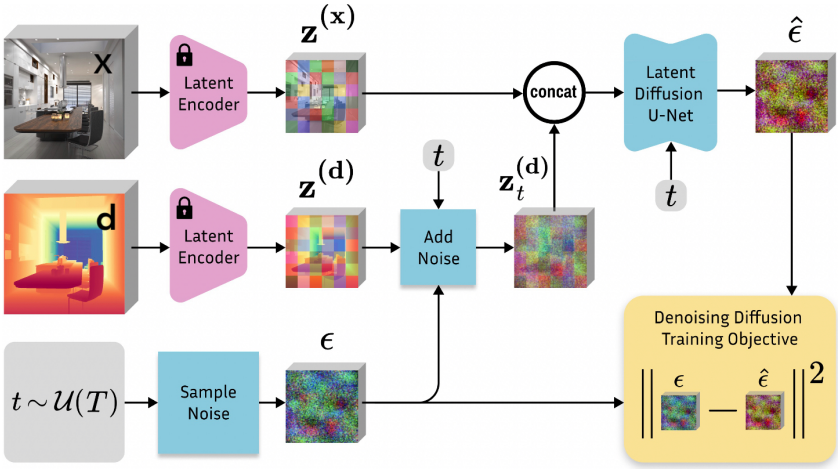


Figure 1. Marigold architecture.

for specific sensor.

The second method to identify trees in the front row is to analyze the depth predicted from the initial images. Existing depth estimation methods can be divided into two types: classical methods based on probabilistic graphical models (Saxena et al., 2005), and methods based on convolutional neural networks (Garg et al., 2016). Compared to classical methods, neural network-based methods significantly increase the accuracy and reliability of predictions. The models can be either supervised or unsupervised. In the first case, the method requires a large number of images equipped with the ground truth depth maps (Ren et al., 2020; Saxena et al., 2007). Therefore, the supervised approach requires considerable time and specialized techniques to create a training dataset with detailed depth information. The unsupervised learning methods look more attractive as they do not require ground truth depth data acquired with depth sensors such as 3D cameras or LIDARs. For example, such methods can utilize information about disparities and differences between pixels in sequential frames.

One of the unsupervised methods is MonoDepth2 (Godard et al., 2019). MonoDepth2 is based on calculating the loss between the input image and the reconstructed one using outputs from subnets: a depth map and a transformation matrix between frames. Another example of an unsupervised model is Marigold (Ke et al., 2023). This is a diffusion model that, unlike MonoDepth2, was trained on a much larger number of datasets. Marigold works by transforming an unknown distribution of training data into a simple, already known distribution by adding noise, and then using the U-Net Latent Diffusion model (Ronneberger et al., 2015; Rombach et al., 2022) to obtain a depth map from the simplified distribution.

In this work, we have developed a method for segmenting foreground trees in images of apple orchards. The segmentation is based on the joint analysis of the depth map estimated using the Marigold model and a model depth map constructed using automatically detected vanishing lines. The method was evaluated on the OrchardAppleDet-MSU dataset[1] as a preprocessing step for the apple detection approach (Nesterov et al., 2023). The results showed that utilizing the segmentation masks obtained by the proposed method to filter detection results improves the quality metrics of apple detection by 1-3%.

[1] https://imaging.cs.msu.ru/en/research/apples

2. Methods

In this section, we describe the pipeline for segmenting foreground row trees. Given the input image I, we construct the binary segmentation mask M_{final}. First, we estimate the depth map of the image (Section 2.1). Then, the model depth map is estimated (Section 2.3) using the automatically detected vanishing lines (Section 2.2). Finally, the binary mask is created by thresholding the depth map using the model depth map (Section 2.4).

(a) Input image (b) Estimated depth map

Figure 2. The result of depth map estimation.

2.1 Depth estimation

We estimated the depth map using the Marigold model (Ke et al., 2023). Marigold is a diffusion model, which means it transforms noise into a representative sample of data. The architecture of the model is shown in Fig. 1. The basic idea is to transform the unknown distribution of the training data into a simple, known distribution, and then reverse the process. At each step, the original image is gradually degraded by adding noise and then passed through a neural network to reconstruct the image. As a result, the model learns to estimate both the original data distribution and the added noise. The trained network, starting with a simple noise distribution, can then create a new image that represents the original training dataset. Marigold uses the Latent Diffusion U-Net model (Ronneberger et al., 2015; Rombach et al., 2022) as the neural network for working with noise. We used the pretrained on Virtual KITTI 2 (Cabon et al., 2020) Marigold model to estimate the depth for our images.

Fig. 2 shows an example of the result from the Marigold model, where warm colors indicate areas close to the camera, and cold colors indicate areas farther from the camera.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-X-2-W1-2024-25-2024 | © Author(s) 2024. CC BY 4.0 License.

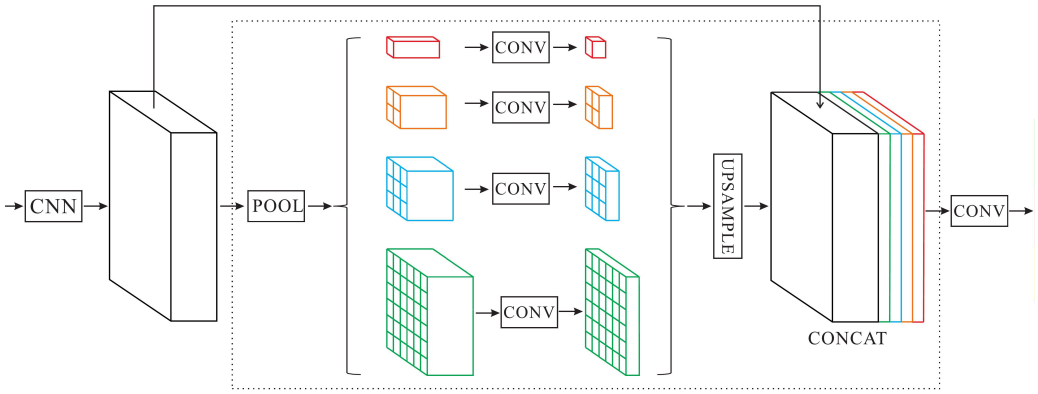


Figure 3. PSPNet architecture.

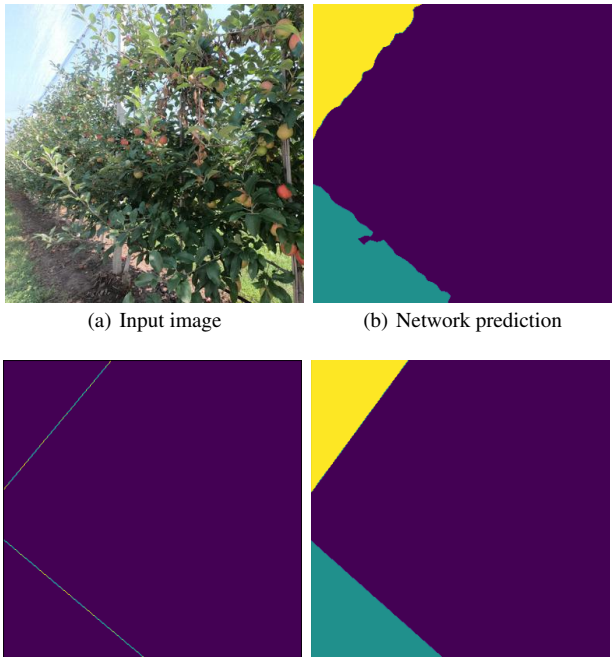
2.2 Vanishing lines detection

To construct the model depth map we first need to detect vanishing lines. Vanishing lines are identified using the Pyramid Scene Parsing Network (PSPNet) (Zhao et al., 2017). The network was trained on the labeled OrchardAppleDet-MSU dataset. The labeling was performed by indicating the endpoints of the vanishing lines corresponding to the tree trunks and tree tops. Than the lines were converted to the ground truth segmentation masks with three classes: sky, ground, background (see Fig. 5). The dataset was divided into 3 parts: training set (127 images), validation set (10 images), testing set (10 images). The images were cropped and downsampled to 384x384. Also, the dataset was augmented (Cubuk et al., 2019) with using the functions ColorJitter, RandomPerspective, RandomRotation. ColorJitter performs arbitrary changes in brightness, contrast, saturation and hue of an image. RandomPerspective performs a random transformation of an image's perspective with a given probability. RandomRotation rotates an image by a random angle from a specified range. The presence of augmentation allows the network to be used for datasets taken at a different angle and in other light conditions.

The PSPNet architecture (Fig. 3) includes two components: the main convolutional network and the pyramid module. The main network, which uses a ResNet architecture (Koonce and Koonce, 2021), extracts features from the input image, resulting in feature maps that are $\frac{1}{8}$ the size of the original image. The pyramid module is needed to combine information from different image scales. It creates a four-level pyramid that covers the entire image, half the image, and smaller parts of it. Each level is processed by additional convolutional layers to obtain a more complete representation of the image at different scales. Information from the pyramid is integrated into a global level and then combined with the original feature map from the ResNet model. The combined information is then passed through a convolutional layer to create the final prediction map.

The network segments the sky and ground regions on the left side of the image. An example of a trained network's prediction can be seen in Fig. 4(b).

All small regions in the prediction were removed using morphological post-processing (Sreedhar and Panlal, 2012). The vanishing lines were estimated using linear regression (Montgomery et al., 2021) over the points defined by the boundaries between the segmented classes. The resulting vanishing lines are depicted in Fig. 4(c).



(a) Input image (b) Network prediction

(c) Vanishing lines extracted with linear regression (d) Ground truth

Figure 4. Vanishing lines detection.

2.3 Model depth map

The trees in the depth image have an unstructured appearance, and the depth map is relatively high frequency. Additionally, there is a strong perspective in the image, making direct depth thresholding insufficient for segmenting the trees in the foreground row. However, we can leverage the fact that the trees in the orchards are planted in straight lines. Thus, we can roughly model the depth map of the nearest planted tree row as a flat surface (as if it were a fence) and use the model depth map in conjunction with the estimated depth map for a joint analysis.

A model depth map is constructed using previously detected vanishing lines. We assume that the depth value decreases uniformly along the straight line passing through a point in the image and the vanishing lines' intersection point O (see Fig. 5). The pixels that fall into the triangular areas cut off by the vanishing lines are filled with zeros. For each pixel in the remaining region, the depth value I is determined using the following formula:

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-X-2-W1-2024-25-2024 | © Author(s) 2024. CC BY 4.0 License.

$$I = \frac{\rho(O,P)}{\rho(O,M)}k\,, \tag{1}$$

where P is the point where it is necessary to determine the depth, M is the intersection point of the line passing through O and P and the vertical line, $\rho(O,P)$ and $\rho(O,M)$ are the lengths of the segments OP and OM, respectively, and the coefficient k represents the depth value at the right edge of the model depth image. The k value is individual for each image, and the method for obtaining it is described below.

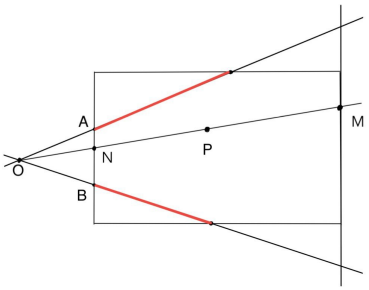


Figure 5. Geometric constructions.

The value of k affects the surface slope of the model depth map. To accurately highlight areas where the values on the real and model depth maps differ the most, k should be selected such that the surface of the model depth map deviates as little as possible from the actual surface of the estimated depth map. Therefore, in selecting k, it is necessary to solve the following minimization problem:

$$\sum_i d_i^2 \longrightarrow min, \tag{2}$$

where d_i is the difference for each ith pixel between the model depth values on the estimated depth values.

Figure 6. The result of constructing a model depth map.

2.4 Foreground Tree Segmentation and Postprocessing

To summarize, the foreground row tree segmentation algorithm contains the following steps for each image:

1. Estimation of the image depth map;
2. Vanishing lines detection;
3. Construction of a model depth map for the foreground row trees based on the estimated depth map and vanishing lines;
4. Creation of a binary mask by choosing the threshold parameters;
5. Post-processing the mask to remove high-frequency information.

Fig. 7 shows two surfaces of the model and estimated depth maps. Both surfaces are plotted on the same graph, making areas where the value of the estimated depth map is significantly less than the value on the model depth map clearly visible. These areas correspond to parts of the image where pixels are not included in the foreground row tree mask.

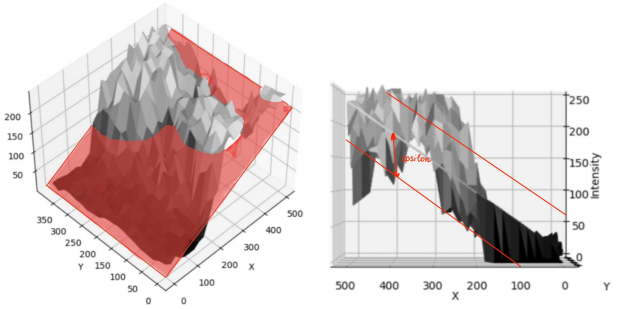


Figure 7. Surfaces of model and real depth maps.

To create the binary mask, the triangular areas highlighted by the vanishing lines are initially zeroed, as there are obviously no trees with apples in them. Then, all areas for which the difference between the value of the real and model depth map is less than ε (Fig. 7), or the value of the real depth map is less than 30 (representing very distant trees), are set to zero.

The resulting mask contains many small, noisy dark and bright regions, so it needs post-processing to fill holes, remove noise, and smooth the mask contours. To post-process the binary mask M, we convert it to a real-valued image, blur it with a Gaussian kernel, and threshold the blurred image with a value of 0.5.

$$M_{final} = \begin{cases} 1, & M * G_\sigma > 0.5\,, \\ 0, & M * G_\sigma < 0.5\,, \end{cases} \tag{3}$$

where

$$G_\sigma = \frac{1}{2\pi\sigma}\exp^{-(x^2+y^2)/2\sigma^2}\,. \tag{4}$$

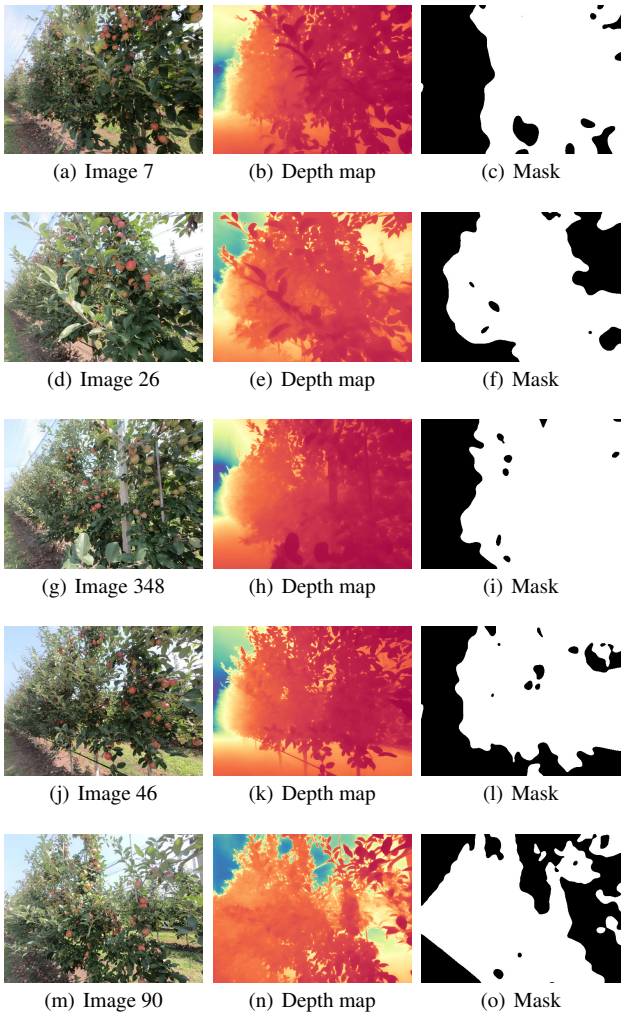
The example of a constructed binary mask before and after post-processing is shown in Fig. 8.

Figure 8. Initial image and binary masks before and after post-processing.

3. Results

The proposed approach was evaluated on the OrchardAppleDet-MSU dataset, which consists of 147 images 3000×4000 pixels of apple orchards collected by the Faculty of Biology, Lomonosov Moscow State University. An example of the resulting depth maps and masks can be seen in Fig. 9.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-X-2-W1-2024-25-2024 | © Author(s) 2024. CC BY 4.0 License.



(a) Image 7 (b) Depth map (c) Mask
(d) Image 26 (e) Depth map (f) Mask
(g) Image 348 (h) Depth map (i) Mask
(j) Image 46 (k) Depth map (l) Mask
(m) Image 90 (n) Depth map (o) Mask

Figure 9. Results of the method.

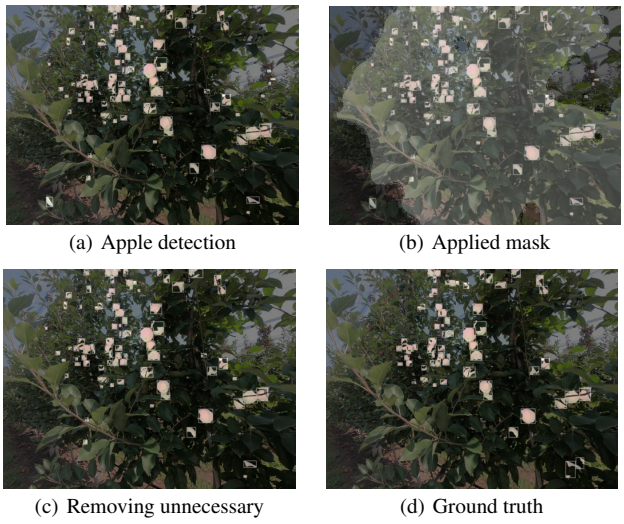
Fig. 9(d) shows an example of an image where a background tree is visible in the gap between the foreground trees. This tree is not segmented in the resulting binary mask.

Some masks contain a large number of small areas that do not carry semantic information and can spoil further results when these masks are used on the apple detection stage of the pipeline. Examples of such masks are shown in Fig. 9(l,o). This flaw mainly appears on masks where there are large areas of sky on the right side of the image.

In the mask in Fig. 9(o) not only the nearest trees were highlighted, but also all the trees in the front row. This happened because the foreground of the image was obscured by a branch, so Marigold highlighted this area as close in the depth map (Fig. 9(n)). This inaccuracy in the mask is not critical, since the branches of nearby trees cover distant trees, therefore all detected apples on the trees will meet the requirements.

As a result, despite the presence of a small number of errors, the majority of the resulting binary masks correctly segment foreground row trees.

To evaluate the performance, the method was used to discard false detections of the Mask R-CNN model (Nesterov et al., 2023). Fig. 10(a) shows the detected apples predicted by the neural network. In Fig. 10(b) the foreground row trees mask obtained by the proposed method is overlaid with the detection results. It can be seen that the apples on the second row trees and the apples that fell to the ground are outside of the obtained foreground row trees mask. Therefore, in Fig. 10(c) these detections have been discarded, that correspond to the ground truth data for apple detection Fig. 10(d).



(a) Apple detection (b) Applied mask
(c) Removing unnecessary (d) Ground truth

Figure 10. Example of applying the method to discard false detections of apples.

In addition, we carried out a quantitative assessment of the effectiveness of the proposed solution. To assess its quality, we calculated the mAP (mean Average Precision) (Varsadan et al., 2009) and IoU (Intersection over Union) (Rezatofighi et al., 2019) metrics for apple detection algorithm with and without application of the foreground row trees mask. The average values of the mAP and IoU metrics without applying the masks were 0.395 and 0.881, respectively. After applying the foreground tree masks, the metric values increased to 0.405 and 0.887. Thus, the quality metrics for fruit detection increased by 2.5% and 0.7% after using the constructed masks. In the example shown in Fig. 10, before applying the mask the metrics were mAP = 0.4664, IoU = 0.873, and after applying the mask the metrics became mAP = 0.4695, IoU = 0.883.

4. Consclusion

In this paper, we have developed a fully automatic method for the segmentation of foreground row trees. The segmentation relies on the joint analysis of the depth map estimated with the Marigold model and the model depth map constructed using automatically detected vanishing lines. The proposed approach was evaluated on the OrchardAppleDet-MSU dataset. The results showed that incorporating this method as a preprocessing step in the apple detection pipeline improves the quality metrics (mAP and IoU) by 1-3%. The code is publicly available[2].

References

Cabon, Y., Murray, N., Humenberger, M., 2020. Virtual kitti 2. *arXiv preprint arXiv:2001.10773*.

Changyi, X., Lihua, Z., Minzan, L., Yuan, C., Chunyan, M., 2015. Apple detection from apple tree image based on BP

[2] https://github.com/marinam34/apple_image_analysis

neural network and Hough transform. *International Journal of Agricultural and Biological Engineering*, 8(6), 46–53.

Chen, L.-C., Papandreou, G., Schroff, F., Adam, H., 2017. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 1–14.

Chen, Z., Ting, D., Newbury, R., Chen, C., 2021. Semantic segmentation for partially occluded apple trees based on deep learning. *Computers and Electronics in Agriculture*, 181, 1–7.

Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., Bharath, A. A., 2018. Generative adversarial networks: An overview. *IEEE signal processing magazine*, 35(1), 53–65.

Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., Le, Q. V., 2019. Autoaugment: Learning augmentation strategies from data. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 113–123.

Garg, R., Bg, V. K., Carneiro, G., Reid, I., 2016. Unsupervised cnn for single view depth estimation: Geometry to the rescue. *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14*, Springer, 740–756.

Godard, C., Mac Aodha, O., Firman, M., Brostow, G. J., 2019. Digging into self-supervised monocular depth estimation. *Proceedings of the IEEE/CVF international conference on computer vision*, 3828–3838.

He, K., Gkioxari, G., Dollár, P., Girshick, R., 2017. Mask r-cnn. *Proceedings of the IEEE international conference on computer vision*, 2961–2969.

Jiang, P., Ergu, D., Liu, F., Cai, Y., Ma, B., 2022. A Review of Yolo algorithm developments. *Procedia Computer Science*, 199, 1066–1073.

Ke, B., Obukhov, A., Huang, S., Metzger, N., Daudt, R. C., Schindler, K., 2023. Repurposing diffusion-based image generators for monocular depth estimation. *arXiv preprint arXiv:2312.02145*, 1–33.

Koonce, B., Koonce, B., 2021. ResNet 50. *Convolutional neural networks with swift for tensorflow: image recognition and dataset categorization*, 63–72.

Majeed, Y., Zhang, J., Zhang, X., Fu, L., Karkee, M., Zhang, Q., Whiting, M. D., 2018. Apple tree trunk and branch segmentation for automatic trellis training using convolutional neural network based semantic segmentation. *IFAC-PapersOnLine*, 51(17), 75–80.

Montgomery, D. C., Peck, E. A., Vining, G. G., 2021. *Introduction to linear regression analysis*. John Wiley & Sons.

Nesterov, D. A., Shurygin, B. M., Solovchenko, A. E., Krylov, A. S., Sorokin, D. V., 2023. A CNN-Based Method for Fruit Detection in Apple Tree Images. *Computational Mathematics and Modeling*, 33(3), 354–364.

Ren, H., Raj, A., El-Khamy, M., Lee, J., 2020. Suwlearn: Joint supervised, unsupervised, weakly supervised deep learning for monocular depth estimation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 1–9.

Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., Savarese, S., 2019. Generalized intersection over union: A metric and a loss for bounding box regression. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 658–666.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B., 2022. High-resolution image synthesis with latent diffusion models. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10684–10695.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, Springer, 234–241.

Saxena, A., Chung, S., Ng, A., 2005. Learning depth from single monocular images. *Advances in neural information processing systems*, 18, 1–8.

Saxena, A., Schulte, J., Ng, A. Y. et al., 2007. Depth estimation using monocular and stereo cues. *IJCAI*, 7, 2197–2203.

Slaughter, D. C., Harrell, R. C., 1987. Color vision in robotic fruit harvesting. *Transactions of the ASAE*, 30(4), 1144–1148.

Sreedhar, K., Panlal, B., 2012. Enhancement of images using morphological transformation. *arXiv preprint arXiv:1203.2514*, 1–18.

Tian, Y., Yang, G., Wang, Z., Wang, H., Li, E., Liang, Z., 2019. Apple detection during different growth stages in orchards using the improved YOLO-V3 model. *Computers and electronics in agriculture*, 157, 417–426.

Varsadan, I., Birk, A., Pfingsthorn, M., 2009. Determining map quality through an image similarity metric. *RoboCup 2008: Robot Soccer World Cup XII 12*, Springer, 355–365.

Wang, Q., Nuske, S., Bergerman, M., Singh, S., 2013. Automated crop yield estimation for apple orchards. *Experimental Robotics: The 13th International Symposium on Experimental Robotics*, Springer, 745–758.

Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J., 2017. Pyramid scene parsing network. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2881–2890.

Zhao, J., Tow, J., Katupitiya, J., 2005. On-tree fruit recognition using texture properties and color data. *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 263–268.

Zhao, Y., Gong, L., Huang, Y., Liu, C., 2016. A review of key techniques of vision-based control for harvesting robot. *Computers and Electronics in Agriculture*, 127, 311–323.