Optical flow-based branch segmentation for complex orchard environments

Alexander You¹, Cindy Grimm¹, Joseph R. Davidson¹

Abstract—Machine vision is a critical subsystem for enabling robots to be able to perform a variety of tasks in orchard environments. However, orchards are highly visually complex environments, and computer vision algorithms operating in them must be able to contend with variable lighting conditions and background noise. Past work on enabling deep learning algorithms to operate in these environments has typically required large amounts of hand-labeled data to train a deep neural network or physically controlling the conditions under which the environment is perceived. In this paper, we train a neural network system in simulation only using simulated RGB data and optical flow. This resulting neural network is able to perform foreground segmentation of branches in a busy orchard environment without additional real-world training or using any special setup or equipment beyond a standard camera. Our results show that our system is highly accurate and, when compared to a network using manually labeled RGBD data, achieves significantly more consistent and robust performance across environments that differ from the training set.

I. INTRODUCTION

Accurate sensing of the environment is an integral component of any robotic system. One common problem hindering many computer vision algorithms for specialty crop production (i.e. fresh market fruits and vegetables) is that outdoor environments — like fruit orchards — are very noisy with variable lighting conditions and highly complex and unstructured environments. In these environments, it is often useful to distinguish between parts of the image that are part of the foreground versus those in the background, i.e. to perform foreground segmentation. Fast and reliable foreground segmentation in orchard environments is directly applicable to a variety of orchard tasks (e.g. canopy estimation and blossom counting) and is also useful for many downstream 2.5D and 3D computer vision and control problems (e.g. localizing objects for pruning, thinning, and picking).

Foreground segmentation in orchards is a difficult problem, and because of this, many algorithms and research experiments are performed in highly controlled environments to minimize the effects of background noise. In our previous pruning experiments [1], we trained a reinforcement learning system for dormant tree pruning using a computer vision system that, while successful, required the lab tree setup to be next to a blank wall to eliminate background noise. Similar examples can be found in many other works: [2] and [3] require backdrops to reconstruct tree models via photogrammetry, while [4] uses a trailer that encompasses the target grape vine, isolating it in the scene from the adjacent vineyard rows and enabling complete control of the, essentially, indoor lighting conditions. Deep learning using depth data is also a common approach. However, training these networks typically requires a large amount of hand-labeled data, and the resulting networks can struggle with environmental conditions that differ even slightly from the training data.

As humans, one way we contend with the issue of background noise is to move our heads to get different views of the scene. Our goal is to apply this principle to a camera mounted in an eye-in-hand robotic configuration, to yield dense optical flow estimates (middle), allowing us to perform robust foreground branch segmentation (right).
the image estimating their speeds across frames. This optical flow serves as a proxy for depth data, enabling us to do real-time foreground object segmentation.

In this paper, we introduce a deep neural network for producing foreground masks for trees in planar orchard environments by augmenting RGB data with optical flow data computed from the previous RGB frame. The augmented image with RGB and optical flow is fed into the neural network to produce a segmentation mask. To our knowledge, our approach is the first one for segmentation of branches in outdoor environments that is trained entirely in simulation, as well as the first to make use of optical flow data instead of relying on depth data. By training our network this way, we eliminate the need to do any manual labeling of ground truth data. We demonstrate that both of these components combined allow for consistent and accurate segmentation results by evaluating the performance of a suite of comparison networks in four outdoor environments and one indoor environment and showing that only our system was able to perform consistently across all of them.

II. RELATED WORK

Branch detection is an active area of research within agricultural robotics. For this paper, we focus only on techniques used to perform segmentation on 2D images, rather than techniques for 3D reconstruction of trees. Many papers have shown success in using machine learning techniques to create branch masks. Amatya and Karkee [5] train a Bayesian classifier to segment and reconstruct branches on images of sweet cherry trees taken at night, while [6] uses an encoder-decoder network to segment out branches, wires, and fruit in a kiwifruit orchard. Depth information is a common modality for filtering out unwanted background noise: [7], [8] use depth information to filter out all points beyond a specified threshold before feeding the RGB image through a neural network, while [9], [10] feed the depth channel directly into the neural network to filter out background noise. Yang et al. [11] produce a mask with a raw RGB image but post-process the mask with depth data to create an accurate tree model for localization. Depth data has a number of notable limitations, particularly that the quality and characteristics of depth data can vary greatly between different depth cameras. Because of this, realistic depth images are difficult to simulate, requiring training data to be labeled manually, a very time-consuming and arduous process. Our work substitutes optical flow as a proxy for depth data, allowing us to train a network using only simulated images and transfer the resulting network to real data.

Our problem is closely related to the issue of background subtraction (or, equivalently, foreground segmentation) in which the goal is to use a stream of images to detect and extract objects of interest in a scene by removing pixels which are located in the background. Background subtraction is a highly studied area of computer vision, with numerous surveillance-related applications as detailed by [12]. Piccardi [13] offers a survey of classical background subtraction techniques, which typically operate by constructing a model of the background and maintaining it against the most current frame. Bouwmans et al. [14] review the large body of research that has shown success with using convolutional neural networks to perform foreground segmentation, often with robustness to issues such as camera jitter, dynamic backgrounds, and weather conditions that traditional methods struggled with. However, in addition to these networks being highly scene-specific, the vast majority of background subtraction algorithms assume the camera is mostly stationary and the object of interest is moving, whereas in our situation, all objects of interest are stationary while the camera itself is moving.

Our research is also closely related to the areas of object tracking and mask propagation for video data. Yao et al. [15] gives a comprehensive overview of many of these methods. Of particular interest to us is the use of deep neural networks such as SegFlow [16] that simultaneously learn optical flow and segmentation data, ultimately resulting in improved object segmentations when compared to learning them individually. Though we do not train our network in the same way as theirs, we follow a similar idea by using motion induced by camera movement to segment out the objects of interest in the foreground.

III. PROBLEM STATEMENT

Our work focuses primarily on modern, high-density orchard systems, such as the one shown in Figure 2. This Figure shows sweet cherry trees (during the dormant season) trained in an upright fruiting offshoot (UFO) system whereby vertical leader branches growing from the nearly horizontal trunk are tied to trellis wires, creating an inclined, V-style planar canopy. Our goal is to generate a mask that will highlight only the branches attached to the front plane of the trellis, filtering out all other objects including the ground, sky, trees and objects in different rows, trellis wires, and tags and non-organic parts of the foreground.

We assume that our setup for operating in these environments consists of a mobile camera oriented roughly parallel to the planar trellis system. We also assume that the trees themselves are mostly stationary (no excessive
wind). Therefore, when the camera moves translationally, the objects in the foreground will move the most.

IV. OPTICAL FLOW-BASED NETWORK DETAILS

In this section, we describe the architecture and training methodology of our segmentation network, which combines an existing optical flow network (FlowNet2) with the pix2pix GAN (Section IV-A). Notably, we train our network using entirely simulated data (Section IV-B), avoiding the need to label data by hand.

A. Architecture

Our segmentation network is shown in Figure 3. We utilize the pix2pix [17] framework, a Generative Adversarial Network (GAN) [18] designed to convert images in one domain into another one, to perform the segmentation. Our previous work [1], as well as others [10], have shown that the pix2pix network is capable of performing robust segmentation.

In addition to passing the 3-channel RGB image into the GAN, which outputs a 1-channel segmentation mask, we also pass in a 3-channel colorized representation of the optical flow between the current RGB frame and a previous one. To compute the optical flows, we use a fully-sized FlowNet2 network with pre-trained weights [19], which we chose due to the high quality of the optical flows produced as well as its ability to work on simulated images and on image pairs with varying distances.

The combined network’s runtime is nearly real-time, mainly limited by the performance of FlowNet2. On our research computer using an NVIDIA GeForce GTX 1050 Ti, we observed runtimes of the pix2pix network to be around 20 FPS and FlowNet2 to be around 4 FPS, though the original paper for FlowNet2 purports to reach 8 FPS using an NVIDIA GTX 1080.

B. Simulated Image Generation

To create a network that can generalize to real world scenarios without requiring any manual labeling of data, we follow the methodology of [20]. This approach trains a GAN to convert an image of a scene to a simplified, canonical representation (essentially a labeled mask). The key to this approach is training the GAN using simulated images with heavy texture and lighting randomization. They showed that the resulting GAN was able to generalize surprisingly well to real images of the scene. We showed in our previous work [1] that this process also worked for tree branches in a simple lab setup. This paper further augments the process with optical flow data to allow our system to work in complicated environments in which a single RGB image alone is not enough to perform a proper segmentation.

For our experiments, we created a multi-row orchard environment in Blender, an open source 3D modeling software. A typical environment with randomized textures is shown in Figure 4. Each environment was created by randomly lining up 1-5 rows of tree models, where the spacing between trees was about 0.6m and the spacing between rows was about 1.1m. These values were based on measurements from a commercial UFO cherry tree orchard. Each row of trees has 3 trellis wires placed at random vertical offsets between 0-1m above the ground, as well as a rectangular plane on the ground representing the soil area. We also placed between 0 and 2 randomly sized rectangular walls behind the final tree row to simulate nearby walls or foreign objects.

Once the environment is generated, we randomize the textures of the trees (each tree receives the same texture),
ground plane, soil area planes, and trellis wires by picking textures from a set of 5000 simple textures. Each of the walls was textured with one of 1000 random images sourced from the Open Images v6 data set [21], and the lighting of the scene was determined using one of 303 randomly chosen high-dynamic-range images (HDRIs). For all textures, we randomly shifted the hue, saturation, and values of the source images to obtain a wide variety of synthetic environments.

After randomizing the textures as described above, we randomly place a camera so that it faces the front row of the scene and apply a random angular perturbation of up to 15 degrees. We then generated two renders of the scene by moving the camera a random direction by 0.5-2 cm. For each render, we created a corresponding ground truth mask by coloring the foreground trees white, hiding all other objects, and setting the background to black.

Each pair of renders is used to generate a corresponding pair of colorized optical flow images from FlowNet2, giving us two sets of corresponding synthetic images and optical flow images. For our training, we generate a total of 4400 6-channel synthetic-flow pairs, 4000 of which are used for the training phase and 400 of which are used for validation; training details are given in Section IV-C. Figure 4 shows an example of a render, mask, and optical flow image generated from a synthetic pair.

C. Training Methodology

After precomputing the optical flows for each pair of renders, we trained the pix2pix network using the synthetic-flow pairs. Since the optical flows are precomputed, the FlowNet2 network weights are not updated during training, decoupling the optical flow computation from the segmentation.

We trained the network for a maximum of 100000 images. Every 1000 images, we computed the average L1 loss for the validation data set and saved the network weights if the average loss was lower than the previous best. Although in general for GANs L1 loss is not the best criterion for determining network convergence, for this relatively simple segmentation task we would expect that good segmentations would be highly correlated with low L1 losses.

V. COMPARISON NETWORK DETAILS

In this section, we cover the real world data set we collected (Section V-A) that we used to train a number of comparison networks (Section V-B). The networks trained using this data serve as a useful comparison baseline for our own network trained using synthetic data.

A. Real World Training Data Set

To train a comparison network with hand-labeled data, we collected video RGB and depth data from a UFO cherry tree orchard on a sunny day at 1 PM in Prosser, Washington using an Intel (Santa Clara, CA, USA) Realsense D435 camera at a resolution of 640 x 480 pixels. A human operator collected the data by holding the camera in their hand and moving it in primarily translational directions at a distance of roughly 0.3-1m from the plane of trees. For ease of data collection, the depth data was colorized using a color map and appended to the RGB video.

After the data collection, we chose a total of 326 training images and 45 validation images from the videos to manually annotate with ground truth data. An example of RGBD and ground truth data is shown in Figure 5. The labeling process was arduous, as due to the complexity of each image and the necessity to cross reference the source video to determine foreground movement, each image could take up to 10 minutes to label, resulting in a combined effort of over 40 hours just to label 371 images. Furthermore, we expect that there was some amount of label noise introduced by the complexity of the scene as well as ambiguities about whether certain branches qualified as foreground branches or not (e.g. a branch extending from the front trellis wall into the background).

B. List of Comparison Networks

Our goal is to evaluate whether our system — using synthetic data only — obtains comparable accuracy to a system trained using hand-labeled RGBD data while also being robust to a wide variety of environments. All trained networks use the pix2pix architecture and training methodology as described in Sections IV-A and IV-C but with the network inputs changed:

- RGB data: Do we use real RGB data with labeled ground truths, or do we use synthetic data?
- Depth/flow: To assist with depth perception, do we use depth data, computed optical flow, or nothing at all (relying on a single frame of RGB data)?

Our baseline networks are as follows:

- Synthetic with optical flow (Syn+F*), 6 channels: Our proposed network (Section IV) trained using the synthetic image-colored flow image pairs described in Section IV-B
- RGB with colorized depth (RGB+D), 6 channels: Trained using the manually labeled RGBD data described in Section V-A. This setup is comparable to the one described in [10].

To further examine the effects of the use of depth and flow data, we train three other networks for an ablation study:

- Synthetic with no flow (Syn), 3 channels: Trained using the synthetically generated images, but without including the colorized optical flow image.
- RGB with no depth (RGB), 3 channels: Trained using the manually labelled data set, but without including the colorized depth image.
• RGB with optical flow (RGB+F), 6 channels: Trained using the RGB images collected from the main data set, but instead of using the depth image, we use the colorized flow associated with either the frame before or the frame after the image. Since there are two optical flow images per RGB image, this doubles the sizes of the training and validation sets.

VI. EXPERIMENTS

In this section we discuss the various test data sets we use to evaluate the comparison networks (Section VI-A), as well as the metrics used to perform the evaluation (Section VI-B). In addition to assessing each network’s relative performance against our own (Syn+F*), our goal is to examine how each network’s performance degrades as the data set changes. These results are covered in Section VII.

A. Test Image Data Sets

To evaluate the robustness of the trained networks, in addition to the training and validation data from the sweet cherry tree orchard described in Section V-A, we labeled 15 ground truth images each from 5 different environments using the same data collection process as in Section V-A. The environments, shown in Figure 6, are listed here in order of their similarity to the training data set:

1) Main: Separate data collected on the same day and at the same time as the training data.
2) Afternoon: Data collected from the same orchard with sunny conditions on a different day, but at 4 PM instead of 1 PM. Notably, the sun was setting, resulting in image glare and exaggerated contrast.
3) Cloudy: Data collected from the same orchard, but under mostly cloudy conditions, at 5 PM on a different day.
4) Envy spindle trees, 12 PM, cloudy: Data collected from a different orchard of Envy apples grown in a tall, spindle configuration. The day was foggy and cloudy. These images differ visually from the main data set both in the weather conditions and the general appearance of the trees.
5) Lab setup: Data collected in our lab using our experimental setup from our previous work [1] which uses branches from cherry trees. For this setup, the camera was mounted onto a Universal Robots (Odense, Denmark) UR5e robot and manually moved using the control panel as the video data was collected. We have moved the experimental setup away from a blank wall so that a large amount of background clutter is present.

B. Evaluation metrics

To measure the quality of the prediction masks, we first create a binary prediction mask by thresholding all predictions (which range from 0 to 1) that are below 0.5 to 0 and all predictions above 0.5 to 1. We denote the set of all positively identified pixels in the ground truth and predictions as $T$ and $P$ and all negative pixels as $T^-$ and $P^-$ respectively. We then use the following standard image segmentation metrics to evaluate the performance:

- Intersection over union (IOU): \( IOU = \frac{|T \cap P|}{|T \cup P|} \in [0, 1] \)
- False-positive rate, relative to image size (FP Rate): \( FP = \frac{|T \cap P^{|}}{|P|} \in [0, 1] \)
- False-negative rate, relative to ground truth size (FN Rate): \( FN = \frac{|T \cap P^-|}{|T|} \in [0, 1] \)

VII. RESULTS

The averaged metrics for each network-environment pairing are shown in Table I. Table II shows p-values for a Welch’s t-test comparing IOUs. The top table compares the performance of other networks against our own (Syn+F*) on each data set, while the bottom table examines performance degradation for each network against the Main data set. (Colors: Green if IOU is significantly higher than baseline, red if significantly lower; p-values below 0.05, 0.01 and 0.001 highlighted)

### Table I: Averaged results for three metrics across all networks and data sets. Our network (Syn+F*) has the most consistent performance across all data sets for each metric.

| IOU       | Main | Afternoon | Cloudy | Envy | Lab |
|-----------|------|-----------|--------|------|-----|
| Syn+F*    | 62.3%| 53.3%     | 69.5%  | 53.7%| 60.1%|
| RGB+D     | 71.3%| 49.5%     | 49.7%  | 37.9%| 3.2% |
| RGB+F     | 74.8%| 66.6%     | 72.5%  | 52.1%| 33.3%|
| RGB       | 60.7%| 50.7%     | 57.9%  | 26.4%| 2.9% |
| Syn       | 54.5%| 40.4%     | 58.6%  | 45.7%| 18.7%|

| FN Rate   | Main | Afternoon | Cloudy | Envy | Lab |
|-----------|------|-----------|--------|------|-----|
| Syn+F*    | 28.7%| 33.2%     | 22.2%  | 30.7%| 26.1%|
| RGB+D     | 21.8%| 42.2%     | 42.2%  | 48.6%| 78.8%|
| RGB+F     | 17.7%| 23.8%     | 18.1%  | 33.0%| 53.1%|
| RGB       | 31.7%| 35.3%     | 29.9%  | 64.5%| 94.2%|
| Syn       | 30.8%| 43.4%     | 26.2%  | 37.8%| 58.6%|

### Table II: p-values for a Welch’s t-test comparing IOUs. The top table compares the performance of other networks against our own (Syn+F*) on each data set, while the bottom table examines performance degradation for each network against the Main data set. (Colors: Green if IOU is significantly higher than baseline, red if significantly lower; p-values below 0.05, 0.01 and 0.001 highlighted)

| Vs. Syn+F* | Main | Afternoon | Cloudy | Envy | Lab |
|------------|------|-----------|--------|------|-----|
| Syn+F*     | 0.004| 0.350     | 0.008  | 0.005| 0.000|
| RGB+D      | 0.000| 0.000     | 0.451  | 0.740| 0.000|
| RGB+F      | 0.746| 0.516     | 0.079  | 0.000| 0.000|
| RGB        | 0.009| 0.005     | 0.034  | 0.199| 0.000|
| Syn        | 1.000| 1.000     | 1.000  | 1.000| 1.000|

| Vs. Main   | Main | Afternoon | Cloudy | Envy | Lab |
|------------|------|-----------|--------|------|-----|
| Syn+F*     | 0.000| 0.000     | 0.000  | 0.000| 0.000|
| RGB+D      | 0.000| 0.000     | 0.000  | 0.000| 0.000|
| RGB+F      | 0.000| 0.000     | 0.000  | 0.000| 0.000|
| RGB        | 0.009| 0.000     | 0.000  | 0.000| 0.000|
| Syn        | 0.000| 0.000     | 0.000  | 0.000| 0.000|
the corresponding versions trained with depth or optical flow data, indicating that adding these features makes it significantly easier to discern the foreground from the background. This was consistent with the annotators’ general observations that without cross referencing the source video, trunks and thick branches in the background could be easily misinterpreted as being part of the foreground plane.

On the Main data set, the Syn+F* network was significantly outperformed by both the RGB+D and RGB+F networks. This is not wholly surprising due to overfitting effects. Due to the manual annotation, the RGB networks also learned to filter out non-branch objects in the foreground, usually ribbons hanging off of the trellis wires, that the Syn networks were not trained to filter out.

However, Table II shows that in almost all other scenarios no network was able to significantly outperform Syn+F*, the one exception to this being the RGB+F network on the Afternoon data set. In particular, the Lab environment proved to be extremely challenging for all networks besides the Syn+F* network. Overall, the RGB+F network proved to be the strongest competitor to Syn+F*, outperforming Syn+F* in the Main and Afternoon data sets, matching it in the Cloudy and Envy data sets, and having the second best (though still highly inaccurate) performance in the Lab setup. This provides support that optical flow is, itself, a powerful alternative to depth for foreground segmentation. However, the performance degradations for RGB+F network are more significant than the degradations for the Syn+F* network. This shows that the consistent performance of the Syn+F* network derives not just from its use of optical flow but also from environmental robustness gained from learning using synthetic data. Meanwhile, the RGB+D network showed very significant degradations in performance for every environment, indicating poor robustness to environments differing from the training set.

We acknowledge some shortcomings in our data collection process that could potentially have improved the results of the analysis. First, when obtaining the depth images, we did not align them with the color image frame, meaning that in order to make use of the depth data the pix2pix network would have to learn to do the depth/color alignment itself. Based on the performance of RGB+D versus RGB, the misaligned depth data still appears to be useful, but it is likely that proper depth alignment would have improved the RGB+D performance. However, we note that, from a qualitative examination of the depth images, they were quite noisy, contained many holes and shadowing, and often completely failed to capture thin structures, problems that optical flow had less frequently. Therefore, we do not believe that correcting the depth data would allow RGB+D to perform better than RGB+F in any scenario.

Also, we would have liked to obtain more training data from different days and conditions to examine the impact of adding more data on a trained network’s robustness, which we were unable to do due to time and labor constraints. However, we have already shown that training on RGB data almost always results in reduced robustness when moving to less familiar environments, and it is unlikely that any amount of data collected from outdoor orchards could have allowed an RGB-trained system to function in the Lab environment.

VIII. CONCLUSION

In this paper, we propose a system for performing branch segmentation in complicated orchard environments in which the foreground and background cannot be easily discerned from a single RGB image alone. In lieu of attempting to control the environment, which is not always feasible, or using depth data, which can be unreliable and requires hand-labeled data to train a neural network, we take advantage of our system design (i.e., an eye-in-hand configuration for robotic pruning). Specifically, we use optical flow in place of depth, allowing our system to be trained using only simulated
data with no manual labeling required and to function with any suitable RGB camera. We verify our system’s performance against various networks trained on manually-labeled data and prove that the joint use of synthetic data and optical flow data results in a highly robust system that performs well even in challenging, unforeseen environments.

ACKNOWLEDGMENT

The authors would like to acknowledge and thank Olsen Brothers Ranches, Inc. (Prosser, WA) for their support during data collection, and Gopala Krishna Josyula and Abhinav Jain for help with the data labeling process.

REFERENCES

[1] A. You, H. Kolano, N. Parayil, C. Grimm, and J. R. Davidson, “Precision fruit tree pruning using a learned hybrid vision/interaction controller,” IEEE International Conference on Robotics and Automation (ICRA), 2022, accepted.

[2] A. Tabb and H. Medeiros, “A robotic vision system to measure tree traits,” in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2017, pp. 6005–6012.

[3] B. Lavaquiol, R. Sanz, J. Llorens, J. Arnó, and A. Escolà, “A photogrammetry-based methodology to obtain accurate digital ground-truth of leafless fruit trees,” Computers and Electronics in Agriculture, vol. 191, p. 106553, 2021.

[4] T. Botterill, S. Paulin, R. Green, S. Williams, J. Lin, V. Saxton, S. Mills, X. Chen, and S. Corbett-Davies, “A robot system for pruning grape vines,” Journal of Field Robotics, vol. 34, no. 6, pp. 1100–1122, 2017.

[5] S. Amatya and M. Karkee, “Integration of visible branch sections and cherry clusters for detecting cherry tree branches in dense foliage canopies,” Biosystems Engineering, vol. 149, pp. 72–81, 2016.

[6] Z. Song, Z. Zhou, W. Wang, F. Gao, L. Fu, R. Li, and Y. Cui, “Canopy segmentation and wire reconstruction for kiwifruit robotic harvesting,” Computers and Electronics in Agriculture, vol. 181, p. 105933, 2021.

[7] Y. Majeed, J. Zhang, X. Zhang, L. Fu, M. Karkee, Q. Zhang, and M. D. Whiting, “Deep learning based segmentation for automated training of apple trees on trellis wires,” Computers and Electronics in Agriculture, vol. 170, p. 105277, 2020.

[8] X. Zhang, M. Karkee, Q. Zhang, and M. D. Whiting, “Computer vision-based tree trunk and branch identification and shaking points detection in dense-foliage canopy for automated harvesting of apples,” Journal of Field Robotics, vol. 38, no. 3, pp. 476–493, 2021.

[9] J. Zhang, L. He, M. Karkee, Q. Zhang, X. Zhang, and Z. Gao, “Branch detection for apple trees trained in fruiting wall architecture using depth features and regions-convolutional neural network (r-cnn),” Computers and Electronics in Agriculture, vol. 155, pp. 386–393, 2018.

[10] Z. Chen, D. Ting, R. Newbury, and C. Chen, “Semantic segmentation for partially occluded apple trees based on deep learning,” Computers and Electronics in Agriculture, vol. 181, p. 105952, 2021.

[11] C. Yang, L. Xiong, Z. Wang, Y. Wang, G. Shi, T. Kuremot, W. Zhao, and Y. Yang, “Integrated detection of citrus fruits and branches using a convolutional neural network,” Computers and Electronics in Agriculture, vol. 174, p. 105469, 2020.

[12] B. García-García, T. Bouwmans, and A. J. R. Silva, “Background subtraction in real applications: Challenges, current models and future directions,” Computer Science Review, vol. 35, p. 100204, 2020.

[13] M. Piccardi, “Background subtraction techniques: a review,” in IEEE International Conference on Systems, Man, and Cybernetics, vol. 4, 2004, pp. 3099–3104.

[14] T. Bouwmans, S. Javed, M. Sultana, and S. K. Jung, “Deep neural network concepts for background subtraction: A systematic review and comparative evaluation,” Neural Networks, vol. 117, pp. 8–66, 2019.

[15] R. Yao, G. Lin, S. Xia, J. Zhao, and Y. Zhou, “Video object segmentation and tracking: A survey,” ACM Transactions on Intelligent Systems and Technology (TIST), vol. 11, no. 4, pp. 1–47, 2020.

[16] J. Cheng, Y.-H. Tsai, S. Wang, and M.-H. Yang, “Segflow: Joint learning for video object segmentation and optical flow,” in IEEE International Conference on Computer Vision, 2017, pp. 686–695.

[17] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” in IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1125–1134.

[18] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” Advances in Neural Information Processing Systems, vol. 27, 2014.

[19] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, “Flownet 2.0: Evolution of optical flow estimation with deep networks,” in IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2462–2470.

[20] S. James, P. Wohlhart, M. Kalakrishnan, D. Kalashnikov, A. Irpan, J. Ibarz, S. Levine, R. Hadsell, and K. Bousmalis, “Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks,” in IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 12 627–12 637.

[21] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M.Malloci, A. Kolesnikov, et al., “The open images dataset v4,” International Journal of Computer Vision, vol. 128, no. 7, pp. 1956–1981, 2020.