APPLE MOTS: Detection, Segmentation and Tracking of Homogeneous Objects using MOTS*

Stefan de Jong[1], Hilmy Baja[1,2], Karsjen Tamminga[1] and João Valente[1]

***Abstract*—Current multi object tracking and segmentation (MOTS) methods made great progress for the simultaneous detection and tracking of *heterogeneous* objects like cars and pedestrians. Nevertheless, all of these scenes consisted of dissimilar objects, which are easier to track than homogeneous and smaller objects, as those are more similar in appearance. Therefore, this is the first paper that explores the implementation of MOTS algorithms for the simultaneous detection and tracking of *homogeneous* objects. Towards this end, video data was acquired in an apple orchard using a wearable camera and unmanned aerial vehicles (UAV). The dataset, called APPLE MOTS, contains almost 86000 manually annotated apple masks and is the first public dataset in which apple instances are temporally consistent labelled across frames. Implementation of the MOTS architectures called TrackR-CNN and PointTrack indicates that they could be suitable for the joint detection (MOTSP: 80.4) and tracking (sMOTSA: 38.7, MOTSA: 52.9) of apples. This paper exposes the challenge of tracking homogeneous objects due to their similar shape and colour while detection performance remains state-of-the-art. The APPLE MOTS code *(https://git.wur.nl/said-lab/rt-obj-tracking)* and datasets *(https://doi.org/10.5281/zenodo.5939726)* have been released to support the scientific community.**

***Index Terms*—Object Detection, Segmentation and Categorization; Agricultural Automation; Robotics and Automation in Agriculture and Forestry; Deep Learning Methods**

I. INTRODUCTION

In recent years, major improvements have been made in the field of deep learning and in particular object detection. Therefore, there are many recent deep learning breakthroughs that have become current state-of-the-art deep learning frameworks, e.g. MaskR-CNN [1] which is an extension of Faster R-CNN [2], and You Only Look Once (YOLO) [3] are popular methods. Nevertheless, tracking of objects remains a difficult task, especially for segmentation purposes. Recent work in the field of multi object tracking and segmentation (MOTS) [4]–[6] made great progress on this topic by simultaneously detecting and tracking *heterogeneous* objects on a pixel level. Their research focused on the application of MOTS on temporally consistent scenes for cars and pedestrians. As an addition, the scenes were crowded and challenging due to the many occlusions. Nevertheless, all of these scenes consisted of dissimilar objects (heterogeneous) which are easier to track than *homogeneous* objects, as those are more similar in appearance in terms of shape and colour (Figure 1).

(a) Homogeneous sample image (b) Homogeneous annotated image

(c) Heterogeneous sample image (d) Heterogeneous annotated image

Fig. 1: Sample image of APPLE MOTS (a) with corresponding instance annotation (b) and KITTI MOTS sample image (c) with corresponding instance annotation (d).

So far, no previous deep learning-based studies were done for applying MOTS on homogeneous scenes, as difficulty increases in terms of detection and tracking. Therefore, there is a need to explore the ability to detect and track instance segmentation masks for homogeneous object scenes. Typical homogeneous object scenes can be found in agriculture, especially in the field of orchard management, consisting of large orchards with homogeneous fruits (e.g. low-hanging fruits like pears, apples, mangoes, and berries). Specifically, apples are one of the most widely farmed fruits in the world, taking over five million hectares of planting space in the world with a yield around 83 million tonnes in 2017 [7]. In apple orchards, yield is commonly estimated in the flowering stage or the mature stage, with the former by counting the flowers, and the latter by counting the apples [8]. This technique is mostly done manually. However, manual yield estimation is labour-intensive and inaccurate. The techniques often rely on extrapolating data from a select number of trees, which

Manuscript received: March, 15, 2022; Revised June, 16, 2022; Accepted July, 18, 2022.

This paper was recommended for publication by Editor Hyungpil Moon upon evaluation of the Associate Editor and Reviewers' comments.

*This work was not supported by any organization.

[1]The authors are with the Information Technology Group, Wageningen University & Research, 6706 KN Wageningen, The Netherlands {stefan.dejong, hilmy.baja, karsjen.tamminga, joao.valente}@wur.nl

[2]The author is with Center of Technology, Universitas Hasanuddin, Makassar, Indonesia.

Digital Object Identifier (DOI): see top of this page.

might not be an accurate representation for the whole orchard. Accurate yield estimation is beneficial to improve fruit quality, making better decisions on harvesting, calculating required labour for efficiency and reduce operating costs by optimizing storage facilities and packaging logistics [9].

A few challenging problems which arises in yield estimation are the difficulties (1) to detect low-hanging fruits due to complex and occluded fruits, (2) to track fruits in order to avoid double counting and (3) to perform on site decision-making. The framework of MOTS can be used to solve these problems, as it provides both instance segmentation with tracking. Next to that, MOTS methods show the possibility of applying near real-time image processing and counting of apples. Common issues like disk space, processing time and uploading time will be potentially resolved with near real-time processing and therefore enable rapid decision-making on site. A speed evaluation is also added in the results to show the feasibility of applying these algorithms for near real-time processing. Overall, this paper explores the possibility of using MOTS for homogeneous objects like apples in order to obtain improvements in future yield estimation of low-hanging fruits.

For heterogeneous scenes, benchmark datasets like KITTI MOTS [4] and Apollo MOTS [10] were constructed to facilitate the research of multi object tracking and segmentation. Moreover, efforts were made to construct benchmark datasets for homogeneous objects like apples and mangoes [11], [12]. Nevertheless, these datasets do only contain bounding boxes or segmentation masks without having temporal consistent information with them. Hence, there is also a lack of suitable datasets in order to perform multi object tracking and segmentation on homogeneous scenes. To summarize, this paper builds a suitable dataset and subsequently provides results on the first attempt to use MOTS for the simultaneous detection, segmentation and tracking of homogeneous objects like apples. Therefore, we make the following **contributions**:

- We construct and provide APPLE MOTS, the first public dataset in which apple instances are labelled across frames with temporal consistency. This dataset contains almost 86000 manually annotated apple masks which is 32% larger than the Apollo MOTS and KITTI MOTS datasets;
- We explore the possibilities to apply MOTS on homogeneous scenes, in specific low-hanging fruits (apples);
- We propose new tracking branches (Kalman filter and Optical Flow) to existing MOTS models to validate its performance on tracking apples.

II. RELATED WORK

Homogeneous object detection. For homogeneous object scenes (and especially fruits) multiple researches and datasets were constructed with the objective of detecting fruits on the canopy of trees, for example apples [13]–[15], citrus [16], strawberries [17], mangoes [18] and blueberries [19]. The MangoYOLO [18] dataset tests detection on several object detection architectures such as YOLOv3 [3] and MaskR-CNN [1] with their own re-design of YOLO.

The fruit datasets mentioned above cannot be used within MOTS as those datasets consist of random images which lack temporal information. Moreover, solely using object detection (without tracking) will cause many repetitions, which will lead to non-precise yield information for the farmer.

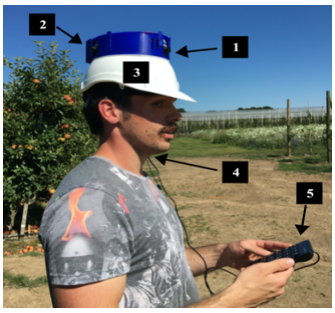
Heterogeneous MOT. For the task of multi-object tracking (MOT) with object detection, the research community has shown a very large interest in tracking heterogeneous objects like cars and pedestrians in urban settings, due to research of autonomous driving technology. Numerous MOT algorithms have been developed utilizing various methods to improve tracking and detection of these objects [20]–[22], which are benchmarked on various datasets. DeepSORT [20] is a MOT tracker that makes use of FasterRCNN [2] to generate detections. The datasets that are commonly benchmarked on new MOT algorithms are, among others, the KITTI [23], MOTChallenge [24], ApolloScape [25], and UA-DETRAC [26] dataset. These datasets are annotated with bounding boxes, so they lacks per-pixel accuracy provided by instance segmentation tasks, causing lower performance in localization of fruits.

Heterogeneous MOTS. As a solution to previously mentioned problems with detection and tracking methods, recent developments in computer vision like MOTS attempt to track objects across frames by having an end-to-end trainable network sharing useful information over the whole pipeline. Currently, this is only tested for pedestrians and cars. The first proposed method that addresses the aspects of the MOTS task is called *TrackR-CNN* [4]. TrackR-CNN consists of two steps, object detection, fulfilled by the MaskR-CNN algorithm, and tracking fulfilled by a tracking mechanism. 3D convolutions were added to the feature maps of the ResNet-101 backbone. The extra dimension, time, was included with the features in order to augment them with temporal context. The second proposed method is *PointTrack* [5]. Its object detection is fulfilled by SpatialEmbedding [27], and tracking by association of instance embeddings. Currently, there are multiple MOTS datasets that are used to benchmark MOTS algorithms. Most of these datasets are extensions of MOT datasets, which include among others, KITTI MOTS and MOTSChallenge [4], Apollo MOTS [10], and CityScapes [28]. These datasets emphasize tracking and detection of heterogeneous objects in urban scenes. Thus far, there is a lack of a MOTS dataset that is suitable in homogeneous scenes (like agriculture).

III. METHOD

A. Data acquisition

Data acquisition was conducted in an apple orchard of Wageningen Plant Research for Flower bulbs, Nursery stock and Fruits in Randwijk (Netherlands, 51°56'19.3"N 5°42'24.4"E). Three different varieties of apples were grown, *Elstar*, *Jonagold* and *Junami*. By using a novel wearable sensor platform, DJI Matrice 210 RTK V2 (UAV) and Parrot Anafi (UAV) (see Figure 2) videos of the orchard plots were acquired. All systems were able to acquire videos at a frame rate between 25-30 FPS. Different acquisition strategies were applied by acquiring videos at a height of 2-3m within or next to the apple rows. The field work was done on four different days, spread over 1.5 month, to acquire video under different light conditions.



(a) Wearable Sensor Platform. The system components of the wearable sensor platform (a) are: 1) front camera, 2) side cameras, 3) customized helmet, 4) headphone to receive sound signals and 5) numpad to send commands or enter data.

(b) DJI Matrice 210 RTK V2

(c) Parrot Anafi

Fig. 2: Overview of the platforms used during our research.

Fig. 3: An image acquired from the Parrot Anafi UAV with an added gray overlay, the so-called *ignore region*. The ignore region is added on top of visible background apples that may be detected as false positives.

B. Dataset construction

Due to the fact that MOTS were not studied before within orchard research, there was a lack of suitable datasets where apple instances were labelled across frames. The annotation tool CVAT was used for the annotation of apples across the frames, having a resolution of 1296x972. CVAT is developed by Intel and has powerful features like interpolation and tracking of objects between frames to decrease the annotation time. The same annotation procedure that was used for the MinneApple dataset [11] was applied. This procedure comprises the labelling of both fully and partially visible fruits in the trees, while ignoring fruits in background trees or on the ground. In total, nine datasets were annotated in 5-6 weeks. Six datasets were used for training and three for testing. This lead to an image data split of 70% for training and 30% for testing. Two train datasets were acquired by the wearable sensor and four by the Parrot Anafi. Moreover, two test datasets were acquired by both UAVs and one was acquired using the wearable sensor platform. Our dataset called *APPLE MOTS* has a substantial dimension of almost 86000 manual annotated masks (see Figure 1 for sample image). The masks are divided over 1673 frames and consist of 2304 unique apple instances. In accordance with the KITTI dataset [23], the so-called *ignore regions* are added to the APPLE MOTS test datasets to make sure that unlabelled apples in the background do not get detected as false positives. An example is shown in Figure 3. The extent of this dataset makes this dataset suitable for current computer vision tasks. As an addition, it is currently one of the largest publicly available orchard datasets.

C. TrackR-CNN implementation

One of the MOTS architectures used in this research was TrackR-CNN, which consisted of three core features: training (with addition of extra tracking branches), forwarding and

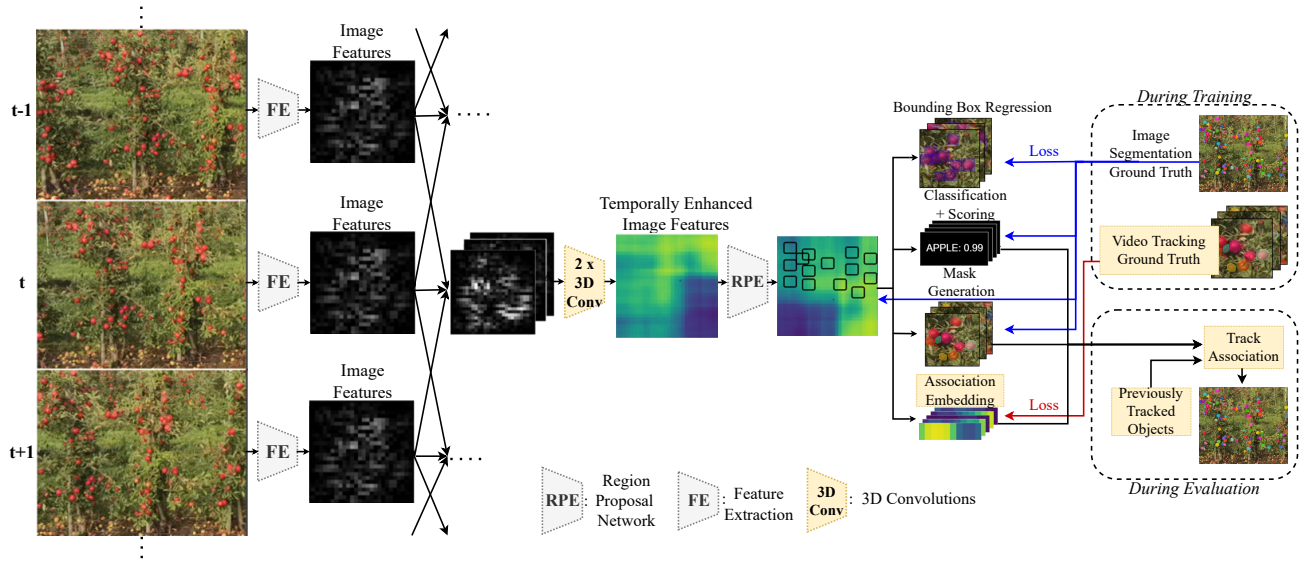


Fig. 4: TrackR-CNN architecture. Differences to Mask R-CNN architecture are highlighted in yellow.

tuning. An overview of TrackR-CNN's architecture can be found in Figure 4.

Training. The best performing model from the research of Voigtlaender et al. [4] is used as a starting point. This baseline model used two stacked 3D convolutions as its temporal component and data association with learned embeddings. As an addition, the MaskR-CNN of the TrackR-CNN algorithm used the pre-trained weights of COCO and Mapillary. This method, called transfer learning, migrated the knowledge learned from the COCO and Mapillary dataset to the target dataset. Likewise, the association head was fine-tuned by using the initialized weights of the KITTI MOTS dataset. The model was trained using an Adam optimizer with a learning rate of $5x10^{-7}$. A batch size of 8 and 40 epochs was used as a starting point for our dataset according to Voigtlaender et al. [4]. Different batch size and epochs were applied to find the optimal value for the dataset in order to avoid overfitting.

Kalman Filter. To improve the tracking capabilities of the algorithm, a linear velocity Kalman filter [29] was implemented, which replaces the association head for the tracking step. The Kalman filter is invoked on the bounding box tracking mechanism using Euclidean distance, due to the inability of mask prediction using the Kalman filter. The estimation model is shown as follows:

$$x = [x_1, x_2, y_1, y_2] \tag{1}$$

where x_1 and x_2 is the coordinate for the top left point of the bounding box, and y_1 and y_2 is the bottom right coordinate of the bounding box. The bounding boxes are then associated to a target instance mask, which will be updated based on subsequent detections in the sequence. If there are no detections associated to a state, then the next position is predicted using the linear velocity model.

Optical Flow. Mask warping is a promising approach to associate masks over the sequence. Mask propagation scores are calculated with images of flow estimations. In accordance to the methods of [4], the optical flow estimation of all pairs of adjacent frames are calculated with the model of PWC-Net [30].

Forwarding. After training, the training dataset was forwarded and tracked. This means that the network was evaluated on the given training datasets and created a tracking output with all images and their corresponding detections.

Tuning. In order to increase the detection and tracking performance of the model, the parameters were fine-tuned by making use of a random search of 1000 iterations. Subsequently, the best detection and tracking parameters were defined for the training dataset and evaluated (using the evaluation metrics) on the test dataset.

D. PointTrack Implementation

The other MOTS architecture used in this research was PointTrack [5]. The implementation consists of two steps: 1) instance segmentation model training, and 2) PointTrack training. An overview of PointTrack's architecture is shown in Figure 5. How the implementation differs from TrackR-CNN is depicted in the flowchart, shown in Figure 6.

Instance segmentation training. Before training the instance segmentation model, spatial embedding [27] mask crops are generated. It is done by selecting masks that are considered unique, then subsequently parsing them as an image and instance PNG. Out of the 62899 apple masks in the train datasets, there are a total of 26631 generated mask crops. Considering one apple is around 30-50 pixels wide, each mask crop has a size of 80x80 pixels to fully encapsulate the apple and its surroundings. Two models were trained on the segmentation network, a model pre-trained on the weights of KITTI MOTS (transfer learning) and a model trained from scratch. The network was trained with a learning rate of $5x10^{-5}$ with an Adam optimizer. It was trained with a batch of 20 mask crops and 400 epochs. Consequently, the trained

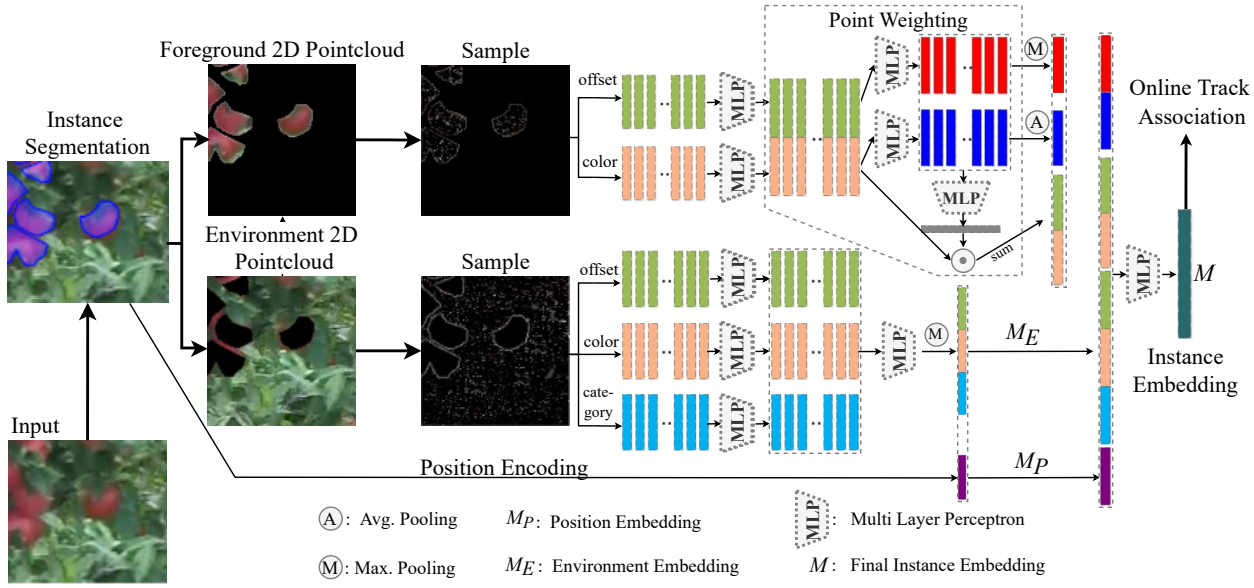


Fig. 5: PointTrack architecture.

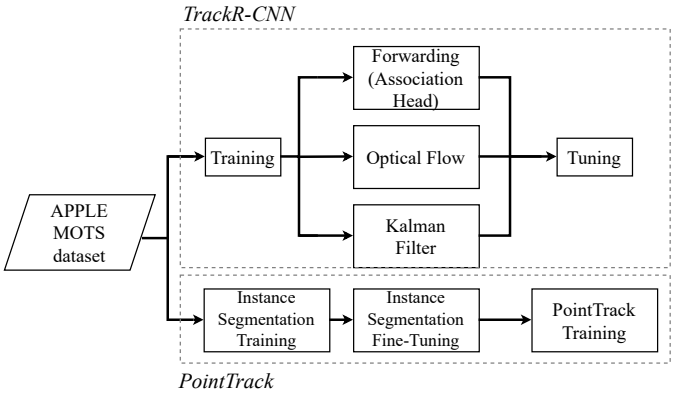


Fig. 6: Flowchart showing the implementation of TrackR-CNN (along with the Optical Flow and Kalman Filter method) and PointTrack.

network is fine-tuned with weights from larger mask crops (160x160) to let the model learn the surrounding features of the masks. The fine-tuning trained with a learning rate of $5\text{x}10^{-6}$ and 1200 epochs.

PointTrack training. In this step, the PointTrack network will learn the embeddings (critical points) obtained from the instance segmentations and assign association weights to help in tracking and reducing ID switches (IDS). The PointTrack training is similar to the instance segmentation training. Pre-trained weights of KITTI MOTS are used with a training of 100 epochs and a batch size of 64.

E. Evaluation

To assess the performance of the algorithms, different evaluation metrics were implemented. Three evaluation measures were used, the multi-object tracking and segmentation accuracy (MOTSA), the multi-object tracking and segmentation precision (MOTSP) and the soft multi-object tracking and segmentation accuracy (sMOTSA). They were calculated as follows:

$$MOTSA = \frac{|TP| - |FP| - |IDS|}{|M|} \tag{2}$$

$$MOTSP = \frac{\tilde{TP}}{|TP|} \tag{3}$$

$$sMOTSA = \frac{\tilde{TP} - |FP| - |IDS|}{|M|} \tag{4}$$

where:

- TP True positives, number of hypothesized masks mapped to a ground truth mask (where IOU>0.5).
- $\tilde{\text{TP}}$ Soft true positives, sum of the IOU of all true positives.
- FP False positives, number of hypothesized masks that are not mapped to any ground truth mask.
- IDS ID switches, ground truth mask whose former ground truth mask (t-1) was tracked with a different id.
- M The number of ground truth masks.

IV. RESULTS

A. Detection

Table I shows the performance of the model of our test set of APPLE MOTS. The highest metrics are highlighted in bold. For the task of detection, the MOTSP does not show much variation between all the methods. The model with the highest MOTSP is PointTrack trained with KITTI MOTS weights.

B. Tracking

The results show that the model with the best tracking metrics (sMOTSA and MOTSA) is the PointTrack model, outperforming the the TrackR-CNN models by 3-5%. On the other hand, the ID switches of PointTrack outperformed TrackR-CNN by almost three times. The tracking metrics for the Kalman filter and mask warping were the worst performing ones.

C. Speed

The speed column in Table I shows the time it takes to process the detection, segmentation and tracking of each frame. The speed results show a significant difference between the two algorithms used. The models of TrackR-CNN and PointTrack have a difference of a factor of 40, with the latter being considerably faster due to the usage of SpatialEmbedding [27] for instance segmentation.

D. Discussion, limitations and outlook

TrackR-CNN models results. Table I shows the best performing TrackR-CNN model. In the case of apples, the bounding box tracking method managed to outperform the association head tracking. These results differ from those of Voigtlaender et al. [4]. In their research, the bounding box tracking method gave lower sMOTSA and MOTSA values (2-4%). This difference might be attributed to the different object type tracked. APPLE MOTS is a dataset with homogeneous objects, compared to KITTI MOTS (cars, pedestrians) a dataset with heterogeneous objects, as such is much easier used for re-identification. Objectively, there are many factors that differentiates KITTI MOTS and APPLE MOTS, which does not make this a straightforward comparison. However, it is a comparison that still needs to be made due to the lack of research on homogeneous MOTS objects. On the other hand, the tracking results of the Kalman filter and mask warping on TrackR-CNN are inaccurate. The Kalman filter had such results due to the bounding box predictions not being able to properly match the assigned apple mask instances, resulting in low IoU scores, hence low tracking scores. Further, the mask warping calculations from the APPLE MOTS dataset using PWC-Net [30] returned very inaccurate optical flow images, due to individual apples not being detected in the flow estimation. The leaves and apple occlusions hindered the flow estimation to detect the apple contours and motion boundaries, hence resulting in inaccurate tracking metrics.

PointTrack models results. Table I shows that both of the PointTrack models achieved slightly lower tracking metrics than the TrackR-CNN model, 3% to 6% lower, which is

(a) PointTrack detecting unlabeled background apples

(b) Ignore regions that censor the unlabeled apples

Fig. 7: Image showing how PointTrack is able to detect background apples quite well, despite them not being labeled in the current iteration of the APPLE MOTS dataset.

unexpected considering PointTrack is the more state-of-the-art algorithm. However, PointTrack makes it up with slightly higher MOTSP and a much better IDS. Inspecting the results qualitatively, it is shown in Figure 7 that PointTrack is able to detect many small unlabeled apples in the background, which in turn is detrimental to the tracking performance. This potentially confirms that PointTrack is better than TrackR-CNN in detecting smaller objects. Comparing both PointTrack models, the model trained from scratch has higher tracking metrics (sMOTSA and MOTSA) by 4-5%.

Results of models with ignore region. With the purpose of eliminating the possibility of false positives in the detection of all models, ignore regions in the form of overlays were added to the background apples of the testing datasets of APPLE MOTS. This technique is also present in the latest iteration of the KITTI [23] dataset, to reduce false positive detections. Table I shows results of the models evaluated with these ignore regions. The tracking metrics of the PointTrack model improved by 6.9%, confirming our suspicion that PointTrack had lower results due to many false positive detections. TrackR-CNN's model also improved slightly in sMOTSA and MOTSP but not as much as PointTrack, which signifies that it didn't suffer that many false positive detections. Ultimately, the tracking metrics of PointTrack's model surpasses TrackR-CNN's model by 5.1%, while also achieving a 64.6% decrease in IDS.

ID assignment. A qualitative visual analysis was done on the detection results, it is found that both MOTS algorithms suffer from the algorithm reusing IDs throughout the sequence, as emphasized in Figure 8. Figure 8 (a) shows the same apple being assigned to ID number 29, 42 and 7 in consecutive frames, moreover, (d) shows another single apple assigned to two different ID numbers, 30 and 14. The models of both algorithms struggled to differentiate unique apples through the sequence, leading to extreme cases where one apple ID is assigned to 9 different apples. These high amount of ID switches (see Table I) resulted in a negative numerator for sMOTSA and MOTSA and therefore a negative metric.

Speed. Currently, TrackR-CNN has a processing speed of 0.2 FPS when forwarding the detections. Meanwhile, on the same machine, PointTrack demonstrates more efficient processing by attaining a speed of 7.8 FPS on average. A possible solution to speed up the process could be to

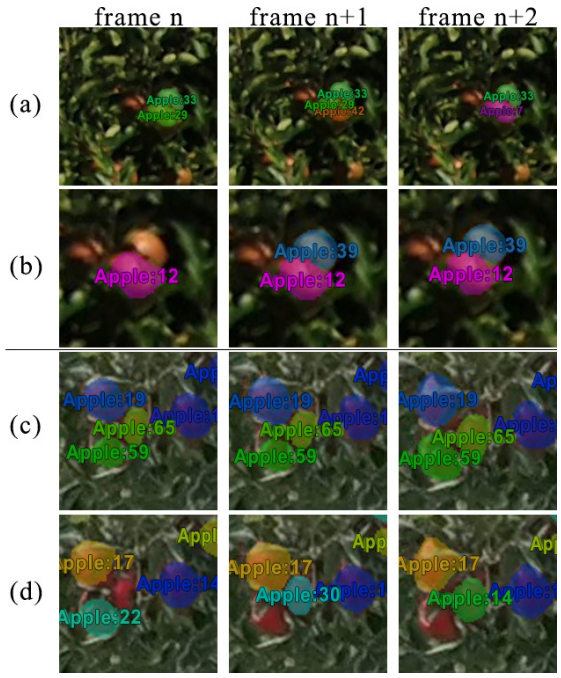


Fig. 8: Comparison of consecutive frames n of the model results on APPLE MOTS. (a) & (c) shows PointTrack results and (b) & (d) shows TrackR-CNN results on the test dataset.

Table I: Performance of the models on the test dataset. One PointTrack model is trained from KITTI MOTS weights (transfer learning) and the other model is trained from scratch.

Method	Speed (FPS)	sMOTSA	MOTSA	MOTSP	ID switches (IDS)
TrackR-CNN (Bounding box)	0.20	34.8	46.8	80.4	1427
TrackR-CNN + Kalman Filter	0.20	-9.7	1.4	79.3	11949
TrackR-CNN + Mask Warping and Optical Flow	0.21	-13.2	-2.9	80.7	11735
PointTrack (KITTI MOTS weights)	**7.81**	28.9	41.2	**81.7**	**349**
PointTrack (Scratch)	**7.81**	31.8	46.0	80.0	469
TrackR-CNN with ignore regions	0.20	35.7	47.8	80.4	1480
PointTrack with ignore regions	**7.81**	**38.7**	**52.9**	80.0	524

(a) Annotated ground truth (b) Result TrackR-CNN

Fig. 9: Image of the wearable sensor platform test set containing the ground truth (a) and TrackR-CNN output (b). The white rectangles highlight the missing predictions.

downsize the image. Nevertheless, this is at the expense of detection accuracy, as apples will also be downsized to a lower resolution.

Detection performance. Figure 9 shows one example frame from the wearable sensor platform. Most predicted masks correspond correctly to the ground truth. However, the white rectangles in the predicted output of TrackR-CNN highlight some areas where predicted masks are not matching the ground truth masks. The corresponding apples are occluded by leaves or branches and therefore difficult to detect for the algorithm. Nevertheless, the detection performance of all methods (MOTSP) shows comparable results with Voigtlaender et al. [4]. A slightly lower detection performance is achieved compared to the car class (85.1) and slightly better than the pedestrian class (75.6). Hence, it reflects the suitability of using both TrackR-CNN and PointTrack for the detection of homogeneous objects even though, the apple annotations had a diameter of only 30-50 pixels in a challenging scene (occlusions). Compared to the resolution of the image (1296x972) this is rather small.

Challenges in tracking homogeneous objects. The results from this letter show that tracking fruits using the MOTS current 2D state-of-the-art method, PointTrack, outperforms the baseline method of TrackR-CNN in terms of performance. Due to a lack of comparable datasets (homogeneous MOTS), the evaluation metrics can only be compared to existing MOTS datasets from previous literature (cars and pedestrians, compared to apples). Therefore, it might be argued that the lower tracking performance is related to the object class, especially due to the heterogeneity and larger objects of car and pedestrians compared to homogeneity and smaller apples. Therefore, further research into tracking homogeneous objects should be done. This impairs that currently little is known about implementation of MOTS in orchard research, as this is the first research that implements MOTS in that setting.

Limitations. A major limitation of the MOTS technique is that it only takes into account the detection and tracking of apples from one side. One should be aware that double counting of apples can still appear if apples are visible from opposite sites. In the long term it is therefore of interest to not only detect and track the apples accurately but also to compare this information with the actual amount of apples within the tree. A prediction model would then be helpful to translate the detected amount of apples to the total number of apples within an orchard [31].

V. CONCLUSION

This research constructed and provided APPLE MOTS, with almost 86000 manual annotated masks, the first public dataset where apple instances are labelled across frames. State-of-the-art algorithms (MOTS) were implemented and tested on the APPLE MOTS dataset to validate the feasibility of detection and tracking of homogeneous objects in agriculture (apples). The task of tracking homogeneous objects is difficult, despite pixel-wise segmentation of the objects. It can be argued that the small size of the apples (∼30-50 pixels wide per segmentation) and the homogeneous nature of the objects make the tracking way more challenging than larger heterogeneous objects like cars and pedestrians. Nevertheless, with evaluation results of a baseline and a state-of-the-art MOTS algorithm, we present a new powerful tool for the detection and tracking of fruits. By making the data and code used in this research publicly available, it can serve as a foundation for further research regarding the implementation of MOTS in agricultural scenes (dataset: *https://doi.org/10.5281/zenodo.5939726*, code: *https://git.wur.nl/said-lab/rt-obj-tracking*).

VI. ACKNOWLEDGEMENTS

The authors would like to thank Chenglong Zhang and Pieter van Dalfsen for help in collecting ground truth data.

REFERENCES

[1] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask r-cnn," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2017-Octob, pp. 2980–2988, 2017.

[2] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," 2015, pp. 91–99.

[3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem, pp. 779–788, 2016.

[4] P. Voigtlaender, M. Krause, A. Osep, J. Luiten, B. B. G. Sekar, A. Geiger, and B. Leibe, "Mots: Multi-object tracking and segmentation," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, pp. 7934–7943, 2019.

[5] Z. Xu, W. Zhang, X. Tan, W. Yang, X. Su, Y. Yuan, H. Zhang, S. Wen, E. Ding, and L. Huang, "Pointtrack++ for effective online multi-object tracking and segmentation," *arXiv*, pp. 2–5, 2020.

[6] S. Qiao, Y. Zhu, H. Adam, A. Yuille, and L.-C. Chen, "Vip-deeplab: Learning visual perception with depth-aware video panoptic segmentation," 2020.

[7] Y. Wang, W. Li, X. Xu, C. Qiu, T. Wu, Q. Wei, F. Ma, and Z. Han, "Progress of apple rootstock breeding and its use," *Horticultural Plant Journal*, vol. 5, pp. 183–191, 2019.

[8] C. Zhang, J. Valente, L. Kooistra, L. Guo, and W. Wang, "Orchard management with small unmanned aerial vehicles: A survey of sensing and analysis approaches," *Precision Agriculture*, vol. 22, no. 6, pp. 2007–2052, 2021.

[9] Q. Wang, S. Nuske, M. Bergerman, and S. Singh, "Automated crop yield estimation for apple orchards," pp. 745–758, 2013.

[10] Z. Xu, W. Zhang, X. Tan, W. Yang, H. Huang, S. Wen, E. Ding, and L. Huang, "Segment as points for efficient online multi-object tracking and segmentation," *European Conference on Computer Vision*, pp. 264–281, 2020.

[11] N. Hani, P. Roy, and V. Isler, "Minneapple: A benchmark dataset for apple detection and segmentation," *IEEE Robotics and Automation Letters*, vol. 5, pp. 852–858, 2020.

[12] Z. Wang, K. Walsh, and A. Koirala, "Mango fruit load estimation using a video based mangoyolo—kalman filter—hungarian algorithm method," *Sensors (Switzerland)*, vol. 19, 2019.

[13] N. Häni, P. Roy, and V. Isler, "A comparative study of fruit detection and counting methods for yield mapping in apple orchards," *Journal of Field Robotics*, vol. 37, pp. 263–282, 2020.

[14] S. Bargoti and J. P. Underwood, "Image segmentation for fruit detection and yield estimation in apple orchards," *Journal of Field Robotics*, vol. 34, pp. 1039–1060, 2017.

[15] J. Gené-Mola, R. Sanz-Cortiella, J. R. Rosell-Polo, J. R. Morros, J. Ruiz-Hidalgo, V. Vilaplana, and E. Gregorio, "Fruit detection and 3d location using instance segmentation neural networks and structure-from-motion photogrammetry," *Computers and Electronics in Agriculture*, vol. 169, p. 105165, 2020.

[16] X. Liu, S. W. Chen, C. Liu, S. S. Shivakumar, J. Das, C. J. Taylor, J. Underwood, and V. Kumar, "Monocular camera based fruit counting and mapping with semantic data association," *IEEE Robotics and Automation Letters*, vol. 4, pp. 2296–2303, 2019.

[17] T. T. Santos, L. L. de Souza, A. A. dos Santos, and S. Avila, "Grape detection, segmentation, and tracking using deep neural networks and three-dimensional association," *Computers and Electronics in Agriculture*, vol. 170, 2020.

[18] A. Koirala, K. Walsh, Z. Wang, and C. McCarthy, "Deep learning for real-time fruit detection and orchard fruit load estimation: Benchmarking of 'mangoyolo'," *Precision Agriculture*, vol. 20, no. 6, pp. 1107–1135, 2019.

[19] S. Gonzalez, C. Arellano, and J. E. Tapia, "Deepblueberry: Quantification of blueberries in the wild using instance segmentation," *IEEE Access*, vol. 7, pp. 105 776–105 788, 2019.

[20] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *2017 IEEE international conference on image processing (ICIP)*. IEEE, 2017, pp. 3645–3649.

[21] E. Yu, Z. Li, S. Han, and H. Wang, "Relationtrack: Relation-aware multiple object tracking with decoupled representation," *arXiv preprint arXiv:2105.04322*, 2021.

[22] Y. Xu, Y. Ban, G. Delorme, C. Gan, D. Rus, and X. Alameda-Pineda, "Transcenter: Transformers with dense queries for multiple-object tracking," *arXiv preprint arXiv:2103.15145*, 2021.

[23] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[24] P. Dendorfer, A. Osep, A. Milan, K. Schindler, D. Cremers, I. Reid, S. Roth, and L. Leal-Taixé, "Motchallenge: A benchmark for single-camera multiple target tracking," *International Journal of Computer Vision*, vol. 129, no. 4, pp. 845–881, 2021.

[25] X. Huang, X. Cheng, Q. Geng, B. Cao, D. Zhou, P. Wang, Y. Lin, and R. Yang, "The apolloscape dataset for autonomous driving," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 954–960.

[26] L. Wen, D. Du, Z. Cai, Z. Lei, M.-C. Chang, H. Qi, J. Lim, M.-H. Yang, and S. Lyu, "Ua-detrac: A new benchmark and protocol for multi-object detection and tracking," *Computer Vision and Image Understanding*, vol. 193, p. 102907, 2020.

[27] D. Neven, B. de Brabandere, M. Proesmans, and L. van Gool, "Instance segmentation by jointly optimizing spatial embeddings and clustering bandwidth," *arXiv*, 2019.

[28] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3213–3223.

[29] R. E. Kalman, "A new approach to linear filtering and prediction problems," 1960.

[30] D. Sun, X. Yang, M. Y. Liu, and J. Kautz, "Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. D, pp. 8934–8943, 2018.

[31] J. Das, G. Cross, C. Qu, A. Makineni, P. Tokekar, Y. Mulgaonkar, and V. Kumar, "Devices, systems, and methods for automated monitoring enabling precision agriculture," *IEEE International Conference on Automation Science and Engineering*, vol. 2015-Octob, pp. 462–469, 2015.