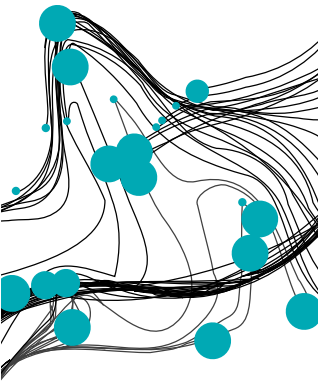
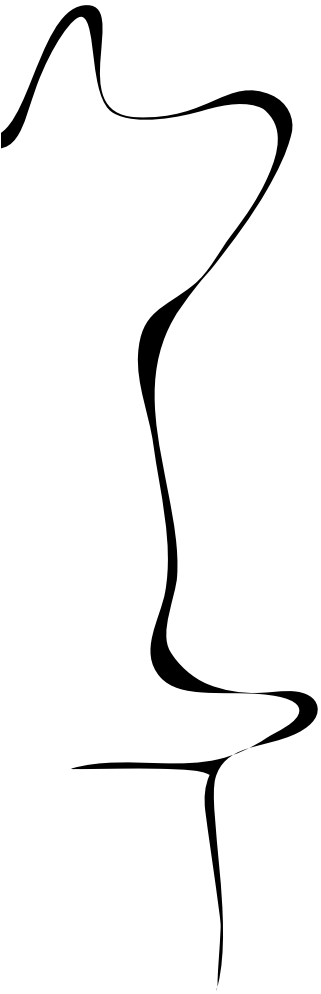


DMB

DATABASE MANAGEMENT
AND
BIOMETRICS

.87885



FRUIT INSPECTION PROGRESS TRACKING USING STEM AND CALYX DETECTION AND 3D SPHEROID MODELS

Ellen den Boer

MASTER'S THESIS ASSIGNMENT

Committee:
Dr.ir. L.J. Spreeuwers
Dr. C.G. Zeinstra
Dr.ir. M. Abayazid
Msc. L. van de Laak

February, 2023

2023DMB0003
Data Management and Biometrics
EEMathCS
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

UNIVERSITY OF TWENTE. | DIGITAL SOCIETY INSTITUTE

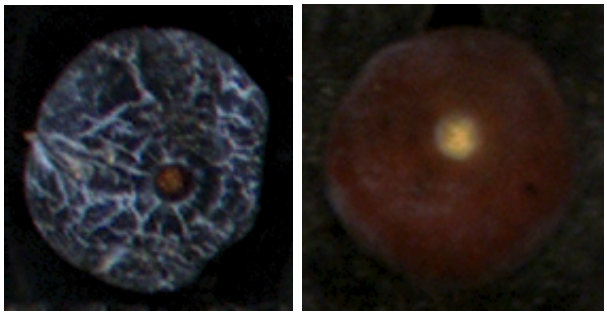
Fruit Inspection Progress Tracking using Stem and Calyx Detection and 3D Spheroid Models

ELLEN DEN BOER

***Abstract*—On an industrial scale fruits are sorted based on their quality, which is often automated using roller conveyors and multiple cameras. These cameras capture consecutive images of the fruits in order to assess their quality. For proper quality control it is of importance to know which part of the fruit has been inspected. In this paper a new approach for fruit inspection tracking is presented, making use of feature detection and a spherical or spheroid 3D model. From the captured image series, the size of the fruit is determined and a 3D spheroid model is fitted. The most distinctive features of fruit, namely the stem and calyx are detected using the well-known YOLOv5 detection network. Based on YOLOv5 nano, a model is trained with a mAP of 0.86 that generalises over a variety of fruits. Using the stem and calyx, the rotation matrix between two consecutive images is calculated, which is used to determine the overall inspection progress. Since the stem and calyx detection yields a maximum of two matching points, the detection algorithm LoFTR is implemented in order to determine whether more matching points lead to a better estimation of the inspected area. The obtained results demonstrate that the overall inspection progress can be tracked using stem and calyx detection or matching points. Results of the progress tracking algorithm based on matching points are similar to the results of the progress tracking algorithm using stem and calyx detection, whilst taking more computational time.**

I. INTRODUCTION

In 2020, 887 million tons of fruit have been harvested worldwide [1]. After harvesting, the fruits need to be sorted for quality assurance. Not only will a degraded fruit quality lead to lower prices, fruits with rotten spots or mold can also impact the quality of an entire batch during storage or transportation [2]. It is therefore vital that the fruit quality is assessed correctly and that spoiled fruits are detected timely. Fruit inspection and selection can be automated using cameras positioned above a roller conveyor, capturing multiple images of a piece of fruit over time. However, it still remains a challenge to determine whether the entire fruit has been inspected or whether certain areas have been inspected multiple times. The resulting miscounting or missing of defects might lead to a misclassification of the fruit quality [3].

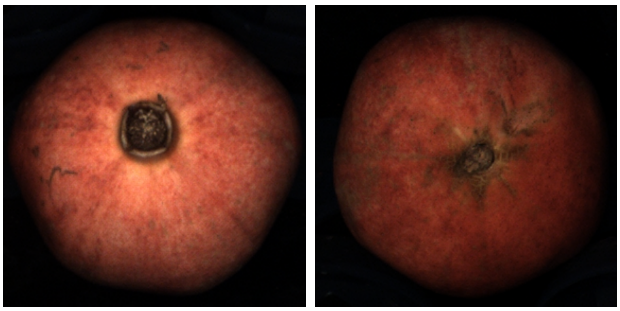


(a) Rich texture. (b) Limited texture.

Fig. 1: A blueberry with rich and limited texture.

When different images over time of the same fruit are captured, motion tracking can be used in order to determine the rotation between consecutive images. The rotation can be used to determine if the complete area is seen. However, conventional approaches that base the motion tracking on the fruit surface texture [4] cannot be generally applied, due to the lack of texture on the skin area of some fruit commodities, as shown in Figure 1b. So other features of fruit are needed in order to determine the rotation. Two distinctive features that appear in all types of fruit are the stem and calyx, as shown in Figure 2, examples for different types of fruits are shown in Appendix A. Since these are positioned on opposite ends of the fruit, they are almost always present in the captured images [5]. Besides being the most distinctive feature of a fruit, the stem and calyx are also important in classifying fruits, since defects often get confused with the stem and calyx. As a result of these properties, we will develop an algorithm that uses the stem and calyx for tracking the inspection progress.

Our research question, *"How can we use stem and calyx detection, feature matching and rotation estimation to predict whether a large proportion of the surface of the fruit has been inspected?"*, will be answered by the four sub-questions: 1) How well can we detect the stem and calyx of different fruit commodities? 2) Which algorithms find reliable matching points between consecutive images of fruits? 3) How reliable is the rotation estimation based on stem and calyx detection



(a) Calyx. (b) Stem.

Fig. 2: The stem and calyx of a pomegranate.

or matching points? 4) How can we use the estimated rotation to calculate the observed area?

In this paper a new approach is proposed for calculating the total observed surface area on a series of images for different types of sphere and oblate spheroid-shaped fruits, such as pomegranate (sphere) and blueberries (oblate spheroid). First, a 3-dimensional sphere or spheroid model of the fruits is fitted using data from the image series. Then the rotation is calculated based on matching points between two consecutive images. Two different methods are used for finding matching points. As a baseline we use the popular and fast YOLOv5 object detection network [6], [7] for detecting the stem and calyx. Also, we create a second set of matching points using the limited-texture feature matching algorithm LoFTR [8] for improved rotation estimation. After mapping the found matching points to 3D, we can determine the rotation between two consecutive images. With the calculated rotations, the total observed surface area of the sphere or spheroid can be determined.

After presenting an overview of relevant works in Section II, we make the following contributions:

- A novel fruit inspection tracking algorithm is proposed in Section III, which generalizes to all spherically and oblate spheroid shaped fruits.
- A stem and calyx detection algorithm is developed in Section III-C based on YOLOv5 [7],
- Problems with the small amount of matching points after stem and calyx detection are tackled by applying the LoFTR algorithm [8] as described in Section III-D.
- A new method for calculating the observed area using rotation estimation, matching points and 3D modelling is presented in Section III-F.

Our methodology is thoroughly evaluated in Section IV. Section VI concludes the paper, followed by Section VII which provides interesting directions for future research.

II. Literature

This section presents relevant works from literature. First, in Section II-A, application-related works are presented. In Sections II-B, II-C and II-D methodology-related works are presented, which provide the basis for stem and calyx detection, rotation estimation and feature matching, respectively. Note that most fruit inspection research concerns commercial product development, therefore limited works involving stem and calyx detection and progress tracking are available.

A. Fruit inspection

On the topic of capturing the entire fruit area, different automated fruit inspection methods have been proposed, of which most focus only on a single type of fruit. For example, Zou et al. [9] proposed a system using multiple cameras which capture multiple images at once. Other systems use mirrors in order to cover the entire surface of a fruit [10] or control the rotations of the fruit [11]. However, controlling the rotation of the fruit, slows down the inspection process.

Recent work from Albiol [4] have used spheroid models in order to model the rotation of fruits. Using the shape and size of tomatoes, each tomato is modeled as a spheroid or sphere. The images of the tomatoes are captured at different places on a roller conveyor, leading to a series of images of each tomato. For each image series, a 3D model is created using the dimensions calculated from the 2D images. Using matching points between images, the rotation can be determined using the 3D model. Two drawbacks of this model are that it can only be used on spherically shaped fruits and that the fruits need to have texture in order to match the series of images.

B. Stem and calyx detection

In the past couple of years different methods to detect the stem and calyx have been explored and evaluated, but most methods are specific to a single type of fruit, instead of presenting a general solution. Due to the similarity, stem and calyxes often get confused with large size defects as both often show a lot of texture and wide variety of color [3]. Most literature focuses on the detection of both the spots and the calyx/stem, in order to reduce misclassification. Sun et al. [8] proposes an approach based on the difference between black spots and the stem. Assuming that the black spots are densely distributed while the stems are complex, they designed a feature skyscraper detector based on this distinction. Other solutions without the use of deep learning have been proposed by for example Zhang et al. [12] in which

near-infrared linear-array structured light is projected on an apple. Due to the deformation of the light being different for the calyx and stem with respect to the rest of the apple, these features can be detected.

Other detection algorithms use different features in combinations with a support vector machine (SVM) [13], [14]. More recently, the field seems to slide towards the use of convolutional neural networks (CNN) instead of classical computer vision techniques, because of the advances in the field of deep learning. For instance Zhang et al. [15] used a CNN for the detection of bruises and calyxes on blueberries. This method used hyperspectral (87 channels) images, as well as three and nine channel images, obtained from the hyperspectral images. The hyperspectral images show an Intersection over Union (IoU) accuracy for calyx detection of 84.1%, while the three and nine channel images showed similar results of 82.8% and 82.6% respectively.

The stem and calyx can also be detected using deep learning. One-stage detectors, such as EfficientDet [16], Faster R-CNN [17] and YOLO [6], [7] might be able to detect the stem and calyx and can work real time. The network architecture with the highest speed is YOLO. Over the years, different YOLO versions have been released, with the latest being YOLOv5. Different sizes of models exist in this version of YOLO, ranging from v5n, v5s, v5m, v5l to v5x, with v5n being the fastest, but less accurate and v5x being more accurate, but slower. YOLOv5 has recently been used to train models for the detection of the stem and calyx in apples [18], leading to a mean average precision (mAP) of 93.89% for the v5s model, with a speed of 177 frames per second (FPS) on a GPU. An overview of all stem and calyx detection algorithms proposed in the past is given in Appendix B, Table IV.

C. Rotation estimation

One method for estimating the rotation of a fruit is to determine the apparent motion between two images, which can be determined using optical flow. Optical flow is an approximation of the physical movement [19]. However, in order to be able to use optical flow, the constant brightness assumption needs to hold and the displacement of the pixels needs to be small. The constant brightness assumptions holds if a pixel has the same brightness in both images. However, due to the spherical shape of fruit, brightness will change when a point moves towards the edge. To calculate the optical flow a second constraint needs to hold, the displacement constraint. Since optical flow is based on linear approximations, the displacement between two images in x and y direction needs to be small, as well as the time between two images. Due to this, methods based on differential equations can not be used in a sequence of images with a displacement of the fruit larger than a few pixels [20]. To overcome this use, large displacement optical flow algorithms are used [20]. However, large displacement optical flow still relies on the brightness assumption.

D. Feature matching

To match images correctly, we need an algorithm to match features. Classical feature matching algorithms like SIFT [21] and SURF [22] base matches on found features, for example corners. Since in a perfect fruit, no or little texture is present on the surface area, these algorithms will find matches on the fruit contour, rather than on the fruit. But even if features are found, there could still be an issue with repeatability between the found points, especially with points rotating in or out of the image. Another issue with classical feature matching algorithms is that they are not position dependent. This can lead to points being close in one image, to be matched with points in another image far away from each other. SuperGlue [23] uses a graph neural network (GNN) to overcome the issue of position. First, a detector is used to find points of interest. A graph is created, using self attention and cross attention [24] to aggregate information between feature points, within an image and between different images. Using the graph, a partial assignment problem is solved in order to match the correct points and reject unmatched points. Even though SuperGlue's performance is better in low textured areas than SIFT or SURF, it still uses a detector to find interest points. LoFTR [8] uses the GNN as proposed in SuperGlue, but it overcomes the issue of repeatability by creating a detector-free algorithm. A local feature CNN is used to extract the coarse- and fine-level feature maps, which are flattened and combined with position encodings. With the position encodings, features are not only dependent on the descriptions of the features but also on the position in the images. This improves the overall results of the matching algorithm on places with little to no texture.

III. Method

This section will first discuss the data available in Section III-A. Then the method to create a 3D model is presented in Section III-B followed by the detection of the stem and calyx using YOLO and the feature matching algorithm LoFTR in Section III-C and III-D, respectively. To finalize our approach, the rotation estimation algorithm and progress tracking algorithm are presented

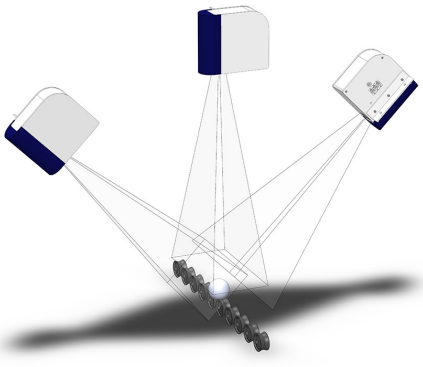


Fig. 3: The camera system used when three cameras are available. In a system with only a single camera, the camera positioned above the roller conveyor is used.

in Section III-E and III-F. An overview of the entire proposed method is shown in Figure 8.

A. Data

This research is conducted in collaboration with Ellips B.V. [25]. This company designs camera modules and software for fruit inspection machines using roller conveyors. The software is designed for machines which can sort small to mid-size ellipse and ellipsoid shaped fruits, such as blueberries, apples and pomegranates.

Different fruit commodities are used for evaluating our methodology. The fruits are captured in series of eight to ten consecutive images, each at different locations on the roller conveyor. For larger fruits, three distinct images are captured at the different locations on the roller conveyor, each at a different angle (-45°, 0°and 45°). An overview of a camera system with three cameras is shown in Figure 3. In case the camera system uses only a single camera, the camera positioned directly above the roller conveyor is used.

The cameras each capture three different images in distinct spectral ranges: 1) visible light, 2) near-infrared (NIR) and 3) soft, a spectral band close to near-infrared. The specific optimal wavelengths used for NIR and soft can differ for each type of fruit.

An overview of the fruit specific details is presented in Table I. An example of the thirty color images captured of a pomegranate is shown in Figure 4.

The number of available fruits in our data set differs for each fruit commodity. The data set available for each commodity is split into a training, validation and test set, according to a 70%, 15% and 15% split, respectively. The exact number of unique instances available for each commodity and each subset is summarized in Table II.

All images are box or point annotated for both the stem and the calyx, which will be used as ground truth to train our model for stem and calyx detection, examples of the stem and calyx annotations for each fruit type are shown in Appendix A.

For all pomegranates in our test and validation set, additional manual landmark annotations have been created, such that each matching pair of images, taken with the top camera, has at least three matching points. For a subset of the test set, two different sets of manual landmark annotations are available. These matching points are used to determine a ground truth for the algorithm.

B. 3-dimensional model

For modeling the 3D structure of the fruits, we resort to the sphere and oblate spheroid model from Albiol et al.[4], as shown in Figure 5a and 5b respectively. However, in order to generalize the method proposed in this paper, an additional prolate spheroid model needs to be made. Table I shows which model is used for which type of fruit. The sphere and oblate spheroid models are characterized by the size of their major and minor principal axes, which are equal for sphere models specifically. The size of these axes in the 3-dimensional model can be determined based on the principal axes of the projected ellipses in the time series as we will show shortly. The principal axes of the images can be computed using the covariance matrix of the pixels corresponding to the object. First the center of the object in the images (c_x, c_y) is calculated using the sample mean as

$$c_x = \frac{1}{N}\sum_{n=1}^{N} x_n \qquad c_y = \frac{1}{N}\sum_{n=1}^{N} y_n \tag{1}$$

where the set $\{(x_n, y_n)\}_{n=1}^{N}$ denotes the coordinates of all pixels in the image mask and N the total number of pixels in this mask. Based on the center of the image (c_x, c_y), we can extract the covariance matrix Σ of the projected ellipse for each image, which we represent as

$$\Sigma = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{bmatrix}, \tag{2}$$

where the individual elements are computed using the sample (co)variance

$$\sigma_x^2 = \frac{1}{N}\left(\sum_{n=1}^{N} x_n^2\right) - c_x^2 \qquad \sigma_y^2 = \frac{1}{N}\left(\sum_{n=1}^{N} y_n^2\right) - c_y^2 \tag{3}$$

$$\sigma_{xy} = \frac{1}{N}\left(\sum_{n=1}^{N} x_n y_n\right) - c_x c_y \tag{4}$$

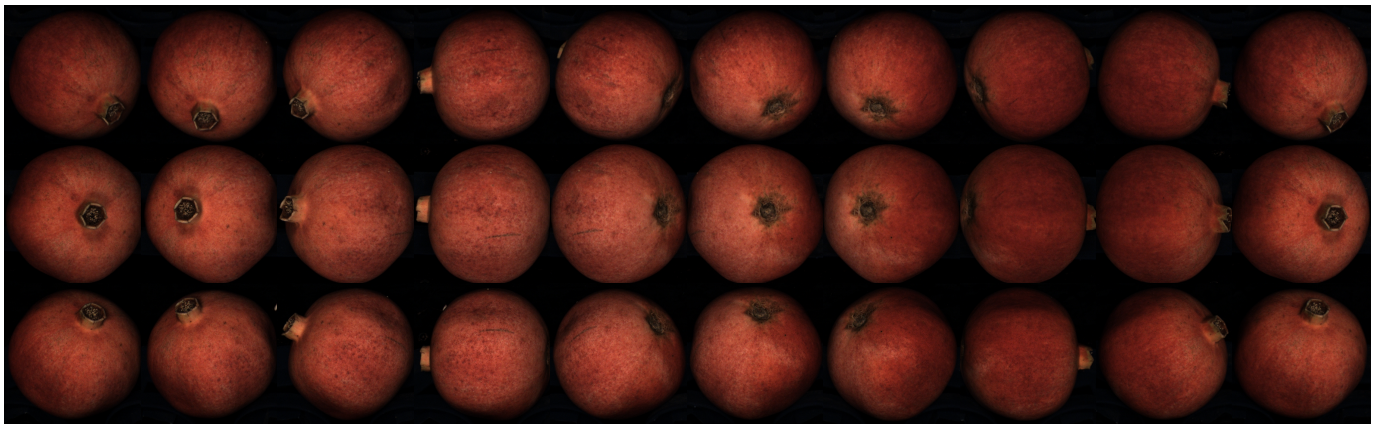


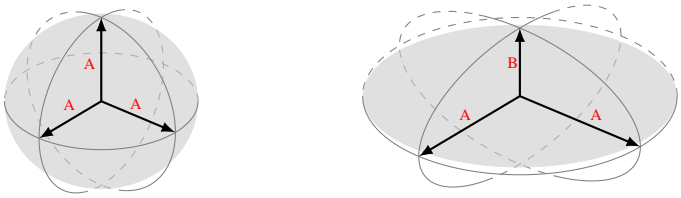
Fig. 4: Example of 30 pomegranate images, the images in the top row are taken with the right camera, the middle row is taken with the top camera and the bottom row is taken with the left camera, as shown in Figure 3.

TABLE I: Details of the available dataset.

Fruit	Number of cameras	Number of channels	Number of images per camera	Average image size [pixels]	Processing speed [fruit/s]	model
Apple	3	3	8/10	349 x 332	10	Sphere
Blueberry	1	3	10	119 x 117	90	Oblate spheroid
Kiwi	3	3	10	276 x 250	12	Prolate spheroid
Lemon	2/3	2/3	10	358 x 290	12	Prolate spheroid
Lime	3	3	10	266 x 241	12	Sphere
Mandarin	3	3	10	251 x 248	12	Oblate spheroid
Pear	3	2	10	518 x 312	4	Prolate spheroid
Pomegranate	3	3	10	339 x 342	8	Sphere

TABLE II: Number of unique fruits for each fruit commodity in each subset.

Fruit	total	train	validation	test
Apple	182	119	33	30
Blueberry	204	143	29	32
Kiwi	402	281	60	61
Lemon	259	175	40	44
Lime	38	26	5	7
Mandarin	48	33	7	8
Pear	103	72	15	16
Pomegranate	200	140	30	30



(a) Sphere model. (b) Oblate spheroid model.

Fig. 5: The 3D models of a sphere and an oblate spheroid.

Based on the covariance matrix, the eigenvectors $\mathbf{v_1}$ and $\mathbf{v_2}$ and eigenvalues λ_1 and λ_2 of the projected ellipse can be calculated, as shown in Figure 6. $\mathbf{v_1}$ corresponds to the direction of the largest variance in the image, in our case the largest axes, with λ_1 being the corresponding magnitude. $\mathbf{v_2}$ is the eigenvector orthogonal to v_1 and corresponds to the smallest axes, with λ_2 being its magnitude. If the fruit can be modeled as a sphere the eigenvalues are approximately equal, i.e. $\lambda_1 \approx \lambda_2$. For oblate spheroids the eigenvalue corresponding to the semi-minor principal axis λ_2 is smaller or equal to the eigenvalue of the semi-major principal axis λ_1. From these eigenvalues the lengths of the semi-major and semi-minor principal axes can be calculated by

$$a = 2\sqrt{\lambda_1} \qquad b = 2\sqrt{\lambda_2}. \tag{5}$$

1) 3D sphere model: In general an ellipsoid which is centered at the origin can be described by

$$\left(\frac{x}{A}\right)^2 + \left(\frac{y}{B}\right)^2 + \left(\frac{z}{C}\right)^2 = 1. \tag{6}$$

For spheres all sides are equal so $A = B = C$. Also, the major and minor axis of each image in an image

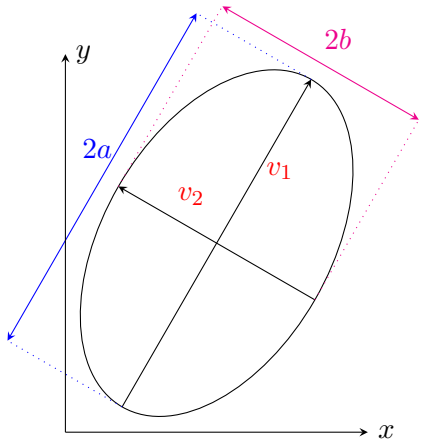


Fig. 6: The principal axes with the corresponding lengths for an oblate spheroid.

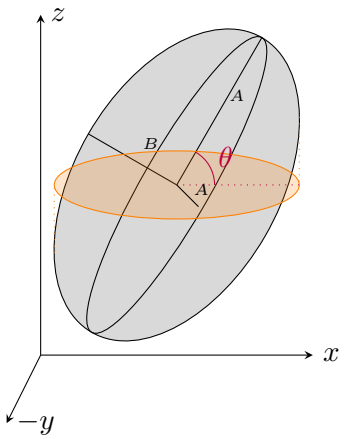


Fig. 7: The elevation angle of a rotated oblate spheroid.

series will be be equal. Therefore, A can be calculated by averaging the major and minor axes over all images

$$A = B = \frac{1}{N_v} \sum_{i=1}^{N_v} \frac{a_i + b_i}{2} \tag{7}$$

with a_i and b_i representing the length of the principal major and minor axis of each image, respectively and N_v the number of available views. Since Equation (6) is formulated for a sphere centered around the origin, we need to center the x and y coordinates around the center of the image, $(x', y') = (x - c_x, y - c_y)$. Based on the length of the principal axes, we can determine the z-coordinate of each pixel coordinate (x_n, y_n) by

$$z_n = \sqrt{A^2 - (x'_n)^2 - (y'_n)^2} \tag{8}$$

2) 3D oblate spheroid model: An oblate spheroid has two axes with length A and one smaller axis with length B, as shown in Figure 5b. Since two axes have length A, the longest principal axis in each image will be of length A. The length of the principal minor axis is obtained as the smallest principal minor axis length over all images, which occurs when one of the major principal axes of the fruit are parallel to the camera orientation, resulting in

$$A = \frac{1}{N_v} \sum_{i=1}^{N_v} a_i \qquad B = \min_i b_i. \tag{9}$$

If the axes lengths of the oblate model are determined, we need to determine the elevation angle θ. This is the angle between the longest axis A of the fruit and the camera plane as shown in Figure 7, of each image by

$$\theta = \arccos \sqrt{\frac{b^2 - B^2}{A^2 - B^2}}. \tag{10}$$

The corresponding derivation is given in Appendix C.

With the eigenvectors, $\mathbf{v}_1$ and $\mathbf{v}_2$, and the elevation angle θ the pose matrix P can be determined. The pose matrix describes the principal axes of the oblate spheroid for each image and can be calculated by

$$\mathbf{P} = \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{bmatrix} \tag{11}$$

with the first row the unit vector in the direction of $\mathbf{v}_b$, as drawn in Appendix C, Figure 17, and the second row the vector $\mathbf{v}_1$

$$\begin{aligned} p_{11} &= \mathbf{v}_{2x} \sin\theta, & p_{12} &= \mathbf{v}_{2y} \sin\theta, & p_{13} &= \cos\theta \\ p_{21} &= \mathbf{v}_{1x}, & p_{22} &= \mathbf{v}_{1y}, & p_{23} &= 0 \end{aligned}$$

With $\mathbf{v}_1$ and $\mathbf{v}_2$ the eigenvectors of the covariance matrix. The third row needs to be perpendicular to the first and second row and can be calculated by

$$\begin{bmatrix} p_{31} & p_{32} & p_{33} \end{bmatrix}^T = \begin{bmatrix} p_{11} & p_{12} & p_{13} \end{bmatrix}^T \times \begin{bmatrix} p_{21} & p_{22} & p_{23} \end{bmatrix}^T$$

The pose matrix can be used to rewrite the general sphere equation, as given in Equation 6, to a general equation for non-axis aligned oblate spheroids. This derivation is given in Appendix D. The z value can then be calculated by solving

$$\begin{bmatrix} x' & y' & z \end{bmatrix} \mathbf{P}^T \begin{bmatrix} \frac{1}{B^2} & 0 & 0 \\ 0 & \frac{1}{A^2} & 0 \\ 0 & 0 & \frac{1}{A^2} \end{bmatrix} \mathbf{P} \begin{bmatrix} x' \\ y' \\ z \end{bmatrix} = 1. \tag{12}$$

C. Stem and calyx detection

For estimating the rotation, we need matching features between images. The most distinctive features of fruit are the stem and calyx. Therefore, detecting those features is of vital importance. Here we present a stem and calyx detection approach based on the YOLOv5 image object

detection network. YOLOv5 consists of models of different sizes, of which we will focus on the three smallest ones, v5n, v5s and v5m, due to speed limitations. More information regarding YOLOv5 is given in Appendix E.

1) Post processing: Post-processing is applied to the results of the detection algorithm to eliminate false positives. Since we are detecting the stem and calyx it is known that there can be a maximum of one stem and one calyx within an image, where they are approximately positioned on opposites sides of the fruit.

Using these two criteria we can post-process the detection to remove any double detections. If a stem and calyx is detected we can check the distance between them, since this distance needs to be approximately equal to the size of the major or minor principal axes, depending on the model used. If the stem and calyx are detected too close to each other, we will keep the detection with the highest confidence. If multiple stems or calyxes are detected, we will also keep the stem or calyx with the highest confidence.

D. Feature detection and matching

If we base our algorithm only on the stem and calyx, we will most likely only find one matching point, while we need at least three to obtain a unique solution for the rotation matrix. Besides this, we will not find a rotation if the fruit rotates around the stem-calyx axis. This will be due to the stem and calyx remaining in approximately the same place, as shown in Appendix F, Figure 20. Since we will assume that perfect fruits do not contain any damage, we need a solution that will be able to match low-texture images. In order to match the images, LoFTR will be used to find matching points between two consecutive images. As described in Section II-D, LoFTR is a network that is able to find matching points in low-textured areas. Since no ground truth data is available for training, we will use a pre-trained LoFTR model. Pre-trained models are available trained on indoor and outdoor images. Since more and better matches will be found with the model trained on indoor images, this model will be used.

When LoFTR has found matching points, the points outside of the fruit region are discarded, as well as the points close to the edges in order to prevent matches based on shape, instead of fruit texture.

E. Rotation estimation

If we have found matching points in the two 2D successive images, as shown in Figure 8a and 8b, we can rewrite them as 3D points as described in Section III-B and shown in Figure 8c and 8d. Based on these 3D points we can calculate the rotation between these points and therefore between the two successive images. Most of the time only one matching point is found when using stem and calyx detection. This is due to the fruit shape, with the stem and calyx being on approximately opposite sides of the fruit. The only time two matching points can be found is when the calyx is sticking out and the fruit has a limited movement. We will use two different methods for calculating the rotation, one based on one or two matching points from the stem and calyx detection and another method that will use the matching points found by LoFTR.

1) Rotation estimation using stem and calyx: In case we only use the result of stem and calyx detection as matching points, we can assume that in almost all cases this leads to only one matching point, either the stem or calyx. In this case the point in both images will be rewritten in 3D coordinates as described in Section III-B. To determine the rotation matrix between these points we need to find the matrix that transforms the initial point to the new point. To determine the rotation matrix, we use Rodrigues' rotation formula [26].

We will rewrite the points as vectors from the origin to the point, denoted by $\mathbf{w_1}$ and $\mathbf{w_2}$. Using the vectors we first need to determine the rotation axis by

$$\mathbf{w} = \frac{\mathbf{w_1} \times \mathbf{w_2}}{||\mathbf{w_1}||||\mathbf{w_2}||} \tag{13}$$

Than we can also calculate the rotation angle α,

$$\cos\alpha = \frac{\mathbf{w_1} \cdot \mathbf{w_2}}{||\mathbf{w_1}||||\mathbf{w_2}||} \tag{14}$$

Using the rotation axis we can determine the cross product matrix of $\mathbf{w}$, $\mathbf{k}$, by

$$\mathbf{k} = \begin{bmatrix} 0 & -\mathbf{w}_z & \mathbf{w}_y \\ \mathbf{w}_z & 0 & -\mathbf{w}_x \\ -\mathbf{w}_y & \mathbf{w}_x & 0 \end{bmatrix} \tag{15}$$

With the cross product matrix $\mathbf{k}$ and the rotation angle we can calculate the rotation matrix R by [26]

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \mathbf{k}\sin(\alpha) + (\mathbf{k}\cdot\mathbf{k})(1-\cos(\alpha)). \tag{16}$$

2) Rotation estimation with multiple matching points: To calculate the rotation between two sets of 3D points, Q_1 and Q_2, we use Kabsch algorithm [27]. First we will need to center the two sets of points around the origin to remove the translation component. We can determine the center of the set of points by

$$c_{Q1} = \frac{1}{q}\sum_{i=1}^{q} Q_1^i, \qquad c_{Q2} = \frac{1}{q}\sum_{i=1}^{q} Q_2^i. \tag{17}$$

With c_{Q1} the center of the points of image 1 and c_{Q2} the center of the points of image 2. Using the centers we can center the points around the origin and calculate the matrix, H, by

$$H = (Q_1 - c_{Q1})(Q_2 - c_{Q_2})^T \tag{18}$$

Using the singular value decomposition (SVD), which will decompose matrix H in three geometric transformations, we can determine U (rotation or reflection), S (scaling) and V^T (rotation or reflection) [28] .

$$[U, S, V^T] = \text{SVD}(H) \tag{19}$$

Using U and V we compute the rotation matrix R as [27], [29]

$$R = V \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & d \end{bmatrix} U^T \tag{20}$$

with

$$d = \text{sign}(\det(VU^T)). \tag{21}$$

This additional matrix, including the sign of the determinant of VU^T, is needed to ensure that the solution is not a reflection, but the right-handed coordinate system that we want.

F. Inspection progress tracking

After the 3D model is created and the rotation matrices between the consecutive images are calculated, we need to determine if the entire fruit area has been seen. For this a grid of points will be created on the first image of the series, as shown in Figure 8e. In a perfect case the points will be on each pixel, but to increase the speed the points will be distributed less densely. In case three cameras are used, this grid consists of the points that can be seen from all three cameras. These points are created by rotating the points of image 1 by the rotation matrix R_{right} to get the points on the top camera and R_{left} to get the points on the bottom camera. Since the cameras are mounted at a 45°angle, these rotation matrices will be similar for all setups.

If we rewrite the area of a sphere to spherical coordinates (r, θ, ϕ), everything has been seen if the plot covers $\theta \in [0, \pi]$ and $\phi \in [0, 2\pi]$. We will use this to determine if the entire area has been seen. We will first rewrite the grid points found in the first image to spherical coordinates. From these spherical coordinates, (ϕ, θ) will be plotted with a block size of $(gridsize, gridsize)$ to correct for the spacing between the points, as shown in Figure 8g. After this the grid points from image 1 will be rotated by the rotation matrices calculated as described in Section III-E. After each rotation each point will be rewritten in spherical coordinates and plotted, as shown in Figure 8h. The result after all rotations will show which area of the fruit has been seen in the images, as in Figure 8i, with white being the areas seen and black the areas that are not seen within the images.

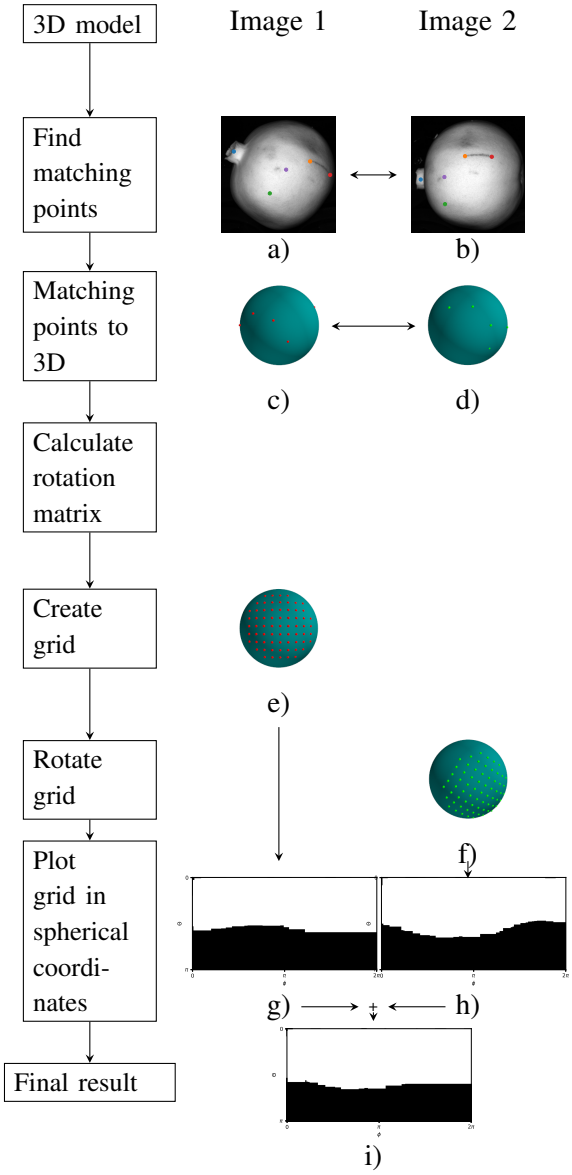


Fig. 8: Overview of the proposed algorithm. **a)** Image 1 with matching points. **b)** Image 2 with matching points. **c)** 3D matching points corresponding to image 1. **d)** 3D matching points corresponding with image 2. **e)** Grid created with a resolution of 30 pixels and a radius of 0.8 $\times$ fruit radius. **f)** Grid rotated by the rotation matrix; **g)** Inspected area using grid shown in e. The white parts is the area seen, while black has not been seen. **h)** Inspected area using the grid shown in f. **i)** Total inspected area after image 1 and 2.

IV. Experiments

In this section, an overview of the experiments will be presented. First the experiments to validate our YOLO model for stem and calyx detection will be presented, followed by the experiment to determine the effectiveness of LoFTR for feature extraction and rotation estimation. Lastly, we will present the experiments to validate the results of our inspection tracking algorithm.

A. Stem and calyx detection

Multiple network structures of different sizes exist within YOLOv5, but only the three smallest networks, nano (v5n), small (v5s) and middle (v5m) are considered as a result of the speed requirements. These three models are trained using all available data, containing all fruit and image types. Their performance in terms of accuracy and speed is reported for image sizes of 384×384 pixels.

The model with the best trade-off between inference time and performance, measured as the mean average precision (mAP), is selected for the next experiments. In order to determine the effect of the input image resolution on the performance results and inference time of the optimal model, this model is also trained using an image resolution of 256×256 and 128×128 pixels. Again, the model with the best trade-off between inference time and performance is selected and is trained on the different image types (NIR, soft and color) separately, to evaluate whether there is a difference in performance. This model is also trained on fruit specific data to determine whether there is a difference in performance between the networks trained on all data and fruit specific data. Both networks are validated for each fruit type separately. All models are trained for 200 epochs, with early stopping, on a GTX 1050Ti GPU, with a batch size of 8. The model corresponding to the epoch with the lowest loss, which consists of Generalized Intersection over Union (GIoU), objectness and classification losses, on the validation set is used for testing.

B. Feature matching

Since LoFTR might find false matches, we need to determine the effect of inaccurate matching points for rotation estimation and determine whether some of these points need to be pruned. For this we will conduct an experiment comparing the different pruning rates with increments of 10%. The matches with the highest confidence scores are used to calculate the rotation matrix. Using this rotation matrix and the annotated landmarks for pomegranates, we will rotate the landmarks between two consecutive images. After this the Euclidean distance between the rotated points and the landmarks is calculated, with the goal of minimizing this distance.

C. Rotation estimation

A similar experiment is conducted as described in the previous subsection to select the number of matching points from LoFTR to check whether the rotation matrices are correct. We will determine the rotation matrix between two consecutive images using the landmark annotations, YOLO and LoFTR. For all three rotation methods the distance between the rotated points and the landmarks is computed. The same experiment is used to determine differences between annotators. For this the subset containing two sets of landmarks is used.

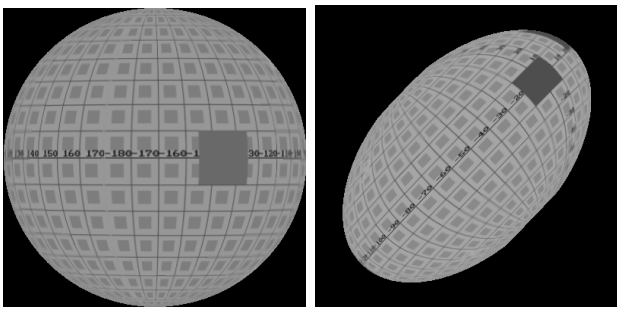
D. Inspection progress tracking - Synthetic data

For verification of the proposed algorithm, synthetic data of a sphere and oblate spheroid is created. An example of the sphere synthetic data and oblate spheroid are given in Figures 9a and 9b, respectively. Image series are created for no rotation, a quarter rotation, half rotation and full rotation using three cameras. For the oblate spheroid additional image series are created with different rotation angles to better match the reality. An example of the synthetic data of an oblate spheroid with full rotation is shown in Appendix G. All image series are annotated with at least three matching landmarks between two consecutive images.

The landmarks are used to calculate the rotation matrices between consecutive images, as described in Section III-E. These rotation matrices are used to determine how much of the area of the fruit is inspected. For all available synthetic data the inspected area is calculated if one or three cameras are used. The area around the edge of the fruit is distorted, even though it is visible. In order to correct for this, different grid sizes are tested, to see which will best match the reality. Specifically, we evaluate our methodology taking into account all points within 100%, 90% 80% and 70% of the radius of the fruit. In order to increase the speed of the algorithm, different grid resolutions are tested, namely: 1, 10, 20, 30, in order to see if this will influence the overall results.

E. Inspection progress tracking - Sphere model

The sphere model will be tested on images of pomegranates. The manual landmark annotations will be used to create a ground truth for the algorithm. Using the manual landmark annotations, three experiments will be conducted.



(a) Synthetic data of a sphere.

(b) Synthetic data of an oblate spheroid.

Fig. 9: An example of the synthetic data created to validate the method proposed.

1) To determine if the manual landmark annotations can be used as a ground truth, a subset of fifteen images of the test set of pomegranate is annotated with manual landmarks by two different annotators. Using these two sets we can determine if there are differences in the results based on these two sets of manual landmark annotations.

2) We compare the results based on the annotations with the results of the algorithm based on stem and calyx rotation and the rotation based on LoFTR.

3) Apart from the landmark annotations for pomegranates, four experts in the field are asked to give their opinion on whether a pomegranate is fully rotated within the image series. Possible answers are "yes", "not fully rotated" and "it can not be determined or seen from the images". For this the pomegranate test set is used, which consists of 30 different image series. We will use these annotations to assert whether there is a relation between the results of the algorithm and these annotations.

F. Inspection progress tracking - Oblate spheroid model

To test the oblate spheroid model, we will use the images of blueberries. For blueberries no landmark annotations are available, so the algorithm based on the rotation between the stem and calyx and LoFTR will be compared. Similarly as with pomegranates, four experts have been asked to look at the rotation of blueberries. Since blueberries have a higher chance of not rotating, four options were given, namely, "no rotation", "not fully rotated", "fully rotated" or "it can not be seen in the images". For this the test set of blueberry is used, which consists of 32 different image series. These annotations are used to determine if the results of the algorithms are similar and if there is a relation between the results and the annotations.

V. Results and discussion

A. Stem and calyx detection

Results of the experiments described previously show that the YOLO v5n network using an image resolution of 256×256 pixels, gives the best results for our data, resulting in a mAP of 0.865 with an inference time of 1.4ms. We can also see that there are no significant differences if different image types are used for training. The complete results are given in Appendix H1.

Using YOLO v5n and image resolution 256×256 pixels, we trained the models for each fruit individually, which we will test on each individual fruit commodity. The model trained on all fruit commodities will also be tested on each individual fruit commodity, to determine if there are any differences. Table III shows the mAP results for each fruit tested on the model trained on all images or individual images, which differ between 0.718 and 0.983 for mandarins and pomegranates, respectively. If the Mann-Whitney U test is applied, all p-values exceed 0.05, so no significant differences are observed. This means that we can use a general model instead of training multiple models. Additional results, including separate results for stem and calyx, are presented in Appendix H2.

We can compare our results with the comparative study by Wang et al. [18]. Using YOLOv5 as their detection algorithm for stem and calyx detection of apples they report better results with a mAP of 93.89%. However, they exclude apples with defects in their test and validation set, artificially improving classification performance.

B. Feature matching

Figure 10 shows that pruning the matches for LoFTR, after disregarding the points close to the edges, does not lead to a smaller distance between rotated and annotated points. From this we conclude that the LoFTR points do not need pruning. Therefore, all matching points are used except for the points found close to the edge of the fruit. Even though no pruning gives the best results, it still shows a median distance of 30 pixels. This could indicate that the matching points found by LoFTR might not be the best matches.

C. Rotation estimation

Figure 11 shows the Euclidean distance between the annotated and rotated points using the different rotation estimation methods. As expected, the manual landmark annotations achieve the best performance. The distance variation for LoFTR is smaller than for stem and calyx

TABLE III: The mAP results for different YOLO v5n models trained on all fruit commodities and one fruit commodity, validated on individual fruit commodities.

Trained	Apple	Blueberry	Kiwi	Lemon	Lime	Manderin	Pear	Pomegranate
All data	0.767	0.846	0.883	0.756	0.766	0.718	0.904	0.983
Own data	0.790	0.851	0.883	0.748	0.753	0.724	0.932	0.984
Mann-Whitney U p value	0.310	1.000	0.896	0.798	0.643	1.000	1.000	0.690

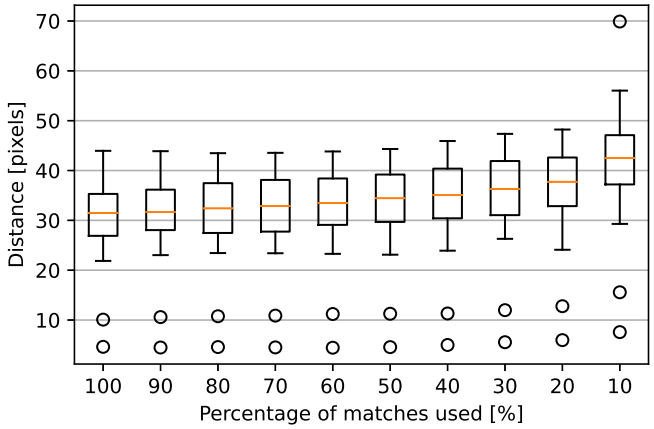


Fig. 10: Average distance between rotated and annotated points for each image series in the pomegranate validation set, using a different number of matching points.

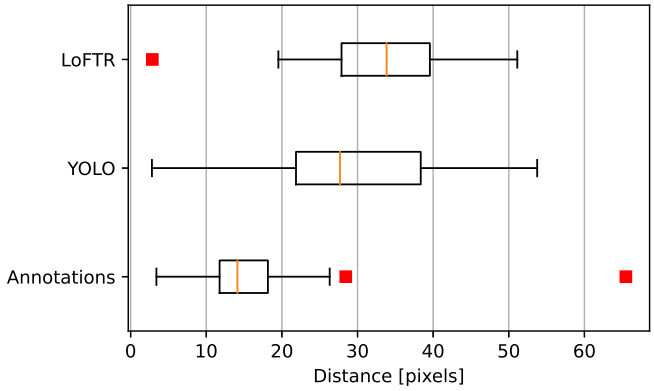
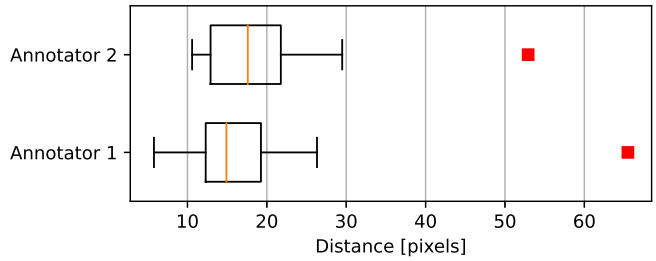
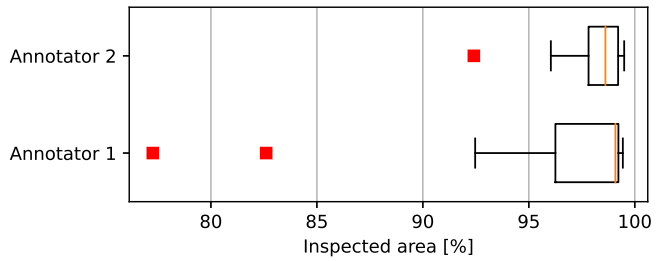


Fig. 11: Average distance between rotated and annotated points on the pomegranate test set, using different methods to calculate the rotation matrices.

rotation, however, the median distance of stem and calyx rotation is smaller. In a perfect case, the distances using annotation would be close to zero, which is not what our results show. This can be due to the assumption that a pomegranate is a perfect sphere or that the manual landmark annotations are placed on the edges of the fruit. At the edges of the fruit the points will be deformed, leading to an incorrect representation of reality. This might also be the cause of the two outliers that are present in the results using the annotations. LoFTR



(a) Difference in distance between annotated and rotated points between different landmark annotations.



(b) Results of the algorithm based on two different sets of landmark annotations.

Fig. 12: The difference in results between different landmark annotations on a subset of 15 images for the test set of the pomegranates.

shows one case where the distance is close to zero, it could be that this specific pomegranate has a lot of texture, leading to better matches.

D. Inspection progress tracking - Synthetic data

We will use the synthetic data to determine the right parameters for the grid. This will show that a grid size of 0.8 * radius is the best fit, this is done to disregard the information around the edge of the fruit since this is distorted. It also shows that a grid resolution of 10 will give similar results as a grid resolution of 1, while increasing the speed. Therefore, these grid parameters will be used for further experiments. The full results of all experiments are given in Appendix I2.

E. Inspection progress tracking - sphere model

Our algorithm, using the previously determined grid parameters, using 80% of the fruit radius and a grid

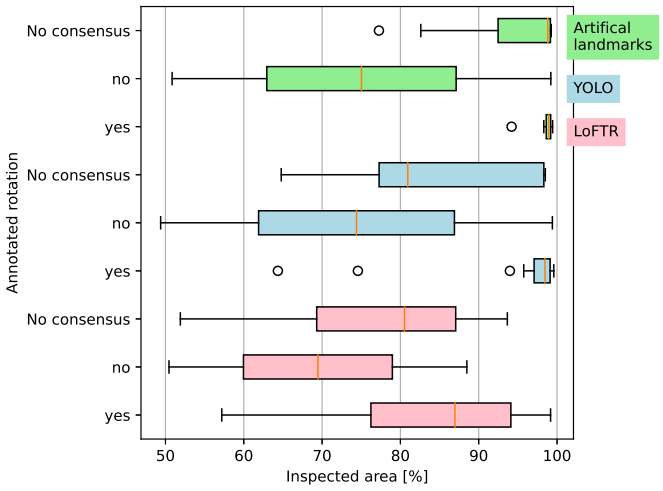


Fig. 13: The results of the algorithm on the test set of pomegranates, using 3 cameras, a grid size of 80% the fruit radius and a grid resolution of 10 pixels, specified for different classification annotations.

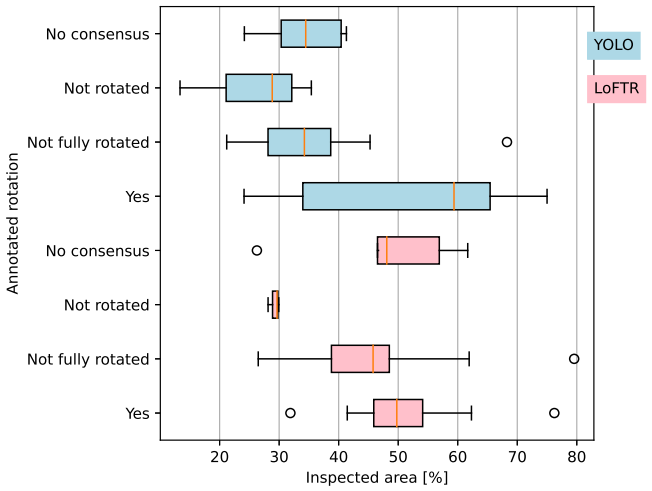


Fig. 14: The results of the algorithm on the test set of blueberries, using 1 camera, a grid size of 80% the fruit radius and a grid resolution of 10 pixels, specified for different classification annotations.

resolution of 10 pixels, is applied on our pomegranate data for the three experiments previously mentioned.

1) The distance between annotated and rotated points and the final results when using two different set of landmarks are shown in Figure 12a and 12b, respectively. The Mann-Whitney U test [30] does not show significant differences, with a p-value of 0.98 and 0.53 for the distance and results, respectively. Therefore, we will use the manual landmark annotations as ground truth.

2) The complete results for the algorithm using annotations, LoFTR and stem and calyx detection are given in Appendix J. It shows that the results of the algorithm using stem and calyx detection are most similar to the results using landmark annotations.

3) If we take the classification annotations into account, we can see that experts in the field do not always agree. In 63% of all images all experts annotated a pomegranate as fully rotated, in 7% all annotators agree that the pomegranate is not fully rotated and in 30% the annotators do not agree on the rotation.

If we plot the results of the algorithm for each annotation, as shown in Figure 13, we can see differences in the inspected area between the different classes. Especially for the annotations and stem and calyx rotation, a larger variation and smaller median can be observed in the set of which annotators do not agree when compared to the fully rotated pomegranates.

The large deviation in the results for the pomegranates that are not fully rotated, is due to the sample size of 2. One of the pomegranates has not rotated at all, while the other rotated for a large amount but not completely.

Using the Mann-Whitney U test, we observe that the difference between annotations and stem and calyx are not significant for the fully rotated pomegranates, with a p-value of 0.16. While between the annotations and LoFTR there are significant differences, with a p-value of < 0.001. For the pomegranates with different annotations there are significant differences between both the annotations and the stem and calyx algorithm and between the annotations and the LoFTR algorithm, with a p-value of 0.02 and < 0.01 respectively.

The algorithm based on stem and calyx detection shows three outliers for the fully rotated pomegranates. These are most likely the results of pomegranates that have rotated around the stem-calyx axes, showing little to no movement in the stem and calyx. Since there is no movement in the stem and calyx, these are likely to be detected as not rotated, while they actually did rotate.

F. Inspection progress tracking - Oblate spheroid model

Even though our synthetic data set showed that a grid resolution of 10 pixels would result in similar results as a grid resolution of 1 pixel, this is not applicable to blueberries due to the difference in image resolutions. Therefore a grid resolution of 1 pixel will be used for the experiments of blueberry, a more elaborate explanation is given in Appendix K.

If we look at the annotation of blueberries, a bigger variation in rotation can be observed. From the test set of blueberries, 22% is annotated as fully rotated, 9% as not rotated, 47% as partly rotated and in 22% of all cases the annotators do not agree with each other.

The results, shown in Figure 14, are similar as with pomegranates, where the inspected area of blueberries that have not rotated is smaller than with fully rotated blueberries. The results of the partly rotated blueberries overall lie in-between the results of the fully and not rotated blueberries. The results based on stem and calyx detection obey a wider spread, while the results within each class using LoFTR are more dense.

VI. Conclusion

After harvesting, fruit needs to be sorted in order to assure the right quality. This can be done using roller conveyors, where fruit rotates while different images are acquired. For correct classification the entire fruit area needs to be seen within the series of images. However, some fruits do not or not fully rotate on the roller conveyor, which can lead to misclassification. In this paper we have shown a method to determine how much of the fruit area is inspected by answering the question: *"How can we use stem and calyx detection, feature matching and rotation estimation to predict whether a large proportion of the surface of the fruit has been inspected?"*

To answer this we have first shown that we can use a general YOLO v5n model to detect the stem and calyx of different fruit commodities, instead of training fruit specific models. After this we have shown that we can use the detection network LoFTR to find more matching points. However, if these points are used to calculate the rotation they are less reliable than the rotation based on stem and calyx. Finally, after fitting a 3D spheroid model, the rotation matrices based on the stem and calyx detections or LoFTR are calculated and used to rotate a grid. Using spherical coordinates, the rotating grid can be used to estimate the inspected area.

Overall, we have shown that inspection progress tracking based on stem and calyx detection using 3D spheroid models, gives a good estimate of the actual rotation. Our method is validated by annotations based on four expert opinions on whether blueberries and pomegranates have rotated. For the rotation estimation of pomegranates, the algorithm based on stem and calyx detection obtains results most similar to the manual landmark annotations. The stem and calyx algorithm also outperforms the LoFTR approach, both in accuracy and speed. For blueberries we have shown that both algorithms can deviate between fully and not rotated blueberries, while the algorithm based on stem and calyx is faster.

VII. Future recommendations

The inspection tracking algorithm based on YOLOv5 shows an overall better result, however for the case where the fruit rotates around the stem-calyx axes, rotation based on stem and calyx alone will not give a good representation of the rotation. Therefore it is important to evaluate how often fruits rotate around their stem-calyx axes in order to quantify the need for a better matching algorithm. For obtaining a better matching algorithm, LoFTR can be re-trained to find better matches. However, this requires ground truth data, which is hard to obtain.

In order to improve and validate the algorithm the correct ground truth is needed. In this paper the ground truth was determined using manual annotations. However this is time consuming and can lead to errors, since limited numbers of matching points are available. A possible method that might simplify the creation of the landmarks is to use a marking that is not visible in the color images, but is visible in other wavelengths. In this way the matching points can be determined in the images at another wavelength and these can be matched to the color images.

Recently, two new YOLO architectures have been published, YOLOv6 [31] and YOLOv7 [18]. There are no comprehensive comparisons between YOLOv5, v6 and v7 available yet, however these might yield better or faster processing. The stem and calyx detection might be improved by using the rotation estimation. If a fruit is rotating approximately constant, the rotation can be used to determine the location of the next stem or calyx. However, to use this more research needs to be done to determine if fruits always rotate constantly.

In order to increase the speed of the algorithm, the grid resolution can be optimized. By determining the optimal grid resolution based on fruit size, the algorithm can use the optimal grid for each individual fruit.

If you want to generalize the algorithm to all types of fruit, it is important to create an additional prolate spheroid model. If the two models presented in this paper are combined with a prolate spheroid model, almost all fruit sorts can be modeled.

References

[1] M. Shahbandeh, "Fresh fruit production worldwide 2020," Jan 2022. [Online]. Available: https://www.statista.com/statistics/262266/global-production-of-fresh-fruit/

[2] J. W. Eckert and N. F. Sommer, "Control of diseases of fruits and vegetables by postharvest treatment," *Annual Review of Phytopathology*, vol. 5, no. 1, pp. 391–428, 1967. [Online]. Available: https://doi.org/10.1146/annurev.py.05.090167.002135

[3] J. Blasco, N. Aleixos, and E. Moltó, "Machine vision system for automatic quality grading of fruit," *Biosystems Engineering*, vol. 85, no. 4, pp. 415–423, 2003.

[4] A. Albiol, A. Albiol, and C. S. de Merás, "Fast 3d rotation estimation of fruits using spheroid models," *Sensors*, vol. 21, no. 6, mar 2021.

[5] C. Flemmer, H. Bakker, and R. Flemmer, "Analysis of the stochastic excursions of tumbling apples," *Computers and Electronics in Agriculture*, vol. 188, p. 106362, 2021. [Online]. Available: https://doi.org/10.1016/j.compag.2021.106362

[6] J. Redmon, S. K. Divvala *et al.*, "You only look once: Unified, real-time object detection," *CoRR*, vol. abs/1506.02640, 2015. [Online]. Available: http://arxiv.org/abs/1506.02640

[7] G. Jocher, A. Stoken *et al.*, "ultralytics/yolov5: v3.0," Aug. 2020. [Online]. Available: https://doi.org/10.5281/zenodo.3983579

[8] X. Sun, G. Li, and S. Xu, "FSD: feature skyscraper detector for stem end and blossom end of navel orange," *Machine Vision and Applications*, vol. 32, no. 1, pp. 1–13, 2021. [Online]. Available: https://doi.org/10.1007/s00138-020-01139-5

[9] Z. Xiao-bo, Z. Jie-wen *et al.*, "In-line detection of apple defects using three color cameras system," *Comput. Electron. Agric.*, vol. 70, no. 1, p. 129–134, jan 2010. [Online]. Available: https://doi.org/10.1016/j.compag.2009.09.014

[10] D. Y. Reese, A. M. Lefcourt *et al.*, "Whole surface image reconstruction for machine vision inspection of fruit," in *Optics for Natural Resources, Agriculture, and Foods II*, Y.-R. Chen, G. E. Meyer, and S.-I. Tu, Eds., vol. 6761, International Society for Optics and Photonics. SPIE, 2007, pp. 140 – 148. [Online]. Available: https://doi.org/10.1117/12.738406

[11] N. Vélez Rivera, J. Gómez-Sanchis *et al.*, "Early detection of mechanical damage in mango using nir hyperspectral images and machine learning," *Biosystems Engineering*, vol. 122, pp. 91–98, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1537511014000506

[12] B. Zhang, W. Huang *et al.*, "Computer vision recognition of stem and calyx in apples using near-infrared linear-array structured light and 3D reconstruction," *Biosystems Engineering*, vol. 139, pp. 25–34, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.biosystemseng.2015.07.011

[13] D. Unay and B. Gosselin, "Stem and calyx recognition on 'Jonagold' apples by pattern recognition," *Journal of Food Engineering*, vol. 78, no. 2, pp. 597–605, 2007.

[14] M. S.H and P. C.J, "Stem - Calyx Recognition of an Apple Using Shape Descriptors," *Signal & Image Processing : An International Journal*, vol. 5, no. 6, pp. 17–31, 2014.

[15] M. Zhang, Y. Jiang *et al.*, "Fully convolutional networks for blueberry bruising and calyx segmentation using hyperspectral transmittance imaging," *Biosystems Engineering*, vol. 192, pp. 159–175, 2020. [Online]. Available: https://doi.org/10.1016/j.biosystemseng.2020.01.018

[16] M. Tan, R. Pang, and Q. V. Le, "Efficientdet: Scalable and efficient object detection," 11 2019. [Online]. Available: http://arxiv.org/abs/1911.09070

[17] S. Ren, K. He *et al.*, "Faster R-CNN: towards real-time object detection with region proposal networks," *CoRR*, vol. abs/1506.01497, 2015. [Online]. Available: http://arxiv.org/abs/1506.01497

[18] Z. Wang, L. Jin *et al.*, "Apple stem/calyx real-time recognition using YOLO-v5 algorithm for fruit automatic loading system," *Postharvest Biology and Technology*, vol. 185, p. 111808, mar 2022. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0925521421003471

[19] P. O'Donovan, "Optical flow: Techniques and applications," *International Journal of Computer Vision*, pp. 1–26, 2005. [Online]. Available: http://www.dgp.toronto.edu/~donovan/stabilization/opticalflow.pdf

[20] T. Brox, C. Bregler, and J. Malik, "Large displacement optical flow," *2009 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009*, pp. 41–48, 2009.

[21] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," 2004.

[22] H. Bay, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features."

[23] P.-E. Sarlin, D. Detone *et al.*, "Superglue: Learning feature matching with graph neural networks."

[24] A. Vaswani, N. Shazeer *et al.*, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 2017-December, no. Nips, pp. 5999–6009, 2017.

[25] "Grade fruit and vegetables more accurately than ever before," Feb 2022. [Online]. Available: https://ellips.com/

[26] O. Rodrigues, "Des lois géometriques qui régissent les déplacements d'un systéme solide dans l'espace, et de la variation des coordonnées provenant de ces déplacements considérés indépendamment des causes qui peuvent les produire," *Journal de Mathématiques Pures et Appliquées*, vol. 5, pp. 380–440, 1840.

[27] W. Kabsch, "A solution for the best rotation to relate two sets of vectors," *Acta Crystallographica Section A*, vol. 32, no. 5, pp. 922–923, Sep 1976. [Online]. Available: https://doi.org/10.1107/S0567739476001873

[28] V. Klema and A. Laub, "The singular value decomposition: Its computation and some applications," *IEEE Transactions on Automatic Control*, vol. 25, no. 2, pp. 164–176, 1980.

[29] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-d point sets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, pp. 698–700, 1987.

[30] H. B. Mann and D. R. Whitney, "On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other," *The Annals of Mathematical Statistics*, vol. 18, no. 1, pp. 50 – 60, 1947. [Online]. Available: https://doi.org/10.1214/aoms/1177730491

[31] C. Li, L. Li *et al.*, "YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications," 2022. [Online]. Available: http://arxiv.org/abs/2209.02976

[32] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-January, pp. 6517–6525, 2017.

[33] ——, "YOLOv3: An Incremental Improvement," 2018. [Online]. Available: http://arxiv.org/abs/1804.02767

[34] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," 2020. [Online]. Available: http://arxiv.org/abs/2004.10934

[35] C. Y. Wang, H. Y. Mark Liao *et al.*, "CSPNet: A new backbone that can enhance learning capability of CNN," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, vol. 2020-June, pp. 1571–1580, 2020.

[36] S. Xie, R. Girshick, and P. Doll, "Aggregated Residual Transformations for Deep Neural Networks http://arxiv.org/abs/1611.05431v2," *Conference on Computer Vision and Pattern Recognition*, pp. 1492–1500, 2017.

[37] T. Lin, M. Maire *et al.*, "Microsoft COCO: common objects in context," *CoRR*, vol. abs/1405.0312, 2014. [Online]. Available: http://arxiv.org/abs/1405.0312

[38] R. Xu, H. Lin *et al.*, "A forest fire detection system based on ensemble learning," *Forests*, vol. 12, no. 2, pp. 1–17, 2021.

[39] W. H. Kruskal and W. A. Wallis, "Use of ranks in one-criterion variance analysis," *Journal of the American Statistical Association*, vol. 47, no. 260, pp. 583–621, 1952. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/01621459.1952.10483441

APPENDIX

A. Annotations

An example for the stem and calyx annotations for all types of fruit used in this research is shown in Figure 15 and 16 respectively.

B. Literature overview

An overview of previous research on stem and calyx detection is given in Table IV.

C. Calculation of the elevation angle

If we want to calculate the elevation angle θ, which is the angle between the longest axes v_A and the camera plane, as shown in Figure 7, we will use the cross section of the fruit at $v_1 = 0$, as described by Albiol et al. [4]. We will first plot the fruit in the z, v_2 plane, as shown in Figure 17. From this plot we can see that we can rewrite v_2 by

$$\begin{aligned} v_2 &= v_a \cos(\theta) - v_b \cos(90^\circ - \theta) \\ &= v_a \cos(\theta) - v_b \sin(\theta) \end{aligned} \tag{22}$$

In order to solve this for θ, we need to compute the variance on both sides. The variance of an eigenvector is the eigenvalue, so

$$\mathrm{Var}[v_2] = \lambda_2.$$

For the right hand side of the equation we first start with simplifying the equation to

$$\begin{aligned} z &= v_a \cos(\theta) - v_b \sin(\theta) \\ &= x + y \end{aligned} \tag{23}$$

with $x = v_a \cos(\theta)$ and $y = -v_b \sin(\theta)$. Using the simplification we can calculate a general equation for the variance by

$$\begin{aligned} \mathrm{Var}[z] &= \mathrm{cov}[z, z] \\ &= \mathrm{cov}[x + y, x + y] \\ &= \mathrm{cov}[x, x] + \mathrm{cov}[x, y] + \mathrm{cov}[y, x] + \mathrm{cov}[y, y] \\ &= \mathrm{cov}[x, x] + 2\mathrm{cov}[x, y] + \mathrm{cov}[y, y] \\ &= \mathrm{Var}[x] + \mathrm{Var}[y] + 2\mathrm{cov}[x, y]. \end{aligned} \tag{24}$$

We can now calculate each term to get the full equation. The variance of x can be rewritten as

$$\begin{aligned} \mathrm{Var}[x] &= \mathrm{Var}[v_a \cos(\theta)] \\ &= \cos^2(\theta)\mathrm{Var}[v_a] \\ &= \cos^2(\theta)\sigma_A^2. \end{aligned}$$

In a similar way the variance of y can be rewritten to

$$\mathrm{Var}[y] = \sin^2(\theta)\sigma_B^2. \tag{25}$$

Now we will calculate $\mathrm{cov}[x, y]$ by

$$\begin{aligned} \mathrm{cov}[x, y] &= \mathrm{cov}[v_a \cos(\theta), v_b \sin(\theta)] \\ &= \cos(\theta)\sin(\theta)\mathrm{cov}[v_a, v_b] \\ &= \cos(\theta)\sin(\theta)\sigma_{AB} \\ &= 0 \end{aligned} \tag{26}$$

since $\sigma_{AB} = 0$ this term will cancel and we can express λ_2 as

$$\lambda_2 = \sigma_A^2 \cos^2(\theta) + \sigma_B^2 \sin^2(\theta) \tag{27}$$

We can now use the relation described by Equation (III-B), $\lambda_2 = \frac{b^2}{4}$, and use a similar relation for $\sigma_A^2 = \frac{A^2}{4}$ and $\sigma_B^2 = \frac{B^2}{4}$ to rewrite this to

$$\frac{b^2}{4} = \frac{A^2}{4}\cos^2(\theta) + \frac{B^2}{4}\sin^2(\theta). \tag{28}$$

From this equation the elevation angle θ can be calculated by

$$\begin{aligned} b &= A^2 \cos^2(\theta) + B^2 \sin^2(\theta) \\ &= A^2 \cos^2(\theta) + B^2(1 - \cos^2(\theta)) \\ &= (A^2 - B^2)\cos^2(\theta) + B^2 \end{aligned} \tag{29}$$

$$\cos^2(\theta) = \frac{b^2 - B^2}{A^2 - B^2} \tag{30}$$

$$\cos(\theta) = \sqrt{\frac{b^2 - B^2}{A^2 - B^2}} \tag{31}$$

D. Introduction pose matrix

The general equation of an oblate spheroid can be derived from the general sphere equation, Equation (6).As described previously, an oblate spheroid has 2 larger axes, A and a smaller axes B, so the sphere equation can be rewritten to

$$\left(\frac{x}{B}\right)^2 + \left(\frac{y}{A}\right)^2 + \left(\frac{z}{A}\right)^2 = 1. \tag{32}$$

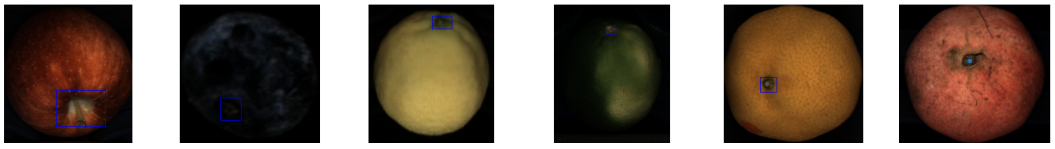
This can be rewritten to matrix form as

$$\begin{bmatrix} x' & y' & z \end{bmatrix} \begin{bmatrix} \frac{1}{B^2} & 0 & 0 \\ 0 & \frac{1}{A^2} & 0 \\ 0 & 0 & \frac{1}{A^2} \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z \end{bmatrix} = 1. \tag{33}$$

However, this only holds for spheroids with their axis aligned to the camera axes. In order to correct for this and obtain a general equation, the pose matrix P is needed. This matrix will describe the spheroid principal axes. Implementing the pose matrix in Equation (33) will lead to

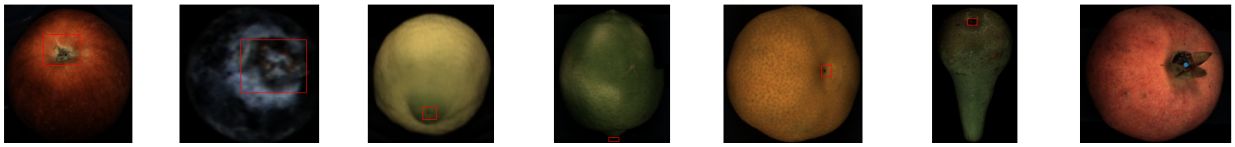
$$\begin{bmatrix} x' & y' & z \end{bmatrix} \mathbf{P}^T \begin{bmatrix} \frac{1}{B^2} & 0 & 0 \\ 0 & \frac{1}{A^2} & 0 \\ 0 & 0 & \frac{1}{A^2} \end{bmatrix} \mathbf{P} \begin{bmatrix} x' \\ y' \\ z \end{bmatrix} = 1. \tag{34}$$

Which can be solved for z in order to obtain the z-coordinate for each 2D coordinate in the images.



(a) Apple. (b) Blueberry. (c) Lemon. (d) Lime. (e) Manderin. (f) Pomegranate.

Fig. 15: Example of stem annotations for each fruit type.



(a) Apple. (b) Blueberry. (c) Lemon. (d) Lime. (e) Manderin. (f) Pear. (g) Pomegranate.

Fig. 16: Example of calyx annotations for each fruit type.

TABLE IV: Literature overview of papers regarding stem and/or calyx detection

Reference	Type of fruit	Methods	Brief description	Results
[13]	Jonagold apples	Support vector machine (SVM)	Statistical, texture and shape features are extracted of the segmented image and classified using a SVM.	99% and 100% correct recognition of stem and calyx respectively, 13% misclassification of defects
[14]	Apples	Shape features	Use shape features to distinguish stem and calyx from defects.	Classification accuracy of 95%
[12]	Apples	NIR linear-array	Detection using the deformation of a linear-array of light.	97.5% recognition accuracy for stem and calyx
[15]	Blueberry	Feature sky-scrapper (FCN)	Segmentation model for the detection of bruised tissue, unbruised tissue and calyx of blueberries. Trained on near-infrared hyperspectral transmittance images. Damage can be detected 30 minutes after impact.	Accuracy of 82.1%
[8]	Navel orange	Feature sky-scrapper detector	Detect stem, blossom ends and black spots on navel oranges. Using the difference in distribution between black spots and stem and blossom ends. Use a feature sky-scrapper based on dense connectivity to distinguish the three classes.	mAP of 87.48%
[18]	Apple	YOLO v5	Different YOLO v5 models are trained to detect the stem and calyx.	mAP of 93.89% for stem and calyx

E. YOLOv5

In this research the commonly known YOLOv5 [7] network is used for stem and calyx detection. This network is based on the previous YOLO versions 1-4 [6], [32], [33], [34]. As all one-stage detection network it consists of a backbone, neck and head. The backbone is used to extract important features, which uses Darknet-53 [33] with a cross stage partial network (CSPnet) [35]. The neck uses a feature pyramid network structure, the path aggregation network (PANet) [36]. The final detection is done using the same head as is used in YOLOv3. This will generate three different sizes of feature maps, making the detection of different sizes of objects possible. An overview of the YOLOv5 architecture is given in Figure 18.

Different sizes of networks are available with YOLOv5, each with different sizes, speed and results. An overview of the different network sizes, trained on a NVIDIA V100 GPU and the Microsoft Common Objects in Context (MS COCO) data set [37] is given in Table V.

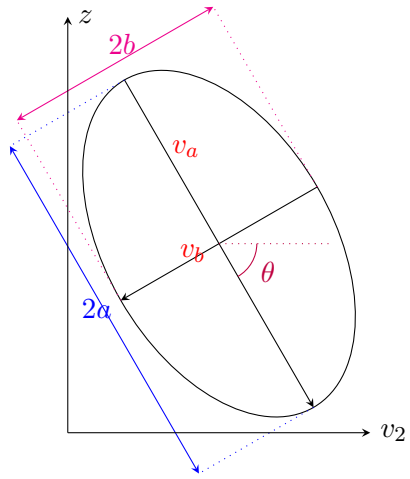


Fig. 17: The (v_2, z) plot of the oblate spheroid

TABLE V: The results on MS COCO data set, speed (NVIDIA V100 GPU) and number of params for different network sizes of YOLOv5 [7].

Model	size (pixels)	mAP	Speed (V100)	Params (M)
v5n	640	45.7	6.3	1.9
v5s	640	56.8	6.4	7.2
v5m	640	64.1	8.2	21.2
v5l	640	67.3	10.1	46.5
v5x	640	68.9	12.2	89.7

F. Example of different rotations

An example of a pomegranate rotating over the stem-calyx axes is shown in Figure 19 and a pomegranate rotation over the stem and calyx is shown in Figure 19.

G. Synthetic data

An example of the oblate spheroid synthetic data is shown in Figure 21

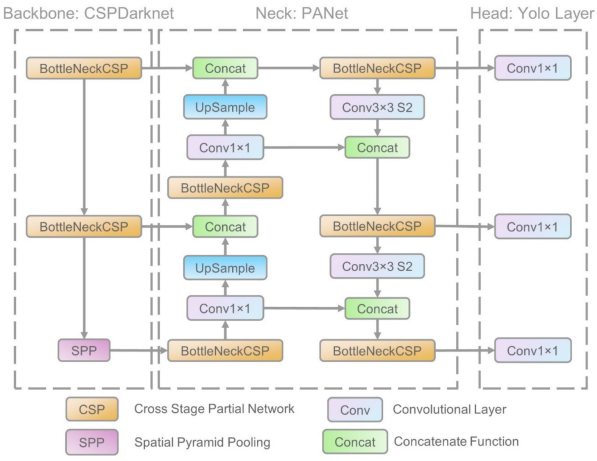


Fig. 18: The YOLOv5 network architecture, adapted from [38]

H. Results stem and calyx detection

1) Result of YOLO v5: Training YOLO v5n, v5s and v5m on all data, all fruit commodities and all image types, results in a mean average precision (mAP) of 0.831, 0.854 and 0.862 with a corresponding inference time of 2.4, 5.7 and 21.1 ms for v5n, v5s and v5m, respectively. The precision (P), recall (R) and mAP for the stem and calyx specifically are summarized in Table VI. The results of the different models do not differ significantly (Kruskal-Wallis H-test [39] with a p-value of 0.16 on subsets of the data), while there is a large difference in inference time. Because of the importance of real-time fruit processing we will use the fastest version, v5n, for further experiments.

The v5n models trained on different image resolutions of 128×128, 256×256 and 384×384 pixels, lead to a mAP of 0.858, 0.865 and 0.825 with an inference time of 2.4, 1.4 and 0.8 ms, respectively. All results of these models are summarized in Table VII. To select the image resolution used for further experiments the trade-off between speed and performance is made again, in this case image resolution 256×256 pixels is selected, since this yields in the best performance, whilst being fast enough to satisfy our requirements.

If we train YOLO v5n for all image types and for the different image types individually, we get similar results with an p-value of 0.5 for the Kruskal-Wallis H-test. All results are summarized in Table VIII.

2) Result for different fruit commodities: All results for YOLO v5n trained on all data and own data is summarized in Table IX.

I. Results synthetic data

1) Sphere model: For synthetic data we will conduct different experiments to validate our method and to determine the right parameters. Figure 22 shows the results for full, half, quarter and no rotation for a different radii of the grid, using one or three cameras and with a grid resolution of 1. It shows that if a grid with a radius of 80% of the fruit radius is used, it is most similar with reality, due to the points on the edge being less visible.

The next experiment is performed with a grid with a radius of 80% of the radius of the fruit, as previously determined. The percentage of the area seen is again calculated, but with a variation in grid resolutions. Figure 23a and 23b show the results for 1 and 3 camera(s) respectively. A bigger grid resolution leads to an over-estimation of the area seen, however, it does lead to a reduction in computational load. The trade-off is made to use a grid resolution of 10 pixels for further experiments, making use of reduction in computational load, while restricting the overestimation.

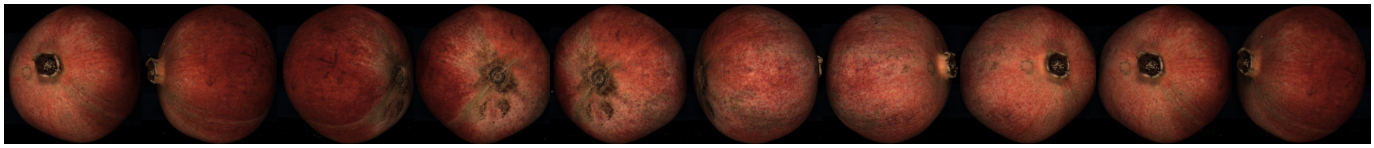


Fig. 19: Example a pomegranate rotation over the stem calyx, taken with the top camera as shown in Figure 3.

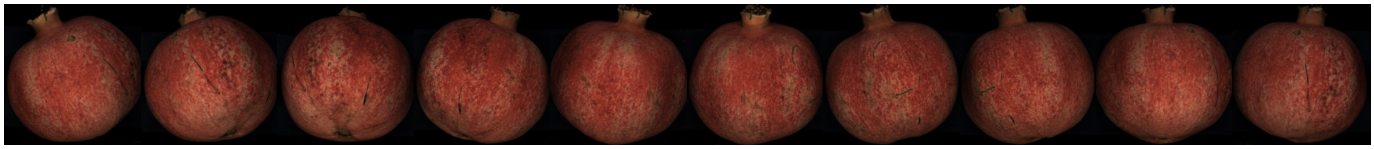


Fig. 20: Example a pomegranate rotation over the stem calyx axes, taken with the top camera as shown in Figure 3.

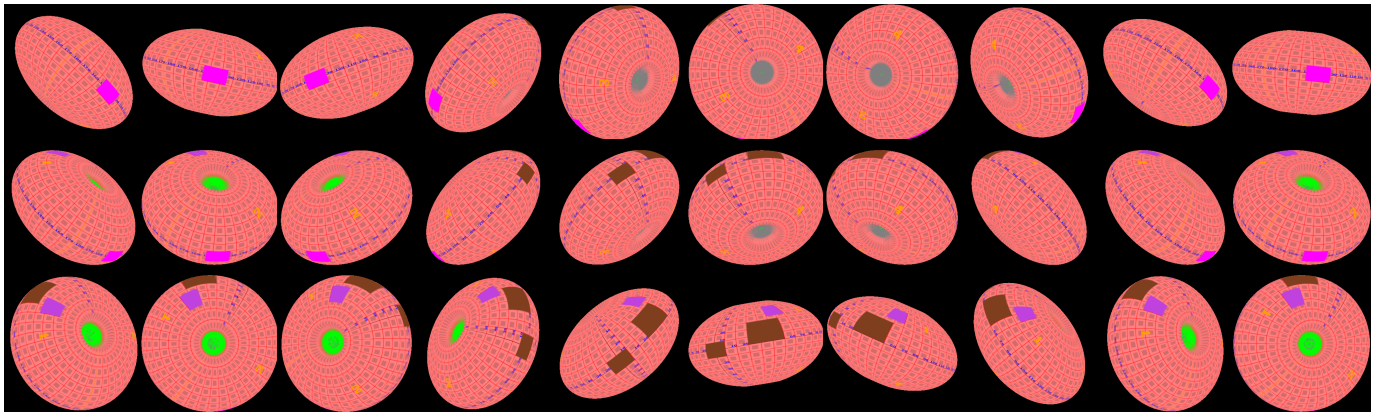


Fig. 21: Example of 30 images of the oblate spheroid synthetic, the images in the top row are taken with the right camera, the middle row is taken with the top camera and the bottom row is taken with the left camera.

TABLE VI: Results for YOLO v5n, v5s and v5m trained and tested on all images and image resolution 384.

Model	$\mathbf{P}_{\text{calyx}}$	$\mathbf{R}_{\text{calyx}}$	$\mathbf{mAP}_{\text{calyx}}$	$\mathbf{P}_{\text{stem}}$	$\mathbf{R}_{\text{stem}}$	$\mathbf{mAP}_{\text{stem}}$	**P**	**R**	**mAP**	**Inference time (ms)**
v5n	0.877	0.845	0.886	0.852	0.815	0.858	0.827	0.784	0.831	2.4
v5s	0.896	0.869	0.897	0.864	0.847	0.875	0.833	0.825	0.854	5.7
v5m	0.892	0.886	0.906	0.862	0.861	0.884	0.833	0.836	0.862	21.1

TABLE VII: Results for YOLO v5n for different image resolutions, trained and tested on all images.

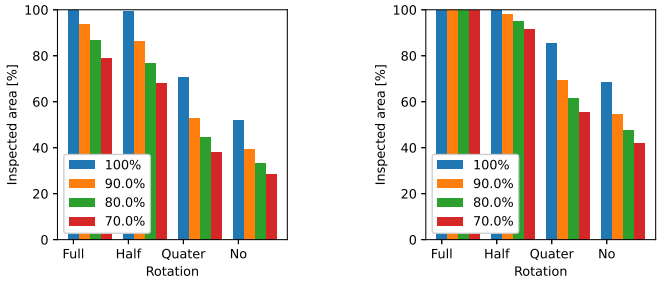
Image resolution	$\mathbf{P}_{\text{calyx}}$	$\mathbf{R}_{\text{calyx}}$	$\mathbf{mAP}_{\text{calyx}}$	$\mathbf{P}_{\text{stem}}$	$\mathbf{R}_{\text{stem}}$	$\mathbf{mAP}_{\text{stem}}$	**P**	**R**	**mAP**	**Inference time (ms)**
384×384	0.827	0.784	0.831	0.877	0.845	0.886	0.852	0.815	0.858	2.4
256×256	0.849	0.791	0.845	0.894	0.854	0.886	0.871	0.823	0.865	1.4
128×128	0.819	0.750	0.801	0.854	0.812	0.354	0.836	0.781	0.825	0.8

TABLE VIII: Results for YOLO v5n and an image resolution of 256×256 for different image types.

Image type	$\mathbf{P}_{\text{calyx}}$	$\mathbf{R}_{\text{calyx}}$	$\mathbf{mAP}_{\text{calyx}}$	$\mathbf{P}_{\text{stem}}$	$\mathbf{R}_{\text{stem}}$	$\mathbf{mAP}_{\text{stem}}$	**P**	**R**	**mAP**
All	0.849	0.791	0.845	0.894	0.854	0.886	0.871	0.823	0.865
Color	0.884	0.838	0.879	0.842	0.750	0.805	0.863	0.794	0.842
NIR	0.880	0.866	0.891	0.815	0.802	0.832	0.848	0.834	0.862
Soft	0.892	0.827	0.880	0.856	0.779	0.857	0.874	0.803	0.868

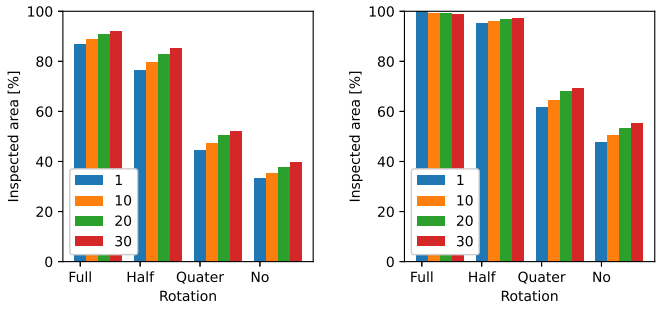
TABLE IX: Results for YOLO v5n when trained on all data of fruit specific data and validated on own data, with image resolution 256×256.

Fruit	Trained on	P_{stem}	R_{stem}	mAP_{stem}	P_{calyx}	R_{calyx}	mAP_{calyx}	P	R	mAP
Apple	All data	0.843	0.832	0.837	0.735	0.768	0.698	0.789	0.800	0.767
Apple	Apple	0.858	0.822	0.844	0.765	0.744	0.736	0.811	0.783	0.790
Blueberry	All data	0.754	0.730	0.729	0.872	0.945	0.963	0.813	0.838	0.846
Blueberry	Blueberry	0.802	0.720	0.747	0.901	0.894	0.955	0.852	0.807	0.851
Kiwi	All data	0.931	0.856	0.922	0.848	0.826	0.844	0.889	0.841	0.883
Kiwi	Kiwi	0.917	0.839	0.904	0.8545	0.861	0.8624	0.886	0.850	0.883
Lemon	All data	0.906	0.883	0.914	0.674	0.578	0.598	0.790	0.731	0.756
Lemon	Lemon	0.931	0.888	0.916	0.657	0.598	0.580	0.794	0.743	0.748
Lime	All data	0.857	0.858	0.854	0.704	0.735	0.678	0.781	0.797	0.766
Lime	Lime	0.946	0.872	0.913	0.679	0.664	0.593	0.812	0.768	0.753
Mandarin	All data	0.760	0.782	0.734	0.737	0.755	0.702	0.748	0.768	0.718
Mandarin	Mandarin	0.741	0.747	0.730	0.745	0.759	0.717	0.743	0.753	0.724
Pear	All data	-	-	-	0.922	0.885	0.904	0.922	0.885	0.904
Pear	Pear	-	-	-	0.920	0.920	0.932	0.920	0.920	0.932
Pomegranate	All data	0.961	0.952	0.976	0.964	0.993	0.990	0.962	0.972	0.983
Pomegranate	Pomegranate	0.960	0.941	0.974	0.971	0.988	0.993	0.966	0.965	0.984



(a) 1 camera. (b) 3 cameras.

Fig. 22: The results of the algorithm on sphere synthetic data for different radii of the grid.



(a) 1 camera. (b) 3 cameras.

Fig. 23: The results of the algorithm on synthetic sphere data for different grid resolutions using 80% of the radius.

2) Oblate spheroid model: The experiments preformed on the sphere synthetic data are repeated for oblate spheroid data. Figure 24 shows the result of the progress tracking algorithm for a different radii of the grid. The results are similar as the results of

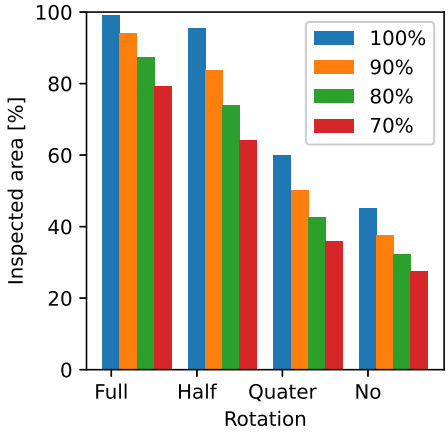


Fig. 24: The results of the algorithm on sphere synthetic data for different radii of the grid, using a grid resolution of 1 and one camera.

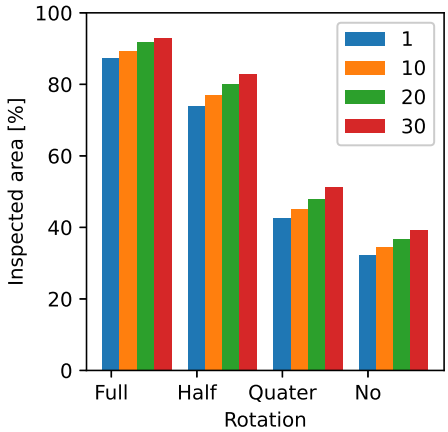
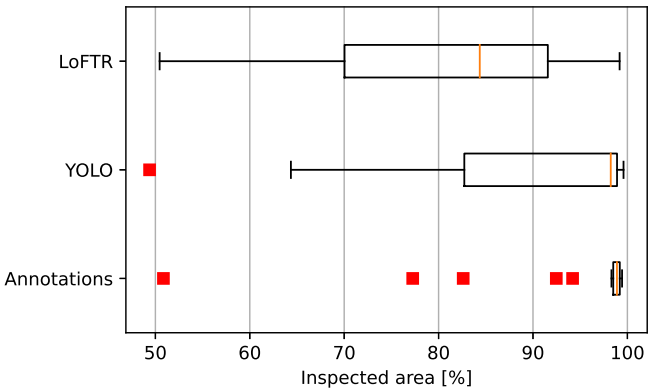


Fig. 25: The results of the algorithm on sphere synthetic data for different grid resolutions, using a grid of 80% the fruit radius and one camera.

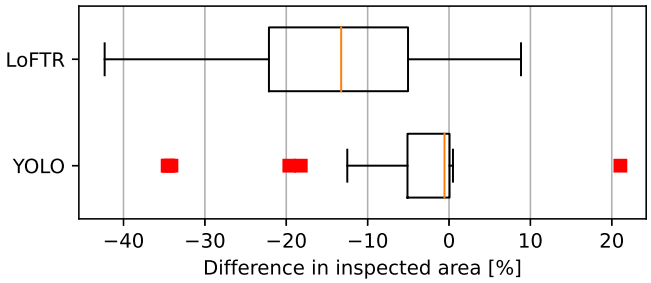
sphere synthetic data, as shown in Figure 22. Figure 25 shows the results of the progress inspection tracking algorithm for different grid resolutions. Similar as with the sphere synthetic data, a larger resolution leads to a overestimation of the area seen.

J. Results sphere model - pomegranates

Figure 26a shows the progress tracking algorithm results for the different methods. It shows that the algorithm based on annotations shows less variation in results and classifies almost all fruits as fully rotated, while the algorithms based on stem and calyx and LoFTR show a larger variation. Even though the algorithm based on stem and calyx rotation shows a larger variation in the results, the median is still close to the median of the annotation-based algorithm, while the median of the LoFTR-based algorithm is lower. Figure 26b shows the differences between the results of the algorithm based on annotations and stem and calyx and LoFTR. Here it can also be seen that the results of the algorithm based on stem and calyx deviate less from the results based on the manual landmark annotations. However, there are some large outliers, which can be due to the fruit rotating around the stem-calyx axes, as shown in Figure 19, this will lead to the stem and calyx being in the same place, showing no rotation if the algorithm based on the stem and calyx, while fully rotating if the rotation is based on manual landmarks.



(a) Results for the algorithm for the different methods.



(b) Difference between the annotated results and the results for the algorithm based on YOLO and LoFTR.

Fig. 26: The results of the algorithm on the test set of pomegranates, using 3 cameras, a grid size of 80% the fruit radius and a grid resolution of 10.

K. Elaboration on the grid resolution for blueberry

Our algorithm shows similar results for the spherical synthetic data set and the oblate spheroid data set. Which leads to the choice to use a grid resolution of 10 in order to increase the speed, while keeping the overestimation to a minimum. However, the resolution of the blueberry images is smaller, on average 119×117 pixels, than the image resolution of the synthetic data, which is on average 357×396 pixels. If in both cases a grid resolution of 10 is used, the blueberry grid will consist of less points, leading to a underestimation of the area seen. In order to check what grid resolution fits blueberries best, an experiment is preformed in which the inspected area is calculated after one image. Results show, as shown in Figure 27, that due to the smaller image resolution of blueberries, a smaller grid resolution is needed to obtain similar results as with synthetic data. Because of this we will use a image resolution of 1, for further experiments.

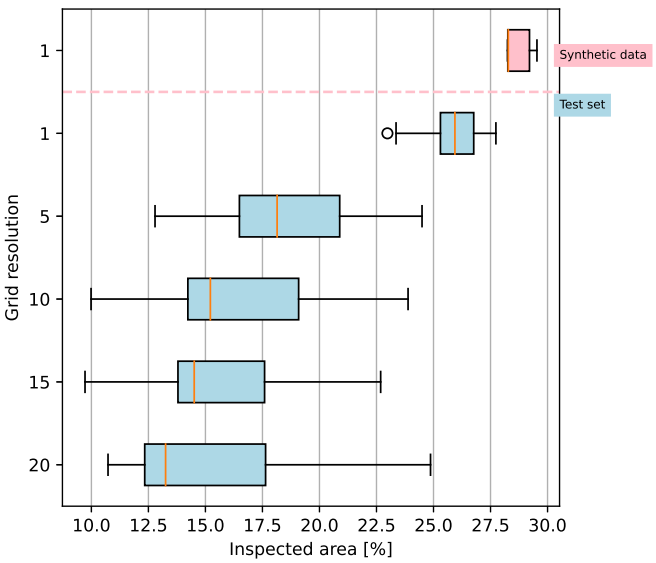


Fig. 27: Inspected area for one image of blueberries for different grid resolutions.