Ben-Gurion University of the Negev

Faculty of Engineering Sciences

Department of Industrial Engineering and Management

Adaptive thresholding algorithm for robotic fruit detection

By: Elie Zemmour

Advisor: Prof. Y. Edan

August 2018

ACKNOWLEDGMENTS

This research was partially supported by the European Commission (SWEEPER GA no. 66313) and by Ben-Gurion University of the Negev through the Helmsley Charitable Trust, the Agricultural, Biological and Cognitive Robotics Initiative, the Marcus Endowment Fund and the Rabbi W. Gunther Plaut Chair in Manufacturing Engineering.

I wish to express my sincere gratitude to my academic advisors, Prof. Yael Edan and Ms. Polina Kurtser, who provided me guidance, support and insights over the course of the research.

Finally, I would like to express my deepest gratitude to my friend, Elad Dan, for support, encouragement and for sharing with me his vast knowledge throughout my studies and research.

Thank you all.

Abstract

This research presents an automatic parameter tuning process for a dynamic adaptive thresholding algorithm for fruit detection. The algorithm enables robust detection in highly variable lighting conditions. The image is dynamically split into variable sized regions, where each region has approximately homogeneous lighting conditions. Nine thresholds were selected to accommodate three different illumination levels for three different dimensions in four color spaces: RGB, HSI, NDI and LAB. The thresholds were selected by quantifying the required relation between the true positive rate and false positive rate. A tuning process was developed to determine the best fit values of the algorithm parameters to enable easy adaption to different fruits, colors and illumination conditions. Extensive analyses were conducted on three different databases: red apples (9 images of 113 apples), green grapes (129 images of 1078 grapes) and yellow peppers (30 images of 73 peppers) acquired in outdoor conditions. Results show the importance of the tuning process for the generalization of the algorithm to different kinds of fruits and environments. In addition, this research revealed that for each kind of fruit the use of a different color space might be superior over the others.

Keywords: Adaptive thresholding, Fruit Detection, Robotic harvesting, Parameter tuning

Publications

Journal papers under review

- Zemmour, E., Kurtser, P. and Edan, Y. Automatic parameter tuning for adaptive thresholding in robotic fruit detection. Submitted July, 2018. (Appendix B).

Conference papers and presentations

- Zemmour, E., Kurtser, P. and Edan, Y., 2017, April. Dynamic thresholding algorithm for robotic apple detection. In Autonomous Robot Systems and Competitions (ICARSC), 2017 IEEE International Conference on (pp. 240-246). IEEE. (Appendix A).

Abstracts

- Zemmour, E., Kurtser, P. and Edan, Y., 2017, September. Dynamic thresholding algorithm for robotic apple detection. ISAE the Israeli Conference on Agricultural Engineering, Bet-Dagan. http://isae.org.il/

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1. Problem description

Object detection is an important and essential task in many agricultural applications including autonomous navigation and obstacles avoidance (Loianno et al., 2018), precision and selective spraying (Tona et al., 2018), weed control (Albert and Michaels, et al., 2017), yield estimation (Liu et al., 2017), ripeness and quality evaluation (Pereira et al., 2018), phenotyping (Ghosal et al., 2018) and fruit detection for robotic harvesters (Almendral et al., 2018; Kapach et al., 2012; Gongal et al., 2015, Bac et al., 2014).

Despite intensive research conducted in identifying fruits, implementing a real time vision system remains a complex task (Gongal et al., 2015;Kapach et al., 2012). Current detection is limited to 87-88% detection rate with 3.8% false alarms (Luo et al., 2018; Vitzrabin and Edan, 2016; Bac et al., 2014) in robotic harvesting.

Features such as shape, texture and location, are subject to high variability in the agricultural domain (Gongal et al., 2015). Moreover, fruits grow in an unstructured environment with highly variable lighting conditions (Vitzrabin and Edan, 2016) and obstructions (Barth et al., 2016) that influence detection performance. Color and texture are fundamental characteristics of natural images and play an important role in visual perception (Arivazhagan et al., 2010).

Images can be represented by different color spaces (e.g., RGB, HSI, LAB, NDI); each one emphasizes different color features (Arivazhagan et al., 2010). RGB is the most common color space, representing each pixel in the image in three color channels as acquired: red, green and blue. HSI represents every color with three components: hue (H), saturation (S) and intensity (I), also known as HSV (Zheng et al., 2009). LAB color space is an approximate of human vision (Shmmala and Ashour,2013). An additional color space commonly employed in the agriculture field (Vitzrabin and Edan, 2016) is the normalized difference index (NDI) space. The NDI is used to differentiate between fruit and the background (Woebbecke et al., 1992) since it helps to overcome changes in illumination and shading due to its normalization technique (Shrestha, 2014). Each dimension in the NDI space is the normalized difference index between 2 colors in the RGB space, resulting in three dimensions (Equation 1). These operations are applied for all pixel locations in the image, creating a new image with this contrast index. These equations yield NDI values ranging between -1 and +1.

$$NDI_1 = \frac{R-G}{R+G}\ ; NDI_2 = \frac{R-B}{R+B} ; NDI_3 = \frac{B-G}{B+G} \qquad (1)$$

One of the most common methods for fruit detection is image segmentation (e.g., Wang et al., 2013; Jiang et al., 2008; Arroyo, et al., 2016; Rong, et al., 2017). Many segmentation algorithms have been developed (Zhang, 1996) including: Kmeans (Shmmala and Ashour, 2013); Mean shift analysis (Zheng et al., 2009); Artificial neural networks (ANN) (Al-allaf, 2014), Support vector machines (SVM) (Sakthivel et al., 2015), Deep learning (Sa et al., 2016), Reinforcement learning (RL) (Ostovar et al., 2018) and several others.

This research focuses on segmenting objects in the image using an adaptive thresholding method. Observing the histogram of the image color implies that a threshold can be determined to best differentiate between the background and the object distributions (Park et al., 2011). The threshold is computed by finding the histogram minimum (Figure 1) separating between two peaks – the object and the background. However, the global minimum between the distributions is very hard to find in most cases (Hannan et al., 2007).

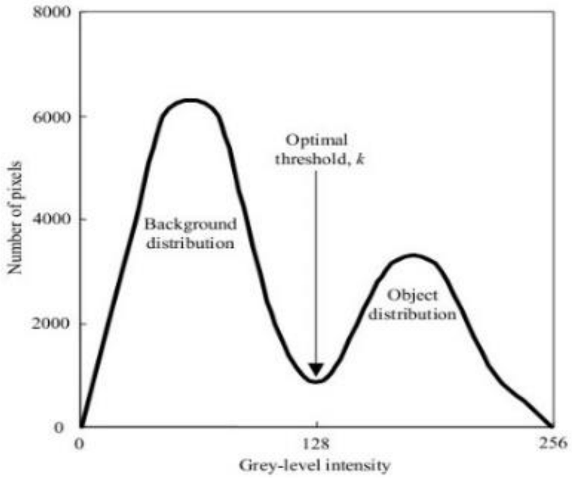


FIGURE 1: OPTIMAL THRESHOLD IN BIMODAL HISTOGRAM

Current most optimal thresholding algorithms determine the threshold only in a one-dimensional space, for example in the RGB space, either R or G or B or a linear combination of their values (e.g. grayscale transformation) will be used (Bulanon et al., 2001). In the transformation from three dimensions into one, information is lost. In this research, a three-dimensional thresholding algorithm based on (Vitzrabin and Edan, 2016) was applied and evaluated also for additional color spaces (RGB, NDI , HSI and LAB color spaces) – a threshold is determined for each dimension in the color space.

There are two common adaptive thresholding algorithm concepts: 1) global thresholding, in which for each image, a different threshold is determined according to specific conditions for the entire image that is then transformed into a binary image; 2) local thresholding, in which the image is divided into sections, and a different threshold is calculated for each section; the sections are then combined to a binary image.

There are several methods that utilize dynamic local thresholding algorithms (Gunatilaka and Baertlein, 2001; Kanungo et al., 2010). A common approach is to use multi-resolution windows which apply a bottom up method, merging pixels while a criteria is met (Gunatilaka and Baertlein 2001; Kanungo et al., 2010). Another approach is the top down method, where the image is divided into sub regions according to criteria. The top down approach reduces execution speed and improves generalization (Hall and McMullen, 2004) and was therefore used in this research.

The adaptive thresholding algorithm presented in this research is based on previous work (Vitzrabin, 2016) that aimed to detect peppers, in which a set of three thresholds were determined for each region of the image according to its lighting setting. The algorithm dynamically divides the image into several regions, each with approximately the same lighting conditions. The main contribution of the adaptive local 3D thresholding is a very high true positive rate (TPR) and low false positive rate (FRP) in the fruit detection task in an unstructured, highly variable, and dynamic crop environment. Another contribution is the ability to change in real time the task objective in the algorithm behavior based on the desired ratio between TPR and FRP; this contributes to a better success rate in the grasping operation itself (Vitzrabin and Edan, 2016). Both the high performances and the changing task objectives are key issues regarding detection in robotic harvesting applications (Vitzrabin and Edan, 2016). This thesis advances the previous research with specific contributions as noted below.

1.2. Objectives

Current research objectives and contributions:

- A new parameter tuning process developed to best-fit the parameters to the specific database.
- Intensive evaluation of the adaptive thresholding algorithm for different color spaces.
- Application and evaluation of the algorithm to different kinds of fruit.

Literature Review

This chapter presents scientific background concerning the research related to object detection, adaptive thresholding algorithms, color spaces used for image representation and robotic applications in agriculture.

2.1. Object detection using computer vison

Object detection is an important and essential task in many agricultural applications including autonomous navigation and obstacles avoidance (Loianno et al., 2018), precision and selective spraying (Tona et al., 2018), weed control (Albert and Michaels, et al., 2017), yield estimation (Liu et al., 2017), ripeness and quality evaluation (Pereira et al., 2018), phenotyping (Ghosal et al., 2018) and fruit detection for robotic harvesters (Almendral et al., 2018; Kapach et al., 2012; Gongal et al., 2015, Bac et al., 2014). This research focuses on vision for stationary cameras and "eye-in-hand" cameras for robotic harvesters (Figures 2,3) in outdoor conditions.

Usually an image includes objects of interest and a background represented by anything else in the image. Objects detection and recognition in images are significant tasks in computer vision field (Jalled and Voronkov, 2016).

Computer vision aims to imitate human vision by perceiving and understanding an image or a video through pixels and their relations using: artificial intelligence, neurobiology, signal processing and more. Although vision seems natural for humans, image interpretation is a challenging task for computers. The transformation of the world from 3D to 2D causes loss of information (e.g. geometric structure, distance between objects). In addition, while measuring the real world (using images and videos), vision systems deal with a large amount of noise caused by physical conditions such as brightness, shadow area or measurement errors. Lastly, analyzing the image pixel by pixel makes hard for a computer to understand the whole picture (Schalkoff, 1989).

Despite all the challenges mentioned above, computer vision systems are used today in a varied field of application including face detection (Li et al., 2015), autonomous cars (Alhaija et al., 2017) and many more.

Computer vision applications include image acquisition , image processing and analysis interpretation (Nuske et al., 2014).

2.1.1. Image acquisition

First step in image acquisition process deals with capturing the image data using a camera. The image is represented by a matrix of discrete picture elements (i.e. Pixels) (Szeliski, 2010). The pixels values represent the light intensity in the image. These values are converted into digital value by an analog digital converter. The digital value depends on the number of bits used in the vision system (Nuske et al., 2014). For instance, binary vision presents the light intensity using only two values: zero for black pixels and one for white pixels (Sonka et al., 2014). Another common image representation is the greyscale vision system showing different shades of grey according to the number of bits in each pixel (Nuske et al., 2014). More advanced vision systems can represent color. This method evaluates for each pixel three number in range [0,256], each one represents a different color intensity red, green or blue (Mohanty et al., 2016)

2.1.2. Image processing and analysis

Processing and analyzing the image are the following steps of object detection process. A variety of techniques have been developed to process and understand the image data. A common method used for object detection is segmentation (e.g., Rong, et al., 2017; Arroyo, et al., 2016; Wang et al., 2013; Jiang et al., 2008). Segmentation is a process of partitioning the image into different objects or connected regions that do not overlap (Jameel and R. Manza, 2012).

Image segmentation algorithms are based on either the discontinuity principle or the similarity principle (Haward, 2016). The discontinuity principle consists in extracting regions that differ in properties such as intensity, color, texture, or any other image statistics. The similarity principle consists in grouping pixels based on a common property (Sonka et al., 2014). Thresholding is an example for segmentation that uses the discontinuity principle.

Thresholding, one of the simplest methods for image segmentation, usually uses the grayscale image to create a binary image that represents objects vs. background in the image (Shapiro et al. 2001). The thresholding method assumes that the density color of the object can be differentiated from the density color of the background (Figure 1). In this method, a histogram is created based on the image color. In a second step the method tries to find the local minimum between the density of the background and the density of the object. This value is set as the threshold value. Using the threshold value, pixels are categorized as white (part of the fruit) or black (representing the background) (Figure 4).

General computer vision literature indicates that predefined global thresholding fails in most scenarios (Haralick et al., 1985; Nalwa, 1993; Zhang et al., 2010). To overcome the problem, several vision applications apply adaptive thresholding algorithms (reviewed in section 2.2) where the threshold adaptively changes to the new illumination conditions (Boulmerka et al., 2014; Hannan et al., 2007).

The outcome of the segmentation procedures is a binary image creating similar regions which helps to define and identify in the image (Sonka et al., 2014). Features extraction process is following the segmentation process . Features can be used by a machine vision algorithm to define an object in the image (Nuske et al., 2014). Some of the features extracted from the segmented image are simple features such as area, width and length and others are more complex, such as center of gravity, shape and aspect ratio (Sonka et al., 2014). The combination of a few features together usually describes the object and helps to recognize it and interpret the image (Szeliski, 2010).

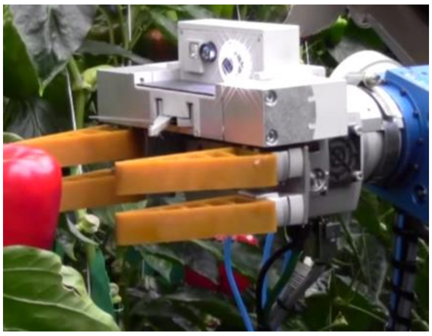
Many techniques have been developed for feature extraction process in computer vision field (Lowe, 1999; Harris and Stephens, 1988) and more modern algorithms have been created such as deep convolutional neural networks (Ruiz-del-Solar et al., 2018).

2.1.3. Image interpretation

The image interpretation process is the final step of the object detection. Based on the extracted features, machine vision algorithms can define objects in the image (Nuske et al., 2014). Usually, this process uses predefined models or values to recognize the objects. A common method is called Template Matching. In this method the system compares the extracted features to a predefined model template that represents object in images (Nuske et al., 2014). Today, advance techniques emerge using machine learning, artificial neural networks (Ruiz-del-Solar et al., 2018) and more.

This research focuses on the detection process using adaptive thresholding methods.

.



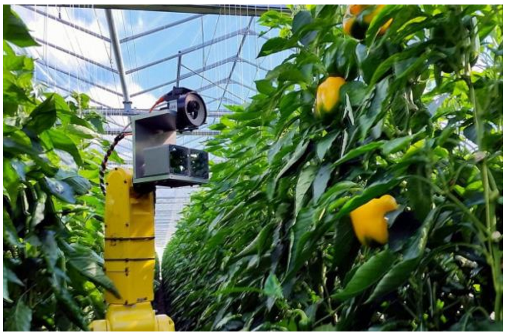


Figure 2: "eye in hand" camera

Figure 3: Stationary camera

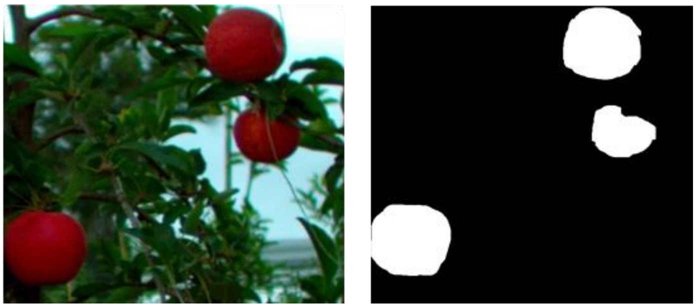


Figure 4: Binary segmentation

2.2. Adaptive thresholding algorithms

Changing lighting conditions in outdoor environments, due to sunlight direction and shadowed areas (caused by object or clouds) impact the robotic vision (Figure 5,6). Several object detection algorithms used thresholds derived specially for images with high variability (Aufrère et al., 2000; Rasolzadeh et al., 2010; Rakun et al., 2011). In addition, target detection algorithms were developed for determining thresholds for images with high variability (Reibman et al., 1987; Wuhib et al., 2008; Rosin et al., 2003; Kong et al., 2010). Another approach to solve the high variability problem is to assign a different threshold for each sub-image (Kong et al., 2010; Kanungo et al., 2010;Shih et al., 2005; Revol-Muller et al., 2002).

Two common adaptive thresholding algorithm approaches are used for object detection:

- **Adaptive global thresholding -** for each image a different threshold is determined according to specific conditions for the entire image, which is then transformed into a binary image.

- **Adaptive local thresholding -** the image is divided into sections and a different threshold is calculated for each section. The division to section can be conducted using different rules (e.g. minimizing stand deviation or entropy on each section). Finally, the sections are combined to a binary image.

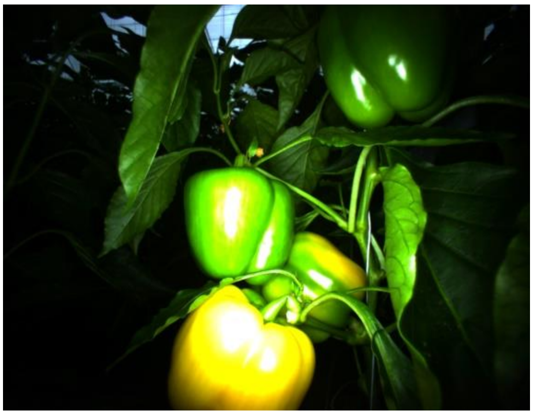


FIGURE 6: CHANGING LIGHTING CONDITIONS

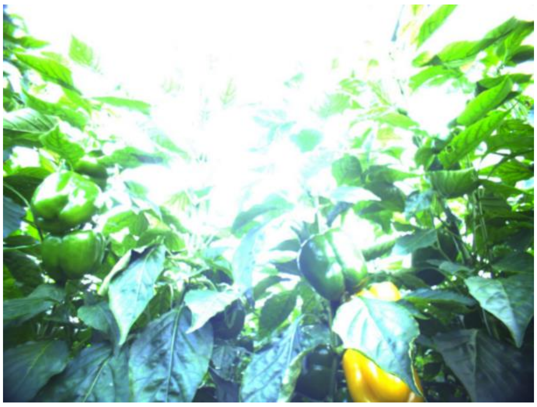


FIGURE 5: SHADOWED AREAS

In this research we adapted a previous developed thresholding algorithm (Vitzrabin, 2016) that adapts to different illumination conditions within and between images. It dynamically applies a local threshold to different areas in the image.

2.3. Color spaces

Color models, like all mathematical representations of physical phenomena, can be expressed in many ways, each with its advantages and drawbacks. Some representations are formulated to help humans select colors and others are formulated to ease data processing in machines using various color space. Historically, whatever the meaning assigned to the color space variables, three of them were enough to describe all colors: Red-Green-Blue (RGB), Hue-Saturation-Brightness (HSB), L*a*b* etc. (Pascale, 2003). This chapter covers RGB, HSI, NDI and LAB color spaces.

RGB - The RGB color space differentiate between Red Green and Blue colors in the image (Figure 7). Each dimension is represented by a value range 0-255. The RGB color space is one of the most commonly used image representation (Pascale, 2003; Shmmala and Ashour, 2013).

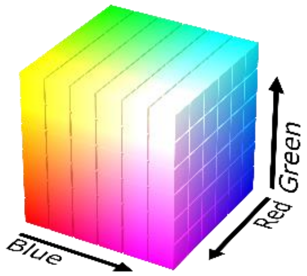


FIGURE 7: RGB COLOR SOLID CUBE

NDI - Each dimension in the NDI space is the **normalized difference index** between 2 colors (e.g. NDI first dimension represents the difference between red and green in RGB space, (Equation 1 in Section 1); second dimension represents the difference between red and blue in RGB space) as aforementioned, resulting in three dimensions. These operations are applied for all pixel locations in the image, creating a new image with this contrast index. These equations yield NDI values ranging between -1 and +1.

Using the NDI space in the dynamic thresholding process instead of the grey scale image enables us to learn 3D information about the threshold that best distinguishes between the background and the fruit (Woebbecke et al., 1992).

HSI - The HSI color space (Hue, Saturation and Intensity) defines a model in terms of its components. This color space is recommended for processing images when they are affected by lighting changes. HSI space can separate the intensity of the intrinsic color information, which would refer to the hue and saturation (Gasparri et al .2011).

HSI space representation is through a double cone, as shown in Figure 8. The center of this double cone is a circumference divided into angles of equal magnitude (Gasparri et al .2011) which range is [0,2π] comparative to angle 0 at red axis, 2 π/3 at green axis, 4π/3 at blue axis and red again at 2π (Shmmala and Ashour, 2013).The distance from the center of the exterior circumference represents the saturation found in every color and takes values from 0 to 1, indicating how the color is diluted with white light. Finally, the axis through the two cones corresponds to the intensity component. This has a normalized value from 0 (black) to 1 (white) and indicates the amount of light in a color. Removing a small circumference of the figure formed by two cones, colors close to an intensity of 1 are lighter than those close to zero. When the saturation component is close to 0, colors only reflect a change between black and white. When this component is close to 1, the color will reflect the true value represented by the hue (Gasparri et al .2011).

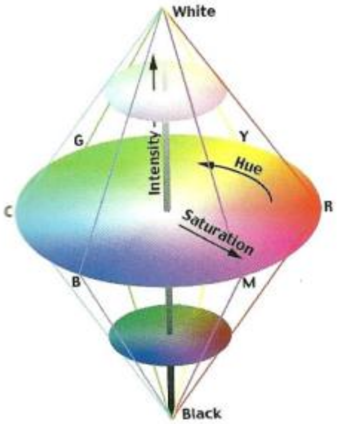


FIGURE 8: HSI COLOR SPACE REPRESENTATION

LAB - The Lab color space describes mathematically all perceivable colors in the three dimensions: L represents lightness while a and b represent the color opponents green–red and blue–yellow.

One of the most important attributes of the Lab model is device independence. This means that the colors are defined independent of their nature of creation or the device they are displayed on.

The LAB space is a three-dimensional real number space that contains an infinite number of possible representations of colors. However, in practice, the space is usually mapped onto a three-dimensional integer space for device-independent digital representation, and for these reasons, the L*, a*, and b* values are usually absolute, with a pre-defined range.

The lightness, L*, represents get a range from 0 (darkest – black) to 100 (brightest-white). The color channels, a* and b*, represent true neutral gray values when a* = 0 and b* = 0. The red/green opponent colors are represented along the a* axis, with green at negative a* values and red at positive a* values. The yellow/blue opponent colors are represented along the b* axis, with blue at negative b* values and yellow at positive b* values. The scaling and limits of the a* and b* axes will depend on the specific implementation of Lab color, but they often run in the range of ±100 or −128 to +127 (Shmmala & Ashour, 2013).

2.4. Robotic applications in agriculture

There is an increasing need for robots in agriculture to respond to the increasing demand of food at a competitive price and ensure high quality food (Jukema and Van de Meer 2009).

Robotic applications can improve yield and agriculture quality by enabling better control over environmental implications. For instance, extracting information such as leaf measures, steam properties, climate condition and yield data per plan helps to take appropriate measures in a proactive approach. In addition, agricultural robots can reduce manual labor and production costs (Edan and Miles, 1994; Kapach et al., 2012). Robotic applications enable new functionality using sensing abilities that obtain better performances than humans in accuracy and consistency (Jukema and Van de Meer 2009). For example, agricultural applications can help in detecting fruits diseases. By enabling focused treatment on a specific area, the use of chemicals can be decreased (Gorbe and Calatayud 2012).

During the past 30 years, robotics applications in agriculture has been a vast research subject (Sarig, 1993). Many robotic systems were developed for different agricultural applications such as transplanting, cultivating, spraying, trimming and selective harvesting (Edan and Miles, 1994; Kapach et al., 2012).

However, agriculture automated system has yet been commercialized for harvesting fruit crops (Gongal et al. 2015). The main reason for low performances of harvesting robots are the complexity of the corps and environment (Henten, C. Wouter Bac and Eldert J. van, Jochen Hemming, 2014). Fruits inherent high variability in size, shape, texture, and location (Figure 9) (Kapach et al., 2012). In addition, unstructured, uneven illumination, and dynamic nature of the environment makes harvesting robot real time system implementation in an outdoor environment is still a complex problem (Figures 3,4) (Bulanon et al. 2002; Kapach et al. 2012; Bac et al. 2014).

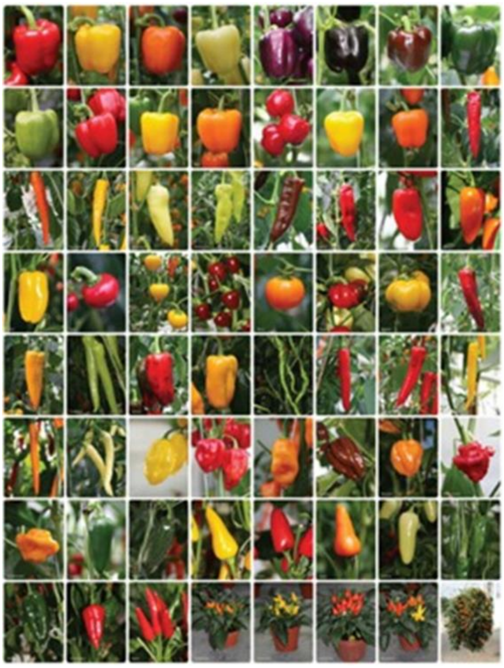


FIGURE 9: HIGH VARIABILITY IN SIZE SHAPE COLOR AND LOCATION

3. Research methods

3.1. Overview

This research is based on a dynamic thresholding algorithm, which incorporated changing task objectives algorithm to improve sweet pepper detection for a robotic harvester (Vitzrabin & Edan, 2015). The algorithm was implemented in Matlab 2015a following the flow presented by (Vitzrabin & Edan, 2015). The algorithm includes an offline process (train), in which the algorithm learns color thresholds for detection followed by an online process responsible for implementing the thresholds learned in the offline process on a new image, in order to detect pixels that represent fruit in real time (Vitzrabin & Edan, 2015) . Previous analyses indicated the superiority of the dynamic thresholding algorithm (Vitzrabin & Edan, 2015) as compared to traditional algorithms such as constant threshold, bimodal histogram and a constant threshold 3D thresholding.

The current developments include:

1. Generalizing the algorithm by applying different color spaces (RGB, NDI, HSI, LAB) to enable robust detection for a wide range of fruit varieties.
2. Developing a tuning process to enable automatic and improved fruit detection.

3.2. Assumptions

- Different objects in the images have different color densities that can be differentiated by local minimum threshold between them.
- The images contain fruits and background.
- Image light level can be represented by computing the average pixel values in the grayscale image.

3.3. Algorithms

As a first step, the previously developed algorithm (Vitzrabin & Edan, 2015) was implemented. Then, the algorithm performances were improved for apple detection by omitting the fusion module which was specific for this database and developing a dynamic use of the NDI color dimensions. The third step in the algorithm development included extension to a multitude of color spaces used for segmentation (RGB, HIS, and LAB). Finally, a dynamic parameter tuning module was developed to ensure employment of best-fit parameters for the algorithm initialization when changing the detection task (detection of different kind of fruit and/or different environment).

3.4. Databases

The algorithms were evaluated on three databases representing three different fruit colors: red (apples), green (grapes) and yellow (peppers) for two environmental settings (greenhouse, field) in different illumination conditions. Images were acquired with different cameras.

Apples - The orchard apples database includes 113 "Royal Gala" apples in 9 images acquired from an orchard in Chile in March 2012 under natural growing conditions with a Prosilica GC2450C camera with 1536x2048 resolution; the camera was attached to a pole. The images were captured in daylight: half of the images were acquired under direct sunlight, and half of the images were acquired in the shade. Ground truth was manually marked (Figure 10).

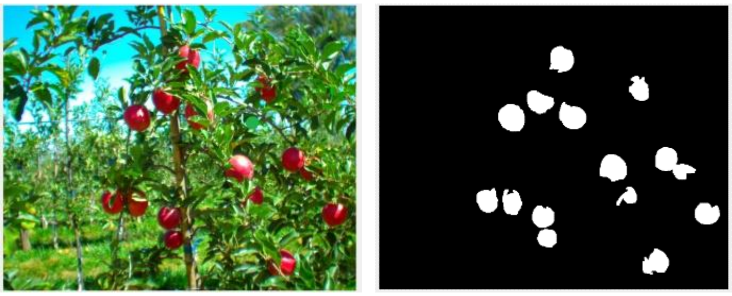


Figure 10: Apples image RGB image (left) and ground truth (right) example

Grapes - The images used originated from a commercial vineyard growing green grapes of the “superior” variety. An RGB camera (Microsoft NX-6000) with 600×800 resolution was manually driven at mid-day along a commercial vineyard in Lachish, Israel, during the summer season of 2011, one month before harvest time. The images were captured from 5 different growing rows. The targets were defined as the grape clusters. A group of three experts was guided to mark the closing perimeter of each grape cluster in the image. The final ground truth was marked using the judge rules criteria (if a given pixel was marked by two or more experts, it was considered a target). A set of 129 images were marked using this technique and used as a ground truth for the following research. The images included 1078 grape clusters (Figure 11).

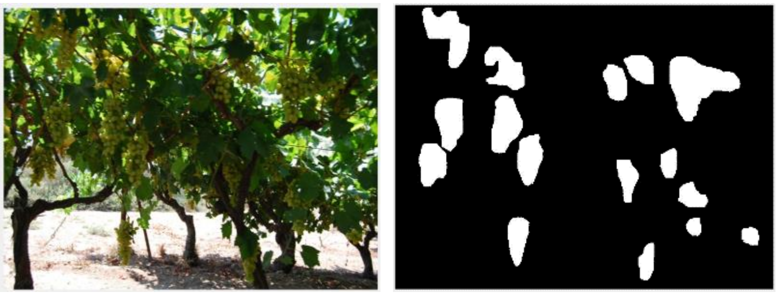


Figure 11: Grapes image RGB image (left) and ground truth (right) example

Peppers – The dataset includes 30 images of 73 yellow peppers acquired in a commercial greenhouse in Ijsselmuiden, Netherlands using a 6 degree of freedom manipulator (Fanuc LR Mate 200iD/7L), equipped with an iDS Ui-5250RE RGB camera with 600x800 resolution. The images were manually marked in order to compare the algorithm performances to manual detection by a human labeler. The database was marked twice, one time marking only peppers with high visibility (denoted as "high visibility peppers", this was done for 10 images of 25 yellow peppers) and a second time marked as well peppers in dark areas that are less visible in the image (will be refer as "including low visibility peppers", done for all 30 images) (Figures 12,13).

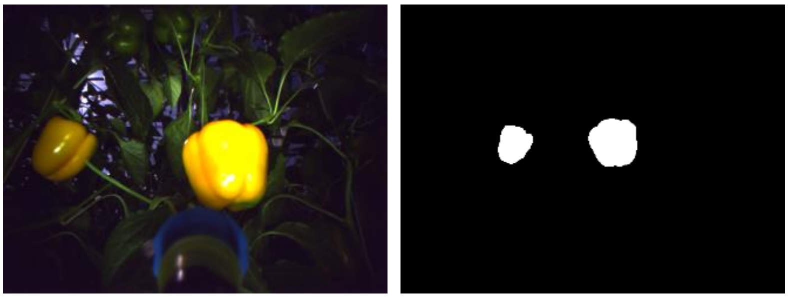


Figure 12: Peppers RGB images (left) and ground truth (right) example

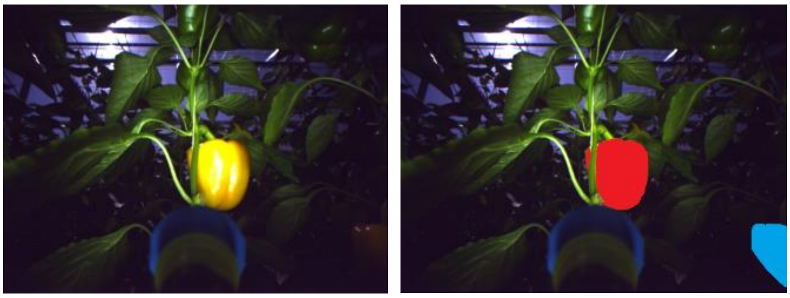


Figure 13: RGB image (right) and labeled image (left)

"High visibility peppers" marked in red and "Low visibility peppers" marked in blue

Note that ‘imclose’ MATLAB function was used (with disk=5) when converting the ground truth to binary images as follows to remove numbers that were written on images when tagged (Figure 14).

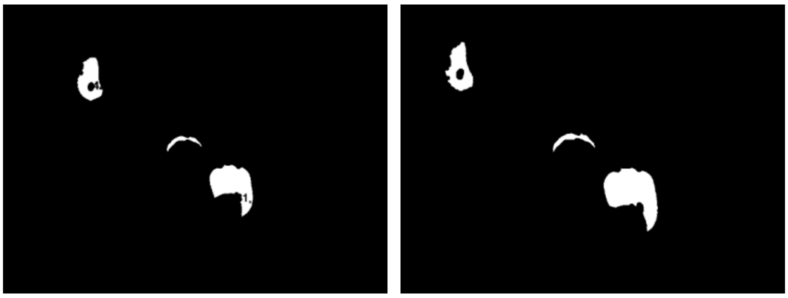


Figure 14: Binary image before “imclose” function (left) and after (right)

3.5. Analyses

The following analyses were conducted for the three databases, apples, grapes and peppers, using 70% of the data for training and 30% for testing (Guyon & Isabelle, 1997). This rate was chosen to be more rigid to the algorithm performances since the number of images in each DB were relativity small. In addition, to ensure robustness of the results each test was repeated five times with random selection of images for each repetition and average results of all performance measures were reported.

- **Tuning parameters**: parameters were computed for each database with procedures defined in section 3.3 and compared to previous predefined parameters.
- **Color spaces analyses**: algorithm performances were tested on all databases for four different color spaces: NDI, LAB, HSI, RGB.
- **Sensitivity analyses:** sensitivity analyses were conducted for all the databases and included:

  a) Noise - noise was created by adding to each pixel in the RGB image a random number from the mean normal distribution for noise values up to 30%. The artificial noise represents the algorithms robustness toward other cameras with more noise, or when capturing images with different camera settings. Noise values of 5%, 10%,20%,30%, were evaluated.

  b) Thresholds learned in offline process - thresholds were changed by ±5%, ±10% and ±15% according to the threshold in each region.

  c) Stop condition - the selected std value was changed by 5%, 10% to test the robustness of the algorithm to these parameters.

  d) Train vs. Test -The algorithm performances were evaluated while using different percentage of DB images for the training and testing processes.

- **Morphological operation contribution –** performances were tested for imaging with and without the morphological operations process.

3.6. Performance measures

Performance measures included TPR (true positive rate, also noted as hit), FPR (false positive rate, also noted as false alarms) and F score (Goutte & Gaussier, 2005). The TPR metric (Equation 2) states the number of correctly detected objects relative to the actual number of objects, while the FPR metric calculates the number of false objects detected relative to the actual number of objects (Equation 3). The F score (Equation 4), balances between TPR and FPR equally.

$$\textbf{TPR}= \frac{pixels\ detected\ correctly\ as\ part\ of\ fruit}{Actual\ number\ of\ pixels\ that\ represent\ the\ fruit} \qquad (2)$$

$$\boldsymbol{FPR} = \frac{false\ detected\ pixels}{Actual\ number\ of\ pixels\ that\ represent\ background} \qquad (3)$$

$$\boldsymbol{F(TPR, FPR)} = \frac{2*\left(TPR*(1-FPR)\right)}{TPR+(1-FPR)} \qquad (4)$$

4. Algorithm

4.1. Algorithm flow

The RGB images are the inputs for the offline process. Some areas in the images contain more illumination than others, depending on the position of the light source and shading caused by leaves, branches and the covering net when exists. To overcome this issue, the algorithm divides each image into multiple sub images, with approximately homogenous illumination conditions (Figure 15). These sub images are categorized into three illumination conditions: low, medium, and high. The illumination level is obtained by calculating the average on the gray scale sub images. The gray scale image shows values between 0 (completely dark) and 255 (completely white). In the previous algorithm (Vitzrabin and Edan, 2016), the sub images were categorized into groups using levels selected empirically as 10, 70, and 130, corresponding to low, medium, and high level images based on manual image analyses. The high value was set as 130 in order to filter overexposed areas in the images. In the current algorithm a tuning parameter process (detailed in section 3.3) is developed to determine these three values.

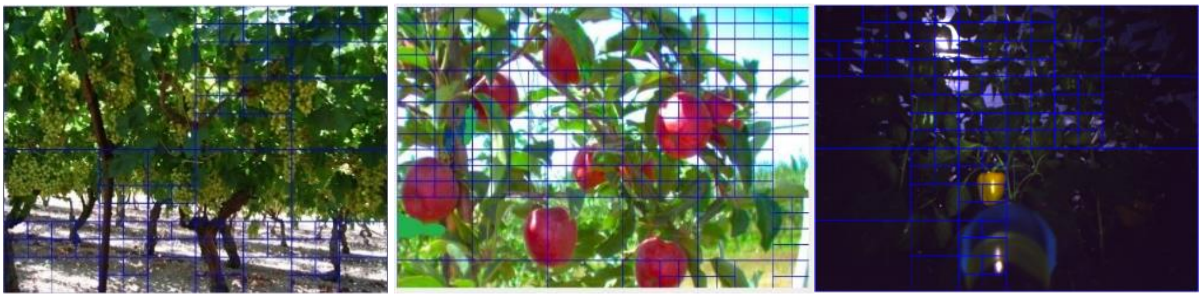


FIGURE 15: IMAGE SPLIT INTO SUB-IMAGES - VISUALIZATION

The algorithm then creates a 3D color space image (can transform the RGB image to NDI, HSI, LAB space or uses directly the RGB space). For each color dimension a binary image (mask) is created, where each pixel that represents the fruit receives a value of one and all other pixels receive a value of zero (Figure 16). Finally, the algorithm creates a ROC (receiver operator characteristics curve) representing TPR as a function of FPR (Siegel & Wu, 2003) representing all the nine thresholds learned from the offline process. Figure 17 presents an example of nine ROC curves computed for three sub images with different light levels (L1, L2, L3) in the NDI color space. In this example, the sub image with light level 2 (L2) in the first NDI dimension obtained the best performances (high TPR and low FPR).

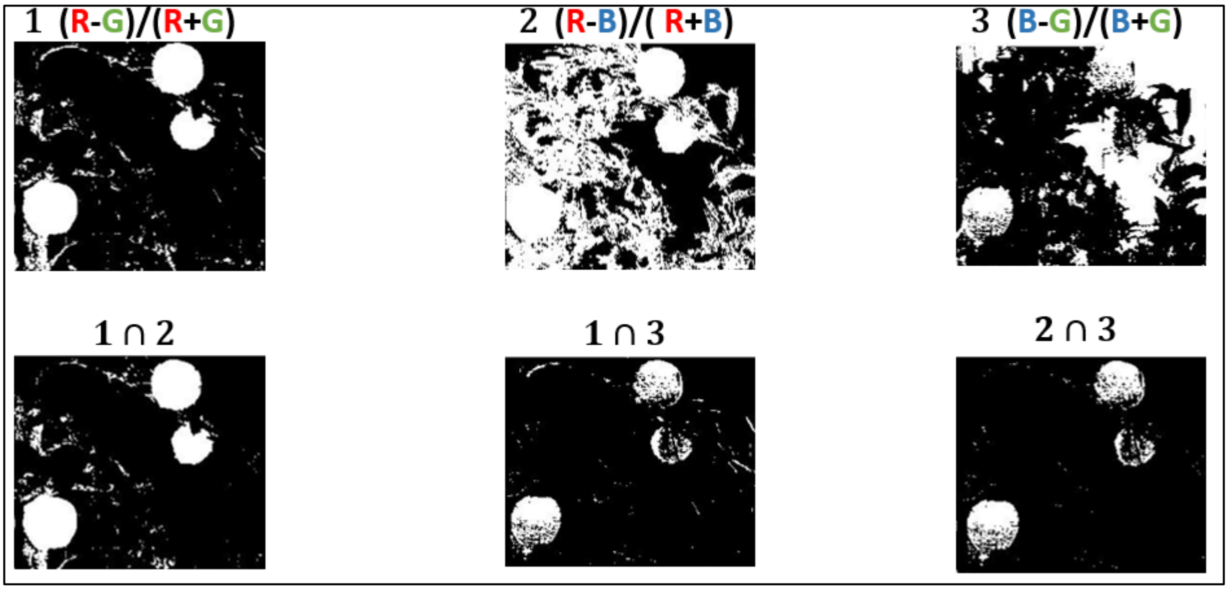


FIGURE 16: USE OF NDI DIMENSION INTERSECTION TO INCREASE PERFORMANCE

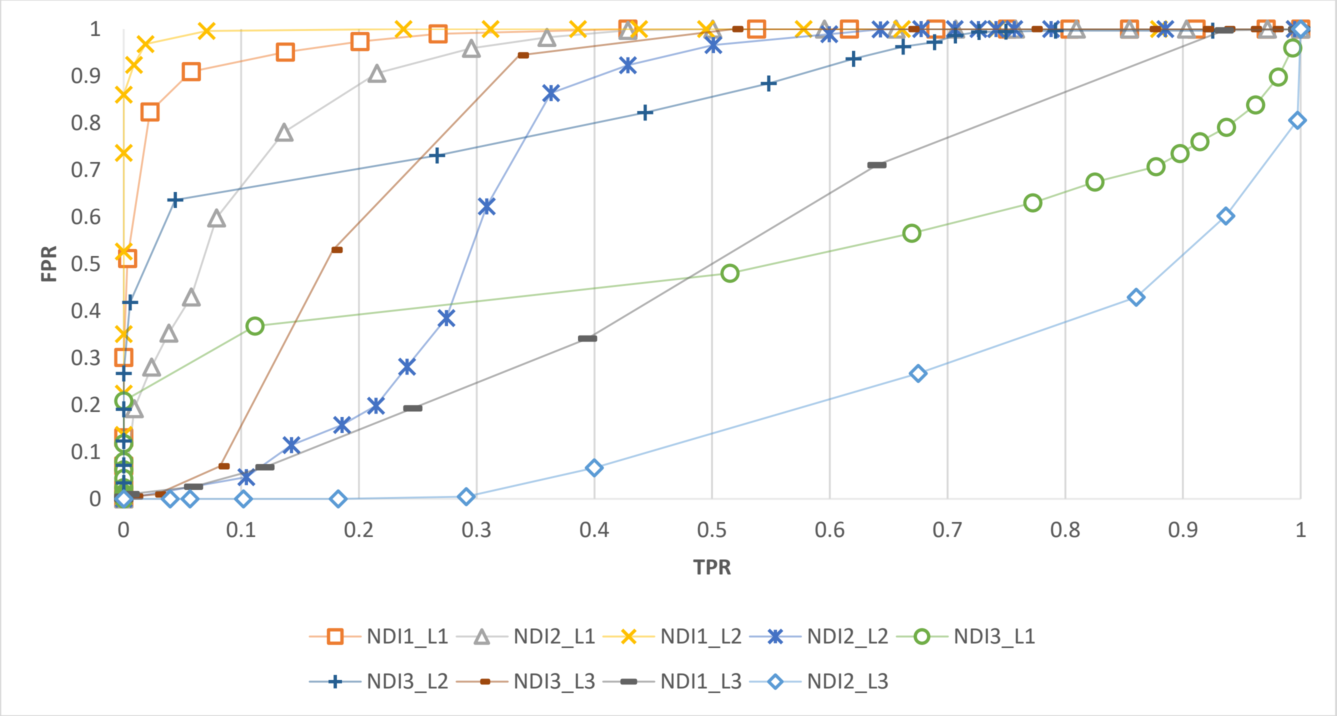


FIGURE 17: 9 ROC CURVE - 3 DIMENSIONS X 3 LIGHT LEVELS

NDII_LJ I REPRESENTS THE COLOR SPACE DIMENSION; J REPRESENTS THE ILLUMINATION LEVEL

In the online process, the algorithm receives RGB images from the camera in real time, transforms the representation to the relevant color space (NDI/HSI/LAB) and creates a binary image by applying the thresholds as following: three thresholds, one for each dimension are calculated from the nine thresholds learned by linear interpolation between two of the three illumination regions (Low, Medium, and High) selected as closest to the calculated illumination level for the specific sub-image from the grayscale image and using Equation 5.

$$T = \frac{T(closest\ from\ below)*(Light\ level - Ligh\ from\ below) + T(closest\ from\ above)(Light\ level\ from\ above - Light\ level)}{Light\ from\ above - Light\ from\ below} \quad (5)$$

For example, if the current light level is 40, and the thresholds in the offline process for the Low, Medium and High light levels were 10, 70 and 130, the threshold would be calculated in the following way:

$$T = \frac{T(10) * (40 - 10) + T(70)(70 - 40)}{70 - 10}$$

4.2. Morphological operations

The algorithm result is a binary image with major fruit detected and small clusters of pixels that were wrongly classified as fruits (e.g., Figure 18 – present figure with this before and after morphological operation). In addition, some fruits are split between several clusters. Several morphological operations were performed to overcome these problems.

Morphological opening operations were executed based on previous research which indicate their contribution (Vitzrabin and Edan, 2016) using the same method: erosion followed by dilation with a neighborhood of 11×11-pixel squares. The square function was used since there was no pre-defined knowledge about the expected fruit orientation. To connect close clusters, the closing morphological operation was then applied by dilation followed by erosion implemented with a 5×5-pixel square neighborhood.

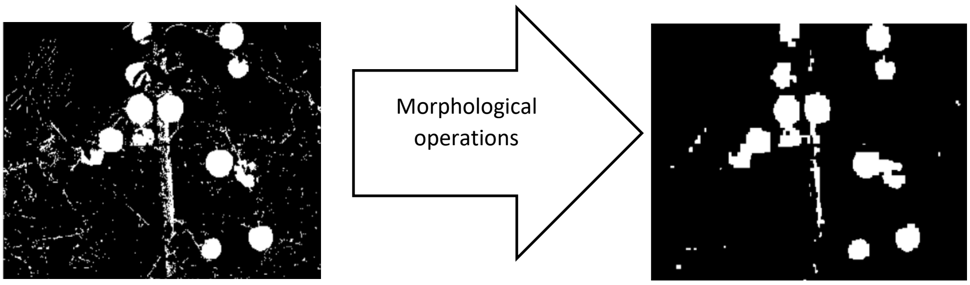


FIGURE 18: MORPHOLOGICAL OPERATION

4.3. Parameter tuning

The algorithm uses several parameters that influence the algorithm performances: T1, T2, Std, Classification rule direction- D1/D2. The following parameter tuning procedure (Figure 20) was developed and should be performed when exploring images from a new database, new operating conditions (cameras, illumination) or when exploring a new color space. The parameters are:

<u>Light level thresholds</u> (T1, T2) –The algorithm splits the images into sub images set to 1% of the entire image. Then, the algorithm computes the light level of each sub-image by calculating the average pixels values of the grayscale sub-image. Finally, the algorithm groups the sub images into three light level categories (see Figure 19) using two thresholds as presented in Equation 6.

$$f(x) = \begin{cases} \text{Low}\,, & 0 < x < T1 \\ \text{Medium}\,, & \text{T1} < x < \text{T2} \\ \text{High}\,, & x > \text{T2} \end{cases} \qquad (6)$$

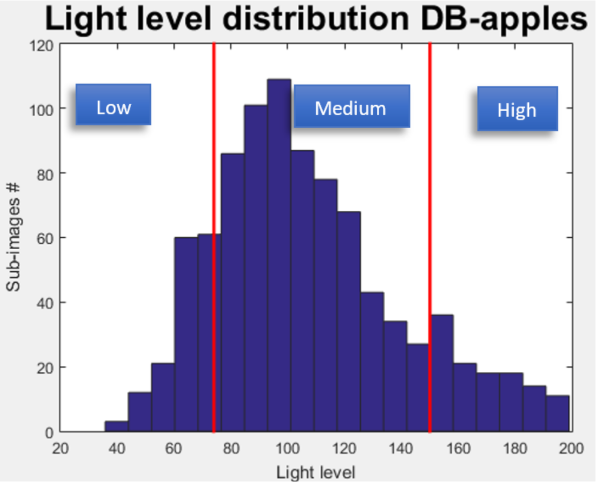


FIGURE 19: SUB-IMAGES LEVEL OF LIGHT DISTRIBUTION

To identify the PDF function of the data distributions of each database a χ^2 goodness of fit tests was conducted for the following distributions: Normal distribution, Poisson distribution or Negative binomial distribution. These distributions were selected based on the nature of the data. Once the distribution is selected for each database, the algorithm chooses the T1, T2 in a different way, as described in Figure 21.

Note that as described in the algorithm flow, the algorithm uses a third threshold. Sub-images above that threshold are ignored in the training process due to their high values (the sub images are almost completely white).

Stop splitting condition (std) - the algorithm splits an image into sub-images until the sub image achieves a predefined standard deviation (STD) value. This approach assumes that a larger sub-image contains higher STD value. To test this assumption, STD was calculated for different sizes of sub images for the different databases. The stop condition value (STD minimum value) is determined by maximizing the F score (Equation 4).

Classification rule direction– (D1, D2, D3) as detailed in the introduction, in the thresholding method, a value that differentiates the intensity of the object from its background is determined. When using different color spaces, one of the issues encountered was to determine for each color dimension if the intensity of the object is greater or smaller than the background. This information was learned as part of the tuning process. For this step a simple heuristic rule was used as follows based on the assumption that the images contain more background pixels than objects: 1. Execute **image>Threshold**. 2. If the pixels categorized as background represent less than 70% of the image, reverse the thresholding direction **images <Threshold**.

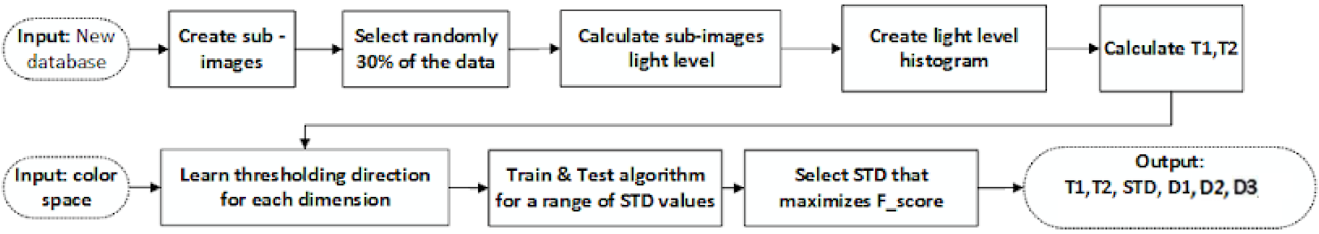

Input: New database
Create sub - images
Select randomly 30% of the data
Calculate sub-images light level
Create light level histogram
Calculate T1,T2
Input: color space
Learn thresholding direction for each dimension
Train & Test algorithm for a range of STD values
Select STD that maximizes F_score
Output: T1, T2, STD, D1, D2, D3

FIGURE 20: PARAMETER TUNING PROCESS

```
light_levels_array = calculate_light_level_for_each_sub_image (sub_images)

#chi tests goodness of fit
is_normal ,mu,sigma= test_normal_distribution(light_levels_array)
is_ poisson,lambda= test_poisson_distribution(light_levels_array)
is_negative_binomial,p,r= test_negative_binomial_distribution(light_levels_array)

if (is_normal):
   T1=mu-sigma; T2=mu+ sigma;
else:
   if (is_ poisson):
      [T1,T2]= get_thresholds($\sum_{0}^{T_1} \frac{\lambda^t}{T_1!} e^{-\lambda}$ ~0.15 , $\sum_{0}^{T_2} \frac{\lambda^t}{T_2!} e^{-\lambda}$ ~0.85)
  else:
     if (is_negative_binomial):
        [T1,T2]= get_thresholds($\sum_{i=0}^{T_1} \binom{T_1 + r - 1}{T_1} (1-p)^r p^{T_1}$ ~0.15 , $\sum_{i=0}^{T_2} \binom{T_2 + r - 1}{T_2} (1-p)^r p^k$ ~0.85)
     else:
        T1 = get_ percentile_0.15(light_levels_array)
        T2 = get_ percentile_0.85(light_levels_array)
```

FIGURE 21 PSEUDO CODE FOR THRESHOLDS SELECTION

5. Results & Discussion

5.1. Sub-image size vs. STD value

Figure 22 confirms the assumption that splitting an image to small sub images (small S) decreases the average STD of the sub images (in all three databases).

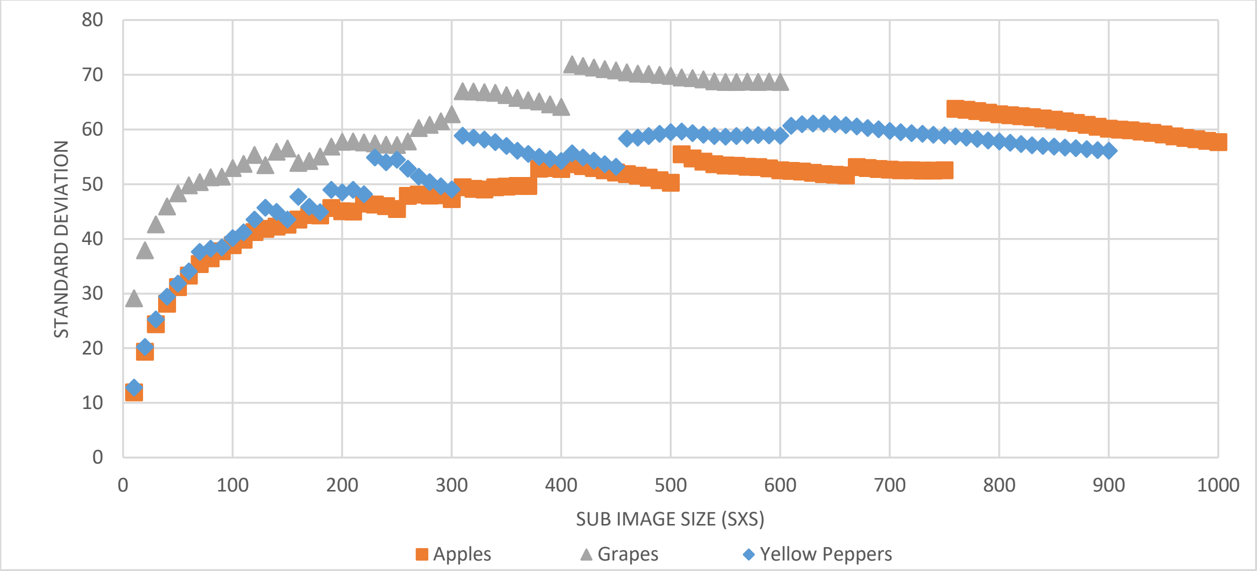


FIGURE 22: SUB IMAGE SIZE VS AVERAGE STD

5.2. Tuning process

This section presents the tuning process results, including thresholds derived to categorize the sub images into light level groups as well as the recursive stop condition that achieved best result for each database.

Light level distribution: As a first step the algorithm tried to fit for each database a known distribution. As shown in Table 1 the hypotheses were rejected in all three databases (significant level α=0.05).

TABLE 1 P-VALUE RESULTS FOR EACH DATABASE AND TESTED DISTRIBUTION

DB / **Hypothesis (α=0.05)**	**Ho: X~N(σ, μ)** **H1: Not**	**Ho: X~Poiss(λ)** **H1: Not**	**Ho: NB~(r,p)** **H1: Not**
Apples	H0 rejected P value: 0.0201	H0 rejected P value: 6.8764e-129	H0 rejected P value: 0
Grapes	H0 rejected P value: 1.2959e-21	H0 rejected P value: 0	H0 rejected P value: 0.002
Peppers	H0 rejected P value: 0.0013	H0 rejected P value: 2.4285e-222	H0 rejected P value: 0.0014

Therefore, the light level distribution was computed empirically for each database (Figure 23) along with T1 and T2 (Table 2). The variation in the light distributions between the different databases are described in Table 3. The variance of light in the grapes databases is significantly higher than in both the apples and the peppers databases, the pepper database is significantly darker and highly skewed. Therefore, for each database T1 and T2 were significantly different implying the importance of the tuning process.

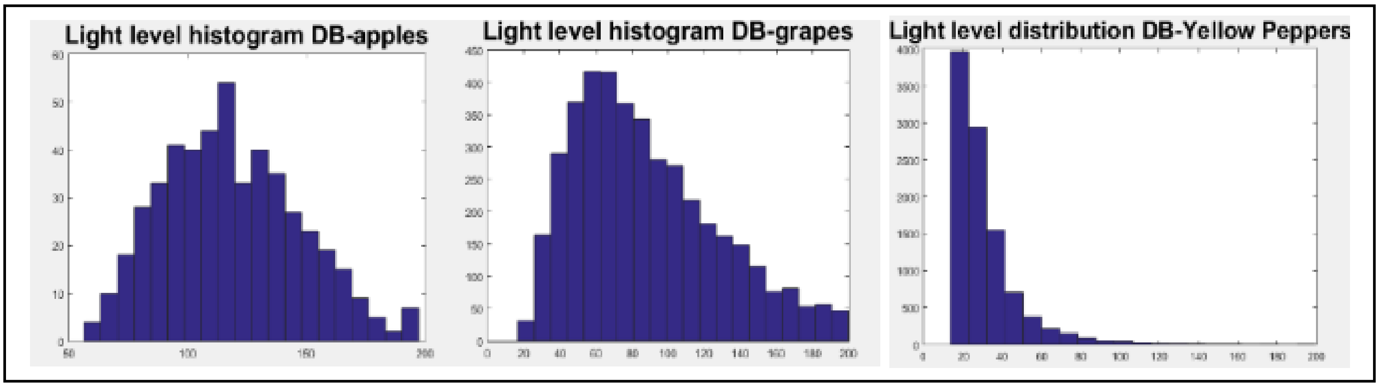


FIGURE 23: LIGHT LEVEL DISTRIBUTION WAS COMPUTED FOR EACH DATABASE

TABLE 2: T1 AND T2 VALUES DETERMINED FOR EACH DATABASE

Measure/DB	Apples	Grapes	Peppers
T1	84	49	18
T2	140	130	47

TABLE 3: DESCRIPTIVE STATISTICS OF THE DIFFERENT LIGHT DISTRIBUTIONS

Measure/DB	Apples	Grapes	Peppers
Mean	118.46	88.00	32.09
Std	28.04	37.90	18.92
Skewness	0.40	0.68	3.16
Kurtosis	-0.17	-0.13	15.36
Median	116.31	81.06	26.93

Stop splitting condition: Using a low STD value as a stop condition, increases performance (Figure 24). This happens since smaller sub images contain less illumination differences. However, small STD values can create also too small sub-images which may not contain fruit and background pixels in the same frame. In these cases, the algorithm cannot learn a threshold that could differ between them. Additionally, results reveal that when using high STD values the performances stay constant. This happens since beyond a certain value the algorithm does not split the image even once.

As part of the parameter tuning process, the STD value is selected by testing the performances of a range of STD [0,100]. For each STD value the algorithm runs five iterations were it randomly select P% of the images, from the selected images it uses 70% for train and 30% test. Final selected STD values are presented in Table 4 for each database and color space (using P=30% and 50%).

Classification rule direction: as shown in Table 4, the direction of the classification rule in the thresholding process can be different for each color dimension, therefore this must be learned as part of the tuning process.

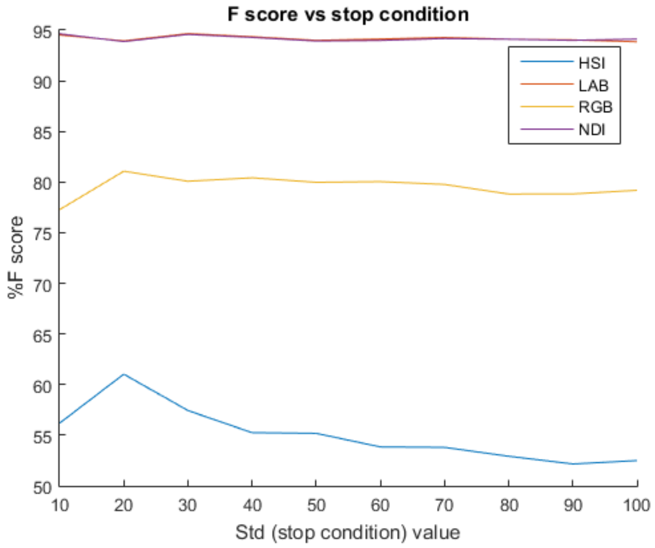


FIGURE 24: F-SCORE VS. INCREASING STD VALUE AS STOP CONDITION FOR THE RECURSIVE FUNCTION ON APPLES DB

Table 4: STD value chosen for each database and color space

DB		Apples				Grapes				Peppers			
Color space		**HSI**	**LAB**	**NDI**	**RGB**	**HSI**	**LAB**	**NDI**	**RGB**	**HSI**	**LAB**	**NDI**	**RGB**
STD (P=30%)		20	30	10	20	10	20	60	20	100	10	10	10
STD (P=50%)		20	10	10	30	20	20	70	20	100	20	10	10
Classification rule direction	D1	>	<	>	>	<	>	>	>	<	>	>	>
	D2	>	>	>	<	>	<	>	>	>	>	>	>
	D3	>	>	>	<	<	>	<	<	>	>	<	<

5.3. Color spaces analyses

In this section algorithm performances results are presented in figures for each color space followed by a table representing the best color space performances including performances for all color space dimensions combinations.

Apples – Results (Figure 25) reveal that NDI and LAB color spaces result in similar best performances. In Table 5 the preferences for each dimension in the NDI color space and the performances when using the intersection between them is shown. The NDI first dimension (see Equation 1) represents the difference between the red and green colors in the image. The objects in this database are red apples and most of the background is green leaves therefore, as expected, the first NDI obtained the best F of 93.17%. In the LAB color space, results (Table 6) reveal that the second dimension (A) yields the best F score of 93.19%.

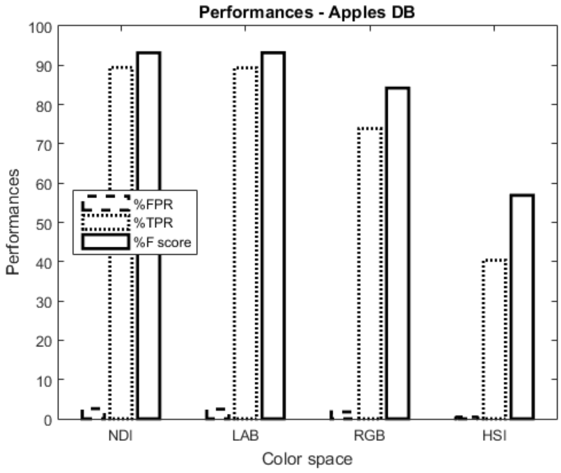


FIGURE 25: : COLOR SPACES PERFORMANCES - APPLES DB

TABLE 5: PERFORMANCES OF EACH NDI DIMENSION AND INTERSECTIONS – APPLES

Measure /Dimension	1	2	3	1∩2	1∩3	2∩3	1∩2∩3
% FPR	2.59	40.91	31.38	1.64	1.32	2.48	0.48
% TPR	89.45	83.53	68.39	78.52	64.82	54.65	54.10
% F	93.17	67.85	67.80	86.75	77.60	69.67	69.69

Table 6: Performances of each LAB dimension and intersections – Apples

Measure /Dimension	1	2	3	1∩2	1∩3	2∩3	1∩2∩3
% FPR	33.58	2.45	77.08	1.78	28.55	1.55	1.07
% TPR	61.26	89.34	85.26	56.59	52.95	76.27	48.80
% F	56.79	93.19	35.85	69.02	54.61	85.37	62.58

Grapes – The NDI color space obtained the best result for grapes (Figure 26) with a F score of 73.52%. The second-best color space is the LAB with a F score of 62.54% The best NDI results were obtained using the second dimension (Table 7).

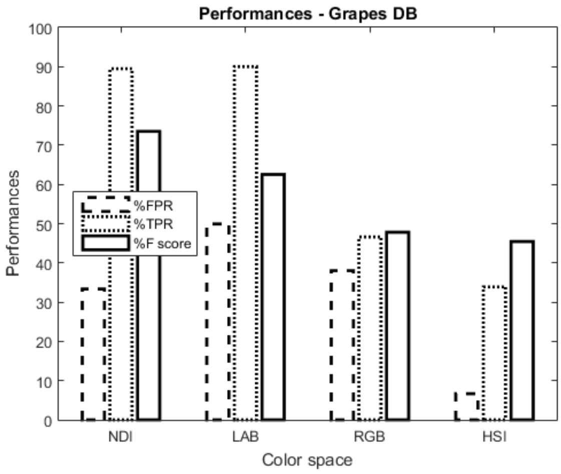


Figure 26: Color spaces performances - Grapes DB

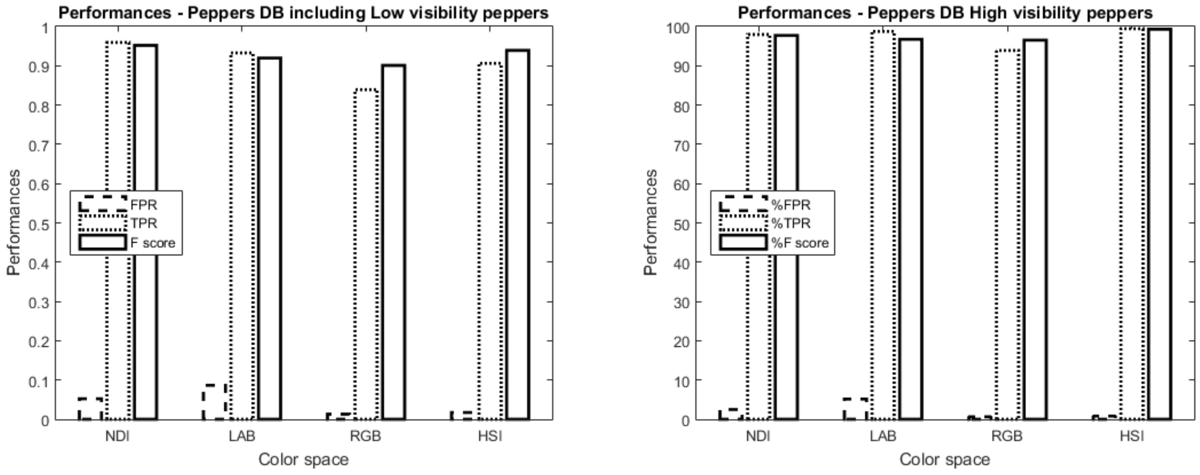
Table 7: Performances of each NDI dimension and intersections – Grapes

Measure /Dimension	1	2	3	1∩2	1∩3	2∩3	1∩2∩3
% FPR	35.86	33.35	52.90	4.86	5.50	32.35	4.09
% TPR	44.52	89.48	89.99	38.53	37.27	87.50	36.70
% F	47.19	73.52	58.05	50.12	48.93	73.20	48.65

Peppers

High visibility – Figure 27a indicates that HSI color space obtained the best results with relatively low FPR (0.81%) and very high TPR (99.43%) resulting in a high F score (99.31%). The second-best color space is NDI with FPR=2.48% and TPR=97.96% (F= 97.72%). The best HSI result, were obtained using the combination of the first and the second dimensions (Table 8).

Including low visibility – Figure 27b indicates that NDI color space obtained the best results with relatively low FPR (5.24%) and very high TPR (95.93) resulting in high F score (95.19%). Although on the "high visibility" peppers HSI obtained the best performances, when trying to detect peppers in dark areas that are less visible, NDI shows better results. The best NDI results were obtained using the intersection between the first and the second dimensions (Table 9).



(a) (b)

FIGURE 27: COLOR SPACES PERFORMANCES - PEPPERS DB

TABLE 8: PERFORMANCES OF EACH HSI DIMENSION AND INTERSECTIONS – PEPPERS HIGH VISIBILITY

Measure/DB	1	2	3	1∩2	1∩3	2∩3	1∩2∩3
% FPR	2.48	5.15	0.64	0.81	2.48	5.15	0.64
% TPR	97.96	98.75	93.91	99.43	97.96	98.75	93.91
% F	97.72	96.73	96.51	99.31	97.72	96.73	96.51

TABLE 9: PERFORMANCES OF EACH NDI DIMENSION AND INTERSECTIONS – PEPPERS INCLUDING LOW VISIBILITY

Measure /Dimension	1	2	3	1∩2	1∩3	2∩3	1∩2∩3
% FPR	66.57%	5.24%	9.23%	1.42%	1.24%	4.57%	0.99%
% TPR	85.61%	95.93%	92.49%	82.20%	78.64%	92.33%	78.61%
% F	46.96%	95.19%	91.24%	88.91%	86.59%	93.51%	86.67%

5.4. Sensitivity analysis

a) Noise – analysis shows that the algorithm is robust to noise in the image up to 15% in the apples and peppers databases (Figure 28). The grapes images are more sensitive to noise and performance drops when noise values of 5% are added. Although better F score values were obtained for NDI and HSI for grapes and peppers, we can see that LAB color space yields more robust performance when adding noise to the images.

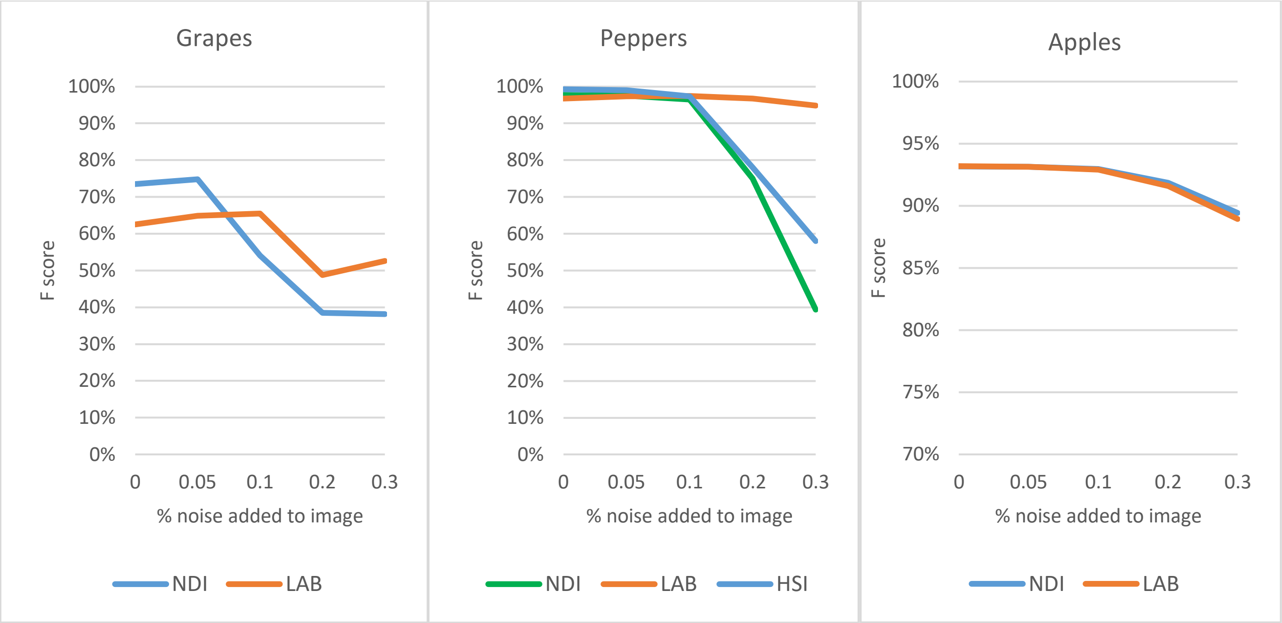


FIGURE 28: SENSITIVITY ANALYSIS - ADDING NOISE TO IMAGES

b) Thresholds learned in offline process - as expected, TPR decreases when the threshold values change. The algorithm is relatively robust to the change in the thresholds for apples and peppers. Performance in the grapes images is more sensitive to threshold changes and yields a significant decrease in TPR when increasing the threshold value (Table 10).

TABLE 10: THRESHOLD VALUES CHANGED BY ±5%, ±10% AND ±15% ACCORDING TO THE THRESHOLD IN EACH REGION

DB	Measure/Change in thresholds	-15	-10	-5	0	5	10	15
Apples	%FPR	3.58	3.43	3.30	2.59	3.06%	2.93	2.81
	%TPR	91.47	91.28	91.07	89.45	90.75%	90.57	90.44
Grapes	%FPR	21.59	18.40	15.53	33.35	11.00%	9.23	7.72
	%TPR	78.02	72.63	66.42	89.48	50.99%	43.63	36.24
Peppers	%FPR	0.98	0.91	0.86	0.81	0.78%	0.70	0.65
	%TPR	99.25	99.22	99.20	99.43	99.12%	99.07	99.04

c) Stop condition - the algorithm shows more robustness to apples and peppers images than grapes (Figure 29).

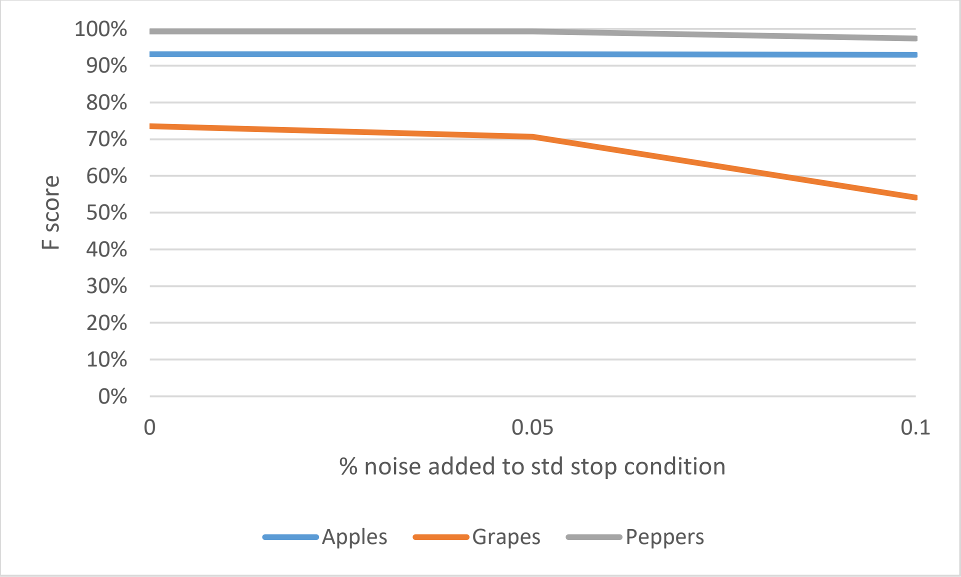


FIGURE 29: SENSITIVITY ANALYSIS - ADDING NOISE TO STD STOP CONDITION

d) Train/Test – the expectation is that more training images leads to better performance until overfitting is accommodated. There is a clear increase in TPR however FPR increases as well at 80%,90% train.

Table 11: Performances vs. different % images database as train set

DB	Measure / %Train	10	20	30	40	50	60	70	80	90
Grapes	%FPR	32.81	37.08	28.54	36.16	31.83	29.10	29.63	40.51	40.80
	%TPR	88.79	89.62	87.01	88.44	87.58	82.55	87.14	94.53	95.85
	%F	73.35	70.20	75.19	69.41	72.49	70.24	73.92	72.55	72.54

The tuning process resulted in increased performances for both the grapes and peppers databases with 40% and 1.49% increase respectively. The results for the apple databased were similar with only a 0.1% increase as expected (since this was similar to the database the previous parameters were derived from).

Table 12: Parameter tuning contribution to algorithm performances

DB	Measure	Performances using Previous params	Performances using Tuning process
Apples	%FPR	2.53	2.59
	%TPR	89.23	89.45
	%F	93.08	93.17
Grapes	%FPR	18.63	33.35
	%TPR	63.70	89.48
	%F	67.30	73.52
Peppers	%FPR	1.00	0.81
	%TPR	97.97	99.43
	%F	98.47	99.31

The morphological operations process increases the F score by 2.85%, 8.59%, and 2.71% for the apples grapes and peppers databases respectively (Figure 30).

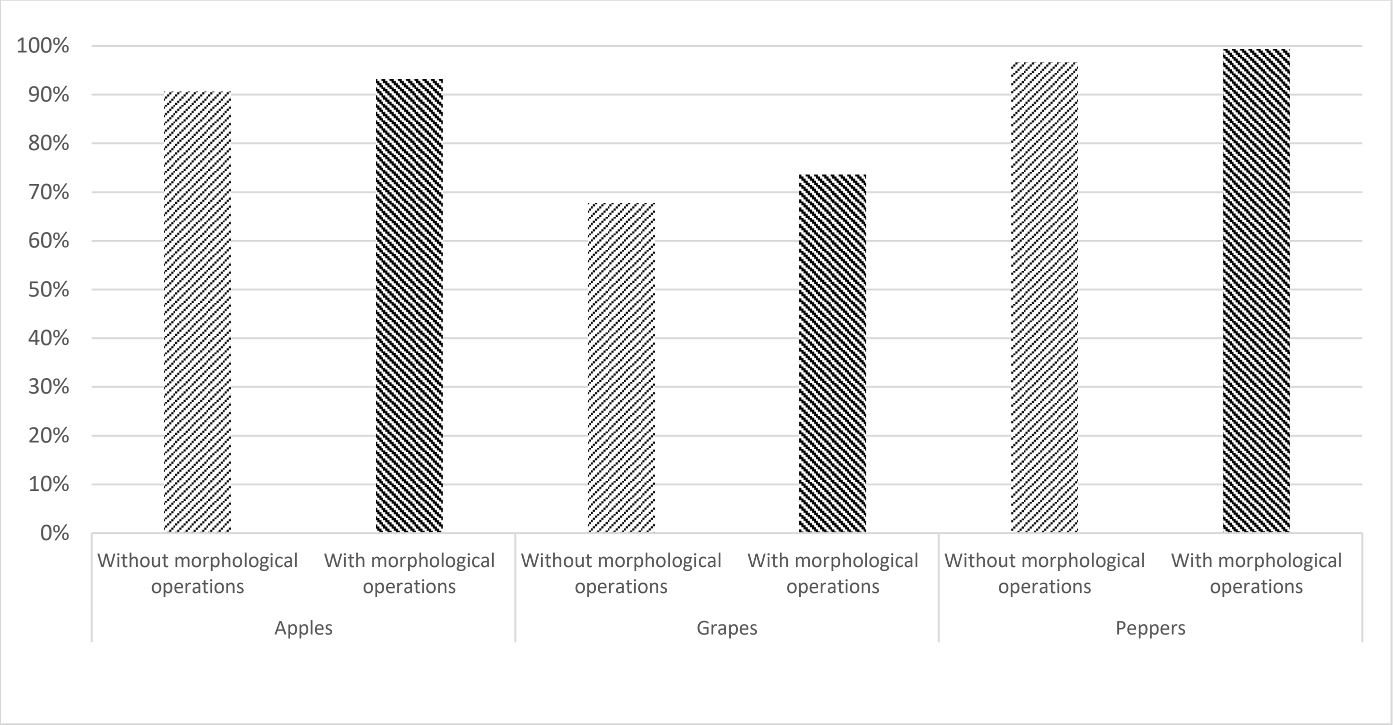


FIGURE 30: MORPHOLOGICAL OPERATION CONTRIBUTION

6. Conclusions & Future work

The algorithm successfully detected apples and peppers in variable lighting conditions with an F-score of 93.15% and 97.40% respectively, resulting in one of the best detection rates achieved to date in fruit detection to the best of our knowledge. Previous research has shown 85%-90% TPR (Bac et al., 2014; Vitzrabin and Edan 2016; Sa et al., 2016). The algorithm has shown less impressive results in the grapes database (F score of 73.52%) due to difficulties associated with differentiating between green fruits and a green background (leaves). In this case, additional features (e.g. morphological operations fitted for grapes see Bernstein, Shahar, Shapiro, Edan, 2010) should be used to increase performance.

Different color spaces yielded best results for each fruit variety, implying that the color space must be analyzed and fitted to the specific fruit. The LAB color space is more robust to noise in images and hence should be used when images are of low quality. The algorithm is robust to changes in the threshold learned by the offline process and to noise effects in images. Morphological operations can improve performance and hence should be utilized.

The tuning process developed in this research enables the algorithm to adapt automatically to changing conditions/objectives (i.e. to detect other fruit with different colors and other outdoor conditions) and hence should be used for improved target detection in highly variable illumination conditions.

7. References

Al-Allaf, O.N., 2014. Review of face detection systems based artificial neural networks algorithms. arXiv preprint arXiv:1404.1292.

Albert, A. and Michaels, A., Robert Bosch GmbH, 2018. Weed control device. U.S. Patent Application 15/576,914.

Alhaija, H.A., Mustikovela, S.K., Mescheder, L., Geiger, A. and Rother, C., 2017. Augmented Reality Meets Computer Vision: Efficient Data Generation for Urban Driving Scenes. International Journal of Computer Vision, pp.1-12.

Almendral, K.A.M., Babaran, R.M.G., Carzon, B.J.C., Cu, K.P.K., Lalanto, J.M. and Abad, A.C., 2018. Autonomous Fruit Harvester with Machine Vision. Journal of Telecommunication, Electronic and Computer Engineering (JTEC), 10(1-6), pp.79-86.

Arivazhagan, S., Shebiah, R.N., Nidhyanandhan, S.S. and Ganesan, L., 2010. Fruit recognition using color and texture features. Journal of Emerging Trends in Computing and Information Sciences, 1(2), pp.90-94.

Arroyo, J., Guijarro, M. and Pajares, G., 2016. An instance-based learning approach for thresholding in crop images under different outdoor conditions. Computers and Electronics in Agriculture, 127, pp.669-679.

Aufrère, R., Chapuis, R. and Chausse, F., 2000. A dynamic vision algorithm to locate a vehicle on a nonstructured road. The International Journal of Robotics Research, 19(5), pp.411-423.

Bac, C.W., van Henten, E.J., Hemming, J. and Edan, Y., 2014. Harvesting robots for high-value crops: State-of-the-art review and challenges ahead. Journal of Field Robotics, 31(6), pp.888-911.

Barth, Ruud, Jochen Hemming, and Eldert J. van Henten., 2016. "Design of an eye-in-hand sensing and servo control framework for harvesting robotics in dense vegetation." Biosystems Engineering, 146, pp.71-84.

Boulmerka, A., Saïd Allili, M., & Ait-Aoudia, S. (2014). A generalized multiclass histogram thresholding approach based on mixture modelling. Pattern Recognition, 47(3), 1330–1348

Bulanon, D.M., Kataoka, T., Ota, Y. and Hiroma, T., 2002. AE—automation and emerging technologies: a segmentation algorithm for the automatic recognition of Fuji apples at harvest. Biosystems Engineering, 83(4), pp.405-412.

Chris Harris and Mike Stephens. 1988. “A combined corner and edge detector.” In: Alvey vision conference. Vol. 15. 50. Citeseer. 1988, pp. 10–5244 (cit. on p. 6)

David G Lowe. 1999 . “Object recognition from local scale-invariant features”. In: Computer vision. The proceedings of the seventh IEEE international conference on. Vol. 2. Ieee. 1999, pp. 1150–1157 (cit. on p. 6).

Edan, Y. and Miles, G.E., 1994. Systems engineering of agricultural robot design. IEEE transactions on systems, man, and cybernetics, 24(8), pp.1259-1265.

Gasparri, J.P., Bouchet, A., Abras, G., Ballarin, V. and Pastore, J.I., 2011. Medical image segmentation using the HSI color space and Fuzzy Mathematical Morphology. In Journal of Physics: Conference Series (Vol. 332, No. 1, p. 012033). IOP Publishing.

Ghosal, S., Blystone, D., Singh, A.K., Ganapathysubramanian, B., Singh, A. and Sarkar, S., 2018. An explainable deep machine vision framework for plant stress phenotyping. Proceedings of the National Academy of Sciences, 115(18), pp.4613-4618.

Gorbe, E. and Calatayud, A., 2012. Applications of chlorophyll fluorescence imaging technique in horticultural research: a review. Scientia Horticulturae, 138, pp.24-35.

Gongal, A., Amatya, S., Karkee, M., Zhang, Q. and Lewis, K., 2015. Sensors and systems for fruit detection and localization: A review. Computers and Electronics in Agriculture, 116, pp.8-19.

Gunatilaka, A.H. and Baertlein, B.A., 2001. Feature-level and decision-level fusion of noncoincidently sampled sensors for land mine detection. IEEE transactions on pattern analysis and machine intelligence, 23(6), pp.577-589.

Guyon, I., 1997. A scaling law for the validation-set training-set size ratio. AT&T Bell Laboratories, pp.1-11.

Hall, D.L. and McMullen, S.A., 2004. Mathematical techniques in multisensor data fusion. Artech House.

Hannan, M.W., Burks, T.F. and Bulanon, D.M., 2007. A real-time machine vision algorithm for robotic citrus harvesting. In 2007 ASAE Annual Meeting (p. 1). American Society of Agricultural and Biological Engineers.

Haralick, R.M. and Shapiro, L.G., 1985, April. Image segmentation techniques. In Applications of Artificial Intelligence II (Vol. 548, pp. 2-10). International Society for Optics and Photonics.

Haward, S.J., 2016. Microfluidic extensional rheometry using stagnation point flow. Biomicrofluidics, 10(4), p.043401.

Hemming, J., Ruizendaal, J., Hofstee, J.W. and van Henten, E.J., 2014. Fruit detectability analysis for different camera positions in sweet-pepper. Sensors, 14(4), pp.6032-6044.

Jalled, F. and Voronkov, I., 2016. Object Detection Using Image Processing. arXiv preprint arXiv:1611.07791.

Jameel, S.K. and Manza, R.R., 2012. Color image segmentation using wavelet. IJAIS, 1(6), pp.1-4.

Jiang, J.A., Chang, H.Y., Wu, K.H., Ouyang, C.S., Yang, M.M., Yang, E.C., Chen, T.W. and Lin, T.T., 2008. An adaptive image segmentation algorithm for X-ray quarantine inspection of selected fruits. Computers and electronics in agriculture, 60(2), pp.190-200.

Kanungo, P., Nanda, P.K. and Ghosh, A., 2010, October. Parallel genetic algorithm based adaptive thresholding for image segmentation under uneven lighting conditions. In Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on (pp. 1904-1911). IEEE.

Kapach, K., Barnea, E., Mairon, R., Edan, Y. and Ben-Shahar, O., 2012. Computer vision for fruit harvesting robots–state of the art and challenges ahead. International Journal of Computational Vision and Robotics, 3(1-2), pp.4-34.

Kong, D.Y., Zhao, D.A., Zhang, Y., Wang, J.J. and Zhang, H.X., 2010, June. Research of apple harvesting robot based on least square support vector machine. In Electrical and Control Engineering (ICECE), 2010 International Conference on (pp. 1590-1593). IEEE.

Li, H., Lin, Z., Shen, X., Brandt, J. and Hua, G., 2015. A convolutional neural network cascade for face detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 5325-5334).

Liu, S., Cossell, S., Tang, J., Dunn, G. and Whitty, M., 2017. A computer vision system for early stage grape yield estimation based on shoot detection. Computers and Electronics in Agriculture, 137, pp.88-101.

Loianno, G., Scaramuzza, D. and Kumar, V., 2018. Special Issue on High-Speed Vision-Based Autonomous Navigation of UAVs. Journal of Field Robotics, 35(1), pp.3-4.

Luo, L., Tang, Y., Lu, Q., Chen, X., Zhang, P. and Zou, X., 2018. A vision methodology for harvesting robot to detect cutting points on peduncles of double overlapping grape clusters in a vineyard. Computers in Industry, 99, p

Mohanty, S.P., Hughes, D.P. and Salathé, M., 2016. Using deep learning for image-based plant disease detection. Frontiers in plant science, 7, p.1419.p.130-139.

Nuske, S., Wilshusen, K., Achar, S., Yoder, L., Narasimhan, S. and Singh, S., 2014. Automated visual yield estimation in vineyards. Journal of Field Robotics, 31(5), pp.837-860.Vincent, L., 1993. Morphological

Ostovar, A., Ringdahl, O. and Hellström, T., 2018. Adaptive Image Thresholding of Yellow Peppers for a Harvesting Robot. Robotics, 7(1), p.11.

Park, J.H., Lee, G.S., Cho, W.H., Toan, N., Kim, S.H. and Park, S.Y., 2010, June. Moving object detection based on clausius entropy. In Computer and Information Technology (CIT), 2010 IEEE 10th International Conference on (pp. 517-521). IEEE.

Pascale, D., 2003. A review of rgb color spaces... from xyy to r’g’b’. Babel Color, 18, pp.136-152.

Pereira, L.F.S., Barbon, S., Valous, N.A. and Barbin, D.F., 2018. Predicting the ripening of papaya fruit with digital imaging and random forests. Computers and Electronics in Agriculture, 145, pp.76-82.

Rakun, J., Stajnko, D. and Zazula, D., 2011. Detecting fruits in natural scenes by using spatial-frequency based texture analysis and multiview geometry. Computers and Electronics in Agriculture, 76(1), pp.80-88.

Rasolzadeh, B., Björkman, M., Huebner, K. and Kragic, D., 2010. An active vision system for detecting, fixating and manipulating objects in the real world. The International Journal of Robotics Research, 29(2-3), pp.133-154.

Reibman, A.R. and Nolte, L.W., 1987. Optimal detection and performance of distributed sensor systems. IEEE Transactions on Aerospace and Electronic Systems, (1), pp.24-30.

Revol-Muller, C., Peyrin, F., Carrillon, Y. and Odet, C., 2002. Automated 3D region growing algorithm based on an assessment function. Pattern Recognition Letters, 23(1-3), pp.137-150.

Ruiz-del-Solar, J., Loncomilla, P. and Soto, N., 2018. A Survey on Deep Learning Methods for Robot Vision. arXiv preprint arXiv:1803.10862.

Rong, D., Ying, Y. and Rao, X., 2017. Embedded vision detection of defective orange by fast adaptive lightness correction algorithm. Computers and Electronics in Agriculture, 138, pp.48-59.

Rosin, P.L. and Ioannidis, E., 2003. Evaluation of global image thresholding for change detection. Pattern recognition letters, 24(14), pp.2345-2356.

Sa, I., Ge, Z., Dayoub, F., Upcroft, B., Perez, T. and McCool, C., 2016. Deepfruits: A fruit detection system using deep neural networks. Sensors, 16(8), p.1222.

Sakthivel, K., Nallusamy, R. and Kavitha, C., 2015. Color Image Segmentation Using SVM Pixel Classification Image. World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering, 8(10), pp.1919-1925.

Sarig, Y., 1993. Robotics of fruit harvesting: A state-of-the-art review. Journal of agricultural engineering research, 54(4), pp.265-280.

Schalkoff, R.J., 1989. Digital image processing and computer vision (Vol. 286). New York: Wiley.

Shapiro, L.G., 2001. GC stockman,“. Computer vision,” Prentice Hall Inc.

Shmmala, F.A. and Ashour, W., 2013. Color based image segmentation using different versions of K-Means in two Spaces. Global Advanced Research Journal of Engineering, Technology and Innovation (GARJETI, 1(1), pp.030-041.

Shrestha, D.S., Steward, B. and Bartlett, E.R.I.C., 2001, November. Segmentation of plant from background using neural network approach. In Intelligent Engineering Systems through Artificial Neural Networks: Proc. Artificial Neural Networks in Engineering (ANNIE) International Conference (Vol. 11, pp. 903-908). sn.

Shih, F.Y. and Cheng, S., 2005. Automatic seeded region growing for color image segmentation. Image and vision computing, 23(10), pp.877-886.

Siegel, M. and Wu, H., 2003, May. Objective evaluation of subjective decisions. In Soft Computing Techniques in Instrumentation, Measurement and Related Applications, 2003. SCIMA 2003. IEEE International Workshop on (pp. 14-18). IEEE.

Sonka, M., Hlavac, V. and Boyle, R., 2014. Image processing, analysis, and machine vision. Cengage Learning.

Szeliski, R., 2010. Computer vision: algorithms and applications. Springer Science & Business Media.

Tona, E., Calcante, A. and Oberti, R., 2018. The profitability of precision spraying on specialty crops: a technical–economic analysis of protection equipment at increasing technological levels. Precision Agriculture, 19(4), pp.606-629.

Vincent, L., 1993. Morphological grayscale reconstruction in image analysis: applications and efficient algorithms. IEEE transactions on image processing, 2(2), pp.176-201.

Vishvjit, S. and Nalwa, A., 1993. A Guided Tour of Computer Vision.

Vitzrabin, E., 2015 Ph.D., Dynamic local thresholding using sensor fusion, Dept. of Industrial Engineering and Management, Ben-Gurion University of the Negev.

Vitzrabin, E. and Edan, Y., 2016. Changing task objectives for improved sweet pepper detection for robotic harvesting. IEEE Robotics and Automation Letters, 1(1), pp.578-584.

Vitzrabin, E. and Edan, Y., 2016. Adaptive thresholding with fusion using a RGBD sensor for red sweet-pepper detection. Biosystems Engineering, 146, pp.45-56.

Wang, J., He, J., Han, Y., Ouyang, C. and Li, D., 2013. An adaptive thresholding algorithm of field leaf image. Computers and electronics in agriculture, 96, pp.23-39.

Woebbecke, D.M., Meyer, G.E., Von Bargen, K. and Mortensen, D.A., 1993, May. Plant species identification, size, and enumeration using machine vision techniques on near-binary images. In Optics in Agriculture and Forestry (Vol. 1836, pp. 208-220). International Society for Optics and Photonics.

Wuhib, F., Dam, M. and Stadler, R., 2008. Decentralized detection of global threshold crossings using aggregation trees. Computer Networks, 52(9), pp.1745-1761.

Zemmour, E., Kurtser, P. and Edan, Y., 2017, April. Dynamic thresholding algorithm for robotic apple detection. In Autonomous Robot Systems and Competitions (ICARSC), 2017 IEEE International Conference on (pp. 240-246). IEEE.

Zhang, Y.J., 1996. A survey on evaluation methods for image segmentation. Pattern recognition, 29(8), pp.1335-1346.

Zheng, L., Zhang, J. and Wang, Q., 2009. Mean-shift-based color segmentation of images containing green vegetation. Computers and Electronics in Agriculture, 65(1), pp.93-98.

8. Appendices

Appendix 1
Dynamic thresholding algorithm for robotic apple detection

(Zemmour et al., 2017)

2017 IEEE International Conference on
Autonomous Robot Systems and Competitions (ICARSC)
April 26-28, Coimbra, Portugal

Dynamic thresholding algorithm for robotic apple detection

Elie Zemmour, Polina Kurtser, Yael Edan

Department of Industrial Engineering and Management, Ben-Gurion University of the Negev
Beer Sheva 84105, Israel

***Abstract*— This paper presents a dynamic thresholding algorithm for robotic apple detection. The algorithm enables robust detection in highly variable lighting conditions. The image is dynamically split into variable sized regions, where each region has approximately homogeneous lighting conditions. Nine thresholds were selected so as to accommodate three different illumination levels for three different dimensions in the natural difference index (NDI) space by quantifying the required relation between true positive rate and false positive rate. This rate can change along the robotic harvesting process, aiming to decrease FPR from far views (to minimize cycle times) and to increase TPR from close views (to increase grasping accuracy). Analyses were conducted on apple images acquired in outdoor conditions. The algorithm improved previously reported results and achieved 91.14% true positive rate (TPR) with 3.05% false positive rate (FPR) using the NDI first dimension and a noise removal process**

Keywords—dynamic thresholding; object detection; apples detection; robotic harvesting.

I. INTRODUCTION

Robotic harvesting systems can increase agricultural productivity by reducing manual labor and production costs [1]. When developing a robotic harvester, a basic step is to be able to identify the fruit [2]. Despite intensive research conducted in identifying fruits, implementing a real time vision system still remains a complex task [1],[2] and current detection is limited to 87% detection rate with 3.8% false alarms [3].

Features such as size, shape, texture and location, which help in object detection are subject to high variability in the agricultural domain [3]. Moreover, fruits grow in an unstructured environment with highly variable lighting conditions and obstructions that influence detection performance [3]. Color and texture are fundamental characteristics of natural images and play an important role in visual perception [9]. An image can be represented by different color spaces, each one emphasizing other color behavior [9].

Most object detection algorithms use a histogram of the image in order to determine the threshold that best differentiates between the background and the object [4]. The threshold is computed by finding the histogram minimum (Fig. 1). However, the global minimum is very hard to find in most cases [5].

Current optimal thresholding algorithms determine the threshold only in a one-dimensional space - either R or G or B or a combination of their values [10]. In the transformation from three color dimensions into one information is lost. In this paper we use a three dimensional thresholding algorithm.

Several "state of the art" approaches deal with image segmentation using the HSI space. The HSI color model represents every color with three components: hue (H), saturation (S) and intensity (I). The H-component describes color itself in the form of an angle between 0∘ and 360∘. The S-component signals how much the color is polluted with white color in a range [0, 1]. The range of I is also in the [0, 1] range, where 0 represents black and 1 represents white [11]. For example, in [12] the segmentation is conducted using a K-means algorithm over the HSI color space. The K-means algorithm is used to locate clusters of colors within a color image. The hue, saturation, and intensity components are used in the segmentation process instead of the red, blue, and green components so that the segmentation is based upon human visual perception of color. Two separate K-means are performed one over the one-dimensional hue space and the other over the two-dimensional saturation-luminance space [12].

K-means segmentation is also commonly used with the LAB color space. The LAB color space is a color-opponent space with dimension L for lightness and A, B for the color opponent dimensions [18]. The LAB color space is designed to approximate human vision. It aspires to perceptual uniformity with its L component closely matching human perception of lightness. It can thus be used to make accurate color balance corrections by modifying output curves in the A and B components, or to adjust the lightness contrast using the L component which models the output of physical devices rather than human visual perception [19].

Another segmentation method [11] uses mean shift analysis. A non-parametric feature-space analysis technique is employed to locate the maxima of a density function, based on multiple color features, such as hue and saturation in HSI color space, as well as red, green and blue value in RGB color space [11].

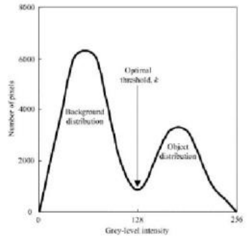


Fig.1. Optimal threshold in bimodal histogram

In the current study the algorithm uses the three dimensional normalized difference index (NDI) space. NDI is commonly employed in agricultural settings to differentiate between the fruit and the background [13] since it helps to overcome changes in illumination and shading due to its normalization technique [14].

These three dimensions are used with a dynamic thresholding algorithm [3]. There are several methods that use dynamic thresholding algorithms [6],[7]. A common approach is to use multi-resolution windows which apply a bottom up method, merging pixels while a criteria is met [6],[7]. Another approach is the top down method, where the image is divided into sub regions according to a criteria. The top down approach reduces execution speed and improves generalization [8].

In this paper a set of three thresholds is determined for each region of the image according to the lighting setting in this region which changes. The algorithm dynamically divides the image into several regions, each with approximately the same lighting conditions.

II. Methods

A. Overview

This research is based on a dynamic thresholding algorithm which incorporated changing task objectives algorithm to improve sweet pepper detection for a robotic harvester [3]. The algorithm was implemented in Matlab 2015a following the flow presented by [3]. The algorithm includes an offline process, in which the algorithm learns color thresholds for detection followed by an online process responsible for implementing the thresholds learned in the offline process on a new image, in order to detect pixels that represent fruit in real time (Fig. 4) [3]. Previous analyses indicated the superiority of the dynamic thresholding algorithm [3] as compared to traditional algorithms such as constant threshold, bimodal histogram and a constant threshold 3D thresholding.

In the current paper, analyses were conducted to determine the NDI dimensions that result in best performance. Furthermore, a noise removal process was added.

B. Database

The orchard apples database includes 113 "*Royal Gala*" apples in 9 images coming from an orchard in Chile in March 2012 under natural growing conditions with a Prosilica GC2450C camera. The images were captured in daylight, half of the images were acquired under direct sunlight, and half of the images were acquired in the shade. Ground truth was manually marked.

C. Performance measures

Metrics included the TPR (true positive rate, also noted as hit), FPR (false positive rate, also noted as false alarms) and the F score [16]. The TPR metric (Equation 1) states the number of correctly detected objects relative to the actual number of objects, while the FPR metric calculates the number of false objects detected relative to the actual number of objects (Equation 2). The F score (Equation 3), balances between TPR and FPR equally.

$$TPR = \frac{pixels\ detected\ correctly\ as\ part\ of\ fruit}{Actual\ number\ of\ pixels\ that\ represent\ the\ fruit} \quad (1)$$

$$FPR = \frac{false\ detected\ pixels}{Actual\ number\ of\ pixels\ that\ represent\ the\ fruit} \quad (2)$$

$$F(TPR, FPR) = \frac{2*(TPR*(1-FPR))}{TPR+(1-FPR)} \quad (3)$$

D. Analyses

Performance was evaluated while exploring the effect of using different FPR values instead of the F score on the algorithm accuracy. This analyses were important since most previous work in agricultural detection did not deal with reporting false alarms at all [17].

In addition, sensitivity analyses were conducted to evaluate the performance and robustness when adding random noise to the images and for slight changes in threshold values.

The noise was created by adding to each pixel in the RGB image a random number from the mean normal distribution for noise values up to 30%. The artificial noise represents the algorithms robustness toward other cameras with more noise, or when capturing images with different camera settings.

Evaluations were conducted using the leave one out method: eight images were used for the training process and one for testing. Average results are presented. Additional analyses were conducted using different sizes of training sets (using the "leave 1-5 out" method).

The algorithm performances were compared to several methods including: constant threshold, bimodal histogram, a constant threshold using the three NDI dimensions and segmentation using K-means with LAB and HSI color spaces.

III. Algorithm

A. Algorithm flow [3]

The offline process of the algorithm gets as input RGB images. Some areas of the images have more illumination than others, depending on the position of the light source and shading caused by leaves, branches and the covering net. In order to overcome this issue, the algorithm divides each image into multiple sub-images, having approximately homogenous illumination conditions (Fig. 2). These sub images are categorized into three illumination conditions: low, medium, and high. The illumination level is obtained by calculating the average on the gray scale sub images. The gray scale image shows values between zero (totally dark) and 255 (totally white). The chosen levels were selected empirically as 10, 70, and 130, corresponding to low, medium, and high level images based on manual image analyses. The high value was set as 130 so as to overcome the noise.

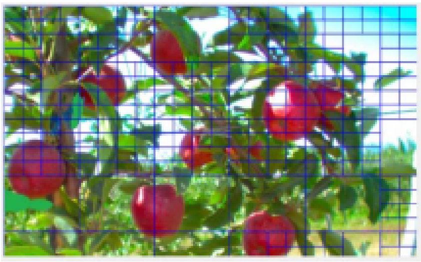


Fig. 2. Image split into sub-images - visualization

The algorithm transforms the RGB images to a 3D NDI space images. For each NDI dimension a binary image is created, where each pixel representing a fruit receives a value of one and all other pixels receive a value of zero (Fig. 4). Finally, the algorithm creates a ROC (receiver operator characteristics curve) representing TPR as a function of FPR [15] representing all the 9 thresholds learned from the offline process (Fig. 3).

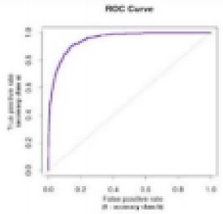


Fig. 3. ROC curve

In the online process the algorithm receives RGB images from the camera in real time and creates a binary image using the thresholds learned where white areas represent the fruits and the black areas represent the background (Fig. 5).

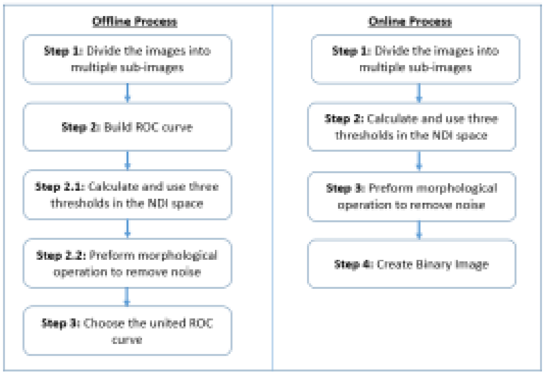

Offline Process
Step 1: Divide the images into multiple sub-images
Step 2: Build ROC curve
Step 2.1: Calculate and use three thresholds in the NDI space
Step 2.2: Preform morphological operation to remove noise
Step 3: Choose the united ROC curve
Online Process
Step 1: Divide the images into multiple sub-images
Step 2: Calculate and use three thresholds in the NDI space
Step 3: Preform morphological operation to remove noise
Step 4: Create Binary Image

Fig. 4. Flowchart of al algorithm

B. NDI color space exploration

As part of the algorithm process, the RGB image is transformed to an NDI image: each dimension in the NDI space is the normalized difference index between 2 colors (e.g. NDI first dimension represents the difference between red and green in RGB space, Fig. 5; second dimension represents the difference between red and blue in RGB space) as aforementioned, resulting in three dimensions (Equation 4). These operations are applied for all pixel locations in the image, creating a new image with this contrast index. These equations yield NDI values ranging between -1 and +1.

Using the NDI space in the dynamic thresholding process instead of the grey scale image enables us to learn 3D information about the threshold that best distinguishes between the background and the fruit.

$$Image_{D1} = \frac{R-G}{R+G}; Image_{D2} = \frac{R-B}{R+B}; Image_{D3} = \frac{B-G}{B+G} \quad (4)$$

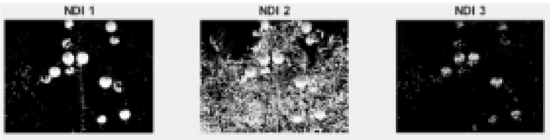


Fig. 5. Example for three dimensions of the NDI space

A binary image is created for each of the NDI dimensions (Equation 1 - D1, D2, and D3). Pixels categorized as part of a fruit are set equal to one; all other pixels are set to zero (Figure 6). In the binary output image of the algorithm pixels that were categorized as part of a fruit in all of the three binary images were set equal to one.

The use of pixels intersection from the three binary images helped reduce noise. However, this procedure can miss pixels that were correctly detected in one or two of the binary images.

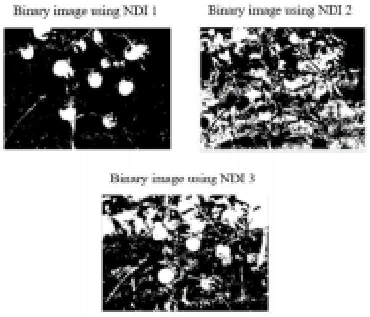


Fig. 6. Example of binary images results from NDI space

C. *Morphological operations*

The algorithm result (binary image) has noise in the shape of small clusters of pixels that were wrongly classified as fruits. In addition, some fruits are split between several clusters. Several morphological operations were performed to overcome these problems.

Morphological opening operations were executed using erosion followed by dilation with a neighborhood of 11×11 pixel squares. The square function was used since there was no pre-defined knowledge about the expected fruit orientation. To connect close clusters, the closing morphological operation was then applied by dilation followed by erosion implemented with a 5×5 pixel square neighborhood.

IV. RESULTS

A. *NDI and HSI color space exploration*

The use of NDI dimensions strongly affected performance as compared to HSI, as can be seen in Tables 1 and 2. The use of HSI space in the current algorithm, leads to higher FPR. The intersection between the binary images created by HSI space decreases the FPR, however it decreases the TPR as well. HSI obtains a higher F score (63.18%) which is obtained by the intersection between the S and I dimensions.

The best F score (91.78%) was obtained for the NDI first dimension (difference between red and green channels) which achieved maximum TPR-88.14% and relatively low FPR-3.05%. Although the intersection between the first and second NDI dimensions decreases FPR by 1.5%, it also decreases TPR by 1.15% and correspondingly the F score by 1.07% placing it as the second best performing algorithm.

B. *Noise removal process*

The noise removal process increased the F score by 1.02%, with increased 1.03% TPR and decreased 0.71% FPR (Table 3), consequently improving detection and minimizing false detection. The removal process increases computation time of both offline and online processes (Table 3). However, the online process is still relatively fast (26.86 sec on an Intel i7 2.6GHz laptop computer with 16GB memory).

C. *Sensitivity analysis*

Results (figure 7) reveal that when adding random noise to the image, noise values up to 10% hardly affect performance (TPR and FRP remain around 88% and 4.28% respectively with an F score over 91% for all noise values). When noise increases to above 30% F score goes below 90% (89.98 with TPR of 86.24 and FPR of 5.94%).

Table 1 – NDI color space exploration results –part 1

Metric		**Combinations of Intersection between binary images from NDI and HSI color spaces**		
		1	**2**	**3**
NDI	**FPR%**	4.27	41.16	30.68
	TPR%	88.14	81.31	66.65
	F%	91.78	68.27	67.96
HSI	**FPR%**	14.02	53.66	52.63
	TPR%	48.82	74.54	72.18
	F%	62.28	57.15	57.20

Results (Figure 8) reveal that the algorithm performances are not significantly influenced by changing the threshold value. Both TPR and FPR remain relatively stable with changes up to 8%. Changes above 10% cause decrease of TPR performance by 1.6%.

Analyses results (Table 4) indicate that when the threshold from the ROC curve maximizes the ratio TPR/FPR instead of the F score, FPR changed by less than 2%, however TPR decreased by 7.68%. In addition, choosing a threshold from the ROC with FPR=0.02 and FPR=0.04 improved TPR by 4.79% and 3.92% respectively. However, results indicate increased false alarms with FPR up to 10% and 16%.

Results shown in Table 5 indicate that changes in the training set do not influence the algorithm performances.

Table 2 NDI and HIS color space exploration results - part 2

Metric		Combinations of Intersection between binary images from NDI and HSI color spaces			
		1∩2	1∩3	2∩3	1∩2∩3
NDI	FPR%	2.82	1.69	0.29	0.29
	TPR%	76.46	62.64	51.01	51.01
	F%	85.59	76.52	67.49	67.4%
HSI	FPR%	3.49	12.52	19.85	3.16
	TPR%	37.48	39.39	52.14	29.24
	F%	54.00	54.32	63.18	44.92

Table 3 – Reduction of noise process results

Metric		Before Noise Removal	After Noise Removal
FPR%		4.27	3.04
TPR%		88.14	91.14
F%		91.78	93.96
Running Time%	Offline process	11.91 sec	556.77 sec
	Online process	5.27 sec	26.86 sec

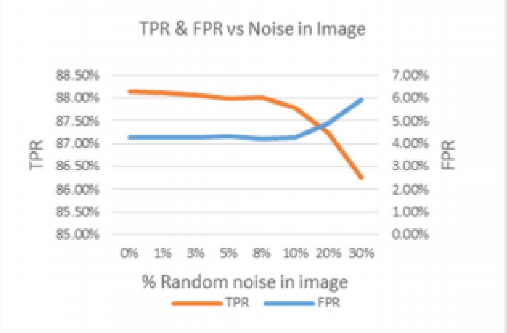


Fig. 7. TPR & FPR vs Noise in Image

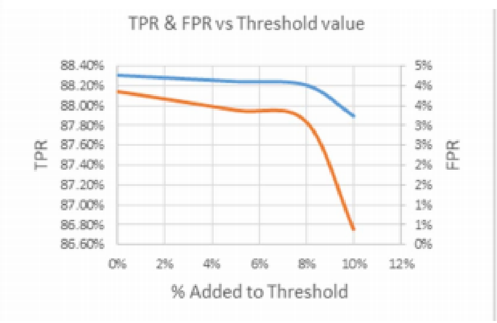


Fig. 8. TPR and FPR vs changes in threshold

Table 4 – Change of FPR effect on performances results

Metric	MAX(F)	$\mathrm{MAX}(\frac{TPR}{FPR})$	FPR=0.02	FPR=0.04
FPR%	4.27	1.72	16.48	10.16
TPR%	88.14	81.37	92.37	91.60
F%	91.78	89.03	87.72	90.71

Table 5 algorithm performances using "Leave 1-5 out "method

Metric	1	2	3	4	5
FPR%	4.27	4.27	4.07	4.29	4.16
TPR%	88.14	88.17	89.67	88.14	87.43
FPR%	91.78	91.78	92.69	91.77	91.44

Performance results of constant threshold, bimodal histogram and constant threshold 3D indicate that best performance was achieved by the dynamic thresholding algorithm for both TPR and the FPR (Table 6). The TPR was 4.9% better than the second best algorithm, which was, in this case, the bimodal histogram, with 10.9% difference in the FPR and an F-score over 90%.

Table 6 Performance results for different algorithms on apples real world orchard database

Metric	Constant threshold	Bimodal histogram	Constant threshold 3D	Current algorithm
FPR%	0.551	0.151	0.149	0.042
TPR%	0.509	0.832	0.819	0.881
F%	0.477	0.840	0.835	0.918

Best segmentation performance for K-Means algorithm was obtained with the use of the LAB color space with F-score 89.19%. Although the use of Hue dimension obtained the lowest FPR, TPR is less than 50% when used.

Table 7 Segmentation using K-Means over Lab and HSI color spaces

Metric	RGB	Lab	H	SI	H∩IS	H∪IS
FPR%	45.91	5.88	1.48	44.81	0.15	48.49
TPR%	72.46	85.48	48.27	66.19	33.66	83.45
F%	54.00	89.19	64.27	57.27	49.67	62.48

V. CONCLUSIONS AND FUTURE WORK

The algorithm successfully detects apples in variable lighting conditions resulting with an F-score of 93.96% using the NDI first dimension (with 91.14% correct detection and 3.04% false alarms). By applying thresholding with the NDI dimension improved results are obtained (as compared to 91.36% obtained by Vitzrabin 2016 [3]). The addition of a noise removal process on the images further increased TPR by 3% and decreased FPR by 29% resulting in increasing performance (F score increased by 2%). The algorithm is robust to changes in the threshold learned by the offline process and to noise effects

The dynamic set of three thresholds derived for different image regions and for a changing task objective enables to apply the algorithm for improved robotic harvesting detection. Adapting the threshold to image regions with homogenous lighting conditions improved TPR and FPR measures. Ongoing research is focused on optimizing the threshold selection process and expanding the algorithm for detecting additional agricultural objects in parallel to its implementation on a sweet pepper harvesting robot.

VI. Acknowledgments

This research was partially supported by the Helmsley Charitable Trust through the Agricultural, Biological and Cognitive Robotics Center, by the by the Marcus Endowment Fund and by the Rabbi W. Gunther Plaut Chair in Manufacturing Engineering, both at Ben-Gurion University of the Negev.

VII. References

[1] K. Kapach, E. Barnea, R. Mairon, Y. Edan, and O. Ben-Shahar, "Computer vision for fruit harvesting robots—State of the art and challenges ahead," Int. J. Comput. Vis. Rob., vol. 3, no. 1, pp. 4–34, 2012.

[2] A. Gongal, S. Amatya, M. Karkee, Q. Zhang, and K. Lewis, "Sensors and systems for fruit detection and localization: A review," Comput. Electron. Agric., vol. 116, pp. 8–19, 2015.

[3] E. Vitzrabin, "Adaptive local 3D thresholding for fruit harvesting robots," Ph.D. thesis, Dept. Ind. Eng. Manage., Ben-Gurion Univ. of the Negev, Israel, 2015.

[4] J. H. Park, G. S. Lee, W. H. Cho, N. Toan, S. H. Kim and S. Y. Park, "Moving Object Detection based on Clausius Entropy," 2010 10th IEEE International Conference on Computer and Information Technology, Bradford, 2010, pp. 517-521.

[6] M. Hannan, T. Burks, and D.M. Bulanon, "A real-time machine vision algorithm for robotic citrus harvesting," in Proc. IEEE Annu. Conf. Amer. Soc. Agric. Biol. Eng., 2007, pp. 1–11, Paper no. 073125.

[6] A. H. Gunatilaka and B. A. Baertlein, "Feature-level and decision-level fusion of noncoincidently sampled sensors for land mine detection," IEEE Trans. on Pattern IEEE Trans. Pattern Anal. Mach. Intell., vol. 23, no. 6, pp. 577-589, Jun 2001.

[7] P. Kanungo, P. K. Nanda and A. Ghosh, "Parallel genetic algorithm based adaptive thresholding for image segmentation under uneven lighting conditions," 2010 IEEE Intl. Conf. on Systems, Man and Cybernetics, Istanbul, 2010, pp. 1904-1911.

[8] D. L. Hall and S. A. H. McMullen, "Mathematical Techniques in Multisensor Data Fusion,"Artech House , vol. 2. p. 466, 2004.

[9] RN, Shebiah. Fruit Recognition using Color and Texture Features. Journal of Emerging Trends in Computing & Information Sciences,vol.1.2, no.1,pp.90-94,2010.

[10] D. M. Bulanon;, T. Kataoka, Y. Ota;, and T. Hiroma, "A Segmentation Algorithm for the Automatic Recognition of Fuji Apples at Harvest D.M.," J. Agric. Eng. Res., vol. 79, no. 3, pp. 265–274, 2001.

[11] Zheng, Liying, Jingtao Zhang, and Qianyu Wang. "Mean-shift-based color segmentation of images containing green vegetation." Computers and Electronics in Agriculture, vol 65 no. 1, pp. 93-98, 2009.

[12] A. Weeks and G. Hague, "Color Segmentation in the HSI Color Space Using the k-Means Algorithm," Proc. SPIE, vol. 3026, pp. 143-154, Feb. 1997.

[13] D. M. Woebbecke, G. E. Meyer, K. Von Bargen, and D. A. Mortensen, "Plant species identification, size, and enumeration using machine vision techniques on near-binary images," Opt. Agric. For., vol. 1836, 1992.

[14] Shrestha, D. S., B. Steward, and E. R. I. C. Bartlett. "Segmentation of plant from background using neural network approach." Intelligent Engineering Systems through Artificial Neural Networks: Proc. Artificial Neural Networks in Engineering (ANNIE) Intl. Conf.. Vol. 11. sn, 2001.

[15] M. Siegel, "Objective evaluation of subjective decisions," in Proc. IEEE Int. Workshop Soft

Comput. Techn. Instrum. Meas. Relat. Appl. (SCIMA'03), May 2003, pp. 14–18.

[16] Goutte, Cyril, and Eric Gaussier. "A probabilistic interpretation of precision, recall and F-score, with implication for evaluation." In European Conference on Information Retrieval, pp. 345-359. Springer Berlin Heidelberg, 2005.

[17] W. C. Bac, J. E. Henten, J. Hemming, and Y. Edan, "Harvesting robots for high-value crops: State-of-the-art review and challenges ahead," J Field Rob., vol. 31, no. 6, pp. 888–911, 2014.

[18] Marques, Blanca NP, Pina P, "Pattern Recognition and Image Analysis" Singh, S.; Singh, M.; Apte, C.; Perner, P. Estoril, Portugal, vol. 2, 2005.

[19] Shmmala, Faten Abu, and Wesam Ashour. "Color based image segmentation using different versions of K-Means in two Spaces." Global Advanced Research Journal of Engineering, Technology and Innovation vol. 1, no. 1. pp. 030-041. 2013

Appendix 2
Automatic parameter tuning for adaptive thresholding in robotic fruit detection

(Zemmour et al., under review)

Automatic parameter tuning for adaptive thresholding in robotic fruit detection

Elie Zemmour, Polina Kurtser, Yael Edan

Department of Industrial Engineering and Management, Ben-Gurion University of the Negev

Beer Sheva 84105, Israel

Abstract - This paper presents an automatic parameter tuning process for a dynamic adaptive thresholding algorithm for fruit detection of a robotic harvester. The algorithm enables robust detection in highly variable lighting conditions. The image is dynamically split into variable sized regions, where each region has approximately homogeneous lighting conditions. Nine thresholds were selected to accommodate three different illumination levels for three different dimensions in four color spaces: RGB, HSI, NDI and LAB. The thresholds were selected by quantifying the required relation between the true positive rate and false positive rate. A tuning process was developed to determine the best fit values of the algorithm parameters to enable easy adaption to different fruits, colors and illumination conditions. Extensive analyses were conducted on three different databases: red apples (9 images of 113 apples), green grapes (129 images of 1078 grapes) and yellow peppers (30 images of 73 peppers) acquired in outdoor conditions. Results show the importance of the tuning process for the generalization of the algorithm to different kinds of fruits and environments. In addition, this research revealed that for each kind of fruit the use of a different color space might be superior over the others.

Keywords: Adaptive thresholding, Fruit Detection, Robotic harvesting, Parameter tuning

1. Introduction

Robotic harvesting systems can increase agricultural productivity by reducing manual labor and production costs (Bac et al., 2014). When developing a robotic harvester, a basic step is to be able to identify the fruit (Kapach et al., 2012; Gongal et al., 2015). Despite intensive research conducted in identifying fruits, implementing a real time vision system remains a complex task (Kapach et al., 2012; Gongal et al., 2015). Current detection is limited to 87-88% detection rate with 3.8% false alarms (Vitzrabin and Edan, 2016; Bac et al., 2014, Luo et al., 2018).

Features such as shape, texture and location, are subject to high variability in the agricultural domain (Gongal et al., 2015). Moreover, fruits grow in an unstructured environment with highly variable lighting conditions (Vitzrabin and Edan, 2016) and obstructions (Barth et al., 2016) that influence detection performance. Color and texture are fundamental characteristics of natural images and play an important role in visual perception (Arivazhagan et al., 2010). Images can be represented by different color spaces (e.g., RGB, HIS, LAB, NDI); each one emphasizes different color features (Arivazhagan et al., 2010). RGB is the most common color space, representing each pixel in the image in three color channels as acquired: red, green and blue. HSI represents every color with three components: hue (H), saturation (S) and intensity (I), also known as HSV (Zheng et al., 2009). LAB color space is an approximate of human vision (Shmmala and Ashour,2013). An additional color space commonly employed in the agriculture field (Vitzrabin and Edan, 2016) is the normalized difference index (NDI) space. The NDI is used to differentiate between fruit and the background (Woebbecke et al., 1992) since it helps to overcome changes in illumination and shading due to its normalization technique (Shrestha, 2014). Each dimension in the NDI space is the normalized difference index between 2 colors in the RGB space, resulting in three dimensions (Equation 1). These operations are applied for all pixel locations in the image, creating a new image with this contrast index. These equations yield NDI values ranging between -1 and +1.

$$NDI_1 = \frac{R-G}{R+G}\,; NDI_2 = \frac{R-B}{R+B}; NDI_3 = \frac{B-G}{B+G} \qquad (1)$$

One of the most common methods for fruit detection is image segmentation (e.g., Wang et al., 2013; Jiang et al., 2008; Arroyo, et al., 2016; Rong, et al., 2017). Many segmentation algorithms have been developed (Zhang, 1996) including: Kmeans (Shmmala and Ashour, 2013); Mean shift analysis (Zheng et al., 2009); Artificial neural networks (ANN) (Al-allaf, 2014), Support vector machines (SVM) (Sakthivel et al., 2015), Deep learning (Sa et al., 2016), Reinforcement learning (RL) (Ostovar et al., 2018) and several others.

This research focuses on segmenting objects in the image using an adaptive thresholding method. Observing the histogram of the image color implies that a threshold can be determined to best differentiate between the background and the object distributions (Park et al., 2011). The threshold is computed by finding the histogram minimum (Fig. 1) separating between two peaks – the object and the background. However, the global minimum between the distributions is very hard to find in most cases (Hannan et al., 2007).

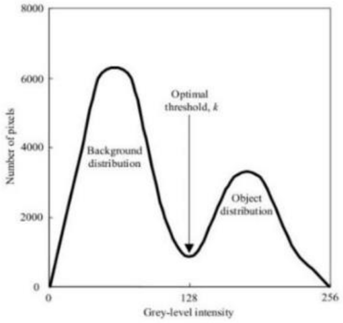


Fig. 1. Optimal threshold in Bimodal histogram

*Corresponding author

E-mail address: eliezemmour@gmail.com (E.Zemmour).

Current most optimal thresholding algorithms determine the threshold only in a one-dimensional space, for example in the RGB space, either R or G or B or a linear combination of their values (e.g. grayscale transformation) will be used (Bulanon; et al., 2001). In the transformation from three colors dimensions into one, information is lost. In this research, a three-dimensional thresholding algorithm based on (Vitzrabin and Edan, 2016) was applied and evaluated also for additional color spaces (RGB, NDI , HSI and LAB color spaces) – a threshold is determined for each dimension in the color space.

There are two common adaptive thresholding algorithm concepts: 1) global thresholding, in which for each image, a different threshold is determined according to specific conditions for the entire image that is then transformed into a binary image; 2) local thresholding, in which the image is divided into sections, and a different threshold is calculated for each section; the sections are then combined to a binary image.

There are several methods that utilize dynamic local thresholding algorithms (Gunatilaka and Baertlein, 2001; Kanungo et al., 2010). A common approach is to use multi-resolution windows which apply a bottom up method, merging pixels while a criteria is met (Gunatilaka and Baertlein 2001; Kanungo et al., 2010). Another approach is the top down method, where the image is divided into sub regions according to criteria. The top down approach reduces execution speed and improves generalization (Hall and McMullen, 2004) and was therefore used in this research.

The adaptive thresholding algorithm presented in this paper is based on previous work (Vitzrabin and Edan, 2016) that aimed to detect peppers, in which a set of three thresholds were determined for each region of the image according to its lighting setting. The algorithm dynamically divides the image into several regions, each with approximately the same lighting conditions. The main contribution of the adaptive local 3D thresholding is a very high true positive rate (TPR) and low false positive rate (FRP) in the fruit detection task in an unstructured, highly variable, and dynamic crop environment. Another contribution is the ability to change in real time the task objective in the algorithm behavior based on the desired ratio between TPR and FRP; this contributes to a better success rate in the grasping operation itself (Vitzrabin and Edan, 2016). Both the high performances and the changing task objectives are key issues regarding detection in robotic harvesting applications (Vitzrabin and Edan, 2016). Previous research (Zemmour et al., 2017) presented preliminary results of this algorithm for apple detection.

The current paper advances previous research (Zemmour et al., 2017) with several new contributions: 1) A new parameter tuning process developed to best-fit the parameters to the specific database. 2) Intensive evaluation of the adaptive thresholding algorithm for different color spaces. 3) Application and evaluation of the algorithm to different kinds of fruit. 4) Comparing the contribution of the new developments (items 1-2) to previous developments.

2. Materials and methods

2.1. Databases

The algorithm was evaluated on three databases representing three different fruit colors: red (apples), green (grapes) and yellow (peppers) for two environmental settings (greenhouse, field) in different illumination conditions. Images were acquired with different cameras.

Apples - The orchard apples database includes 113 "Royal Gala" apples in 9 images acquired from an orchard in Chile in March 2012 under natural growing conditions with a Prosilica GC2450C camera with 1536x2048 resolution; the camera was attached to a pole. The images were captured in daylight: half of the images were acquired under direct sunlight, and half of the images were acquired in the shade. Ground truth was manually marked (Fig 2).

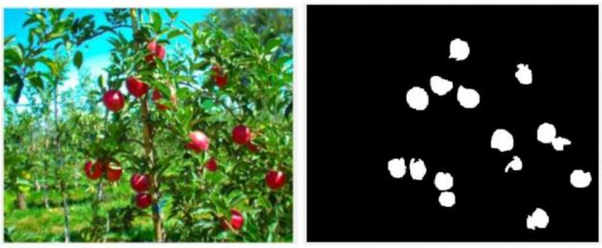


Fig. 2. Apples image RGB image (left) and ground truth (right) example

Grapes - The images used originated from a commercial vineyard growing green grapes of the "superior" variety. An RGB camera (Microsoft NX-6000) with 600×800 resolution was manually driven at mid-day along a commercial vineyard in Lachish, Israel, during the summer season of 2011, one month before harvest time. The images were captured from 5 different growing rows. The targets were defined as the grape clusters. A group of three experts was guided to mark the closing perimeter of each grape cluster in the image. The final ground truth was marked using the judge rules criteria (if a given pixel was marked by two or more experts, it was considered a target). A set of 129 images were marked using this technique and used as a ground truth for the following research. The images included 1078 grape clusters (Fig 3).

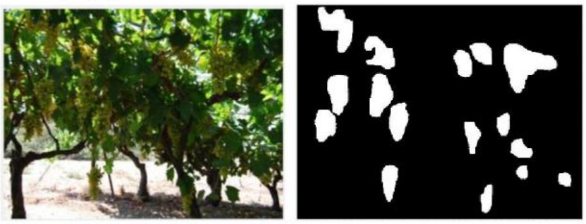


Fig. 3. Grapes image RGB image (left) and ground truth (right) example

Peppers – The dataset includes 30 images of 73 yellow peppers acquired in a commercial greenhouse in Ijsselmuiden, Netherlands using a 6 degree of freedom manipulator (Fanuc LR Mate 200iD/7L), equipped with an iDS Ui-5250RE RGB camera with 600x800 resolution. The images were manually marked in order to compare the algorithm performances to manual detection by a human labeler. The database was marked twice, one time marking only peppers with high visibility (denoted as "high visibility peppers", this was done for 10 images of 25 yellow peppers) and a second time marked as well peppers in dark areas that are less visible in the image (will be refer as "including low visibility peppers", done for all 30 images) (Fig 4,5).

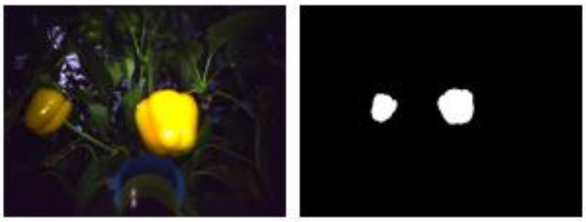


Fig. 4 Peppers RGB image (left) and ground truth (right) example

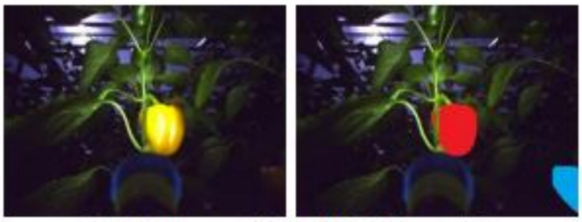


Fig. 5 RGB image (right) and labeled image (left).
"High visibility peppers" marked in red and "Low visibility peppers" marked in blue

2.2. *Performance measures*

Metrics included the TPR (true positive rate, also noted as hit), FPR (false positive rate, also noted as false alarms) and the F score (Goutte & Gaussier, 2005). The TPR metric (Equation 2) states the number of correctly detected objects relative to the actual number of objects, while the FPR metric calculates the number of false objects detected relative to the actual number of objects (Equation 3). The F score (Equation 4), balances between TPR and FPR equally.

$$\mathrm{TPR}=\frac{pixels\ detected\ correctly\ as\ part\ of\ fruit}{Actual\ number\ of\ pixels\ that\ represent\ the\ fruit} \qquad (2)$$

$$FPR = \frac{false\ detected\ pixels}{Actual\ number\ of\ pixels\ that\ represent\ background} \qquad (3)$$

$$F(TPR,FPR) = \frac{2*(TPR*(1-FPR))}{TPR+(1-FPR)} \qquad (4)$$

2.3. *Analyses*

The following analyses were conducted for the three databases, apples, grapes and peppers, using 70% of the data for training and 30% for testing (Guyon & Isabelle, 1997). This rate was chosen to be more rigid to the algorithm performances since the number of images in each DB were relativity small. In addition, to ensure robustness of the results each test was repeated five times with random section of images for each repetition and average results reported.

- **Tuning parameters**: parameters were computed for each database with procedures defined in 3.3 and compared to previous predefined parameters.
- **Color spaces analyses**: algorithm performances were tested on all databases for four different color spaces: NDI, LAB, HSI, RGB.
- **Sensitivity analyses:** sensitivity analyses were conducted for all the databases and included:
  a) Noise - noise was created by adding to each pixel in the RGB image a random number from the mean normal distribution for noise values up to 30%. The artificial noise represents the algorithms robustness toward other cameras with more noise, or when capturing images with different camera settings. Noise values of 5%, 10%,20%,30%, were evaluated.
  b) Thresholds learned in offline process - thresholds were changed by ±5%, ±10% and ±15% according to the threshold in each region.
  c) Stop condition - the selected std value was changed by 5%, 10% to test the robustness of the algorithm to these parameters.
  d) Train vs. Test -The algorithm performances were evaluated while using different percentage of DB images for the training and testing processes.
- **Morphological operation contribution** – performances were tested for imaging with and without the morphological operations process.

3. Algorithm

3.1. Algorithm flow

The RGB images are the inputs for the offline process. Some areas in the images contain more illumination than others, depending on the position of the light source and shading caused by leaves, branches and the covering net when exists. To overcome this issue, the algorithm divides each image into multiple sub images, with approximately homogenous illumination conditions (Fig. 6). These sub images are categorized into three illumination conditions: low, medium, and high. The illumination level is obtained by calculating the average on the gray scale sub images. The gray scale image shows values between 0 (completely dark) and 255 (completely white). In the previous algorithm (Vitzrabin and Edan, 2016), the sub images were categorized into groups using levels selected empirically as 10, 70, and 130, corresponding to low, medium, and high level images based on manual image analyses. The high value was set as 130 in order to filter overexposed areas in the images. In the current algorithm a tuning parameter process (detailed in section 3.3) is developed to determine these three values.

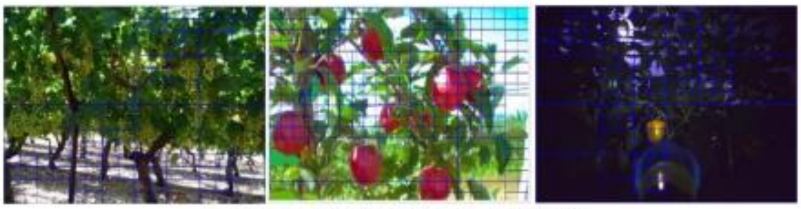


Fig. 6. Image split into sub-images - visualization

The algorithm then creates a 3D color space image (can transform the RGB image to NDI, HSI, LAB space or uses directly the RGB space). For each color dimension a binary image (mask) is created, where each pixel that represents the fruit receives a value of one and all other pixels receive a value of zero. Finally, the algorithm creates a ROC (receiver operator characteristics curve) representing TPR as a function of FPR (Siegel and Wu 2003) representing all the nine thresholds learned from the offline process. Fig. 7 presents an example of nine ROC curves computed for three sub images with different light levels (L1, L2, L3) in the NDI color space. In this example, the sub image with light level 2 (L2) in the first NDI dimension obtained the best performances (high TPR and low FPR).

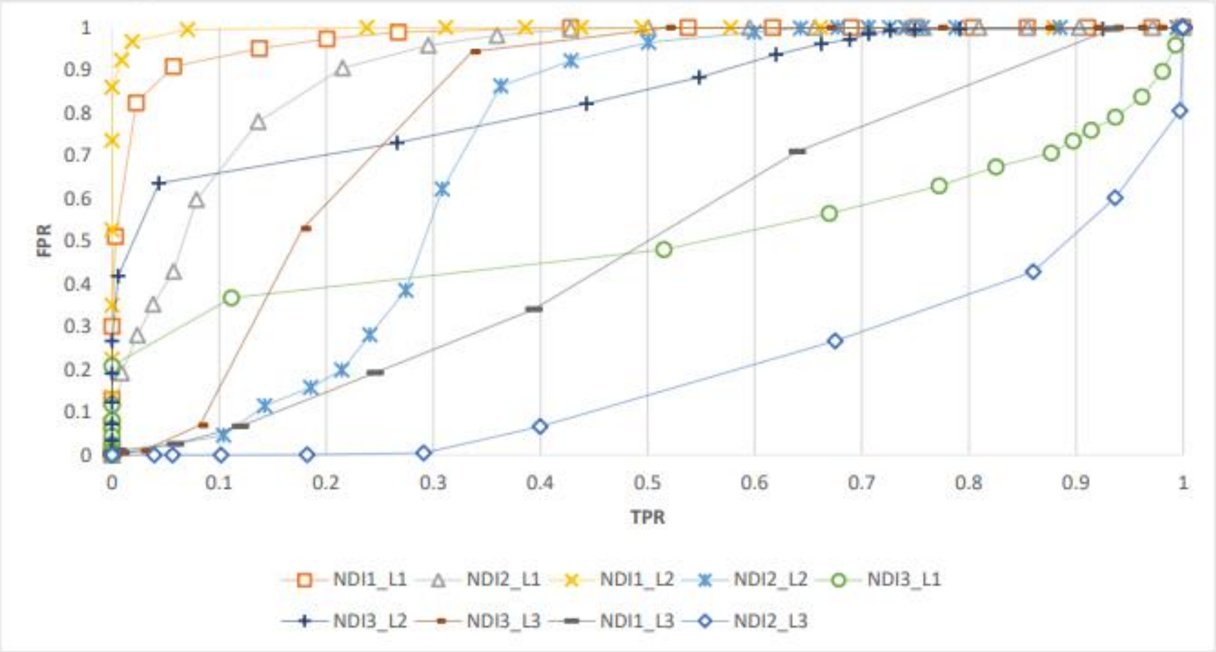


Fig. 7. 9 ROC curve - 3 Dimensions X 3 Light levels

NDIi_Lj I represents the color space dimension; j represents the illumination level

In the online process, the algorithm receives RGB images from the camera in real time, transforms the representation to the relevant color space (NDI/HSI/LAB) and creates a binary image by applying the thresholds as following: three thresholds, one for each dimension are calculated from the nine thresholds learned by linear interpolation between two of the three illumination regions (Low, Medium, and High) selected as closest to the calculated illumination level for the specific sub-image from the grayscale image and using Equation 5.

$$T = \frac{T(closest\ from\ below)*(Light\ level - Ligh\ from\ below) + T(closest\ from\ above)(Light\ level\ from\ above - Light\ level)}{Light\ from\ above - Light\ from\ below} \qquad (5)$$

For example, if the current light level is 40, and the thresholds in the offline process for the Low, Medium and High light levels were 10, 70 and 130, the threshold would be calculated in the following way:

$$T = \frac{T(10) * (40 - 10) + T(70)(70 - 40)}{70 - 10}$$

The end of the process results in a binary image where white areas (1 value pixels) in the binary image represent the fruits and the black areas (0 value pixels) represent the background (see Fig.2,3,4).

3.2. Morphological operations

The algorithm result is a binary image with major fruit detected and small clusters of pixels that were wrongly classified as fruits (e.g., Fig 8 – present figure with this before and after morphological operation). In addition, some fruits are split between several clusters (e.g., Fig. 8). Several morphological operations were performed to overcome these problems.

Morphological opening operations were executed based on previous research which indicate their contribution (Vitzrabin and Edan, 2016) using the same method: erosion followed by dilation with a neighborhood of 11×11-pixel squares. The square function was used since there was no pre-defined knowledge about the expected fruit orientation. To connect close clusters, the closing morphological operation was then applied by dilation followed by erosion implemented with a 5×5-pixel square neighborhood.

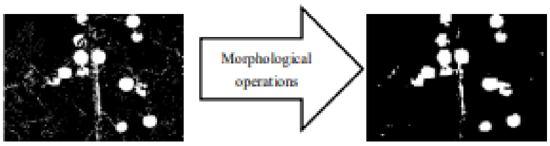


Fig. 8 Morphological operation

3.3 Parameter tuning

The algorithm uses several parameters that influence the algorithm performances: T1, T2, Std, Classification rule direction- D1/D2. The following parameter tuning procedure (Fig. 10) was developed and should be performed when exploring images from a new database, new operating conditions (cameras, illumination) or when exploring a new color space. The parameters are:

Light level thresholds (T1, T2) –The algorithm splits the images into sub images set to 1% of the entire image. Then, the algorithm computes the light level of each sub-image by calculating the average pixels values of the grayscale sub-image. Finally, the algorithm groups the sub images into three light level categories (see Fig. 9) using two thresholds as presented in Equation 6.

$$f(x) = \begin{cases} \text{Low}\,,\ 0\ < x < T1 \\ \text{Medium}\,,\ \text{T1} < x < \text{T2} \\ \text{High}\,,\ x > \text{T2} \end{cases} \qquad (6)$$

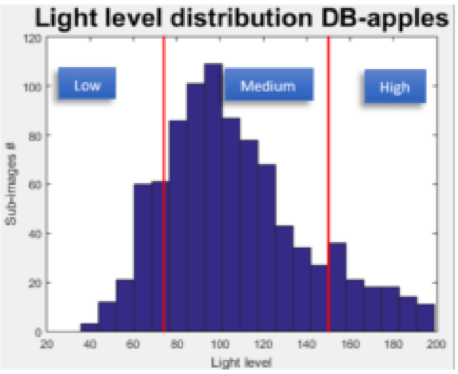


Fig. 9. Sub-images level of light distribution

Research was done to identify the PDF function of the data distributions of each database through a χ^2 goodness of fit tests. However, due to the lack of significant level in the tests the thresholds were selected empirically as follows: T1 and T2 were chosen so that 15% of the data will be categorized as Low, 15% as High and 70% will be categorized as Medium.

Note that as described in the algorithm flow, the algorithm uses a third threshold. Sub-images above that threshold are ignored in the training process due to their high values the sub images (are almost completely white).

Stop splitting condition (std) - the algorithm splits an image into sub-images until the sub image achieves a predefined standard deviation (STD) value. This approach assumes that a larger sub-image contains higher STD value. To test this assumption, STD was calculated for different sizes of sub images for the different databases. The stop condition value (STD minimum value) is determined by maximizing the F score (Equation 4).

Classification rule direction– (D1, D2, D3) as detailed in the introduction, in the thresholding method, a value that differentiates the intensity of the object from its background is determined. When using different color spaces, one of the issues encountered was to determine for each color dimension if the intensity of the object is greater or smaller than the background. This information was learned as part of the tuning process. For this step a simple heuristic rule was used as follows based on the assumption that the images contain more background pixels than objects: 1. Execute **image>Threshold**. 2. If the pixels categorized as background represent less than 70% of the image, reverse the thresholding direction **images <Threshold**.

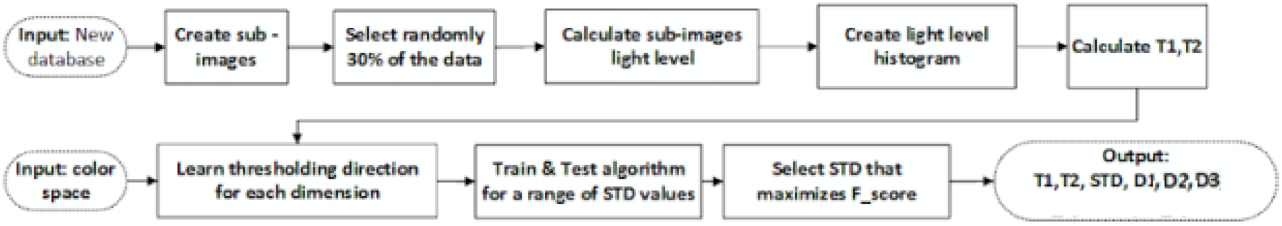

Input: New database
Create sub - images
Select randomly 30% of the data
Calculate sub-images light level
Create light level histogram
Calculate T1,T2
Input: color space
Learn thresholding direction for each dimension
Train & Test algorithm for a range of STD values
Select STD that maximizes F_score
Output: T1,T2, STD, D1, D2, D3

Fig. 10 Parameter tuning process

4. Results and discussion

4.1 Sub-image size vs. STD value

Fig. 11 confirms the assumption that splitting an image to small sub images (small S) decreases the average STD of the sub images (in all three databases).

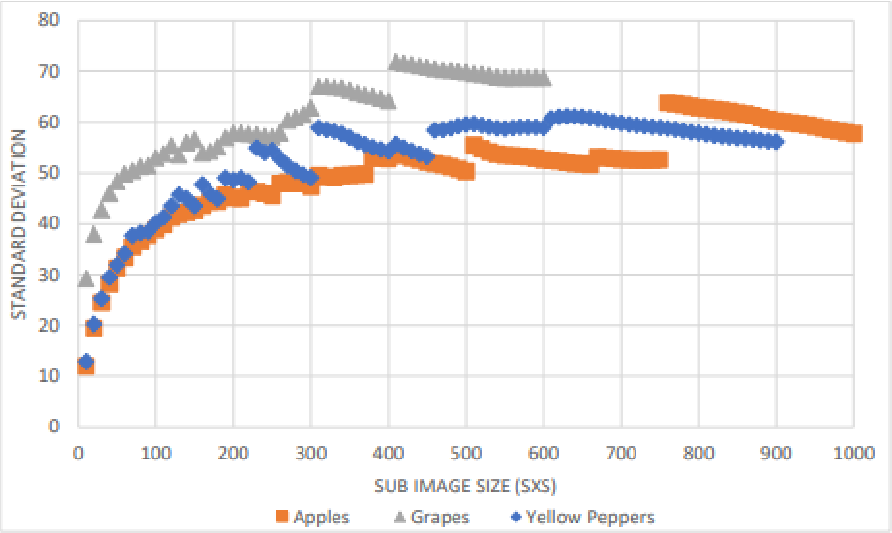


Fig. 11. Sub image size vs average STD

4.2 Tuning process

This section presents the tuning process results, including thresholds derived to categorize the sub images into light level groups as well as the recursive stop condition that achieved best result for each database.

Light level distribution: The light level distribution was computed for each database (Fig. 12) along with T1 and T2 (Table 1). The variation in the light distributions between the different databases are described in Table 2. The variance of light in the grapes databases is significantly higher than in both the apples and the peppers databases, the pepper database is significantly darker and highly skewed. Therefore, for each database T1 and T2 were significantly different implying the importance of the tuning process.

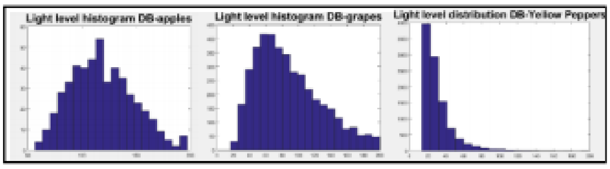


Fig. 12. Light level distribution was computed for each database

Table 1 T1 and T2 values determined for each database

Measure/DB	Apples	Grapes	Peppers
T1	84	49	18
T2	140	130	47

Table 2 Descriptive statistics of the different light distributions

Measure/DB	Apples	Grapes	Peppers
Mean	118.46	88.00	32.09
Std	28.04	37.90	18.92
Skewness	0.40	0.68	3.16
Kurtosis	-0.17	-0.13	15.36
Median	116.31	81.06	26.93

Stop splitting condition: Using a low STD value as a stop condition, increases performance (Fig. 13). This happens since smaller sub images contain less illumination differences. However, small STD values can create also too small sub-images which may not contain fruit and background pixels in the same frame. In these cases, the algorithm cannot learn a threshold that could differ between them. Additionally, results reveal that when using high STD values the performances stays constant. This happens since beyond a certain value the algorithm does not split the image even once.

As part of the parameter tuning process, the STD value is selected by testing the performances of a range of STD [0,100]. For each STD value the algorithm runs five iterations were it randomly select P% of the images, from the selected images it uses 70% for train and 30% test. Final selected STD values are presented in Table 3 for each database and color space (using P=30% and 50%).

Classification rule direction: as shown in Table 3, the direction of the classification rule in the thresholding process can be different for each color dimension, therefore this must be learned as part of the tuning process.

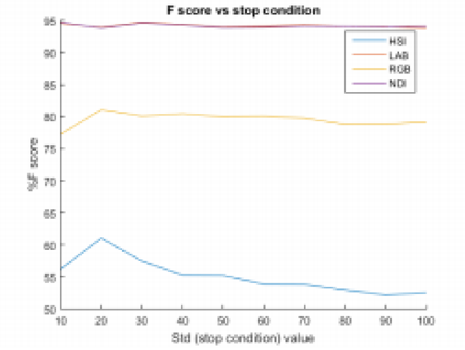


Fig. 13. F-score vs. increasing STD value as stop condition for the recursive function on apples DB

Table 3 STD value chosen for each database and color space

DB		Apples				Grapes				Peppers			
Color space		**HSI**	**LAB**	**NDI**	**RGB**	**HSI**	**LAB**	**NDI**	**RGB**	**HSI**	**LAB**	**NDI**	**RGB**
STD (P=30%)		20	30	10	20	10	20	60	20	100	10	10	10
STD (P=50%)		20	10	10	30	20	20	70	20	100	20	10	10
Classification rule direction	D1	>	<	>	>	<	>	>	>	<	>	>	>
	D2	>	>	>	<	>	<	>	>	>	>	>	>
	D3	>	>	>	<	<	>	<	<	>	>	<	<

4.3 Color spaces analyses

In this section algorithm performances results are presented in figure for each color space followed by a table representing the best color space performances including performances for all color space dimensions combinations.

Apples – Results (Fig 14) reveal that NDI and LAB color spaces result in similar best performances. In Table 4 the preferences for each dimension in the NDI color space and the performances when using the intersection between them is shown. The NDI first dimension (see Equation 1) represents the difference between the red and green colors in the image. The objects in this database are red apples and most of the background is green leaves therefore, as expected, the first NDI obtained the best F of 93.17%. In the LAB color space, results (Table 5) reveals that the second dimension (A) yields the best F score of 93.19%.

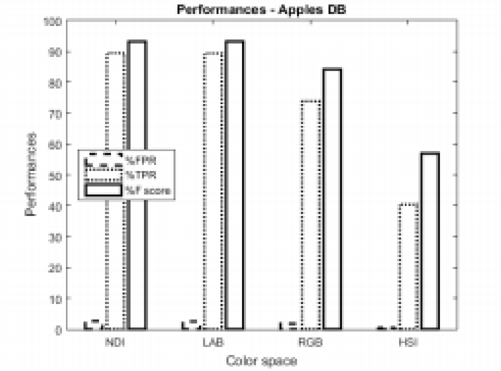


Fig. 14. Color spaces performances - Apples DB

Table 4 - Performances of each NDI dimension and intersections – Apples

Measure /Dimension	1	2	3	1∩2	1∩3	2∩3	1∩2∩3
% FPR	2.59	40.91	31.38	1.64	1.32	2.48	0.48
% TPR	89.45	83.53	68.39	78.52	64.82	54.65	54.10
% F	93.17	67.85	67.80	86.75	77.60	69.67	69.69

Table 5 - Performances of each LAB dimension and intersections – Apples

Measure /Dimension	1	2	3	1∩2	1∩3	2∩3	1∩2∩3
% FPR	33.58	2.45	77.08	1.78	28.55	1.55	1.07
% TPR	61.26	89.34	85.26	56.59	52.95	76.27	48.80
% F	56.79	93.19	35.85	69.02	54.61	85.37	62.58

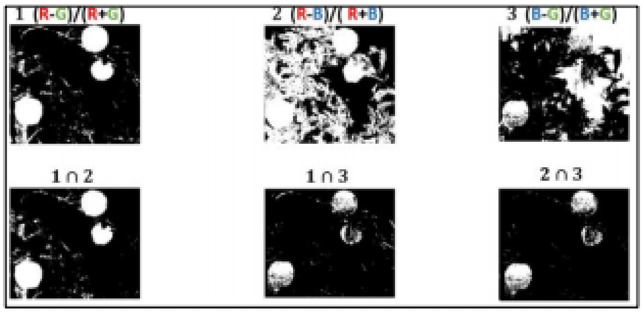


Fig. 15. Use of NDI dimension intersection to increase performance

Grapes – The NDI color space obtained the best result for grapes (Fig. 16) with a F score of 73.52%. The second-best color space is the LAB with a F score of 62.54% The best NDI results were obtained using the second dimension (Table 6).

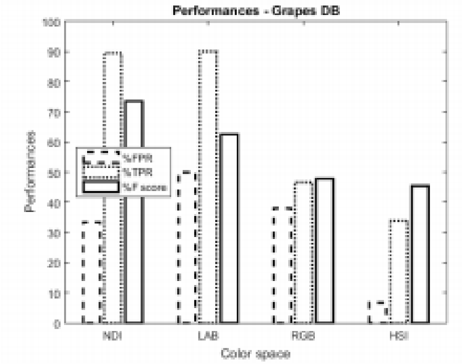


Fig. 16. Color spaces performances - Grapes DB

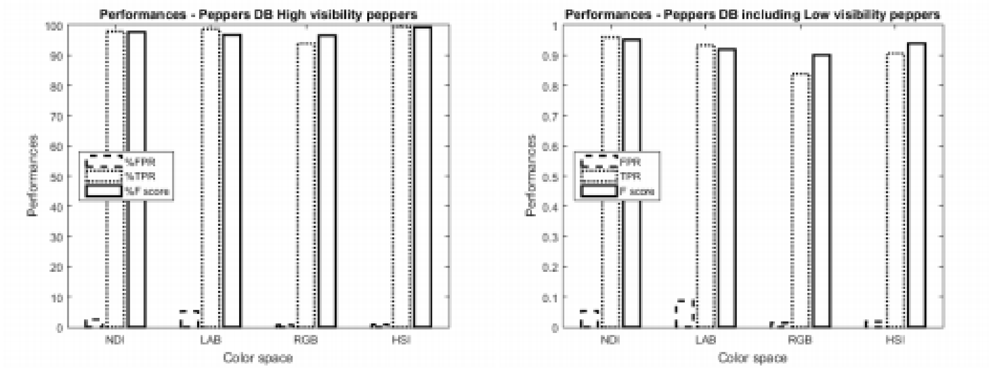
Table 6 - Performances of each NDI dimension and intersections – Grapes

Measure /Dimension	1	2	3	1∩2	1∩3	2∩3	1∩2∩3
% FPR	35.86	33.35	52.90	4.86	5.50	32.35	4.09
% TPR	44.52	89.48	89.99	38.53	37.27	87.50	36.70
% F	47.19	73.52	58.05	50.12	48.93	73.20	48.65

Peppers

High visibility – Fig. 17a indicates that HSI color space obtained the best results with relatively low FPR (0.81%) and very high TPR (99.43%) resulting in a high F score (99.31%). The second-best color space is NDI with FPR=2.48% and TPR=97.96% (F= 97.72%). The best HSI result, were obtained using the combination of the first and the second dimensions (Table 7).

Including low visibility – Fig. 17b indicates that NDI color space obtained the best results with relatively low FPR (5.24%) and very high TPR (95.93) resulting in high F score (95.19%). Although on the "high visibility" peppers HSI obtained the best performances, when trying to detect peppers in dark areas that are less visible, NDI shows better results. The best NDI result, were obtained using the intersection between the first and the second dimensions (Table 8).



(a) (b)

Fig. 17. Color spaces performances - Peppers DB

Table 7 - Performances of each HSI dimension and intersections – Peppers high visibility

Measure/DB	1	2	3	1∩2	1∩3	2∩3	1∩2∩3
% FPR	2.48	5.15	0.64	0.81	2.48	5.15	0.64
% TPR	97.96	98.75	93.91	99.43	97.96	98.75	93.91
% F	97.72	96.73	96.51	99.31	97.72	96.73	96.51

Table 8 – Performances of each NDI dimension and intersections – Peppers including low visibility

Measure /Dimension	1	2	3	1∩2	1∩3	2∩3	1∩2∩3
% FPR	66.57%	5.24%	9.23%	1.42%	1.24%	4.57%	0.99%
% TPR	85.61%	95.93%	92.49%	82.20%	78.64%	92.33%	78.61%
% F	46.96%	95.19%	91.24%	88.91%	86.59%	93.51%	86.67%

4.3 *Sensitivity analysis*

a) Noise – analysis shows that the algorithm is robust to noise in the image up to 15% in the apples and peppers databases (Fig. 18). The grapes images are more sensitive to noise and performance drops when noise values of 5% are added. Although better F score values were obtained for NDI and HSI for grapes and peppers, we can see that LAB color space yields more robust performance when adding noise to the images.

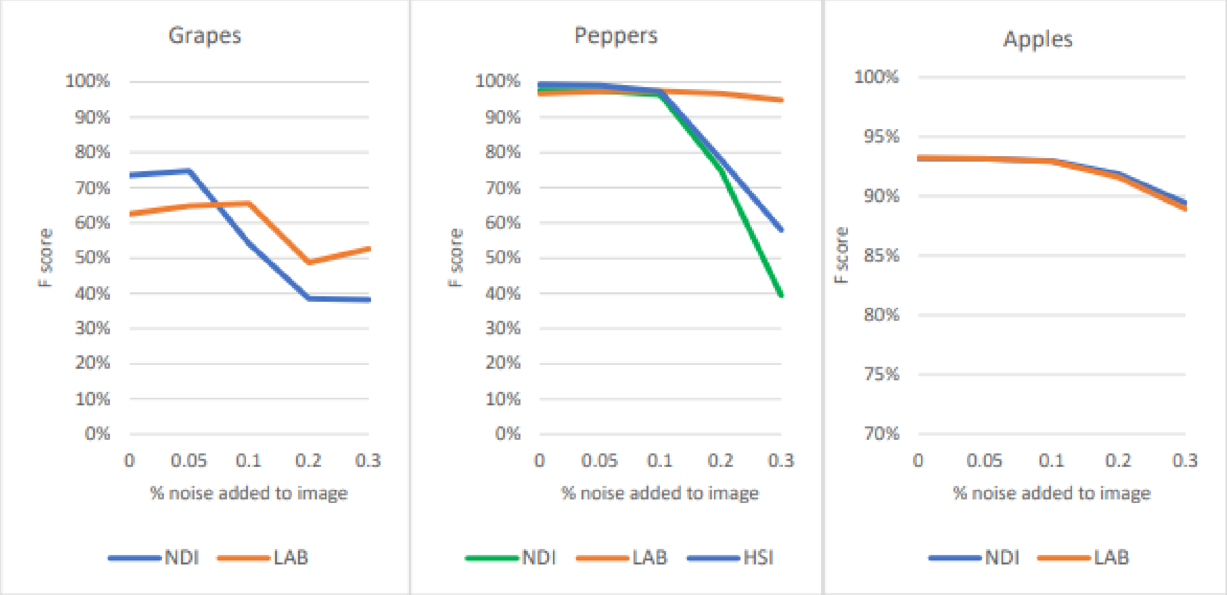


Fig. 18. Sensitivity analysis - adding noise to image

b) Thresholds learned in offline process - as expected, TPR decreases when the threshold values change. The algorithm is relatively robust to the change in the thresholds for apples and peppers. Performance in the grapes images is more sensitive to threshold changes and yields a significant decrease in TPR when increasing the threshold value (Table 9).

Table 9 – Threshold values changed by ±5%, ±10% and ±15% according to the threshold in each region

DB	Measure /Change in thresholds	-15	-10	-5	0	5	10	15
Apples	%FPR	3.58	3.43	3.30	2.59	3.06%	2.93	2.81
	%TPR	91.47	91.28	91.07	89.45	90.75%	90.57	90.44
Grapes	%FPR	21.59	18.40	15.53	33.35	11.00%	9.23	7.72
	%TPR	78.02	72.63	66.42	89.48	50.99%	43.63	36.24
Peppers	%FPR	0.98	0.91	0.86	0.81	0.78%	0.70	0.65
	%TPR	99.25	99.22	99.20	99.43	99.12%	99.07	99.04

c) Stop condition - the algorithm shows more robustness to apples and peppers images than grapes (Fig. 19).

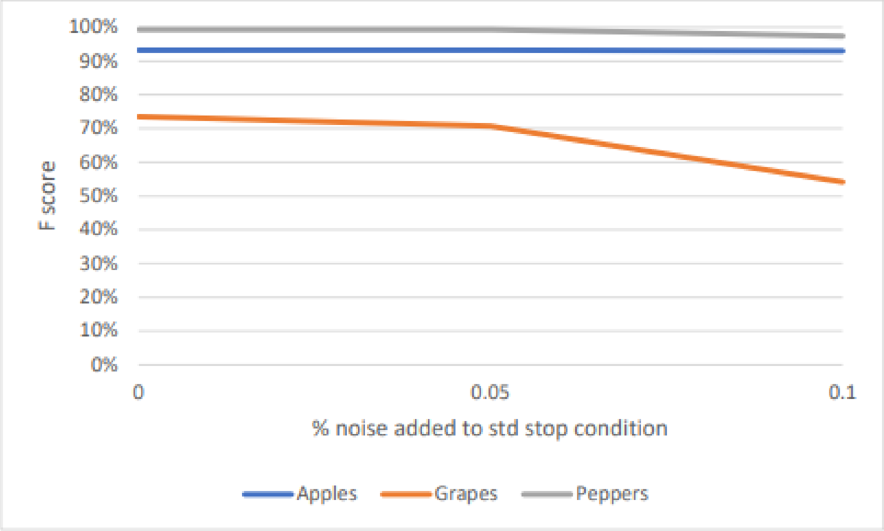


Fig. 19. Sensitivity analysis - adding noise to std stop condition

d) Train/Test – the expectation is that more training images leads to better performance until overfitting is accommodated. There is a clear increase in TPR however FPR increases as well at 80%,90% train.

Table 10 - Performances vs. different % images database as train set

DB	Measure / %Train	10	20	30	40	50	60	70	80	90
Grapes	%FPR	32.81	37.08	28.54	36.16	31.83	29.10	29.63	40.51	40.80
	%TPR	88.79	89.62	87.01	88.44	87.58	82.55	87.14	94.53	95.85
	%F	73.35	70.20	75.19	69.41	72.49	70.24	73.92	72.55	72.54

The tuning process resulted in increased performances for both the grapes and peppers databases with 40% and 1.49% increase respectively. The results for the apple databased were similar with only a 0.1% increase as expected (since this was similar to the database the previous parameters were derived from).

Table 11 – Parameter tuning contribution to algorithm performances

DB	Measure	Performances using Previous params	Performances using Tuning process
Apples	%FPR	2.53	2.59
	%TPR	89.23	89.45
	%F	93.08	93.17
Grapes	%FPR	18.63	33.35
	%TPR	63.70	89.48
	%F	67.30	73.52
Peppers	%FPR	1.00	0.81
	%TPR	97.97	99.43
	%F	98.47	99.31

The morphological operations process increases the F score by 2.85%,8.59%,2.71% for the apples grapes and peppers databases respectively (Figure 20).

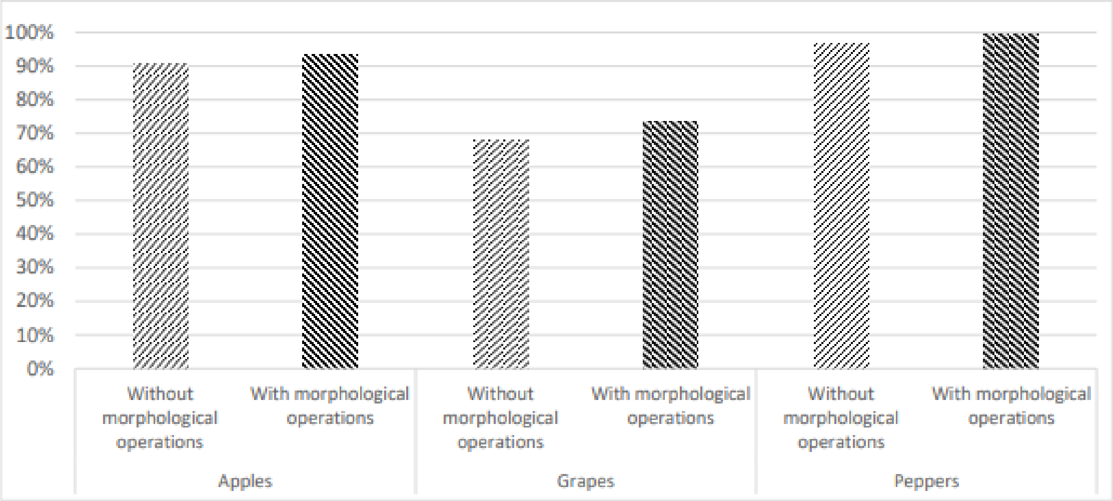


Fig. 20. Morphological operation contribution

5. Conclusions and future work

The algorithm successfully detected apples and peppers in variable lighting conditions with an F-score of 93.15% and 97.40% respectively, resulting in one of the best detection rates achieved to date in fruit detection to the best of our knowledge. Previous research has shown 85%-90% TPR (Bac et al., 2014; Vitzrabin and Edan 2016; Sa et al., 2016). The algorithm has shown less impressive results in the grapes database 73.52% due to difficulties associated with differentiating between green fruits and a green background (leaves). In this case, additional features (e.g. morphological operations fitted for grapes see Bernstein, Shahar, Shapiro, Edan, 2010) should be used to increase performance.

Different color spaces yielded best results for each fruit variety, implying that the color space must be analyzed and fitted to the specific fruit. The LAB color space is more robust to noise in images and hence should be used when images are of low quality. The algorithm is robust to changes in the threshold learned by the offline process and to noise effects in images. Morphological operations can improve performance and hence should be utilized.

The tuning process developed in this paper enables the previous algorithm (Zemmour, Kurtser, and Edan 2017) to adapt automatically to changing conditions/objectives (i.e. to detect other fruit with different colors and other outdoor conditions) and hence should be used for improved target detection in highly variable illumination conditions.

Acknowledgments

This research was partially supported by the European Commission (SWEEPER GA no. 66313) and by Ben-Gurion University of the Negev through the Helmsley Charitable Trust, the Agricultural, Biological and Cognitive Robotics Initiative, the Marcus Endowment Fund and the Rabbi W. Gunther Plaut Chair in Manufacturing Engineering.

REFERENCES

Al-Allaf, O.N., 2014. Review of face detection systems based artificial neural networks algorithms. arXiv preprint arXiv:1404.1292.

Arivazhagan, S., Shebiah, R.N., Nidhyanandhan, S.S. and Ganesan, L., 2010. Fruit recognition using color and texture features. Journal of Emerging Trends in Computing and Information Sciences, 1(2), pp.90-94.

Arroyo, J., Guijarro, M. and Pajares, G., 2016. An instance-based learning approach for thresholding in crop images under different outdoor conditions. Computers and Electronics in Agriculture, 127, pp.669-679.

Barth, Ruud, Jochen Hemming, and Eldert J. van Henten., 2016. "Design of an eye-in-hand sensing and servo control framework for harvesting robotics in dense vegetation." Biosystems Engineering, 146, pp.71-84.

Bulanon, D.M., Kataoka, T., Ota, Y. and Hiroma, T., 2002. AE—automation and emerging technologies: a segmentation algorithm for the automatic recognition of Fuji apples at harvest. Biosystems Engineering, 83(4), pp.405-412.

Gongal, A., Amatya, S., Karkee, M., Zhang, Q. and Lewis, K., 2015. Sensors and systems for fruit detection and localization: A review. Computers and Electronics in Agriculture, 116, pp.8-19.

Gunatilaka, A.H. and Baertlein, B.A., 2001. Feature-level and decision-level fusion of noncoincidently sampled sensors for land mine detection. IEEE transactions on pattern analysis and machine intelligence, 23(6), pp.577-589.

Guyon, I., 1997. A scaling law for the validation-set training-set size ratio. AT&T Bell Laboratories, pp.1-11.

Hall, D.L. and McMullen, S.A., 2004. Mathematical techniques in multisensor data fusion. Artech House.

Hannan, M.W., Burks, T.F. and Bulanon, D.M., 2007. A real-time machine vision algorithm for robotic citrus harvesting. In 2007 ASAE Annual Meeting (p. 1). American Society of Agricultural and Biological Engineers.

Jiang, J.A., Chang, H.Y., Wu, K.H., Ouyang, C.S., Yang, M.M., Yang, E.C., Chen, T.W. and Lin, T.T., 2008. An adaptive image segmentation algorithm for X-ray quarantine inspection of selected fruits. Computers and electronics in agriculture, 60(2), pp.190-200.

Kanungo, P., Nanda, P.K. and Ghosh, A., 2010, October. Parallel genetic algorithm based adaptive thresholding for image segmentation under uneven lighting conditions. In Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on (pp. 1904-1911). IEEE.

Kapach, K., Barnea, E., Mairon, R., Edan, Y. and Ben-Shahar, O., 2012. Computer vision for fruit harvesting robots–state of the art and challenges ahead. International Journal of Computational Vision and Robotics, 3(1-2), pp.4-34.

Luo, L., Tang, Y., Lu, Q., Chen, X., Zhang, P. and Zou, X., 2018. A vision methodology for harvesting robot to detect cutting points on peduncles of double overlapping grape clusters in a vineyard. Computers in Industry, 99, pp.130-139.

Ostovar, A., Ringdahl, O. and Hellström, T., 2018. Adaptive Image Thresholding of Yellow Peppers for a Harvesting Robot. Robotics, 7(1), p.11.

Park, J.H., Lee, G.S., Cho, W.H., Toan, N., Kim, S.H. and Park, S.Y., 2010, June. Moving object detection based on clausius entropy. In Computer and Information Technology (CIT), 2010 IEEE 10th International Conference on (pp. 517-521). IEEE.

Rong, D., Ying, Y. and Rao, X., 2017. Embedded vision detection of defective orange by fast adaptive lightness correction algorithm. Computers and Electronics in Agriculture, 138, pp.48-59.

Sa, I., Ge, Z., Dayoub, F., Upcroft, B., Perez, T. and McCool, C., 2016. Deepfruits: A fruit detection system using deep neural networks. Sensors, 16(8), p.1222.

Shmmala, F.A. and Ashour, W., 2013. Color based image segmentation using different versions of K-Means in two Spaces. Global Advanced Research Journal of Engineering, Technology and Innovation (GARJETI, 1(1), pp.030-041.

Shrestha, D.S., Steward, B. and Bartlett, E.R.I.C., 2001, November. Segmentation of plant from background using neural network approach. In Intelligent Engineering Systems through Artificial Neural Networks: Proc. Artificial Neural Networks in Engineering (ANNIE) International Conference (Vol. 11, pp. 903-908). sn.

Sakthivel, K., Nallusamy, R. and Kavitha, C., 2015. Color Image Segmentation Using SVM Pixel Classification Image. World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering, 8(10), pp.1919-1925.

Siegel, M. and Wu, H., 2003, May. Objective evaluation of subjective decisions. In Soft Computing Techniques in Instrumentation, Measurement and Related Applications, 2003. SCIMA 2003. IEEE International Workshop on (pp. 14-18). IEEE.

Vincent, L., 1993. Morphological grayscale reconstruction in image analysis: applications and efficient algorithms. IEEE transactions on image processing, 2(2), pp.176-201.

Vitzrabin, E. and Edan, Y., 2016. Adaptive thresholding with fusion using a RGBD sensor for red sweet-pepper detection. Biosystems Engineering, 146, pp.45-56.

Wang, J., He, J., Han, Y., Ouyang, C. and Li, D., 2013. An adaptive thresholding algorithm of field leaf image. Computers and electronics in agriculture, 96, pp.23-39.

Woebbecke, D.M., Meyer, G.E., Von Bargen, K. and Mortensen, D.A., 1993, May. Plant species identification, size, and enumeration using machine vision techniques on near-binary images. In Optics in Agriculture and Forestry (Vol. 1836, pp. 208-220). International Society for Optics and Photonics.

Zemmour, E., Kurtser, P. and Edan, Y., 2017, April. Dynamic thresholding algorithm for robotic apple detection. In Autonomous Robot Systems and Competitions (ICARSC), 2017 IEEE International Conference on (pp. 240-246). IEEE.

Zhang, Y.J., 1996. A survey on evaluation methods for image segmentation. Pattern recognition, 29(8), pp.1335-1346.

Zheng, L., Zhang, J. and Wang, Q., 2009. Mean-shift-based color segmentation of images containing green vegetation. Computers and Electronics in Agriculture, 65(1), pp.93-98.