



HAL
open science

Sustainable Yield Prediction in Agricultural Areas Based on Fruit Counting Approach

Amine Saddik, Rachid Latif, Abedallah Zaid Abualkishik, Abdelhafid El
Ouardi, Mohamed Elhoseny

► **To cite this version:**

Amine Saddik, Rachid Latif, Abedallah Zaid Abualkishik, Abdelhafid El Ouardi, Mohamed Elhoseny. Sustainable Yield Prediction in Agricultural Areas Based on Fruit Counting Approach. Sustainability, 2023, 15 (3), pp.1-14. 10.3390/su15032707. hal-04162988

HAL Id: hal-04162988

<https://cnam.hal.science/hal-04162988v1>

Submitted on 17 Jul 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Brief Report

Sustainable Yield Prediction in Agricultural Areas Based on Fruit Counting Approach

Amine Saddik ^{1,2,*} , Rachid Latif ¹, Abedallah Zaid Abualkishik ³ , Abdelhafid El Ouardi ⁴ 
and Mohamed Elhoseny ^{5,6} 

¹ Laboratory of Systems Engineering and Information Technology LISTI, National School of Applied Sciences, Ibn Zohr University Agadir, Agadir 80000, Morocco

² Laboratoire d'Innovation Durable et de Recherche Appliquée, International University of Agadir, Bab Al madina, Quartier Tillila, B.P. 8143, Agadir 80000, Morocco

³ College of Computer and Information Technology, American University in the Emirates, Dubai P.O. Box 503000, United Arab Emirates

⁴ SATIE, CNRS, ENS Paris-Saclay, Université Paris-Saclay, 91190 Gif-sur-Yvette, France

⁵ College of Computing and Informatics, University of Sharjah, Sharjah P.O. Box 27272, United Arab Emirates

⁶ Faculty of Computers and Information, Mansoura University, Mansoura 35516, Egypt

* Correspondence: amine.saddik@edu.uiz.ac.ma

Abstract: A sustainable yield prediction in agricultural fields is a very critical task that aims to help farmers have an idea about agricultural operations. Generally, we can find a variety of applications proposed for this purpose that include fruit counting. These applications are based on Artificial Intelligence, especially Deep Learning (DL) and Machine Learning (ML) approaches. These approaches give reliable counting accuracy, but the problem is the use of a large database to achieve the desired accuracy. That makes these approaches limited. For this reason, in this work, we propose a low-complexity algorithm that aims to count green and red apples based on our real dataset collected in the Moroccan region, Fes-Meknes. This algorithm allowed us to further increase sustainability in agricultural fields based on yield prediction. The proposed approach was based on HSV conversion and the Hough transform for fruit counting. The algorithm was divided into three blocks based on image acquisition and filtering for the first block. The second block is the conversion to HSV and the detection of fruits. Finally, the counting operation for the third block. Subsequently, we proposed an implementation based on the low-cost Raspberry system and a desktop. The results show that we can reach 15 fps in the case of the Raspberry architecture and 40 fps based on the desktop. Our proposed system can inform agricultural policy by providing accurate and timely information on crop production, which can be used to guide decisions on food supply and distribution.

Keywords: sustainable yield prediction; agricultural operations; artificial intelligence; Fes-Meknes region; sustainability



Citation: Saddik, A.; Latif, R.; Abualkishik, A.Z.; El Ouardi, A.; Elhoseny, M. Sustainable Yield Prediction in Agricultural Areas Based on Fruit Counting Approach. *Sustainability* **2023**, *15*, 2707. <https://doi.org/10.3390/su15032707>

Academic Editor: Piotr Prus

Received: 3 January 2023

Revised: 22 January 2023

Accepted: 30 January 2023

Published: 2 February 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Sustainable yield in agriculture is imperative, as crop production has been damaged due to the imbalanced use of farm inputs and extreme weather events [1–5]. Likely, due to crops, fruit counting is a very important task for yield prediction in agricultural fields. This application has seen a huge development in several agricultural products, as well as counting as an application based on several techniques [6,7]. These techniques are improving with the development of algorithms in precision agriculture. However, the most used are the applications based on Deep Learning and machine learning [8]. These learning methods show great progress in the case where the precision of results is required. The difference between these algorithms is based on the inputs of the algorithm and the number of delivered data. These data are then exposed to a training process in order to achieve the desired performance. Additionally, these techniques require a huge amount of data in order

to have a high accuracy in the desired application. Among these applications, we can find the detection of weeds, the counting of agricultural products, the monitoring of vital signs, and disease detection [9–12]. Artificial intelligence (AI) can be used to predict crop yields by analyzing various factors such as weather data, soil conditions, and historical yield data. Machine learning algorithms can be trained in this data to make accurate predictions about future crop yields. However, it is important to note that the accuracy of these predictions will depend on the quality and quantity of data available, as well as the specific algorithm used. Additionally, AI systems need to be continuously updated with new data to improve their accuracy [13,14].

Several applications have been proposed to improve the accuracy in the algorithmic aspect. For example, U-O. Dorj et al., 2017 proposed a dedicated algorithm for calculating and predicting agricultural yields [15]. Similarly, P.Y. Ramos et al., 2017 proposed a system for counting coffee branches based on computer vision [16]. J.G-Mola et al., 2020 were based on 3D localization for fruit detection. The proposed algorithm was developed using R-CNN and a structure-from-motion (SFM) mask [17]. In the same context, A. Aquiro et al., 2020 proposed an algorithm based on CNN to identify different olive fruit. This application is based on Deep Learning and machine learning approaches to count, identify and detect different agricultural products [18–25]. Fruit counting in precision agriculture is the process of using technology to count the number of fruits on a tree or plant in order to optimize crop yields. This can be conducted using various techniques, including. Image processing Using cameras or other imaging devices, images of the fruits can be captured and analyzed using computer algorithms. This can be conducted either by taking images of the entire tree or plant and counting the fruits manually or by using image processing techniques such as object detection to identify and count the fruits automatically. Machine learning algorithms can be trained to recognize and count fruits in images. This can be carried out by providing the algorithm with a large dataset of images that have been labeled with the number of fruits in each image. Computer vision techniques can be used to detect and count fruits in images automatically. This can be conducted by using techniques such as edge detection and blob analysis to identify the fruits in the image. Robotics technology can be used to count fruits by using cameras or other sensors to detect and count fruits on a tree or plant. These techniques can be used to count the number of fruits, which can then be used to optimize crop yields by making decisions such as when to harvest or how to best care for the crops. However, the problem with these proposed techniques is the high algorithmic complexity as well as the use of embedded architecture that can increase the reliability of the developed system. For this reason, we propose in this work a low-complexity and real-time counting algorithm that does not require any pre-collected data or training of DL or ML models.

In this work, we propose an algorithm based on image processing to count green and red apples. This algorithm is based on the color detection method and the geometrical shapes of the apples. Thus, we propose an implementation of the proposed algorithm in a low-cost system. Then a comparison was presented between the raspberry Pi B+ embedded architecture and a desktop. The proposed algorithm is based on RGB to HSV image conversion and Hough transform. The reason for using these algorithms is to avoid the DL and ML techniques that require a large database. As well as GPU boards that require large energy consumption. In addition, another constraint is based on the long time required for the data training. The proposed algorithm is based on C/C++ language and a low-cost embedded architecture. Our contribution is as follows:

- (1) Proposition of a fruit detection and counting algorithm based on intelligent vision approaches.
- (2) The implementation of the proposed algorithm in a low-cost embedded architecture of the Raspberry model and a desktop.

The proposed work is divided into four parts, the first part for the general introduction to the topic to be developed. The second part focuses on the material and algorithmic study proposed in this work. Then we have the results obtained based on the evaluation

of processing time, accuracy and results in the real agricultural field. Finally, we have the conclusion and future work.

2. Material and Methods

2.1. Material

The agricultural product studied in this paper is based on red and green apples, and this agricultural field is located in the Fes-Meknes region near Lake Daiit Aoua. This region is characterized by the high production of apples in Morocco. This field is located at $33^{\circ}38'41''$ N and $5^{\circ}02'04''$ W with a surface of 44.12 ha and a perimeter of 3.13 KM. The data collection was based on an RGB camera with a resolution of 4608×3456 and 16 Mpx. The camera has a high resolution in order to obtain accurate results. The images used in this work were collected on Thursday, 3 September 2020, around 13.30. The choice of this region is due to the high production of green and red apples. Figure 1 shows the location of the studied field.

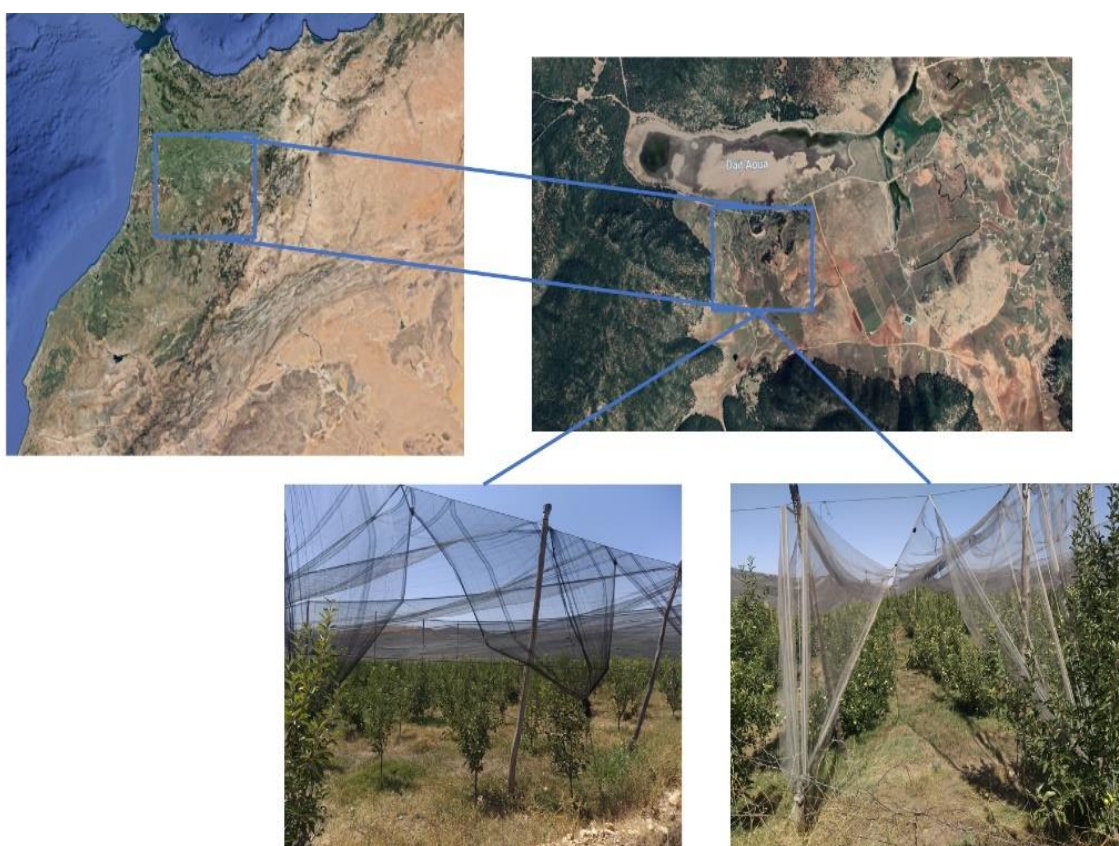


Figure 1. A real agricultural area based on the Fes-Meknes region in Morocco.

Counting fruits such as apples can help farmers have an idea about the yield of agricultural fields. These two agricultural products are characterized by the green and red color of the apples. Generally, we have more than 20 types of apples, namely fuji characterized by a red color. We also have a pink lady produced a little late due to the sugar content and the acid necessary to produce good apples and Honeycrisp, which combine yellow and red colors. In addition, we have the Gala apples, which are bigger and have a clear red color with a sweet taste.

On the other hand, we have the Red Delicious apples, which have a very red type of apple. Additionally, we have Golden Delicious, which has a green color. Our study will focus on Golden Delicious and Red Delicious due to the high demand for these two types of apples. Figure 2 shows images from the studied agricultural field.

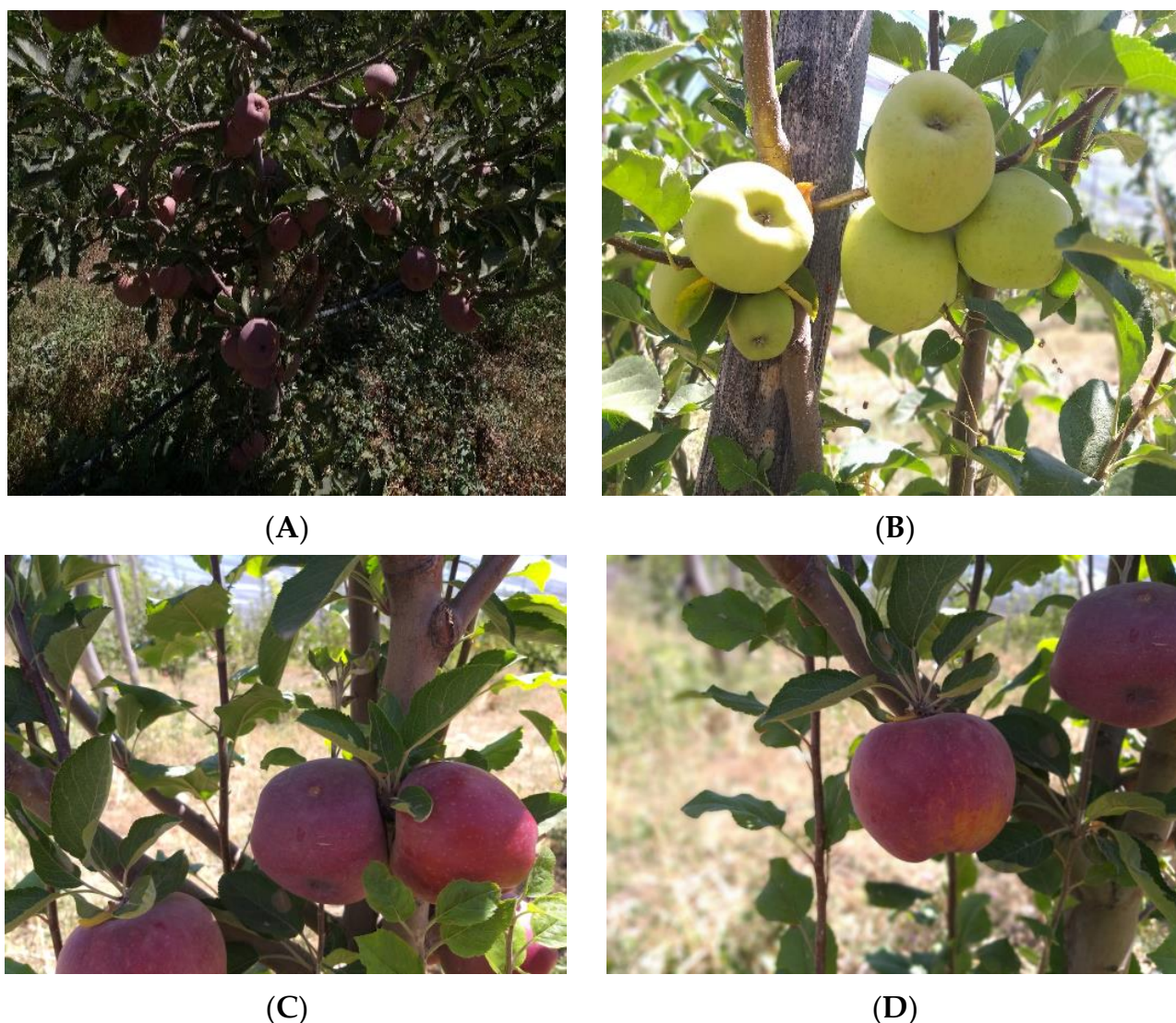


Figure 2. Red and green apples from agricultural area: (A,C,D) present the Red Delicious apples and (B) the Golden Delicious apples that have been used in this study.

2.2. Algorithm Study

The proposed algorithm in this work is based on the acquisition of images and then filtering operation to eliminate noise. We have the conversion of RGB images to HSV to select the desired color. For example, if we have red apples, we must choose the red pixels; in the opposite case where we have green apples, we must select the green color.

The selection of the color is made with such parameters in the HSV conversion domain. Once the color is selected, we move to the operation of detection of apples via the geometric shape that merges with the circle; this detection is based on the transformation of Hough [26]. Then, once the shape is detected, we go to the counting operation to predict the different fruits of the agricultural fields. Figure 3 shows an overview of the proposed algorithm.

The proposed algorithm is divided into three blocks, the first block for the acquisition of images and then applying the median filter to filter the images. The second block is based on the conversion to HSV, as shown in Equations (1)–(5). Then, in the same block, we have fruit detection based on the Hough transform [26]. The last block is dedicated to the counting of fruits. Algorithm 1 shows the different steps to follow for the counting of fruits.

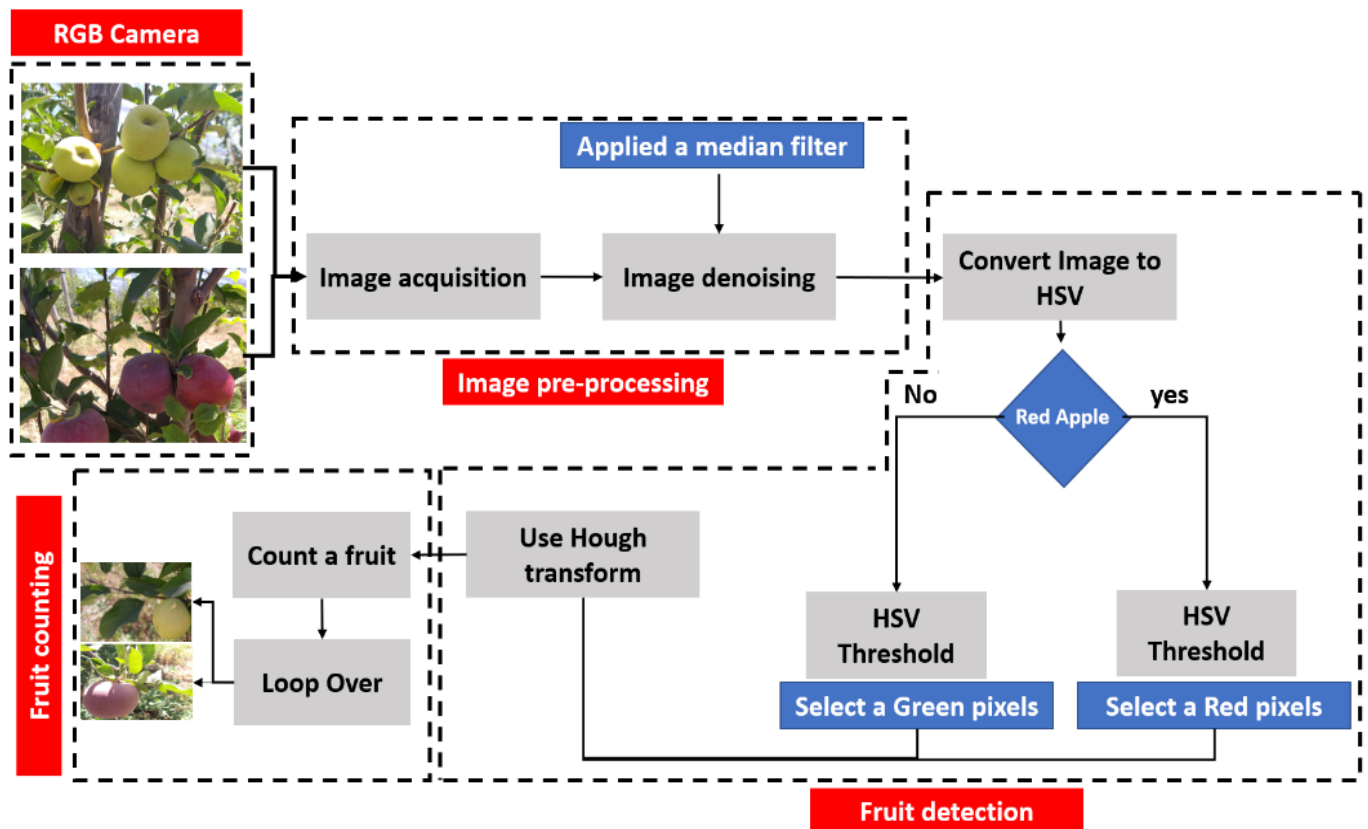


Figure 3. Algorithm overview.

The algorithm proposed in our work is based on low algorithmic complexity. On the other hand, the algorithms based on ML or DL require a huge database to handle all the cases. As well as the training of the weights generated in the ML or DL algorithms also requires a permanent update which influences the database collection, especially when dealing with seasonal fruits. In this case, we must wait one year or more to collect the new database. The algorithm in our case does not require any database. It is based first on acquiring images emitted by the RGB camera. Then these images will be filtered to have clear images without noise that will facilitate the recognition later. The filter used in our case is based on the median filter. This filter is a non-linear image processing technique used to remove noise from an image. It works by replacing the value of each pixel with the median value of the pixels in a defined neighborhood around the pixel. This neighborhood is typically a square or circular region surrounding the pixel. The median filter is particularly effective at removing “salt and pepper” noise, which appears as randomly dispersed white and black pixels. The median filter is a robust method, as it is not affected by outliers. It is often used in combination with other techniques such as the mean filter. After filtering the images we move to the HSV conversion. HSV (Hue, Saturation, Value) is a color model used in image processing and computer vision to represent an image in terms of its color characteristics. It is an alternative representation of the RGB (Red, Green, Blue) color model, which is widely used in electronic displays and digital imaging. In the HSV model, a color is represented by three components:

Hue: represents the dominant wavelength of light and corresponds to the traditional concept of color (e.g., red, yellow, green, blue, etc.). It is typically represented as an angle on a color wheel, with red at 0 degrees and the other colors increasing in value.

Saturation: represents the purity of color and ranges from 0 (gray) to 1 (fully saturated). It is a measure of how much white is mixed with the color.

Value: represents the brightness of a color and ranges from 0 (black) to 1 (white).

The process of converting an image from the RGB color model to the HSV color model is called color space conversion. The conversion is performed using mathematical formulas that map the RGB values of each pixel in the image to the corresponding HSV values. The resulting HSV image can then be used for various image processing tasks such as color-based object detection, image segmentation, and color correction. Because the HSV model separates color information from brightness information, it can be more useful for certain tasks than the RGB model.

After the HSV conversion, we move on to define the threshold to select the red and green colors. The exact range of hue values for red can vary depending on the specific HSV color model being used, but in general, a hue value of around 0–20 or 340–360 degrees is used to represent red. To select a specific red color, you would need to define a range of hue values that correspond to red and then filter out pixels that fall within that range. Firstly, we start by converting the image from the RGB color space to the HSV color space, and after that, we define the range of hue values for red and create a binary image mask to filter out pixels that fall outside the defined range of hue values. Then we used the Hough transform to detect the geometrical shape of apple fruits. A Hough transform is a technique used in image processing for detecting shapes or patterns within an image. It is commonly used for detecting lines, circles, and other geometric shapes in images. The technique is based on the idea of representing a shape in an image as a set of points in a Hough space, where each point represents a possible location of the shape. The Hough transform then identifies clusters of points in the Hough space, corresponding to the shape(s) present in the image. The Hough transform is a powerful technique for detecting shapes in images, particularly when the shapes are partially obscured or have low contrast. Finally, we have fruit counting based on the used Hough and HSV techniques.

For the RGB to HSV, we have Hue calculation [27]:

$$H = \begin{cases} 60^\circ \times \left(\frac{g-b}{\Delta} \bmod 6 \right); D(\max) = r \\ 60^\circ \times \left(\frac{b-r}{\Delta} + 2 \right); D(\max) = g \\ 60^\circ \times \left(\frac{r-g}{\Delta} + 4 \right); D(\max) = b \end{cases} \quad (1)$$

Equation (1) is related to color image processing and is specifically related to hue, saturation and value/brightness. H represents hue, and it is calculated by taking the difference between the green and blue color channels, dividing it by some delta values and then using modulus 6. $D(\max)$ is the max value among r, g and b, and it is calculated in terms of the difference between the other two-color channels and some delta values. For the Saturation, we have the following:

$$S = \begin{cases} 0; D(\max) = 0 \\ \frac{\Delta}{D(\max)}; D(\max) \neq 0 \end{cases} \quad (2)$$

Finally for the Value:

$$V = D(\max) \quad (3)$$

With

$$\begin{cases} r = \frac{R}{255} \\ g = \frac{G}{255} \\ b = \frac{B}{255} \end{cases}$$

and

$$\begin{cases} D(\max) = \max(R, G, B) \\ D(\min) = \min(R, G, B) \\ \Delta = D(\max) - D(\min) \end{cases}$$

For the inverse operation to convert HSV images to RGB, we have the following:

$$(r, g, b) = \begin{cases} (Y, Z, 0), 0^\circ \leq H < 60^\circ \\ (Z, Y, 0), 60^\circ \leq H < 120^\circ \\ (0, Y, Z), 120^\circ \leq H < 180^\circ \\ (0, Z, Y), 180^\circ \leq H < 240^\circ \\ (Z, 0, Y), 240^\circ \leq H < 360^\circ \end{cases} \quad (4)$$

$$(R, G, B) = ((r + N) \times 255, (g + N) \times 255, (b + N) \times 255) \quad (5)$$

Algorithm 1. Outlines the step-by-step methodology utilized in our proposed approach, which details the logical sequence of operations to be performed in order to achieve the desired outcome.

1. Read the RGB image in three bands: red, green and blue.
 2. Test if the image has been loaded as well as the existence of data in the image.
 3. Apply the filtering operation based on the median filter.
 4. Convert the RGB images to HSV to select the pixels.
 5. Apply the Hough transform for fruit detection.
 6. Apply a looping operation to detect all the fruits that exist in the image.
 7. Count the detected fruits.
-

3. Results and Discussion

3.1. Experimental Results

The evaluation of the results was based on the algorithm mentioned in Figure 3 to count the different fruits of the apples. We have added a temporal evaluation to the proposed algorithm to study the specific architecture. The embedded architecture used in this evaluation is the Raspberry Pi 3B+ embedded system with a Broadcom Cortex-A53 @ 1.4 Ghz processor and 1 GB LPDDR2 RAM. Additionally, we added a comparison with a desktop in order to extract the strong points of using low-cost embedded architecture.

The approach followed in the evaluation of this work is divided into two parts; the first part is dedicated to evaluating the real database collected in the agricultural field. The second part focuses on temporal and architectural studies. The language used in this study is C/C++ based on the OpenCV library. The temporal evaluation was based on a sequence of 50 images to extract the maximum, minimum, and average time. The apples selected in this study are green and red.

Figure 4 shows the agricultural products counted in the field. As well as we have shown the conversion to HSV. The top right figure shows the Red Delicious apples and the bottom Golden Delicious. Additionally, on the left, we have the conversion of these images to HSV.

After the HSV conversion, we have to select the pixels according to the type of apples, and then we have to detect the fruits and the count. Figure 5 shows the results obtained after the evaluation of the proposed algorithm. Generally, the algorithm has a strong point in detecting the fruits with error due to the use of an RGB camera that cannot calculate the depth; therefore, if we have apples in the back, we cannot estimate them due to this type of camera. The solution here is to try to make a 360° turn on all the plants in order to count all the products with the minimum error.



Figure 4. RGB and HSV conversion of the image.



Figure 5. Cont.



Figure 5. Counting result.

3.2. Processing Time Result

The temporal evaluation was based on two architectures, a desktop, and an embedded Raspberry architecture. The sequential implementation of the desktop showed a benefit compared to the embedded architecture due to the high frequency of these machines. Yet the major problem of this type of architecture is the high-power consumption and the high weight for applications based on unmanned aerial vehicles and ground robots [28,29]. Figures 6 and 7 show the results of the global processing time from the acquisition to the final count.

Figure 6 shows the results obtained from the evaluation of our algorithm based on the desktop. The evaluation showed that we obtain a processing time that varies between 14 ms and 38 ms for the minimum and maximum values, respectively, as well as an average of 25.5 ms for the processing of each image which implies the processing of 40 images/s. Similarly, Figures 7 and 8 represent the algorithm's evaluation of the Raspberry embedded architecture with a maximum time of 88 ms and a minimum of 39 ms. The obtained average of 50 images gives 63.08 ms, which implies the processing of 15 frames/s. We have evaluated each block of the algorithm separately, and we have found that block 1 consumes 19.02 ms in the Raspberry architecture as well as 6.6 ms on the desktop. For the second block, we have 16.4 ms on the desktop and 34.05 on the Raspberry board. Similarly, we

have 2.5 ms consumed on the desktop and 10.01 on the Raspberry board for the third block. Figure 8 shows the comparison between the different blocks.



Figure 6. Processing time based on Desktop.

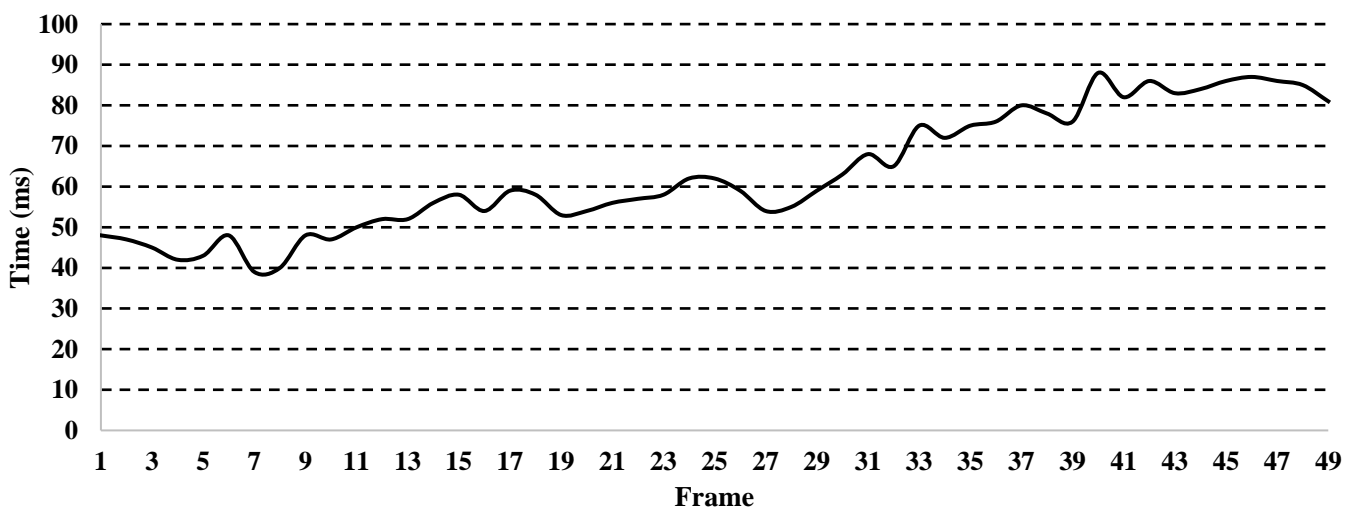


Figure 7. Processing time based on Raspberry.

In Figure 7, we tried to evaluate the processing time on the Raspberry embedded architecture. The results on a sequence of images allow us to extract the max, min, and average time to increase processing time accuracy. After calculating the processing time, we evaluate each block proposed in the algorithm. This technique allows us to locate the part that consumes the majority of processing time in order to accelerate it later.

Figure 8 shows the detection accuracy and processing time on the desktop. The system demonstrated exceptional accuracy, with a final score of 98.45%. This falls within the desired range of 94.98% to 98.91% and is among the highest levels of accuracy ever recorded for this type of model. The results confirm that the model is able to make highly accurate predictions and can be trusted to perform well in real-world scenarios.

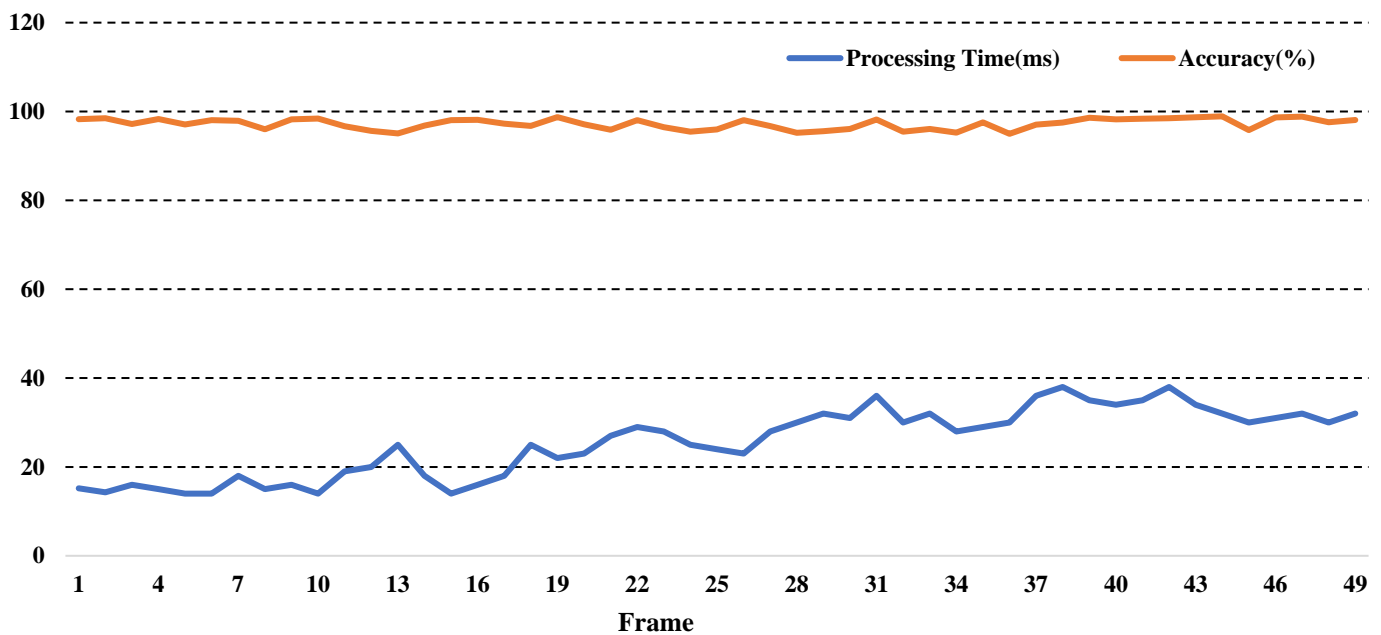


Figure 8. Detection accuracy and processing time (Desktop).

Figure 9 shows the detection accuracy and processing time in Raspberry. The system's performance was evaluated using a dataset, and it was found to have an accuracy of 95.04%. This falls within the expected range of 91.05% to 98.77%. The high accuracy of the system indicates that it is able to classify and predict outcomes with a high degree of reliability. The results demonstrate that the model is well-suited for practical applications and can be trusted to produce accurate results.

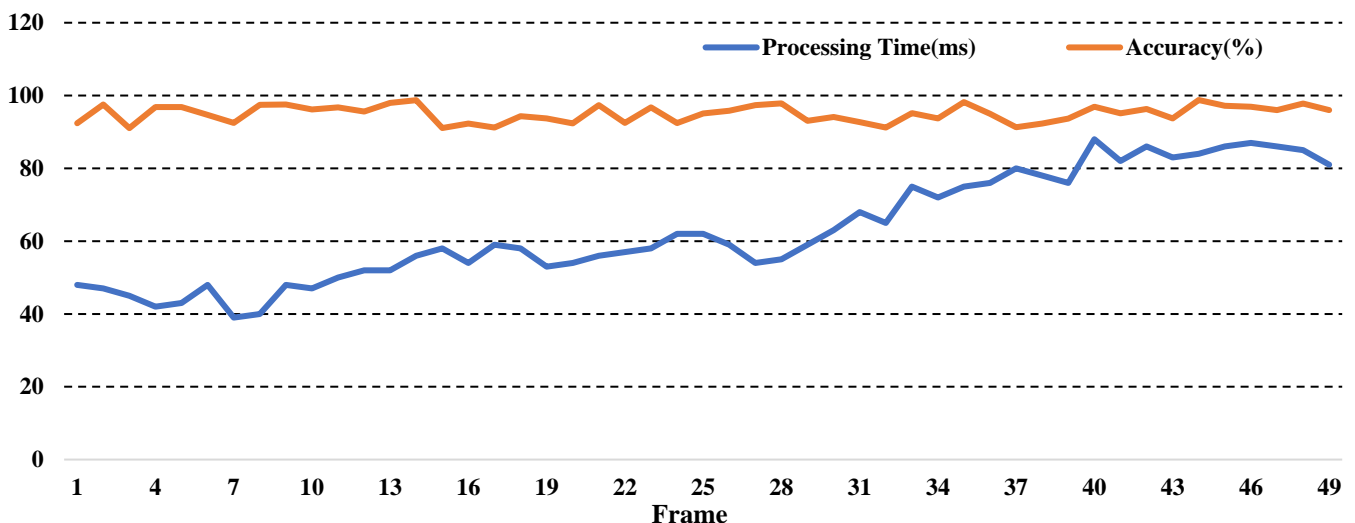


Figure 9. Detection accuracy and processing time (Raspberry).

In Figure 10, a comparison of processing time has been proposed between the Raspberry embedded architecture and the desktop. This comparison showed that the desktop consumes less processing time. However, the major problem is the high-power consumption compared to the low-cost and low-power consumption Raspberry embedded architecture. For this reason, the choice of Raspberry hardware is justified by the different characteristics of this system. Figure 11 present a comparative study between Desktop and Raspberry embedded architecture for each block in our proposed algorithm

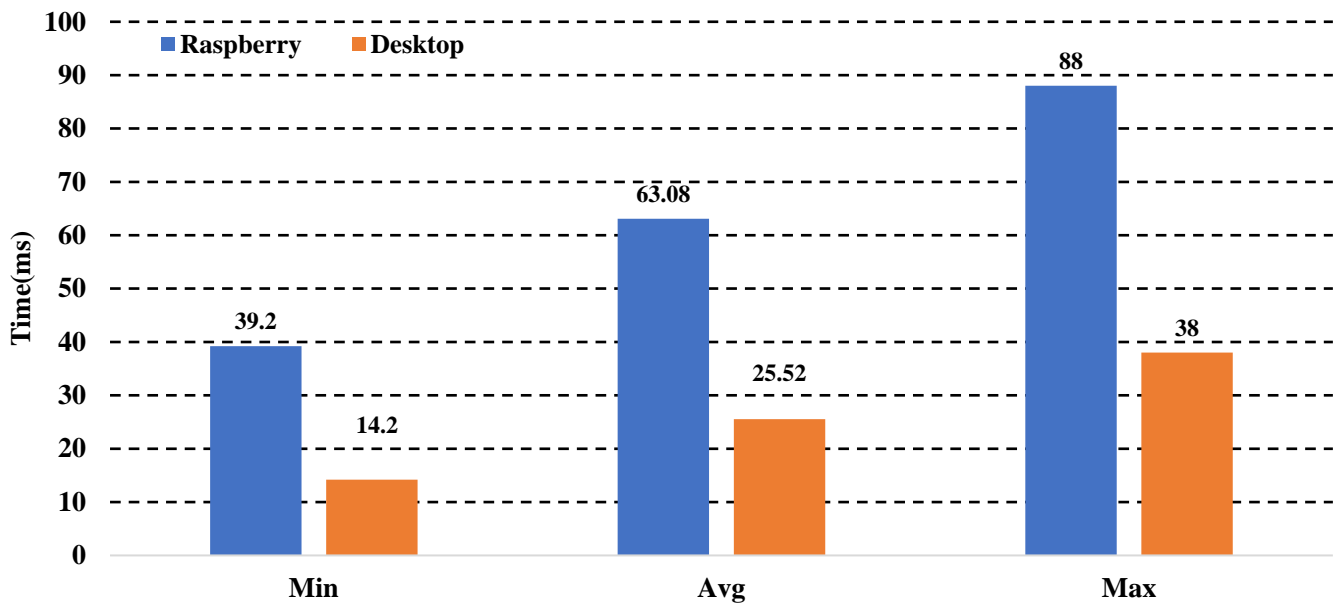


Figure 10. Processing time based on Raspberry.

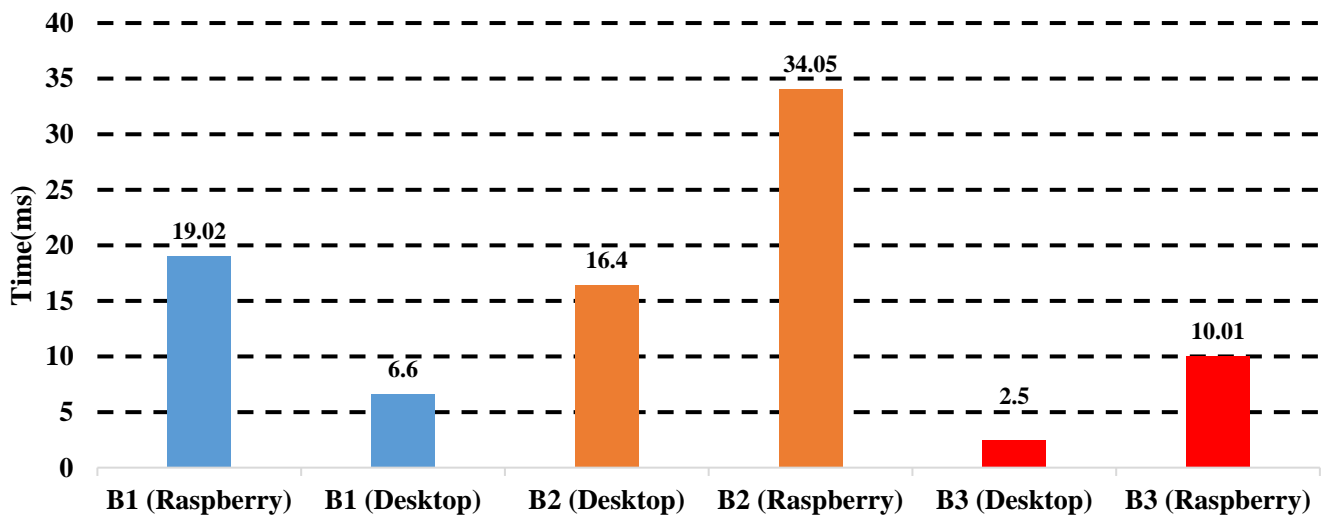


Figure 11. Processing time based on Raspberry and Desktop for each block.

Table 1 shows a comparative study between the results obtained in our algorithm and the published works in this field. The comparison shows that our algorithm has good accuracy and can process up to 15 frames/s in the Raspberry low-power embedded architecture. In the case of the desktop, the algorithm processes 40 frames/s. However, the works presented in Table 1 are based on conventional architectures that prevent real-time processing.

Table 1. A comparative study.

Reference	Accuracy	System Specification	Algorithm	FPS
N. Häni et al., 2020 [30]	95.56–97.83%	NVIDIA Tesla K20X GPU	Gaussian Mixture Model	5
F.Gao et al., 2022 [31]	91.49%	NVIDIA GTX 1080 GPU	YOLOv4-tiny	2–5
J. Villacrés et al., 2022 [32]	93–97%	NVIDIA Tesla K40 GPU	Faster R-CNN	5–17
N. Häni et al., 2018 [33]	96–97%	NVIDIA GTX 1080 GPU	Gaussian Mixture Model	-
Proposed method	95.04–97.22%	Raspberry Pi	HSV-Hough	15

4. Conclusions

In this work, we proposed a green and red apple-counting algorithm based on selecting pixel colors and the Hough transform for fruit detection. Then we apply the processing of counting fruit to predict the yield in agricultural fields. The proposed algorithm is based on three blocks that aim to process the desired application. Thus, we have evaluated our algorithm in an embedded architecture type Raspberry Pi B3+ and a desktop. The results show that we can achieve 40 frames/s on the desktop and 15 frames/s on the Raspberry board. The datasets used in this work are based on an RGB camera and an agricultural field located in the Moroccan region, Fes-Meknes. In future work, we aim to accelerate this algorithm in a low-cost system type CPU-GPU or CPU-FPGA. This acceleration will help us ensure the real-time constraint in our processing. The system achieved accuracy on the test dataset, falling within the desired range of 95.04% to 97.22%. This level of accuracy indicates that the model is able to identify and classify the majority of the data points within the dataset. The results demonstrate that the model can be used for reliable and accurate prediction in practical applications. As a future suggestion, we aim to use a combination of weather data and historical yield information to make accurate predictions using our developed approach. This includes data on temperature, precipitation, soil moisture and other environmental factors that can affect crop growth.

Author Contributions: Conceptualization, A.S. writing—original draft preparation, A.S. and A.S.; methodology, A.S. and A.E.O.; software, A.S. and A.E.O.; validation, M.E. and A.Z.A.; formal analysis, R.L.; data curation, A.S.; writing—review and editing, M.E. and R.L.; visualization, A.Z.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by [National Center for Scientific and Technical Research of Morocco (CNRST)] grant number [19 UIZ2020].

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data used in this paper are available under requests.

Acknowledgments: We owe a debt of gratitude to the National Center for Scientific and Technical Research of Morocco (CNRST) for their financial support and for their supervision (grant number: 19 UIZ2020).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Abbas, A.; Waseem, M.; Ahmad, R.; Khan, K.A.; Zhao, C.; Zhu, J. Sensitivity analysis of greenhouse gas emissions at farm level: Case study of grain and cash crops. *Environ. Sci. Pollut. Res.* **2022**, *29*, 82559–82573. [\[CrossRef\]](#)
2. Mohammad, A.T.; Zhendong, M. Intelligent Web Information Extraction Model for Agricultural Product Quality and Safety System. *J. Intel. Syst. Internet Things* **2021**, *4*, 99–110. [\[CrossRef\]](#)
3. Elahi, E.; Khalid, Z.; Tauni, M.Z.; Zhang, H.; Lirong, X. Extreme weather events risk to crop-production and the adaptation of innovative management strategies to mitigate the risk: A retrospective survey of rural Punjab, Pakistan. *Technovation* **2022**, *117*, 102255. [\[CrossRef\]](#)
4. Abedallah, Z.A.; Rasha, A.; William, T. Evaluating Smart Agricultural Production Efficiency using Fuzzy MARCOS method. *J. Neutrosophic Fuzzy Syst.* **2022**, *3*, 8–18. [\[CrossRef\]](#)
5. Elahi, E.; Zhang, H.; Lirong, X.; Khalid, Z.; Xu, H. Understanding cognitive and socio-psychological factors determining farmers' intentions to use improved grassland: Implications of land use policy for sustainable pasture production. *Land Use Policy* **2021**, *102*, 105250. [\[CrossRef\]](#)
6. Vong, C.N.; Conway, L.S.; Zhou, J.; Kitchen, N.R.; Sudduth, K.A. Early corn stand count of different cropping systems using UAV-imagery and deep learning. *Comput. Electron. Agric.* **2021**, *186*, 106214. [\[CrossRef\]](#)
7. Saddik, A.; Latif, R.; El Ouardi, A.; Elhoseny, M.; Khelifi, A. Computer development based embedded systems in precision agriculture: Tools and application. *Acta Agric. Scand. Sect. B Soil Plant Sci.* **2022**, *72*, 589–611. [\[CrossRef\]](#)
8. Vasconez, J.; Delpiano, J.; Vougioukas, S.; Cheein, F.A. Comparison of convolutional neural networks in fruit detection and counting: A comprehensive evaluation. *Comput. Electron. Agric.* **2020**, *173*, 105348. [\[CrossRef\]](#)
9. Saddik, A.; Latif, R.; Elhoseny, M.; El Ouardi, A. Real-time evaluation of different indexes in precision agriculture using a heterogeneous embedded system. *Sustain. Comput. Inform. Syst.* **2021**, *30*, 100506. [\[CrossRef\]](#)

10. Latif, R.; Saddik, A.; Elouardi, A. Evaluation of agricultural precision algorithms on UAV images. In Proceedings of the 2019 International Conference of Computer Science and Renewable Energies (ICCSRE), Agadir, Morocco, 22–24 July 2019; pp. 1–4.
11. Latif, R.; Jamad, L.; Saddik, A. Implementation of Hybrid Algorithm for the UAV Images Preprocessing Based on Embedded Heterogeneous System: The Case of Precision Agriculture. In *Enabling Machine Learning Applications in Data Science: Proceedings of Arab Conference for Emerging Technologies*; Hassanien, A.E., Darwish, A., Abd El-Kader, S.M., Alboaneen, D.A., Eds.; Springer: Singapore, 2021. [\[CrossRef\]](#)
12. Guerrouj, F.Z.; Latif, R.; Saddik, A. Evaluation of NDVI and NDWI parameters in CPU-GPU Heterogeneous Platforms based CUDA. In Proceedings of the 2020 5th International Conference on Cloud Computing and Artificial Intelligence: Technologies and Applications (CloudTech), Marrakesh, Morocco, 28–30 May 2020; IEEE: Piscataway Township, NJ, USA, 2020; pp. 1–6. [\[CrossRef\]](#)
13. Rania, A.G.; Saad, M. A Proposed Framework for Effective Risk Management in Egyptian Sustainable Development Projects. *Am. J. Bus. Oper. Res.* **2019**, *1*, 26–42. [\[CrossRef\]](#)
14. Tang, Y.; Zhou, H.; Wang, H.; Zhang, Y. Fruit detection and positioning technology for a Camellia oleifera C. Abel orchard based on improved YOLOv4-tiny model and binocular stereo vision. *Expert Syst. Appl.* **2023**, *211*, 118573. [\[CrossRef\]](#)
15. Dorj, U.-O.; Lee, M.; Yun, S.-S. An yield estimation in citrus orchards via fruit detection and counting using image processing. *Comput. Electron. Agric.* **2017**, *140*, 103–112. [\[CrossRef\]](#)
16. Ramos, P.; Prieto, F.; Montoya, E.; Oliveros, C. Automatic fruit count on coffee branches using computer vision. *Comput. Electron. Agric.* **2017**, *137*, 9–22. [\[CrossRef\]](#)
17. Gené-Mola, J.; Sanz-Cortiella, R.; Rosell-Polo, J.R.; Morros, J.-R.; Ruiz-Hidalgo, J.; Vilaplana, V.; Gregorio, E. Fruit detection and 3D location using instance segmentation neural networks and structure-from-motion photogrammetry. *Comput. Electron. Agric.* **2020**, *169*, 105165. [\[CrossRef\]](#)
18. Saddik, A.; Latif, R.; El Ouardi, A. Low-Power FPGA Architecture Based Monitoring Applications in Precision Agriculture. *J. Low Power Electron. Appl.* **2021**, *11*, 39. [\[CrossRef\]](#)
19. Shahzaib, A.; Saleem, A. Decision Support Modeling For Agriculture Land Selection Based On Sine Trigonometric Single Valued Neutrosophic Information. *Int. J. Neutrosophic Sci.* **2020**, *9*, 60–73. [\[CrossRef\]](#)
20. Albornoz, V.M.; Véliz, M.I.; Ortega, R.; Ortíz-Araya, V. Integrated versus hierarchical approach for zone delineation and crop planning under uncertainty. *Ann. Oper. Res.* **2019**, *286*, 617–634. [\[CrossRef\]](#)
21. Petridis, N.E.; Digkas, G.; Anastasakis, L. Factors affecting innovation and imitation of ICT in the agrifood sector. *Ann. Oper. Res.* **2018**, *294*, 501–514. [\[CrossRef\]](#)
22. Saddik, A.; Latif, R.; Taher, F.; El Ouardi, A.; Elhoseny, M. Mapping Agricultural Soil in Greenhouse Using an Autonomous Low-Cost Robot and Precise Monitoring. *Sustainability* **2022**, *14*, 15539. [\[CrossRef\]](#)
23. Nagendra, N.P.; Narayanamurthy, G.; Moser, R. Satellite big data analytics for ethical decision making in farmer’s insurance claim settlement: Minimization of type-I and type-II errors. *Ann. Oper. Res.* **2020**, *315*, 1061–1082. [\[CrossRef\]](#)
24. Saddik, A.; Latif, R.; El Ouardi, A.; Alghamdi, M.I.; Elhoseny, M. Improving Sustainable Vegetation Indices Processing on Low-Cost Architectures. *Sustainability* **2022**, *14*, 2521. [\[CrossRef\]](#)
25. Latif, R.; Saddik, A. SLAM algorithms implementation in a UAV, based on a heterogeneous system: A survey. In Proceedings of the 2019 4th World Conference on Complex Systems (WCCS), Piscataway Township, NJ, USA, 22–25 April 2019; pp. 1–6. [\[CrossRef\]](#)
26. Shapiro, S.D. Use of the Hough transform for image data compression. *Pattern Recognit.* **1980**, *12*, 333–337. [\[CrossRef\]](#)
27. Qin, X.; Wang, H.; Du, Y.; Zheng, H.; Liang, Z. Retinex Structured Light Image Enhancement Algorithms in HSV Color Space. *J. Comput. Aided Des. Comput. Graph.* **2013**, *4*, 488–493.
28. Liu, M.; Zhu, D.; Nie, X.; Shi, C.; Tu, H.; Jing, B.; Fan, H.; Zhao, H. Design and Testing of a Remote Control-Based Integrated Flow Regulation and Measurement Device. *Sustainability* **2022**, *14*, 15672. [\[CrossRef\]](#)
29. Rondelli, V.; Capacci, E.; Franceschetti, B. Evaluation of the Stability Behavior of an Agricultural Unmanned Ground Vehicle. *Sustainability* **2022**, *14*, 15561. [\[CrossRef\]](#)
30. Häni, N.; Roy, P.; Isler, V. A comparative study of fruit detection and counting methods for yield mapping in apple orchards. *J. Field Robot.* **2020**, *37*, 263–282. [\[CrossRef\]](#)
31. Gao, F.; Fang, W.; Sun, X.; Wu, Z.; Zhao, G.; Li, G.; Li, R.; Fu, L.; Zhang, Q. A novel apple fruit detection and counting methodology based on deep learning and trunk tracking in modern orchard. *Comput. Electron. Agric.* **2022**, *197*, 107000. [\[CrossRef\]](#)
32. Villacrés, J.; Viscaino, M.; Delpiano, J.; Vougioukas, S.; Cheein, F.A. Apple orchard production estimation using deep learning strategies: A comparison of tracking-by-detection algorithms. *Comput. Electron. Agric.* **2023**, *204*, 107513. [\[CrossRef\]](#)
33. Häni, N.; Roy, P.; Isler, V. Apple counting using convolutional neural networks. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Piscataway Township, NJ, USA, 1–5 October 2018; pp. 2559–2565. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.