

Original Article

# Detection of Peruvian Golden Berry Quality using a Convolutional Neural Network

Jhon Taquila-Velásquez<sup>1</sup>, Alexis Vela-Guerra<sup>2</sup>, Pedro Portillo-Mendoza<sup>3</sup>, Carlos Sotomayor-Beltran<sup>4</sup>

<sup>1,2,3,4</sup>Technological University of Peru, Lima, Peru.

<sup>4</sup>Corresponding Author : c22137@utp.edu.pe

Received: 29 September 2023

Revised: 15 December 2023

Accepted: 16 December 2023

Published: 07 January 2024

**Abstract** - In Peru, the National Strategic Export Plan seeks to promote exports of many fruits, among them the Peruvian golden berries. To increase the production speed of golden berries to match foreign competition, the design of a convolutional neural network (CNN) algorithm applied to the selection of this type of berry is proposed. The classification of the golden berry quality has been defined according to its condition, a good one and in poor condition. The developed classification software, at its core, is based on a pre-trained CNN called GoogLeNet, which was implemented using Matlab development software; additionally, the Matlab App designer extension was used to create a graphical user interface. The architecture of the CNN allowed us to obtain the characteristic parameters of the fruit, where the criterion for using the database considered 80% images for training, 10% for validation, and 10% for post-training. For this work, 241 images were used, divided into two blocks, as NO\_GOOD (in poor condition and immature) and as GOOD (mature and in good condition). The best-trained CNN, with a validation percentage of 100%, was embedded in the user interface. A total of 6000 iterations and 500 epochs were used for the training of the CNN. The development of the user interface system allowed to select golden berries based on their quality at greater speeds than by doing it manually.

**Keywords** - Golden berry quality, Convolutional neural network (CNN), GoogLeNet, Matlab, User interface system.

## 1. Introduction

In some departments of Peru, the improvement of the production of golden berries is considered in the National Export Strategic Plan (PENX 2025), which was prepared by the Ministry of Foreign Trade and Tourism (MINCETUR) and the Permanent Mixed Multisectoral Commission. This plan allows monitoring of the commercial productive context of the regions, giving priority to the dynamics of golden berry production for exporting purposes. A clear example of golden berry production is observed in the Huanuco department when observing the annual growth from 722 to 1573 tons of golden berries between the years 2015-2020 [1]. Faced with this incentive imposed by the state, many Peruvian production plants are at a disadvantage compared to foreign competition, which possess mainly fully automated systems of production. The production process of golden berries in Peru has been carried out manually since its inception, unlike other products such as avocado, lemon or orange that recently have automated plants. This directly affects Peruvian farmers of golden berries, who are faced with great foreign demand and need to increase the speed of production. To optimize the production of golden berries at export levels, Artificial Intelligence (AI) can be used. Many studies worldwide have used AI algorithms such as Radial Base [2], BackPropagation [3,4,5] or Convolutional Neural

Networks (CNN) [6,7] to allow for high levels of recognition of diverse types of fruits. None of these schemes have been applied for the recognition of golden berry quality.

One of the most common applications for digital image processing (DIP) within the field of agroindustry is to find defects on the surface of fruits. This is used both to identify them more easily, helping the general count and to discard them if they compromise the condition and quality of the product [9]. Fruit spots can be used to recognize their position within the sample image, just as they can be used to estimate the quantities and positions of all fruits in an image [4]. The application of DPI in the detection and acquisition of images in fruits presents certain characteristics, such as shape, size, color, and features in the images, where pixel segmentation, which is based on the division of blocks, is applied. Furthermore, to estimate the number of real fruits, a statistical method such as linear regression is applied [4]. Also, physical properties such as volume and mass must be considered, as well as chemical properties such as conductivity, solidity, density of dissoluble solids and titratable acid [10]. After selecting the features to be considered (weight, color and size), machine learning algorithms are used to learn the product classes [9].



Consequently, automatic food selection represents a considerable advance in almost any industry. Given the case of the selection module of [3], this is a peach sorting machine that works using digital signal processing to discriminate fruits by color and size. After analyzing the results obtained, a notable improvement in productivity was obtained compared to manual work, given the speed and precision of the machinery. In the same way, [11] obtains an accuracy of 94%, using the Mask Region-based CNN (R-CNN), in its module focused on the automated selection of peaches, whose properties have certain similarities with golden berries. Similarly, [5] achieved a detection rate of 94%. This result is outstanding and has a great projection for the practice of the model in automated systems using feature extraction methods and transfer learning in Deep Learning. In turn, [4] with its complete recognition of tomato fruits by their state of maturity, obtained the lowest percentage among the reviewed studies using a Radial Base Network; the authors of the study applied an image segmentation method, where in the tests, they obtained an accuracy of 83%. However, it must be considered that the study was carried out with fruits not picked from the tree, which greatly increases the difficulty of recognition. Finally, in ripe fruits, a 100% detection rate was obtained due to their characteristic red color that facilitated their differentiation [4]. When considering diseases, obtained a 96% validation using the pre-trained network GoogLeNet CNN for the classification of five sugarcane diseases. On the other hand, for the correct approach of a network focused on object detection, a wide variety of factors must be considered. For example, always taking into account the shape of the product influences the type of system that must be implemented; that is, if the object is spherical, it is more convenient to use a

hemispherical diffuser camera to illuminate the scene uniformly, facilitating the visualization of the image, since the measurement of this attribute in spherical objects is less complicated, but the complexity varies depending on the regularity in the shape of the products [3]. The normality of the surface and its curvature are the most important characteristics. The smooth areas allow an easier calculation to represent the properties with respect to the geometric qualities of the fruit [12]. The color of the fruit of organic products is a key factor because customers are, to a certain extent, susceptible to choosing or not choosing a particular fruit based on its surface color [9]. A critical step in defining the training of the network is the selection for the application of a learning model such as CNN, EfficientNetV2, Radial Basis Neural Network (RBNN), and multilayer CNN. Then, depending on the adjustable model, the number of convolutional layers grouped and connected to the network is defined [13]. Also important in the design is the application of concatenation, convolution, and additional operations, in addition to the assignment of biases and weights to the variables for optimal training of the algorithm; these allow for acquiring high precision for the detection of a contiguous feature map, translated into colors and edges of images [13]. In addition, transfer learning is taken into consideration; these are applied to the neural network models VGG16, VGG19, InceptionV3 and MobileNetV2, which have demonstrated that when working with modified images, they achieve better results in a short time for the corner detection method by Shi-Tomas [6]. It should be noted that with the composition of the CNNs Mask R-CNN and VGG-19 models, good results are obtained in the detection of diseases in peaches for automatic harvesting since it has a 94% accuracy in detection.

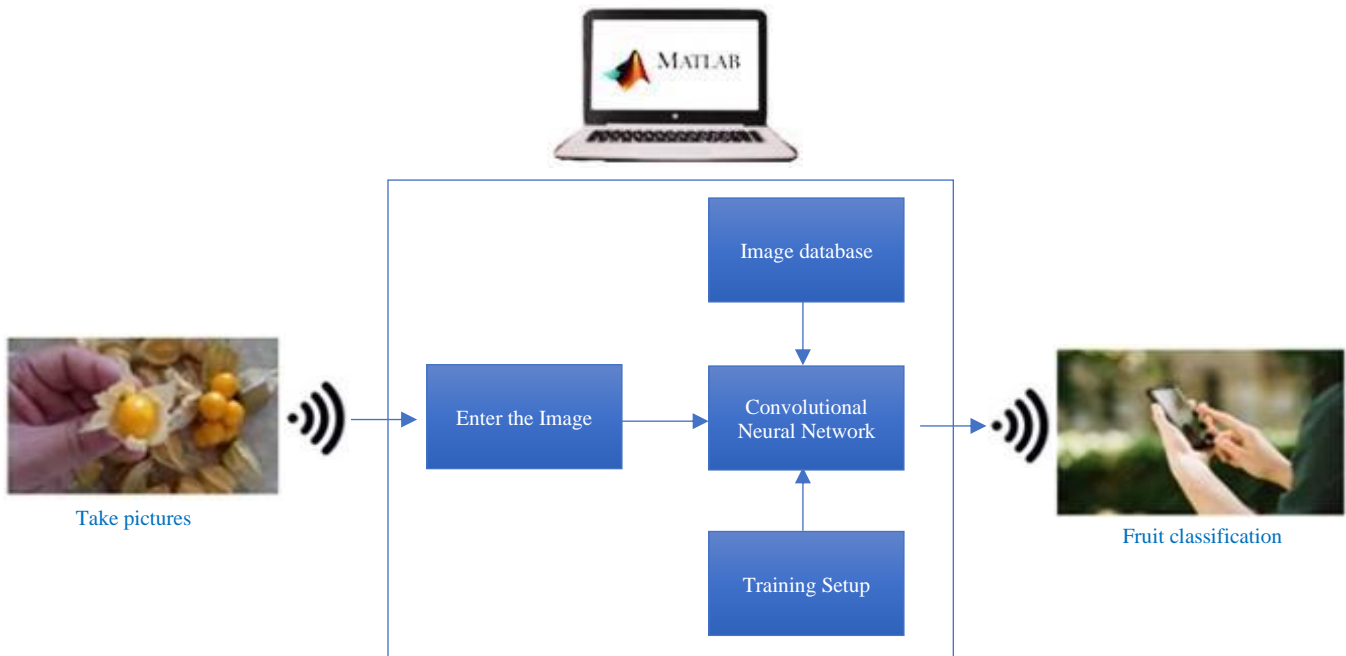


Fig. 1 Block diagram of the golden berry classification system



Fig. 2 Phenotypic states of golden berry. (a) Mature, (b) Immature, (c) Mouldy, and (d) Decomposed



Fig. 3 Images of peruvian golden berries

This showed that the combination of models is useful for having a regular training time and a medium error rate compared to other models [11]. On the other hand, the use of pre-trained networks such as GoogLeNet CNN is becoming very common due to its ease of implementation, which can be embedded in a Matlab App designer web application. In view of the above, in this work, the design and implementation of a convolutional neural network algorithm embedded in a user interface applied to the selection of golden berries based on their quality to optimize their production in the Peruvian food industry is presented.

## 2. Methodology

The proposed system to recognize and classify the quality of the Peruvian golden berry is shown in Figure 1.

### 2.1. Training Parameters

Parametric data is defined in measurable characteristics such as ripe, shape, and color. Also, for an appropriate classification according to maturity, it is of utmost importance to identify the variation of colors [14]. In short, by adequately selecting the characteristic parameters of the golden berry, they help generate measurable data for the CNN training, thus describing the phenotypic state of the fruit. Figure 2 shows four images of golden berries taken with a hemispherical camera. Some phenotypic traits of golden berries are identified in Figure 2: mature (a), immature (b), mouldy (c), and decomposed (d).

### 2.2. Image Acquisition

To take images of golden berries for subsequent classification, a hemispherical camera will be used because it provides optimal resolution and smoothness for its preprocessing, which facilitates the manipulation of its data [15]. For a better result, artificial lighting using LED lights will be used since they represent lower energy consumption. The images are sent directly to the computer, which must be connected to the network for processing through a server.

### 2.3. Database

The database is made up of a sample of 241 images for pre-training, where 50% will correspond to fruits in good condition, and the rest will be in poor condition. This is intentionally done to generate recognition data for the entries of the artificial neural network. After performing the first training iteration, the number of samples will be doubled so that the CNN improves in accuracy for each iteration. Each image file must have an .xml file extension for the reading and interpretation of the information by the CNN input layer. Figure 3 shows 20 images from the database of golden berries in good condition, decomposed, of irregular shape and of varied colors taken by the camera at different angles and with varied light intensities so that the CNN can identify cases in the event of an environmental disturbance and obtain better precision. These images represent 8% of the database that will be used for training.

### 2.4. Convolutional Neural Network (GoogLeNet)

#### 2.4.1. Introduction to GoogLeNet

GoogLeNet is a convolutional network widely used due to its availability and flexibility. This CNN has 22 completely customizable layers. For data entry, the images are required to be sized at 224x224 pixels.

#### Input Layer

In this layer, the preprocessing of the images is carried out to adapt them to the extraction of features. Therefore, the images are subjected to resizing and rotations until achieving a resolution of 224x224 pixels with a depth of 3 layers, since it is an RGB image [16].

#### Extraction Layer of Characteristics

Within this layer, filters are applied to extract the desired features. In this project, the detection of edges, shapes and colors will be sought [13]. The size of the filters varies incrementally. Finally, convolution and normalization are applied. It should be taken into account that at the end of each section of the extraction, we seek to create an unidimensional vector. This is so that the layers can be kept connected for classification.

#### Output Layer

Within this layer the success rate regarding the classification is shown to avoid overfitting due to unexpected data, something normal in image processing.



Fig. 4 Block diagram of the CNN programming

A layer called “Drop-out” is used, which will add the stability to the network and will only be activated in the training.

2.4.2. CNN Programming

The programming of the CNN was carried out in five stages (Figure 4).

Data Preprocessing

To enter the data into the GoogLeNet CNN, the *imresize* function was used; with this, the properties of the data obtained can be modified, thereby resizing the images to 224x224 pixels, which is the standard format.

Data Augmentation

The Data Augmentation method is used to obtain enough training data. With this, the number of samples can be increased by applying transformations for data augmentation, such as cutting, white balance, scaling, rotation, and reflection [8]. For this, the *ImageDataAugmenter* function was applied, which allows the performing of these same transformations mentioned on a large scale. With this, enough data could be gathered for optimal training.

Data Partition

Partitions will be used from the total data collected in our database. 80% of the data will be used for training, 10% for validation and the remaining 10% for post-training tests. This is achieved using the *SplitEachLabel* function.

GoogLeNet Requirements

The layers of the GoogLeNet network must be modified with respect to the needs of the proposed design, especially the fully connected layer. On the other hand, a classification layer calculates the cross-entropy loss of classification tasks. The learning rate of the weights per layer is obtained with the *WeightLearnRateFactor*. Similarly, *BiasLearnRateFactor* is the product of the rate obtained to find the learning rate of the biases in the layers.

CNN Training

In this last stage, the *TrainingOptions* function is used, within which the training parameters are entered; these can be changed later to refine the results. The main purpose is to establish an initial learning rate for training that should be on a positive scale. The lower this index is, the longer the training will take, but the more reliable the results will be. The *ValidationFrequency* parameter corresponds to the number of iterations between evaluations of the validation

metrics. This is found by dividing the total number of training data by the *MiniBatchSize*. *MiniBatchSize* is used to set the size of the data groups for each training iteration. The *ValidationData* parameter corresponds to the portion of the database that will be used in training to verify its validation; this corresponds to 10% of our total data. Although it is true that the greater the number of iterations, the more precise result is obtained, to avoid overfitting (overtraining) during training, the training ratio is varied, or the number of iterations is decreased.

2.4.3. Image Capture and Real-Time Recognition

A block diagram of the processing of the images in real-time is presented in Figure 5. Initially, the images are loaded. At this point, it is suggested that the previously performed training of the network with the best validation result must be used.

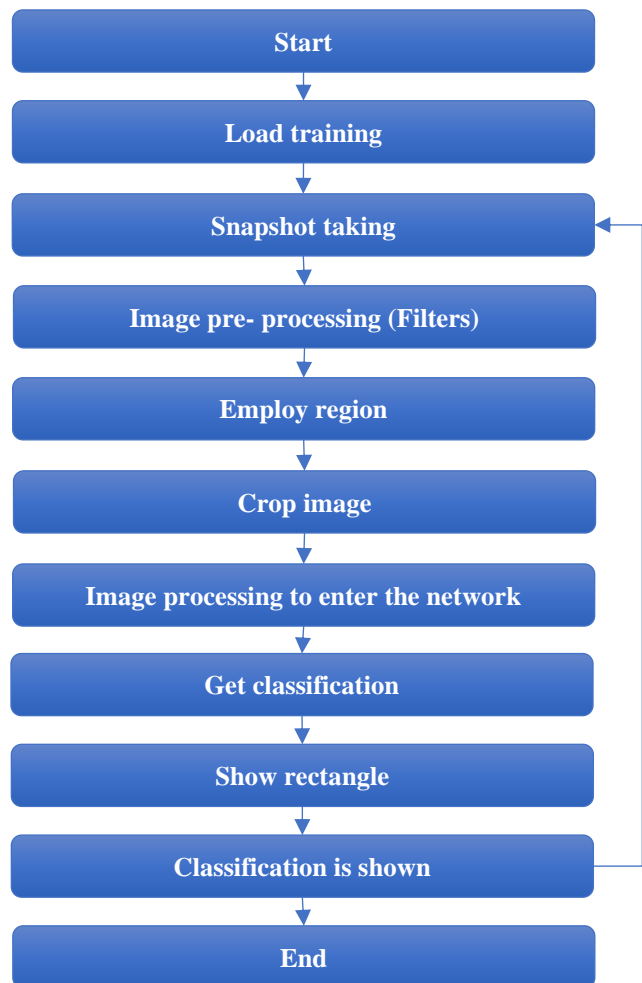


Fig. 5 Block diagram of image processing in real-time



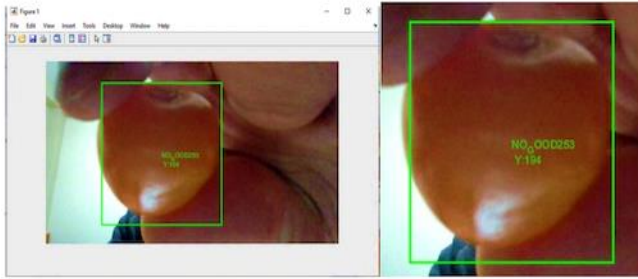


Fig. 6 Recognition and classification of the golden berry fruit

This will be called using the *Load* command and will load all the layers of the network. The way to obtain images directly from the webcam is through the “MATLAB Support Package for USB Webcams” toolbox, which will allow access to video inputs. And using the *snapshot* command, the image taken from the webcam is captured and saved in real-time with the *capture* variable. Once the image capture is obtained in real time, it must be processed for the detection of the quality of the golden berry. This begins by dividing the image into RGB layers, and then the layers are operated until a contrast range is obtained that allows only the desired object to be displayed. Once this is found, binarization is applied. Subsequently, a *medfilt2* filter is applied to filter the median of the image. To further improve the detection, the squeeze morphological transformation is applied, which proceeds to erode the image, eliminating undesirable particles and then dilating it, recovering the initial proportions. As a morphological element for this transformation, a circle of radius 1 was used since this adds softness to the contour obtained and fits perfectly with the golden berry samples.

To obtain the image of the golden berry identified as a set, *bwlabel* is applied to obtain a matrix of labels, which contains the labels of the objects found in the image. To obtain the characteristics of these identified objects, the *regionprops* function is used, which obtains properties such as the centroid and the area. With *regionprops*, the 4 points that enclose the detected figure in a rectangle are calculated. These 4 values are stored in a 1x4 matrix, which is named *location*. These values are updated with each webcam capture. Using the *imcrop* function, the original image is cropped using the *location* variable as limiting points. The cropped image is saved for uploading later to the network. The dimensions must be adjusted to make it compatible with the network input. Using the *imresize* tool, the image size is modified to 224x224 pixels.

Once the image has been conditioned, it is inserted into the network to obtain the golden berry classification. The trained network returns 2 values, “Label” and “Prop”. The first indicates the class to which the entered image belongs, and the second indicates its recognition rate. Keep in mind that “Label” is a 1x1 matrix, so to return its value in

character, the *char* command is used. To show a rectangle once the recognition is done, the *location* variable is used again, where the 4 points that surround the recognized object are saved. With the *rectangle* function, a rectangle is drawn on the screen, and its *location* is used to position it around the detected figure. Additionally, the lines of the rectangle are given a green color. The *regionprops* command returns the calculated centroid in the form of a 1x2 matrix. This will be saved under the *center* variable. For visualization, the *text* command is used, indicating its position as the *center*. In addition, it is indicated that it returns its position and the class obtained from the network saved in the “Label” variable, as shown in an example of a golden berry image in Figure 6.

#### 2.4.4. User Interface with Matlab App Designer

The graphical user interface is an environment where the end-user interacts with the recognition algorithm in such a way as to acquire information on the natural state of the quality of the golden berry. Figure 7 presents the flow chart of the interface design. Diverse components were selected from the Matlab App designer library to generate the interface. In the program function stage, the shot of the real-time image was processed. Once the “Select” button is clicked, the embedded CNN will classify the golden berry fruit as GOOD (mature and in good condition) or NO\_GOOD (in poor condition and immature).

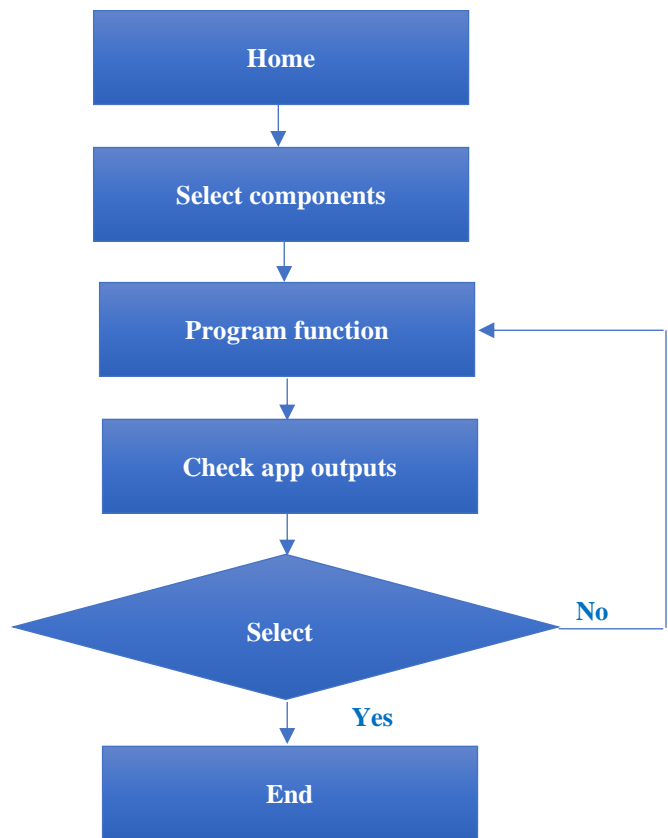


Fig. 7 Flow chart of the design of the graphical user interface

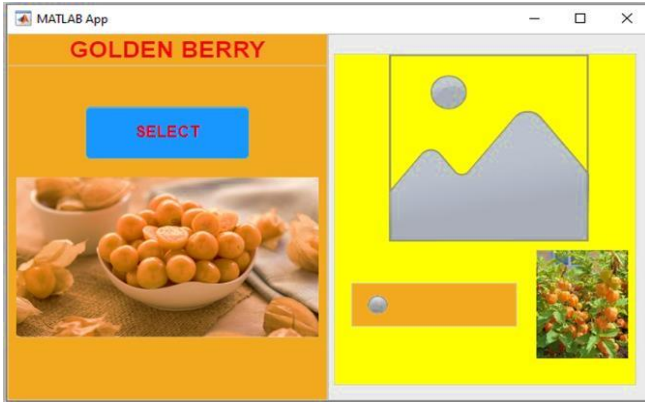


Fig. 8 Graphical user interface generated with Matlab App designer

The graphical user interface produced is presented in Figure 8. In the left panel, there is a “Select” button that, once clicked, takes a picture in real-time, and in the right panel, the picture, along with the appropriate classification, is shown.

### 3. Results and Discussion

#### 3.1. Training Tests

In Figure 9, the first training of the network was carried out, where the precision curves (traces that follow the blue curve) and loss (traces that follow the orange curve) are

verified. What is expected from training is that the learning curves begin to converge with each other. The epoch number is 14 with 3 iterations each, which corresponds to a total of 42 iterations. This is a fairly low number for training, but the advantage is that the training time is greatly reduced to 4 minutes. The consequence is that the learning curves converge with each other to a certain extent.

Still, it is not a perfect similarity, and this is reflected in the validation rate of only 67.4%, which tells us that the recognition will have a questionable but positive accuracy. This result is far from that obtained by [11] and [5], who both obtained a recognition rate of 94% for their training. However, it must be considered that in the case of [11], they are working with peaches that are larger in size, and the intrinsic characteristics of the ripe ones vary among themselves, which added difficulty to their training.

It was previously established that 10% of the total database would be used in testing the trained network, so, using a simple selection code, a random sample of 16 photographs that were not introduced in the network was collected. In Figure 10, they are shown with their respective classification (GOOD for the mature ones and NO\_GOOD for the ones in bad condition), along with their validation rate.

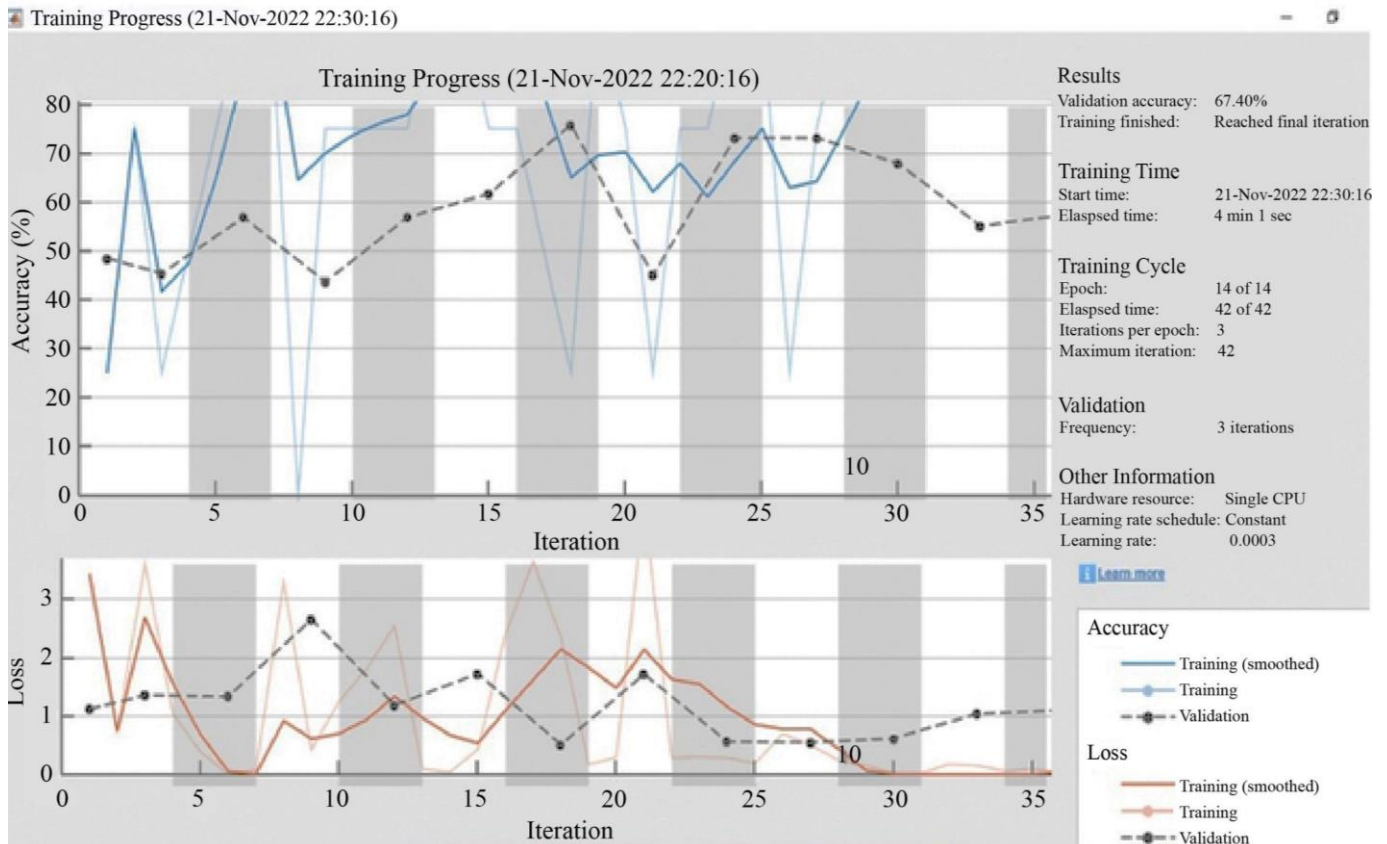


Fig. 9 CNN test training (validation rate 67.4%)

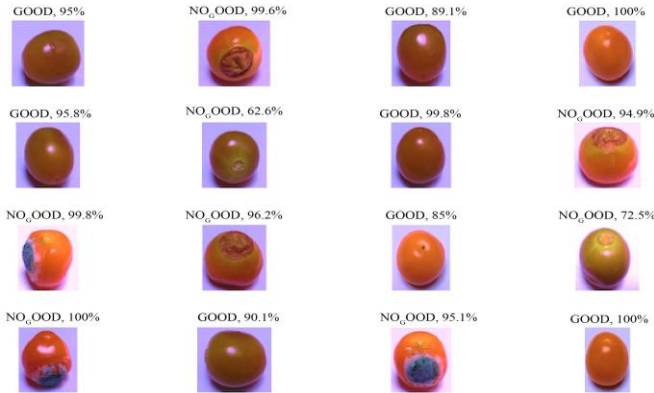


Fig. 10 Classification of golden berry quality (Random 16 Samples)

In Figure 10, it can be seen that the network has a correct response by detecting deformations and putrid sections of golden berries. But it is also imprecise in some images. Something that these images have in common is that the shape remains similar to that of a mature golden berry, and only the saturation of the fruit varies slightly. Therefore, it can be stated that the network has problems identifying the color ranges close to the correct range of the mature golden berries, but it correctly identifies the deformations. This is mentioned by [4], who indicates that the minimum contrast variation makes the correct classification of the network difficult. In order to obtain a better validation rate, it was decided to increase the number of iterations since the more iterations the network has, the more precise the validation rate is, generally. In Figure 11, we have the completion of a second CNN training with a time of 1457 minutes, and it can

be seen how, instead of converging, the learning curve diverges completely.

Due to the divergence of the learning curve, the training becomes obsolete, and this is reflected in the learning rate obtained, 50.66%, a lower percentage than the previous training that only had 47 iterations. This phenomenon is known as overfitting, and it usually happens when working with a lot of data. For this case, the number of total iterations of 7000 with 1000 epochs of 7 iterations each was the cause. The problem with working with such high quantities is that the network starts not only learning the features but also memorizing them, as well as patterns such as noise or inaccurate inputs [5]. This causes the model to not categorize properly data outside of the training set, and this is why it diverges from the validation data. One of the possible solutions to overfitting is to increase the learning rate or decrease the number of total iterations. Past study had already increased its learning rate to avoid overfitting as a result of a longer training time. For its part, [16] indicates avoiding an excessive number of iterations, in addition to considering the epoch proportion as a determining factor.

Figure 12 shows the result of the second CNN training with the testing set; again, 16 random samples are extracted from this set. As in the previous training, it is seen how it successfully recognizes and detects golden berries with superficial marks and deformations, but it fails to detect immaturity due to certain similarities in shape and color with mature ones. Hence, based on what was observed, this training can be discarded due to its low validation rate, lower than the first training.

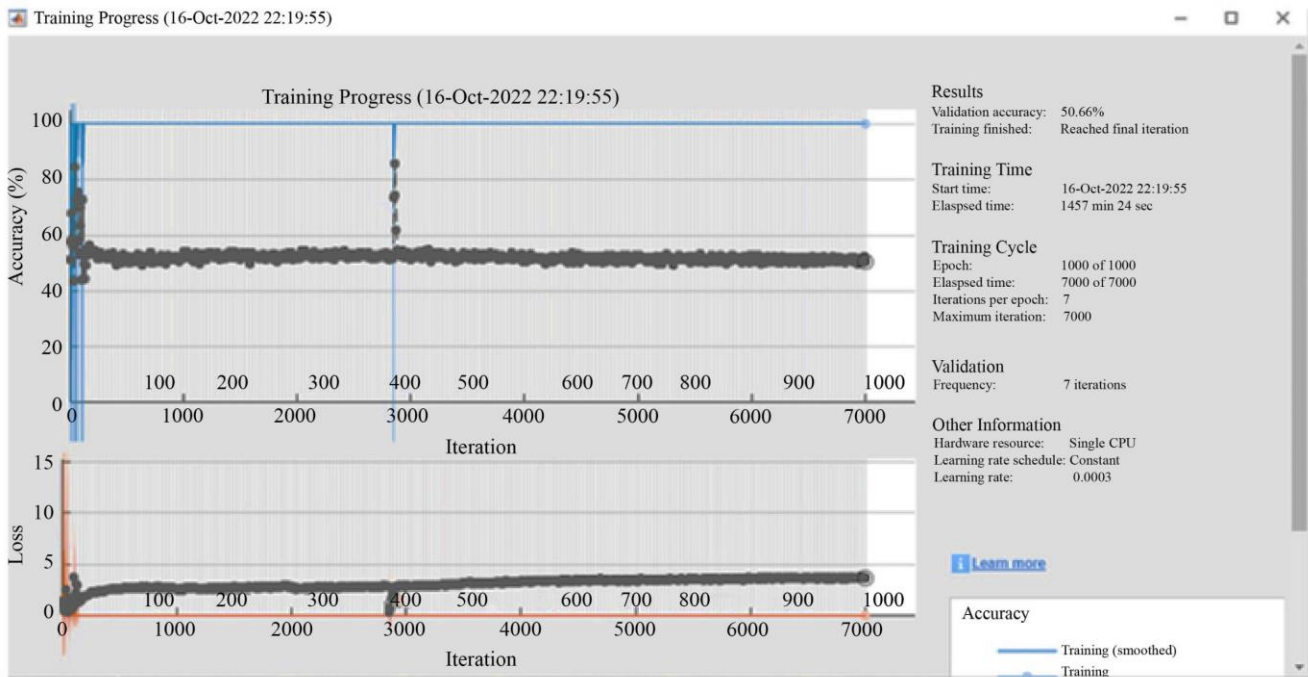


Fig. 11 CNN test training (validation rate 50.66%)



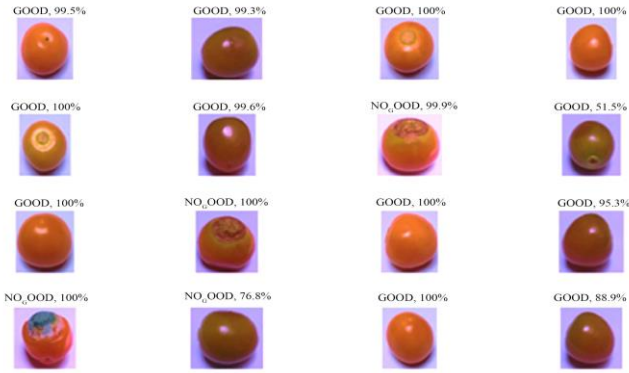


Fig. 12 Random 16 samples extracted from the second CNN training

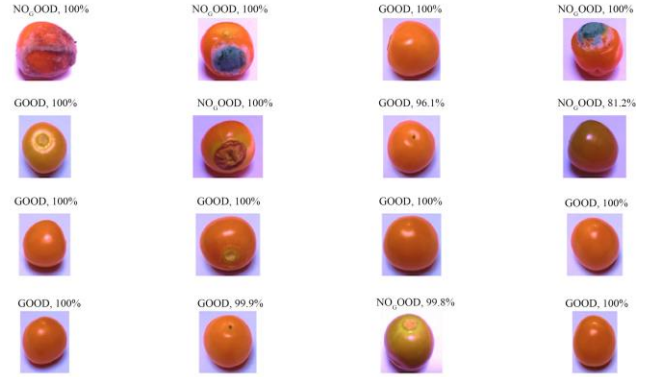


Fig. 14 Results of 16 random samples extracted from the testing database

After the failure of the previous training, the overfitting problem had to be solved, for which it was decided to reduce the number of total iterations from 7000 to 6000. Although it may seem like it is a similar value, the distribution of these iterations was also modified, This time being 500 epochs with 12 iterations each, to balance the training. In Figure 13, a result of this can be seen.

In Figure 13, this third training shows how the curves converge in their entirety, unlike the previous training. Therefore, this training was successful. The completion time and total duration of the training amounted 396 minutes and

38 seconds. Furthermore, a 100% validation rate is obtained, making it the best result.

Figure 14 shows the testing results with 16 random samples extracted from the testing database. These images were not used to train the network. In Figure 14, it can be seen how the network is able to correctly classify all the images, even those in which the previous training failed. It is then stated once again that the network is successful, and it is this that will be used for the detection system of golden berry quality.

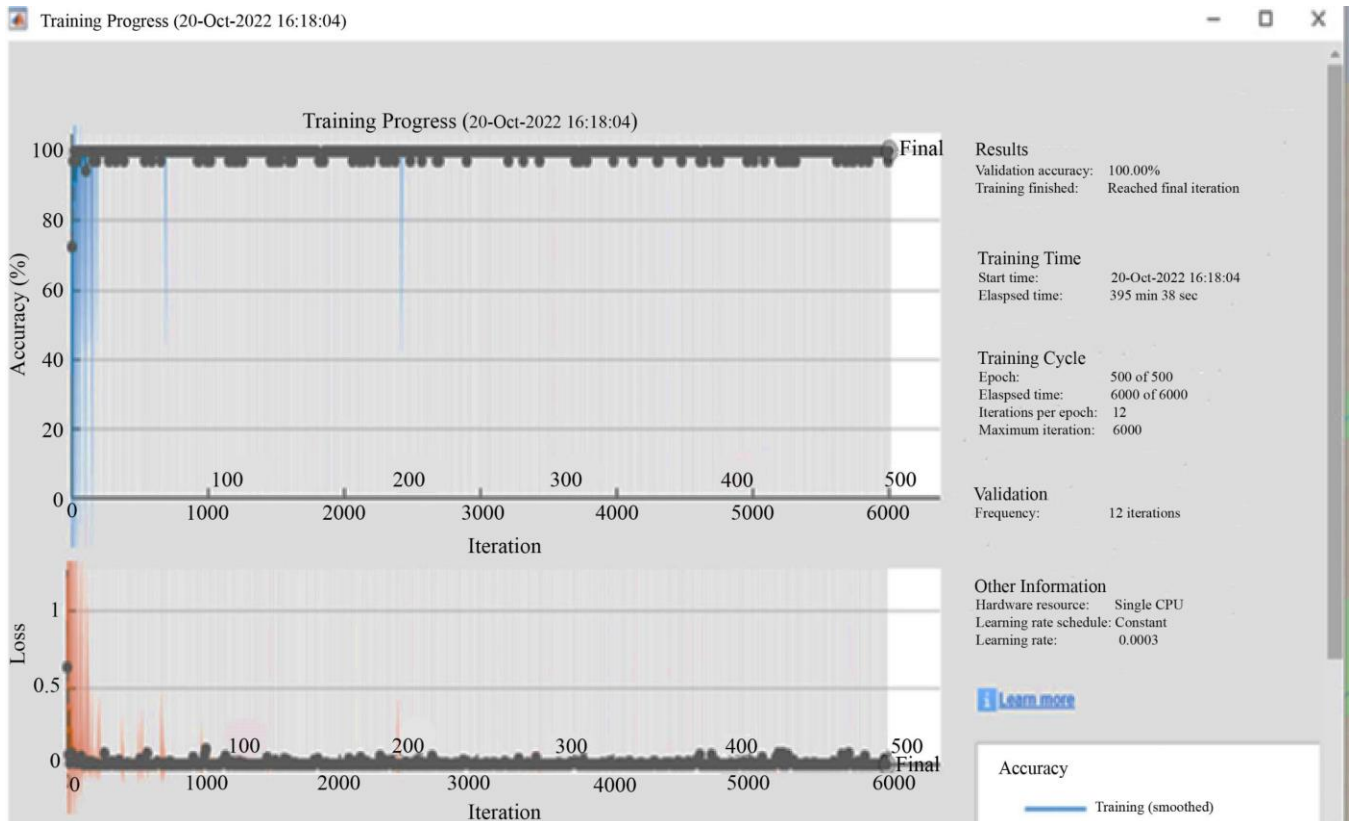


Fig. 13 CNN test training (validation rate 100%)



### 3.2. Real-Time Recognition and Classification

Once the best training model was chosen, the verification of the model was carried out, as can be seen in Figure 15; under stable lighting conditions, the network correctly recognizes and surrounds the fruit. Regarding the classification, this is carried out correctly at a refresh rate of 0.0001 seconds.

Even though the validation rate of the network was 100%, it is not predisposed to failures in detection and classification since this depends on the environment in which it is carried out and the quality of the camera. The real-time detection system depends on the color and contrasts with the environment for the detection of the object.

Thus, very high brightness or too many shadows affect the recognition in the images, which is why the crop given by the detection of the *bwlabel* function is wrong since it will only consider the set of pixels that meet the established color parameters in the labels.

In Figure 16, an excessive light source can be seen that oversaturates the image, affecting the color reception of the detection system. Also, it can be seen how the recognition region does not surround the entire fruit; this is because the *bwlabel* function does not detect the oversaturated or overly opaque parts since they were not considered in the binarization, which, in result, gives an imprecise classification.

Another problem that may arise is the motion blur because it depends on the quality of frames per second that the camera can handle. If it is too low, the image is deformed by the movement itself. Figure 17 shows how the blur produced by the speed of movement deforms the image, causing a wrong recognition and classification.

The designed graphical user interface was tested with the third trained network, which is the one that obtained the best results. Unlike previous detection systems, the tracking box or coordinates will not be displayed in the interface. The trimming will be carried out to obtain the classification of the golden berry quality using an indicator LED in the interface. As can be seen in Figure 18, the detection is correct under other distinguishable features. In addition, it is possible to distinguish that the LED is on (green color).

In Figure 19, the recognition of the golden berry in a poor condition is clearly identified. The red LED indicates a NO\_GOOD detection. Once again, the image was cropped and entered the network, so external objects do not affect the recognition as long as they are not widely invasive. In Figure 20, it can be seen how when the *bwlabel* function does not detect any object, and thus the system interprets this as an absence of data, the LED turns gray to indicate the absence of a golden berry.

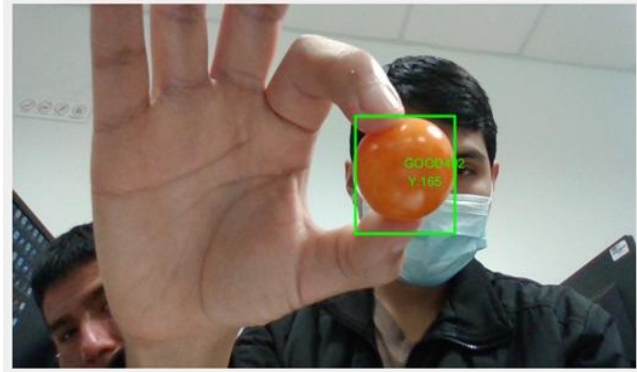


Fig. 15 Real-time recognition and classification

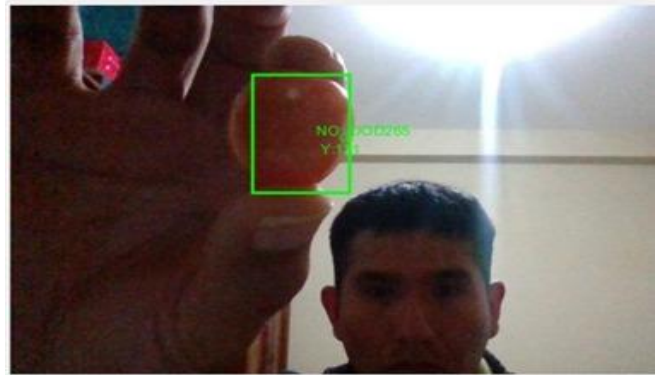


Fig. 16 Faulty real-time recognition due to excessive light

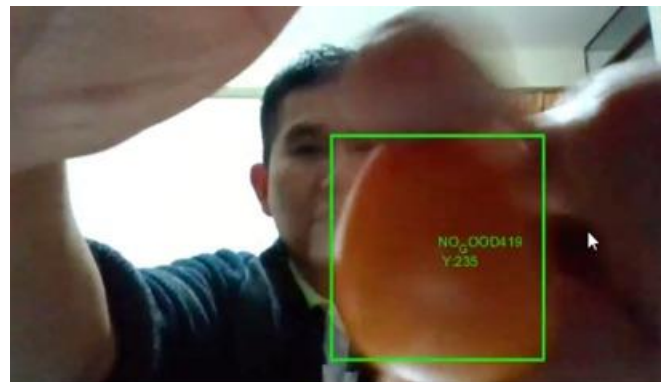


Fig. 17 Faulty real-time recognition due to motion blur

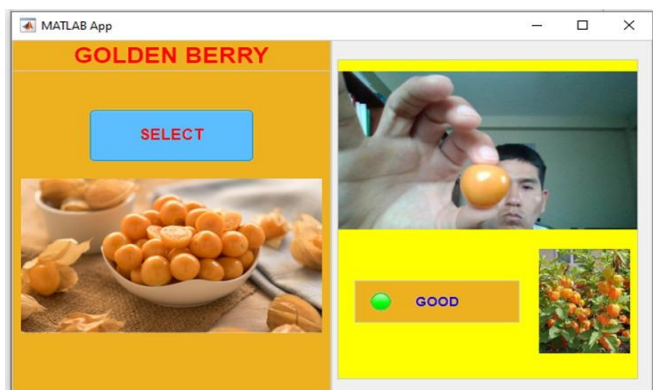


Fig. 18 Real-time recognition of a golden berry in GOOD condition using the graphical user interface

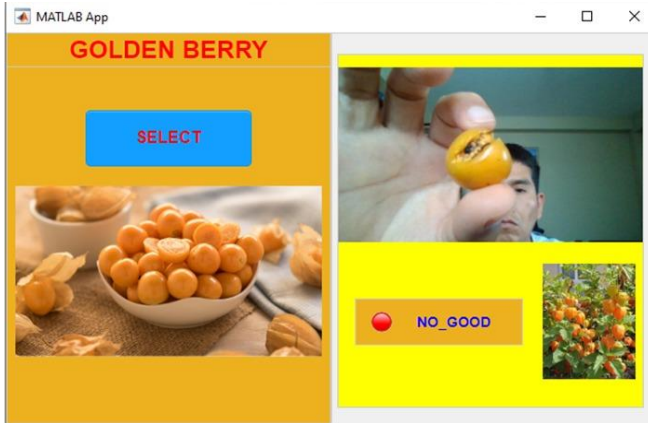


Fig. 19 Real-time recognition of a golden berry in a NO\_GOOD condition using the graphical user interface

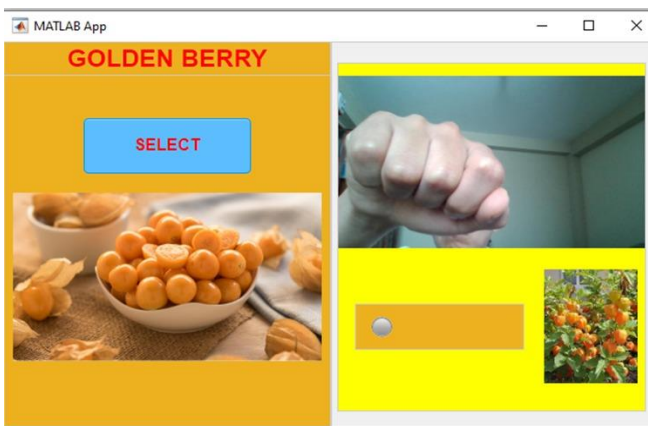


Fig. 20 No real-time recognition in the absence of a golden berry

The tests in the graphical user interface for the recognition and classification of the golden berry quality indicate that the model performs the classification correctly on the images that it manages to capture in real-time. It is worth mentioning that although the environment where the classification tests were carried out contains objects of similar colors that could confuse the network, such for instance a yellow curtain located in the back background, the CNN recognizes the characteristic patterns of the golden berry without alterations, that could be produced due to ambient lighting, and the type and resolution of the camera [3].

#### 4. Conclusion

In this work, it was possible to correctly identify and classify the Peruvian golden berry based on its quality (GOOD and NO\_GOOD). Likewise, the factors that make the selection difficult were identified, such as the quality of the camera and the poor lighting of the environment. The real-time classification tests with the Matlab App designer interface system were successful by performing the classification of the golden berry quality appropriately in a low-light environment and with background objects that could confuse the network. The recognition speed range of the designed system proved to be higher than that obtained with manual selection. However, this will depend on the configuration given by the operator since another virtue of the developed algorithm is that it allows defining the recognition speed. Finally, it is recommended to increase the database for the training of the network and to include images with lighting variations in the environment to perform optimal training and reduce overfitting.

#### References

- [1] Market Analysis, Commercial Intelligence Unit – Ministerio de Desarrollo Agrario y Riego (MIGRADI), 2021. [Online]. Available: <https://cdn.www.gob.pe/uploads/document/file/1720425/An%C3%A1lisis%20de%20Mercado%20del%20Aguaymanto.pdf>
- [2] Juan Cevallos Ampuero, “Radial Based Neural Networks Applied to Quality Improvement,” *Revista de la Facultad de Ingeniería Industrial*, vol. 11, no. 1, pp. 63-72, 2008. [CrossRef] [Google Scholar] [Publisher Link]
- [3] Granados Giraldo et al., “Design and Construction of an Export-Type Peach Sorting Machine Using Artificial Intelligence,” *Universidad Autónoma de Bucaramanga, Bucaramanga, Colombia*, pp. 1-74, 2020. [Google Scholar] [Publisher Link]
- [4] Kyosuke Yamamoto et al., “On Plant Detection of Intact Tomato Fruits using Image Analysis and Machine Learning Methods,” *Sensors*, vol. 14, no. 7, pp. 12191-12206, 2014. [CrossRef] [Google Scholar] [Publisher Link]
- [5] Khaled Marji Alresheedi et al., “Dates Fruit Recognition: From Classical Fusion to Deep Learning,” *Computer Systems Science and Engineering*, vol. 40, no. 1, pp. 151-166, 2022. [CrossRef] [Google Scholar] [Publisher Link]
- [6] Farian Severine Ishengoma, Idris A. Rai, and Rutabayiro Ngoga Said, “Identification of Maize Leaves Infected by Fall Armyworms using UAV-Based Imagery and Convolutional Neural Networks,” *Computers and Electronics in Agriculture*, vol. 184, pp. 1-8, 2021. [CrossRef] [Google Scholar] [Publisher Link]
- [7] Cristopher Bazan Yaranga, Zaid Sanchez, and Ricardo Rodriguez, “Transfer Learning Using Convolutional Neural Networks for Distracted Driver Recognition,” *TECNIA*, vol. 28, no. 2, pp. 1-5, 2018. [CrossRef] [Google Scholar] [Publisher Link]
- [8] Agnieszka Mikołajczyk, and Michał Grochowski, “Data Augmentation for Improving Deep Learning In Image Classification Problem,” *2018 International Interdisciplinary PhD Workshop (IIPhDW)*, Świnouście, Poland, pp. 117-122, 2018. [CrossRef] [Google Scholar] [Publisher Link]
- [9] Sergio Cubero et al., “Advances in Machine Vision Applications for Automatic Inspection and Quality Evaluation of Fruits and Vegetables,” *Food Bioprocess Technology*, vol. 4, no. 4, pp. 487-504, 2011. [CrossRef] [Google Scholar] [Publisher Link]

- [10] Guoxiang Zhang et al., "A Comprehensive Peach Fruit Quality Evaluation Method for Grading and Consumption," *Applied Sciences*, vol. 10, no. 4, pp. 1-11, 2020. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [11] Wael Alosaimi, Hashem Alyami, and M. Irfan Uddin, "PeachNet: Peach Diseases Detection for Automatic Harvesting," *Computers Materials and Continua*, vol. 67, no. 2, pp. 1665-1677, 2021. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [12] Jianhong Yu et al., "Target Positioning and Sorting Strategy of Fruit Sorting Robot Based on Image Processing," *Traitment du Signal*, vol. 38, no. 3, pp. 797-805, 2021. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [13] C.K. Sunil, C.D. Jaidhar, and Nagamma Patil, "Cardamom Plant Disease Detection Approach Using EfficientNetV2," *IEEE Access*, vol. 10, pp. 789-804, 2021. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [14] M. Ruiz-Altisent et al., "Sensors for Product Characterization and Quality of Specialty Crops-A Review," *Computers and Electronics in Agriculture*, vol. 74, no. 2, pp. 176-194, 2010. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [15] Hancong Wang et al., "A Novel Wireless Leaf Area Index Sensor Based on a Combined U-Net Deep Learning Model," *IEEE Sensors Journal*, vol. 22, no. 16, pp. 16573-16585, 2022. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [16] Fidel L'opez Saca et al., "Convolutional Neural Network with Multi-Column Feature Extraction for Image Classification," *Research in Computing Science*, vol. 148, no. 7, pp. 391-404, 2019. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]