VISIÓN ELECTRÓNICA VOL. 18 NO. 2 (2024) • JULY-DECEMBER • P.P. 180-187 • ISSN 1909-9746 • ISSN-E 2248-4728 • BOGOTÁ (COLOMBIA)

UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS

Visión Electrónica

https://revistas.udistrital.edu.co/index.php/visele

VISIÓN ELECTRÓNICA

A RESEARCH VISION

Smart system for recognition of ripening level in blackberry fruits

Sistema inteligente de reconocimiento del nivel de maduración en frutos de mora

Rafael A. Núñez R. [1], *Karen J. Camargo* [2], *Oscar D. Rangel J.* [3]

INFORMACIÓN DEL ARTÍCULO

Historia del artículo:
Enviado: 20/06/2024
Recibido: 05/07/2024
Aceptado: 06/10/2024

Keywords:
Blackberry crop
Neural network
TensorFlow
Convolutional Neural network
Artificial Intelligence

Palabras clave:
Cultivo de Mora
Red Neuronal
TensorFlow
Red Neuronal Convolucional
Inteligencia Artificial

ABSTRACT

The present work proposes the development of an algorithm focused on improving the blackberry fruit classification process, given that the harvesting of this fruit is done manually and depends mainly on the tradition and experience of the personnel in charge of the harvesting process. The design of an algorithm capable of recognizing the ripeness level of blackberry fruit through a VGG-16 convolutional neural network is described.

For the design of the algorithm, Item was necessary to create to bank of images to train the network. The network is then trained according to the design and performance parameters of the neural network, and finally it is implemented in an embedded system (Raspberry Pi) using the TensorFlow Lite framework. Field tests were carried out on blackberry crops to validate the performance and efficiency of the proposed algorithm, which indicate an accuracy percentage close to 96.66%.

RESUMEN

El presente trabajo plantea el desarrollo de un algoritmo enfocado a mejorar el proceso de clasificación del fruto de mora, dado que la recolección de este fruto se realiza de forma manual y depende principalmente de la tradición y experiencia del personal encargado de los procesos de cosecha. Se describe el diseño de un algoritmo capaz de reconocer el nivel de maduración del fruto de mora a través de una red neuronal convolucional VGG-16.

Para el diseño del algoritmo fue necesario de la creación de un banco de imágenes para entrenar la red. Seguidamente se entrena la red de acuerdo con los parámetros de diseño y desempeño de la red neuronal, finalmente se implementa un sistema embebido (Raspberry Pi) a partir de marco de trabajo TensorFlow Lite. Se realizaron pruebas de funcionamiento en campo sobre los cultivos de mora para validar el desempeño y la eficiencia del algoritmo propuesto, los cuales indican un porcentaje de exactitud cercano al 96,66%.

1 **Engineer Electronic, Units Technological of Santander, Colombia. Master in Engineering Electronics, University Pontifical Bolivarian, Colombia. teacher Associated: Units Technological of Santander, Colombia. E-mail: rrodriguez@correo.uts.edu.co**

2 **Engineer Electronic, Units Technological of Santander, Colombia. E-mail: ccamargo@uts.edu.co**

Citar este artículo como: R. A. Núñez R., K. J. Camargo, y O. D. Rangel J., "Smart system for recognition of ripening level in blackberry fruits", Visión electrónica, vol. 18, no. 2, pp. 180-187, july-december 2024, https://doi.org/10.14483/22484728.23628

1. Introduction

The technologies of vision artificial they have used widely in the agriculture with end to improve the performance of agricultural production units. In [1] it was proposed use of an algorithm Otsu, that It allows pull apart fruit of the background for then train the CNN Caffenet, resulting in 95% accuracy in recognizing the status of maturation.

Similarly, in [2] they use hyperspectral images to train the Alexnet CNN and detect the variety of the plum together with state of maturity, in conclusion, gets a 89.45% of accuracy in the prediction. By other side, implement a detector of defects in the Mangosteen by using a 4-layer convolutional neural network for validation, with results of 97.5% accuracy, according to [3], light intensity has a very important role important when taking images. Meanwhile, in [4] a method was established to improve the classification of the tomato by half of a CNN with end of improve the accuracy, the Results are 91.9% accurate and a prediction time of 0.01 seconds. For his part in [5], 3 classification methods were used (Bayesian – K neighbors and neural network artificial) reaching the conclusion that the perceptron-type neural network allows the feijoa maturity classification with 90% accuracy. Moving on from the progress reported in the literature, the development of low-cost technologies and portable devices with good performance, as decision-making support tools during the harvest of floors of fruit of Blackberry.

This work presents a descriptive research, with a quantitative approach, and systematic observation method, given that the training process of a convolutional neural network and the algorithm will be evaluated based on the accuracy in terms of percentage of successes.

Firstly, the design of the recognition algorithm that integrates the following phases: image bank compilation, digital image processing, transformation of images digital and the creation of a new model CNN to leave of a pre-trained structure. Second is the implementation of the algorithm previously designed in an embedded Ri system from the TensorFlow framework Lite. Next, the way in which the performance of the algorithm is evaluated is presented. recognition, taking into account different validation scenarios. Finally present the results of the different phases developed with the conclusions of the job.

2. Methodology

2.1. Design of the algorithm of recognition

An algorithm is proposed to recognize the level of blackberry ripening using convolutional neural networks. This consists of four stages: data collection, prosecution of data, transformation of data and creation of a new model CNN.

2.1.1. Collection bank of images

An image bank is built with the specific characteristics of the levels of maturation established by the rule ICONTEC chord with the board of maturation of the blackberry fruits [6]. The images were taken of different blackberry crops located in the sidewalk Planned in municipality of Piedecuesta, Santander.

2.1.2. Prosecution of images digital

The first step of data processing is to review the photographs taken in the previous step and discard those that do not have good quality, for example, images that by his intensity of light distort the colors real of the Blackberry either that for his sharpness No They guarantee correct classification by the algorithm.

Secondly, they extract sections of the image initial belonging to the fruit of Blackberry using the tool of cropping and rotation, with the purpose of eliminating areas that do not provide relevant information to the algorithm. Finally, the data set is constructed according to the characteristics of maturation as shown in Figure 1, a total of 3000 photographs were categorized in 3 classes or labels data of output: Green, Red and Purples.

Figure 1. Tags of exit CNN.

Source: own.

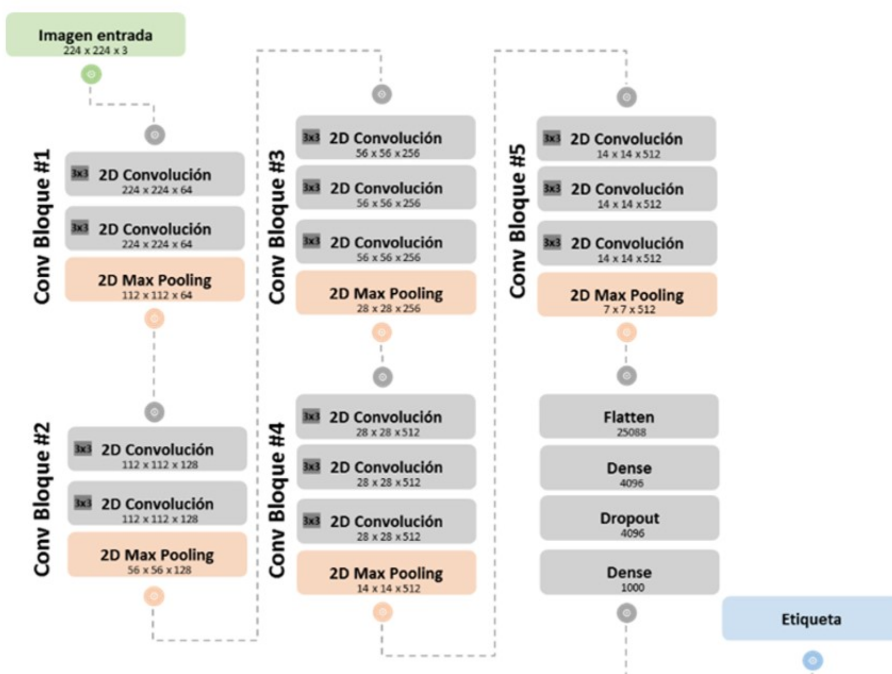
2.1.3. Transformation of images digital

A time classified the images, the CNN still No finds list for develop transfer learning algorithm, given that the images to be input do not comply with the parameters of entrance that requires the grid, without embargo, one of the settings further important is the size of the images which must be 224 x 224 pixels, since is value for flaw of the network, chord with kernel of the grid pre - trained VGG-16.

2.1.4. Parameterization of the grid neural

This section presents how the CNN model should be tuned, trained, and evaluated. For it the pre-trained network VGG - 16 is established. According to Subham Tewari, it achieves a precision of the 92.7% in recognition of objects [7]. The grid neural VGG - 16 this structured in language of programming Python, by it which is necessary count with an editor of code for modify the output parameters of the neural network,

Figure 2. Architecture of the CNN VGG16.



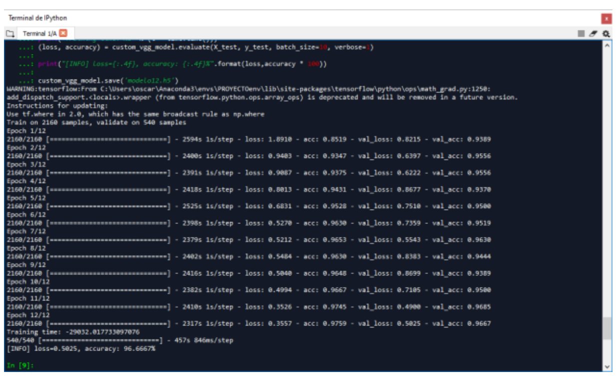
Source: own.

Figure 2 summarizes the parameters and architecture of the grid.

The number of classes to work is defined taking into account that the original network has with 1000 classes as exit. Then, the images selected for train distribute equally between the classes to which they correspond, and a label is assigned to each data set. To guarantee better results when training the algorithm, it is important that the data is organized randomly; Likewise, the data is divided into two sets, 80% of the images are taken for training and the twenty% for validation of the algorithm.

The stage of transfer of learning start to the set-up size of the images input. Subsequently, the pre-trained VGG-16 network is imported from the library. TensorFlow. This network allows you to enter a new image bank and modify the labels of exit of agreement with problem to treat.

Figure 3. Results of training CNN.



Source: own.

This is called transfer learning. To apply this learning, they keep the parameters learned from the original network and only the output block is extracted for modify them with new labels. Finally compile the new architecture of the grid neuronal.

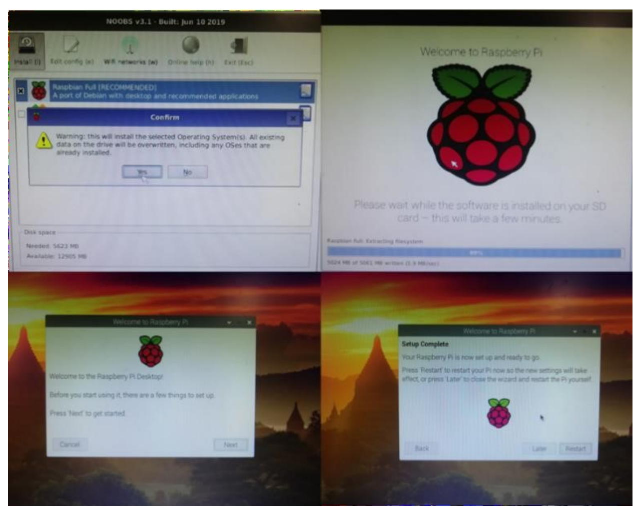
By having an adjusted network architecture, the training stage begins. This It consists of training the network with 80% of the previously processed images. Once finalized training, valid algorithm with twenty % of the images remaining, giving results in terms of percentage of loss and accuracy as observe in the weight matrix adjusted after the training process is stored for then implement them in system embedded Rpi.

2.2. Implementation of the algorithm in system embedded

2.2.1. Setting of the atmosphere of development

With end of develop this section, is necessary the acquisition of a system embedded and of a device that allow visualize functioning of the algorithm previously designed. For this project used the Raspberry Pi 3 model b.

Figure 4. Settings initials YOU Raspberry Pi.



Source: own.

In first place, is necessary install respective system operational (Raspbian) in a SD storage card, said system can be obtained from the official website of Raspberry. Finished the facility configure some parameters of the system, such as, set up language, change location, setting of grids, between others; the which they can see in Figure 4. Then, we proceed to download the necessary drivers for the display to use.

In second place, for achieve implement algorithm precise of the bookshop of OpenCV, the which It allows develop Applications in how much to the recognition of objects, besides of be the major bookshop for achieve with aim proposed. By last, is necessary install the of Tensorflow and Keras libraries, since these allow executing the code that contains the CNN of shape efficient.

2.2.2. Implementation of CNN in atmosphere of development

A time gets design of the algorithm, following passed is find out operation of this. This section describes how the implementation is carried out in the Raspberry Pi, in addition, a graphical interface is designed with the Python tool "Tkinter", so that the user can visualize the operation of the algorithm. In Table 1 describe procedure of the implementation in the Rpi.

Table 1. Variable versus concentrations.

Code implementation CNN in Rpi
Start
1. Matter functions for development of the code
2. Create items Interface visual (Qualification, buttons, typography)
3. Create function for transmission of video
4. Create function for take, keep and sort out Photography (code CNN)
End

Source: own.

Tests are carried out in two different scenarios which allow check the behavior of the trained network. First of all, the operation of the algorithm on the computer, for which it is validated in 4 different weight files using 50 images for each level of maturation; It is important to mention that the images classify not were used during training of the CNN.

According to the results of the previous step, it is determined which of these has the greatest efficiency when classifying images and then being implemented in testing. field, with the purpose of verifying the behavior of the network against various factors climate like sample in the Figure 5.

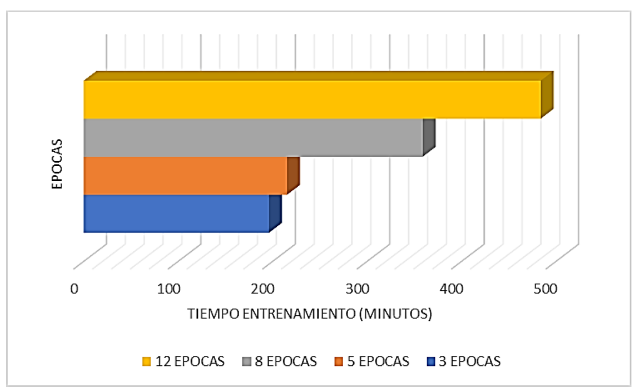
Figure 5. Crops of Blackberry, Planed – Santander.

Source: own.

3. Results

In this section present the results of the process of training of the algorithm, which sample time of duration by each training done together with their percentages of accuracy. As evidence in the Figure 6, to the increase 3 either 4 eras, time that late in train increases in a 25% approximately.

Figure 6. Time of training evidence CNN.



Source: own.

As for the second trajectory, the results are shown in Figure 7, which shows that the control by sliding modes presents better performance for all actuators because it has a smaller error.

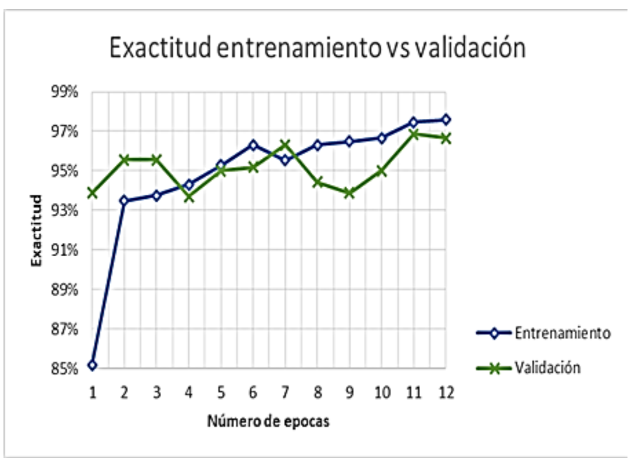
Table 1. Variable versus concentrations.

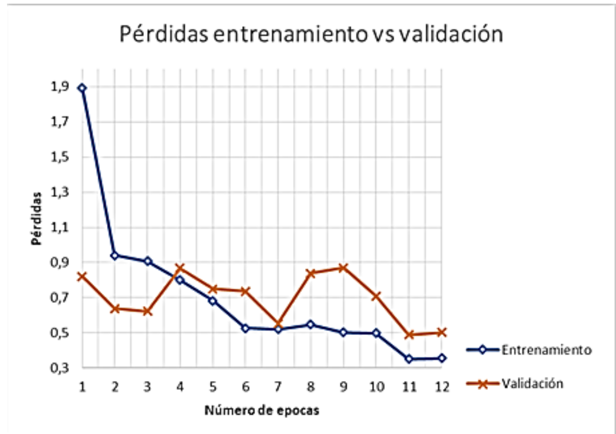
		Training		Validation	
Epoch #	Time (Min)	Data loss	Percentage Accuracy	Data loss	Percentage Accuracy
1	43.23	1,891	85.19%	0.8215	93.89%
2	40	0.9403	93.47%	0.6397	95.56%
3	39.85	0.9087	93.75%	0.6222	95.56%
4	40.3	0.8013	94.31%	0.8677	93.70%
5	42.08	0.6831	95.28%	0.751	95.00%
6	39.97	0.527	96.30%	0.7359	95.19%
7	39.65	0.5212	95.53%	0.5543	96.30%
8	40.03	0.5484	96.30%	0.8383	94.44%
9	40.27	0.504	96.48%	0.8699	93.89%
10	39.7	0.4994	96.67%	0.7105	95.00%
eleven	40.17	0.3526	97.45%	0.49	96.85%
12	38.62	0.3557	97.59%	0.5025	96.66%

Time total of training	483.87 min
Loss	0.5025
Percentage final of accuracy	96.66%

Source: own.

Figure 7. Graphics of accuracy and loss training CNN with 12 epochs.





Source: own.

By other part, in the Table 3, show the percentages of accuracy to the train the grid with different quantities of eras, 3,5,8 and 12; with end of determine which of the present greater efficiency. As can be seen, when training the model with 12 epochs, the percentage of accuracy was of the 96.66%. He same behavior presents in training of the grid neural with 3, 5 and 8 epochs.

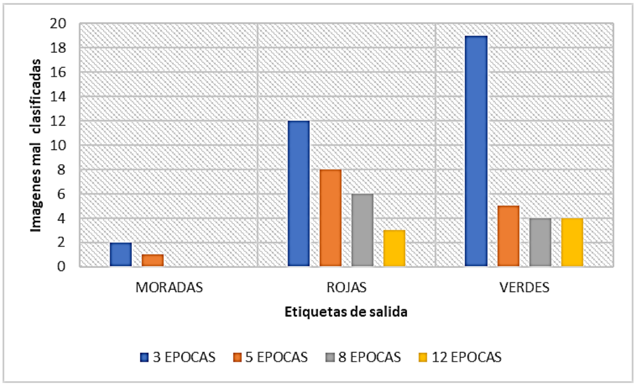
Table 3. Percentage of accuracy training CNN.

Number of eras	Percentage of accuracy (hits)
3	85.55%
5	95.74%
8	95.74%
12	96.66%

Source: own.

Having in account model of elderly accuracy, to continuation, relate the results of the tests described during the validation of the algorithm. Initially, they took 150 images with which the algorithm was validated. As evident in Figure 8, the variables are inversely proportional, that is, as the epochs increase, the bad images classified decrease significantly.

Figure 8. Number of images classified wrongly in CPU.



Source: own.

By other part, they made evidence directly in the crops of Blackberry with the Raspberry Pi and its respective camera module. For a total of 60 images, only 7 of these were wrong classified as seen in Table 4, which demonstrates that the classifier algorithm fruit has precision of the 88.3% approximately.

Table 4. Number of images classified wrongly in RPi.

Label	Epoch 12
Purples	1
Red	4
Green	2

Source: own.

4. Conclusions

TO weigh of the diverse factors environmental with the which believe bank of images, and of agreement with the results described previously concludes that, to elderly amount of eras in training of the algorithm, the greater the the accuracy of this.

The implementation of GPU is recommendable for train the CNN, due to that any modification in the parameters of the grid can take further of 6 hours to complete training again. Therefore, it is logical to conclude that fine-tuning the algorithm No is feasible carry it to cape in development of the project.

To the implement algorithm in the plate, presented drawbacks that are related to the Raspberry Pi camera, since it, as it does not have a light filter and does not count with a system of autofocus, gender images of low quality to the hour of capture fruit directly in the crops. Of mode that, in the classification of images, algorithm daring a 11.6% of mistake during the proof performed.

Recognitions

The results raised in this job derive of the project of degree "System of recognition of the level of maturation in fruits of Blackberry through a grid neural convolutional", framed in the line of investigation of automation and systems of control advanced of the Cluster of Investigation in Control Advanced – GICAV, of the program of Engineering electronics of the Technological Units Santander.

References

[1] X. Li, J. Li, and J. Tang, “A deep learning method for recognizing elevated mature strawberries,” in *2018 33rd Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, May 2018, pp. 1072-1077. https://doi.org/10.1109/YAC.2018.8406530

[2] M. Campos et al., “Detección de variedad y estado de maduración del ciruelo japonés utilizando imágenes hiperespectrales y aprendizaje profundo,” *XVIII Conf. la Asoc. Española para la Intel. Artif. (CAEPIA 2018)* Av. en Intel. Artif. 23-26 oct. 2018 Granada, España, 2018, ISBN 978-84-09-05643-9, pp. 139-144, 2018.

[3] L. M. Azizah, S. F. Umayah, S. Riyadi, C. Damarjati, and N. A. Utama, “Deep learning implementation using convolutional neural

network in mangosteen surface defect detection," in *2017 7th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, nov. 2017, pp. 242-246. https://doi.org/10.1109/ICCSCE.2017.8284412

[4] L. Zhang, J. Jia, G. Gui, X. Hao, W. Gao, and M. Wang, "Deep Learning Based Improved Classification System for Designing Tomato Harvesting Robot," IEEE Access, vol. 6, pp. 67940-67950, 2018. https://doi.org/10.1109/ACCESS.2018.2879324

[5] J. P. Bonilla-González and F. A. Prieto-Ortiz, "Determinación del estado de maduración de frutos de feijoa mediante un sistema de visión por computador utilizando información de color," *Rev. Investig. Desarro. e Innovación*, vol. 7, no. 1, pp. 111-126, Dec. 2016. https://doi.org/10.19053/20278306.v7.n1.2016.5603

[6] ICONTEC, Frutas frescas. Mora de castilla. Especificaciones - NTC 4106. Colombia: Instituto Colombiano de Normas Técnicas y Certificación, 1997.

[7] S. Tewari, "CNN Architecture Series - VGG-16 with implementation (Part I)," Medium, 2019. https://medium.com/datadriveninvestor/cnn-architecture-series-vgg-16-with- implementation-part-i-bca79e7db415