Identificación Automática de la calidad del banano usando clasificación por Redes Neuronales Profundas (DNN) *(Automatic Identification of banana quality with Deep Neural Network Classification (DNN))*

Deyner Julian Navarro Ortiz[1] y Silvia Alejandra Martínez López[2]

Material original autorizado para su primera publicación en la revista Ciencia y Tecnología de la Facultad de Ingeniería de la Universidad de Palermo.
Campo temático: Machine Learning.
Recepción: 27/04/2021 | Aceptación: 12/1/2022

Resumen

La industrialización de la agricultura ha ido tomando lugar en Colombia de forma paulatina, esto surge por la necesidad de optimizar el tiempo y esfuerzo requeridos para los procesos implicados desde el cultivo de los alimentos hasta la distribución y comercialización, bien sean exportaciones o comercio interno. La clasificación de las frutas es un proceso importante antes de ser sacadas como un producto final, ya que se deben seguir normativas predeterminadas por entidades reguladoras que definen la calidad del producto cosechado. Los autores proponen un prototipo de bajo costo para la clasificación automática del banano según la norma NTC 1190 (normativa colombiana), utilizando una red neuronal convolucional (CNN) de arquitectura MobileNetV2, entrenada mediante transfer learning e implementada en una Raspberry Pi 3B + con una cámara para monitorear los ejemplares y una sencilla interfaz de interacción con el usuario, además de una carcasa diseñada para la protección del hardware. El conjunto de datos utilizados en este trabajo para los procesos de entrenamiento, validación y prueba consta de imágenes tomadas de dos bases de datos de frutas de acceso libre, y también imágenes adquiridas por los investigadores. La precisión alcanzada es del 87%, suficiente para garantizar confiabilidad y bajo costo computacional.

Palabras Clave: Bananos, Redes Neuronales Convolucionales, Clasificación, Transfer Learning.

[1] Universidad Industrial de Santander. deyner1996@gmail.com
[2] Universidad Industrial de Santander. silvia141496@gmail.com

Abstract

The industrialization of agriculture has gradually taken place in Colombia, because of the need to optimize the time and effort required for the processes involved from growing fruits and vegetables to their distribution and commercialization, whether they are meant for exportation or commercialization within the country. Fruit classification is a very important process before they are being taken out as a final product, since predetermined guidelines issued by regulatory entities that define the quality of the harvested product must be followed. The authors propose a low-cost prototype for the automatic classification of bananas according to the NTC 1190 standard (Colombian normative), using a convolutional neural network (CNN) of MobileNetV2 architecture trained through transfer learning and implemented in a Raspberry Pi 3B+ with a camera to monitor the specimens and an easy interface for interaction with the user, as well as a case designed to contain the hardware and allow access to its ports in the most compact way possible. The datasets utilized in this work for training, validation, and testing, consists of images taken from two free access fruit database and others acquired by the researchers. The achieved precision is 87 %, enough to ensure reliability and low computational cost.

Keywords: Bananas, Convolutional Neural networks, Classification, Transfer learning.

Introduction

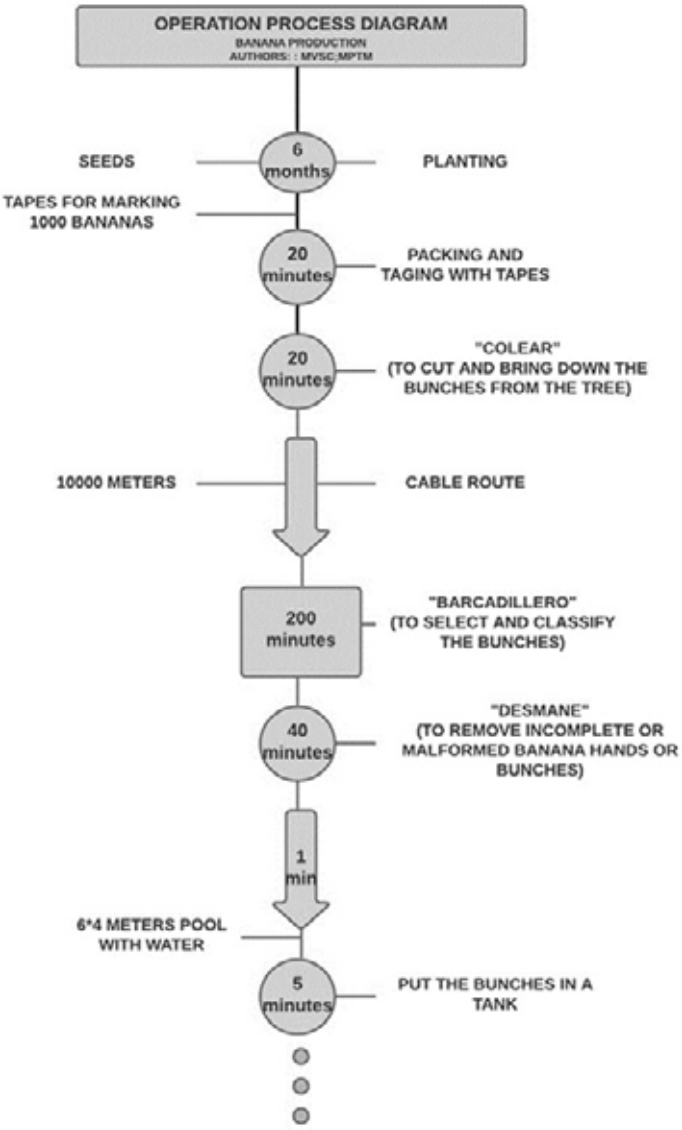
According to the Colombian Agricultural Institute (ICA), the bananas produced in Colombia are licensed to be commercialized in countries such as Germany, Saudi Arabia, Albania, Aruba, Australia, and recently Argentina (Tobón Rojas & Vanegas Cabrera, 2019), therefore, this product is of great importance within Colombian exportation.

The production has been increasing in recent years: 93.4 million boxes of 20 kilos in 2016 for a total value of 811.6 million dollars, 96.4 million boxes in the course of 2017 and 100.491.531 boxes of bananas in 2018, of which 2.472.738 were equivalent to exported boxes (Sociedad de Agricultores de Colombia, s.f.). In addition to this, banana exportation made a total of USD 852.8 million with 100.2 million boxes at the end of 2019 and has contributed with more than 140.000 jobs in total.

Although the above indicates significant numbers for the Colombian economy, several stages involved in banana production are not efficiently performed due to the lack of automation. As the raw material flows consecutively from the beginning of the process until it comes out as a product ready for distribution, each procedure is executed regardless of the time it takes.

Figure 1 shows the first stages of the operation process according to a study carried out in the banana company Banafrut (Sierra Castro & Torrenegra Murcia, 2017). Note that it is specified the execution time in each one.

Figure 1



Process diagram on banana production. Adapted from (Sierra Castro & Torrenegra Murcia, 2017).

The "barcadillero" (selection and classification of the bunches by quality) takes approximately 200 min (almost 4 hours), because the procedure depends on the empirical inspection of the workers in charge of classifying the bananas. In (Sierra Castro & Torrenegra Murcia, 2017) research recommends implementing a machine that helps the worker to recognize the patterns of the fruit that determine the classification according to the established regulations, in this way, the time

normally invested in such activity would be considerably reduced.

One of the most developed applications in recent years in the field of artificial intelligence and machine learning are the images recognition and the images classification. The purpose of this research is to develop a prototype that reduces the time of selection and classification of bananas by quality according to the NTC 1190 standard (ICONTEC, 1976), by means of a CNN implemented in a Raspberry Pi 3b+ and a camera to capture the image. The proposed network model is MobilenetV2, a lightweight architecture made to be implemented on mobile devices, while preserve the accuracy. The network was trained with the transfer learning technique to save processing time and computational cost. The proposed model achieves an accuracy of 85%, specificity of 93.3% and a F1-score of 86.3 %.

1. Data Base

The database is a composition of two opensource fruit databases and self-acquired images. The result is an augmented database with 350 images in three categories, for a total of 1050 photos. The Tier-based Dataset (Cabinatan, et al., 2020) consists of 1164 photos taken on a white background from different angles and size of 640x480. This database consists of bunches of green bananas classified into four categories (extra, class I, class II, rejected) that scale depending on the perceivable defects. The Fruits Fresh and Rotten for Classification (Kaggle.com, 2020) is a public and collective dataset from Kaggle that consists of images of apples, oranges and bananas classified in rotten and fresh with different resolutions. We discarded a group of images due they did not fit the categories in the context of Colombian banana production.

The total of images was grouped into three categories according to the features described in the NTC 1190 standard (ICONTEC, 1976). The labeling was made manually, checking the physical characteristics of each class, and then adding them to folders named as the three labels: extra, first and second. Preprocessing consisted of a normalization between 0-1 and a resolution adjustment into 224x224 pixels.

The distribution of the database is according to the following terms:

- 80% of the images for training (840 in total).
- 20% of the images for validation (210 in total).

For the test performance, 60 photos were acquired by us with the Raspberry generic camera and not included in the training set. Figures 2 3 and 4 are examples of the bananas for each class.

Figure 2

Extra class bananas.

Figure 3

First class bananas.

Figure 4

Second class bananas.

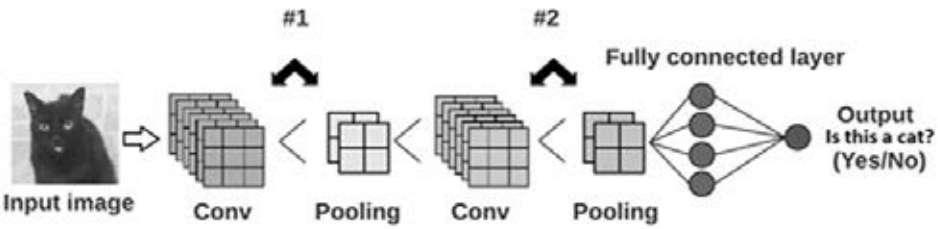
2. Convolutional Neural Networks (CNN)

Convolutional neural networks (CNN) are a class of deep neural networks which consist of multiple layers of convolutional filters that processes an input image with arrays named kernel for extracting characteristics.

Japanese scientist Kunihiko Fukushima published the first CNN model in 1980 (Fukushima, 1980). Eight years later, Yann LeCun improved that model using three types of layers: convolutional, sub-sampling and fully connected layers (Bengio, Bottou, Haffner, & LeCun, 1998). The convolutional layers are meant to extract features from the input images, sub-sampling layers reduce both spatial size and computational complexity, and the fully connected layers classify the data.

CNN have been used to solve miscellaneous problems, such as: fruits classification (Hou, Li, Sun, Wu, & Yang, 2016), object recognition (Okamoto, Tanno, & Yanai, 2016), and even medical applications (Castillo, Fajardo, & Granados, 2020). Figure 5 shows the basic structure of a convolutional neural network. Note that both modules 1 and 2 have convolutional and sub-sampling layers, while the fully connected layer appears at last.

Figure 5



Basic configuration of a convolutional neural network.

3. Transfer Learning

The Transfer learning technique takes a pre-trained model and makes certain modifications to adapt it to a new problem, "freezing" the parameters of the deep layers and training only the last dense layer. The purpose of this process is to avoid training a model from scratch, thus saving processing time and computational resources. Additionally, the researcher can estimate the results such as the weight of the network from architectures already designed for the application of interest, which is a crucial criterion in mobile or memory applications limited.

Recent works have used this technique to address medical applications like histopathological images of breast cancer tissues classification, where the results of

a network trained from scratch and a pre-trained fine-tuned network are compared, concluding that transfer learning outperforms the full training technique in most of the performed experiments (Mehra, 2018).

For this research, the CNNs were loaded with the pretrained weights obtained with ImageNet through the Keras libraries, and later were trained with our own dataset.

4. Selecting the CNN architecture

Since there is a direct relationship between the number of parameters that the CNN architecture has and the batch of images available for training, architectures like AlexNet or VGG get over-fitted easily on the small or medium size training data (Fan, Han, & Liu, 2018), therefore, models with less parameters were taken into consideration.

The selected models are MobileNetV2 and InceptionV3, which have 2.261.827 and 21.808.931 parameters, respectively.

4.1. MobileNet V2

The MobileNet architecture was designed to perform without many computational resources like mobile devices, it supports RBG channels.

This network considerably reduces the number of operations and memory used without diminishing precision. One of the biggest developments of the MobileNetV2 is its residual connection, which was created to help with the flow of gradients through the network, by adding these states, the network may access activations that were not modified in the convolutional block.

In addition to this, the technique of depthwise separable convolutions is incorporated, in order to reduce computational cost without having considerable losses in precision (Chen, G. Howard, Kalenichenko, & Zhu, 2020).

4.2. Inception V3

The InceptionV3 is an upgrade of the GoogleNet architecture (known as Inception V1) that was designed to keep computational efficiency in mind, so it can be run on individual devices even with low computational. This architecture takes 224 x 224 size images with RGB color channels. The overall architecture is 22 layers deep, and all the convolutions use Rectified Linear Units (ReLU) as their activation functions.

InceptionV3 incorporates the structure of its precursor in addition of some improvements like RMSProp optimizer to improve learning rate and restricting the oscillations on the gradient descent so it will converge faster, batch normalization

in the auxiliar classifiers to earn speed and stability and label smoothing to prevent overfitting and a bad generalization.

5. Training

5.1. General aspects

To perform the training process, we used the Colab platform which provides virtual environments with GPUs and TPUs from the Google platform. This platform allows the users to work from the cloud in an environment based on Jupyter Notebooks with Tensorflow, Keras and OpenCV libraries.

When using transfer learning, the last layer on the pre-trained models correspond to the original categories in which the model was initially trained, and the weights and neurons of this layer" learned" those classes. Therefore, this layer must be removed and replaced by a new one with the output number according to the classes of interest.

5.2. Data augmentation.

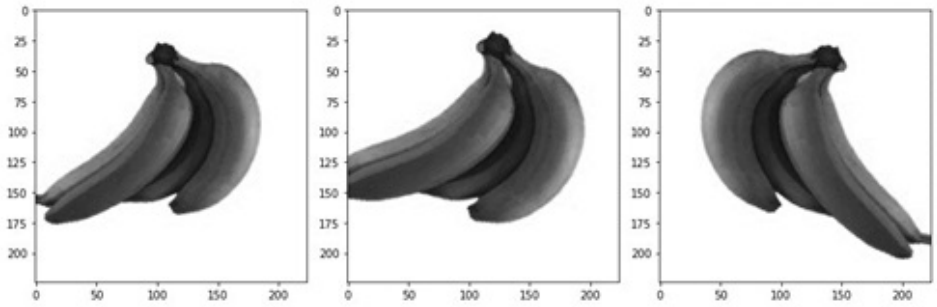
One of the limitations that can arise when addressing a problem with CNN is the amount of data available for training, since not enough information is entered for a particular problem, the network will learn global characteristics that any item belonging to a category would has and eventually could find difficult to classify new data.

Data augmentation is a technique which through transformation effects performed on the training set, we can obtain more data without adding new ones. This can be helpful to ensure an adequate set of data when the database is formed from scratch, with specific characteristics such as specific color backgrounds or capture angles to avoid irregularities.

This technique has been shown to have good results when using transfer learning, showing improvement on classification strength, and helping in preventing overfitting (Perez & Wang, 2017).

For this research, we applied to our training data rotation effects, horizontal flips and a 10% zoom to the images randomly. Figure 8 shows an example of data augmentation for an extra class bunch of bananas.

Figure 6



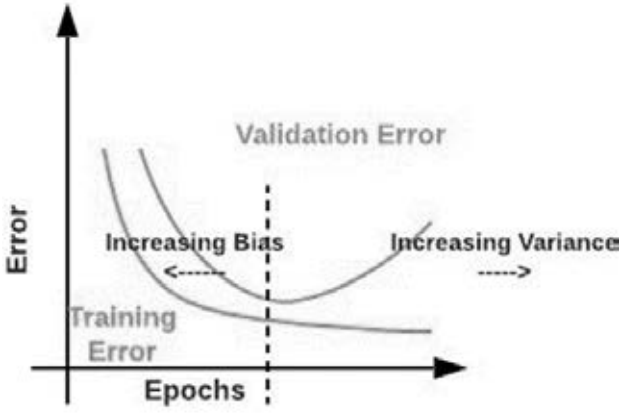
Data augmentation on an extra class image

5.3. Early stopping

Early stopping is a regularization technique used to prevent the model to overfit, which consists of stopping the training process when the model performance stops improving during the validation stage, this is also knows as learning speed slowdown.

Figure 9 is an error vs epochs graph, notice that there is an optimal point where both the training error and the validation error are minimal after a certain number of epochs elapsed during the process, and after this point, the validation error curve starts to increase (although the training error curve may keep decreasing).

Figure 7



Early Stopping. Adapted from Jain (2020)

In conclusion, what is expected with the early stopping technique is that the process stops at the optimum point described above, both to avoid overfitting and

to not keep training needlessly, which finally translates into savings processing time and computational resources (Caponnetto, Rosasco, & Yao, 2005).

5.5. Testing different batch values.

Batch size is the parameter that represents the number of samples that will be propagated through the CNN to get one update to the network parameters.

Some works have concluded that there is a direct relationship between batch size, computational cost, and network training efficiency. Hence, a smaller batch size can yield a fast computation, but it also implies fewer updates per training iteration, which will require more time to converge, and the testing accuracy will tend to fluctuate significantly.

On the other hand, the greater the parameter value, the higher the image recognition accuracy, but it also implies a huge computational cost (Radiuk, 2017).

For both architectures, different batch values will be tested and compared to obtain the most appropriate value.

6. Implementation

The prototype consists mainly of a Raspberry Pi 3B+, which offers:

- • High processing speed.
- • Easy hardware connectivity utilities, being able to monitor the activity on any screen with an HDMI connection without requiring a large working space.
- • Delivers a solid development platform for Python and Tensorflow Lite. Additionally, it has a wide documentation repositories and development community.

Figure 10 shows the interface designed. Notice that it shows the capture in real time and the probabilities of classification results. It also has a clean button so another shot can be taken, and a camera select list in case there are other cameras connected that can be recognized by the Raspberry and used for capturing images.

Figure 8

Interface

Figure 11 shows the structure for the prototype: A Raspberry case with a tab for the 5MP camera to position and hold it, space for adding a cooler on the top, access to ports and terminals for connections and screw holes to secure the device and attach it to a surface if required.

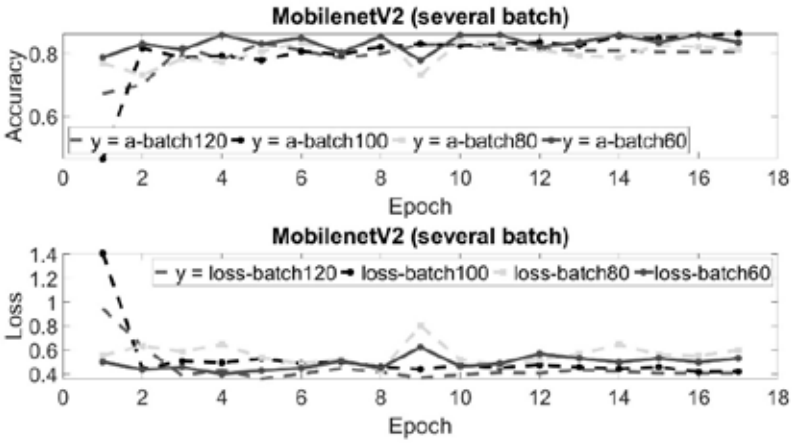
Figure 9

Prototype structure

7. Results

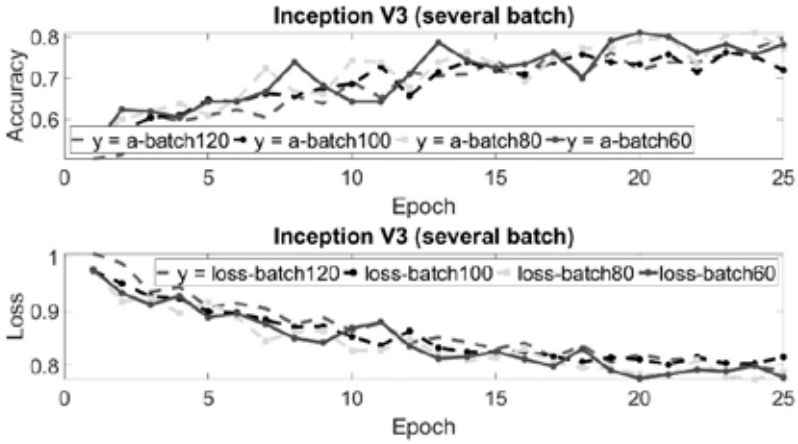
Figures 12 and 13 show the results for Accuracy vs Epochs and Loss vs Epochs during validation for the MobileNetV2 and InceptionV3 with different batch values as mentioned in the previous chapter. Notice that for values over 100 there is an improvement in the process and the most convenient to use is 100 for MobileNetV2 and 120 for InceptionV3.

Figure 10



Validation of MobileNetV2

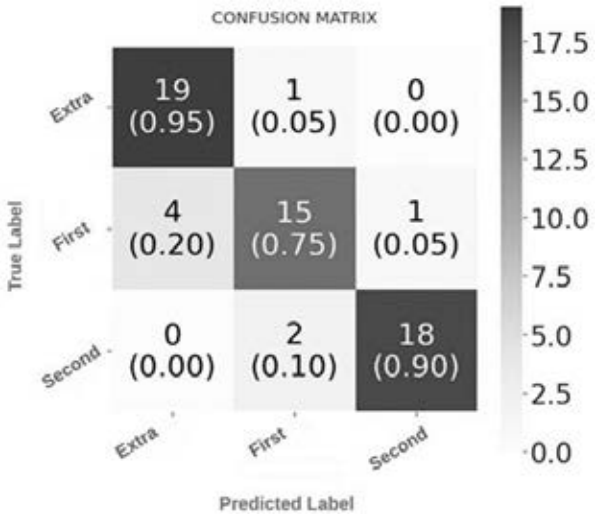
Figure 11



Validation of InceptionV3

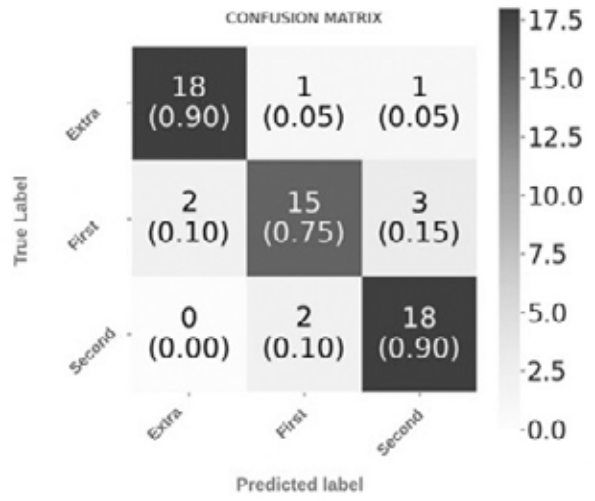
The confusion matrix is a practical way to visualize the performance of classification for a model, with a set of data which the model has never seen before and their true values are known, so we can see know which values were well predicted (matching the known labels) and which were the miss-matching predictions. Figures 14 and 15 show the confusion matrices performed with the test data for both architectures.

Figure 12



Confusion matrix for MobileNetV2

Figure 13



Confusion matrix for InceptionV3

Some specific metrics were taken into consideration from the matrix and presented in Table 3 and 4 as follows:

- Correctly matching the label in question (True Positives, TP).
- Incorrectly matching the label in question (False Negatives, FN).
- When the model predicted that it was not a label, and the prediction was correct (True Negatives, TN).
- When the model predicted a label, but the prediction was not correct (False Positives, FP).

Table 1
Metrics for MobileNetV2

Class	TP	FP	TN	FN
Extra	19	4	36	1
First	15	3	37	5
Second	18	1	39	2

Table 2
Metrics for InceptionV3

Class	TP	FP	TN	FN
Extra	18	2	38	2
First	15	3	37	5
Second	18	4	36	2

Since the purpose of the study is to classify a product according to a standard, and the less desirable events are the false positives, we also measured the specificity or TRN (true negative rate). Specificity tells how well the model can detect a class, taking into consideration when the algorithm is correct in predicting that a data does not belong to a certain class as follows:

$$Specificity = \frac{TN}{TN+FP} \quad (1)$$

So, this calculation must be done for all the three classes, as is shown in Table 5 and Table 6.

Table 3
Results for MobileNetV2

	Precision	Recall	F1-score	Specificity
Extra	0.83	0.95	0.88	0.90
First	0.88	0.70	0.78	0.925
Second	0.90	0.95	0.93	0.975
Accuracy			0.87	
Weighted Avg	0.87	0.87	0.86	0.933

Table 4
Results for InceptionV3

	Precision	Recall	F1-score	Specificity
Extra	0.90	0.90	0.90	0.95
First	0.83	0.75	0.79	0.925
Second	0.82	0.90	0.86	0.90
Accuracy			0.85	
Weighted Avg	0.85	0.85	0.85	0.925

The MobileNetV2 outperforms InceptionV3 in terms of specificity, confusion matrix results, F1-score, but also has the lowest number of FP, thus, it is proven to be the most effective model for this classification task, and the current data sample.

Table 7
Results for MobileNetV2

Test Accuracy	**0.87**	**0.76**	**0.71**
Optimizador	Adam	RMSprop	SGD
Batch	120	120	120
Learning-rate	1e-3	1e-3	1e-3
β1	0.9	0.9	
β2	0.99		

Table 8
Results for InceptionV3

Test Accuracy	0.76	0.85	0.85
Optimizador	Adam	RMSprop	SGD
Batch	120	120	120
Learning-rate	1e-3	1e-3	1e-3
β1	0.9	0.9	
β2	0.99		

After performing the testing process with the CNN MobileNetV2 and the InceptionV3 using the hyperparameters shown in Table 7 and Table 8, it can be noticed that the highest accuracy is achieved with the Adam optimizer, using a batch size of 120 and learning rate, β1and β2 with the default values on CNN MobileNetV2.

8. Conclusions

We have developed a prototype for banana classification according to the NTC 1190 using a CNN running in a Raspberry Pi3 B+. We tested several batch values for two different architectures and our results suggest that with a batch of 100 for the MobileNetV2 the most consistent accuracy is reached. On the other hand, the confusion matrix results show a good generalization for the three categories.

The result is a MobileNetV2 model with an accuracy of 87%, a F1-score of 86.3%, specificity of 93.3%, and precision of 87%, so the present work contributes to the possibility of a low-cost, rapid, and automatic classification device to help the banana company workers to perform this task efficiently in order to save time.

From the results obtained, we can conclude that transfer learning is an excellent technique that allows obtaining an adequate and sufficient model to address a problem while reducing the training iterations. For problems with a small amount of data available for the training and validation tasks as ours, using pre-trained models is a significant advantage, since by reducing processing time and having a defined architecture with evidence of a good performance, the researcher can dedicate squarely to acquire more data instead of building a model from scratch.

It should also be noted that although the MobileNet V2 is a small model compared to other architectures, it was the one that obtained the best performance; therefore, using large networks may not be equivalent to an optimal result.

The "barcadilleo" is entirely done by empirical means (Sierra Castro & Torrenegra Murcia, 2017), therefore the precision achieved is acceptable in a real context of

banana production Since we are training our model in accordance with the guidelines of the standard that define the features for the correct classification in Colombia.

9. Future research

Some of the limitations of this study can be overcome in future research, for instance, more images can be acquired to complement the current dataset, so the model can be trained with a larger set and earn a better performance. A more interesting approach for future research would be to focus on distinguishing the defects on the bananas like physical deformities (of which there is not enough data of easy and free access) included in the sample or rejected category in the NTC 1190.

Bibliographic References

Agrawal, P., Efros, A., & Minyoung, H. (2016). What makes imagenet good for transfer learning? *arXiv preprint arXiv:1608.08614*.

Bengio, Y., Bottou, L., Haffner, P., & LeCun, Y. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE, 86*(11), 2278-2324.

Cabinatan, I. P., Cascabel, H. F., Ferrer, L. V., Larada, J., Pantilgan, R., Piedad, E. J., & Pojas, G. (2020). *Tier-based dataset: Musa-acuminata banana fruit species.* Obtenido de https://data.mendeley.com/datasets/zk3tkxndjw/2

Caponnetto, A., Rosasco, L., & Yao, Y. (2005). ON EARLY STOPPING IN GRADIENT DESCENT LEARNING. *Web.mit.edu*.

Castillo, J., Fajardo, C., & Granados, Y. (2020). Patient-specific detection of atrial fibrillation in segments of ecg signals using deep neural networks. (U. M. Granada, Ed.) *Ciencia E Ingenieria Neogranadina, 30*(1), 45-58.

Chen, B., G. Howard, A., Kalenichenko, D., & Zhu, M. (2020). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *Arxiv.org*. Obtenido de https://arxiv.org/pdf/1704.04861.pdf

Colombian Institute of Technical Standards and Certification ICONTEC. (1976). *NTC 1190, banana classification.*

Fan, W., Han, D., & Liu, Q. (2018). A new image classification method using cnn transfer learning and web data augmentation. *Expert Systems with Applications, 95*, 43-56.

Fukushima, K. (Febrero de 1980). Neocognitron: A self-organizing neural network

model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics, 36*, 193-202. doi:10.1007/BF00344251

Hou, L., Li, P., Sun, Q., Wu, Q., & Yang, H. (2016). Fruit recognition based on convolution neural network. En *2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)* (págs. 18-22). Changsta, China. doi:10.1109/FSKD.2016.7603144

Jain, R. (2020). *Why "early-stopping" works as Regularization?* Obtenido de https://medium.com/@rahuljain13101999/why-early-stopping-works-as-regularization-b9f0a6c2772

Kaggle.com. (2020). *Fruits fresh and rotten for classification.* Obtenido de https://www.kaggle.com/sriramr/fruits-fresh-and-rotten-forclassification

Mehra, R. (2018). Breast cancer histology images classification: Training from scratch or transfer learning? *ICT Express, 4*(4), 247-254.

Okamoto, K., Tanno, R., & Yanai, K. (2016). "Efficient mobile implementation of a cnn-based object recognition system. En *Proceedings of the 24th ACM international conference on Multimedia* (págs. 362-366).

Perez, L., & Wang, a. (2017). The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*.

Radiuk, P. M. (2017). Impact of training set batch size on the performance of convolutional neural networks for diverse datasets. *Information Technology and Management Science, 20*(1), 20-24.

Sierra Castro, M. V., & Torrenegra Murcia, M. (2017). *Análisis de la producción del banano en Banafrut.* Universidad del Rosario.

Sociedad de Agricultores de Colombia. (s.f.). *Sector bananero colombiano creció en 2018*. Obtenido de https://sac.org.co/sector-bananero-colombiano-crecio-en-2018/

Tobón Rojas, M. C., & Vanegas Cabrera, J. M. (2019). *La industria bananera colombiana: retos y oportunidades de mejora.* Universidad del Rosario.