A coffee bean classifier system by roast quality using convolutional neural networks and computer vision implemented in an NVIDIA Jetson Nano

Conference Paper · October 2022

DOI: 10.1109/CONIITI57704.2022.9953636

Gerardo Vilcamiza
University of Buenos Aires

A coffee bean classifier system by roast quality using convolutional neural networks and computer vision implemented in an NVIDIA Jetson Nano

Gerardo Vilcamiza
Escuela de Ingeniería Mecatrónica
Universidad Peruana de Ciencias Aplicadas
Lima, Perú
gerardo.vilcamiza@ieee.org
0000-0001-6109-5515

Nicolas Trelles
Escuela de Ingeniería Mecatrónica
Universidad Peruana de Ciencias Aplicadas
Lima, Perú
u201523821@upc.edu.pe
0000-0001-9310-9774

Leonardo Vinces
Escuela de Ingeniería Mecatrónica
Universidad Peruana de Ciencias Aplicadas
Lima, Perú
leonardo.vinces@upc.pe
0000-0002-3518-591X

José Oliden
Escuela de Ingeniería Mecatrónica
Universidad Peruana de Ciencias Aplicadas
Lima, Perú
pceljoli@upc.edu.pe
0000-0003-2643-327X

***Abstract*—This article proposes the development and implementation of an intelligent system for the automatic classification of coffee beans by roast level or quality, which complements a previous project on the development of a coffee bean detector. Likewise, this work, in subsequent articles, will be part of a more complex system that will carry out the selection of beans by means of electro-pneumatic actuators. According to different study sources, currently, the coffee agro-industrial sector employs labour to discern the quality of the beans that manage to go through the selection process to the final packaged product. The expenses of the coffee growing companies are considerably affected by the investment in personnel for the selection of bean, in addition, the results of this are not always exact and uniform, since they are influenced by the subjectivity in the vision, and in the criteria or judgment, of each operator. For this reason, this work presents as a solution a classification system for coffee beans according to their level of roasting, which is highly linked to their final quality. This system was developed in Python and implemented on an NVIDIA Jetson Nano development board, where computer vision libraries such as OpenCV and artificial intelligence libraries such as Pytorch were used, the latter to design a convolutional neural network (CNN) and train it with our own dataset, obtained from real samples of coffee beans differentiated into 3 degrees of roasting (under-roasting, optimum roasting and over-roasting).**

***Keywords*— artificial intelligence, convolutional neural networks, computer vision, deep learning, object classification**

I. Introduction

In recent years, Peru has managed to position itself as one of the main references in the export of coffee worldwide, due to its high quality and the good care that is given in its cultivation [1]. However, this sector has seen its production schedule affected due to the tedious task of selecting and filtering the coffee fruits to ensure optimal quality control. This activity, in many parts of the Peruvian coffee agribusiness, is still done manually, which means excessive labour costs and long production times [2]; nevertheless, in other sectors of the industry, less orthodox options have already been introduced, such as the implementation of semi-automatic machinery of an electromechanical nature. However, this alternative has solved the problem only in the pre-roasting stage of the process, since this is where the coffee bean is filtered for shape and size. On the other hand, a smart system is needed that is capable of filtering the beans in the post-roasting stage, where the selection of the coffee is made according to the colour and degree of roasting. And since good visual perception is needed for this activity, it is still carried out by an operator who is instructed with a colour guide table, so that he can compare it with the colour of each bean.

This leads, in the first place, that the production times are delayed because, in the post-roasting selection, the beans are processed at a low speed due to human slowness. Additionally, workers require breaks and cannot work continuously 24 hours a day. Secondly, despite all the time that production took, in the final batch you can still find beans that do not correspond to their assigned roasting quality, due to the ever-present human error, since each person's view and perception is subjective, and does not result in a uniform selection, even more so if it is considered that at a given point in the process, the human eye gets tired, which causes selection failures due to operator fatigue, causing, in turn, the reduction of great profit potential economical due to possible waste of good quality coffee. And, thirdly, the loss of labour that occurs in other important areas of the plant, since the workers in charge are only dedicated to the selection of coffee.

It is also worth noting that the impact of an over-roasted or insufficiently roasted coffee bean affects considerably the flavour of the product that the customer will taste, since roasting is the most important process that the coffee bean will undergo. Within this process, different chemical changes can be caused within the bean, which is manifested in the tendency to increase the concentration of caffeine as the degree of roasting increases, and in the same way the direct influence on the chemical composition of coffee beans, on the concentration of phenolic compounds, which in turn affects their antioxidant capacity [3].

Obtaining a good classification of objects through digital image processing goes hand in hand with the implementation of artificial intelligence algorithms. This methodology is continuously used in various areas of agribusiness and the food industry. For example, Koklu, M., & Ozkan, I. A. [4] tested 4 different AI models to classify a variety of dry bean types (Barbunya, Bombay, Cali, Dermason, Horoz, Seker, and Sira). In this work, an accuracy of 91.73%, 93.13%, 87.92% and 92.52% was obtained for the multilayer perceptron (MLP), support vector machine (SVM), k nearest neighbors (kNN) and decision tree (DT) respectively. As can be seen, the

978-1-6654-6525-0/22/$31.00 ©2022 IEEE

best result obtained was with the SVM algorithm; however, it is estimated that, if the MLP had been complemented with a previous convolution process, thus forming a CNN, as will be done in the present work, a higher precision in the classification of beans would have been obtained

Fuentes, M.S. et al. [5] developed a system for recognizing the stages of maturation of various coffee fruits, which were labelled as green 1, green 2, green 3, green yellow, pinton, ripe, overripe and dry. In this system, neural networks and artificial vision libraries such as OpenCV were used, and after 42 tests, an efficiency of 97.6% was obtained. However, the recognition would not be as precise if the fruits were not so different in their colour tone, so if you wanted to differentiate closer shades, as in the present work, training with better techniques and a larger dataset should be carried out.

Pizzaia, J.P.L. et al. [6] following the line of coffee, propose using the MLP neural network for a coffee bean quality classification system, based on its affected area, roundness and colour. In this project, the Raspberry Pi development board was used, which was programmed in Python, in addition, an accuracy of 92.86% was obtained for the recognition of good beans and one of 95.35% for the recognition of defective beans. However, the algorithm is made for unroasted beans, unlike the present project. On the other hand, the precision for the good beans is not so optimal, since it does not exceed 95%, this would be solved with a CNN, as well as the training time would be reduced using a board with a powerful GPU like the one from Jetson Nano board.

Finally, it is worth mentioning the work carried out by Khattak, A. et al [7], where convolutional neural networks (CNN) were used for the development of an automatic detection system for citrus fruits. In this, different AI models were also considered, such as the SVM and the kNN model. After analysing all the models, the one that obtained the best result was the training of a CNN, achieving an accuracy of 95.65%, which was the highest. However, the algorithm has still room for improvement, since a data set of only 213 images was used.

After investigating previous research works, it was shown that CNN is the most efficient and accurate method for classification, this added to a large dataset with a large number of images for each type. Therefore, this artificial intelligence model is proposed to classify coffee beans after the roasting process in order to detect samples that do not meet the appropriate quality standard for export distribution.

II. Methodology

For the present study, the choice of two main devices is highlighted. For the acquisition of images, the Razer Kiyo Pro camera was used, which was chosen for its great image quality and its good capture speed. This is capable of taking up to 60 images per second at a resolution of 1920x1080 pixels. On the other hand, the chosen development board was the NVIDIA Jetson Nano with 4GB of RAM, due to its power for training a CNN thanks to its GPU. In this, an AI algorithm developed in Python with the Pytorch and OpenCV libraries was implemented.

Figure 1 shows all the stages to achieve the training of the convolutional neural network.

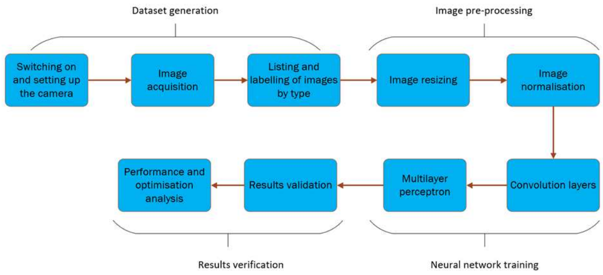

Fig. 1. Block diagram of the training process

Figure 2 shows the stages of the real-time classification process of the coffee bean samples.

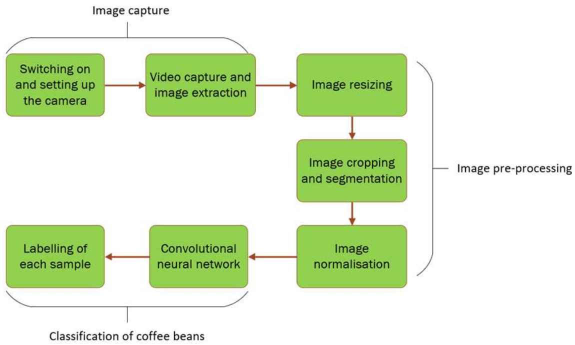

Fig. 2. Block diagram of the real-time classification process

One of the main reasons why the NVIDIA Jetson Nano was chosen as the development board to be used, was its GPU and its CUDA cores, the latter are a set of parallel processors that can perform the processing of a huge amount of data. These cores perform complex calculations much more easily than a CPU. That is why the Jetson Nano stands out against other types of boards, such as the Raspberry Pi.
Thanks to the CUDA cores, the training of the neural network, and its use for real-time classification, it will be able to be much faster. And because Pytorch can access this feature, the GPU will take full advantage of it.

III. Convolutional Neural Network Design

In this work, it was decided to choose the convolutional neuronal network (CNN) as a Deep Learning model to implement, since this is the most optimal model to use in image processing. And what is sought is to specifically classify images, for which characteristics are extracted thanks to operations with convolution kernels.

To design a CNN, before the values enter the perceptron, they must go through two main layers, the convolutional layer and the Max-Pooling layer. This pair of layers can be repeated as many times as they are considered relevant to the proper functioning of the CNN.

A. *Convolutional layer*

Convolution is a spatial operation that serves to filter and extract the characteristics of an image. This process consists that a kernel or filter performs a sweep in the original image, where for each pixel the values of each neighboring pixel are multiplied, with the parameters of the kernel, and then add them and thus obtain the pixel of the new image.

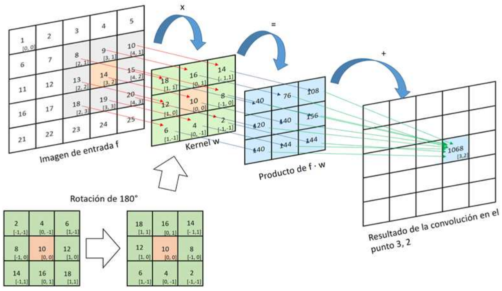

Fig. 3. Convolution of two matrices

The filtering of an input image or matrix f is represented by equation 1, where the kernel w must be inverted to scan the original image and thus obtain the final or output image I.

$$I(x,y) = \sum_{i=0}^{M}\sum_{j=0}^{N} f(i,j)w(x-i,y-j) \qquad (1)$$

Depending on the size of the kernel and its parameters, a wide variety of features can be extracted better or worse, such as edges, lighting patterns and textures, straight or diagonal axes, geometric shapes, etc. That is why, in order to obtain the greatest number of features of a class and thus be able to distinguish it from another in a better way, up to 3 convolution layers were used in the training of this CNN.

B. Max-Pooling layer

In this stage, the resolution of an image is scaled to a lower one, in such a way that, with this subsampling, only the most relevant values (the highest) of a group of pixels prevail, thus eliminating the least outstanding ones, without losing important information to the image. This is to aim that only influential values in the extraction of characteristics of a class are highlighted in the characteristics map.

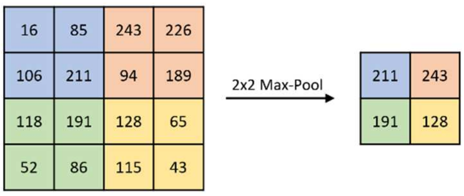

Fig. 4. Max Pooling example

In the image, a 2x2 Max-Pooling process is observed, this means that both the width and the height of the image will be divided by 2. This also helps to reduce the processing time for the following operations.

C. Dropout technique

This technique consists in that, during the training of the network, in each epoch, the neurons will have a certain probability of being deactivated. This makes the network more robust and helps to avoid that after training, some weights assigned to the network are very large, while others are very small, which means that the latter are hardly used as they have very little influence on the network decisions, resulting in a huge waste.

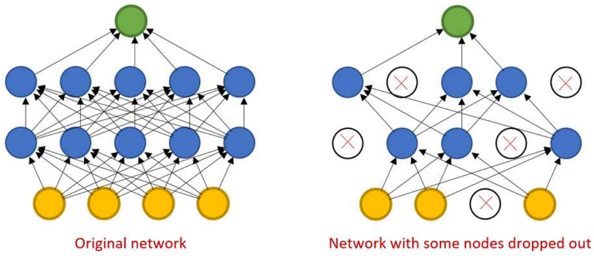

Fig. 5. Graphic representation of the dropout technique

Dropout also helps reduce overfitting, since nearby neurons often learn correlated patterns, and these relationships can form very specific patterns with the training data, this dependency between neurons will be less in the overall neural network due to dropout, so the neurons in the neural network will need to function better individually and will be less dependent on relationships with neighbouring neurons. The selection of neurons that turn off per epoch is random, with a configurable probability. For this work, a probability of 50% was chosen.

D. CNN sequence structure

Next, the sequence chosen for the design of the CNN is shown.

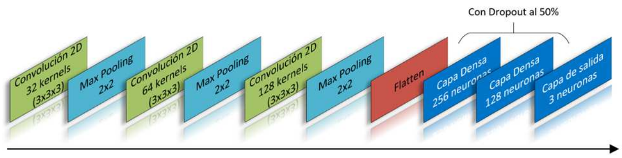

Fig. 6. Convolutional neuronal network process

Colour images in RGB colour are input to the model, as for this project the emphasis on colour is indispensable.

It starts with a convolution layer, where 64 kernels of 3x3x3 are generated. This means that each kernel is 3 pixels wide, 3 pixels high and because an RGB image is made up of 3 matrices (one for each colour channel), the kernel will have a depth of 3 matrices. It is worth noting that in each convolution layer the ReLU activation function is used, so that the system is not limited to a linear process.

$$ReLU(x) = max(0,x) \qquad (2)$$

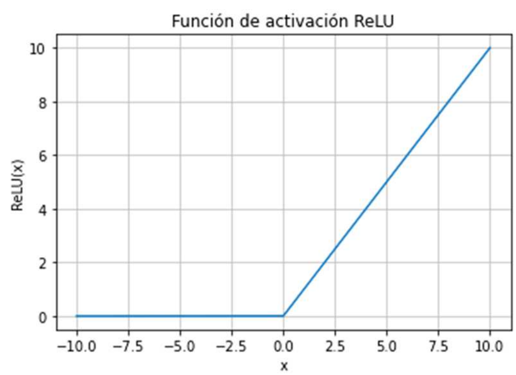

Fig. 7. ReLU activation function

Then follows the 2x2 Max Pooling layer, which means that both the width and height of the image will be halved.

These two processes are repeated a total of 3 times, doubling the number of kernels each time. Afterwards, a Flatten layer converts the matrices into vectors so that they can enter the multilayer perceptron (MLP).

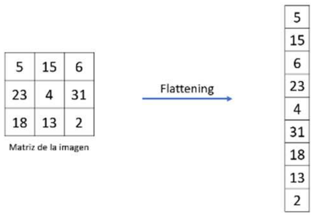

Fig. 8. Matrix Flatten process

The perceptron has two dense layers containing 256 and 128 neurons respectively. And finally, an output layer containing 3 neurons, since each neuron provides the probability that the input image is one of the 3 classes (under-roasting, optimum roasting and over-roasting). For this, the softmax activation function was used, since it allows us to know the probability of each class, and not just one, as with the sigmoid function, which allows us to make predictions with systems with more than two classes.

$$s(y_i) = \frac{e^{y_i}}{\sum_{k=1}^{K} e^{y_k}} \qquad (3)$$

Equation 3 shows the formula for the softmax function, where s is the probability that the image belongs to a certain class, y_i is the value that comes out of the previous hidden layer and K is the number of classes, and no matter how many classes there are, the sum of probabilities will always be 1. For this case K equals 3. The neural network was compiled with the Adam optimiser and the *CrossEntropyLoss* function.

E. Dataset for classification

To train the convolutional neural network, several samples of coffee beans were captured for the 3 types of roasting. These images were captured with a resolution of 1920x1080 pixels, but then scaled to a resolution of 400x400 pixels and a total of 2489 samples were obtained as shown in table I.

Likewise, the criterion for choosing the number of samples for training and validation, from each roasting level, was that the latter should present a proportion close to 20% of the total. However, the number is not exact, since the number of beans selected was not exact either, because it was decided to consider beans with small defects in shape within the set of training images, in order to check whether, even with these defects, the system was capable of making a good classification.

TABLE I. DATASET COMPOSITION

	Coffee beans samples			
	Under-roasting	***Optimum roasting***	***Over-roasting***	**TOTAL**
Training	404	840	758	2002
Validation	112	180	195	487
TOTAL	516	1020	953	2489

The number of 2489 is due to the fact that in preliminary tests of the system, it had very low accuracy, being trained on around 300 samples. It was then tested with about 1000 samples, in which case the result improved, but was still not accurate enough. It was therefore decided to increase the number of samples to more than 2000, in order to ensure a good percentage of effective classification. In addition to this, to increase the robustness of the dataset, functions of the torchvision module of Pytorch were used, which allow randomly applying to the images operations of rotation, reflex, zoom, etc. In order to have a more varied dataset before starting to train.

Figure 9 shows some of the samples used in the training, which were used to get an overview of the coffee beans. On the other hand, figure 10 shows another batch of samples, which were used to get a more specific view of the coffee beans and their roast level. Furthermore, in this batch a controlled illumination inside an image acquisition booth was used, and these sample images obtained were also used to train a coffee bean detector, which belongs to a stage prior to the coffee bean classifier by roast quality, nevertheless, the development of this detector will not be addressed in the present work.

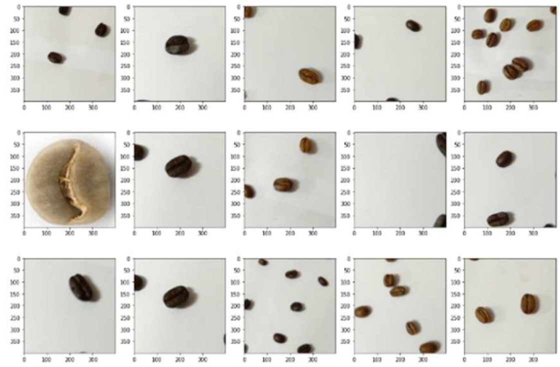

Fig. 9. Some samples of dataset I

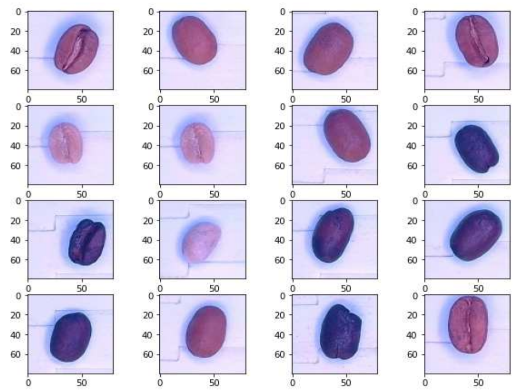

Fig. 10. Some samples of dataset II

IV. RESULTS AND VALIDATION

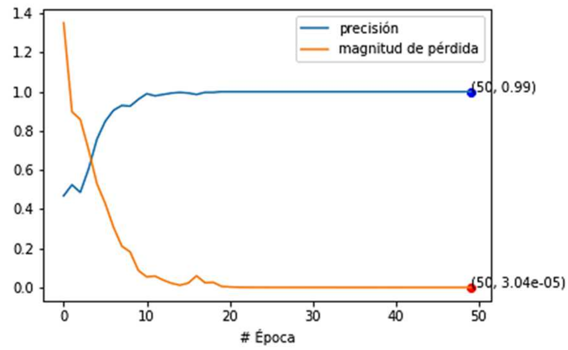

Fig. 11. Accuracy and loss graphs

After the network training, which lasted 50 epochs. Finally, a magnitude of loss of 3.04×10^{-5} was obtained, which is practically zero, and a precision of 0.99 in the classification of beans, that is, of 99%. Previously, 2 other training runs were carried out with permutations of the same dataset and the results were identical.

The accuracy of the CNN calculated by the algorithm is quite high, this remains to be verified experimentally. Therefore, tests were carried out empirically in order to validate that precision and have a more real and functional result.

Below are some images of the tests that were carried out to prove the effectiveness and accuracy of the developed system. These tests make use of a coffee bean detector algorithm that was previously developed in Pytorch and later implemented in a OpenCV small interface, to then use the final classification model, which we have covered throughout this project.

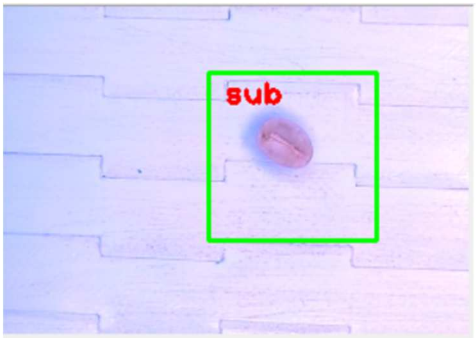

Fig. 12. Classification of a bean with under-roasting defect

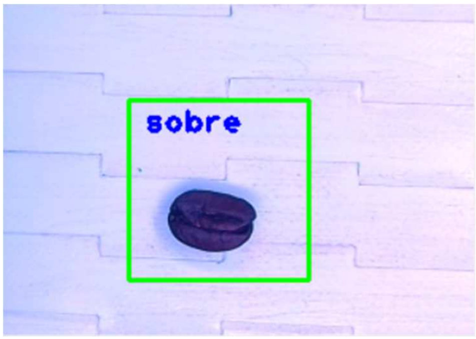

Fig. 13. Classification of a bean with over-roasting defect

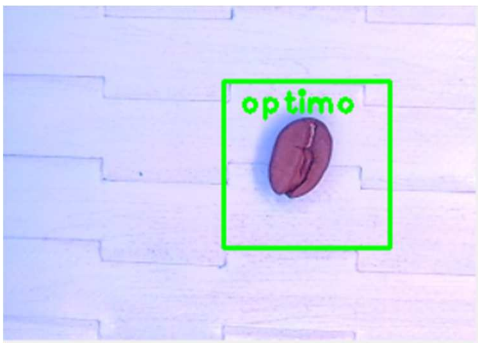

Fig. 14. Classification of a bean with optimal roasting

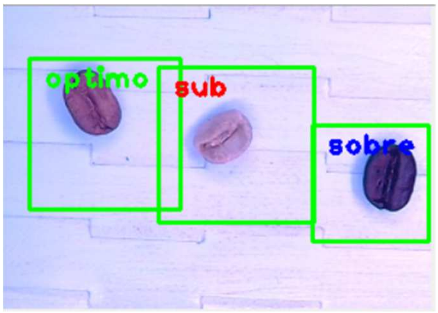

Fig. 15. Classification of the three roasting levels at the same time I

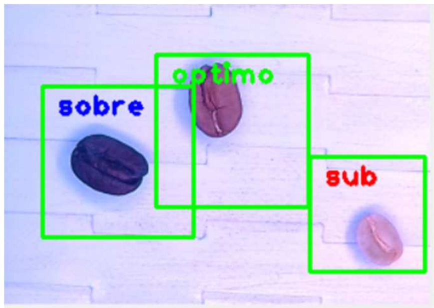

Fig. 16. Classification of the three roasting levels at the same time II

As can be seen in figures 15 and 16, 3 beans of different roasting qualities were placed at the same time, and even then the system was able to detect and correctly classify each of the beans. This fact indicates that the system is good and robust.

TABLE II. SUMMARY OF EXPERIMENTAL EVIDENCE OF BEAN CLASSIFICATION ACCURACY

	Coffee beans samples			
	Under-roasting	***Optimum roasting***	***Over-roasting***	**TOTAL**
Success	45	44	48	137
False Under-roasting	0	2	0	2
False Optimum Roasting	4	0	2	6
False Over-roasting	1	4	0	5
TOTAL	50	50	50	150

$$Accuracy = \frac{137}{150} \times 100 = 91.33\% \quad (4)$$

Equation 4 shows a more realistic value of the CNN accuracy, which is 91.33%, which is a little far from the theoretical 99% calculated by the algorithm. However, the new calculated value is still quite good and satisfactory.

V. CONCLUSIONS

A system capable of classifying images of coffee beans according to their degree of roasting was developed using convolutional neural networks, in order to automate the visual and manual inspection process for post-roasting quality control in the coffee agro-industrial sector of Peru.

It was shown that thanks to a convolutional neural network (CNN) and the application of techniques such as dropout and transformations to increase the dataset, an accuracy percentage of 99% theoretical can be obtained, which means a fairly high accuracy. However, the value closest to reality is obtained with experimental tests, and the designed CNN provided us with an accuracy of 91.33% experimentally, in addition, the calculated loss function was very close to zero, so it can be said that the implementation of this algorithm is a fairly efficient, precise and viable alternative to be used in conjunction with sorting machines in the coffee industry.

Performance indicators already mentioned (accuracy and loss), were sufficient to obtain a clear idea of the behaviour of the classifier, and this is demonstrated by observing that the experimental accuracy value is not too far away from the theoretical one. While it is true that more indicators can be evaluated, for this work, which is not extremely complex, it was not found essential or relevant to do so.

Although it is true that very good results were obtained with the CNN developed, there are other architectures that were evaluated for this work, such as the Residual Neural Network (ResNet), which allows us to obtain a good result using fewer convolution layers. However, in this case, the difference in optimisation and accuracy would not be too great, which is why we did not resort to using another type of architecture.

Using an NVIDIA Jetson Nano development board for this work made training easier and greatly reduced processing time, thanks to its GPU and CUDA cores. So, using this board is one of the best alternatives on the market for AI and computer vision projects. By far outperforming its direct competitors such as the Raspberry Pi in terms of speed.

VI. ACKNOWLEDGEMENTS

To the Dirección de Investigación de la Universidad Peruana de Ciencias Aplicadas for the support given to carry out this research work UPC-EXPOST-2022-1, and to the IEEE UPC Student Branch for making the acquisition of the NVIDIA Jetson Nano development board possible.

VII. REFERENCES

[1] D. Choque, “Evolución de la Exportación de Café: 2010-2020” B.S. Thesis, Universidad de Lima, Lima, Perú, 2021

[2] D. Llano, “Análisis de los Costos Logísticos de la Cadena de Valor del Café en Chanchamayo” B.S. Thesis, Universidad del Pacífico, Lima, Perú, 2021

[3] F. Pacheco-Coello, R. Torres, T. Arvelo, y I. Velasquez, “Variación de la actividad antioxidante por efecto del tostado en granos de café (Coffea arabica), estado Miranda, Venezuela”, cac, vol. 3, n.º 2, pp. 49–56, December 2020

[4] Koklu, M., & Ozkan, I. A. “Multiclass classification of dry beans using computer vision and machine learning techniques”. *Computers and Electronics in Agriculture*, 174. July 2020

[5] M. S. Fuentes, N. A. L. Zelaya and J. L. O. Avila, "Coffee Fruit Recognition Using Artificial Vision and neural NETWORKS," *2020 5th International Conference on Control and Robotics Engineering (ICCRE)*, 2020, pp. 224-228

[6] J. P. L. Pizzaia, I. R. Salcides, G. M. d. Almeida, R. Contarato and R. d. Almeida, "Arabica coffee samples classification using a Multilayer Perceptron neural network," *2018 13th IEEE International Conference on Industry Applications (INDUSCON)*, 2018, pp. 80-84

[7] A. Khattak et al., "Automatic Detection of Citrus Fruit and Leaves Diseases Using Deep Neural Network Model," in IEEE Access, vol. 9, pp. 112942-112954, 2021

[8] Suprapto, & Riyanto, E. “Grape drying process using machine vision based on multilayer perceptron networks”. *Indonesian Journal of Science and Technology*, 5(3), 382–394, December 2020

[9] S.-J. Chang and C.-Y. Huang, “Deep Learning Model for the Inspection of Coffee Bean Defects,” Applied Sciences, vol. 11, no. 17, p. 8226, September 2021

[10] N. -F. Huang, D. -L. Chou and C. -A. Lee, "Real-Time Classification of Green Coffee Beans by Using a Convolutional Neural Network," *2019 3rd International Conference on Imaging, Signal Processing and Communication (ICISPC)*, 2019, pp. 107-111

[11] R. S. Latha et al., "Automatic Fruit Detection System using Multilayer Deep Convolution Neural Network," *2021 International Conference on Computer Communication and Informatics (ICCCI)*, 2021, pp. 1-5

[12] T. Abu-Jamie, S. Abu-Naser, M. Alkahlout y M. Aish, “Six Fruits Classification Using Deep Learning,” *International Journal of Academic Information Systems Research (IJAISR)*, vol. 6, pp. 1-8, 2022.

[13] J. -Y. Lee and Y. -S. Jeong, "Prediction of Defect Coffee Beans Using CNN," *2022 IEEE International Conference on Big Data and Smart Computing (BigComp)*, 2022, pp. 202-205

[14] P. -Y. Yang, S. -Y. Jhong and C. -H. Hsia, "Green Coffee Beans Classification Using Attention-Based Features and Knowledge Transfer," *2021 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW)*, 2021, pp. 1-2

[15] M. Hakim, T. Djatna and I. Yuliasih, "Deep Learning for Roasting Coffee Bean Quality Assessment Using Computer Vision in Mobile Environment," *2020 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, 2020, pp. 363-370