XLVI CILAMCE
VITÓRIA - ES - BRAZIL - 2025
XLVI IBERO-LATIN AMERICAN CONGRESS ON COMPUTATIONAL METHODS IN ENGINEERING

A Proposal for Quality Classification of Cocoa Beans Using Convolutional Neural Networks

Brenda B. Pinheiro[1], Luiz A. Pinto[1]

[1]*Graduate Program in Control and Automation Engineering (ProPECAut)*
Instituto Federal do Espírito Santo (IFES), Campus Serra
Av. Dos Sabiás, 330, 29166-630, Espírito Santo, Brazil
brendabayerlp@gmail.com, pintoluizalberto@gmail.com

Abstract. Given the need for technological advancement in the cocoa sector and the standardization of physical quality analysis of cocoa beans, this work proposes an automated classification tool based on convolutional neural networks, using digital images. A database with five classes of beans was created, evaluating accuracy and performance through precision, recall, and F1-score. Three models were implemented: a CNN with MobileNetV2, another with ResNet101V2, and one without pretrained models, also applied to a second dataset for comparison. For image segmentation, Otsu's method was used, facilitating bean identification via thresholding and enabling real-time classification. The main network achieved 67.14% accuracy, with F1-scores of 0.81 for White and Defective classes, 0.60 for Brown, 0.49 for Partially Brown, and 0.78 for Violet, demonstrating feasibility for application on mobile devices.

Keywords: cocoa bean classification, convolutional neural networks, image segmentation.

1 Introduction

The cocoa tree (Theobroma cacao L.), native to the tropical forests of South and Central America [1], holds high economic importance in tropical regions. Currently, Ivory Coast and Ghana account for about 50% of global production, with forecasts for 2024/25 crop growth driven by improved climatic conditions. In the Americas, Ecuador and Brazil stand out, with Brazil responsible for 4.5% of global production [2]. In Brazil, according to IBGE, production reached 293,474 tons in June 2025, with Pará and Bahia states concentrating over 90% of this volume [3].

The quality of the beans is decisive for market value, directly influencing product competitiveness and producer remuneration. Despite an average annual growth of 3.5% between 2018 and 2025, the production chain suffered significant impacts from global events such as the COVID-19 pandemic and the Russia-Ukraine conflict, which affected logistics, input supply, and international demand [4]. Moreover, local factors such as climate, pests, and management continue to influence productivity, prompting producers to invest in modernization [5].

Quality assessment of the beans, traditionally performed by visual inspection in the cut test, presents operational and subjective limitations. In this context, the use of automated methods based on computer vision and machine learning, especially convolutional neural networks (CNNs), stands out for their ability to interpret complex visual patterns [6].

This study proposes using CNNs to automate visual analysis of cut beans, according to the Cocoa Bean Technical Regulation [7], focusing on the productive reality of the state of Espírito Santo, Brazil. The proposal aims to offer an accessible technological solution with potential to add value to the fine cocoa chain, which can be valued in the market up to three times higher than common cocoa [8].

2 Theoretical Framework

2.1 Convolutional Neural Networks: transfer learning and architectures

Convolutional Neural Networks (CNNs) are an advanced class of deep learning models inspired by the structure of the human visual cortex, widely applied in visual data analysis. Unlike traditional neural networks, CNNs leverage the strong local spatial correlation present in natural images through their three-dimensional architecture (width, height, and depth), ensuring local connectivity and shared weights among neurons, which promotes efficiency and robustness in visual pattern recognition [9, 10]. Key components of this architecture include convolutional layers, which automatically learn filters to identify hierarchical features — from simple edges to complex structures and objects; pooling layers, which reduce the dimensionality of feature maps and provide local translation invariance; and fully connected layers, which capture global patterns for classification.

This ability to extract and represent visual features with increasing complexity makes CNNs particularly effective in computer vision applications, including tasks in the agricultural sector such as grain classification, plant disease detection, and fruit segmentation — where automation facilitates monitoring and supports precise decision-making [11].

The transfer learning technique has been crucial in popularizing and expanding the use of CNNs, especially in contexts with limited labeled data, which is common in agriculture. By reusing the weights of models previously trained on large datasets such as ImageNet, transfer learning speeds up training time, greatly reduces computational costs, and enhances the model's ability to generalize to new, related tasks. This approach has proven advantageous compared to training from scratch, as it requires fewer data and helps avoid issues like overfitting.

Among the most widely adopted architectures, MobileNetV2 stands out for being designed for devices with hardware and computational constraints. It uses depthwise separable convolutions to reduce the number of parameters and operations, as well as inverted residual blocks and linear bottlenecks that preserve information and speed up processing, enabling real-time applications on smartphones and embedded systems [12, 13].

On the other hand, ResNet101V2 is a deep architecture with 101 layers that incorporates residual connections (skip connections) to mitigate gradient-related issues in very deep networks, facilitating training and achieving high accuracy in complex visual classification tasks [14, 15]. The V2 version of ResNet improves data preprocessing and normalization strategies, further enhancing performance for different applications, including those in agriculture.

While the use of pre-trained networks is an efficient solution for scenarios with operational constraints and limited data, training CNNs from scratch offers greater flexibility to customize architectures to the unique characteristics of a problem. However, this approach demands a considerable amount of data and more robust computational resources to avoid overfitting and ensure convergence during training — which can be a challenge in agricultural environments with limited labeled images [16].

Therefore, selecting the most appropriate architecture and training strategy should take into account task complexity, data availability, and computational capacity, balancing efficiency and performance to maximize the benefits of CNNs in agricultural contexts and other computer vision domains.

2.2 Image Segmentation and Otsu's Method

Image segmentation is a crucial step in computer vision systems, enabling separation of objects of interest from the background, facilitating subsequent analysis and processing. Among methods, Otsu's automatic thresholding, proposed by Nobuyuki Otsu in 1979, is widely recognized for simplicity and computational efficiency. It determines a global threshold dividing gray levels into object (foreground) and background classes by maximizing inter-class variance or equivalently minimizing intra-class variance, assuming a bimodal intensity distribution [17]. Due to its robustness with good contrast images, Otsu's method is broadly used in pattern recognition preprocessing stages across domains like food classification, industrial parts, and plant tissues.

2.3 State of the Art Review

Studies show convolutional neural networks (CNNs) and other machine learning techniques effectively classify cocoa beans. Malcher and Guedes [18] used deep CNNs on Cut Test images (1,400 balanced images in 14 classes), with Inception, VGG-16, and EfficientNetB0 architectures reaching up to 68.28% accuracy, surpassed by a network ensemble at 89.79%, showing practical potential. Fikri et al. [19] applied the YOLO (You Only Look Once) architecture for automatic detection and classification of cocoa beans, considering visual features such as color and shape. Using bounding boxes and an integrated CNN classifier, the method achieved 99.8% accuracy

on a dataset of 2,880 labeled images, demonstrating its effectiveness in automating moisture control and its potential for application to other quality variables.

Hybrid approaches combining hardware and supervised algorithms include Santos and Remot [20] who integrated the a cocoa bean fermentation device using an Arduino microcontroller with a KNN classifier, achieving 85.3% accuracy in distinguishing "Good" and "Poor Quality" beans, with high functional performance per ISO 9126. Similarly, Pino et al. [21] used CNNs to classify fermentation degree with 82% accuracy, noting confusion between visually similar classes, highlighting the need for robust datasets.

Other promising techniques involve spectroscopy combined with machine learning and genetic algorithms, with models like logistic regression and random forest surpassing 84% accuracy, presenting spectral analysis as an efficient alternative to manual screening [22]. Oliveira et al. [23] developed a model using Otsu segmentation, HSV feature extraction, and Random Forest classification on 1,800 images across four classes, reaching 97.6% accuracy, reinforcing the viability of computer vision techniques in bean quality assessment.

3 Methodology

3.1 Image Database Creation and Class Definition

Bean classes were defined following MAPA Instruction No. 38/2008. A lighting box (40×20×30 cm) with matte black interior and green base was constructed to provide contrast. Two white LED lamps (6500K) in DC eliminated shadows and aliasing. Images were captured with a 48 MP smartphone camera positioned vertically over a top opening. Figure 1 shows the box and bean visibility comparison on white and green backgrounds.



Figure 1. Box used for image acquisition and comparison of the same bean on different backgrounds.

Beans, from northern Espírito Santo, underwent primary processing. After longitudinal cutting, beans were visually inspected on an IPS calibrated monitor (RGB coverage > 99%) and classified into five categories: White (82 images), Brown (100), Partially Brown (100), Violet (100), and Defective (99, including germinated, molded, and damaged due to low individual sampling). Image segmentation cropped beans to highlight edges and visual features, composing the Segmented Bean Database (SBD) with 481 records, divided into 336 for training and 145 for validation. Additionally, a Comparison Database (CD) from background-free images per Santos, Palmeira, and Jesus [24] was used and comprises: Compartmentalized Brown, Compartmentalized White, Compartmentalized Partially Purple, Compartmentalized Purple, Compartmentalized Slatty, Plated Brown, Plated White, Plated Partially Purple, Plated Purple, Plated Slatty, Moldered, Flattened, Brittle, and Agglutinated. Figure 2 presents sample class images from SBD.

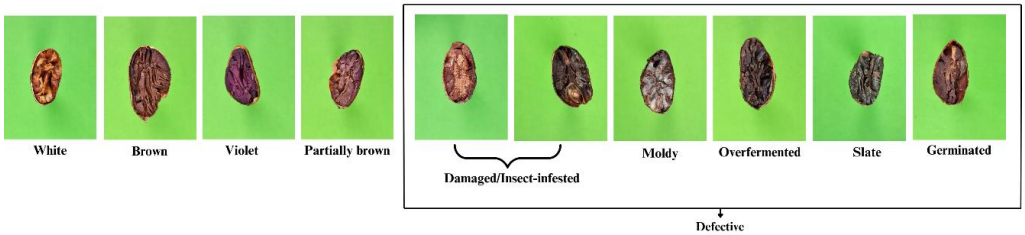


Figure 2. Classes of beans that comprise the Segmented Bean Database (SBD).

3.2 CNN Architecture and Training

Three CNN models were developed: one using the pretrained MobileNetV2 architecture [12], one with ResNet101V2 [14, 15], and a network built from scratch, without transfer learning. The pretrained models were used as feature extractors based on the ImageNet dataset [25], applying the transfer learning technique to optimize performance [26].

MobileNetV2 was prioritized due to its lightweight nature, suitability for mobile devices, and better performance compared to previous architectures such as AlexNet, VGGNet, and GoogLeNet [27, 28, 29]. Its structure is composed of residual blocks with different strides and convolutions with 1x1 and 3x3 filters, utilizing the ReLU6 activation function. ResNet101V2 contains 101 layers with deep residual connections.

In the model without pretraining, the architecture starts with a 2D convolution (16 filters of 3x3), followed by batch normalization [30], ReLU, max pooling [31], and dropout [32]. After flattening, two dense layers with 32 neurons were used, with normalization, ReLU, and dropout. The last dense layer applies the Softmax function for multiclass classification, with 5 neurons for SBD and 14 for CD [9].

Training considered data augmentation techniques [27], including random rotations, translations, shearing, and zooms, in addition to pixel normalization (scale 0–1). The following parameters were set: batch size of 28, 100 epochs, cross-entropy loss function, RMSProp optimizer [33], and accuracy as the evaluation metric. Figure 3 illustrates the general network architecture focusing on MobileNetV2.

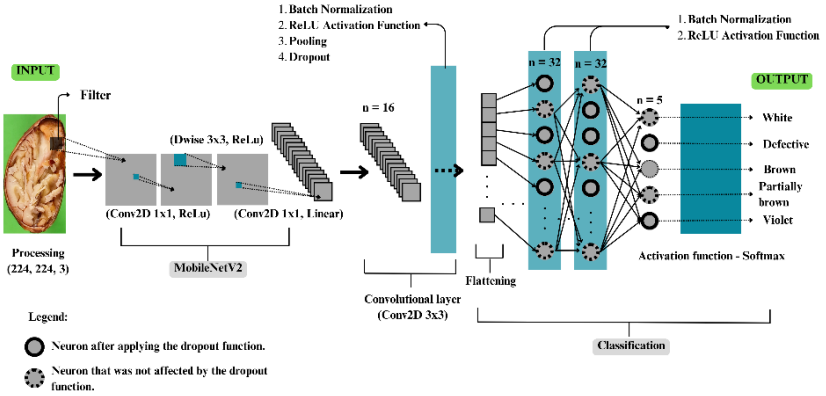


Figure 3. Detailed architecture of the developed convolutional neural network.

3.3 Performance Evaluation Metrics

The performance of the classification models was evaluated using metrics commonly applied to multiclass problems, supported by the Scikit-learn and TensorFlow libraries. Accuracy was initially considered, representing the proportion of correct predictions, but due to dataset imbalance, additional analyses were required. To provide a more detailed view, the confusion matrix was employed, relating actual and predicted classes and visually represented as a blue heatmap to highlight class-wise accuracy and misclassifications.

Complementary metrics—precision, recall, and F1-score—were also used to better assess performance under class imbalance. These are derived from the key elements true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). Following Muller and Guido [34], precision corresponds to TP / (TP + FP), reflecting the accuracy of positive predictions; recall to TP / (TP + FN), indicating the model's ability to identify positive cases; and F1-score, the harmonic mean of both, provides a balanced measure between precision and recall. Together, these metrics enable a more comprehensive and reliable evaluation of the classifier, which is especially important for complex and imbalanced tasks such as cocoa bean quality classification.

3.4 Development Environment and Real-Time Network Performance Test

The development of the code was carried out in the Google Colab and Spyder environments, both compatible with Python and widely used in data science projects. Colab enabled cloud execution with access to GPUs and integration with Google Drive and GitHub, while Spyder was used locally for stages that required greater control over the environment. The main libraries included NumPy, Matplotlib, Scikit-learn, TensorFlow, and Keras.

To validate the model in a practical application, a real-time test was performed with simultaneous detection and classification of multiple cocoa beans. Live video capture was done using the IP Webcam app, transforming a mobile phone into an IP câmera, which is a camera that transmits images over the network and can be accessed remotely. Bean segmentation utilized Otsu thresholding [17] after applying a Gaussian filter, assisted by an MDF board for contrast. The segmented beans were automatically classified by the network, with the labels overlaid on the video. The image backgrounds were adjusted to greenish tones, as in training. The results demonstrated the robustness and potential of the convolutional network for automated visual inspections in the agricultural sector.

4 Results and Discussion

In the Segmented Bean Database (SBD), composed of five classes with unequal distribution, a 60% threshold was adopted for precision and recall metrics. The CNN without pretraining achieved the highest validation accuracy (72.86%), followed by MobileNetV2 (67.14%) and ResNet101V2 (66.43%). However, the non-pretrained network showed signs of overfitting, evidenced by the discrepancy between validation and training errors, while MobileNetV2 stood out for its greater stability during the learning process.

In terms of precision and recall, MobileNetV2 showed more consistent performance, outperforming the others in four classes per metric, with an average F1 of 0.698. These values are presented as vectors in the format [Precision; Recall; F1], with **White** [1.00; 0.68; 0.81], **Defective** [0.92; 0.73; 0.81], and **Violet** [0.70; 0.87; 0.78], while showing lower performance in **Partially Brown** [0.40; 0.63; 0.49]. ResNet101V2 stood out in three classes, notably **Defective** [0.96; 0.77; 0.85] and **White** [0.80; 0.80; 0.80], but had low results in **Brown** [0.44; 0.53; 0.48] and **Partially Brown** [0.56; 0.50; 0.53]. The non-pretrained network achieved the highest average recall (74.6%), with good results in **White** [0.89; 0.96; 0.92], **Violet** [0.54; 0.97; 0.69], and **Brown** [0.90; 0.60; 0.72], but greater variability, with a drop in **Partially Brown** [0.81; 0.43; 0.57], indicating possible overfitting.

The analysis of the confusion matrices, which can be viewed in Figure 4, revealed that MobileNetV2 correctly classified most samples in the **Violet**, **White**, and **Defective** classes, with misclassifications mainly occurring between the **Brown**, **Partially Brown**, and **Violet** classes due to visual similarity. ResNet101V2 achieved better performance in the **White**, **Defective**, and **Violet** classes, showing similar confusions to those of MobileNetV2. The non-pretrained network showed poorer results in the **Partially Brown** class, corroborating the overfitting effect. Considering false positives and false negatives, MobileNetV2 stood out as the most balanced network for this dataset.

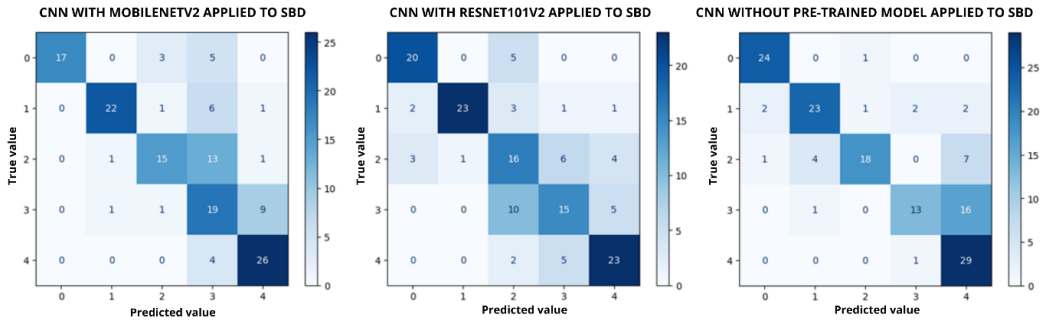


Figure 4. Confusion matrix for each of the three developed networks applied to the SBD.

In the Comparison Database (CD), composed of 14 classes with 70 images for training and 30 for validation in each class, ResNet101V2 achieved the highest validation accuracy (55%), followed by MobileNetV2 (48.1%) and the non-pretrained network (42.62%). Signs of overfitting were detected due to the higher validation accuracy compared to the test accuracy, while MobileNetV2 presented greater computational stability.

Regarding precision, recall, and F1-score, ResNet101V2 and MobileNetV2 achieved precision above 60% in five classes, with ResNet101V2 standing out for having the highest average recall (54.85% versus 48.0% for MobileNetV2) and the highest average F1 (0.53 versus 0.43). In MobileNetV2, the best-performing classes were **Agglutinated** [0.73; 1.00; 0.85], **Broken** [0.96; 0.83; 0.89], and **Slate Plate** [0.61; 0.93; 0.74], whereas **Rotten** and **Brown Plate** 0.00 in all metrics. In ResNet101V2, the top classes were **Agglutinated** [1.00; 0.93; 0.97], **Broken** [0.96; 0.87; 0.91], and **Slate Plate** [0.84; 0.70; 0.76], but there were drops in **Brown Plate** [0.27; 0.10; 0.15] and **Partially Purple Plate** [0.29; 0.13; 0.18]. The non-pretrained network showed limited performance, with several classes scoring zero, except for **Agglutinated** [0.43; 0.77; 0.55], **Rotten** [0.40; 0.07; 0.11], **Purple Plate** [0.30; 0.47; 0.37], and **Slate Plate** [0.11; 1.00; 0.20].

In the CD confusion matrices, which can be seen in Figure 4, MobileNetV2 perfectly classified the **Agglutinated** class and performed well in **Slate Plate**, but showed failures in classes such as **Rotten** and **Brown Plate**, mainly due to false positives. ResNet101V2 demonstrated the best balance, with high hit rates in key classes and lower incidence of errors, although five classes registered less than 50% accuracy. Finally, the non-pretrained network showed low robustness, with a tendency to misclassify several classes as **Slate Plate**.

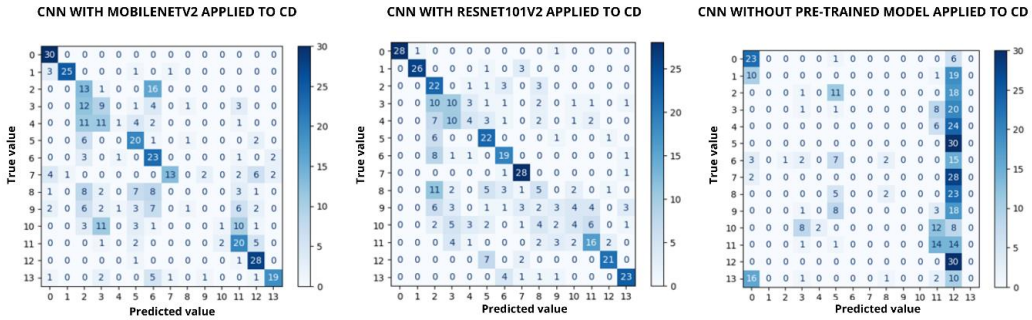


Figure 5. Confusion matrix for each of the three developed networks applied to the CD.

The results highlight the benefits of using pretrained models in CNNs, especially considering the F1 metric, which is suitable for imbalanced datasets. The best performance was observed in the SBD, with a smaller number of classes. MobileNetV2 stood out for its stability during training, with reduced variation between accuracy and cross-entropy. ResNet101V2, although more robust, did not present computational gains proportional to its complexity. The findings reinforce the importance of architectural adaptation of the networks to the dataset used, especially for non-pretrained networks, which showed inferior performance and greater susceptibility to overfitting.

5 Conclusions

The present study aimed to develop and evaluate CNNs for the automatic classification of cocoa beans from digital images, using two datasets. The objective was fully achieved, with the best scenario—using the pretrained MobileNetV2 model—achieving a validation accuracy of 67.14%, precision of 66.89%, recall of 66.52%, and F1-score of 66.70%. Comparative analysis indicated that while ResNet101V2 achieved competitive results, its high computational cost limits its practicality for low-cost or resource-constrained applications. The non-pretrained CNN performed worse, mainly due to overfitting, highlighting the need for refined architectural adjustments for efficient deployment.

In addition, the results of this study show competitive performance in cocoa bean classification. While previous models, such as the ensemble network by Malcher and Guedes [18] and YOLO by Fikri et al. [19], achieved up to 89.79% and 99.8% accuracy respectively, this work demonstrated notable results given the complexity and imbalance of the BAS and BC datasets. MobileNetV2 stood out for its stability and balance between precision and recall (average F1 = 0.698 on BAS). Unlike previous studies using balanced datasets and binary classes, this study addressed multiple visual categories with significant variation, reflecting real conditions. Confusion matrix analysis highlighted the CNNs' ability to distinguish subtle visual differences, such as Brown, Partially Brown, and Violet, supporting the practical use.

The results reinforce the robustness and applicability of CNNs in the automatic classification of cocoa bean quality, demonstrating that computer vision-based solutions can be integrated into support systems for the cocoa industry, providing greater reliability and agility in the evaluation process. This is particularly relevant when compared to the traditional method, which relies on human visual inspection and the individual analysis of each bean, making the process more time-consuming and prone to errors.

As a prospect for future work, it is proposed to expand and validate the SBD in collaboration with cocoa bean quality experts, as well as explore new models and architectures. Among the planned steps, the implementation of YOLOv12 [35], an advanced pretrained model, stands out, aiming to improve detection accuracy and speed by leveraging its real-time processing capability—an essential feature for mobile applications. In this way, it is expected to contribute to the advancement of more precise, efficient, and accessible technological solutions capable of optimizing quality analysis and control in the cocoa production chain.

Acknowledgements. The authors would like to express their sincere thanks to the Universidade Aberta Capixaba (UnAC), a program of the Government of the State of Espírito Santo, linked to the Secretaria de Estado da Ciência, Tecnologia, Inovação e Educação Profissional (Secti), for the institutional and financial support (T.O. 1064/2023) that enabled the realization of this study. We also thank the Federal Institute of Espírito Santo (Ifes) for the infrastructure provided.

References

[1] R. Mororó. "Utilization of cocoa derivatives, by-products, and residues." CEPLAC, P.O. Box, vol. 7, pp. 456–970, 2012.
[2] M. Vidal. "COCOA:." vol. 10 no. 389, June 2025. *ETENE Sectorial Notebook*, vol. 10, 2025.
[3] Brazilian Institute of Geography and Statistics – IBGE. "Systematic Survey of Agricultural Production – June 2025." Rio de Janeiro: IBGE, 2025. Available at: https://sidra.ibge.gov.br/pesquisa/pam/tabelas. Accessed on: Aug 6, 2025.
[4] J. Kongor; M. Owusu; C. Oduro-Yeboah. "Cocoa production in the 2020s: Challenges and solutions." *CABI Agriculture and Bioscience*, vol. 5, no. 1, p. 102, 2024.
[5] J. De Souza Júnior. "Cocoa: cultivation, research, and innovation." SciELO-Editus-UESC Publisher, 2018.
[6] G. Mageto et al. "Applications of artificial intelligence in the era of Agriculture 4.0." *Agronomic Science Journal*, vol. 51, p. e20207701, 2021.
[7] BRAZIL Ministry of Agriculture, Livestock and Supply. "Normative Instruction No. 38, of June 23, 2008." Approves the Technical Regulation of Cocoa Beans. *Official Gazette of the Union: section 1*, Brasília, DF, year 145, no. 120, p. 32, June 24, 2008.
[8] F. Santos et al. "Supervised methods applied to the construction of a vision system for classifying cocoa beans in the cut test." *In: National Meeting of Artificial and Computational Intelligence (ENIAC)*. SBC, 2019. pp. 72-83.
[9] I. Goodfellow; Y. Bengio; A. Courville. "6.2. 2.3 softmax units for multinoulli output distributions." Deep learning, MiT Press Cambridge, MA, USA, vol. 180, 2016.
[10] P. Sermanet; S. Chintala; Y. Lecun. "Convolutional neural networks applied to house numbers digit classification." *In: IEEE. Proceedings of the 21st international conference on pattern recognition (ICPR2012).* [S.l.], 2012. pp. 3288–3291.
[11] S. Pan; Q. Yang. "A survey on transfer learning." *IEEE Transactions on knowledge and data engineering.* vol. 22, no. 10, pp. 1345-1359, 2009.
[12] M. Sandler et al. "Mobilenetv2: Inverted residuals and linear bottlenecks." *In: Proceedings of the IEEE conference on computer vision and pattern recognition.* [S.l.: s.n.], 2018. pp. 4510–4520.
[13] A. G. Howard et al. "Mobilenets: Efficient convolutional neural networks for mobile vision applications." arXiv preprint arXiv:1704.04861, 2017.
[14] K. He et al. "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification." *In: Proceedings of the IEEE international conference on computer vision.* [S.l.: s.n.], 2015. pp. 1026–1034.
[15] K. He et al. "Deep residual learning for image recognition." *In: Proceedings of the IEEE conference on computer vision and pattern recognition.* [S.l.: s.n.], 2016. pp. 770–778.
[16] Y. Lecun et al. "Lenet-5, convolutional neural networks." *URL: http://yann. lecun. com/exdb/lenet*, vol. 20, no. 5, p. 14, 2015.
[17] N. A. Otsu. "A threshold selection method from gray-level histograms." *IEEE transactions on systems, man, and cybernetics*, IEEE, vol. 9, no. 1, pp. 62–66, 1979.
[18] D. Malcher; E. Guedes. "Intelligent classification of the cut test of cocoa using deep convolutional neural networks." *In: Workshop on Computing Applied to Environmental Management and Natural Resources (WCAMA)*. SBC, 2022. pp. 31-40.
[19] M. Fikri et al. "Detection of dry and wet cocoa beans to improve quality using the You Only Look Once architecture based on Convolutional Neural Networks." *Jurnal Nasional Teknik Elektro*, pp. 24-31, 2025.
[20] E. D. Santos; D. B. Remot. "Classification of Cocoa Beans Based on the Grade Level of Fermentation using KNN Algorithm." *Indian Journal of Science and Technology*, vol. 17, no. 21, pp. 2166-2176, 2024.
[21] A. E. C. Pino et al. "Deep Learning applied to the classification of cocoa beans (Theobroma cacao L.) according to fermentation quality." *Revista Científica y Tecnológica UPSE*, vol. 11, no. 2, pp. 92-104, 2024.
[22] K. Ayikpa et al. "Classification of cocoa beans through spectral measure analysis using machine learning and genetic algorithm." *Journal of Imaging*. vol. 10, no. 1, p. 19, 2024.
[23] M. M. Oliveira et al. "Classification of fermented cocoa beans (cut test) using computer vision." *Journal of Food Composition and Analysis*, vol. 97, p. 103771, 2021.
[24] F. A. Santos; E. S. Palmeira; G. Jesus. "An image dataset of cut-test-classified cocoa beans." Data in brief, Elsevier, vol. 24, p. 103916, 2019.
[25] O. Russakovsky et al. "Imagenet large scale visual recognition challenge." International journal of computer vision, Springer, vol. 115, pp. 211–252, 2015.
[26] P. Azunre. "Transfer learning for natural language processing." [S.l.]: Simon and Schuster, 2021.
[27] A. Krizhevsky; I. Sutskever; G. E. Hinton. "2012 alexnet". *Adv. Neural Inf. Process.* Syst, p. 1–9, 2012.
[28] K. Simonyan; A. Zisserman. "Very deep convolutional networks for large-scale image recognition". arXiv preprint arXiv:1409.1556, 2014.
[29] C. Szegedy et al. "Going deeper with convolutions". *In: Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2015.
[30] S. Ioffe; C. Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift". *In: PMLR. International conference on machine learning.* [S.l.], 2015. p. 448–456.
[31] Zhou and Chellapa. "Computation of optical flow using a neural network". *In: IEEE. IEEE 1988 international conference on neural networks*. [S.l.], 1988. p. 71–78.
[32] N. Srivastava et al. "Dropout: a simple way to prevent neural networks from overfitting". *The journal of machine learning research*, JMLR. org, v. 15, n. 1, p. 1929–1958, 2014.
[33] T. Tieleman; G. Hinton. "Lecture 6.5-rmsprop, coursera: Neural networks for machine learning". University of Toronto, Technical Report, v. 6, 2012.
[34] A. C. Mueller; S. Guido. "Introduction to machine learning with Python: a guide for data scientists". [S.l.]: "O'Reilly Media, Inc.", 2016.
[35] Y. Tian; O. Ye; D. Doermann. "Yolov12: Real-time object detectors centered on attention". arXiv preprint arXiv:2502.12524 , 2025.