CNN-BASED SOLUTION FOR MANGO CLASSIFICATION IN AGRICULTURAL ENVIRONMENTS

Beatriz Díaz Peón[1], Jorge Torres Gómez[2], Ariel Fajardo Márquez[3]

[1-3]Havana University of Technology "José Antonio Echeverría" (CUJAE), Havana, Cuba, [2]TU Berlin, Berlin, Germany
[1]e-mail: eatrizdi@automatica.cujae.edu.cu
[2]e-mail: jorge.torresgomez@tu-berlin.de

ABSTRACT

This article exemplifies the design of a methodology for the detection and classification of fruits using Convolutional Neural Networks (CNN). The goal is to develop a methodology for a system capable of automatically assessing fruit quality to optimize agricultural inventory management. Specifically, a method for mango fruit classification was developed using image processing, ensuring both accuracy and efficiency. Resnet-18 was selected as the preliminary architecture for classification, while a cascade detector was used for detection, balancing execution speed and computational resource consumption. Detection and classification results were displayed through a graphical interface developed in MatLab App Designer, streamlining system interaction. The integration of convolutional neural networks and cascade detectors proffers a reliable solution for fruit classification and detection, with potential applications in agricultural quality control.

INDEX TERMS: convolutional neural networks, MatLab, ResNet-18, cascade detector, regression with convolutional neural networks.

SOLUCIÓN BASADA EN REDES NEURONALES CONVOLUCIONALES PARA LA CLASIFICACIÓN DE MANGOS EN ENTORNOS AGRÍCOLAS

RESUMEN

Este artículo ejemplifica el diseño de una metodología para la detección y clasificación de frutas utilizando CNN. El objetivo es desarrollar una metodología para un sistema capaz de evaluar automáticamente la calidad de las frutas para optimizar la gestión del inventario agrícola. En concreto, se desarrolló un método de clasificación para el mango utilizando el procesamiento de imágenes, garantizando tanto la precisión como la eficiencia. Se seleccionó Resnet-18 como arquitectura preliminar para la clasificación, mientras que se utilizó un detector en cascada para la detección, equilibrando la velocidad de ejecución y el consumo de recursos computacionales. Los resultados de la detección y la clasificación se mostraron a través de una interfaz gráfica desarrollada en MatLab App Designer, lo que agilizó la interacción con el sistema. La integración de redes neuronales convolucionales y detectores en cascada ofrece una solución fiable para la clasificación y detección de frutas, con posibles aplicaciones en el control de calidad agrícola.

PALABRAS CLAVES: redes neuronales convolucionales, MatLab, Resnet-18, detector en cascada, regresión con redes neuronales convolucionales.

1.   INTRODUCTION

Mango crops span most tropical and subtropical regions, with a global production exceeding 43×10^6 t, it is the third most produced and imported tropical fruit. In the case of Cuba, mangoes hold significant

relevance, representing 42 % of the cultivated fruit area and 30 % of overall fruit production [1]. However, this fruit export is subject to strict international regulations, such as those outlined by the Codex Alimentarius,[1] which establishes quality criteria including ripeness, classification, and contaminant control [2]. Compliance with these standards poses a challenge in agricultural settings where fruit grading is performed manually, often resulting in subjective assessments and classification errors [1]. To address this issue, it is essential to develop an automated system that ensures objective evaluation of mango quality. As a solution, this work proposes the use of image processing techniques, specifically convolutional neural networks (CNN). This strategy enables precise fruit classification aligned with international quality standards.

Various image processing methods have proven effective for automated fruit classification. For instance, approaches combining Principal Component Analysis (PCA) and Artificial Neural Networks (ANN) have achieved approximately 79 % accuracy in detecting surface defects in apples [3]. Likewise, techniques that integrate k-means segmentation with feature extraction via Gray-Level Co-occurrence Matrix (GLCM) have demonstrated outstanding results, achieving disease recognition rates in apples of nearly 99.60 % [4]. However, these methods rely on manual feature extraction or selection algorithms that require expert knowledge, thereby limiting their generalization capacity under varying lighting conditions, camera angles, or image noise.

In contrast, CNN architectures offer significant advantages for classification and detection tasks, as they automatically extract hierarchical image features, ranging from simple patterns to complex structures [5]. Their basic components include convolutional, pooling, and fully connected layers, with activation functions such as Softmax,[2] and ReLU,[3] which optimize learning and prediction processes [5].

The use of pretrained networks such as ResNet-50, ResNet-18, AlexNet and GoogLeNet, available through the MatLab Deep Learning Toolbox, facilitates adaptation to new tasks via transfer learning [6]. For problems that require both classification and localization, architectures such as Regression with Convolutional Neural Networks (R-CNN) and its optimized variants (Fast R-CNN, Faster R-CNN, and Mask R-CNN) are employed [7]. Cascaded detectors are also used to discard irrelevant regions and accelerate processing, which is particularly useful for real-time applications [8]. These detectors focus only on regions with a high probability of containing relevant information, providing a viable solution for real-time environments.

These architectures have demonstrated high precision levels in complex environments. Deep models such as VGG and ResNet-50 have shown accuracies between 92 % and 93 % in detecting orange diseases [9] and up to 99.6 % in strawberry disease classification [10]. Similarly, the Faster R-CNN model, which combines region proposals with the robustness of CNN, has proven effective in detecting blueberry defects for export classification [11]. These capabilities justify the selection of CNN-based models as the foundation of the proposed system.

Following this trend, the main objective of this research is to develop the methodology for a system based on CNN models for the detection and classification of mangoes. To this end, suitable architectures were selected, preprocessing algorithms were applied to the datasets and hyperparameters[4] were tuned through experimental trials. Additionally, a graphical interface was developed using MatLab App Designer to integrate the trained models, enabling practical use in real environments.

The obtained results validate the effectiveness of the proposed methodology. The Resnet-18 architecture achieved an accuracy of 89.51 % for ripeness classification and 90.65 % for disease classification. These figures indicate strong performance both in assessing mango maturity and in identifying potential diseases. Moreover, the cascaded detector reached an accuracy of 90 % in mango detection within images,

---

[1]Codex Alimentarius: `http://www.codexalimentarius.org/codexhome/en/`. Specifically, CODEX STAN 184-1993: `www.codexalimentarius.org/download/standards/315/CXS\_184e.pdf`

[2]Calculates the probability of each class relative to all possible classes.

[3]Enables the model to approximate nonlinear functions by linking pixels with the network's semantic content.

[4]Configurations that are not learned directly during training but must be set beforehand or during network construction.

confirming its applicability in agricultural environments. It was also observed that excessively deep networks may lead to overfitting,[5] which impairs model generalization. Thus, smaller and more task-specific architectures were favored.

Finally, different detection models were evaluated. Although R-CNN-based architectures offered good precision, their high execution time makes them less suitable for real-time applications and systems with limited computing resources. In contrast, the cascaded detector showed a better balance between speed and accuracy, making it more appropriate for resource-constrained environments.

The rest of the article continues as follows: In Section 2, the methodology used to create the models is presented. In Section 3, the performance of the proposed system is discussed. Finally, the last section concludes the article, highlighting the main features of this proposal.

2. MODEL DEVELOPMENT FOR MANGO CLASSIFICATION AND DETECTION

This section outlines the methodology used to develop the mango detection and classification system. It details the selection and preprocessing of the datasets, as well as the architectures employed for each task. For detection, models based on R-CNN and cascade detectors were used, while classification was addressed through CNN architectures adapted for identifying ripeness stages and diseases. Modifications made to the architectures and the training process are also explained.

Dataset search and selection for training

Selecting an appropriate dataset is essential for effective neural network training, as it directly impacts model accuracy and generalization. In this research, datasets were analyzed according to three specific tasks: 1) classification of mango ripeness, 2) disease classification, and 3) mango detection in images. Since each task requires different label types and structural formats, a single dataset could not meet all requirements. Therefore, a comprehensive search was conducted across repositories such as Roboflow,[6] GitHub,[7] and Mendeley Data,[8] prioritizing datasets with sufficient volume, visual diversity and well-defined labels.

After selecting a general set of datasets for the various tasks, three relevant options were evaluated and compared for mango ripeness classification. The considered datasets were: "Mango detection system yolov8.v2-yolov8-new-dataset",[9] "Mangos.v6i",[10] and "Mango Variety and Grading Dataset".[11]. Table 1 summarizes their main characteristics.

Based on this analysis, the "Mango detection system yolov8.v2-yolov8-new-dataset" was selected. Its high resolution and diversity of lighting and background conditions contribute to stronger model performance in real-world environments. Moreover, it is considerably larger than the other datasets evaluated, allowing better class representation and minimizing the risk of overfitting. It contains 17 009 images at 640×640 pixels resolution, divided into three categories: bad mango (6856 images), raw mango (4772 images) and ripe mango (4672 images), as shown in Fig. 1. Notably, this dataset includes three variations per image. Approximately 50 % of the images are flipped either vertically or horizontally, randomly rotated between $-15°$ and $15°$ and subjected to random Gaussian blur in the range of $0 - 2.3$ pixels. These transformations, integrated into the dataset, improve model robustness against visual variations.

---

[5] https://enclavedeciencia.rae.es/sobreajuste
[6] https://universe.roboflow.com/
[7] https://github.com/
[8] https://data.mendeley.com/
[9] https://universe.roboflow.com/mango-yolov8/mango-detection-system-yolov8-mcocq
[10] https://universe.roboflow.com/mango-detection/mangos-kyo1z
[11] https://data.mendeley.com/datasets/5mc3s86982/1

Table 1: Comparison of datasets for mango ripeness classification.

Dataset	Number of categories	Number of images	Noise conditions	Lighting and background variations
Mango detection system yolov8.v2-yolov8-new-dataset	3	17 009	Included	Included
Mangos.v6i	3	8953	Not included	Limited
Mango Variety and Grading Dataset	11	2200	Not included	Not included

(a) Bad mango   (b) Raw mango   (c) Ripe mango
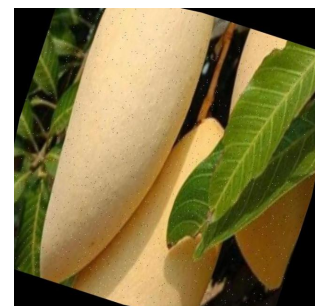
Figure 1: Digital images of the categories: bad mango, raw mango and ripe mango.

For mango disease classification, three datasets were identified and evaluated independently of the one used for ripeness classification, as they address a distinct task within the study. The datasets considered were: "project > 2023-12-04 1:29pm",[12] "Mango.v6i.multiclass",[13] and "mangos.v7i.multiclass".[14] Table 2 summarizes them based on the number of classes, image volume, and disease representativity.

Table 2: Comparison of datasets for mango disease classification.

Dataset	Number of categories	Number of images	Noise conditions	Lighting and background variations
project > 2023-12-04 1:29pm	5	4034	Yes	Yes
Mango.v6i.multiclass	5	9825	No	Yes
mangos.v7i.multiclass	3	485	No	No

Among the analyzed datasets, "project > 2023-12-04 1:29pm" was selected due to its wide range of image variants. It includes 4034 images with 640×640 pixels resolution, divided into five categories: alternaria (795 images), anthracnose (614 images), black mold rot (877 images), healthy (980 images) and stem end rot (766 images), as illustrated in Fig. 2. This dataset also contains three image variants per sample. The transformations applied include horizontal flipping (50 %), random rotation of 90° (none, clockwise, or counterclockwise), random cropping of up to $0 - 20\%$ and Gaussian blur within $0 - 2.5$ pixels. These embedded variants prepare the model for real-world conditions, where images may exhibit noise, slight blur, or rotation.

---

[12]https://universe.roboflow.com/project-khw09/project-airhk
[13]https://universe.roboflow.com/cocoa-f6c3y/mango-qhzr4
[14]https://universe.roboflow.com/yolo-pzqub/mangos-zjjq7

(a) Alternaria

(b) Anthracnose

(c) Black mold rot

(d) Healthy

(e) Stem end rot

Figure 2: Digital images of the categories: alternaria, anthracnose, black mold rot, healthy and stem end rot.

Finally, for mango detection in images, the dataset "572 FRUITS VEGETABLES.v1i.tensorflow"[15] was selected. It consists of 11 489 images at 640×640 pixels resolution. Unlike the previous datasets, it is not folder-organized but includes a `CSV` file specifying each image's filename, width, height, class, and bounding box coordinates. The existing categories include mango, strawberry, banana, guava, tomato, onion, egg, avocado, cherry, peach, apple, pineapple, kiwi, among others. Some examples are shown in Fig. 3. As an alternative, the dataset "FruitMate.v2i.coco"[16] was considered. It contains 5977 images and 10 categories. However, its smaller size and limited variability in background and lighting conditions could hinder model training for classification in real environments. Thus, it was discarded in favor of the first dataset.
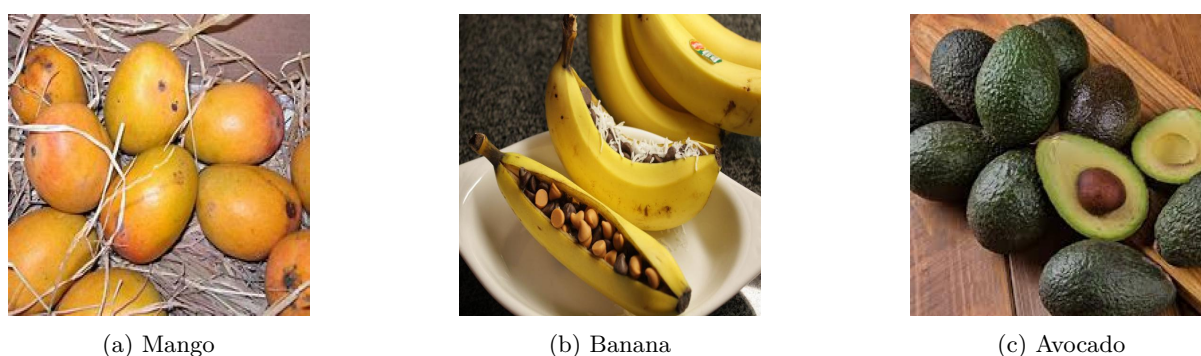

(a) Mango

(b) Banana

(c) Avocado

Figure 3: Digital images from different categories: mango, banana and avocado.

---

[15]https://universe.roboflow.com/ml-datatrain/572_fruits_vegetables
[16]https://universe.roboflow.com/tharushi-qpqyp/fruitmate

Data preprocessing for neural network training

Data preprocessing is a crucial step to ensure image compatibility with the selected models. This process involves several fundamental operations:

1. Dataset splitting: Images are divided into training and validation sets (MatLab recommends a $70-30\%$ split)[17] [12]. This separation can be performed in two ways:

   1.1 Random/Proportional Sampling: Random selection of images or selection based on predefined proportions from the original dataset.

   1.2 Predefined Structure: Use of organized folders such as `train`, `valid` and `test`, if the dataset already follows that structure.

2. Image resizing: Images are adjusted to the standard input sizes required by the pretrained models used (224×224 pixels and 227×227 pixels), such as ResNet-50 , GoogLeNet and AlexNet [13]. This step ensures compatibility and computational efficiency. The original dataset structure is preserved during this process to facilitate handling.

3. Bounding box adjustment (for object detection): In detection tasks, it is necessary to modify the xy coordinates of the bounding boxes to maintain their proportions relative to the new image size. This ensures that the regions of interest continue to correspond correctly to the detected objects.

Finally, the MatLab Image Labeler tool was used to verify the correct modification of both images and bounding boxes. This tool allows manual inspection and correction of alignment errors or missing labels, although the process can be time-consuming.

Architectures for mango classification and detection

The proposed solution consists of two sequential stages, as illustrated in Fig. 4. First, all mangoes present in the whole image are detected, generating individual crops for each detection. Then, each crop is analyzed separately during the classification stage, where both ripeness and disease presence are determined. This workflow ensures that the system processes only regions validated as mangoes, optimizing computational resources and enabling more precise object-level analysis.
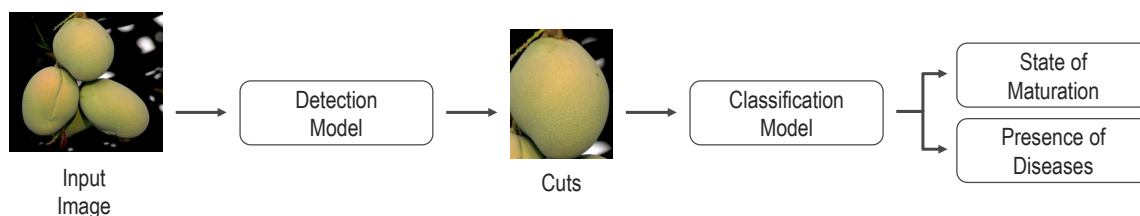

Figure 4: Workflow of the mango detection and classification system.

To classify ripeness and diseases, the pretrained Resnet-18 architecture was used. It comprises 18 layers, including convolutions, batch normalization, and ReLU activation functions. Its residual block structure,[18] facilitates gradient flow during training, enhancing model stability and accuracy [14]. Fig. 5 shows a schematic representation of this architecture.

---

[17]This division can be adjusted based on the problem. Validation should not significantly exceed the training set, as it may slow down the model's learning process.

[18]These blocks directly connect the output of specific layers with subsequent layers.

As Resnet-18 was initially designed to classify up to 1000 categories, its final layers were modified to match the number of classes required for this problem [13]. These layers combine extracted features from the convolutional blocks to generate class probabilities, compute the loss function and produce predicted labels. In ResNet-18, the last trainable layer is fully connected and it was replaced with a new one containing the appropriate number of output nodes (3 for ripeness, 5 for disease detection). Additionally, the classification layer was replaced so that the model could learn the dataset labels directly [12]. This adaptation ensures that Resnet-18 is optimized for the specific classification tasks presented.

Figure 5: Representation of the Resnet-18 architecture.

For detection, two classic CNN architectures—AlexNet and VGG-16—were evaluated and modified, as illustrated in Fig. 6. AlexNet consists of 5 convolutional layers, 3 pooling layers, 3 fully connected layers and a Softmax layer [15]. VGG-16, on the other hand, has 16 layers in total (13 convolutional and 3 fully connected), uses 3×3 convolutions with a large number of filters and contains approximately 138×10^6 parameters [16].

Figure 6: Representation of the AlexNet and VGG-16 architectures.
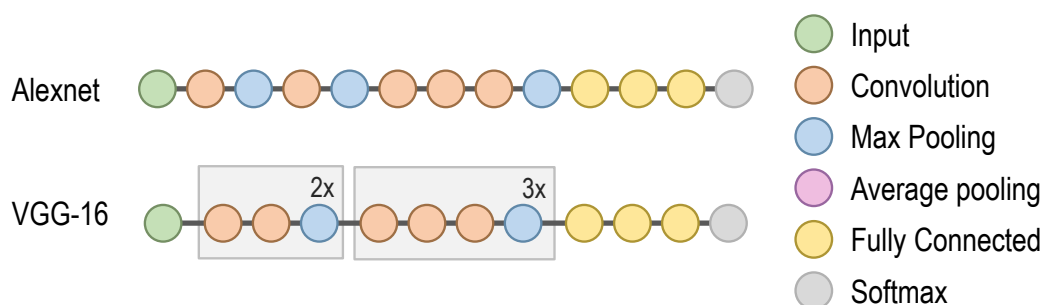
Both networks were adapted to the R-CNN framework by removing their original final layers (fully connected, Softmax and classification), which were replaced with new layers configured to classify proposed regions, including an additional background class. This process is similar to the one used to adapt CNN to new datasets. The strategy retains the pretrained convolutional layers for feature extraction, while the added layers specialize in detecting relevant mango-related objects [17].

In addition to R-CNN-based models, a cascade object detector was employed to provide a lightweight alternative for real-time environments. The cascade detector is particularly effective for recognizing object categories with relatively stable aspect ratios (faces, traffic signs, and vehicles). It operates using a hierarchical architecture that classifies image windows through multiple stages. Each stage quickly eliminates regions unlikely to contain the target object, allowing the system to focus computational resources on the most promising areas. At every step, object-specific features are evaluated to determine whether the region advances to the next classifier or is discarded [8].

Training for detection and classification

Two approaches were employed for object detection in images: the R-CNN method using pretrained networks (VGG-16 and AlexNet) and cascade detector. The CNN architectures were adjusted to create an `rcnnObjectDetector` object through an initial training phase. Final training was conducted using the Stochastic Gradient Descent with Momentum (SGDM) solver, with a mini-batch size of 32, over 10 epochs and an initial learning rate of 0.000 001.

In the case of the cascade detector, different parameter configurations were evaluated, as shown in Table 3. Initial tests produced multiple false detections; however, by increasing the value of `NumCascadeStage`,[19] and reducing the `FalseAlarmRate`,[20] detection accuracy significantly improved in Tests 5 and 6. Ultimately, the configuration in Test 6 (`FalseAlarmRate` of 0.05, `NumCascadeStage` of 10) was selected for providing the best balance between precision and computational efficiency.

Table 3: Training parameters of the cascade detector.

Hyperparameter	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6
FalseAlarmRate	0.01	0.1	0.2	0.05	0.15	0.05
NumCascadeStage	10	5	15	5	20	10
ObjectTrainingSize	auto	[24 24]	auto	auto	auto	auto

For ripeness classification (raw, ripe, and rotten), the Resnet-18 architecture described earlier was trained. A total of 14 884 images were used for training, 1416 for validation and 709 for testing. To balance the training and validation data, 750 images per class were randomly selected for training and 100 images per class for validation. This strategy contributed to a robust model, enabling accurate performance evaluation.

Various hyperparameter configurations were tested, as summarized in Table 4. The final model was trained using the SGDM solver, with an initial learning rate of 0.001, a mini-batch size of 32 and 10 epochs, applying a progressive learning rate drop of 0.01 per epoch.

Table 4: Hyperparameter configuration.

Hyperparameter	Test 1	Test 2	Test 3	Test 4	Test 5
InitialLearnRate	0.001	0.01	0.01	0.001	0.001
LearnRateSchedule	not applied	piecewise	piecewise	not applied	piecewise
LearnRateDropPeriod	not applied	3	1	not applied	4
LearnRateDropFactor	not applied	0.1	0.01	not applied	0.1
L2Regularization	not applied	not applied	not applied	0.01	50

Once the ripeness was classified, the next step was to identify possible diseases. The "spoiled" category includes deteriorated fruits regardless of ripeness, as damage may result from either senescence or disease. Therefore, further analysis is necessary to identify the specific cause and assess the risk of total crop loss. Following experimentation, two approaches were considered:

1. Combined multi-category model: A single network for ripeness and disease (higher complexity, lower precision).

---

[19] Number of cascade stages: Specifies how many stages the cascade detector applies.
[20] False alarm rate: Indicates the percentage or frequency of incorrect detections.

2. Dedicated model for disease classification: A separate network for disease detection (more interpretable and accurate).

Since the first approach resulted in lower precision and higher ambiguity (e.g., mistaking ripening spots for pathological symptoms), the second alternative was chosen. Resnet-18 was again employed, adapted to five disease categories. A total of 3537 images were used for training, 331 for validation and 166 for testing. A total of 300 images per class were randomly selected for training and 50 per class for validation. Training was performed using hyperparameters similar to those used in the previous stage: the SGDM solver, initial learning rate of 0.01, mini-batch size of 32 and 10 epochs, with a learning rate reduction of 0.01 to enhance convergence.

3.   RESULT ANALYSIS

This section presents the results obtained in the detection and classification of mangoes according to their ripeness and disease status. For object detection, the dataset "572 FRUITS VEGETABLES.v1i.tensorflow" was used, evaluating both R-CNN models (based on VGG-16 and AlexNet) and cascade detectors. Classification tasks were addressed using Resnet-18 models, trained with the datasets "Mango detection system yolov8.v2-yolov8-new-dataset" (for ripeness) and "project > 2023-12-04 1:29pm" (for diseases). Additionally, precision, stability, and computational efficiency metrics were analyzed to determine the most suitable configurations.

Training and validation of the mango detection model

Two R-CNN-based models—VGG-16 and AlexNet—were evaluated. Both training processes were manually stopped to prevent overfitting due to excessive iterations. Table 5 summarizes the main results obtained for the training phase. It can be observed that both models exhibit quite similar accuracy, while the VGG-16 model excels with the lowest training loss.

Using these two models, detection results for each model are illustrated in Fig. 7. Green boxes indicate ground truth, while red boxes represent predictions generated by the models. By comparing both results, a better correspondence is observed for the VGG-16 model compared to AlexNet.

Table 5: Training results of R-CNN.

Models	Training Accuracy (%)	Training Loss	Training Time	Iteration
VGG-16	79.00 %	0.75	417 min 7 s	597 from 7490
AlexNet	80.00 %	0.5	100 min 54 s	1695 from 7490

AlexNet demonstrated good generalization capability, detecting even unlabeled objects, though with slight misalignments in some bounding boxes. VGG-16, due to its greater depth, achieved better alignment of detections, albeit with higher computational cost. Both models produced acceptable results, balancing precision, speed and complexity.

As a lighter alternative, a cascade object detector was evaluated. Fig. 8 shows that initial testing produced unsatisfactory results, with multiple incorrect detections and irrelevant regions, indicating low discrimination capability. However, after adjusting the parameters, by increasing the number of stages and lowering the false alarm rate, accuracy improved significantly, as illustrated in Fig. 9. Predicted boxes were better aligned with objects of interest, and false detections on leaves and other structures were notably reduced.
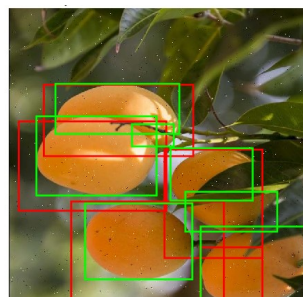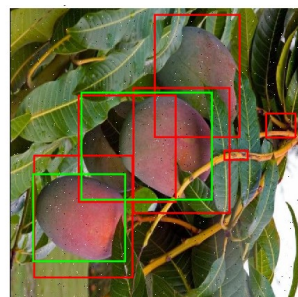
(a) AlexNet   (b) VGG-16

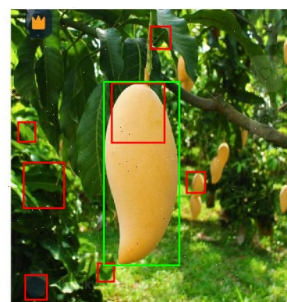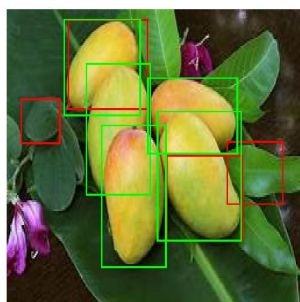Figure 7: Mango detection using R-CNN-based models: AlexNet and VGG-16

Figure 8: Mango detection in initial tests with the cascade detector

The cascade detector was ultimately selected for its simplicity, speed, and strong performance even with limited training data. Its ability to operate in real time and under variable lighting and scale conditions makes it ideal for practical applications. Although R-CNN models offer higher precision in complex environments, their high computational cost makes them less suitable for resource-constrained systems.

Training and validation of the ripeness classification model

A model based on Resnet-18 was trained to classify mangoes into three ripeness stages: raw, ripe, and rotten. The model achieved a validation accuracy of 89.51 %, with no evidence of overfitting, indicating a strong generalization capability. Fig. 10(a) shows a representative result: the model assigns a probability of 91.1 % to the "ripe mango" class, with significantly lower values for the other categories.
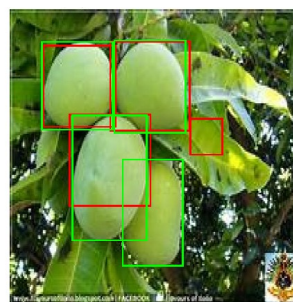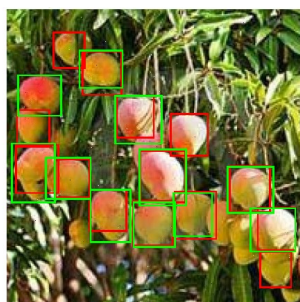

Figure 9: Mango detection in images using the cascade detector

This distribution indicates that the model has successfully learned to identify distinctive visual patterns, even in samples with ambiguous features. The confusion matrix[21] shown in Fig. 10(b) supports this outcome: the "bad mango" class reached 95.3 % accuracy, the ripe mango class 93.3 % and the "raw mango" class 89.3 %, demonstrating clear category differentiation. This consistent performance confirms ResNet-18's suitability as the base model for ripeness classification.
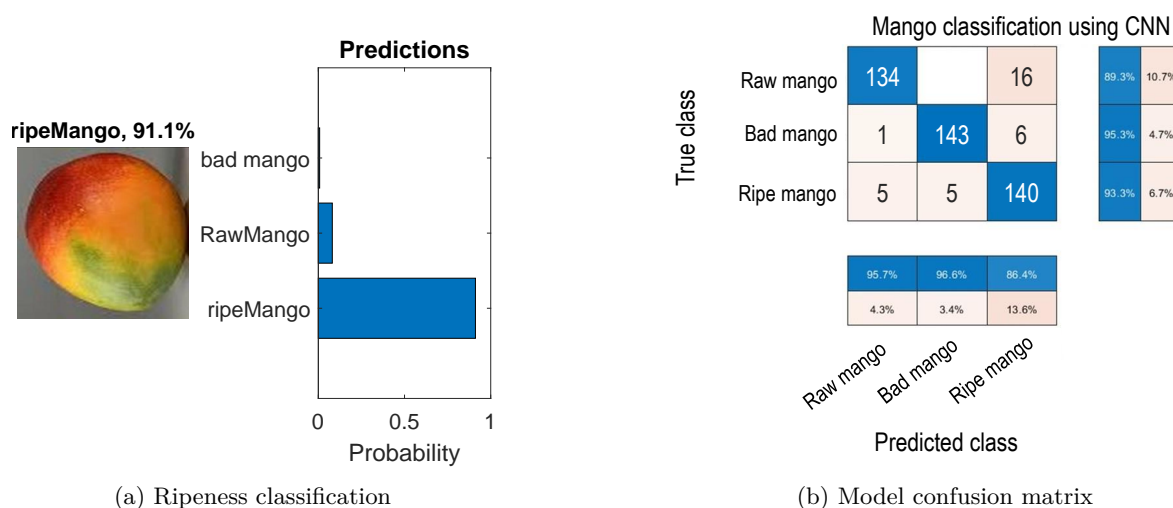

(a) Ripeness classification

(b) Model confusion matrix

Figure 10: Ripeness classification of mangoes using the Resnet-18 model.

As part of the comparative analysis, alternative architectures such as ResNet-50, ResNet-101, GoogLeNet, and AlexNet were evaluated across five independent tests. Table 6 summarizes the resulting accuracies. It is observed that the less complex networks—Resnet-18 and AlexNet—produced better and more stable results than the more complex architectures—ResNet-50 and ResNet-101—even though validation accuracies were relatively low across the board. This suggests that for this relatively simple task, smaller models are more effective and less prone to overfitting.

Table 6: Model Accuracy.

Model	Test 1	Test 2	Test 3	Test 4	Test 5
ResNet-50	86.67 %	78.33 %	43.33 %	78.33 %	33.33 %
Resnet-18	93.33 %	55.00 %	33.33 %	83.33 %	16.67 %
ResNet-101	81.67 %	30.00 %	40.00 %	85.00 %	33.33 %
GoogLeNet	76.67 %	NaN[22]	NaN	90.00 %	33.33 %
AlexNet	73.33 %	NaN	33.33 %	46.67 %	33.33 %

AlexNet and GoogLeNet were selected alongside Resnet-18 for a performance comparison, due to a favorable balance between accuracy, stability, and generalization. Their respective confusion matrices are presented in Fig. 11. For AlexNet (Fig. 11(a)), strong performance was observed in the "raw mango" class (92.0 %), though higher confusion appeared in the "bad mango" category (84.7 %), indicating limitations when classifying samples with ambiguous visual traits. GoogLeNet (Fig. 11(b)) showed moderate performance overall, with frequent errors in the "ripe mango" class (85.3 %), likely due to

---

[21]Values on the main diagonal represent correct classifications; off-diagonal values correspond to classification errors, including false positives and false negatives.

[22]Not a Number. This error may arise from exploding gradients, invalid data (such as infinite values), or optimization issues where the algorithm fails to reach a global minimum.

subtle image variations or mild overfitting. Collectively, the obtained results and matrix patterns confirm that Resnet-18 provides the best balance of accuracy, robustness and computational efficiency. As such, Resnet-18 was selected as the most suitable model for mango ripeness classification.
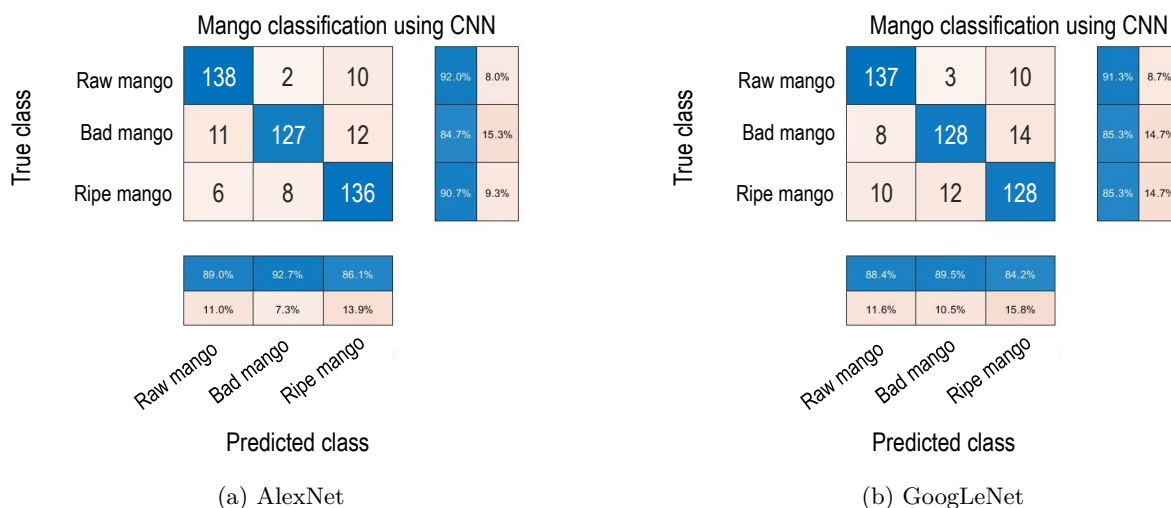

(a) AlexNet   (b) GoogLeNet

Figure 11: Confusion matrices for AlexNet and GoogLeNet models.

Training and validation of the disease classification model

A Resnet-18 model was trained to classify mangoes into five categories: alternaria, anthracnose, black mold rot, stem end rot and healthy. In Fig. 12(a), a prediction example is shown in which the model assigns a probability of 58 % to the "stem end rot" class, consistent with the visible characteristics in the image. The low probabilities assigned to the remaining classes indicate that the model can distinguish between pathological symptoms and normal conditions correctly.

(a) Disease classification   (b) Confusion matrix of the model
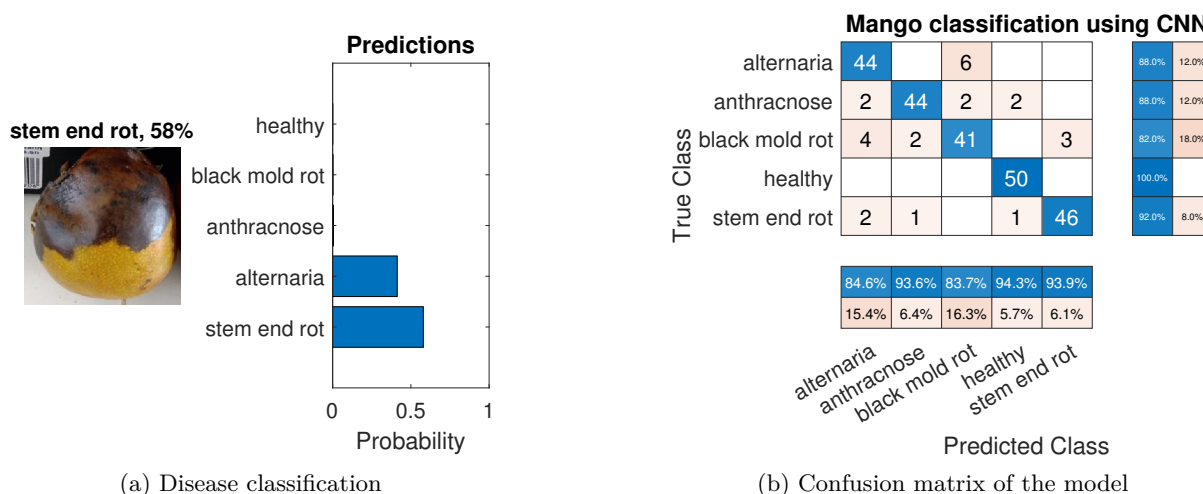
Figure 12: Classification of mango diseases using a dedicated disease model.

The confusion matrix shown in Fig. 12(b) demonstrates solid performance, with accuracies above 88.0 % for all categories and relatively low misclassification rates. The "healthy" class was correctly classified

in 100 % of cases, indicating a clear distinction between healthy and diseased fruits. The "alternaria", "anthracnose" and "black mold rot" categories showed some confusion among themselves, which was expected given the morphological similarity of their symptoms. These results confirm that the dedicated disease classification model is reliable, especially for classes with well-defined visual patterns.

Subsequently, a combined model was trained to classify both diseases and ripeness stages. Fig. 13(a) shows a case where the model assigns the highest probability (28 %) to "alternaria," although visually the fruit exhibits more distinctive symptoms of "stem end rot". This prediction reflects higher uncertainty, likely due to overlapping features between classes.

(a) Disease classification
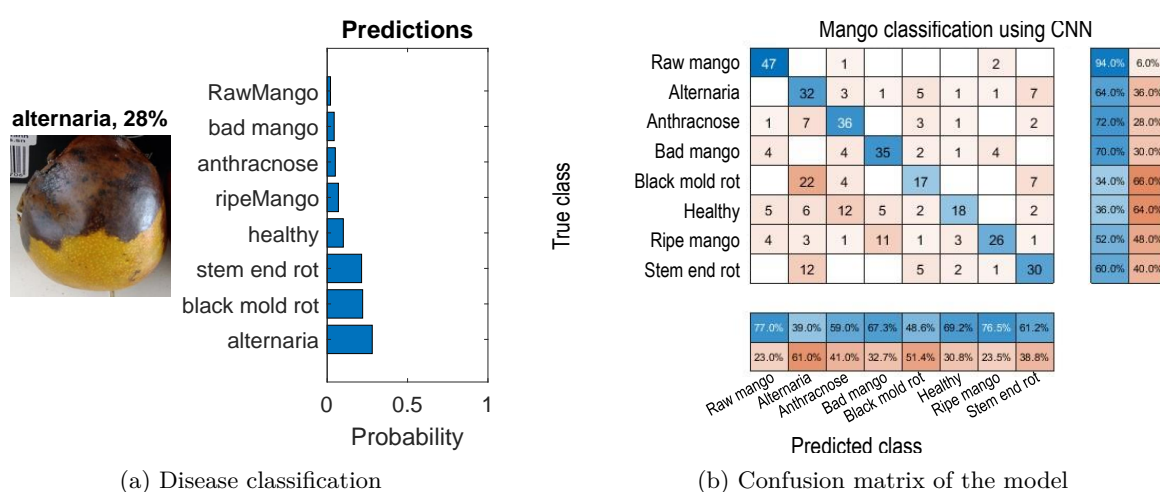

(b) Confusion matrix of the model

Figure 13: Classification of mangoes using a combined model with multiple categories.

The confusion matrix shown in Fig. 13(b) confirms this trend: a greater dispersion of errors is observed, particularly among diseases with similar symptoms and between diseased and ripe fruits. The "alternaria", "anthracnose" and "black mold rot" classes show frequent cross-classification errors, as do "bad mango" and "stem end rot". Additionally, in some cases, diseased fruits were classified as "ripe mango", suggesting that the model may interpret certain symptoms as part of the natural ripening process.

This behavior indicates that including multiple categories in a single network may hinder discrimination between classes with visually similar symptoms, affecting the model's specialization. Although the combined model yields functional results, its overall performance is lower in terms of precision and interpretability. Therefore, it is not recommended to address ripeness and disease classification within a single model simultaneously. Dividing the tasks allows for more targeted treatment of each problem, reducing complexity and improving the clarity of the results.

Visual application

The developed models were integrated into a visual application, shown in Fig. 14, using MatLab's App Designer to optimize result visualization. The application includes several functions to facilitate user interaction with images and the classification process:

- Load Image: allows the user to select images from the device.

- Open/Close Camera: enables real-time image capture.

- Select Fruit: applies the detection algorithm to identify mangoes within the image.

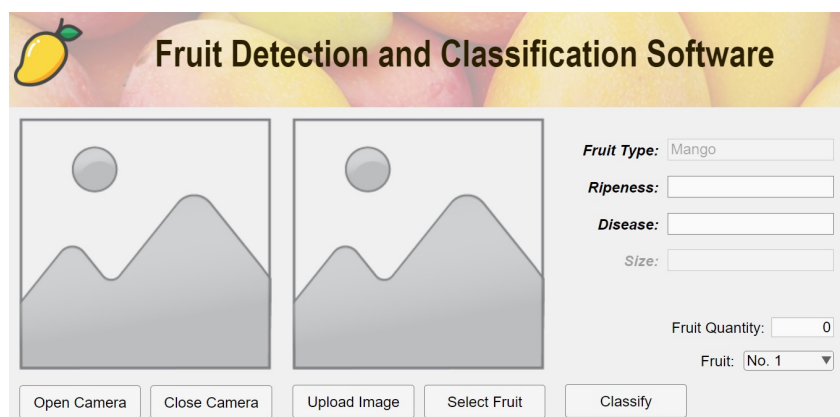■ Classify: runs the classification model on the selected image fragment.

Figure 14: Visual interface for mango classification.

MatLab App Designer was chosen due to its convenience for designing and programming visual interfaces. Additionally, using the same software for both model training and the visual application helps avoid potential compatibility issues in future implementations.

4. CONCLUSIONS

The results introduced in this paper confirm that CNN are effective for mango detection and classification, integrating a cascade object detector and Resnet-18 into a visual application developed with MatLab App Designer. Although the system improves classification accuracy, reduces human error and eliminates the need for destructive methods, its effectiveness compared to traditional methods remains to be validated in real agricultural environments. Additionally, the developed interface facilitates access to information and supports the export of mangoes by meeting international quality standards. Future work will consider the integration of this solution within a real-time system pipeline. Examples include the monitoring of crops with drones, where integrated cameras support the collection of images that this CNN-based autonomous system can evaluate. Likewise, its incorporation into automated sorting lines in processing plants is considered, where the system could analyze moving fruits to make fast and accurate decisions without human intervention. These implementations will require adapting the model to real production scenarios, optimizing it for ripeness level, pest presence, system cost, and other practical criteria, and developing proprietary databases to evaluate performance against alternative methods. These improvements will enhance generalization and ensure practical applicability.

5. References

[1] Instituto de Investigaciones de Fruticultura Tropical (IIFT) y Grupo Agrícola (GAG). "Technical Manual for Mango Cultivation [in Spanish]". In: *Project for Supporting the Strengthening of Local Fruit Value Chains (AGROFRUTALES)* (2023), p. 68.

[2] Codex Alimentarius. "International Food Standards: Mango Standard (CODEX STAN 184-1993)". In: *Codex Standard for Mango* (2005), pp. 184–1993.

[3] BS Bennedsen DL Peterson and A Tabb. "Identifying Apple Surface Defects Using Principal Components Analysis and Artificial Neural Networks". In: *Transactions of the ASABE* 50.6 (2007), pp. 2257–2265.

[4] Sangeetha. "Detection and Classification of Apple Fruit Disease Using K-NN Classification and GLCM Features". In: 2018.

[5] Luis Rodrigo Barba Guamán. "Use of Deep Learning Techniques for Object Recognition in Rural Areas [in Spanish]". PhD thesis. ETSI Sistemas Informáticos, 2021.

[6] The MathWorks Inc. *Deep Learning Toolbox, version 14.2 (R2022b)*. Natick, Massachusetts, United States, 2022.

[7] MathWorks. *Getting Started with R-CNN, Fast R-CNN, and Faster R-CNN*. 2024.

[8] MathWorks. *Get Started with Cascade Object Detector*. 2024.

[9] R Agrawal, K Kumar, and S Vashishth. "Orange Fruit Disease Detection Using Deep Convolutional Neural Networks". In: *International Journal of Advanced Science and Technology* 29.05 (2020), pp. 11146–11153.

[10] Jia-Rong Xiao et al. "Detection of Strawberry Diseases Using a Convolutional Neural Network". In: *Plants* 10.1 (2020), p. 31.

[11] Wily Aldo Narciso Horna and Edgar A Manzano. "Computer Vision System Based on Convolutional Neural Networks for Export-Grade Blueberry Selection [in Spanish]". In: *Revista Campus* 26.32 (2021).

[12] MathWorks. *Retrain Neural Network to Classify New Images*. 2024.

[13] MathWorks. *Pretrained Deep Neural Networks*. 2024.

[14] Product Teacher. *ResNet18 and ResNet50 in Computer Vision*. 2024. URL: https://www.productteacher.com/quick-product-tips/resnet18.

[15] Viso.ai. *AlexNet: A Revolutionary Deep Learning Architecture*. Dec. 2024. URL: https://viso.ai/deep-learning/alexnet/.

[16] Viso.ai. *Essential Guide to Very Deep Convolutional Neural Networks (VGG)*. Dec. 2024. URL: https://viso.ai/deep-learning/vgg-very-deep-convolutional-networks/.

[17] MathWorks. *Create R-CNN Object Detection Network*. 2024.

AUTHORS SECTION

Beatriz Díaz Peón: Professor at the Havana University of Technology "José Antonio Echeverría" (CU-JAE), Havana, Cuba. Automation Engineer. Member of the Image Processing Research Group. Teaching category: Instructor-in-training. ORCID ID 0009-0007-7672-591X

Jorge Torres Gómez: Senior Researcher at the School of Electrical Engineering and Computer Science, TU Berlin, Berlin, Germany. His research interests are in the fields of 6G wireless technologies, molecular communications, and teaching methodologies. ORCID ID 0000-0001-9523-048X

Ariel Fajardo Márquez: Professor at the Havana University of Technology "José Antonio Echeverría" (CUJAE), Havana, Cuba. Master of Science Automation Engineer. Member of the Image Processing Research Group. Teaching category: Assistant Professor. ORCID ID 0009-0003-4586-677X

CONFLICT OF INTEREST

The authors declare that there are no conflicts of interest regarding the content of this article.

CONTRIBUTIONS OF THE AUTHORS

- Author 1: Beatriz Díaz Peón - 60 % Conceptualization, writing and development of the article and critical review of each draft version.

- Author 2: Jorge Torres Gómez - 20 % Conceptualization, critical review of each draft version and approval of the final version to be published.

- Author 3: Ariel Fajardo Márquez - 20 % Conceptualization, critical review of each draft version and approval of the final version to be published.