Engineering Research Express

INCLUSIVE PUBLISHING TRUSTED SCIENCE

PURPOSE-LED PUBLISHING™

PAPER • **OPEN ACCESS**

Machine learning system for commercial banana harvesting

To cite this article: Ahatsham Hayat *et al* 2024 *Eng. Res. Express* **6** 035202

View the article online for updates and enhancements.

You may also like

- SuNeRF: 3D Reconstruction of the Solar EUV Corona Using Neural Radiance Fields
Robert Jarolim, Benoit Tremblay, Andrés Muñoz-Jaramillo et al.
- Fine-tuning of interlayer time interval in WAAM process to enhance the wear resistance of Al5356 walls
M Saravana Kumar, N Jeyaprakash and Che-Hua Yang
- Establishment proper of the balanced scorecard indicators to support decision making in a university: a case study in Institut Teknologi Indonesia
L Theresia, A H Lahuddin and R Bangun

This content was downloaded from IP address 193.136.232.254 on 09/05/2025 at 12:16

Engineering Research Express

PAPER

Machine learning system for commercial banana harvesting

Ahatsham Hayat[1], **Preety Baglat**[2,3], **Fábio Mendonça**[2,3], **Sheikh Shanawaz Mostafa**[3] and **Fernando Morgado-Dias**[3,4]

[1] Department of Electrical and Computer Engineering, University of Nebraska-Lincoln, NE, United States of America
[2] University of Madeira, 9000-082 Funchal, Portugal
[3] Interactive Technologies Institute (ITI/LARSyS) and ARDITI, 9020-105 Funchal, Portugal
[4] Faculty of Exact Sciences and Engineering, University of Madeira, 9000-082 Funchal, Portugal

E-mail: preety.baglat@iti.larsys.pt

Keywords: banana bunch harvesting, deep learning, computer vision, agriculture industry

OPEN ACCESS

RECEIVED
21 February 2024

REVISED
1 June 2024

ACCEPTED FOR PUBLICATION
27 June 2024

PUBLISHED
8 July 2024

Original content from this work may be used under the terms of the Creative Commons Attribution 4.0 licence.

Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

Abstract

The conventional process of visual detection and manual harvesting of the banana bunch has been a known problem faced by the agricultural industry. It is a laborious activity associated with inconsistency in the inspection and grading process, leading to post-harvest losses. Automated fruit harvesting using computer vision empowered by deep learning could significantly impact the visual inspection process domains, allowing consistent harvesting and grading. To achieve the goal of the industry-level harvesting process, this work collects data from professional harvesters from the industry. It investigates six state-of-the-art architectures to find the best solution. 2,685 samples were collected from four different sites with expert opinions from industry harvesters to cut (or harvest) and keep (or not harvest) the banana brunch. Comparative results showed that the DenseNet121 architecture outperformed the other examined architectures, reaching a precision, recall, F1 score, accuracy, and specificity of 85%, 82%, 82%, 83%, and 83%, respectively. In addition, an understanding of the underlying black box nature of the solution was visualized and found adequate. This visual interpretation of the model supports human expert's criteria for harvesting. This system can assist or replace human experts in the field.

1. Introduction

In the food sector, fruit quality is of the utmost importance. Meeting customer demands and producing high-quality fruits at a rapid rate necessitates the use of high-performance technology [1]. In addition, the food business is one of the few industries with restrictive criteria and limits due to its reliance on weather conditions and the labor market [2]. For instance, if the fruits were not harvested at the optimal time owing to weather circumstances, the quality and quantity of the harvest may decline due to inclement weather and excessive ripening of the fruits.

Fruit harvesting is a decision-making job that determines whether the fruit is ready to harvest based on the various characteristics, such as visual features of the fruit [3]. Human harvesters have overcome the majority of technical procedures in the food [4] harvesting business, frequently using a mixture of automated and manual operations. However, these harvesters are occasionally vulnerable to fatigue due to lack of sleep and overwork, which can adversely impact their performance when visually making decisions. Therefore, it could be treated as a computer vision issue that can be potentially addressed by using machine learning [5].

In current years, different visions based on fruits such as apple [6] were done using convex methods [6] and deep learning method [7]. These fruit detection methods are the first step of the fruit harvesting process. After detecting the fruit, the decision is made to harvest or not. These problems are also majorly present in the banana-related industry, which uses the most widely cultivated fruit crop for commercial purposes and is the predominant staple food in numerous developing nations [8].

In the banana harvesting process, as it is known for each ripening stage, bananas have different nutritional properties. Therefore, selecting ripe bananas is essential to acquire fruit that meets all requirements [9]. Before

© 2024 The Author(s). Published by IOP Publishing Ltd

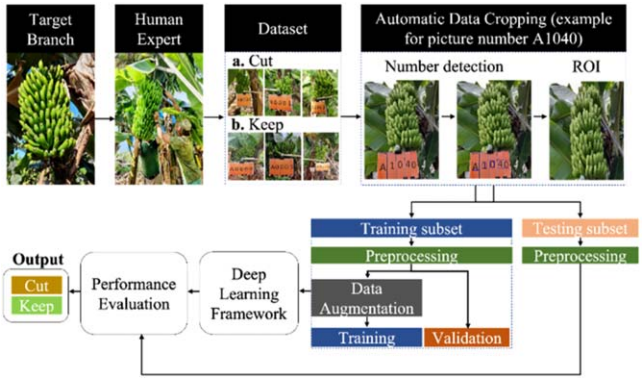

Figure 1. Proposed system for banana branch harvestability classification.

ripening begins, bananas are often collected in the matured green stage, where they stay firm and green without major changes in peel color, texture, or composition [10].

Commercial harvesting is a balancing act where the bananas are harvested in green stages for transport and storage. It also needs to keep the nutritional content. Furthermore, different factors, such as weather, price, moisture, and age of the bananas transport, play a role. Normally, human experts classify the right banana stage for harvesting using visual clues. Such a procedure is likely to be subjective and error-prone. These types of problems can be solved by computer vision with machine learning. Previously, computer vision was used in agriculture for sorting and grading banana crops [11].

Various computer vision-based techniques were also developed to evaluate the ripening stage of bananas based on the bananas' morphology [12]. These computer vision algorithms have the potential to provide an automated and non-destructive tool for banana sorting based on the visual features of the banana. A detailed literature review was done by Baglat *et al* [13]. The literature overview shows that various studies have been conducted in the past for banana grading using deep-learning approaches. Still, none focused on when the banana bunch should be harvested from the field on an industrial scale, which is one of the major concerns of banana-based industries.

Harvesting banana bunch at the right maturity age is crucial because if it is cut before the proper aging time, it might not be appropriately matured in the greenhouse or may not have adequate nutrition. If cut after, it might be prone to disease, and shipping might not be feasible as it can rote during transportation [10]. Therefore, the main objective of this paper is to classify banana bunches for harvesting based on their right maturity stage using deep learning approaches, leading to a less error-prone, non-destructive approach.

The following is an outline of this research work structure: section 2 describes the dataset collection, how we classified banana bunch harvests, and the criteria used to evaluate our findings. The experimental results and conclusions we have drawn from this study are presented in section 3. The interpretability features of the machine learning model we have used are covered in detail in section 4. Section 5 concludes the debate by providing a finding of the study, exploring possible avenues for future research, and outlining the inherent limitations of the current study.

2. Methodology

The proposal started with data collection from different sites multiple times. Annotation of the data created after consulting with the expert team. Then, different data preprocessing processes, such as data cropping, resizing, and argumentation, were done to prepare the data for machine learning models. In addition to that, different evaluation metrics were used for training and testing the system properly. Figure 1 shows the flow of the proposed system, composed of three main components: (1) Dataset Creation; (2) Data Pre-processing; (3) Machine Learning Models; (4) Performance evaluation.

2.1. Dataset creation

A total of four fields were chosen for this study, and pictures were captured with mobile phones (Samsung Galaxy A12, Samsung Galaxy Note 9, and OnePlus 9) in multiple environmental conditions throughout the year to represent the practical situation better and make the system more robust and able to generalize.

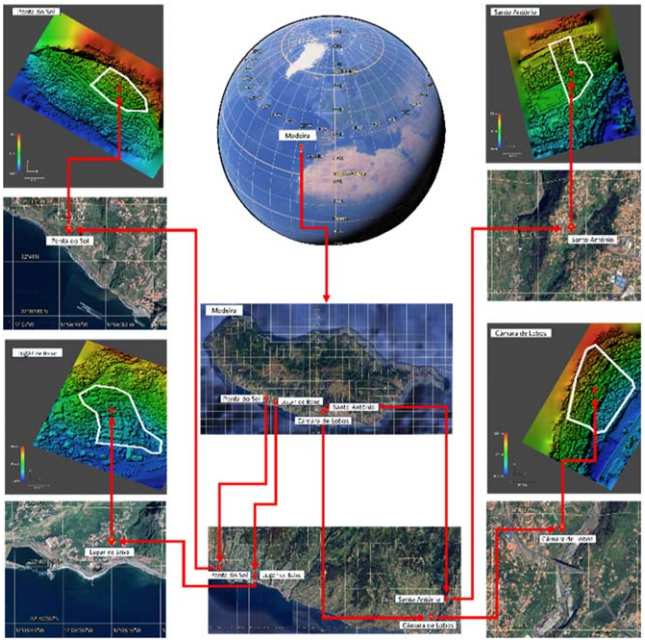

Figure 2. Combination of Google Earth and drone images of the field. The marked area represents the exact location of the field.

Table 1. Description of the produced dataset.

Fields name	Area thousands m^2 [a]	Altitude m [b]	Num of visits	Num of images	
				Cut	Keep
Santo António	1	297–308	4	20	177
Ponta do Sol	4	65–110	9	148	330
Lugar de Baixo	5	21–42	13	581	655
Câmara de Lobos	9	69–103	3	394	380
Total	19	21–308	29	1,143	1,542

[a] Approximate.
[b] Approximate from sea level.

The fields were chosen to represent conditions at four different altitudes from 21 to 308 m from sea level and are designated according to the location: Santo António; Câmara de Lobos; Ponta do Sol; and Lugar de Baixo, which cover 19 thousand m^2 area. Figure 2 shows the fields by a drone and Google Earth.

A summarization of the data collection is shown in table 1. A dataset of 2,685 images of banana bunches was collected along with the banana harvesting experts' team classification in Madeira Island, Portugal. The research team visited the fields with the harvesting expert team 29 times, taking pictures of all bunches before they were harvested and later (on the same day) taking photographs of non-harvested bunches.

All images were collected in the RGB color space and in the Joint Photographic Experts Group (JPEG) format. Among the collected images, 1,143 were harvested or 'Cut' images, whereas 1,542 images of unharvested bananas were indicated as 'Keep'. To differentiate between pictures, a number plate system was used with 'X' and 'A' for 'Cut' and 'Keep', respectively. A sample of harvested and non-harvested pictures indicated by 'Cut' and 'Keep' are shown in figure 3. The data was made available online as a Mendeley dataset [14].

2.2. Data pre-processing

Data preprocessing is one of the main steps in the machine learning process. In this work, an Automatic Data Cropping technique is developed for processing a large number of images. The dataset is divided into three subsets, with 70% of data (1,879 images) used for training, 15% (403 images) for testing, and 15% for validation employed for the early stopping criteria. Later, data resizing and normalization with argumentation were done for the machine learning process.

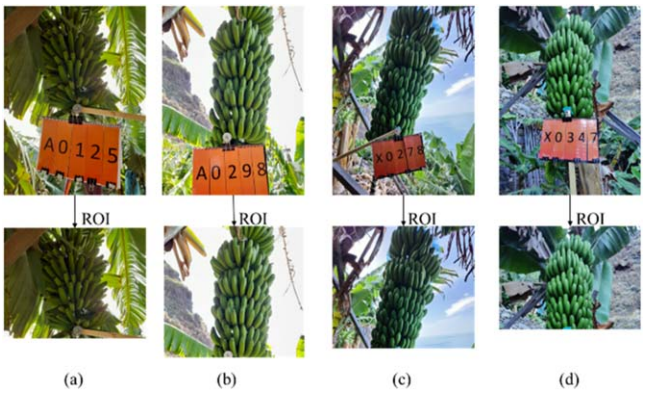

Figure 3. Sample images with numbering plate and after cropping the region of interest (ROI) (a) Cut, (b) Cut, (c) Keep, and (d) Keep.

2.2.1. Automatic data cropping

From the sample images from figure 3, it is visible that the number plate is evident in the collected pictures. If the model is trained on these images directly, instead of learning the features of the banana, in the worst-case scenario, it learns the numbering system 'X' and 'A'. In addition, in the case of data collection in the fields, it was not always possible to keep the banana brunch in focus. To solve these problems, an automatic region of interest with the transformation of the annotation from images to the data set is developed. This was done in a three-step process.

Initially, the system identifies number plates within the images based on color information. Subsequently, it employs the Easy Optical Character Recognition (EasyOCR) [15] method, leveraging the Character-Region Awareness For Text (CRAFT) [16] algorithm for character detection and a Convolutional Recurrent Neural Network (CRNN) [17] for recognition. This step enables the precise localization and extraction of characters from the number plates.

Finally, utilizing the known positioning of the number plates at the bottom part of the banana bunch during data collection, the system employs the detected characters to automatically locate and crop the image, preserving only the region of interest the banana bunch.

By implementing this methodology, the dataset is enriched with accurately annotated images, ensuring that only relevant portions containing the banana bunch are retained for further analysis and training. This way, it was possible to only keep the region of interest (ROI) banana brunch in the images (figure 3).

2.2.2. Data resizing and normalization

All the images were resized to 256 $\times$ 256 pixels to be suitable for the input of the pre-trained deep learning model. Furthermore, the resulting pictures were normalized using the min-max method Field [18] to reduce the environmental-related effects. Subsequently, the intensity value of all images was normalized in the 0 to 1 range.

2.2.3. Data augmentation

The collected image data were augmented to mimic real-world issues such as camera angle when the pictures were taken, zoom range of 0.2 used to accommodate the change in the distance when the picture was taken, horizontal flip compensates the mirror image of the banana bunch, and width shift range of 0.2 shifts the image horizontally as in some picture's banana bunch is not in the center of the image. These augmentations increase the diversity and are used to build a more generalized and robust system [18].

2.3. Comparative study of CNN architectures

This research aims to identify the optimal harvesting point of banana bunches by examining images. Finding image-based features manually is usually a cumbersome task that can lead to suboptimal results [19]. The deep learning-based models are suitable for this task as they automatically detect relevant features. However, training the deep learning-based models requires finding the optimum model structure, which requires a huge computation power and a lot of data.

It is well established in ML that Transfer Learning (TL) is suitable for transferring the knowledge of one problem to another. It reduces the training time as well as the requirement of the amount of data. Therefore, six ML architectures named DenseNet121, VGG19, MobileNetV2, Inception-V3, Xception, and NASNetLarge were used for TL in this research.

In this study, Transfer Learning (TL) was utilized to repurpose a pre-trained machine learning model for the task of detecting the harvestability of banana bunches. TL involves two stages: freezing and fine-tuning the network. During the freezing stage, the pre-existing weights and parameters of the model were kept unchanged, as they had been trained on a large image dataset [20]. In the fine-tuning stage, the fully connected layer of the model was substituted with two fully connected layers, each with two neurons at the output layer, to classify whether the banana bunch should be harvested ('Cut') or not ('Keep').

Model training was performed on the training set created after the data augmentation technique. It is important to note that the test data was not augmented and that no image was ever selected to be in both train and test data. Each batch contained 32 images in the training process. The Adaptive Moment (Adam) algorithm [21] optimized the network with a learning rate of 0.001. The model was set for 100 epochs in training with an early stopping criterion (with patience of 20), which was used to overcome the overfitting problem and help to generalize the model better. Since the dataset collected in this study was imbalanced, the default binary threshold of 0.5 can result in suboptimal performance. Therefore, the threshold tuning [22] was carried out to improve the system's performance. Threshold tuning was performed on the validation dataset, giving the best threshold value for classification instead of the default 0.5 value.

A brief description of ML models used in this work is followed:

2.3.1. DenseNet121

In DenseNet121 , an input image is processed through several convolutional layers in the conventional version of CNN to obtain high-level features [21]. Unlike other popular pre-trained CNN networks like VGG [23] and ResNet [24], DenseNet uses dense connections where each layer receives an extra input feature map from all the previous levels. Since each layer gets a feature map from the preceding layers, the network reuses those feature maps, resulting in less computation time. Furthermore, the DenseNet architecture was designed to address the rampant vanishing gradient problem in the deep learning [21].

The DenseNet architecture comprises DenseBlocks and transition layers. It uses four DenseBlock, each consisting of 1×1 convolution followed by 3×3 convolution layers. The feature maps from the previous layers are densely coupled to achieve a more complex feature map, increasing the system's efficiency. The Transition layer consists of batch normalization followed by the Rectified Linear Unit (ReLU) activation function, followed by 1×1 convolution and by the average pooling layer that will downsample the network to stop it from becoming too large due to dense connection.

2.3.2. VGG19

The Simonyan and Zisserman-developed VGG-19 is a crucial convolutional neural network and a key member of the larger VGG family [23, 25]. With a layered design alternating between multiple convolutional layers and non-linear activation layers, VGG-19 outperforms single-convolution methods regarding network depth compared to standard convolutional neural networks [26]. VGG-19, a deep learning neural network, has 19 interconnection layers, including 16 convolutional layers for extracting important features from input images and 3 fully connected layers for classifying images using these features. Max pooling layers are carefully integrated [25] to decrease feature dimensionality and lessen the danger of overfitting.

The ImageNet dataset, a sizable collection of one million images arranged into 1000 classifications, trains the VGG-19. It uses several 3×3 filters efficiently in each convolutional layer with an input size of $224 \times 224 \times 3$, so it is effective at classifying images [27].

2.3.3. MobileNetV2

The foundation of MobileNetV2 is built upon that of MobileNetV1 [28]. The depth-wise separable convolution concept was first developed for MobileNetV1. It divides convolution into two independent tasks: a depth-wise convolution for input filtering and a point-wise convolution (1×1) to combine these filtered values into new features, and 13 depth-wise separable convolution blocks were placed below a standard (3×3) convolution layer in the original MobileNetV1 architecture [29].

For mobility, Depth-wise Separable Convolution (DSC) is still used in MobileNetV2. Linear bottlenecks are used to overcome the problem of information deterioration in non-linear layers within convolution blocks, and a unique structure known as 'inverted residuals' is also introduced to retain information [28]. The network's gradients can move more easily thanks to the integration of residual connections. Batch normalization and ReLu6 activation layers come after each convolution layer; ReLu6 is a variation of the ReLu activation function with a maximum value of 6 and 17, and these bottleneck blocks make up the entire MobileNetV2 design, which is followed by standard (1×1) convolutions of an average pooling layer and the final classification layer [29, 30].

The parameter counts and computational cost of MobileNetV2 are dramatically reduced to about 18% of the standard convolutions [28] by retaining the use of DSC from MobileNetV1 together with depth-wise and

pointwise convolutions. This design is preferred because it is straightforward, has good memory efficiency, and is primarily suited for mobile applications [30].

2.3.4. Inception-V3

A popular convolutional neural network architecture in the field of image identification is represented by the Inception-V3 model [31]. It is a pre-trained model created by Google that was trained using a sizable dataset of more than 1.4 million photos divided into more than 1,000 different classes. The input image size for Inception-V3 is set at 299 × 299 pixels, with the main goal of developing the model to address computing efficiency and parameter limits while implementing it in practical applications.

However, Inception-V3 outperforms VGGNet in terms of processing time despite this input size being 78% larger than the 244 × 244 pixels employed in the VGGNet design. Convolutional layers, average pooling, max pooling, concatenation operations, dropout layers, and fully linked layers are some of the building components that make up the Inception-V3 model. The activation inputs are heavily subjected to batch normalization throughout the model, which promotes more effective training. The Softmax function is used in Inception-V3 loss computation.

Using convolutional kernels of various sizes by Inception-V3 gives it a unique feature that allows the model to create receptive fields with various spatial extents. Intending to streamline the network architecture, Inception-V3 uses a modular strategy that is later consolidated, making it possible to combine features at various scales.

Between the auxiliary classifier and the fully connected layer in Inception-V3, a Batch Normalization (BN) layer has been deliberately introduced as a regular component [32]. Due to BN's compatibility with batch gradient descent optimization techniques, this addition of BN improves deep neural network training time and encourages model convergence. In addition, Inception-V3 uses a method that sequentially divides big convolutional kernels into smaller ones.

Label smoothing regularization is also added to guarantee smoother model training, along with convolution and pooling procedures being combined in parallel. Inception-V3 adds BN to address the problem of distribution inconsistency between inputs and outputs in standard deep neural networks. This normalization method optimizes the learning process by ensuring that the inputs to each layer are scaled correctly [32, 33].

2.3.5. Xception

The Inception-V3 model served as the basis for the Xception architecture, a notable innovation in convolutional neural networks that added depth-wise separable convolution layers with residual connections to improve computational efficiency and feature representation. By separating the learning channel-wise and spatial characteristics, the depth-wise separable convolution is essential to Xception. Additionally, by establishing shortcuts inside the sequential network, adding residual connections addresses the issues related to vanishing gradients and representational bottlenecks [34].

Consider the Inception-V3 technique to clarify the architectural progress. It first processes cross-channel correlations using 1 × 1 convolution kernels, then separates 3 × 3 convolution kernels to record spatial correlations on each 1 × 1 convolution output channel.

Traditional convolutions, in contrast, regard each input channel as a separate segment, whereas depth-separable convolutions do the opposite. Hundreds of channels are often split into three or four segments by inception modules. To construct a separable convolution, the 'extreme' inception module in Xception, on the other hand, adopts a novel strategy by treating all input channels as a single segment. An additional mechanism for linear residual connections is introduced by Xception [35].

Given these developments, the Inception module can be replaced by the depth-wise separable convolution, providing a chance to improve the Inception series architecture by building models using stacked depth-wise separable layers [34]. This change enhances feature learning capability and computational efficiency in the Xception model.

2.3.6. NASNetLarge

NASANet, also known as the Google-launched Neural Architecture Search Network-Large (NASNetLarge), pioneered the use of Reinforcement Learning (RL) to address the CNN architecture design difficulty. The main goal was to determine the best combination of filter sizes, output channels, strides, layer counts, and other architectural parameters within a predetermined search area. A convolutional neural network named NASNetLarge was trained on a sizable dataset comprising more than a million photos from the ImageNet collection. With its accurate classification of photos into 1,000 different object categories, including commonplace items like pencils and keyboards, as well as a wide variety of animals, this network has astounding capabilities. It should be noted that the network uses photos that have been reduced in size to 331 × 331 pixels, creating rich feature representations [36].

The fully linked layer from the initial pre-trained network must be removed as part of the main feature extraction procedure in NASNet. NASNet differs from traditional CNNs by using blocks or cells, despite sharing many components with them. Notably, these cells are found using a reinforcement learning-based search method rather than being predefined by the authors.

Normal and reduction cells comprise the two types of cells that make up NASNet. Reduction cells are convolutional cells that produce feature maps with a two-fold reduction in both height and width as opposed to normal cells, which produce feature maps with the same dimensions [37]. With the help of this innovative architecture optimization method, a convolutional neural network can perform various image recognition tasks.

2.4. Evaluation metrics

Five standard matrices were used to evaluate the proposed architecture's performance: accuracy; precision; recall; F1 score; specificity. All of these matrices were calculated using the four standard parameters: True Positive (T_p); False Positive (F_p); True Negative (T_n); False Negative (F_n). These metrics can be represented by

$$Accuracy\,(Acc) = (T_p)/(T_p + F_p + T_n + F_n) \tag{1}$$

$$Precision(Pre) = (T_p)/(T_p + F_p) \tag{2}$$

$$Recall(Rec) = (T_p)/(T_p + F_n) \tag{3}$$

$$F1\ score = 2 \times (Pre \times Rec)/(Pre + Rec) \tag{4}$$

$$\text{Specificity}(Spe) = T_n/(T_n + F_p) \tag{5}$$

3. Result and discussion

This study uses six deep learning state-of-the-art deep learning models, namely DenseNet121 [21] , VGG19 [23], MobileNetV2 [38], InceptionV3 [33], Xception [39], and NASNetLarge [40]. The obtained results are presented in figure 4 and table 2. The optimal result would have only T_p and T_n. It was observed that MobileNetV2 and InceptionV3 reached the utmost T_p but struggled with more with F_n values, especially the InceptionV3 model. The DenseNet121 maximized the T_p and T_n values while minimizing the F_p and F_n values, leading to the best performance. These observations are further stressed when examining the results shown in table 2.

Though there is no exact work in the literature to compare with the propsed research. State of artwork such as Saranya *et al* [41] proposed a Convolutional Neural Network (CNN) that can grade the banana into four ripeness stages. The proposed architecture was also compared with state-of-the-art deep CNN networks, Visual Geometry Group16 (VGG16), and Residual Network50 (ResNet50), achieving an utmost accuracy of 96.1%, where this work archives 83.3%. However, a limitation of this work is that a small dataset (273 images) was used, which can lead to poor assessment regarding the generalization potential of the model.

Saragih and Emanuel, [8] used two pre-trained CNN networks based on MobileNetV2 and Neural Search Architecture NetMobile (NASNetMobile) architectures. A bilateral filter for noise removal was employed, and it was concluded that MobileNetV2 is the solution that gives higher accuracy (96.2%) with 436 images. However, this work focused on single banana fingers and based on color, which is not suitable for industrial solutions. Similar issue also occurs in CNN networks based on the VGG16 architecture. Ramadhan *et al* [42] classify the banana into four stages with an accuracy of 72.0%.

Zhu *et al* [43] created a dataset containing 150 images with three classes (unripened, ripened, and over-ripened) based on maturity. The Cycle Generative Adversarial Network (CycleGAN) was used to enlarge the dataset. Rank filtering and log transformation were used to remove noise and improve contrast. Afterward, a Support Vector Machine (SVM) was used for classification and achieved an accuracy of 98.5%. The ripened class was further divided into two categories (mid-ripened and well-ripened) using You Only Look Once v3 (YOLOv3) with an accuracy of 85.7%.

Marimuthu *et al* [44] classified the banana hand (cluster banana) instead of the single banana in three stages: unripe, ripe, and overripe, which is likely more practical with real-time applicability. A Particle Swarm Optimized (PSO) fuzzy model was used for classification, and an accuracy of 93.1% was attained.

Mohamedon *et al* [45] created a mobile application using CNN based on the selected EfficientNet-Lite model because it presents low computational time with high accuracy, making it feasible for mobile applications. A dataset consisting of 571 samples were categorized into three classes, and the proposed model attained an accuracy of 98.3%.

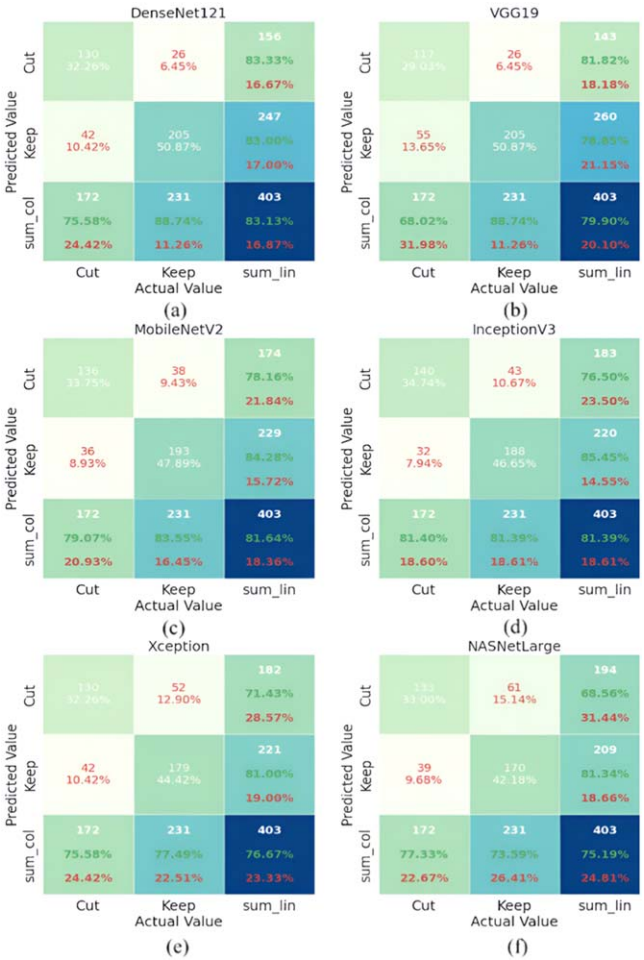

Figure 4. Confusion matrix reached using (a) DenseNet121, (b) VGG19, (c) MobileNetV2, (d) InceptionV3, (e) Xception, and (f) NASNetLarge.

Table 2. Performance metrics in percentage of the examined deep learning models.

Model	Pren	Rec	F1score	Acc	Spe
VGG19	80.5	78.5	79.0	80.0	79.0
MobileNetV2	81.0	81.5	81.5	81.7	84.4
InceptionV3	81.5	81.5	81.5	81.5	85.6
Xception	76.0	77.0	76.0	76.8	81.2
NASNetLarge	75.5	75.5	75.0	75.3	81.5
DenseNet121	85.4	82.0	82.5	83.2	83.2

Though some of the previous works' accuracy was considerably high, these works defined ripe and non-ripe bananas, while none at the commercial level, where, as previously discussed, banana is harvested much earlier than the ripeness level of bananas (due to the transport and storage of the banana distribution chain line). In the commercial case, most of the time, the banana is green, with no color change compared to normal ripeness, where color changes due to the ripeness, which makes the proposed work more difficult than the state of art. Therefore, it is not feasible to directly compare this work with any other state-of-the-art works.

DenseNet outperforms other pretrained models for banana harvesting due to its unique architecture that fosters dense connections between layers. This facilitates feature reuse and enhances gradient flow, enabling more effective information propagation throughout the network. Additionally, DenseNet's dense connectivity reduces the risk of vanishing gradients, thereby mitigating the challenges associated with training deep neural networks. These advantages culminate in improved model performance, robustness, and generalization capabilities, making DenseNet an optimal choice for the demanding task of banana bunch classification.

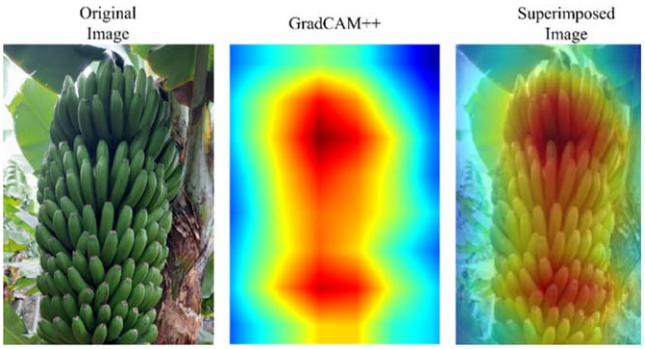

Figure 5. Visualization of proposed system for banana harvesting.

Figure 6. Predictions of the best performing in (a) T_p, (b) T_n, (c) F_p, and (d) F_n samples.

4. Explainability of ML

One of the main limitations of deep learning architectures was their black box characteristics [43]. However, to further improve the models, it is important to understand why they behave in a particular way and which aspects lead to the result. A common practice is performing a visual explainability, for example, where it is possible to visualize which section of the image the deep learning architecture focuses more on. For this purpose, the Gradient Weighted Class Activation Mapping++ (GradCAM++) [44] method was used to examine the best-performing model. Figure 5 highlights the region of interest (where the model responds more). For better visualization, it also shows the superimposed image.

Astonishingly, the model focuses on the banana bunch in figure 5, especially on the top part of the bunch. Such suggests that these regions are the most significant in the analysis and agree with the feedback provided by the specialists, as they focus more on the top region of the banana bunch for making harvesting decisions. Most of the decisions related to fruit harvesting were taken by visual inspection and after seeing the excellent results shown using GradCAM++, suggesting that the DenseNet121 model can be used for fruit harvesting.

Another relevant examination is to understand which characteristics lead the model towards misclassifications. Figure 6 shows the forecasts of the best-performing model in four conditions: correctly predicted 'Cut'; correctly predicted 'Keep'; misclassified 'Cut'; misclassified 'Keep'. After examining the

misclassified samples and backtracking with the experts, it was concluded that these occurred when the harvesting decision was strongly affected by external factors in bordering samples.

The most substantial factor involves the weather conditions, as in bothering samples, the decision was based on when the cutting team was coming back to the field and the weather forecast for these days. In most cases, it was decided to cut the bunch if it was expected for it to be too ripped in the subsequent cutting operation (which could be in several weeks). These samples lead to an F_p. The converse happened when the weather dictated that the bunch would be ready for harvesting in the subsequent visit, leading to F_n samples. Therefore, adding weather-based information could further improve the performance of the model.

5. Conclusion

The agriculture industry needs an automatic banana bunch harvestability appraisal system. The significance of such a system emerges from its critical requirement in the agriculture industry, which is experiencing tremendous demand at a rapid pace. Such a system should be determined to be the most efficient, dependable, accurate, and flexible system for the agriculture sector.

This article proposed a deep learning-based automatic system for banana bunch harvestability analysis based only on the visual appearance of the banana bunch. Various state-of-the-art deep learning-based models were studied and evaluated on a created banana bunch dataset. The results indicate that the DenseNet-based model outperforms all the other deep models. Furthermore, the proposed study has shown good performance with an accuracy higher than 80%, only struggling in the borderline samples.

In the future, the system can be built to include external factors such as weather conditions. Furthermore, the model can be trained on a larger dataset for more generalized and robust results, which can be used for various applications like robot-based automatic banana bunch harvesting.

Acknowledgments

Acknowledgment to the Bolsa de Investigação (BI) within Project BASE: BAnana Sensing (PRODERAM20-16.2.2-FEADER-1810).

Bolsa de Investigação (BI) within Project PRR (TD- C16-i03-SIH).

ITI/Larsys - Funded by FCT (Fundação da Ciência e da Tecnologia) projects: 10.54499/LA/P/0083/2020; 10.54499/UIDP/50009/2020 & 10.54499/UIDB/50009/2020.

Acknowledgment to ARDITI—Agência Regional para o Desenvolvimento da Investigação, Tecnologia e Inovação under the scope of the project M1420-09-5369- FSE-000002 Post-Doctoral Fellowship, co-financed by the Madeira 14–20 Program - European Social Fund.

Conflicts of interest

The authors declare no conflict of interest.

Data availability statement

This dataset used under this study can be made publicly available under Open Data Commons Attribution License v1.0 via a Mendeley data repository [14].

ORCID iDs

Ahatsham Hayat https://orcid.org/0000-0003-2000-5557
Preety Baglat https://orcid.org/0000-0002-3348-262X
Fábio Mendonça https://orcid.org/0000-0002-5107-3248
Sheikh Shanawaz Mostafa https://orcid.org/0000-0002-7677-0971
Fernando Morgado-Dias https://orcid.org/0000-0001-7334-3993

References

[1] Trienekens J and Zuurbier P 2008 Quality and safety standards in the food industry, developments and challenges *Int. J. Prod. Econ.* **113** 107–22

[2] Bac C W, Henten E J V, Hemming J and Edan Y 2014 Harvesting robots for high-value crops: state-of-the-art review and challenges ahead *J. Field Robot.* **31** 888–911

[3] Helwan A, Ma'Aitah M K S, Abiyev R H, Uzelaltinbulat S and Sonyel B 2021 Deep learning based on residual networks for automatic sorting of bananas *J. Food Qual.* **2021** 1–11
[4] Ismail N and Malik O A 2022 Real-time visual inspection system for grading fruits using computer vision and deep learning techniques *Inf. Process. Agric.* **9** 24–37
[5] Sa I, Ge Z, Dayoub F, Upcroft B, Perez T and McCool C 2016 Deepfruits: a fruit detection system using deep neural networks *Sens. Switz.* **16** 1222
[6] Kelman E (E and Linker R 2014 Vision-based localisation of mature apples in tree images using convexity *Biosyst. Eng.* **118** 174–85
[7] Ferrer-Ferrer M, Ruiz-Hidalgo J, Gregorio E, Vilaplana V, Morros J-R and Gené-Mola J 2023 Simultaneous fruit detection and size estimation using multitask deep neural networks *Biosyst. Eng.* **233** 63–75
[8] Saragih R E and Emanuel A W R 2021 Banana ripeness classification based on deep learning using convolutional neural network *3rd 2021 East Indones. Conf. Comput. Inf. Technol. EIConCIT 2021* 85–9
[9] Dewi C, Mahmudy W F, Arisoesilaningsih E and Solimun S 2021 Review of non-destructive banana ripeness identification using imagery data *ACM Int. Conf. Proc. Ser.* 348–54
[10] Dadzie B K and Orchard J E 1997 'Routine post-harvest screening of banana/plantain hybrids: criteria and methods' *INIBAP Tech. Guidel. 2 Rome Intl Plant Genet. Resour. Inst* 5–30
[11] Altaf S, Ahmad S, Zaindin M and Soomro M W 2020 Xbee-based WSN architecture for monitoring of banana ripening process using knowledge-level artificial intelligent technique *Sensors* **20** 4033
[12] Zhang Y, Lian J, Fan M and Zheng Y 2018 Deep indicator for fine-grained classification of banana's ripening stages *Eurasip J. Image Video Process.* **2018** 1–10
[13] Baglat P, Hayat A, Mendonça F, Gupta A, Mostafa S S and Morgado-Dias F 2023 Non-destructive banana ripeness detection using shallow and deep learning: a systematic review *Sensors* **23** 738
[14] Hayat A, Baglat P, Mendonça F, Mostafa S S and Morgado-Dias F 2024 Banana bunch harvesting dataset *Mendeley Data* **V1**
[15] Jaided AI. (n.d.). EasyOCR. GitHub repository. Accessed: 28-03-2023, from https://github.com/JaidedAI/EasyOCR
[16] Baek Y, Lee B, Han D, Yun S and Lee H 2019 Character region awareness for text detection *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.* 2019-June, 9357–66
[17] Shi B, Bai X and Yao C 2015 An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition *IEEE Trans. Pattern Anal. Mach. Intell.* **39** 2298–304
[18] Shorten C and Khoshgoftaar T M 2019 A survey on image data augmentation for deep learning *J. Big Data* **6** 1–48
[19] Le T T, Lin C Y and Piedad E J 2019 Deep learning for noninvasive classification of clustered horticultural crops—A case for banana fruit tiers *Postharvest Biol. Technol.* **156** 110922
[20] Russakovsky O *et al* 2015 ImageNet large scale visual recognition challenge *Int. J. Comput. Vis.* **115** 211–52
[21] Huang G, Liu Z, Maaten L V D and Weinberger K Q 2016 Densely connected convolutional networks *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognit. CVPR 2017* 2017-January, 2261–9
[22] Coenen F, Leng P and Zhang L 2005 Threshold tuning for improved classification association rule mining *Lect. Notes Comput. Sci. Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinforma.* **3518 LNAI** 216–25
[23] Simonyan K and Zisserman A 2015 Very deep convolutional networks for large-scale image recognition *3rd Int. Conf. on Learning Representations (ICLR 2015)* (San Diego) Accessed: 11, 2023. [Online]. Available: http://arxiv.org/abs/1409.1556
[24] He K, Zhang X, Ren S and Sun J 2015 Deep residual learning for image recognition *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.* 2016-December, 770–8
[25] Nguyen T-H, Nguyen T-N and Ngo B-V 2020 AgriEngineering | free full-text | a VGG-19 model with transfer learning and image segmentation for classification of tomato leaf disease *AgriEngineering* **4** 871–87
[26] Xiao J, Wang J, Cao S and Li B 2020 Application of a novel and improved VGG-19 network in the detection of workers wearing masks *J. Phys. Conf. Ser.* **1518** 012041
[27] Bansal M, Kumar M, Sachdeva M and Mittal A 2023 Transfer learning for image classification using VGG19: Caltech-101 image data set *J. Ambient Intell. Humaniz. Comput.* **14** 3609–20
[28] Dong K, Zhou C, Ruan Y and Li Y 2020 MobileNetV2 model for image classification *2020 2nd Int. Conf. on Information Technology and Computer Application (ITCA)* 476–80
[29] Shahi T B, Sitaula C, Neupane A and Guo W 2022 Fruit classification using attention-based MobileNetV2 for industrial applications *PLoS One* **17** e0264586
[30] Gulzar Y 2023 Fruit image classification model based on mobileNetV2 with deep transfer learning technique *Sustainability* **15** 1906
[31] Nikhitha M, Roopa Sri S and Uma Maheswari B 2019 Fruit recognition and grade of disease detection using inception V3 model *2019 3rd Int. conference on Electronics, Communication and Aerospace Technology (ICECA)* (Coimbatore, India: IEEE) 1040–3
[32] Joshi K, Tripathi V, Bose C and Bhardwaj C 2020 Robust sports image classification using INCEPTIONV3 and neural networks *Procedia Comput. Sci.* **167** 2374–81
[33] Szegedy C, Vanhoucke V, Ioffe S, Shlens J and Wojna Z 2015 Rethinking the Inception Architecture for Computer Vision *2016 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*
[34] Wu X, Liu R, Yang H and Chen Z 2020 An xception based convolutional neural network for scene image classification with transfer learning *2020 2nd Int. Conf. on Information Technology and Computer Application (ITCA)* (Guangzhou, China: IEEE) 262–7
[35] Tan Z, Hu Y, Luo D, Hu M and Liu K 2020 The clothing image classification algorithm based on the improved Xception model *Int. J. Comput. Sci. Eng.* **23** 214–30
[36] Mehmood M *et al* 2022 Improved colorization and classification of intracranial tumor expanse in MRI images via hybrid scheme of Pix2Pix-cGANs and NASNet-large *J. King Saud Univ. - Comput. Inf. Sci.* **34** 4358–74
[37] Dahouda M K and Joe I 2022 Neural architecture search net-based feature extraction with modular neural network for image classification of copper/ cobalt raw minerals *IEEE Access* **10** 72253–62
[38] Sandler M, Howard A, Zhu M, Zhmoginov A and Chen L C 2018 MobileNetV2: inverted residuals and linear bottlenecks *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.* 4510–20
[39] Chollet F 2016 Xception: deep learning with depthwise separable convolutions *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognit. CVPR 2017* 2017-January, 1800–7
[40] Zoph B, Vasudevan V, Shlens J and Le Q V 2017 Learning transferable architectures for scalable image recognition *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.* 8697–710
[41] Saranya N, Srinivasan K and Kumar S K P 2021 Banana ripeness stage identification: a deep learning approach *J. Ambient Intell. Humaniz. Comput. 2021 138* **13** 4033–9

[42] Ramadhan Y A, Djamal E C, Kasyidi F and Bon A T 2020 'Identification of cavendish banana maturity using convolutional neural networks *Int. Conf. on Industrial Engineering and Operations Management* (Detroit, Michigan, USA)
[43] Zhu L and Spachos P 2020 Food grading system using support vector machine and YOLOv3 methods *Proc. -IEEE Symp. Comput. Commun.* pp1–6,
[44] Marimuthu S and Roomi S M M 2017 Particle swarm optimized fuzzy model for the classification of banana ripeness *undefined* **17** 4903–15
[45] Mohamedon M F, Rahman F A, Mohamad S Y and Khalifa O O 2021 Banana ripeness classification using computer vision-based mobile application *Proc. 8th Int. Conf. Comput. Commun. Eng. ICCCE 2021* 335–8