International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

Recognizing Mangoes and Determining their Ripeness Through the Application of Image Processing and Machine Learning Techniques

Ashwini B. Gavali[1], Somnath B. Thigale[2], Pradeep S. Togrikar [3], Swagat M. Karve[4], Swapnaja A. Ubale[5], Sonali B. Gavali[6]

Submitted: 10/01/2024 **Revised**: 16/02/2024 **Accepted**: 24/02/2024

Abstract: The fruit market faces challenges in grading, with existing commercial systems being prohibitively expensive. In contrast, smaller businesses often rely on manual grading systems, which are susceptible to human errors and inaccuracies. This research introduces an innovative approach to identify and grade Mango, aligning with the principles of Industry 4.0. Employing a Faster Region-based Convolution Neural Network (Faster R-CNN) object detection algorithm through Tensor Flow, the system efficiently detects the fruit and utilizes image processing to assess the likely percentage of ripeness. This allows for the categorization of the fruit into specific classes. The study demonstrates that the proposed methods are not only effective but also cost-efficient for accurately determining fruit ripeness. Moreover, with effective training, the same system can be adapted for multiple fruits, showcasing its versatility and applicability across various produce.

Keywords: *Ripeness, Machine Learning, Image Processing, Cost-efficient, Object detection algorithm, Faster R-CNN*

1. Introduction

Agriculture plays a crucial role in ensuring national economic stability, employing two-thirds of the population in India. However, technological advancements in agriculture have lagged behind other sectors like technology. Currently, fruit grading, a vital aspect of the fruit market, is carried out manually by human experts, leading to potential errors and inefficiencies due to human fatigue and subjectivity. Commercially available grading systems are expensive, prompting the need for a more cost-effective and efficient solution.

To address this, there is a proposal to introduce a technology-driven fruit grading system that leverages machine learning stated by fadillah et al. [1]. This system aims to enhance efficiency, reduce human errors, and contribute to the overall growth of the economy. The demand in the fruit market is closely tied to the quality of the product, and the proposed system can ensure accurate grading based on factors such as ripeness, shape, and size.

[1]*Assistant Professor, Computer Engineering Department, S. B. Patil College of Engineering, Indapur (MH), India,dnyane.ash@gmail.com*
[2]*Associate Professor, Computer Science & Engineering (AIDS) Department, Shree Siddheshwar Women's College of Engineering, Solapur (MH), India, drsomnaththigale@gmail.com*
[3]*Assistant Professor, E & TC Engineering Department, S. B. Patil College of Engineering, Indapur (MH), India, pradeep.togrikar@gmail.com*
[4]*Assistant Professor, E & TC Engineering Department, S. B. Patil College of Engineering, Indapur (MH), India, swagatkarve@gmail.com*
[5]*Associate Professor, Information Technology Department, Marathwada Mitra Mandal College of Engineering, Pune (MH), India, swapnaja.b.more@gmail.com*
[6]*Assistant Professor, AI & DS Department, Dr. D. Y. Patil, Institute of Technology, Pimpri, Pune, sonygavali1991@gmail.com*

Ripeness is a critical factor in fruit quality, especially for fruits like mangoes, which are highly valued in India. The proposed system would utilize machine learning algorithms to identify and determine the quality of fruits by analyzing factors such as color, a key indicator of ripeness. Machine learning has been successfully used in agriculture for trend analysis, pattern recognition, and managing large volumes of data, as highlighted by previous research.

While the current manual grading system suffers from inefficiencies, high costs, and slow performance, machine learning algorithms offer a promising solution. These algorithms can address issues related to human fatigue, subjectivity, and the need for a fast and accurate grading process. Researchers have identified multiple machine learning algorithms that can be applied to resolve these challenges.

In summary, the introduction of a machine learning-based fruit grading system has the potential to revolutionize the agricultural sector, ensuring faster, more accurate, and cost-effective fruit grading, ultimately contributing to the economic development of the country. Then, the job will be efficient and error-free. Multiple Machine Learning algorithms can be used to resolve this problem as stated by Ranjit et. al [2] and Jang et. al.[3].

2. Literature Survey

The evaluation of Mango maturity is based on the color transition from green to yellow or, in some instances, red. In the examination of tomato ripeness, Jaramillo et al. [4] identified six stages, as outlined in the table 1. Artificial intelligence is widely employed for detection, with Jana et al. [5] emphasizing edge-based segmentation methods involving image pooling, pre-processing, image segmentation, feature extraction, and Support Vector Machine (SVM) for detection.

Table 1. Stage, maturity, range comparison

Stage	Color/Maturity	Range
1	Green	>90%
2	Breaking	90% Green color; <10% otherthan Green
3	Turning	10%- 30% Yellow
4	Pink	30% - 60% Yellow or Redcolor
5	Light Red	60% - 90%
6	Red	90% - 100%

Nandi et al. [6] applied a similar approach but utilized a Charge Coupled Device to capture video on a conveyor with a Mango. Susnjak et al. [7] enhanced color evaluation by measuring quality through color segmentation within the region of interest and pixel blob analysis to identify defective fruits. Fojlaley et al. [8] focused on tomatoes, employing feature extraction based on R and G color, fruit shape, and classifying mangoes based on the first, second, and third moments.

Rismayati et al. [9] explored deep learning using Convolutional Neural Networks (CNN) for sorting salak fruits. Their method involves a region of interest for salak images and CNN classification, achieving a detection accuracy of 81.45% with a 6-filter-layer architecture. Inspired by these studies, we delved into new frameworks like TensorFlow and MobileNet, ultimately implementing the Faster R-CNN. This approach incorporates a convolutional module for region proposal Networks and sigmoid functions to enhance system speed. The expectation is improved accuracy, efficiency, and overall performance in this experimental setup.

3. Proposed Method

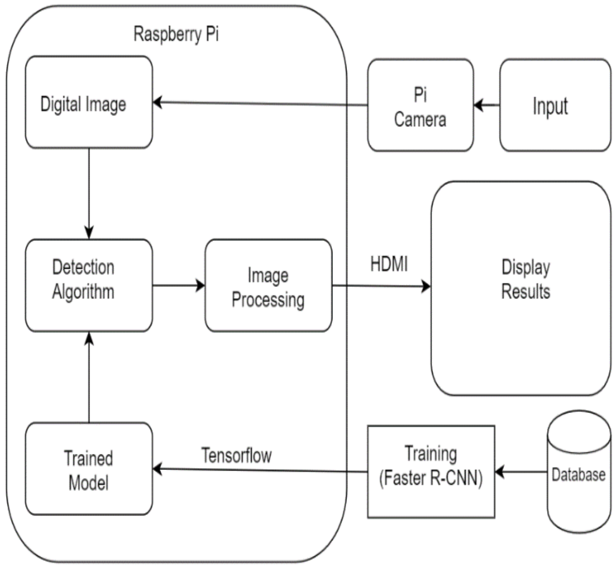

Fig 1. Proposed Architecture

In our case study, we introduce a system that utilizes a Raspberry Pi with a connected Pi camera to process images through a Faster R-CNN algorithm implemented in TensorFlow. This algorithm is specifically trained for fruit identification. The process involves capturing an image using the Pi camera, feeding it through the trained model for fruit detection, and subsequently analyzing the detected image to determine the ripeness of the identified fruit. The results are then displayed on a monitor.

The Faster R-CNN algorithm, grounded in deep learning, plays a crucial role in discerning the characteristics of the fruit, enabling it to distinguish the fruit from other objects and effectively categorize it. In our scenario, the focus is on identifying mangoes in real-time. The software developed for this purpose classifies the detected mangoes into three ripeness categories:

Class 1: High Ripeness

Class 2: Medium Ripeness

Class 3: Low Ripeness

The hardware setup involves the Raspberry Pi 3 B+ model with 512MB of inbuilt SDRAM. The operating system used is Raspbian, a Linux variant optimized for Raspberry Pi. Any HDMI monitor or TV can be connected as a display for the system. Figure 2 illustrates the Raspberry Pi 3 B+ model used in this case [11].

Fig 2. Raspberry Pi [11]

An exemplar of a highly ripened mango is employed to illustrate the functionality of the application. Consequently, the algorithm is adept at detecting a mango and determining its state of ripeness.

The system's primary function is the detection of mangoes and the subsequent categorization of their ripeness. This involves the development and utilization of a softmax classifier. The dataset used for training and testing the model comprises a total of 458 images. Of these, 346 are designated for training, and the remaining 112 are allocated for testing. The images are sourced through manual capture or downloaded from Google.

The process is outlined as follows:

A. Faster R-CNN Object Detection:

The operational model of the Faster R-CNN is depicted in Figure 3. The convolution layer computes feature maps using the data conveyed through the input layer.

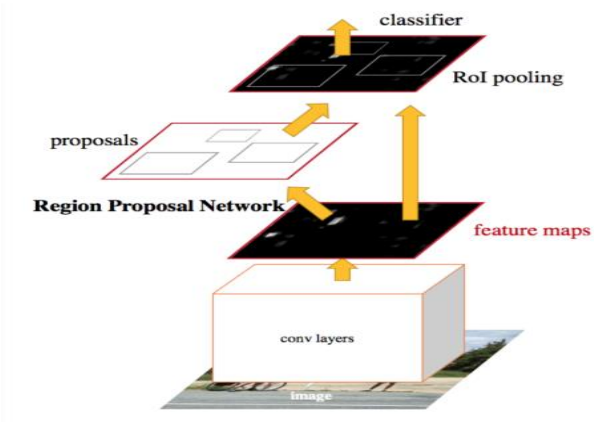

Fig 3. Proposed Faster R-CNN model Architecture

The convolution process is carried out within the convolution layer, as explained by reference [12]. The Region Proposal Network (RPN) examines images to predict groups of objects and their objectness through feature maps. During each proposal, a specific portion of these feature maps is assessed by the Region of Interest (RoI). The resulting feature vector is fed into a fully connected layer with two output layers. The proposed system verifies the presence of the fruit, and generates four real values to define the proposal's location.

B. Region of Interest (ROI) Pooling

To transform the trained features, image pooling is employed, necessitating feature selection. For Region of Interest (ROI), each image is individually labeled using a labeling application. The functionality of this application is illustrated in Figure 4.

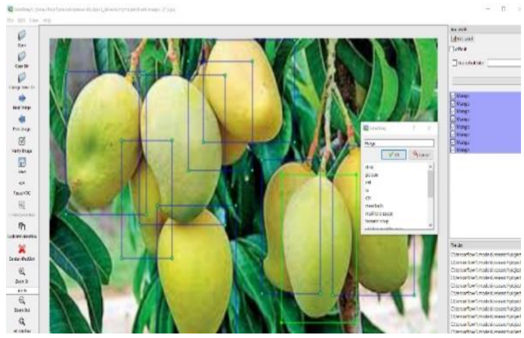

Fig 4. Labeling Image

For every image, an XML file is generated and stored in the root folder, containing information about the height and width of each label. Correspondingly, a CSV file is created, capturing fixed length displacement data for height and width. Both height and width serve as layered variables for all the images. In order to enhance data processing speed and efficiency, all the data has been converted into CSV format, as depicted in Figure 5

filename	width	height	class	xmin	ymin	xmax	ymax
Mango1.jpg	275	183	Mango	15	15	140	134
Mango2.jpg	275	183	Mango	127	42	266	154
Mango3.jpg	216	233	Mango	51	66	158	173
Mango4.jpg	59	50	Mango	1	1	58	50
Mango5.jpg	75	50	Mango	12	8	50	27

Fig 5. Labeled image csv format

C. Binary Classifier

Faster R-CNN is primarily categorized into two modules: the fully convolutional network module and the detector module. The detector module employs Softmax, where the Softmax estimator calculates the probability for the object class using

$$\boldsymbol{\sigma_i(m) = \frac{\exp(m_i)}{\sum_j^z \exp(m_j)}, i = 1, \dots, z}$$

Where:

$m = input\ vector$

$$m_i = elements\ of\ the\ input\ vector$$

$$j = \ normalization\ term$$

$$z = number\ of\ classes$$

We have implemented a two-class Softmax binary classification for fruit detection. Additionally, we incorporate the sigmoid function to enhance the model's efficiency. The sigmoid function plays a crucial role in binary classification and probability assessment. Therefore, the formula can be expressed as follows:

$$\boldsymbol{\sigma_i(m) = \frac{1}{1+e^{-m}}}$$

We employ a multi-task loss to guide the training process, wherein the multi-task loss is utilized.

$$\boldsymbol{ML = ML_{cls} + ML_{reg}}$$

Whereas**,** $\mathbf{ML_{cls}}$ represents a classification loss logged across two losses, and $\mathbf{ML_{reg}}$ signifies a regression loss over the regressor target, which traverses each pixel to predict its location. For $\mathbf{ML_{cls}}$, we opt for sigmoid entropy rather than softmax or multinomial entropy. Subsequently, a sigmoid function is employed to ascertain whether the object contains fruit or not. In cases where training data is insufficient, we consider this method to be the most suitable for fruit classification.

D. Implementation

The approach has been trained and tested on an Intel i7 processor, NVIDIA's Quadro P1000 GPU, and its cuDNN with 32GB RAM. The model utilized for training is ssd_mobilenet_v1_coco. Images are selected arbitrarily and processed in batches. The training model converts the image into an 800x1200 pixel format, followed by the execution of the Faster R-CNN algorithm. Each training step reports the loss, which initially starts high in classification loss and gradually decreases during training. Starting at approximately 2, the loss drops below 0.1. The training was conducted until the loss consistently reached below 0.05, taking approximately 70,000 steps, equivalent to around 24 hours. This progression is depicted in Figure 6 below.

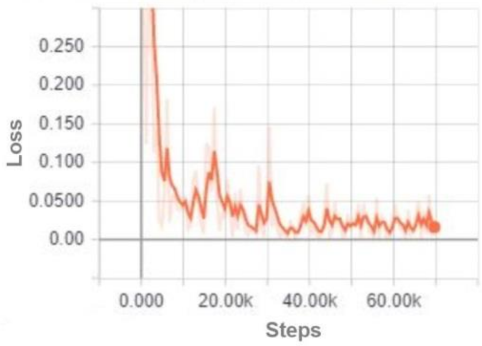

Fig 6. Classification Loss

E. Testing

Our model for Mango detection is prepared, and we executed the object detection code using Python as shown in Figure 7.

1. Input:

We utilized Faster R-CNN to construct an identification model based on a Mango dataset consisting of approximately 500 images. The illustration is demonstrated using a sample fruit image. The modeling is implemented using Python, employing MobileNet as the base, as described by Howard et al. [13], and TensorFlow library by Goldsborough et al. [14].

2. ***Training:***

The images provided in the train and test folders serve as input for the chosen model discussed earlier. Utilizing labeled images in these folders, we initiate the training process, which takes some time to generate an inference graph. This graph is subsequently used by the Python code to detect the Region of Interest (RoI) in the image.

3. Result:

Approximately 80% of the images are used for training, and the remaining 20% are reserved for testing. The training process yields a 99% accuracy rate, which is considered quite satisfactory. It's important to note that this accuracy may vary for other input images due to the arbitrary nature of the training process. Accurate labeling is crucial for achieving high accuracy, as highlighted by He et al. [15].

Fig 7. Training Result

F. Ripeness Determination

Upon finishing the detection process, the subsequent stage involves determining ripeness using image processing. This ripeness determination process primarily consists of three stages: preprocessing, feature extraction, and classification, as illustrated in Figure 8.

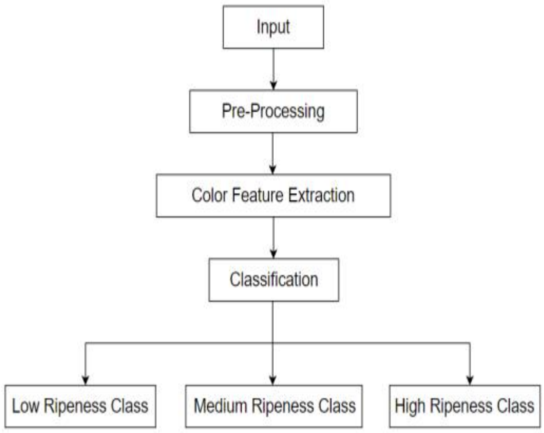

Fig 8. Flowchart for Ripeness Detection

Algorithm Outline:

1. Begin
2. Receive input image from the output of the preceding sections.
3. Convert the image into an HSV image.
4. Apply thresholding to the hue for red (R), green (G), and blue (B) colors.
5. Set saturation values to 0 and 255.
6. Extract blobs from the image.
7. Generate masks for redness, yellowness, and greenness.
8. Determine the color that is predominant.
9. Calculate and display the ripeness.
10. End

4. Result Discussion

We now proceed with implementing ripeness detection for the fruit using the input image from the previous section. Figure 9 illustrates the RGB image along with its corresponding HSV image, utilized to generate masks.

Fig 9. RGB Image (Left) and HSV Image (Right)

The HSV image is employed to create redness, yellowness, and greenness masks by specifying minimum and maximum threshold values for each color (RYG). The purpose of these masks is to transform specific colors into different colors. For example, the redness mask turns areas with red color to white, while non-red pixels are set to black based on the specified threshold. Counting the white pixels in the redness mask provides the exact number of red pixels in the image, indicating the level of mango ripeness.

Figure 10(a) displays the redness mask, showcasing areas where the color red is prominent. Similar masks are created for yellow and green colors as shown in Figure 10(b) and 10(c) respectively.

(a) (b) (c)

Fig 10. (a) Redness mask (b) Yellowness mask (c) Greenness mask

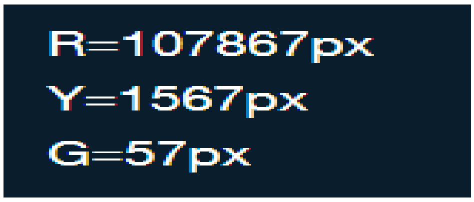

Fig 11. Pixel Count

The HSV conversion and masking aim to assess mango ripeness by counting the pixels of each color, as depicted in Figure 11.

In above figure, the red color stands out prominently, suggesting that the mango is highly ripened. Depending on the dominance of red or yellow color, mangoes can be categorized as highly ripened.

The application classifies the mango in as Class 1, indicating a high level of ripeness as shown in Figure 12.

Fig 12. Results

5. Conclusion

Furthermore, the success of this case study highlights the transformative potential of integrating cutting-edge technologies into the agricultural landscape, marking a significant stride toward precision farming in the era of Industry 4.0. The utilization of Google's MobileNet convolutional network showcases the adaptability of state-of-the-art models, paving the way for innovative solutions to longstanding challenges in the fruit industry.

The study's meticulous classification of fruits based on color serves as a robust foundation for future advancements in quality assessment methodologies.

This not only contributes to the efficiency of fruit grading processes but also lays the groundwork for the development of automated systems capable of enhancing overall agricultural productivity. As the system proves its prowess in distinguishing between fruit classes, the roadmap includes exploring its potential integration into smart farming systems. The seamless extension of this technology to diverse regional and imported fruits holds promise for streamlining agricultural practices on a global scale, promoting sustainability, and ensuring consistent high-quality produce.

Looking ahead, the research team envisions collaborative efforts with industry stakeholders to implement this technology in real-world scenarios, fostering a paradigm shift in fruit quality evaluation.

The continuous refinement of the model and its subsequent adaptation to various fruits underscore its role as a transformative tool that has the capacity to revolutionize not only quality assessments for specific varieties but for a myriad of fruits, fostering a more resilient and technologically advanced agricultural sector.

References

[1] L. Ma, S. Fadillah Umayah, S. Riyadi, C. Damarjati, and N. A. Utama, “Deep Learning Implementation using Convolutional Neural Network in Mangosteen Surface Defect Detection,” 7th IEEE International Conference on Control System, Computing and Engineering (ICCSCE 2017), 24–26 November 2017, Penang, Malaysiano. November, pp. 24–26, 2017.

[2] K. N. Ranjit, H. K. Chethan, and C. Naveena, “Identification and Classification of Fruit Diseases,”Int. Journal of Engineering Research and Application, ISSN- 2248-9622 vol. 6, no. 7, pp. 11–14, 2016.

[3] H. Jang, H. Yang, and D. Jeong, “Object Classification using CNN for Video Traffic Detection System,” International Conference21st Korea-Japan Joint Workshop on Frontiers of Computer Vision (FCV) no. 1, pp. 1–4, 2015.

[4] Jaramillo J., Rodriguez V., Guzman M., Zapata M. and Rengifo T., "Technical Manual: Good Agricultural Practices in the Production of Tomato under Protected Conditions".Food and Agriculture Organization (FAO) 2007.

[5] S. Jana and S. Basak, “Automatic Fruit Recognition from Natural Images using Color and Texture Features,” IEEE Devices Integrated Circuit, pp. 620–624, 2017.

[6] C. S. Nandi, B. Tudu, and C. Koley, “A machine vision-based maturity prediction system for sorting of harvested mangoes,” IEEE Trans. Instrum. Meas., vol. 63, no. 7, pp. 1722–1730, 2014.

[7] S. Basak, “An Improved Bag-of-Features Approach for Object Recognition from Natural Images,” Int. Journal Computer Appl. (0975 – 8887), vol. 151, no. 10, pp. 5–11, 2016.

[8] M. Fojlaley, P. A. Moghadam, and A. niaSaeed, “Tomato Classification and Sorting with machine vision using SVM, MLP, and LVQ,” Int. Jouranl. Agriculture Crop Science, vol. 4, no. 15, pp. 1083–1088, 2012.

[9] Rismiyati and S. N. Azhari, “Convolutional Neural Network implementation for image-based Salak sortation,” 2nd International Conferenceon Science and Technology Computer, ICST 2016, pp. 77–82. 2017

[10] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 1096, pp. 1137–1149, 2015.

[11] https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/

[12] Bengio, Y., Goodfellow, I. danCourville, A. “Convolutional Networks, In Deep Learning,” Book in preparation for MIT Press, pp. 199–216,2015.

[13] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications,” ArXiv,pg.1-9,Apr. 2017.

[14] P. Goldsborough, “A Tour of Tensor Flow,” ArXiv,pg.1-12,Oct. 2016.

[15] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” Arxiv.Org, pg.1-12,Dec.2015.