Fruit Ripeness Image Classification using Convolutional Neural Network

John Eliezer Cruz
College of Computing and Information Technologies
National University
Sampaloc, Manila
cruzjd4@students.national-u.edu.ph

Ej Kristian Guevara
College of Computing and Information Technologies
National University
Sampaloc, Manila
guevaraeo@students.national-u.edu.ph

***Abstract*—Ensuring the quality of fruit is crucial for consumers and distributors, with ripeness being a key factor for quality assurance. Traditional methods of determining fruit ripeness rely on manual inspection, which is labor-intensive and prone to errors. This research explores the use of Convolutional Neural Networks (CNNs) to automate the classification of fruit ripeness, focusing on apples, bananas, and oranges. Utilizing a dataset from Kaggle, the study preprocesses images and trains a CNN model to classify fruits as either fresh or rotten. The model architecture includes various image preprocessing techniques and a sequence of convolutional and pooling layers, optimized using TensorFlow and Keras modules. The trained model achieved a validation accuracy of 95**

***Index Terms*—deep learning, cnn, image classification, fruit ripeness, agricultural technology**

I. Introduction

Ensuring the quality of a fruit is essential for consumers to know if the fruit is good to be served on the table; and for distributors to know if the fruit is good to be distributed on the market. For fruits, ripeness is the basis and important factor for quality assurance. The traditional method of determining the ripeness of fruits is through manual inspection which is a labor-extensive and error-prone process [1]. Moreover, with these challenges, the classification of fruit ripeness has garnered significance for its potential applications in the agricultural industry, supply chain management, and consumer health.

The advancement of machine learning techniques, specifically Convolutional Neural Network, has helped the field of computer vision in image recognition and classification, and even in object detection. Convolutional Neural Network has demonstrated outstanding performance in regards to different domains such as medical image analysis, autonomous driving, and facial recognition [2]. With this machine learning technique, classification of different fruit ripeness will help quality assurance of distributors and consumers. Furthermore, this advancement will simplify the labor-intensive and error-prone task of assessing fruit ripeness.

The research explores the use of Convolutional Neural Network to determine the ripeness of the three fruits - Apple Banana, and Orange- to classify whether they are fresh or rotten. The main objective of the paper is to create a machine learning model that can precisely classify the ripeness of the fruits based on their visual characteristics. Furthermore, the study aims to provide a dependable model that can be used in practical applications in the real world specifically for fruit ripeness.

Identify applicable funding agency here. If none, delete this.

The significance of this research is it has the potential to enhance the effectiveness and precision of fruit classification processes. This could lead to reduction of food wastage and ensure consumers receive fresh produce. Moreover, adopting such a system could simplify the supply chain, improve quality control, and support sustainable agricultural practices.

II. Related Work

A. Definition and basic architecture of CNNs

Convolutional Neural Network (CNN) is a deep learning algorithm that excels in the field of computer vision, particularly used for visual data analysis [3]. It uses convolutional operations to extract different features of an image and find patterns within. CNN's basic architecture consists of three groups: convolutional layers, pooling layers, and fully connected layers . Convolutional layers use a filter called kernels that move across the input to identify different features from the image. After the Convolutional layer, the Pooling layer follows which reduces the spatial dimensions of the input image while keeping the important features of the image. Lastly, Fully Connected layers are responsible for classifying the images based on the extracted features from the layers that come before it [4].

B. Advantages of using CNNs compared to traditional neural network

Multilayer perceptrons (MLPs), traditional neural networks with fully connected layers, struggle with spatial data like images due to inefficiency and a tendency to overfit. As this traditional neural network requires extensive computational resources and often learns irrelevant information. On the other hand, Convolutional Neural Networks (CNNs) connect nodes selectively and use fewer parameters, making them more efficient for image processing . including mobile and edge computing scenarios [5]. CNNs use parameter sharing, where fixed-weight filters traverse images, reducing the number of parameters and boosting efficiency. They also incorporate pooling layers to lower data dimensionality, which enhances performance and generalizability. Due to their capability to automatically extract features, capturing basic features in initial layers and complex patterns in later ones, CNNs excel in computer vision tasks like image recognition [6]. They are particularly well-suited for transfer learning, allowing for the fine-tuning of pretrained models for new tasks, thereby conserving computational resources. The efficient architecture of CNNs facilitates their deployment on various devices, including mobile and edge computing environments [7].

C. CNN on image processing and classification

Convolutional Neural Networks (CNNs) have demonstrated exceptional adaptability and performance across a range of applications, such as object detection, autonomous cars, and medical imaging. By identifying patterns and abnormalities in medical scans like MRIs and CT scans, CNNs are used in medical imaging to evaluate complicated visual data and aid in the diagnosis and treatment of disorders [8]. CNNs are crucial to the operation of self-driving cars because they evaluate visual data from cameras, allowing the car to identify and react to objects such as pedestrians and traffic signs, enhancing safety and navigation [9]. Furthermore, CNNs are particularly good at object identification tasks, which require them to precisely identify and locate things in pictures or video streams. These tasks have shown to be crucial in a variety of industries, including augmented reality and security and surveillance [10].

D. Studies specifically focusing on CNNs for fruit image processing and classification

Khatun et al. [11] explored the use of Convolutional Neural Networks (CNNs) for fruit classification, employing various image processing techniques. Their study underscores the significance of automation in fruit classification for numerous applications. They utilized the Fruits 360 dataset from Kaggle, selecting classes including Apple, Banana, Orange, Pear, Mango, Pineapple, and Strawberry. The dataset was divided into a training set with 180 images and a test set with 20 images per fruit. The study focused on key techniques such as image pre-processing, filtering, segmentation, feature extraction, and knowledge-based decision-making. These methods simplify image recognition and classification by extracting essential features like color, shape, size, and texture, and then using CNN algorithms to make decisions based on these features. This approach achieved an impressive 98.74% accuracy in fruit classification.

This study by Selvakumari and Gomathy [12] employs a CNN architecture to classify fruits, utilizing a dataset comprising 1877 images spanning ten distinct categories named: Cashew, Diamond Peach, Fuji Apple, Granny Smith Apple, Honeydew Melon, Kiwi, Nectarine, Orange, Plum, and Spanish Pear. The CNN model is designed with sixteen layers dedicated to feature extraction from the images, followed by the application of a support vector machine (SVM) classifier for the classification process. The proposed system demonstrates a classification accuracy of 99.2% and a recognition accuracy of 99.02%, showcasing its high efficacy in fruit image recognition tasks.

III. METHODOLOGY

A. Dataset

The dataset used in this study was from a Machine Learning and Data Science Community website Kaggle. The dataset uploaded by the user “SRIRAM REDDY KALLURI” 6 years ago contains images of 6 classes. These 6 classes are fresh bananas, fresh apples, fresh oranges, rotten bananas, rotten apples, and rotten oranges. The dataset has a total of 13599 images split among train and test folders. Moreover, the training class has files of 1581, 1466, 1693, 2224, 2342, and 1595 respectively. Additionally, the testing class has 381 files for fresh bananas, 395 files for fresh apples, 388 files for fresh oranges, 530 files for rotten bananas, 601 files for rotten apples, and 403 files for rotten oranges.

TABLE I
TRAINING DATASET SUMMARY

Fruits	**Fresh(File Count)**	**Rotten(File Count)**
Bananas	1581	2224
Apples	1466	2342
Oranges	1693	1595

TABLE II
TEST DATASET SUMMARY

Fruits	**Fresh(File Count)**	**Rotten(File Count)**
Bananas	381	530
Apples	395	601
Oranges	388	403

These images contain different instances of fruit orientation, variations of the fruit specifically for the color of the apple, and a group of fruits seen with bananas and oranges. However, the downside is almost all of the images have the same background of white which is not good for the learning process of the model. This problem raises an issue in the generalization of the dataset in the learning process. Lastly, there is no folder for validation in this dataset which is an important key in validation accuracy, the issue will be discussed further in the next section.

B. Preprocessing

Since the dataset was garnered online, the image preprocessing was done before training the model in the Python notebook. However, the dataset does not include a validation folder which is an important part of validating the model's accuracy while training. To solve this, the *train-test-split* function splits the training folder into a new train folder and a validation folder. For the new folders, the researcher used a standard utility module from Python, shell utility(shutil). Shutil is used for files and directory functions such as moving, renaming, and deleting files, which were used to create a validation folder. Now the files are split into three folders, the training folder having 8718 files, the validation folder with 2183 files, and lastly, the test folder with 2698 files.

C. Model Architecture

This section will explain the model's architecture that was experimented with by the researchers. This will contain a comprehensive description of the best configuration of the model for better accuracy. Ripeness analysis falls under the classification class and image classification works well with the Convolutional Neural Network(CNN) Architecture. Hence, the researchers utilised CNN using different image pre-processing techniques and experimented with the layers to obtain the best output.

TensorFlow provides different modules to optimise the computing process and learning efficiency while training the model, pre-processing layers from Keras module were used for this purpose. To normalise the images, the data were rescaled into 0 to 1 values of the RGB Channels that is usually from 0 to 255. This way, the training process will be much faster while providing a more accurate computation. To implement this, a preprocessing layer was used – Keras Rescaling Layer enables rescaling the images' pixels from 0-255 to 0-1. Moreover, since the images of the fruits show different orientations, a Keras RandomFlip Layer enables a random flipping function of the images. The random flipping can be done horizontally, and or, vertically. This is to ensure that the model can classify the fruits even with different orientations. Lastly for image pre-processing, a Keras RandomContrast Layer was used. This layer randomly increases or decreases the contrast of the images while training. Randomly adjusting images' contrast makes the model more adaptive in different environments and lighting. All these image pre-processing techniques were used to prevent the model from overfitting in the training to ensure better accuracy for real-life applications.

As for the hidden and output layers, a sequence of 14 layers was utilized to train the model. Moreover, 5 pairs of two-dimensional convolutional and pooling layers are present in this model. All of the convolutional layers consist of 128 units with a 3x3 filter, and it uses Relu as its activation function. On the other hand, the pooling layers all consist of a 2x2 filter. Then, these layers were followed by a Dense Layer containing 200 neurons. Additionally, a dropout layer with a unit of 0.5 was added to prevent the model from overfitting. Lastly, the last dense layer has 6 units representing the 6 classes of the dataset and using softmax as its activation function. The diagram below shows the model summary.

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 rescaling_7 (Rescaling)     (None, 128, 128, 3)       0

 random_flip_7 (RandomFlip)  (None, 128, 128, 3)       0

 random_contrast_3 (RandomC  (None, 128, 128, 3)       0
 ontrast)

 conv2d_18 (Conv2D)          (None, 126, 126, 128)     3584

 max_pooling2d_18 (MaxPooli  (None, 63, 63, 128)       0
 ng2D)

 conv2d_19 (Conv2D)          (None, 61, 61, 128)       147584

 max_pooling2d_19 (MaxPooli  (None, 30, 30, 128)       0
 ng2D)

 conv2d_20 (Conv2D)          (None, 28, 28, 128)       147584

 max_pooling2d_20 (MaxPooli  (None, 14, 14, 128)       0
 ng2D)

 conv2d_21 (Conv2D)          (None, 12, 12, 128)       147584

 max_pooling2d_21 (MaxPooli  (None, 6, 6, 128)         0
 ng2D)

 conv2d_22 (Conv2D)          (None, 4, 4, 128)         147584

 max_pooling2d_22 (MaxPooli  (None, 2, 2, 128)         0
 ng2D)

 flatten_6 (Flatten)         (None, 512)               0

 dense_12 (Dense)            (None, 200)               102600

 dropout_6 (Dropout)         (None, 200)               0

 dense_13 (Dense)            (None, 6)                 1206

=================================================================
Total params: 697726 (2.66 MB)
Trainable params: 697726 (2.66 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Fig. 1. Summary of the layers used in the model.

D. Model Training

Moving on, this section will explain how the model has been trained using the layers stated in the previous section. Compiling the layers, the optimizer used was adam. To calculate the loss, the SparseCategoricalEntropy was used, this loss function is commonly used in image classification models. Lastly, accuracy was used as a metric in evaluating the model's training performance. Additionally, a callback function was included in the model training i.e. the Early Stopping module. This module will monitor if there are no changes in the validation loss of the training in a span of 5 epochs; no changes will force the training to stop. Lastly, the number of epochs provided for the training is 15.

E. Model Evaluation

While training and even after, evaluation techniques should be used to monitor the model's performance. In this study, the researchers used different methods to evaluate the trained model for image classification in the ripeness of apples, bananas, and oranges. Another way of the model's performance evaluation is through the use of plotting. Plotting the predicted labels, and true labels with their confidence score can provide information regarding the model's accuracy. Moreover, a line graph was used to review the performance of the model through every epoch. Lastly, a confusion matrix with precision, recall, f1-score, and overall accuracy scores was utilized to show the model's accuracy with new or unseen data.

IV. Results and Discussion

The previous chapter explained all the methodologies used for this study. This chapter will now focus on presenting the findings and reports about the model for image classification in the ripeness of bananas, apples, and oranges.

In the training, the model's performance was shown to be continuously improving hence the early stopping function was not used. After 15 epochs of training, the model had 94.55% of training accuracy and 0.16 loss. Additionally, the validation accuracy of the model is 95.40% while having a 0.14 validation loss.

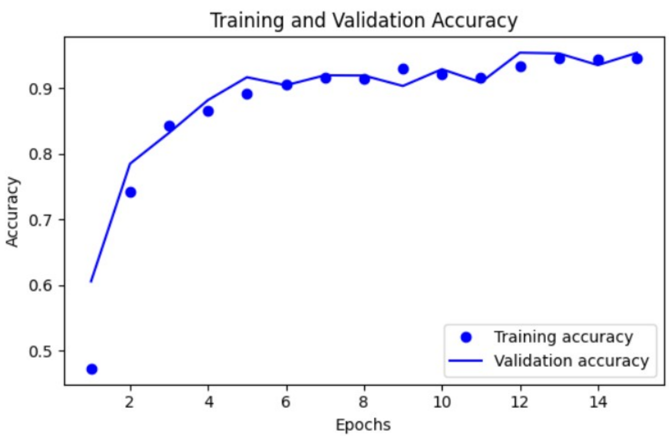

Fig. 2. Model Training and Validation Accuracy

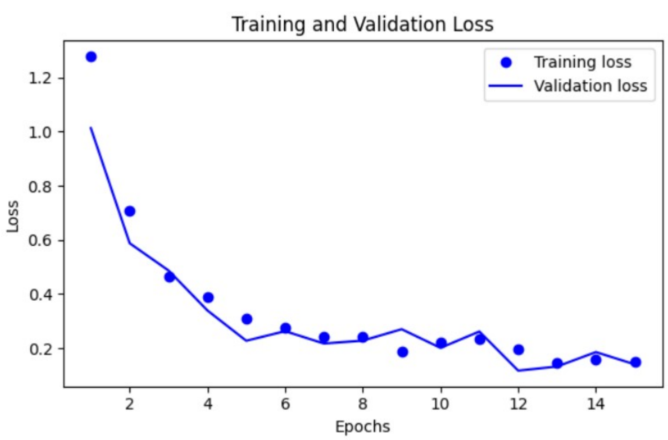

Fig. 3. Model Training and Validation Loss

The images seen below show the validation test the model went. It presents the true labels and the predicted labels with its confidence score. The corresponding value the number represents is 0 - fresh apples, 1 - fresh banana, 2 - fresh oranges, 3 - rotten apples, 4 - rotten bananas, 5 - rotten oranges.

Fig. 4. Model Validation Confidence Score

Lastly, the confusion matrix provides a definite visualization of the performance of the model in classifying the fruits. Out of 395 fresh apple test cases, it correctly predicted 374 of them. Fresh banana has 381 samples and all of it has been predicted correctly by the model. Fresh oranges consist of 381 samples and only 377 have been correctly predicted. The model has predicted 563 samples out of the 601 samples of rotten apples. The rotten banana has been misclassified 3 times out of 530 samples, 527 being the correct prediction. Finally, rotten oranges have only 353 correct predictions out of 403 samples. Looking at the data, it shows that classifying the rotten oranges is the least accurate out of all fruits. However, the model still showed a good performance having an accuracy of 95%

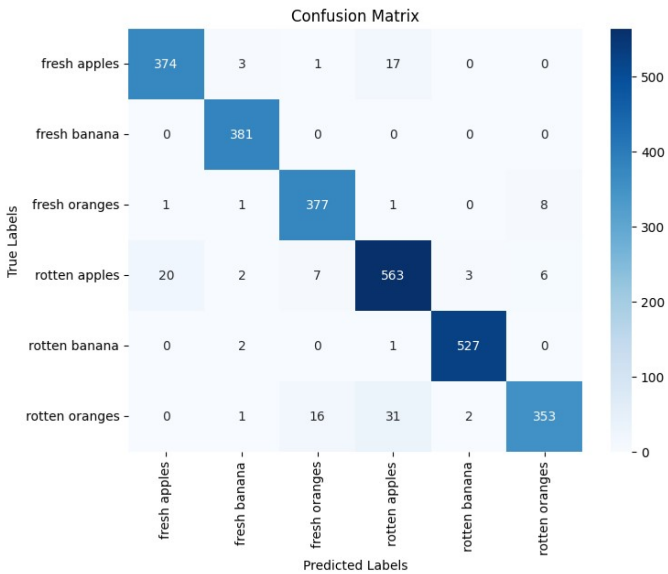

Fig. 5. Confusion Matrix of the Model

TABLE III
CLASSIFICATION REPORT AND ACCURACY SCORE

	Precision	Recall	F1-Score	Support
fresh apples	0.95	0.95	0.95	395
fresh bananas	0.98	1.00	0.99	381
fresh oranges	0.94	0.97	0.96	388
rotten apples	0.92	0.94	0.93	601
rotten bananas	0.99	0.99	0.99	530
rotten oranges	0.96	0.88	0.92	403
accuracy			0.95	2698
macro avg	0.96	0.95	0.95	2698
weighted avg	0.95	0.95	0.95	2698

V. CONCLUSION

In this study, the researchers utilize the use of TensorFlow library to train an image classification model. Building a CNN Model would be a difficult task if not for the available functions and modules provided by TensorFlow. Moreover, image classification using CNN proved useful for detecting the ripeness of the fruits, specifically, apples, bananas, and oranges. Feature extraction of convolutional layers enables the identification of whether the fruit is rotten or still fresh. However as stated, the study only tested this on three fruits, there are still more fruits that are usually consumed in the Philippines to apply these techniques. This model can be used for household environments and quality control for better production and contribution of supplies.

However, the researchers believe that the dataset in this study still lacks variability because some of it is augmented or synthetic. This proves to be a limitation for further improving the model since it can misclassify some of the new data that contains real-life images of fruits. For future reference, adding new fruits may require more convolution layers since some of the fruits have a more complex feature that requires more computing processes. Lastly, this model can extend up to the use of vegetables as the sample since vegetable freshness is harder to distinguish rather than fruits but still with the same objective of providing quality control for these products.

REFERENCES

[1] M. Rizzo, M. Marcuzzo, A. Zangari, A. Gasparetto, and A. Albarelli, “Fruit ripeness classification: A survey,” Artificial Intelligence in Agriculture, vol. 7, pp. 44–57, Mar. 2023, doi: 10.1016/j.aiia.2023.02.004.

[2] “Convolutional Neural Networks: What it is and Why it Matters?” https://clanx.ai/glossary/convolutional-neural-networks-cnns

[3] L. Craig and R. Awati, “convolutional neural network (CNN),” Enterprise AI, Jan. 11, 2024. https://www.techtarget.com/searchenterpriseai/definition/convolutional-neural-network

[4] M. Mishra, “Convolutional neural networks, explained - towards data science,” Medium, Dec. 15, 2021. [Online]. Available: https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939

[5] Dinesh, “CNN vs MLP for Image Classification - Analytics Vidhya - Medium,” Medium, Apr. 22, 2024. [Online]. Available: https://medium.com/analytics-vidhya/cnn-convolutional-neural-network-8d0a292b4498

[6] A. Biswal, “Convolutional Neural Network Tutorial,” Simplilearn.com, Nov. 07, 2023. https://www.simplilearn.com/tutorials/deep-learning-tutorial/convolutional-neural-network

[7] Z. Zhao, L. Alzubaidi, J. Zhang, Y. Duan, and Y. Gu, “A comparison review of transfer learning and self-supervised learning: Definitions, applications, advantages and limitations,” Expert Systems With Applications, vol. 242, p. 122807, May 2024, doi: 10.1016/j.eswa.2023.122807.

[8] F. A. Mohammed, K. K. Tune, B. G. Assefa, M. Jett, and S. Muhie, “Medical Image Classifications using Convolutional Neural Networks: A survey of current methods and statistical modeling of the literature,” Machine Learning and Knowledge Extraction, vol. 6, no. 1, pp. 699–736, Mar. 2024, doi: 10.3390/make6010033.

[9] A. Agnihotri, P. Saraf, and K. R. Bapnad, “A convolutional neural network approach towards Self-Driving cars,” ResearchGate, Sep. 2019, [Online]. Available:

[10] R. Galvez, A. Bandala, E. Dadios, R. Vicerra, and J. Maningo, “Object detection using convolutional neural networks,” TENCON 2018 - 2018 IEEE Region 10 Conference, Oct. 2018. doi:10.1109/tencon.2018.8650517 https://doi.org/10.1109/tencon.2018.8650517

[11] Khatun, M., Ali, F., Turzo, N. A., Nine, J., and Sarker, P..” Fruits classification using convolutional neural network”. GRD Journals-Global Research and Development Journal for Engineering, 5(8), 1-6.

[12] R. Shantha Selvakumari, and V. Gomathy, “View of Fruit Shop Tool: Fruit Classification and Recognition using Deep Learning.” https://cigrjournal.org/index.php/Ejounral/article/view/7847/4067