Banana Ripeness Classification

Machine Learning Project Team 19

Audrey Suhadinata (A.S.)
EECS-GS
NTHU
Jakarta, Indonesia
audreychung@gapp.nthu.edu.tw

Chalanthorn Chanmathikornkul (C.C.)
EECS-GS
NTHU
Bangkok, Thailand
cc109006243@gapp.nthu.edu.tw

Keith Saudjana (K.S.)
EECS-GS
NTHU
Jakarta, Indonesia
knsaudjana@gmail.com

Kevin Karnadi (K.K.)
EECS-GS
NTHU
Surabaya, Indonesia
kevin.karnadi2002@gmail.com

Kittipon Leerujanont (K.L.)
EECS-GS
NTHU
Bangkok, Thailand
kittipon2002@gmail.com

Porama Rittibunlee (P.R.)
EECS-GS
NTHU
Bangkok, Thailand
porama2001.11@gmail.com

***Abstract*—One of the most significant qualities in determining the appropriate time of fruit for consumption is its ripeness. Throughout generations, fruit agriculture has developed various techniques in assessing ripeness through biochemical, optical, or mechanical means. However, the most commonly implemented method - mechanical, is not consistent or time efficient. This paper focuses on applying Machine Learning preprocessing techniques involving computer vision image segmentation, using R-CNN model, to enhance the classification of banana's ripeness. Such machine learning models include artificial neural networks (ANN), convolutional neural networks (CNN), and a pre-trained MobileNetV2 model. The banana's ripeness is categorized into four labels: green, yellowish-green, mid-ripen, and over-ripen. The results show the different impact that image masking and segmentation have on each models during classification. The findings shows that CNN without segmentation to yield the highest validation accuracy of 91.46%. However, the accuracy drops to 77.5% when applied on the testing data.**

***Index Terms*—Banana ripeness, ANN, CNN, MobileNetV2, R-CNN, Image Segmentation**

I. Introduction

Making sure that the fruit is sold at its peak quality depends greatly on determining the maturity of the fruit. To do this, the fruit must be harvested at the proper time so that it is not overripe or under-ripe when it reaches the grocery store. At various stages of ripeness, fruits like bananas contain a variety of nutrients. Bananas lose their disease-preventive tannins, alkaloids, and disease-fighting chemicals as they ripen, while their sugar and vitamin contents rise. The fruit may be ready to be eaten when it turns from green to yellow and develops more brown patches. Nutrient contents among profitability and food waste reduction could be optimized with the correct ripeness classification methods.

Due to recent advancements in technology, commonly used manual technique weighs far below in terms of reliability and consistency when compared with automated deep learning [5]. The emergence of computer vision achieved through image-processing tools and CNN could serve to be the cost and time effective solution. Using the dataset provided by [1], this paper aims to classify Egyptian banana species into different labels such as green, yellowish-green, mid-ripen, and over-ripen. The steps include preprocessing the data, augmenting the data and training the models. The data preprocessing stage involves applying gaussian blur, image masking and segmentation, and resizing. The trainable models consist of CNN and ANN trained from scratch, and transfer learning from a pre-trained MobileNetV2 model. After training, the models are evaluated by their accuracy, precision, recall, and f1 score. Insights on applying additional image masking and segmentation to guided filter are gained from analyzing the experimental results.

II. Methods

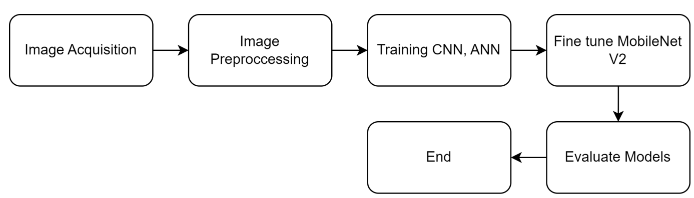

Fig. 1. Research methodology

Figure 1 shows the stages in the research methodology: image acquisition, image preprocessing, training the CNN and ANN models, fine tuning the pre-trained MobileNetV2 model, and finally evaluating the results of the models. The libraries used include, but are not limited to Pytorch, Skimage, Tensorflow, and cv2.

A. Data Acquisition

The first stage of image acquisition is obtained from a dataset provided by [1]. The dataset consists of 273 banana images divided into four ripeness classes: 104 green, 48 yellowish-green, 88 mid-ripen, and 33 over-ripen. A testing dataset with size of 40 data collected from images taken on the field and various websites is also collected to measure how well the model detects and classifies the banana in the image outside of the training dataset.

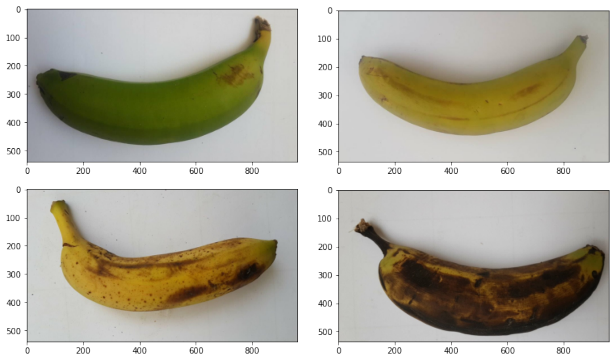

Fig. 2. Training and validation dataset with 4 classes of banana. Green, yellowish-green, mid-ripen and over-ripen (from left to right)

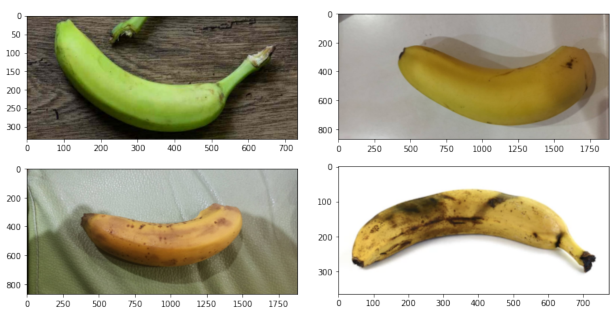

Fig. 3. Testing dataset with 4 classes of banana. Green, yellowish-green, mid-ripen and over-ripen (from left to right)

B. Data Preprocessing

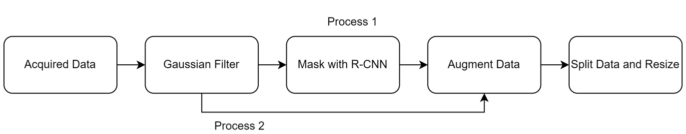

Fig. 4. Image Preprocessing

In this stage, two different processing steps were included where the first one includes all the steps, while the second one skips image masking and segmentation step after applying the filter, and goes directly to splitting data and resizing. Then the data are further processed by dividing the pixel value of each channel.

The first step applies the Gaussian blur filter, Fig. 5 on acquired data to preserve edge and perform smoothing on image to reduce noises. The next step utilizes transfer learning from R-CNN to mask and segment the banana from the background. Fig. 6 shows the process on how the background

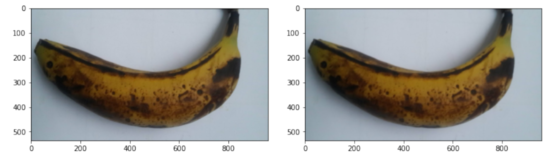

Fig. 5. Applying gaussian blur filter

removal works. There is a special case when R-CNN may detect more than one banana in the picture, the bigger mask is chosen for this case as shown in Fig. 7.

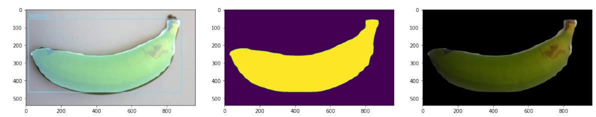

Fig. 6. masking and background removal

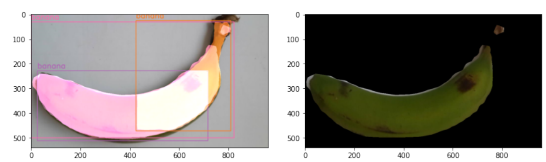

Fig. 7. multiple bananas detected

The segmented data is then augmented by random uniform distribution and rotation to increase the number of data in the dataset. Finally, the data is split into training and validation dataset in an 80-20 ratio. Each processed image is also resized to the shape of (64,64,3). Then, the pixel value in each channel is further preprocessed into the range of [0, 1] before being fed to CNN and ANN models. However, for MobileNetV2 the pixel value range is [-1, 1] as stated in Keras Tensorflow library.

C. Model Implementation

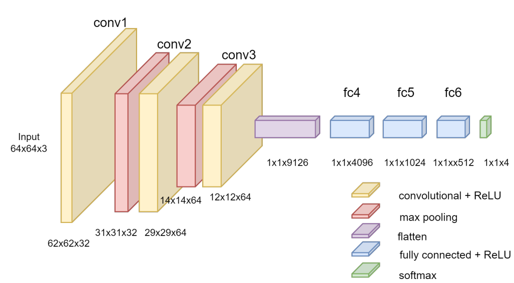

Fig. 8. CNN

The three models trained are CNN, ANN, and a transfer learning model using MobileNetV2. The models were all implemented with the TensorFlow library. The CNN model of 6-hidden layers, as shown in Fig. 8, consists of 3 convolution and 3 fully connected layers. It has a total of 42.5M trainable parameters. While the ANN model of 4-hidden layers, as shown in Fig. 9, consists of 4 dense layers. It has a total of 219.6K trainable parameters. Lastly, the MobileNetV2 model, also the smallest model in the Keras library, has a total of 3.4M parameters.

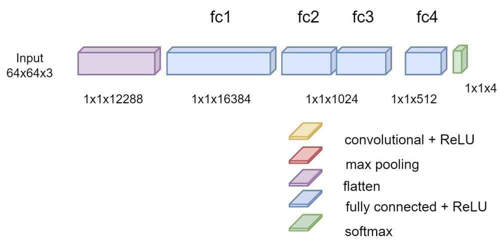

Fig. 9. ANN

The MobileNetV2 model is picked due to its small size and used as a standard to compare with the proposed CNN and ANN models. The model architecture includes the base model of MobileNetV2 without classifier in addition to some added top layers including global average pooling2D, dropout layer at a rate of 0.3, and a dense output layer of 4 classes with softmax activation. There are a total of 5124 trainable parameters in the top layers.

D. Evaluation of Model

The trained models are evaluated in terms of performance by accuracy, precision, recall, and f1 score with methods provided by Sci-kit library.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (1)$$

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

$$F1Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

III. Results

A. Training

The experiments were conducted using Google Colab with a GPU hardware accelerator and libraries such as Tensorflow and scikit-learn. The dataset used was split into training, validation dataset of ratio 80-20 respectively. In addition, testing data was also added to measure how well the model fared in real life applications. Furthermore, each proposed models were trained on two different sets of preprocessed data.

The CNN model ran for 55 epochs at the learning rate of $3e^{-5}$, while the ANN model ran for 110 epochs at the learning rate of $1e^{-6}$. Both models use an Adam optimizer, and a cross categorical entropy loss function.

In contrast, the transfer learning of MobileNetV2 model consists of two steps: training the classifier and fine-tuning. Training the classifier runs for 20 epochs with a learning rate of 0.003, which decays every 500 steps, an Adam optimizer, and a categorical cross entropy loss function, the base model is frozen during this process. The fine-tuning step unfreezes the base model from the 100^{th} layer, and continues to be trained as a whole for an additional 40 epochs with learning rate of 0.00003, which also decays every 500 steps, an RMS prop optimizer, and a categorical cross entropy loss function. After the second step of fine-tuning, the number of trainable parameters increases from the initial 5124 to 1866564.

B. Experimental Results

The result collected from training the proposed models with two different preprocessing methods: 1) with image segmentation, 2) without image segmentation are shown in table I.

TABLE I
RESULT OF PROPOSED MODELS

	Models	**Accuracy**	**Precision**	**Recall**	**F1**
Segment	CNN	91.46%	91%	91%	91%
	ANN	81.1%	82%	81%	79%
	MobileNetV2	84.15%	84%	84%	83%
No Segment	CNN	91.46%	92%	91%	91%
	ANN	90.85%	82%	81%	79%
	MobileNetV2	88.41%	88%	88%	87%

The best performing model on validation dataset is CNN with 91.46% accuracy, 91% precision, 91% recall, and 91% f1 score, followed by MobileNetV2 and ANN respectively. However, the image-segmentation proved indifferent to CNN, but harmful MobileNetV2 and ANN. Each model's behavior will be further analyzed with results from visual confusion matrix and loss curve.

C. CNN Results

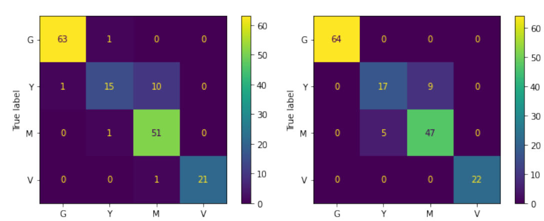

Fig. 10. CNN model (Left: without segment. Right: with segment)

Upon closer examination at the confusion matrix for CNN, or Fig. 10, it can be seen that performing mask-segmentation strengthens the classification for correctly predicting the extreme end classes such as green and over-ripen bananas at the cost of having weaker prediction towards the yellowish-green and mid-ripen bananas. From Fig. 11, the accuracy and loss curve appears more stabilized, but converges earlier with image-segmentation data. However, recall that its f1 score remains the same from Table I.

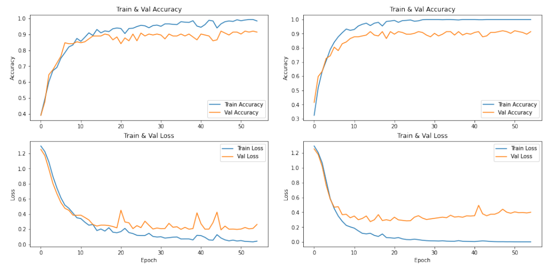

Fig. 11. CNN model (Left: without segment. Right: with segment)

D. ANN Results

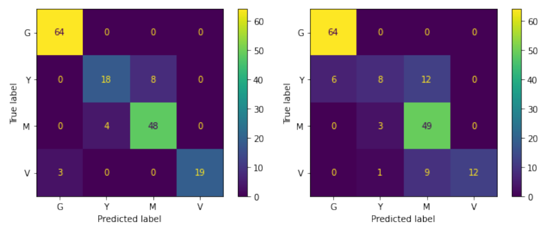

Fig. 12. ANN model (Left: without segment. Right: with segment)

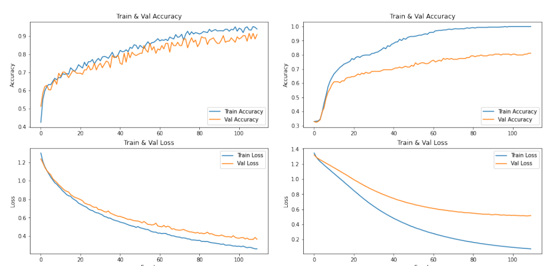

Fig. 13. ANN model (Left: without segment. Right: with segment)

Although the f1 score for ANN trained with and without image-segmentation are the same as shown in Table I, the accuracy drops significantly when compared with the latter. From Fig. 13, the accuracy curve for ANN with image-segmentation converges significantly earlier. Fig. 12 also shows 3 instances of severe mispredicting over-ripen as green banana. All this could imply over-fitting to the training dataset and bias towards certain class due to data imbalance.

E. MobileNetV2 Results

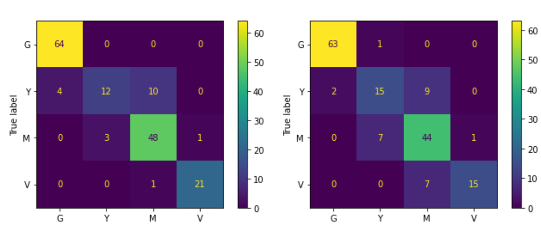

Fig. 14. MobileNetV2 (Left: without segment. Right: with segment)

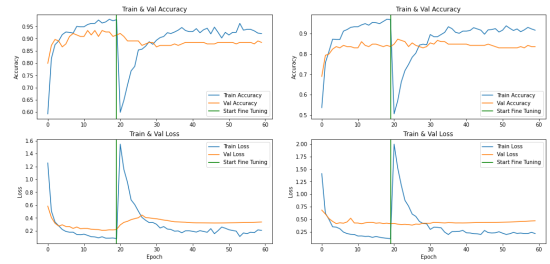

Fig. 15. MobileNetV2 (Left: without segment. Right: with segment)

Recall from Table I. that both accuracy and f1 score drops for the transfer learning model, however the drop in accuracy is not as drastic compared to ANN model. From Fig. 14, the classification on the edge classes improved a little, but at a greater cost of misprediction in yellowish-green and mid-ripen bananas. From Fig. 15, the accuracy and loss curve of transfer learning converges earlier. The negative impact on image-segmentation on transfer learning model is presumably because the imported weights on base model, MobileNetV2, were trained on different characteristics from the input.

F. Testing

TABLE II
RESULT OF PROPOSED MODELS

	Models	Accuracy	Precision	Recall	F1
Segment	CNN	60%	66%	60%	59%
	ANN	55%	54%	55%	52%
	MobileNetV2	67.5%	69%	68%	67%
No Segment	CNN	77.5%	79%	78%	77%
	ANN	45%	44%	45%	44%
	MobileNetV2	70%	72%	70%	69%

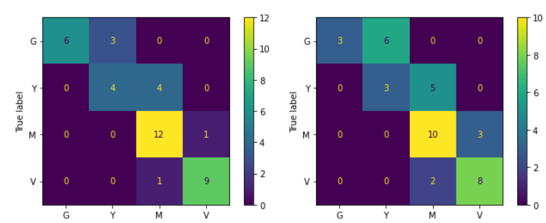

Fig. 16. CNN (Left: without segment. Right: with segment)

The result above shows that the CNN and MobilenetV2 models are far superior than the basic ANN models. Compared to each other on the other hand, their results improve depending on whether segmentation was used or not.

Another point to take note of is the fact that performing segmentation actually worsens both the CNN and MobileNetV2 models and improves the ANN model, it is presumably because the CNN model highlights the important features of the images on the convolution layer. However, because the training

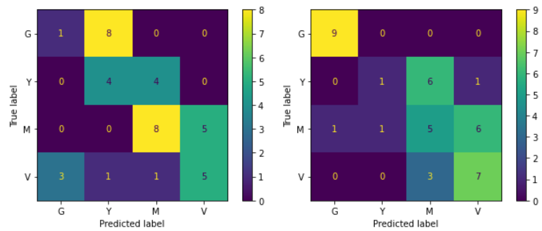

Fig. 17. ANN (Left: without segment. Right: with segment)

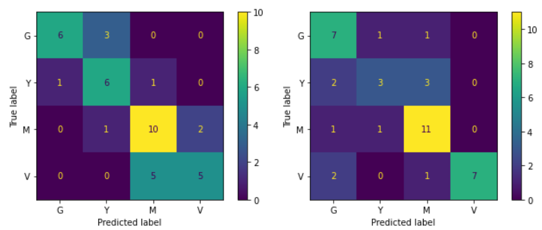

Fig. 18. MobileNetV2 (Left: without segment. Right: with segment)

data collectively shares the same monotone background, the ANN model, which is simpler in nature compared to the CNN, will believe that the grey background is a component of the banana as a whole. Hence, removing the background works more efficiently for the ANN. When looking at their respective confusion matrices however, it can be observed that the predictions made by the model which does not utilize segmentation are more volatile as they would make some very estranged guesses, such as identifying a green banana as an over-ripen one. This issue does not persist in the models that do use segmentation on its data.

IV. Discussion

The major issue that can be concluded from the experimental results is that the training and validation dataset are just very different to the testing data, in that their backgrounds are always in the same grey colour. The testing data has backgrounds with a much larger variation of colours and shadows. This ends with the models producing lackluster results as the training dataset does not accommodate for the diversity of the testing dataset.

Additionally, another possible attribute to the low accuracy of the testing data might be due to using a loss function that is not suited for the job. If the ripeness of the bananas is considered as a range from green to over-ripen and by penalizing the model based on how far its predictions are from the ground truth, the loss calculated for the model would be better and would more properly correct the model. That way the issue of volatile predictions could be fixed as they would be more penalizing.

Future research will focus on gathering larger, more balanced, and more diverse sets of data for training, validating, and testing the implemented models in order to develop better image segmentation and classification techniques for bananas. Extensive research could focus on incorporating the proposed techniques with other transfer learning models to develop more efficient supervised and semi-supervised machine learning models for better classification results.

Contributions

- C.C. (18.5%) - Image preprocessing, fruit segmentation using RCNN, and project template
- P.R. (16%) - Research on fruit segmentation using Otsu's thresholding and presentation preparation
- A.S. (18.5%) - CNN Model, devised the algorithm, gathering testing data and collecting result for report
- K.S. (16%) - ANN Model, image augmentation, writing the report, and proof reading
- K.K. (16%) - Transfer learning implementation of MobileNetV2, and project topic idea
- K.L. (15%) - Result analysis, model evaluation and writing the report

References

[1] F. Mazen., A. Nashat. (2019, January). "Ripeness Classification of Bananas Using an Artificial Neural Network." Arab J Sci Eng, 44(8), 6901–6910. Available: https://www.researchgate.net/publication/330154837

[2] R. E. Saragih., A. W. R. Emaneul. (2021, April). "Banan Ripeness Classification Based on Deep Learning using Convolutional Neural Network." EIConCIT, 2021. Available: https://www.researchgate.net/publication/351644885

[3] S. R. Rath. (2020, Nov. 23). Instance Segmentation with PyTorch and Mask R-CNN[Online]. Available: https://debuggercafe.com/instance-segmentation-with-pytorch-and-mask-r-cnn/.

[4] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L. Chen. (2018, Jan. 13). MobileNetV2: Inverted Residuals and Linear Bottlenecks[Online]. Available: https://arxiv.org/abs/1801.04381v4

[5] Saranya, N., Srinivasan, K. Kumar, S.K.P. Banana ripeness stage identification: a deep learning approach. J Ambient Intell Human Comput 13, 4033–4039 (2022). https://doi.org/10.1007/s12652-021-03267-w