Classification of Fruit Ripeness Grades using a Convolutional Neural Network and Data Augmentation

Mauricio Rodriguez
Universidad Peruana de Ciencias Aplicadas (UPC)
u201510971@upc.edu.pe

Franco Pastor
Universidad Peruana de Ciencias Aplicadas (UPC)
u20141a052@upc.edu.pe

Willy Ugarte
Universidad Peruana de Ciencias Aplicadas (UPC)
willy.ugarte@upc.pe

***Abstract*—Currently the classification processes of the degree of maturity of fruits require the use of complex systems, which, most of the times, are not within the reach of small farmers or consumers who do not have knowledge of the characteristics that a fruit must have in order to be catalogued as immature, mature or rotten. For this reason, a tool that can be accessed by anyone, was designed and implemented through a mobile application that served as an interface. This article describes the use of a convolutional neural network for the classification of the degree of maturity of the following fruits: red apple, green apple, banana, orange and strawberry. First, two sets of images were constructed. Secondly, the data augmentation technique was performed and then the training of the convolutional neuronal network was performed using the dataset images as input. In order to know the performance of the different models generated, the following metrics were used: precision, accuracy, recall, log loss, and f1 score. The best average precision obtained was 96.34%.**

I. Introduction

Fruits have always occupied an important place in human nutrition, due to the properties they offer, the amount of supply and the easy acquisition of the product [1]. Currently, the processes of selection of harvested fruits that are available for sale require expert inspections or complex systems, procedures that are costly and demand controlled environments [2]. In the best cases, the selected fruits are in the physiological maturity, therefore it is a fundamental task of the consumer to select the one that is in the best possible state (consumption maturity) [3]. This identification process is carried out by means of a visual inspection of the consumers, based on knowledge of the color, texture and odor that it should have, procedures that are usually unhygienic and invasive [4]. Consequently, our research focuses on classifying the degree of ripeness, taking into account the immature, ripe and rotten states, of the following fruits: red apple, green apple, banana and strawberry. For this purpose, image processing techniques, specifically Otsu Thresholding and data augmentation, and a convolutional neural network have been used. Likewise, the literature of similar research was reviewed, as the case of [5], where by means of automatic convolutional encoders it was determined the ripening of Lycopersicon (tomato family), while [6] implemented a fuzzy classification architecture based on the RGB color model with descriptors. In addition, they provided an expert system for the detection of oil palm maturity using K-Nearest neighbour. This research presents a mobile application for the classification of the degree of maturity of four fruits (red and green apple, banana, strawberry and orange), grouping them in 3 states (immature, mature and rotten) using a convolutional neural network. In addition, a data set proper to the fruits in question was constructed, collecting images from different sources, having a total of 100 images for each class. In summary, as contributions to the research, we present:

- As far as we know, this is the first time that a multi-class classifier is made for the degree of degradation of the selected fruits using a CNN.
- We present a set of images collected from different sources and a second set of data with the images but without the background.

The Section II is a compilation of terms and theory used for the development of the research. Section III details the contribution of the research, how it was carried out and the instruments needed for this objective. Section IV presents the research and methods used for the classification of the degree of fruit degradation. Section V provides the results obtained from the experimentation. Section VI presents the conclusions and future perspectives on the research carried out.

II. Background

The study of automated fruit maturity categorization is extremely important for sorting and selection tasks for industries and end users. Therefore, for this research, the following definitions and terms were considered to develop this study.

A. Stages of fruit development

The fruits experience stages throughout their cycle of growth and maturation, being the following [7]:

1) **Growth stage:** The irreversible increase in the physical attributes (characteristics) of a developing plant or plant part.

2) **Maturation stage:** The stage of development that leads to the achievement of physiological and consumer maturity.
   - **Physiological maturity:** Stage where a plant or part of a plant will continue to develop even if it is separated from it.
   - **Consumption maturity:** Stage of development where a plant or part of a plant has the prerequisites for use by consumers.
3) **Ripening stage:** It is the set of processes that occur from the last stages of growth to the first stages of senescence, and result in a characteristic food quality and aesthetics, can be visualized in the changes in composition, color, texture or other sensory attributes.
4) **Stage of senescence:** It is the last stage of development during which degradation of biological components occurs, such as loss of chlorophyll, degradation of cell walls and alteration of membrane composition.

B. Fruit Ripeness Indicators

Establishing the maturity index involves determining consistent physicochemical changes. After monitoring the product throughout its development, measures were found that show a strong correlation between maturity and postharvest yield [8].

1) **Size and shape:** The fact that the fruit reaches a certain size can be used as a possible index of maturity; however, it cannot be used alone since the size of any fruit variety can be influenced by the load of fruit on the tree, climatic conditions and cultivation practices.
2) **Firmness of the pulp:** The firmness of the pulp decreases as the fruit reaches the stage of maturity and ripeness, either on the tree or separated from it. However, flesh firmness alone is not a satisfactory minimum maturity index, as flesh firmness, among fruit varieties, varies with fruit size, climatic conditions, and growing practices.
3) **Color:** It is determined by the variety of pigments present in the skin tissue and pulp. As the fruit reaches maturity and ripeness, the color changes from green to red or yellow. Since the development of red color in nectarines and peaches depends on their exposure to light, the position of the fruit on the tree influences the degree of final coloration. Changes in background color and fruit flesh are not affected by sunlight.
4) **Soluble solids content:** The soluble solids content, or SSC, increases as the fruit reaches maturity and its use as a maturity index is limited by variations among varieties, production areas, and seasons.

C. Computer Vision

This problem belongs to the area of artificial intelligence, since it consists of designing algorithms that allow computers to simulate the human capacity to perceive the world. Computer vision becomes a problem since there is a big gap between pixels and their meaning, in fact, what computer sees in a 200×200 RGB image is a set of 120, 000 values. The path from numbers to meaningful information is very complicated. Therefore, it could be said that the visual cortex of the human brain solves a difficult problem that consist on understanding images that are projected on our retina and converted into neural signals. In this context, it should be noted that the three major problems that computer vision encompasses are image segmentation, image classification, and object detection [9].

D. Background Subtraction

In the research of [10], it is referred that the background subtraction generally fulfills the following structure

1) **Background initialization:** First you have to build a background model based on a fixed number of frames. This model can be designed in several ways (statistical, fuzzy, inspired by neurology, etc.).
2) **Foreground detection:** In the following tables, a comparison between the current frame and the background model is processed. This subtraction leads to the calculation of the foreground of the scene.
3) **Background maintenance:** During this detection process, the images are also analyzed to update the background model learned in the initialization step, with respect to the learning rate.

E. Deep Learning

Deep learning belongs to the group of automatic learning techniques, where the main feature is the large number of layers of information processing stages in hierarchical architectures for learning unsupervised features, and for pattern analysis. Deep learning has the essential purpose of calculating hierarchical characteristics or representations of observation data, where higher level characteristics or factors are defined from lower levels [11].

F. Convolutional neural networks

Convolutional neural networks or CNN (see Fig 1) are a widely used deep learning framework, which was inspired by the visual cortex of animals. Initially they were used for object recognition tasks, but now their range of application has been extended to domains such as object tracking, pose estimation, text detection and recognition, and scene tagging [8].

In addition, CNN is a class of advanced deep artificial neural networks that have been used to produce accurate performance in computer vision tasks, such as image classification and detection. CNN's have weights, biases, and outputs through nonlinear activation. In addition, the neurons are arranged in a volumetric fashion: height, width, and depth [12].

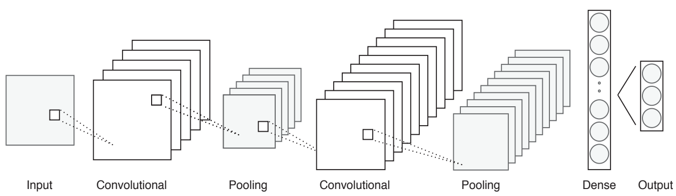

Fig. 1. Basic structure of a CNN

G. Data Augmentation

Data augmentation prevents model over adjustment by applying transformations to training data set images. This is done under the assumption that more information can be extracted from the original data set through magnification. These augmentations artificially inflate the size of the training data set through data warping or oversampling. Data warping magnifications transform existing images so their label is preserved. This includes enhancements such as geometric and color transformations, random deletion, opponent training, and neural style transfer. Oversampling augmentations create synthetic instances and add them to the training set [13].

III. Main Contribution

In this section we will detail our research in detail.

A. Preprocessing

It was used the technique of data increase, since there were only 100 samples for each fruit state, these images were extracted from Internet sources, in order to increase the volume of information for the training of CNN. Some examples of the technique applied can be seen in Fig 2. Using the ImageDataGenerator class from the Keras library [14], the volume of data was increased by 600%, applying the following random transformations to the images shown in Table I.

This 6 transformations were chosen because of their presence on previous works from our state of the art. Out of these transformations, the only ones that could alter images to a point of no recognition are the shear and zoom transformations. That´s why we limited their range to a maximum of 0.2. Re-scaling, rotating, and flipping the image vertically or horizontally are transformations that could be used without altering the image to a point of no recognition.

Fig. 2. Example of data Augmentation applied to the dataset

TABLE I: Example of data augmentation ap-plied to the dataset

Parameter	Value
rescale	1.0/255
shear range	0.2
zoom range	0.2
rotation range	180
horizontal flip	True
vertical flip	True

B. Convolutional neural network

Our motivation is to simulate the human vision in a system capable of classifying the degree of ripeness of a fruit by analyzing a picture of it. After reviewing the literature, it was determined to use a 9-layer convolutional neural network as shown in Fig 3. This neural network structure is inspired on the VGG-16 architecture. In a similar way as the VGG-16 architecture, this neural network consists of 3 blocks of convolutional layers interconnected with 3 max pooling layers. For each convolutional layer, a 3×3 pixel mask was used as

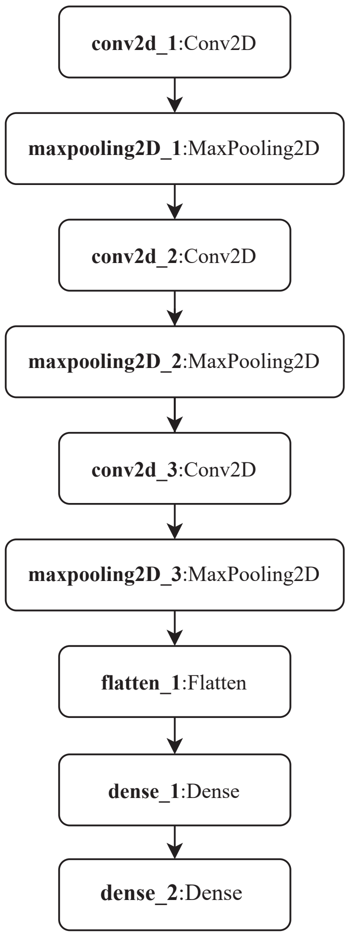

Fig. 3. Proposed CNN structure

a convolution kernel to find the most important characteristics of the images. Connected to the last max-pooling is our output pipeline, which consists of one flatten and two dense layers to reduce the dimensionality and perform the classification of the ripening states. Following the VGG-16 architecture, we intended to maintain the structure of blocks of convolutional-2d layers. Because of the 3×3 size of our kernel, the output of each convolution would result in an image of $(n-2) \times (n-2)$ dimension per each $n \times n$ image input. Also, the connected max pool-2d layers would result in a $(n/2) \times (n/2)$ output dimension per each $n \times n$ input. In that sense, having as input images of 32×32 pixels, we limited our structure to a total of 3 convolutional layers and 3 max pool-2d layers for our connected convolutional layers. After our flattening layers, we implemented two dense layers. The first one works as a hidden layer of 32 neurons which receives the flattened output image from the last max-pooling layer. Given the reduced size of our data-set, the amount of neurons of this hidden layer was reduced to 32 to reduce over-fitting. Finally, a dense layer with a shape of 3 was used as our output layer for each of the classes to be predicted.

IV. Related Works

The classification or estimation of fruit ripening degrees for the industrial sector is a problem that seeks to improve the times of product selection with the lowest cost of equipment or people. Based on this, the researches [15]–[18] prepared specialized environments to take the fruits photographs, having the same conditions of illumination, angle and background color. Compared to our work, that is focused on the final users (vendors and small farmers), thus special environments are not required, giving a flexibility for image capture process, only needing to establish rules for the way the photo is captured.

In the researches [15], [16], K-Nearest Neighbor and Multi-class Support Vector Machine are used to determine the maturity grades of the oil palm and passion fruit respectively. For the first job [15], 70 images are counted for 7 levels of ripeness, then the size of the input image is validated, after that, the filter for the detection of sobel edges is applied, the image characteristics such as intensity, contrast, entropy, standard deviation are extracted and the Euclidean distance from the image is calculated, finally the degree of ripeness is determined. In the same way in [16], to classify the 3 selected maturation states (unripe, near ripe and ripe) through an SVM, there are 40 5-second videos for each class. First each frame of the videos are transformed to the L*a*b* color space to remove the background, using only the a*b* values, using K means clustering method determine the pixels to be removed (background) and the pixels of the passion fruit. Second, for SVM training they use all 6 sides of the fruit and RGBa*, thus obtaining 24 characteristics from each sample. In a similar way as in research [16] our approach shares the same amount of maturation states. However, instead of classifying from an unripe to a rip state, our picked classes go from in mature to rotten. Another main difference between our work and the mentioned researches is the approach for the construction of our second set of images. For this process, we manually removed the background of the images using the Paint3D magic selection tool, as a result we got rid of the variation of the backgrounds of our images from the data set. In addition to this, another main difference in our approach is the usage of a convolutional neural network as the classification model.

In the investigations [17], [18], with the same purpose of determining the degrees of ripeness of bananas and apples respectively, an artificial neural network is used as a method. In the study of [17] they work with a total of 300 images, establishing 4 degrees of ripeness (unripe, yellowish green, mid-ripe, and overripe), then they perform the transformation of the RGB A HSV color space, then they use Otsu's method to remove the background of the images and the ripening factor of the bananas is determined. Then, they extract the texture characteristics and proceed to perform the neural network training with these values: The Hue color space and the ripening factor. Similarly in [18], in case of apples, there was a total of 600 images, 200 for each of the classes (unripe, turning-ripe and ripe), then to remove the background the images are converted to the L*a*b* color space, taking b* as the threshold. Then the characteristics of the images are extracted based on the RGB color space and these are used as input parameters for the training of the artificial neural network. In comparison to these investigations, our approach skips the pre-processing steps for distinctive feature recognition mentioned. In our approach, relevant features of the images are found during the training of our convolutional neural network. The weights of the kernel of each convolutional layer and the weight of our max-pooling layer are updated on each training step in order to automatically find the distinctive features of each class.

V. Experiments

In this section we will describe the process of experimentation, first we will detail the equipment used and configuration, second the training's made with the 2 sets of images of the project and finally the metrics used for the validation of the model.

A. Experimental protocol

For the training of the convolutional neural network was used a macbook pro with an Intel Core i7 2.5Ghz CPU, 16 RAM and an AMD Radeon R9 M370X 2048 MB graphics. In addition, Python was used for the development of CNN, specifically version 3.6 and the keras library. As a project basis we rely on the work of the user markbass, in order to have a convolutional neural network structure implemented. Our project is public in the following link: https://github.com/Fruitapp/FruitApp

We used Keras and Tensor-flow as the main frameworks to implement the proposed Convolutional Neural Network architecture (see Fig 4). Taking advantage of the Keras API, we used the Sequential class, which let us construct the data flow at a high level of logic, focusing of the input and output shape that each layer from our neural network should have.

In this way, the logic of each layer can be treated and tuned as an independent instance. The same sequential model was used in the training process of each experiment.

In order to prepare the training and validation sets for our experiments the flow_from_directory method [14] from the data generator was used. This method automatically identifies the classes to which each image from the dataset pertains according to the directory structure of the dataset. Then it sets it as expected output for training and validation. This method also receives the parameters of the input shape to which all images must be resized before entering the neural network and sets the batch size that will be used per class. In this case, we set the batch size to 10% of the total amount of images per class.

The training process of the experiments was executed using Keras fit_generator [14]. This function enable us to specify the conditions under which the training will take place. The training set was passed as a parameter into this function to indicate which data set must be use for training. A total of 15 epochs were configured for the training process, with a total of 2000 steps per epoch. In each of the number of images evaluated equals the amount specified on the data generator batch size. To prevent running out of input data, a repeat method was configured in keras implementation of the fit_generator method [14].

The validation process was executed using Keras evaluate_generator. This method works with the weights obtained from the training process and the images from the validation set. In a similar way as with the configuration the fit_generator [14], the amount of steps for validation must be passed as a parameter of this function. For these experiments, a total of 1500 validation steps were assigned.

```
Model: "sequential_10"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_28 (Conv2D)           (None, 30, 30, 32)        896
_________________________________________________________________
max_pooling2d_28 (MaxPooling (None, 15, 15, 32)        0
_________________________________________________________________
conv2d_29 (Conv2D)           (None, 13, 13, 64)        18496
_________________________________________________________________
max_pooling2d_29 (MaxPooling (None, 6, 6, 64)          0
_________________________________________________________________
conv2d_30 (Conv2D)           (None, 4, 4, 64)          36928
_________________________________________________________________
max_pooling2d_30 (MaxPooling (None, 2, 2, 64)          0
_________________________________________________________________
flatten_10 (Flatten)         (None, 256)               0
_________________________________________________________________
dense_19 (Dense)             (None, 32)                8224
_________________________________________________________________
dropout_10 (Dropout)         (None, 32)                0
_________________________________________________________________
dense_20 (Dense)             (None, 3)                 99
_________________________________________________________________
activation_10 (Activation)   (None, 3)                 0
=================================================================
Total params: 64,643
Trainable params: 64,643
Non-trainable params: 0
```

Fig. 4. Sequential model

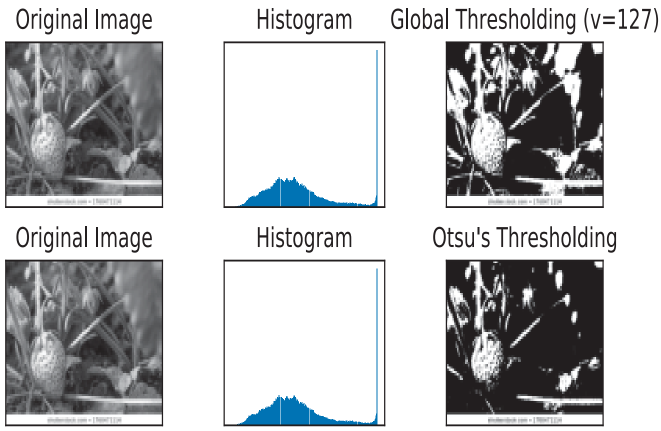

Fig. 5. Otsu's Method Results

B. Image Segmentation

The Otsu threshold selection method is one of the most widely used procedures for performing image segmentation. This method finds the overall optimal threshold using the histogram of a gray scale image, which means, it selects the pixels whose gray level is lower or higher to be assigned as the image background. In the scenario where the histogram is bi modal and the image has uniform illumination, the method will work ideally [19].

Based on the state of the art consulted it was convenient to use the Otsu method to be able to remove the background of the images in an automated way and thus carry out the training, but with the result of having images of the fruits in different contexts, it was concluded that background subtraction results were not optimal and for convenience this method was not used in the pre-processing of the information.

As you can see in Fig 5, since we do not have a uniform background and the pixels of interest in the image are clear, we did not use this technique to remove the background of the images automatically, but rather used the Paint 3D tool, magic selection to achieve our purpose of building a second dataset with the most relevant pixels.

C. Results

Three cases of experimentation with the CNN structure mentioned in Section III-B were taken into account.

1) The first case was the training of the CNN with image set 1 without data augmentation,
2) the second the image set 1 with data augmentation and
3) finally the image set 2 with data augmentation.

For each of these cases, the following metrics were considered:

- **Precision:** measure of the true patterns that are correctly predicted, from the total of predicted patterns in a true class.

$$Precision = \frac{tp}{tp + fp} \quad (1)$$

Where:

tp : true positives
tn : true negatives
fp : false positives
fn : false negatives

- **Accuracy:** measure the ratio of correct predictions over the total number of instances evaluated.

$$Accuracy = \frac{tp + tn}{tp + fp + fn + tn} \tag{2}$$

- **Recall:** used to measure the fraction of positive patterns that are correctly classified.

$$Recall = \frac{tp}{tp + tn} \tag{3}$$

- **Log Loss:**

$$LogLoss = -\sum_{c=1}^{M} y_{o,c} \log(p_{o,c}) \tag{4}$$

Where:

M : number of classes
$y_{o,c}$: binary indicator (0 or 1) if class label c is the correct classification for observation o
$p_{o,c}$: predicted probability observation o is of class c

- **F1:** measure that provides the absolute average between accuracy and recall.

$$F_1 = \frac{tp}{tp + \frac{1}{2}(fp + fn)} \tag{5}$$

As shown in the Table 6a, this experiment obtained the lowest average scores. The highest value of each metric was placed in bold. Without data augmentation, it is really hard for our model to identify significant features in the data set. Considering the size of the data set, a batch of 10 images is relatively low for each step per epoch, which explains why it´s hard for the CNN to extract significant features out of the images. The best performance of this model was achieved with the orange data set. However, when we compare its results with the ones from the other fruits we can see that we are dealing with an over-fitted model. The low loss value obtained and the high precision despite the reduced size of our batch size reaffirms this claim of over-fitting. Taking a look at the data-set of this fruit, we can see see that the similarities of the images per each state show an improper imbalance of the data.

As shown in the Table 6b, this experiment got the best results out of all three. The highest value of each metric was placed in bold. With data augmentation, the data set size is multiplied by a factor of 6 after applying the transformations described in Section III-A, giving a significant improvement on the training and validation of the model. However, the size of the data set limits our predictions on real scenarios. Even though data augmentation increments the amount of images to be evaluated, the data set is not representative enough in relation to the background and context of the fruits. The low loss values obtained on some fruits can be a sign of over fitting of the model. This can be clearly seen on the banana, strawberry and orange results. The model is clearly recognizing distinctive features within our dataset, however the small data set size rises the chance of over-fitting.

	Precision	Accuracy	Recall	Log Loss	F1
Red Apple	0.8737	0.8888	0.8736	1.6030	0.8736
Green Apple	0.8417	0.7777	0.8416	3.8444	0.8416
Banana	0.5555	0.7545	0.6545	7.1245	0.6009
Strawberry	0.6651	0.8888	0.6651	3.0674	0.6651
Orange	**0.9492**	**0.9778**	**0.9708**	**0.5371**	**0.9598**
Average	0.7770	0.8575	0.8011	3.2352	0.7882

(a) First Experiment

	Precision	Accuracy	Recall	Log Loss	F1
Red Apple	0.9573	0.9005	0.9552	1.6502	0.9562
Green Apple	0.9412	0.7831	0.9388	2.7951	0.9399
Banana	**0.9827**	**0.9903**	**0.9811**	**0.2538**	**0.9818**
Strawberry	0.9760	0.9504	0.9749	0.4914	0.9754
Orange	0.9600	0.9787	0.9589	0.2558	0.9594
Average	0.9634	0.9206	0.9617	1.0892	0.9626

(b) Second Experiment

	Precision	Accuracy	Recall	Log Loss	F1
Red Apple	0.9670	0.9089	0.9655	0.00006	0.9662
Green Apple	0.9260	0.8260	0.9198	1.2299	0.9229
Banana	**0.9819**	**0.9912**	**0.9796**	**0.00006**	**0.9807**
Strawberry	0.9443	0.9333	0.9283	0.0004	0.9362
Orange	0.9546	0.9816	0.9529	0.0652	0.9536
Average	0.9548	0.9282	0.9492	0.2591	0.9520

(c) Third Experiment

Fig. 6. Results for experiments

As shown in the Table 6c, even though background removal was applied, this experiment got lower results than than the second one, indicated by the metrics highlighted in bold. However, the results from this experiment were not that far from the previous one, falling back only a percentage point in the average metrics. In a similar way as the previous experiment, the lowest loss values were obtained on the orange, strawberry and banana data sets. In general, the loss values were much lower. The size of the data set could also be a determining factor on why background removal was not that influential on the results. The background removal could also be considered as an influential factor on the loss values obtained, since it makes the images more similar to each other.

VI. Conclusions and Perspectives

This research was based on a 9-layer convolutional neural network and data augmentation for the classification of red and green apples, bananas, strawberries and oranges. It was demonstrated that the use of the data augmentation technique allowed to obtain better results, obtaining an average precision of 96.34% against 77.70% without using this technique. These results indicate that the presented approach can be used to help end users, such as small farmers or people who shop in supermarkets, when selecting their products. However, the limited data set used could show a sign of over-fitting in our model. Future work would involve first, training the proposed convolutional neural network using a data set with a bigger amount of images that present a representative sample in relation to the image conditions (lighting, background color, etc.), and then challenging the implementation of transfer learning to improve model classification rates.

References

[1] W. E. Soto-Silva, E. Nadal-Roig, M. C. González-Araya, and L. M. Plà-Aragonés, “Operational research models applied to the fresh fruit supply chain,” *Eur. J. Oper. Res.*, vol. 251, no. 2, pp. 345–355, 2016.

[2] K. Kipli, H. Zen, M. Sawawi, M. S. Mohamad Noor, N. Julai, N. Junaidi, M. I. Shafiq Mohd Razali, K. L. Chin, and S. M. Wan Masra, “Image processing mobile application for banana ripeness evaluation,” in *2018 International Conference on Computational Approach in Smart Systems Design and Applications (ICASSDA)*, 2018, pp. 1–5.

[3] A. A. Kader, *Postharvest technology of horticultural crops*. University of California Agriculture and Natural Resources, 2002, vol. 3311.

[4] R. Hasan and S. M. G. Monir, “Fruit maturity estimation based on fuzzy classification,” in *ICSIPA*. IEEE, 2017, pp. 27–32.

[5] “Determination of lycopersicon maturity using convolutional autoencoders,” *Scientia Horticulturae*, vol. 256, p. 108538, 2019.

[6] M. J. Villaseñor-Aguilar, J. E. Botello-Álvarez, F. J. Pérez-Pinal, M. Cano-Lara, M. F. León-Galván, M. Bravo-Sánchez, and A. I. B. Gutiérrez, “Fuzzy classification of the maturity of the tomato using a vision system,” *J. Sensors*, vol. 2019, pp. 3 175 848:1–3 175 848:12, 2019.

[7] F. Zakharov, “Fruit ripening and quality relationships,” *Aula-material pedagógico online*, 2015.

[8] C. Crisosto, G. Crisosto, and J. Bermejo, “Applying non-destructive sensors to improve fresh fruit consumer satisfaction and increase consumption,” in *ISHS Acta Horticulturae*, 2016.

[9] A. de Rezende Rocha, “Classifiers and machine learning techniques for image processing and computer vision,” Ph.D. dissertation, University of Campinas, Brazil, 2009.

[10] A. Sobral and A. Vacavant, “A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos,” *Comput. Vis. Image Underst.*, vol. 122, pp. 4–21, 2014.

[11] C. C. Aggarwal, *Neural Networks and Deep Learning - A Textbook*. Springer, 2018.

[12] R. L. Galvez, A. A. Bandala, E. P. Dadios, R. R. P. Vicerra, and J. M. Z. Maningo, “Object detection using convolutional neural networks,” in *TENCON*. IEEE, 2018, pp. 2023–2027.

[13] C. Shorten and T. M. Khoshgoftaar, “A survey on image data augmentation for deep learning,” *J. Big Data*, vol. 6, p. 60, 2019.

[14] F. Chollet *et al.*, “Keras,” https://keras.io, 2015.

[15] I. F. Astuti, F. D. Nuryanto, P. P. Widagdo, and D. Cahyadi, “Oil palm fruit ripeness detection using k-nearest neighbour,” *Journal of Physics: Conference Series*, vol. 1277, p. 012028, jul 2019.

[16] N. M. S. Iswari, Wella, and Ranny, “Fruitylicious: Mobile application for fruit ripeness determination based on fruit image,” in *HSI*. IEEE, 2017, pp. 183–187.

[17] F. Mazen and A. Nashat, “Ripeness classification of bananas using an artificial neural network,” *Arabian Journal for Science and Engineering*, vol. 44, 01 2019.

[18] R. Hamza and M. Chtourou, “Apple ripeness estimation using artificial neural network,” in *HPCS*. IEEE, 2018, pp. 229–234.

[19] J. T. Barron, “A generalization of otsu’s method and minimum error thresholding,” in *ECCV (5)*, ser. Lecture Notes in Computer Science, vol. 12350. Springer, 2020, pp. 455–470.