המכללה האקדמית להנדסה בכרמיאל
בראודה

Capstone Project Phase A

Fruit image Classification and identification of its ripeness
Using Deep Learning

Final Project in Software Engineering (Course 61998)

Project code: 23-1-R-7

Shani Froik, mail: shanifroik@gmail.com

Lior Saghi, mail: liorsaghi@gmail.com

Supervisor: Mr. Ilya Zeldner, mail: zeldner26@gmail.com

Content

Abstract

In recent years Fruit classification has become a common subject in the computer science field. Fruits can be identified by an image using data- it extracts features that are processed in deep learning and gives information based on the image.

In this project we focus on fruit classification with an image, particularly in fruit type and its ripeness, using deep learning. The first stage is to classify the fruits type by using Convolution Neural Network (CNN) to extract features of the fruits from the image, Recurrent Neural Network (RNN) to create appropriate labels, Long Short-Term Memory (LSTM) to integrate the information and classify the fruit type. After we know the fruit type, the second stage is to understand the fruit's level of ripeness. It will be classified by extracting features from the fruit image such as RGB color space and marked boundaries and using a decision tree.

There are many uses to this project. It can help common people choose the finest fruit at the supermarket using the ripeness assessment of the image. It can also prevent people from eating unripe fruit. Another use is to help disabled populations with vision impairments and even blindness giving them a way to overcome their disability.

<u>*Keywords:*</u>

image classification, fruit classification, fruit ripeness, deep learning, neural networks, CNN, RNN, LSTM, decision tree.

1. Introduction

Eating fruit is a recommended way to improve everyone's health and in the long run reduce the risk of disease. Fruits are a good source of minerals, energy, and vitamins and dietary fiber. There are many types of fruits such as: apple, orange, banana, pear, avocado, watermelon, mango and many more. Fruit classification plays an important role in many applications - whether industrial such as factories, agriculture or services such as supermarkets - whether it helped the common man to choose ripe fruit and a person with a disability to identify the fruit he intended to purchase or whether it helped the cashier to automatically select the right type of fruit and determine Priced accordingly.

Determining the condition of the fruit, whether ripe, unripe or scaled, is often performed by a manual sorting method. This method has been around for years, but this method is time-consuming and always involves human intervention. For example, in agriculture, a lot of labor is required to classify the fruits, and despite this, the sorting manual does not give good results in many cases. The same is the case in many other fields and again also for the common man who wants to eat a fruit that is already ripe but has not started the process of rotting.

Therefore, technological techniques are required in order to classify which fruit it is and then determine its condition. In this project, a machine learning-based approach is presented for the classification and identification of fruit by using an image and, after the classification, the assessment of the ripeness of the fruit.

The classification process will be done in several stages when first we will improve the image by Fuzzy Type-II, then CNN will extract its convolution features from the image - it will discover the edges, the color features and then the texture and shape features of the fruit. After that, RNN will be used for labeling Optimal features and LSTM will classify the images using the optimal features that CNN and RNN extracted from the image with the help of deep learning.

After we have found which fruit it is, we will check whether it is ripe, unripe or scaled. This check is done by extracting characteristics from the image of the fruit such as RGB color space, its borders in the image in order to insert this data into a decision tree. The decision tree will give us as an output the ripeness of the fruit.

2. Background and Related Work

2.1 Neural Network

Neural network is a mathematical method that teaches itself to process data. The method is inspired by the cognitive function that occurs in the human brain and simulates a network of neurons. It is a network that acts as a response to certain stimuli so that they take in information, process it and respond. A neural network deals with building virtual models of the nerve cell layers that contain several information units that are linked together and represent input and output. The input is the received information and passes through hidden units that process it, and then an output comes out. Each such processing basically performs a simple mathematical operation, but when there are connections between them, they can perform complex operations.

2.2 Convolutional Neural Network (CNN)

Convolutional Neural Network is a type of neural network that is primarily used to receive image input and process it.

A convolutional network often uses convolution layers, pooling layers and fully connected layers.

2.2.1 Convolution layer

Receives as input a tensor (multilinear function) representing an image. Outputs as a feature map - a tensor with a more abstract image.

This is done cyclically and each tensor input comes out as an output for the input of the next convolution layer.

2.2.2 Pooling layer

Receives a tensor as input and outputs a smaller tensor.

In the pooling layer, a lot of information is lost, but this layer reduces the computational complexity and thus saves processing time.

There are many types of aggregation layers where the main ones are-

- Maximum pooling: selecting the pixel with the maximum value of the elements present in the feature map.
- Average Pooling: Average calculation of the elements present in the feature map.

2.2.3 Fully connected layer

Connects every neuron in one layer to every neuron in another layer. In addition, the image is usually classified by using the SoftMax function according to input from the previous layer.

SoftMax - get as input a value between 0-1 and normalizes it into new vectors that give a total sum of 1.

2.3 Activation function

The activation function calculates the weights of the neuron's input and adds an additional bias to it, thus deciding whether to activate the neuron or not. The purpose of the activation function is to make the neuron's output non-linear.

2.3.1 ReLU Function

The rectified linear unit (ReLU) is the most common activation function used in deep learning models. When the input value is negative the function will return 0, otherwise it will return the positive value back.
The function can be defined as follows: $f(x)=max(0, x)$.

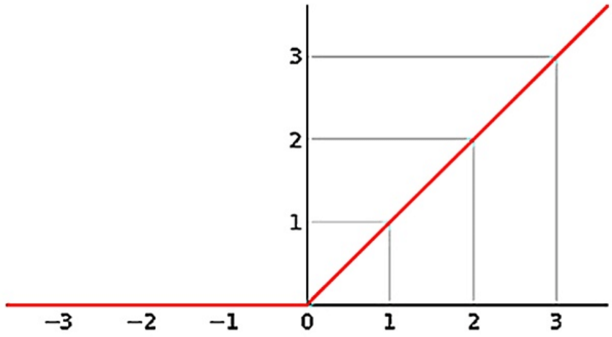


Fig. 1. ReLu activation function

2.4 Flatten layer

This is a layer in a neural network that changes the output from the convolutional layers to a one-dimensional vector which is used as an input to the fully connected layer.

2.5 Softmax

The softmax function transforms the outputs of the neural network into a vector of probability distributions over the input classes. That is, given N classification classes, the softmax operation will return a vector of length N entries, where each entry i in the vector represents the probability that a certain input belongs to class i.

2.6 Recurrent Neural Network (RNN)

It is a type of neural network designed to process continuous data.

An important feature of the RNN is its ability to remember information from previous time steps and use that information to inform its current output.

RNN networks consist of a series of recurrent layers that each have a set of weights and biases that are common to all time steps.

At each time step the input is processed by the recurrent layer and the output is passed to the next time step in addition to the internal state in the network.

2.7 Long short-term memory (LSTM)

Long short-term memory (LSTM) is a type of RNN designed to overcome the limitations of RNNs in processing continuous data when it comes to long-term dependencies.

The network consists of cells similar to RNN but with a more complex structure, where each cell contains a group of gates that control the flow of information into and out of the cell and allow the network to selectively remember or forget information from previous time steps. This allows the network to retain information over a longer period of time.

2.8 Image enhancement techniques

2.8.1 Adaptive Trilateral Contrast Enhancement (ATCE):

According to [16] ATCE is a digital image processing technique used to improve the contrast and visual appearance of an image. It works by adjusting the contrast of the image locally, using a trilateral filter.

The trilateral filter works by smoothing the image, while preserving the edges and other high-frequency features. This helps to reduce noise and other artifacts, while enhancing the overall contrast and clarity of the image. The filter is adaptive, meaning that it adjusts the smoothing strength based on the local image content, resulting in a more natural and visually pleasing enhancement.

In order to simultaneously improve the quality of the image, ATCE uses three different types of image characteristics which are contrast, intensity and sharpness.
This method consists of image contrast manipulation, image sharpness manipulation and image intensity manipulation.
The following contrast manipulation used image segmentation, clipped histogram equalization, image reconstruction and feature extraction.
High-boost filtering is used, which is sharpness manipulation.
Range filtering is used, which is intensity manipulation.

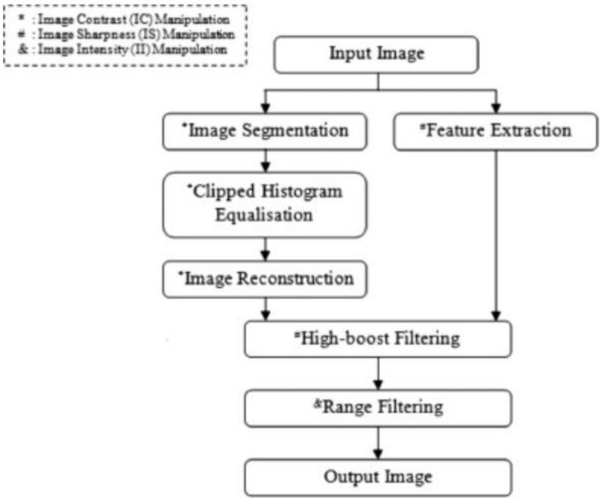


Fig. 2. Adaptive triple contrast enhancement procedure (ATCE)

2.8.2 Brightness Preserving Dynamic Histogram Equalization (BPDHE):

BPDHE is a digital image processing technique used to improve the contrast and visual appearance of an image. It is a variant of histogram equalization, which is a method for adjusting the intensity values of an image such that the resulting image has a more uniform distribution of intensity values.

Histogram equalization works by mapping the intensity values of an image to a new set of intensity values such that the resulting image has a more uniformly distributed histogram. This can improve the contrast of the image by stretching the intensity values across the full range of possible values. However, traditional histogram equalization can sometimes result in over-saturation of the image, causing loss of detail and unnatural looking images.

BPDHE addresses the problem by maintaining the overall clarity of the image while enhancing the contrast. It uses a dynamic mapping function that adjusts the mapping based on the local image content, ensuring that the resulting image has good contrast while maintaining a natural appearance.

2.8.3 Gamma correction (GC):

GC is a digital image processing technique used to encode and decode luminance or chrominance values in digital images. It is designed to compensate for the nonlinear response of some display devices, such as CRT monitors and LCD displays, which can cause images to appear too dark or too light when displayed on these devices.

GC works by adjusting the exponent of the pixel intensity values in the image. In gamma encoding, the pixel values are raised to the power of the gamma value, resulting in a brighter image with more detail in the highlights and shadows. In gamma decoding, the pixel values are raised to the power of 1/gamma, resulting in a darker image with more detail in the midtones.

This method makes it possible to control the brightness of the image, meaning that by using this method the images can appear brighter or darker. In order to improve images using gamma correction, use the following formula:

$$O = \left(\frac{I}{255}\right)^{\frac{1}{\gamma}} * 255$$

I - represents an input pixel value between 0 to 255.
O - represents an output pixel value between 0 to 255.
gamma - gamma value affects the brightness, if the gamma value is less than 1 then the image will be darker and if the gamma value is greater than 1 the image will be brighter. If gamma is equal to 1, the brightness of the image will not change.

2.8.4 Contrast Limited Adaptive Histogram Equalization (CLAHE):

CLAHE is a digital image processing technique used to improve the contrast and visual appearance of an image. It is a variant of adaptive histogram equalization, which is a method for adjusting the intensity values of an image such that the resulting image has a more uniform distribution of intensity values.

Adaptive histogram equalization works by dividing the image into small blocks, or tiles, and applying histogram equalization to each tile individually. This allows the method to adjust the contrast of the image locally, rather than globally, which can result in a more natural and visually pleasing enhancement. However, traditional adaptive histogram equalization can sometimes produce over-saturated images with distorted contrast, particularly in regions with high contrast.

2.8.5 Fuzzy Type-II:

Fuzzy type-II uses membership functions to deal with imprecise or fuzzy data. Fuzzy Type-2 is a generalization of Fuzzy Type-1.
Fuzzy Type-1 uses a single membership function to define the degree of membership of a given element in a set. Fuzzy Type-2 uses a set of membership functions to define the degree of membership of a given element in a set.

Fuzzy Type-2 contributes to a more accurate representation of data uncertainty and fuzzy information.
One of the uses of Fuzzy Type-2 is image processing.
The enhancement of the image using Type-II fuzzy is done by automatically extracting the local atmospheric light and roughly eliminating the details hidden in local detail enhancement.

2.9 Decision Tree

A decision tree is a supervised learning algorithm, which is used for classification tasks. The tree consists of a root node, branches, internal nodes and leaf nodes.
A decision tree starts from the root node, the branches coming out of the root node connect to the internal nodes, which are called decision nodes. According to the available features, the nodes form subgroups which eventually reach the final nodes which are the leaf nodes. The leaf nodes represent all possible outcomes.

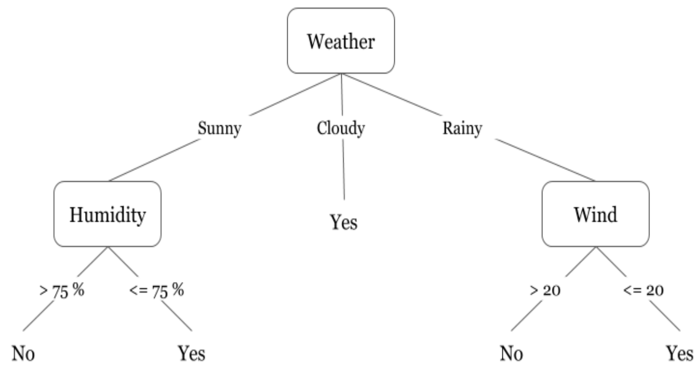


Fig. 3. An example of decision tree

3. Expected Achievements

Our expectation from the project is that we will be able to diagnose a fruit, identify the type of fruit and finally show the user the level of ripeness of the fruit. For fruit classification we will first use the following image enhancement techniques: Fuzzy Type-II.
After that, we will use a model for image classification that combines architectures of CNN, RNN and LSTM as in [1].
Then in order to determine whether a fruit is considered ripe or not we will use a decision tree, where each leaf in the tree symbolizes a label of a certain class (classification according to the labels of the ripeness of the fruit) and the

branches symbolize attributes that lead to those labels so that the model can predict whether the fruit is ripe or not based on the input features.

4. Research

4.1 Process

During the research for the implementation of the project, we realized that we have to divide the model into two stages, where in the first stage we have to build and train a model that will classify images of fruits according to the type of fruit and in the second stage we have to build and train a model that will classify and return information regarding the ripeness of the fruit.
We will detail each step separately.

4.1.1 First stage - Classification fruits images according to the type of fruit:

According to [1], the authors of the article suggest performing a process of image enhancement before using a model that will classify the fruit images. They suggest enhancing the image because a low-quality image can hide important details and image enhancement can help classify the fruit effectively.
There are several techniques for enhance an image, we will mention the techniques we studied:

1. Adaptive Trilateral Contrast Enhancement (ATCE)
2. Brightness Preserving Dynamic Histogram Equalization (BPDHE)
3. Gamma Correction (GC)
4. Contrast Limited Adaptive Histogram Equalization (CLAHE)
5. Fuzzy Type-II

After reading [1] and [22], and after researching the image enhancement techniques, it seems that using Fuzzy Type-II offers improved image quality compared to the other image enhancement techniques (ATCE, BPDHE, GC and CLAHE).
The problem with the ATCE, BPDHE, GC and CLAHE techniques is that these techniques fail to address enhancement problems during histogram calculation, these techniques use a global contrast enhancement that does not improve all areas and the complexity of the calculations is high and this may make it difficult to perform the algorithm.

We will use the following image enhancement algorithm using Fuzzy Type-II:

1. Input a fruit image.
2. Initialize L = 256, where L represents 256 gray level values.

3. Calculate histogram, normalize histogram and calculate the probability of the histogram normalization by using the cumulative distribution function (CDF).
4. Calculating a first-order fuzzy moment (a measure of the center of mass of a set of values with a membership degree between 0 and 1, where membership degree is used as a weight), this value is used to describe a fuzzy set-in term of a single value.

   Calculate the first-order fuzzy moment (m) by using the following formula:

$$m = \sum_{i=0}^{L=1} t(i) * \widehat{h_I}\,(i)$$

$\widehat{h_I}$ - The probability of the histogram normalization

t(i) - Calculate by using the following formula:

$$t(i) = \frac{i}{L-1}$$

i - Gray level i.

5. Calculate membership function by using the following formula:

$$\mu_I(ij) = \frac{L_{ij} - L_{min}}{L_{min} L_{max}}$$

Lmin - Minimum gray level value.

Lmax - Maximum gray level value.

Lij - Gray level of pixel ij, i = 0, 1, 2, …, N and j = 0, 1,2, …, M

6. Calculate histogram.
7. Calculating the probability of a gray level occurring in an image, by counting the number of pixels that have a gray level in the image and then dividing by the total number of pixels in the image.
8. Normalize histogram.
9. Initialize e = 1 - m, ϵ is fuzzy safe entropy, meaning it is a measure of the uncertainty or randomness of a fuzzy set.
10. Obtaining the optimal α and ϒ values by performing the following actions:

    forα= 1 to $\frac{254}{255}$, compute $\lambda = \frac{1-2\alpha}{\alpha^2}$

Using the following membership function:

$$\mu = \begin{cases} \mu_{dark} & \frac{1}{\alpha}x^2 \quad if\, x\, \varepsilon\, [0, \alpha] \\ & 0 \quad if\, x\, \varepsilon\, (\alpha, 1] \\ \mu_{bright} & 0 \quad if\, x\, \varepsilon\, [0, \alpha] \\ \frac{1-\frac{1}{\alpha}(\frac{1-x}{1+\lambda x})^2}{1+\frac{\lambda}{\alpha}(\frac{1-x}{1+\lambda x})^2} & if\, x\, \varepsilon\, (\alpha, 1] \end{cases}$$

We want to segment the image into dark and light pixels, where the meaning of 0 is a dark pixel and the meaning of 1 is a bright pixel, and therefore G describes two fuzzy sets, one dark set and one bright set.

$$G \in \left[0, \frac{1}{255} \frac{2}{255} \frac{3}{255} \frac{255}{255}\right]$$

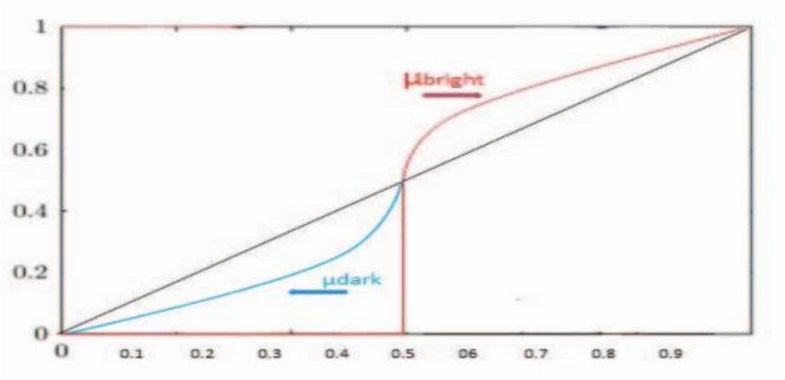


Fig. 4. "The membership function μ_{dark} and μ_{bright} " [From [22] page 4]

Calculation of probability functions for dark and light Fuzzy events, mark these values by M(d) and M(b).
If the current M(d) is greater than the maximum M(d), replace it with the current M(d), same as above for M(b).

11. Modifying the value ϒ according to the following formula:

$$\epsilon = \begin{cases} \frac{p(bright)max}{2} & if m \leq 0.5 \\ \frac{p(dark)max}{2} & otherwise \end{cases}$$

The histogram may be higher if m is less than 0.5 or lower if m is greater than 0.5.

12. To obtain optimal α and ϒ values return to step 10.

According to [1] and [22], it can be seen in the following images that the fuzzy type-II algorithm achieves a better improvement compared to the other techniques.

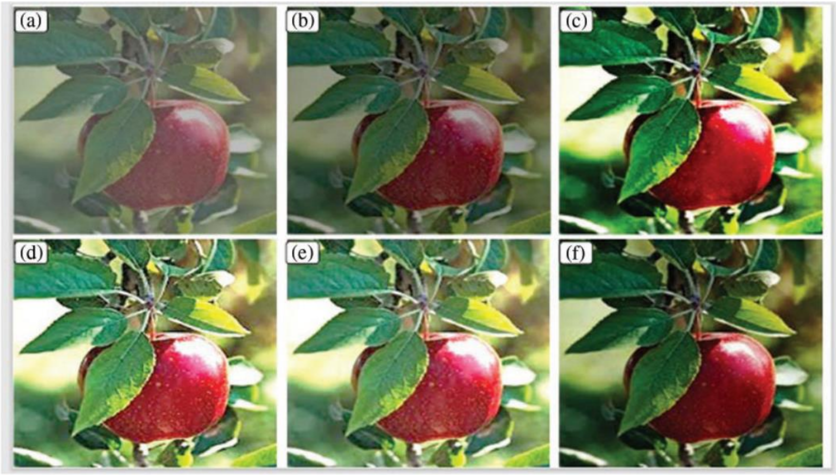


Fig. 5. "Fruit image visual analysis [28–30] (a) input image, (b) (b) enhanced by ATCE, (C) enhanced by BPDHE, (d) enhanced by GC, and (e) by CLAHE, and (f) by Type-II Fuzzy" [From [1] page 9]

Fig. 6. "Results of green apple image, (a) Input (b), (c), (d) (e) and (f) are ATCE, BPDHE, GC, CLAHE and Proposed Scheme output" [from [22] page 6]

4.1.2 Fruit Image Classification model:

There are several models that are used to classify images such as Support Vector Machine (SVM), Feed-forward Neural Network (FFNN), Convolutional Neural Network (CNN), Recurrent Neural Network (RNN) and Adaptive Neuro-Fuzzy Inference System (ANFIS).

After reading [1] and [23], we have seen that a combination of the CNN, RNN and LSTM models gives better performance than the models we mentioned above.

CNN extracts specific features from the fruit images, RNN improves the classification of the labels according to coarse and fine categories, LSTM defines the optimal parameters during fruit classification.

After input the enhanced fruit image to the CNN network, convolution layers integrating a RELU activation function were used to extract the features, pooling layers used to reduce the computation time, a flatten layer to convert the extracted features into a one-dimensional vector and a fully connected layer to combine the extracted features.

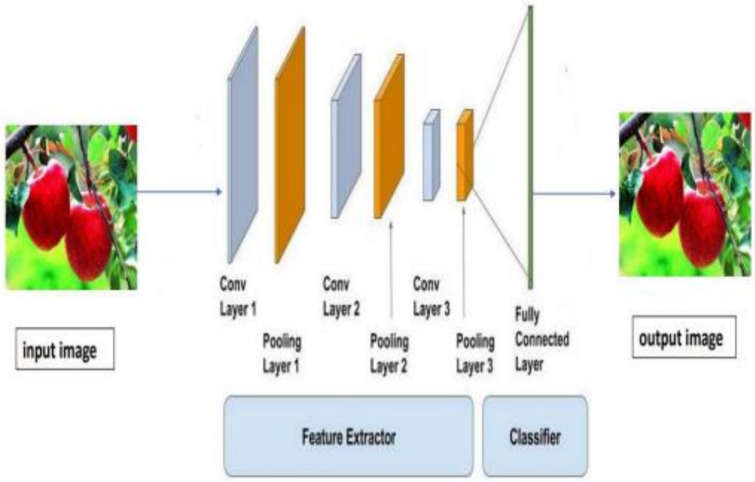


Fig. 7. “Fruit image feature extraction using CNN” [From [23] page 6]

The combination of CNN -RNN -LSTM means that the CNN is used to obtain optimal features, RNN is used to obtain labels and speed up the classification rate and LSTM incorporates a memory cell in order to deal with the vanishing gradient problem.
The RNN is integrated in the model with the CNN to overcome the limitations of the CNN in fruit classification.

Features extracted using CNN and labeled using RNN are fed into LSTM for fruit classification.

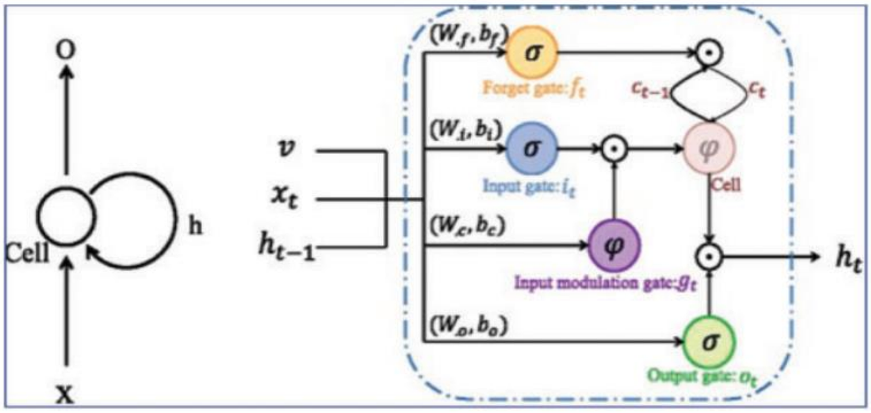


Fig. 8. “RNN (left) and LSTM (right) collaboration” [From [1] page 12]

<u>Details of the stages of the model:</u>
Step 1 - Features will be extracted using the convolution layer, linear layers and fully connected layers of CNN. The extracted properties are based on strength, texture and shape.
Step 2- The features will be labeled by "coarse" and "fine" categories using RNN.
Step 3 - A filtering process used by the LSTM is done and with the help of soft-max-layer the fruits are classified.

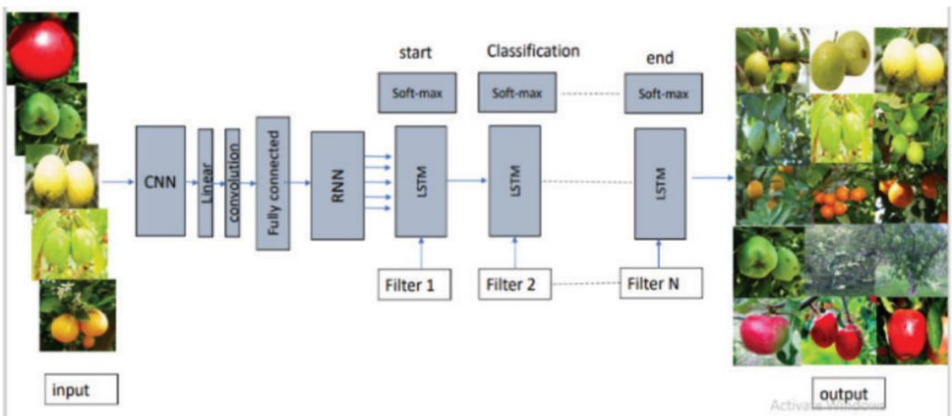


Fig. 9. “Fruit classification model” [From [1] page 13]

To measure the difference between the predicted output of the model and the actual output, a loss function is used.
The purpose of the training model is to minimize the loss function so that the predictions made by the model will be closer to the actual output.
Therefore, in this model we will use the following loss function:

$$Loss = -\frac{1}{N}\sum_{i=1}^{N}\left(\sum_{t=1}^{T}\sum_{j=1}^{C+F} 1\{x_t^i = j\}\ log p_j\right)$$

This loss function will consider the truth labels (T) among all labels (N) and then, the model will use this model to make predictions for the image labels which are defined as "coarse" and "fine" labels (C+F).

Where "coarse" labels represent more general categories of fruits, for example the coarse labels in our model represent "apple", "banana", "orange", etc.
"Fine" labels represent more specific labels. For example, if it is apples, the dataset can contain several types of apples and the subtle labels in our model represent the type of apple, such as "Delicious Red", "Granny Smith", "Fuji", "Pink Lady", etc.

We will use the F measure to evaluate the accuracy of the fruit classification, where a value of 1 is the best classification result and 0 is the worst classification result.
F measure is defined as follows:

$$F\ measure = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Precision - the ratio of the true positive cases to the total number of predicted positive cases (true positives and false positives). Gives information about what the rate of positive identifications is correct.

Recall - the ratio of the true positive cases (the positive and correctly predicted cases) to the total number of actual positive cases (true positives and false negatives). This gives us information about what proportion of correct cases are correctly identified.

Precision and recall are defined as follows:

$$Precision = \frac{T_P}{T_P + F_P}$$

$$Recall = \frac{T_P}{T_P + F_N}$$

<u>4.1.3 Architecture performance comparison</u>:
According to [1] and [23], When classifying images of fruits, an architecture that combines CNN, RNN and LSTM shows high accuracy compared to other architectures.
For our system, we would like the fruit classification to be done in the most accurate way, so we chose this model which combines CNN, RNN and LSTM.

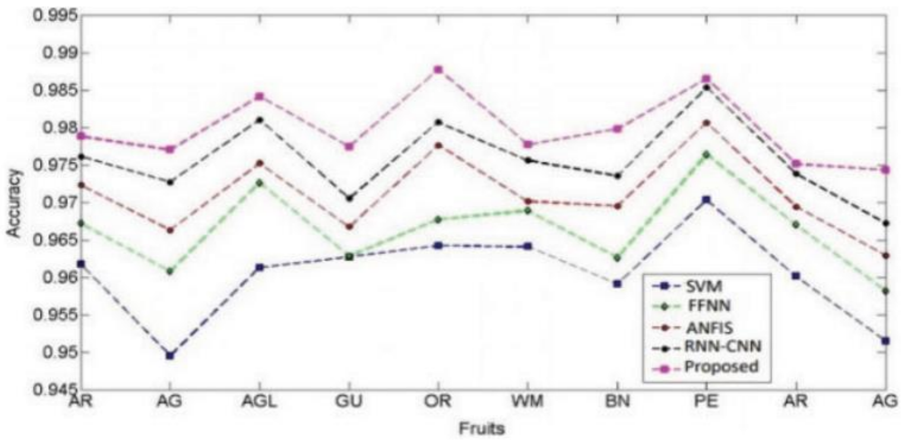


Fig. 10. “Accuracy analysis” [from [1] page 14]

4.1.4 Second stage - Classification fruits images according to ripeness level:

After reading the article [2] we realized that there are several options to classify the fruit according to its ripeness and we checked what would be the best method that would achieve the best classification results. The possible methods:

1. Naive Bayes classifier:

   Naive Bayes classification is a collection of classification methods based on Bayes' law and the assumption that there is no dependence between the properties of the classified objects when their classification is already known.

   Using Bayes' Law, we find the probability that A will happen, given that B has happened. $P(A|B) = \frac{P(B|A)P(A)}{P(B)}$ Here, B is the evidence and A is the assumption. The assumption is that the attributes are independent - that is, one attribute does not affect the other. That's why it's called naive.

2. Artificial Neural Network (ANN):

   Artificial neural network is a mathematical model that includes calculations that simulate brain processes similar to the processes that occur in a natural neural network in the human brain. (There is an additional explanation above in the background for Neural Network).

3. Decision Tree:

   Decision tree is from the field of statistics, like a tree used as a prediction model.

A decision tree is a binary tree that maps observations to an item and creates inferences about the target value of the item. The tree consists of decision nodes where a condition is tested on a certain characteristic of the observations and leaves that contain the predicted value for the observation corresponding to the route that reaches them in the tree. (There is an additional explanation above in the background).

After testing and classification as it appears in the article on a database of images and calculation of an accuracy formula ($\text{Accuracy} = \frac{(TP + TN)}{(TP + FP + TN + FN)}$) it can be concluded that a decision tree achieved the best diagnosis of ripe and unripe fruit.

Looking at the three techniques in the three types of fruit condition - unripe, ripe, scaled, it can be concluded that a decision tree gave the best results in distinguishing what is a ripe fruit and what is an unripe fruit.

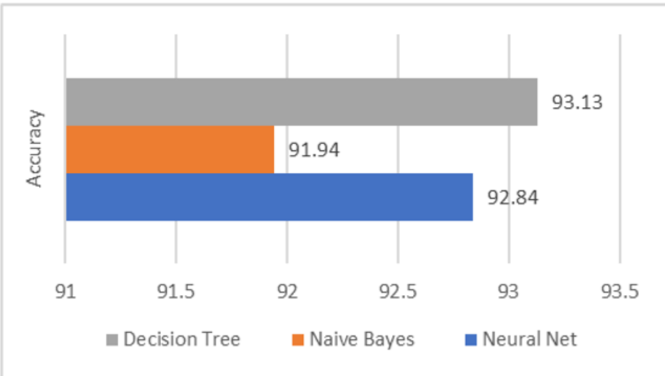


Fig. 11. (From article [2] page 4)

Although a decision tree is behind ANN in the context of scaled condition accuracy and behind Naive Bayes for the condition of an unripe fruit, but still in accuracy it can be seen that a decision tree has the best identification of the condition of the fruit.

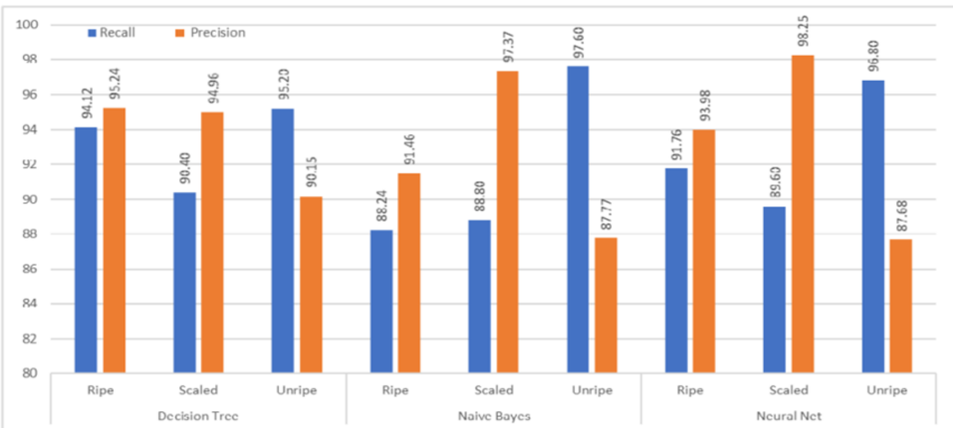


Fig. 12. (From article [2] page 4)

Therefore, choosing a decision tree will be the preferred choice for identifying whether the fruit is ripe or not.

In a decision tree, to classify fruits as ripe or unripe, we need to extract different features from the images such as color, shape, texture in order to have nodes in the tree.

The decision tree may contain many features and extracting these features from images will require some image processing techniques such as color analysis, object detection, and more.

A simple tree for example:

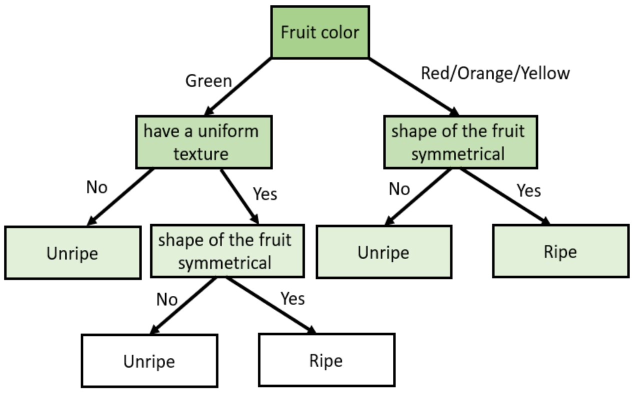


Fig. 13. An example of a simple tree

In fruit identification, color is an important feature because it allows us to evaluate the level of ripeness of the fruit and also indicates defects that may be in the fruit.

We will measure the color level using the basic color components - red, green and blue by using equations –

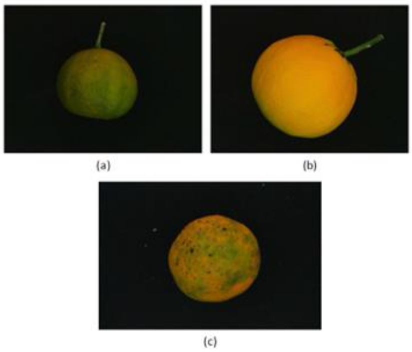
$$\mu_r = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} f_r(i,j)$$

$$\mu_g = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} f_g(i,j)$$

$$\mu_b = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} f_b(i,j)$$

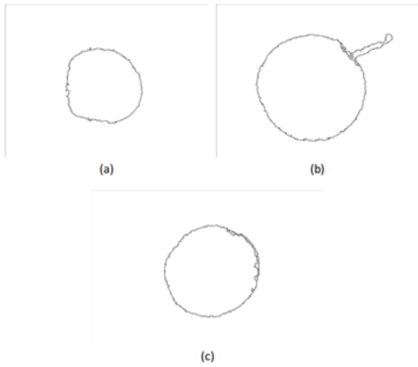
(From [2] page 3)

In addition, to extract the feature of the fruit borders, a border/interior pixel classification (BIC) is used.



Original images

Fig. 14. (From [2] page 3)



After BIC

Fig. 15. (From [2] page 3)

After extracting the features, the tree must be trained and the decisions it makes must be evaluated by measures such as precision, recall, and more.

If the result that comes out of the tree will not be as good as we expected, additional features must be added or some of the existing ones must be replaced.

In order to show the distance between predicted labels and actual labels in the decision tree, we will use the Confusion matrix.

We will use the following matrix:

	Actual True	Actual False
Predicted True	True Positive	False Positive
Predicted False	False Negative	True Negative

True positive (TP) - cases that are expected to be positive and are actually positive.

True negative (TN) - cases that are expected to be false and are actually false (do not belong to a certain class).

False Positive (FP) - cases that are expected to be positive (belong to a certain class) but are actually negative (do not belong to the class).

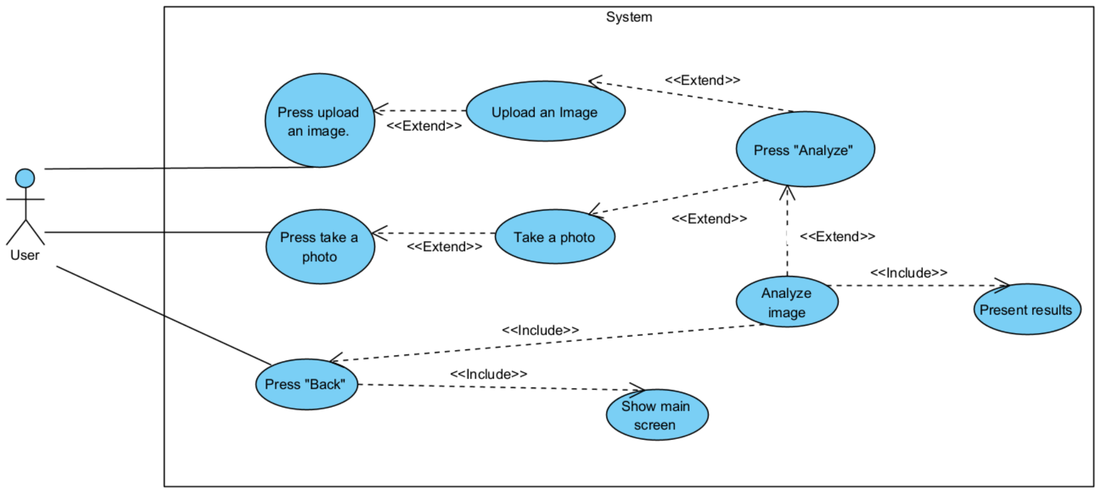
False negative (FN) - cases that are expected to be negative (do not belong to a certain class) but are actually positive (belong to a class).

4.2 Product

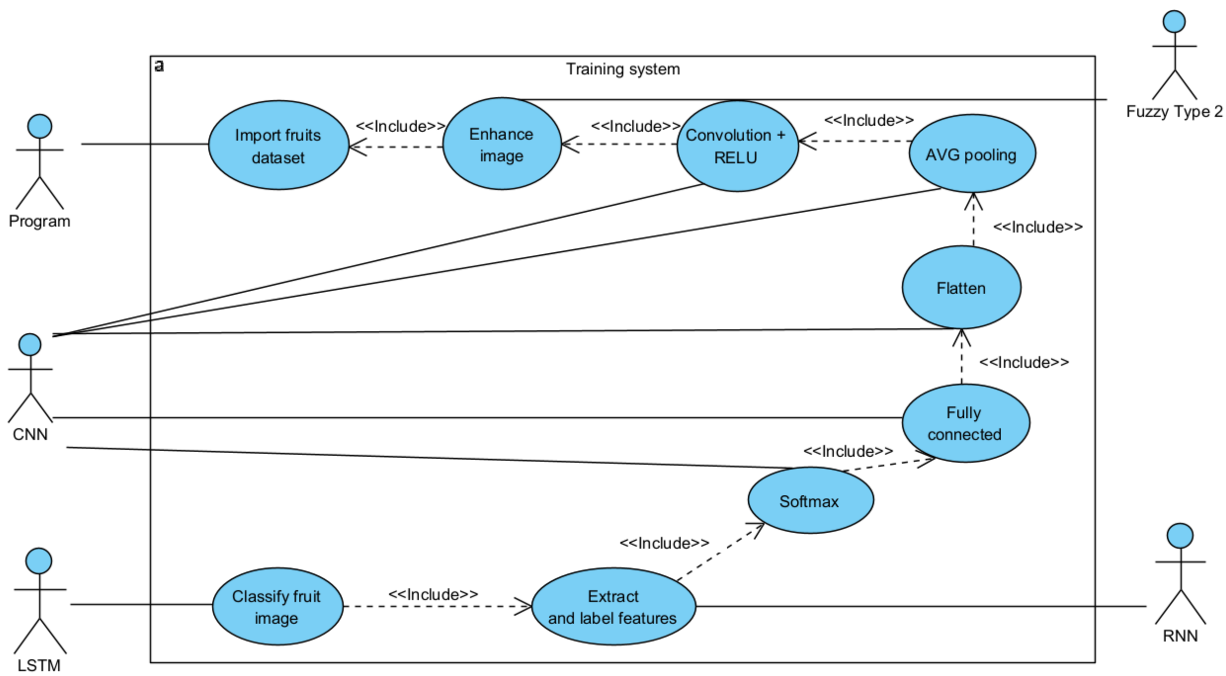
4.2.1 UMLs

4.2.1.1 Use Case:

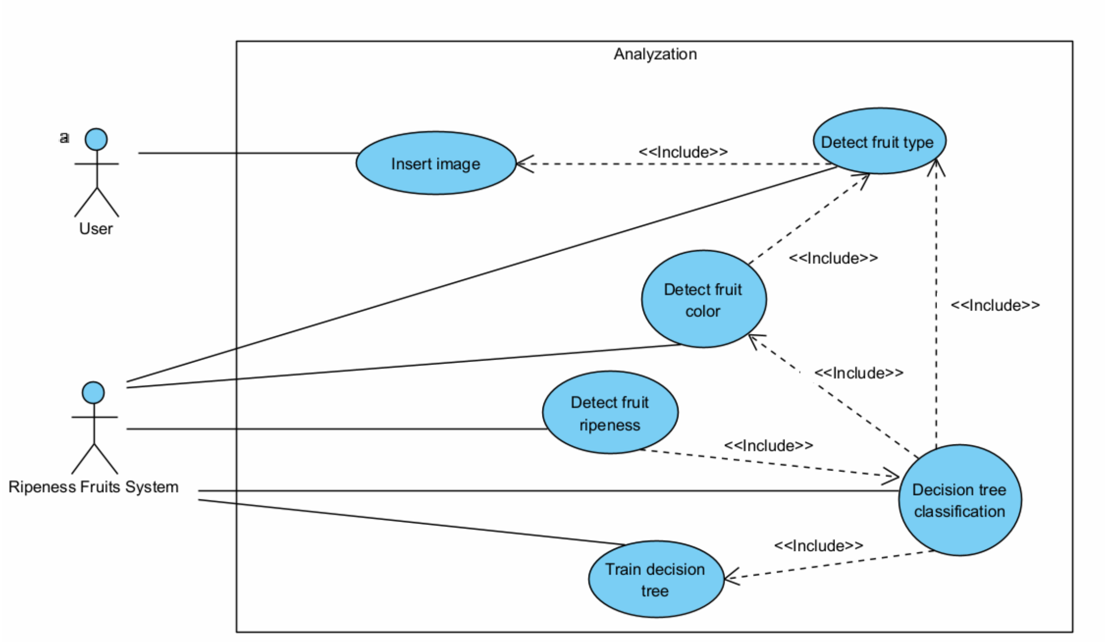
Ripeness Fruits System:



Training:

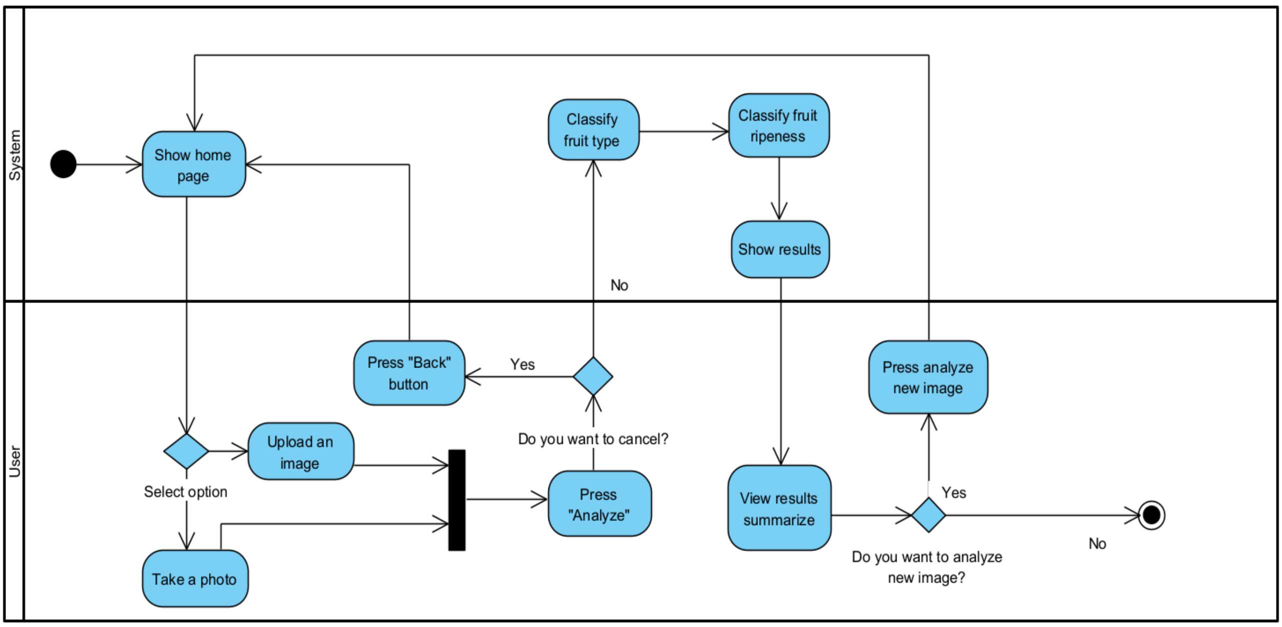


Analyzation:

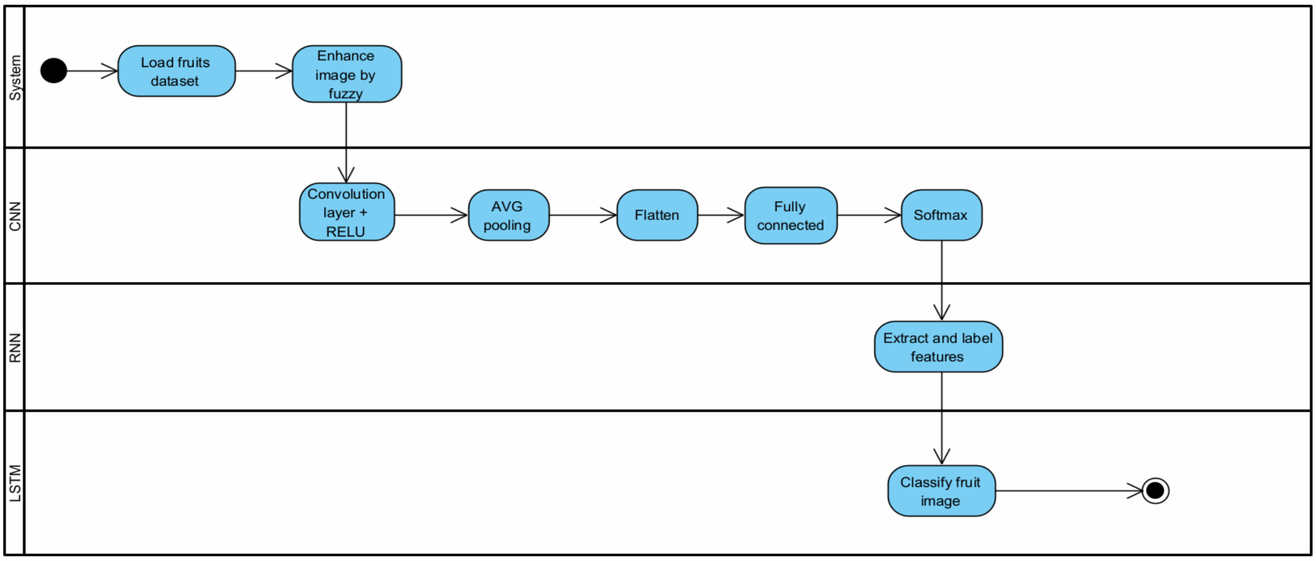


4.2.1.2 Activity:

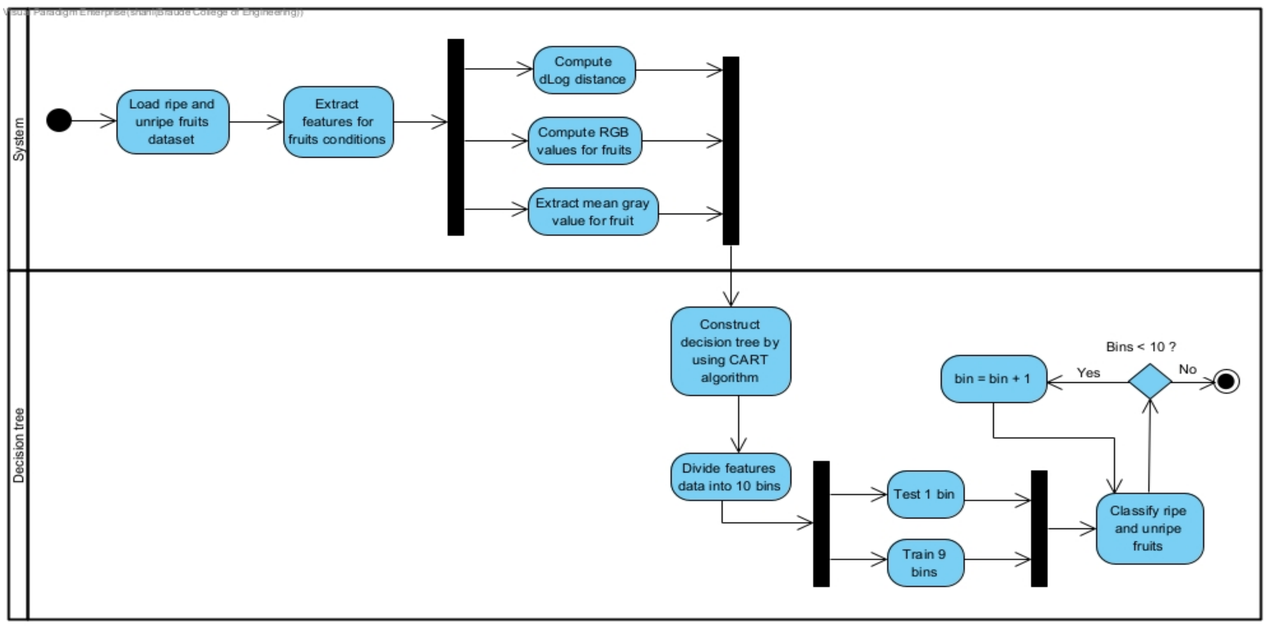
Ripeness Fruits System:



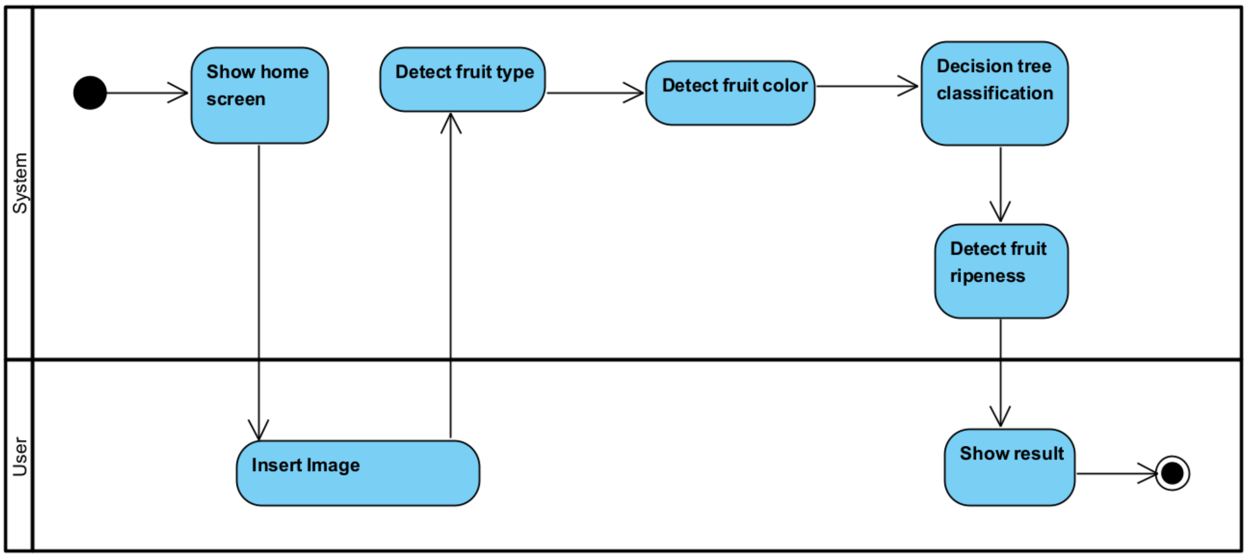
Training classifies fruits model:



Training ripe and unripe fruits model:

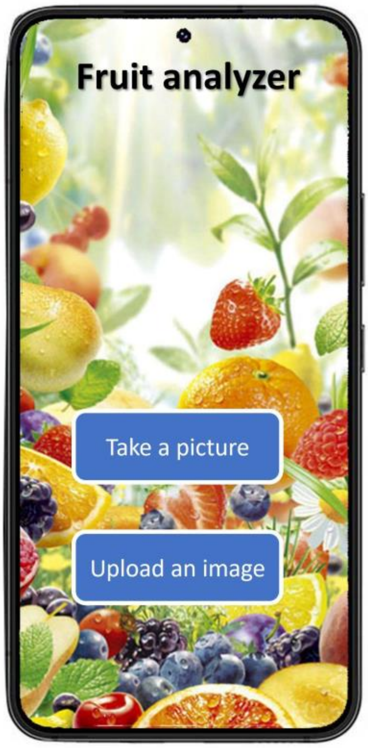


Analyzing:



4.2.2 Screens

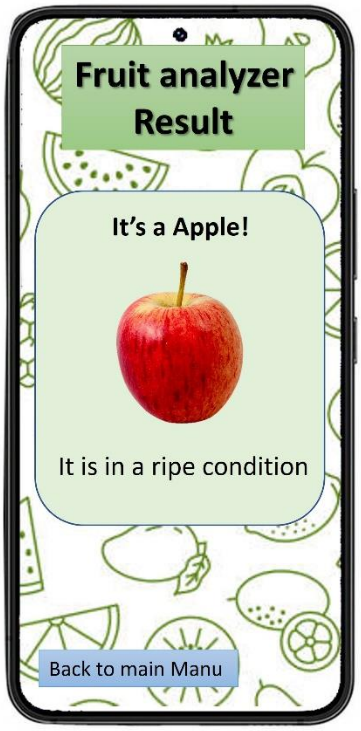
4.2.2.1 Main Screen:



4.2.2.2 Take a picture screen:

4.2.2.3 Analyzation Screen:

4.2.2.4 Result Screen:



5. Evaluation Plan

Test ID	Test Description	Expected results
1	User uploads a picture that is not of a fruit	System shows a popup error message. Output: "No fruit detected".
2	User uploads a file that is not an image	System shows a popup error message. Output: "Please only upload a picture of a fruit".
3	User uploads a picture of a ripe apple	System analyzes the image and shows that it is an apple and it is ripe
4	User uploads a picture of an unripe apple	System analyzes the image and shows that it is an apple and it is unripe
6	User uploads a picture of a scaled apple	System analyzes the image and shows that it is an apple and it is scaled
7	User takes a photo of ripe orange	System analyzes the image and shows that it is an orange and it is ripe
8	User takes a photo of unripe orange	System analyzes the image and shows that it is an orange and it is unripe
9	User takes a photo of a scaled orange	System analyzes the image and shows that it is an orange and it is scaled
10	User takes a photo that is not of a fruit	System shows a popup error message. Output: "No fruit detected".

6. References

[1]Harmandeep singh Gill, Youseef Alotaibi, Osamah Ibrahem Khalaf, Saleh Mohammed Alghamdi, Fruit Image Classification Using Deep Learning, 2022.

[2]Abdul Wajid, Niraj Kumar Singh, Pan Junjun, Muhammad Ali Mughal, Recognition of ripe, unripe and scaled condition of orange citrus based on decision tree classification, 2018.

[3]https://www.geeksforgeeks.org/activation-functions-neural-networks/

[4]https://www.kaggle.com/code/dansbecker/rectified-linear-units-relu-in-deep-learning/notebook

[5]https://www.simplilearn.com/tutorials/deep-learning-tutorial/convolutional-neural-network

[6]https://www.tutorialspoint.com/keras/keras_flatten_layers.htm

[7]https://deepai.org/machine-learning-glossary-and-terms/softmax-layer

[8]https://www.pinecone.io/learn/softmax-activation/

[9]https://www.ibm.com/topics/recurrent-neural-networks

[10]https://colah.github.io/posts/2015-08-Understanding-LSTMs/

[11]https://www.sciencedirect.com/science/article/abs/pii/S0952197620302487

[12]https://www.ibm.com/topics/decision-trees

[13]https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html

[14]https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c

[15]https://www.ibm.com/topics/neural-networks

[16]https://www.sciencedirect.com/science/article/pii/S0167865514002852?casa_token=rFVPVZdyYo8AAAAA:JzohJizy-IBJckmv1KHaQID_1smOlMQGGr-R77WIOph_HPWMBZuFAwEwA5XHzwfTm9JBZb8-Zvg

[17]https://ieeexplore.ieee.org/abstract/document/4429280

[18]https://lindevs.com/apply-gamma-correction-to-an-image-using-opencv

[19]https://www.sciencedirect.com/topics/earth-and-planetary-sciences/artificial-neural-network

[20]https://www.sciencedirect.com/science/article/abs/pii/S1568494614004438

[21]https://philarchive.org/archive/ALKCOA

[22]Dr.Harmandeep Singh Gill, Dr.Baljit Singh Khehra, Er.Bhupinder singh Mavi, Fruit images Visibility enhancement using Type-II Fuzzy, 2021.

[23]Dr.Harmandeep Singh Gill, Dr.Baljit Singh Khehra, Fruit Image classification using Deep Learning, 2022.

7. Git link

https://github.com/shaniFr/fruit-ripeness