**TomatoX: Vision-based Judgment of Tomato Maturity using Extreme Learning Machines**

Presented to the Faculty of the

Division of Natural Sciences and Mathematics

University of the Philippines Visayas

Tacloban College

In Partial Fulfillment of the Requirements

for the Degree of

Bachelor of Science in Computer Science

Presented by:

Roberto D. Saldaña Jr.

2009-09630

November 2013

**Approval Sheet**

This project entitled “TomatoX: Vision-based Judgment of Tomato Maturity using Extreme Learning Machines” by Roberto D. Saldaña Jr. was presented to the Faculty of the Natural Sciences and Mathematics Division of the University of the Philippines Visayas Tacloban College.

This project is hereby accepted and approved in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science.

John Paul T. Yusiong

Adviser

Date

**Acknowledgement**

This research has been made possible through the help of certain individuals. I would like to take this opportunity to extend my deepest gratitude to the people who have helped me in every step of the way.

First of all, I would like to thank the Lord God for everything; may it be good or bad, they have in one way or another contributed to the success of this research. Thank you for the people who have helped me complete this research.

To my family, thank you for the never ending support in my endeavors in life. Thank you for the guidance and constant source of motivation and inspiration.

To my adviser, Prof. John Paul T. Yusiong, thank you for the guidance and support. Thank you for trusting in my capabilities even if sometimes I am in doubt. Thank you for giving me the idea of using Extreme Learning Machine in Tomato Classification.

To kuya Jake and ate Abby, thank you for the resources needed to complete this research. Thank you for the dataset, Tomato Classification and ELM papers.

To my classmates and friends, thank you for all the encouragements. Thank you for making this research memorable.

**Abstract**

Tomato is one of the economically important fruit in the world. Major supplies of tomatoes come from farms and greenhouses. The traditional way of sorting tomatoes are done by human graders. However, this procedure has a lot of shortcomings. Manual sorting is a very time-consuming, tedious, and expensive task. Additionally, the human identification of colors is affected by many factors like brightness, intensity, etc. A lot of research has been conducted on the automation of this procedure. This research proposes the use of Extreme Learning Machines (ELM) in sorting and classifying tomatoes. The study utilizes color features as input to the ELM Network. The study proposes the use of L\*, a\*, Hue, and red-green difference. The experiment results show that ELM can be an efficient method in the classification of tomatoes. Also, the training time for ELM is fast compared to other researches. The proposed color features also performed better than other color feature combinations.

**Keywords:** Tomato Classification, Extreme Learning Machine, Image Processing, Color Models

**Contents**

**List of Figures 3**

**List of Tables 5**

**1 Introduction 6**

1.1 Tomato 6

1.1.1 Tomato Maturity Stages 8

1.2 Image 9

1.2.1 Grayscale Image 9

1.2.2 Binary Image 10

1.2.2.1 Otsu Thresholding 10

1.3 Image Processing 13

1.3.1 Scaling 13

1.3.2 Cropping 14

1.4 Color Space 14

1.4.1 RGB 15

1.4.2 HSI 16

1.4.3 CIE L\*a\*b\* 16

1.5 Neural Networks 17

1.6 Extreme Learning Machines 19

1.6.1 Introduction of a Regularization Coefficient 21

**2 Literature Review 22**

**3 Statement of the Problem 27**

**4 Objectives 28**

**5 Proposed Approach 29**

5.1 Process Flow 29

5.2 Tomato Image 29

5.3 Scaling 30

5.4 Blue Channel 30

5.5 Grayscale Conversion 31

5.6 Masking 31

5.7 Cropping 34

5.8 Feature Extraction 34

5.9 ELM Training 35

5.9.1 Input Layer 35

5.9.2 Hidden Layer 35

5.9.3 Output Layer 36

5.9.4 ELM Training 36

5.10 Tomato Classification 37

**6 Experiments and Results 39**

6.1 System Specifications 39

6.2 Experimental Settings 39

6.3 Experimental Results and Discussion 40

**7 Conclusion and Future Works 48**

**References 50**

**Appendix: User’s Manual 53**

**List of Figures**

1.1 : Ripe Tomatoes 7

1.2 : USDA tomato ripening stages 8

1.3 : Original image converted into grayscale 10

1.4 : A 6-level grayscale image and its histogram 12

1.5 : Result of Otsu’s method 12

1.6 : Example of Otsu method in use 12

1.7 : Original image and the image decreased by 25% 13

1.8 : Original image and cropped image 14

1.9 : Color subspace of RGB and CMY color models 15

1.10 : The RGB color model 15

1.11 : The neuron model 17

1.12 : Multilayer perceptron 18

1.13 : Single-hidden layer feed-forward network 19

5.1 : The pipeline of TomatoX 29

5.2 : Tomato image 30

5.3 : Blue Channel 30

5.4 : Grayscale image 31

5.5 : Image Histogram 31

5.6 : Calculating the histogram 32

5.7 : Binary Mask 32

5.8 : Masking Operation (a) Original Image (b) Output of Masking 33

5.9 : Cropped Image 34

5.10 : 64 x 64 Image 34

5.11 : ELM Network Structure of TomatoX 35

5.12 : ELM Network with sample values 38

6.1 : Results for the fixed and random data 41

6.2 : Results for varying the number of hidden nodes for the test set 43

6.3 : Results for varying the number of hidden nodes for the train set 43

6.4 : Results for varying the values of the regularization coefficient (C) 45

6.5 : Results for varying the input features 46

**List of Tables**

5.1 : Color Features extracted from Figure 5.8 35

6.1 : ELM Standard parameter values 40

6.2 : Results for fixed and random data 41

6.3 : Varying the number of hidden nodes 42

6.4 : Results of varying the number of hidden nodes for the test set 42

6.5 : Results of varying the number of hidden nodes for the train set 43

6.6 : Results of varying the values of the regularization coefficient 44

6.7 : Varying the input features 46

6.8 : Results of varying the input features 46

6.9 : Comparison to TotoBee [15] 47

**Chapter 1**

# Introduction

# Tomato is arguably one of the economically important fruit in the world. This is because exports of tomato and its lateral products such as ketchup and tomato sauce have considerable income. In turn, tomato is the most investigated member of the fleshy fruit regarding fruit development and ripening [25].

# Before transporting to the market, tomatoes are sorted to several grades. Different tomato products have distinct requirements for maturity to achieve standards. Therefore, tomato maturity is one of the important factors associated with the quality of processed tomato products.

# Traditionally, the sorting of tomatoes is accomplished by human power. However, this type of scheme has many shortcomings like cost, time and inconsistency. Thus, in recent years, numerous studies have been conducted using computer techniques for assessing the quality of agricultural crops.

# Tomato maturity can be related to the surface color. Thus, evaluating the maturity level by visual analysis is a feasible approach. With the fast advancement of technology, a number of image classification techniques have been made. Some of the most popular techniques are the use of neural networks (NN) and support vector machines (SVM) [1] [5]. However, both of these popular learning techniques face some challenging issues such as intensive human intervention and slow learning speed.

# Huang et al. [9] introduced the extreme learning machines (ELM) which overcomes some of the issues faced by both NN and SVM. This study will explore the feasibility of using ELM in the classification of tomatoes.

# Tomato

# Tomato, scientifically known as *Lycopersicon esculentum,* is a nutritious fruit known as a vegetable. The English word *tomato* comes from the Spanish word, *tomate*, derived Nahuatl (Aztec language) word, *tomatl*. A member of the deadly nightshade family, tomatoes were erroneously thought to be poisonous (although the leaves are poisonous) by Europeans who were suspicious of their bright, shiny fruit [23]. Having originated in America, tomato spread around the world and many varieties are now widely grown.

# Tomatoes are consumed in diverse ways, including eaten raw, as ingredient in many dishes and sauces, and in drinks and pastes. Although botanically it is a fruit, it is considered a vegetable for culinary purposes which may have caused some confusion. The fruit is rich in nutrients such as lycopene, which is beneficial to health.

# tomato history, recipes, vegetable, fruit, receipts Depending on the market and production area, tomatoes are harvested at stages of maturity ranging from physiological maturity (mature-green stage) through full-ripe [19]. The maturity process can be related to the color of the surface starting from green to red coloration. Figure 1.1 shows an example of ripe tomatoes.

# Taken from [23]

# Figure 1.1: Ripe Tomatoes

# The major sources of tomatoes are commercial farms and greenhouses. Traditionally, the classification and sorting of tomatoes is done using human labor. This process is very time consuming, tedious and expensive. In fact, the high labor cost has been the main obstacle in the expansion of greenhouses. Additionally, human identification of colors is affected by sensations like brightness, intensity, lightness and vividness*,* thus in many cases color definition is a matter of subjective interpretation.

# Automation of quality control is highly significant because saving time and expenses is always a necessity in industrial applications [5]. Moreover, it could improve the quality of the product, abolish inconsistent manual evaluation, and reduce dependence on available manpower. This could be accomplished through the aid of machine vision.

# 1.1.1 Tomato Maturity Stages

# http://www.agrofresh.com/assets/images/smartfresh/tomatoes/tomato_ripening_stages.jpgThe surface color of a tomato is the most important external characteristic to assess maturity and is a major factor in the consumer’s purchase decision. Degree of ripening is usually estimated by color charts [12]. Based on the external color, the United States Department of Agriculture (USDA) established six ripening stages reflecting human ability to differentiate ripeness. Figure 1.2 shows the different stages of a tomato as described in the USDA chart. It can be seen that the surface color changes from green to red.

# Taken from [21]

**Figure 1.2:** USDA tomato ripening stages

The summary of the description for each stage according to [6] is discussed below.

1. **Green / Stage 1 –** The surface is completely green in color. The shade of the green color may vary from light to dark.
2. **Breakers / Stage 2 –** There is a definite “break” in color of not more than 10%.
3. **Turning / Stage 3 –** The change from green to tannish-yellow, pink, red or a combination is more than 10% but not more than 30%.
4. **Pink / Stage 4 –** More than 30% but not more than 60% of the surface shows pink or red color.
5. **Light Red / Stage 5 –** More than 60% of the surface shows pinkish-red or red, provided that not more than 90% of the surface is red.
6. **Red / Stage 6 –** More than 90% of the surface is red.

# Image

It is not easy for humans to imagine what the world looks like to a bat, eel, or mole. Even the word “imagine” demonstrates the problem. The root word “image” implies a picture, or scene, constructed inside the mind [18]. Images allow people to communicate without using words. In every aspect of human life, images are always present, i.e. we always tend to construct images in our minds based on some information. For example, when we are listening to music, we tend to visualize and imagine the lyrics.

With the rapid increase in technology, image entered the world of 1s and 0s, and became “digital”. Digital Images are electronic representations of images that are stored in a computer [24]. They are composed of pixels (picture elements), each representing a color at a single point in the image.

# Grayscale Image

Grayscale Images are images whose pixel values range from 0 to 255, which is equivalent to 8 bits. The values represent the different shades of gray. Grayscale images contain less amount of information. Also, with less range of value and memory consumption, grayscale images ease the computation and memory needs of any image processing technique.



# Taken from [22]

# Figure 1.3: Original image converted into grayscale

# There are several methods to convert color images to grayscale. One of the most common methods is the luminance method as described in equation (1). Figure 1.3 shows an example of a color image converted into a grayscale using the luminance method.

|  |  |
| --- | --- |
|  | **(1)** |

# 1.2.2 Binary Image

# Binary images are images that have been quantized to two values, usually denoted by 0 and 1, but often with pixel values 0 and 255, representing black and white [2]. These images are used in many applications because they are very simple to process. Binary images are often used in digital image processing as masks. Converting a grayscale image to binary is a common image processing task. The conversion is typically done by using a threshold, i.e. pixels with value above the threshold are set to 1 (or 255), while the rest are set to 0.

# Otsu Thresholding

# Otsu’s Method, named after its inventor Nobuyuki Otsu, is a clever method of binarization. The algorithm assumes that the image contains two classes of pixels, the foreground and background. Otsu’s thresholding method involves iterating through all possible threshold values and calculating a measure of spread for the pixel levels of each side of the threshold, i.e. the pixels either falls in the foreground or background [16]. The aim is to find the threshold value that minimizes the within-class variance, i.e. the sum of the foreground and background spreads is at minimum. The within-class variance is computed using equation (2).

|  |  |
| --- | --- |
|  | **(2)** |

# The problem of minimizing the within-class variance can be expressed as maximizing the between-class variance. The threshold with the maximum between-class variance has the minimum within-class variance. The between-class variance is far quicker to calculate; therefore it is a much better approach to use. The between-class variance is computed using equation (3). The remaining equations are used in computing the class probabilities and class means.

|  |  |
| --- | --- |
|  | **(3)** |
|  | **(4)** |
|  | **(5)** |
|  | **(6)** |
|  | **(7)** |

# Figure 1.4 shows a 6 x 6 image with 6 grayscale levels. The histogram for the image is shown next to it. Figure 1.5 shows that the threshold value equal to 3 has the lowest sum of weighted variances. Therefore, this is selected as the optimal threshold. Pixel values less than 3 are considered as background, while the rest are foreground. Figure 1.6 shows an example of binarizing a grayscale image using Otsu’s method.

# 

# http://www.labbookpages.co.uk/software/imgProc/files/otsuResult.pngTaken from [16]

# Figure 1.4: A 6-level grayscale image and its histogram

# Taken from [16]

# http://www.labbookpages.co.uk/software/imgProc/files/otsuExamples/harewood_BW.pnghttp://www.labbookpages.co.uk/software/imgProc/files/otsuExamples/harewood.jpgFigure 1.5: Result of Otsu’s method for Figure 1.4

# Taken from [16]

# Figure 1.6: Example of Otsu method in use

# Image Processing

Image processing is the manipulation of an image to produce either an image or a set of characteristics/parameters related to the image. According to [18], image processing is used in applications for two purposes:

1. Improving the visual appearance of images to a human observer, including their printing and transmission.
2. Preparing images for measurement of their features and structures which they reveal.

# Image processing has provoked a tremendous level of interest in the past years, making it a vast field of study. Thus, various techniques in image processing have been introduced, including visual enhancement, data extraction and the context of reproduction and transmission. These techniques include scaling and cropping.

# Scaling

# http://www.compuphase.com/graphic/lena-96sb.pnghttp://www.compuphase.com/graphic/lena-org.pngImage scaling is the process of resizing a digital image [17]. It increases or decreases the size of the image. This process always involves a trade-off between efficiency, smoothness, and sharpness. Image size is commonly decreased or sub-sampled to fit a smaller display area. Enlarging or interpolating an image, on the other hand, is common for making smaller images fit a bigger screen. Figure 1.7 shows an example of decreasing the size of an image.

# Taken from [17]

# Figure 1.7: Original image and the image decreased by 25%

# Cropping

# http://digital-photography-school.com/wp-content/uploads/old/IMG_2218.jpghttp://digital-photography-school.com/wp-content/uploads/old/IMG_2218_2.jpgCropping an image removes some parts of the image to improve framing or to highlight objects of interest [7]. Usually, this is useful for discarding areas with less useful information. This can also help address the problem of unbalanced images. Figure 1.8 shows an example of cropping an image.

# Taken from [7]

# Figure 1.8: Original image and cropped image

# Color Space

# A color space, also known as color model, is a mathematical or numerical representation of a set of colors. The purpose of a color model is to facilitate the specification of colors in some standard generally accepted way. In essence, a color model is a specification of a 3-D coordinate system and a subspace within that system where each color is represented by a single point [3]. Figure 1.9 shows the color subspace of the RGB and CMY color models. Past researches under tomato classification have utilized the RGB, HSI, and CIE L\*a\*b\* color models. All of the color spaces can be derived from the RGB information.

# F:\My Websites\TomatoX\Color Models_files\ch6_color_models_2.jpgTaken from [3]

# Figure 1.9: Color subspace of RGB and CMY color models

# RGB

# The red, green, and blue (RGB) color space is the most basic and well-known color model. In this model, each color appears as a combination of red, green and blue, which are called the primary colors. This model is called additive because the primary colors can be added to produce the secondary colors of light (see Figure 1.10). The combination of red, green and blue at full intensities makes white [3].

# F:\My Websites\TomatoX\The RGB (CMY) Color Model - Color Models - Technical Guides_files\rgb_model.gif The RGB color space is widely used in computer graphics because color displays use red, green, and blue to create the desired color. Also, this model can simplify the architecture and design of the system. However, the RGB color space fails in dealing with “real world” images.

# Taken from [20]

# Figure 1.10: The RGB color model

# HSI

# The HSI color model was developed to be more “intuitive” in manipulating colors and was designed to approximate the way humans perceive and interpret colors [3]. In this model, a color is represented by three components: Hue (*H*), Saturation (*S*), and Intensity (*I*). Hue defines the color itself ranging from 0 to 360 degrees. Saturation indicates the degree to which the Hue differs from a neutral gray running from 0 (no color saturation) to 1 (fullest saturation). Intensity indicates the illumination level. Equations (8) to (10) are used to convert RGB to HSI.

|  |  |
| --- | --- |
|  | **(8)** |
|  | **(9)** |
|  | **(10)** |

# CIE L\*a\*b\*

# In 1976, the *Comission Internationale de l'Eclairage* (International Commission on Illumination) introduced the CIE L\*a\*b\*, also known as CIELAB. The CIELAB color model is considered to be perceptually uniform and is referred to as a uniform color model [3]. This color model is a uniform derivation from the standard CIE XYZ space. The L\* component represents *Lightness*, ranging from 0 to 100. The a\* and b\* components are the chrominance, sometimes referred to as red / blue and yellow / blue chrominance respectively. Equations (11) to (15) are used to convert RGB to CIELAB. First the RGB is converted to CIE XYZ and then CIE XYZ to CIELAB.

|  |  |
| --- | --- |
|  | **(11)** |
|  | **(12)** |
|  | **(13)** |
|  | **(14)** |
|  | **(15)** |

# Neural Networks

# http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.artn.jpgA Neural Network (NN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information [14]. It is a collection of highly interconnected processing elements (neuron) working in unison to solve specific problems. With their remarkable ability to derive meaning from complicated or imprecise data, NNs can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques [14]. Figure 1.11 shows the neuron model of a NN.

# Taken from [14]

# Figure 1.11: The neuron model

# NNs are modeled as a collection of nodes connected via directed links. Each link has an associated weight that represents the strength of a connection between the connected nodes. The value of the output node is determined by an activation function. This function takes the summation of the input nodes and the corresponding weights. There are many activation functions in use. The most common is the logistic function, also known as Log-Sigmoid written mathematically as

|  |  |
| --- | --- |
|  | **(16)** |

# where .

# http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.multil.jpg In terms of architecture, there are two types of NN, feed-forward networks and feedback (recurrent) networks. Feed-forward networks allows signal to travel only in one way, i.e. from input to output. On the other hand, Feedback networks can have signals travelling in both directions. The simplest form of a feed-forward network is the single-layer neural network, more commonly known as a perceptron. The network only has an input layer and an output layer. If a hidden layer is added, the feed-forward network is then referred to as a multilayer feed-forward network, or a multilayer perceptron. Figure 1.12 shows an example of a multilayer perceptron.

# Taken from [14]

# Figure 1.12: Multilayer perceptron

# Like people, NN learns by example. In order for it to make generalizations, it must be trained using a series of sample inputs and outputs. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons [14]. Same is true with NN; knowledge is contained in the values of the connections weights. Therefore, modifying the knowledge stored implies changing the values of the weights. NNs are usually trained using Backpropagation (BP) algorithm and other metaheuristics such as Genetic Algorithm (GA) [14].

# NN is very popular because of its superior generalization capability. However, the major bottleneck in its application is the slow learning speed. The algorithms that are used to train NN consumes a lot of time and some require intensive human intervention.

# Extreme Learning Machines

# The Extreme Learning Machines (ELM) is a recent breakthrough in machine learning and data analysis. It tries to solve the pressing issues encountered by traditional learning techniques such as neural networks (NN) and support vector machines (SVM). These issues include intensive human intervention and slow learning speed. Developed by Huang et al. [4], ELM works for the “generalized” single-hidden layer feed-forward networks (SLFNs) but the hidden layer (feature mapping) in ELM need not be tuned. Figure 1.13 shows the structure of a SLFN.

# C:\Users\Gabrielle\Pictures\thesis\slfn2.png

# Taken from [9]

# Figure 1.13: Single-hidden layer feed-forward network

# In NN, all hidden nodes in SLFNs need to be tuned. ELM however shows that these hidden nodes need not be tuned and can be randomly generated. In fact, all parameters of ELM can be analytically determined. The ELM algorithm is summarized on the next paragraph.

Given a training set , hidden node output function , and hidden node number *L*,



*step* 1Randomly generate hidden node parameters (), *i* = 1 ,…, *L*.



*step* 2Calculate the hidden layer output matrix **H**.

*step* 3 Calculate the output weight vector **β**: **β =**



# Hidden node parameters and are randomly generated. The parameter corresponds to the input weights connecting the input layer and the hidden layer. The parameter corresponds to the bias for each hidden nodes. These parameters remain fixed after randomly generated.

# The calculation of the hidden layer output matrix H is done using the hidden node output function . denotes the activation function. The matrix H is written as:

|  |  |
| --- | --- |
|  | **(17)** |

# where represents the number of samples and is the number of hidden nodes.

# The output weight vector β and target output T is written as:

|  |  |
| --- | --- |
|  | **(18)** |
|  | **(19)** |

# where is the number of output nodes. is the Moore-Penrose generalized inverse of matrix , computed as:

|  |  |
| --- | --- |
|  | **(20)** |
|  | **(21)** |

# where is the transpose of matrix and is the inverse of the matrix.

# 1.6.1 Introduction of a Regularization Coefficient

# If a positive value 1/*λ* is added to the diagonal of or in the calculation of, the resultant solution is more stable and tends to have better generalization performance. The formula for the calculation of the output weight is as described in equations (22) and (23):

|  |  |
| --- | --- |
|  | **(22)** |
|  | **(23)** |

# In contrast with the Basic ELM, in this implementation, the condition on the number of hidden nodes does not closely depend on the number of training samples. It works for both cases, whether or [9].

**Chapter 2**

# Literature Review

There is a growing interest in the field of image classification. It has been applied to various fields and one of which is in fruit production. A lot of interests have been given to tomato classification. Researchers have used different methods of classification including colorimeters, NaiveBayes, Multilayer Perceptron, RBF Network, Neural Binary Tree, Random Tree, Random Forest, Instance Base K-Nearest Neighbor, K-Star (K\*), Neuro Fuzzy Inference, morphological features, model-based approach, LVQ Networks, and SVM.

Lopez et al. [12] used colorimeters obtained from the CIELAB color space to calculate color indexes. The relationship between these indexes and the visual color classification of tomatoes were compared. The researchers used the USDA color chart for the tomato ripening stages. They used the L\*, a\* and b\* data to calculate the (a) hue, (b) chroma, (c) color index, (d) color difference with pure red, (e) a\*/b\*, and (f) (a\*/b\*)2. They found out that the color changes in the ripening stages were the result of the changes in the values of L\*, a\* and b\*; the more important ones were along the a\* axis. Their experiments showed that the chroma and (a\*/b\*)2 are not good parameters to express tomato ripeness. On contrary, the hue, color index, color difference and a\*/b\* could be used as objective ripening indexes.

Asadollahi et al [1] used image processing techniques in classifying tomatoes. The dataset is composed of 90 images of tomato. Ten (10) characteristics were extracted from each image namely (a) greenness grade, (b) redness grade, (c) yellowness grade, (d) average of greenness, redness, and yellowness grade, (e) entropy, (f) energy, (g) contrast, (h) sympathetic, (i) circularity, and (j) area. The researchers classified tomatoes into three groups: high quality, medium quality, low quality. They compared classifier methods for data classification which include NaiveBayes, Multilayer Perceptron, RBF Network, Neural Binary Tree, Random Tree, Random Forest, Instance Base K-Nearest Neighbor and K-Star (K\*). Their experiments showed that some characteristics do not affect the results. These characteristics include (a) yellowness, greenness, redness average, (b) sympathetic, (c) circularity and (d) area. They concluded that the Multilayer Perceptron and random tree classifier method have better results compared to the other classifiers.

Iraji et al [11] used both fuzzy inference system (FIS) and adaptive neuro fuzzy inference system (ANFIS) for an accurate and appropriate decision on tomato classification. The researchers utilized 7 factors obtained from the images which are (a) maximum radius, (b) difference between green average and red average, (c) difference between red average and blue average, (d) difference between green average and blue average, (e) ratio of maximum radius over minimum radius, (f) image entropy and (g) length of the boundary of the largest defect on the tomato. The output were classified into 9 different classes namely (1) large circular red tomato, (2) medium circular red tomato, (3) small circular red tomato, (4) decayed tomato, (5) non-red tomato, (6) red tomato with bad texture, (7) large non-circular red tomato, (6) medium non-circular red tomato and (9) small non-circular red tomato. They concluded that the ANFIS has less error and is more accurate compared to FIS.

Wang et al [26] used vision-based judgment for tomato maturity under growth conditions. They used near-infrared images and RGB/HSI images. In their study, five maturity stages were used: breakers, turning, pink, light-red, and red. Tomato images were taken under natural illumination and growth conditions. Their research showed that the average intensity of near-infrared image cannot be used to directly judge the maturity of tomatoes. They concluded that with the change of maturity stage, the hue average tends to decrease progressively and the average of the red-green color difference tends to increase progressively. Also, both of these methods have the maximum standard deviations at the pink stage. Therefore, the hue and red-green color difference averages can be used as standards in tomato maturity judgment. Moreover, the red-green mean method is more satisfactory as it achieved an accuracy of over 96%.

Van de Poel et al. [25] used a model-based approach to classify tomatoes according to their physiological maturity. The researchers used the concept of biological age which is based on two parameters: color (expressed as hue) and mass. The researchers utilized two datasets for their study which are the calibration dataset and application dataset. The first is used for model calibration and the other is for classifying tomatoes. They concluded that the calibrated model is suitable for classifying individual tomato fruit for which only single point mass and hue observations are known.

Fojlaley et al. [5] analyzed the qualities of tomato using three different methods: Learning Vector Quantization (LVQ), multilayer perceptron (MLP), and support vector machines (SVM). Their dataset included 142 tomato images acquired using a digital camera; 86 of which are training data and 56 were testing data. Prior to the feature extraction step, the images underwent a series of image processing techniques which include denoising and contrast improvement. The outputs are classified to three: good, fair, bad. Eight (8) features were extracted: (1) degree of redness and yellowness, (2) degree of greenness, (3) first moment, (4) second moment, (5) third moment, (6) average of the three moments, (7) roundness value, and (8) surface area. The features are fed to the classifiers and results were compared. Their experiments showed that support vector machines (SVM) outperformed the other classifiers.

Followed by its conception, Extreme Learning Machines (ELM) has attracted the attention of more and more researchers. As an emergent technology that overcomes some of the challenges faced by other computational intelligence techniques, ELM has certainly proven its worth. Providing better generalization performance at a much faster learning speed and with the least human intervention, ELM has been applied to a vast number of fields. According to [4], ELM has been successfully used in the following applications: (1) Biometrics, (2) Bioinformatics, (3) Image processing, (4) Signal processing, (5) Human action recognition, (6) Disease prediction and eHealthCare, (7) Location positioning system, (8) Brain computer interface, (9) Human computer interface, (10) Feature selection, (11) Time-series, (12) Real-time learning and prediction and (13) Security and data privacy.

Minhas et al. [13] used ELM in object detection and categorization. In their study, they combined global and local object information. To attest the credibility of their study, they used a number of common publicly available image databases such as UIUC, Caltech, MIT, GRAZ and PASCAL. Their initial experiments demonstrated the extremely fast classification capability of ELM using global features. However, the accuracy achieved using global features were not stable. Thus, they combined global and local information to attain reliable classification. Prior to extracting information, images were converted to grayscale and resized to square dimension matrices. For the global information they used the bidirectional 2D-PCA (principal component analysis). For the local information, images were deformed for possible pose variations. Their experiments showed 95% accuracy for the MIT database. However, an inconsistent classification is observed for the rest of the datasets. To compare the results obtained from their study to different researches, the Caltech dataset was used. The proposed method achieved accuracy above 97% and outperformed other methods. They conclude that the combination of the global and local information offer high speed and is capable of handling changes in pose, illumination, inter-class and intra-class attributes.

Huang et al [8] used ELM for classification problems. The ELM was extended to support vector networks. The performance of ELM was tested against 11 UCI datasets and 2 Gene expression datasets. They compared the results to SVM and SVM with ELM kernel. For the UCI datasets, their experiments showed that ELM achieves the best generalization performance. The SVM and SVM with ELM kernel achieved similar performance. However, the SVM is faced with the issue of finding the best combination of parameters. For the DNA microarray datasets, ELM achieved the best generalization performance. They concluded that both SVM and ELM shares the same objective, i.e. to maximize the separating margin and minimize the training errors. However, the generalization performance of ELM is less sensitive to the parameters unlike the traditional SVM. Thus, the ELM can be implemented easily.

Huang et al. [10] also applied ELM to regression and multiclass classification problems. To test the performance of ELM, results were compared to the least square support vector machine (LS-SVM) and proximal support vector machine (PSVM). In order to verify the performance, a wide variety of datasets have been tested including 12 binary classification cases, 12 multi-classification cases, and 12 regression cases. Their experiments showed that ELM achieved comparable performance to LS-SVM and PSVM at a much faster learning speed. They also conducted an experiment on the XOR problem to verify if the ELM can handle rare cases such as extremely few training data sets. Their experiment showed that ELM can solve the XOR problem well. They concluded that ELM achieved similar or better generalization performance for regression and binary class classification cases, and much better generalization performance for multiclass classification cases.

**Chapter 3**

# Statement of the Problem

Major supplies of tomatoes come from farms and greenhouses. They deal with large number of tomatoes and each one will be sorted and classified accordingly. Traditionally, the sorting and classification of tomatoes are done by human graders. This is a very time-consuming, tedious, and expensive task. In fact, the high labor cost has been the main obstacle in the expansion of greenhouses. Additionally, the human identification of colors is affected by sensations like brightness, intensity, lightness and vividness. Thus, in many cases color identification is a matter of subjective interpretation.

A lot of research has been conducted on the automation of the sorting and classification of tomatoes. This is because automation offers a lot of benefits such as reduced identification time, more consistent classification, and improved quality of tomato products. A number of methods have been used to accomplish this task and the most successful includes Neural Networks (NN) and Support Vector Machines (SVM). This is due to the fact the both NN and SVM have superior generalization capabilities. However, both methods face some challenging issues such as intensive human intervention and slow learning speed. In fact, the slow learning speed of NN has been the major bottleneck in its applications for the past decades.

Extreme Learning Machines (ELM) as an emergent technology has gained a lot of interest from researchers. ELM overcomes some of the challenges faced by both NN and SVM. In fact, researches on ELM have shown that it provides better generalization performance at a much faster learning speed and with least human intervention.

This study will try to use ELM in judging the maturity level of tomatoes based on their visual characteristics.

**Chapter 4**

# Objectives

The objectives of this study are as follows:

1. To assess the performance of ELM in the judgment of tomato maturity.
2. To determine the effect of varying the parameters of ELM such as the number of hidden nodes.
3. To determine the effect of introducing a regularization coefficient to the performance of ELM in the judgment of tomato maturity.
4. To determine the effect of using the components L\*, a\*, Hue, red-green difference from the three color models (RGB, HSI, CIE L\*a\*b\*) as inputs to the ELM network.

**Chapter 5**

# Proposed Approach

**5.1 Process Flow**

The flow of the processes for this study is illustrated in Figure 5.1. The tomato image undergoes a number of preprocessing steps before it is fed into the ELM network.

**Figure 5.1:** The pipeline of TomatoX

**5.2 Tomato Image**

The input is a tomato image of any dimension. The tomato images used in this study are acquired using a digital camera. Four features will be extracted from the RGB, HSI, and CIELAB color space. These features are L\*, a\*, Hue, and red-green difference. These features represent the input vector for the ELM network. These features were utilized in the past researches in tomato classification. Figure 5.2 shows a 200 x 200 tomato image.

****

**Figure 5.2:** Tomato image

**5.3 Scaling**

Since the images taken from a digital camera are quite large, they must be reduced appropriately. Tomato images will be reduced to a 200 x 200 dimension. In this way, some of the processing steps will be much faster.

**5.4 Blue Channel**

The next step is to extract the values from the blue channel. The blue channel is the one being manipulated in the other steps. We can easily accomplish this step since the blue channel is just a subset of the RGB color space. We just set the values of the red and green channel to zero. In turn, only the blue channel will be retained. Figure 5.3 shows the blue channel of Figure 5.2.

**Figure 5.3:** Blue channel

**5.5 Grayscale Conversion**

After extracting the blue channel, the image will be converted to grayscale. The luminance method shown in equation **(1)** will be used to convert the pixels of the blue channel image to grayscale. A grayscale image contains the same values for the red, green, and blue channel. Figure 5.4 shows the grayscale image obtained from Figure 5.3.

**Figure 5.4:** Grayscale image

**5.6 Masking**

The next step is to make the binary mask using the Otsu thresholding technique. First, the histogram of the grayscale image is computed. It is usually represented as a one dimensional array. The indexes of the array indicate the pixel values of the image. Since the image is grayscale, the array is only of size 256. Figure 5.5 shows the histogram of Figure 5.4.

**Figure 5.5:** Image Histogram

The histogram is calculated by iterating through all the pixels of the image and incrementing the array index corresponding to the pixel value. Figure 5.6 shows the code snippet of calculating the histogram.

**Figure 5.6:** Calculating the histogram

After the computation of the histogram, the optimal threshold can be calculated using Otsu’s method. This method will iterate from 0 to 255 finding the threshold that maximizes the between-class variance using equation **(3)**. For Figure 5.4, it has been found out that at index 12, the between-class variance is at maximum at 1.04418419E11. Therefore, the optimal threshold is 12. This threshold will determine which pixels belong to the foreground and which pixels belong to the background. Pixels to the left of the threshold are considered as foreground (1 or 255) while the ones to the right belong to the background (0). Figure 5.7 shows the binary mask obtained by thresholding the grayscale image in Figure 5.4 using the optimal threshold.

**Figure 5.7:** Binary Mask

After creating the binary mask, the next step is the masking operation. This is done by applying the binary mask (Figure 5.7) to the original tomato image (Figure 5.8 (a)), i.e. multiplying the pixels of the images. If the pixel value of the binary mask is 1, the product is the same as the original pixel. On the other hand, if the pixel value of the binary mask is 0, the product will be 0 which is the same as the binary mask pixel. Figure 5.8 (b) shows the output of masking.

(a)

(b)

**Figure 5.8:** Masking Operation (a) Original Image (b) Output of Masking

**5.7 Cropping**

The image needs to be cropped since the background is of no use as can be seen in Figure 5.8 (b). The background pixels will just contribute to the processing time and may affect the features to be extracted. To remove the background, the bounds of the tomato (foreground) must be calculated. This can be done easily since the background was already replaced by a black color. Figure 5.9 shows the cropped image of Figure 5.8 (b).

****

**Figure 5.9:** Cropped image

**5.8 Feature Extraction**

Before the cropped image undergoes feature extraction, it must be resized to a 64 x 64 dimension as shown in Figure 5.10. The features that will be extracted from the image are the average L\*, average a\*, average Hue (H), and average red-green difference (R-G). Extracting the RGB features are the easiest since the input image is already in the RGB format. However, for the HSI and CIE L\*a\*b\* features, the RGB image must first be converted to the respective color spaces. The conversion to the HSI color model is done by using equations **(8), (9)** and **(10)** while the conversion to the CIE L\*a\*b\* color model is done using equations **(11), (12), (13), (14), (15).** Table 5.1 shows the values of the features extracted from Figure 5.10.



**Figure 5.10:** 64 x 64 Image

**Table 5.1:** Color Features extracted from Figure 5.8

|  |  |  |  |
| --- | --- | --- | --- |
| L\* | a\* | H | R-G |
| 40.52 | **46.48** | **6.16** | **118.97** |

**5.9 ELM Training**

The training set for this study consists of 70% of the dataset. The network structure for TomatoX is shown in Figure 5.11.

**Figure 5.11:** ELM Network Structure of TomatoX

**5.9.1 Input Layer**

The number of input nodes corresponds to the number of color features. In this study, the four (4) features extracted from the tomato image constitute the number of input nodes.

**5.9.2 Hidden Layer**

The number of hidden nodes is specified by the user. According to [9], the number of hidden nodes in the basic ELM algorithm should not exceed the number of samples, i.e. where corresponds to the number of hidden nodes and denotes the number of samples. In this study, the number of hidden nodes should not exceed 70% of the dataset.

**5.9.3 Output Layer**

The number of output nodes corresponds to the maturity stages of the tomato. In this study, six stages of tomato maturity are distinguished, namely: green, breakers, turning, pink, light red, red. Therefore, the number of output nodes is equal to six (6).

**5.9.4 ELM Training**

The steps for training the network using ELM is explained in Chapter 1.6. The first step is to randomly generate hidden node parameters, (input weights connecting the input layer and hidden layer) and (bias of each hidden node), with values ranging from -1 to 1. Below illustrates an example of the hidden node parameters. We assumed in the following example that the number of hidden nodes **.**

where is the number of input nodes and is the number of hidden nodes.

After calculating the hidden node parameters, we can now proceed to the next step which is the calculation of the hidden layer output matrix using the expression shown in equation **(17).** The activation function used in this study is the Log-Sigmoid function calculated using equation **(16).** Then we can proceed to the calculation of the output weight vector which is derived using and the target output vector (*see section 1.5*). Below illustrates an example of derived from example hidden node parameters above. We assumed in the following example that the number of training samples which corresponds to Figure 5.2. was computed using a special API in the Java Programming Language.

where is the number of samples and is the number of hidden nodes.

Unlike the traditional learning algorithms, ELM happens in just once cycle. The hidden node parameters and and the output weight vector are then concatenated and stored in a file. This file is known as a classifier which is used in the classification of a tomato.

**5.10 Tomato Classification**

The hidden node parameters and the output weight vector are first retrieved from the classifier specified by the user. The features extracted from the tomato will serve as the input nodes of the neural network. In this example, the hidden node parameters generated earlier will be used and the features extracted from Figure 5.1, i.e. the values from Table 5.1 will be used as the values to the input nodes. The hidden layer output matrix is computed in the same way as in the previous section. The values of the output nodes are calculated by multiplying and the output weight vector . The output values are normalized to determine which tomato stage corresponds to the inputted values. Normalization is done by getting the index with the maximum value, which in this example is 5 or the Red stage. Below illustrates an example of derived from the sample values stated above. Figure 5.12 shows the ELM Network with some of the calculated sample values.

**Figure 5.12:** ELM Network with sample values

**Chapter 6**

# Experiments and Results

# 6.1 System Specifications

The experiments conducted on this study were performed on a machine with the following specifications:

**Hardware:**

AMD E-350 Processor 1.60 GHz

2.00 GB of RAM

**Software:**

Microsoft Windows 8 Pro 64-bit (6.2, Build 9200)

Eclipse Juno IDE

Java Virtual Machine

**6.2 Experimental Settings**

The program was coded using the Java programming language through Eclipse Juno IDE. The dataset is composed of six hundred (600) images, one hundred (100) images per tomato stage. The images were acquired using a digital camera and their maturity stage was manually classified. The effectiveness of the ELM classifier will be tested against this manual classification. The dataset was divided into two subsets, the train set and the test set. The train set is composed of seventy percent (70%) of the dataset whilst the test set is composed of the remaining thirty percent (30%). The division is done by selecting 70% of the images per tomato stage to be used as train set and the remaining 30% for the test set. Since every stage is composed of 100 images, 70 of this are randomly selected to be used in training and the remaining 30 for testing.

**6.3 Experimental Results and Discussion**

To assess the performance of ELM in the evaluation of tomato maturity, a standard ELM configuration was defined. Table 6.1 shows the standard parameter values of the ELM parameters.

**Table 6.1:** ELM Standard parameter values

|  |  |
| --- | --- |
| Parameter | Standard Value |
| Number of Hidden Nodes | 105 |
| Regularization Coefficient (C) | Not set |

Four (4) experiments were conducted in this study. The experiments are as follows:

1. Experiment I – comparison of fixed and random dataset
2. Experiment II – variation in the number of hidden nodes
3. Experiment III – variation in the values of the regularization coefficient
4. Experiment IV – variation in the number of input color features

For statistical acceptability, all experimental setups were subjected to 30 trials. In all the experiments, for the ELM network training time reached a minimum of 0.2 seconds and a maximum of 4 seconds. This is due to the fact that the ELM algorithm happens in just one sweep, i.e. no iterations are needed in training.

**Experiment I**

For experiment I, two datasets were compared, the fixed data and the random data. For the fixed data, a fixed train and test set were used for all 30 trials. On the other hand, for the random data, a random train and test set were used for every trial. The standard ELM configuration was used in this experiment. Both the train and test set were used in the comparison. Table 6.2 shows the results for this experiment. Figure 6.1 shows the graph for the results of the 30 trials.

**Table 6.2:** Results for fixed and random data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Fixed | | Random | |
|  | **Test set** | **Train set** | **Test set** | **Train set** |
| Mean | 97.593% | 98.865% | 97.185% | 98.905% |
| Median | 97.778% | 99.048% | 97.222% | 98.929% |
| Minimum | 95% | 98.333% | 93.333% | 97.619% |
| Maximum | 99.444% | 99.524% | 100% | 99.524% |
| STDEV | 1.266 | 0.399 | 1.458 | 0.445 |
| Variance | 1.604 | 0.159 | 2.127 | 0.198 |

**Figure 6.1:** Results for the fixed and random data

For the test set, the fixed data has a higher average than that of the random data but the random data got 100% accuracy on one of the trials. The accuracy for the random data could go as low as 93.333336%. This is due to the fact that since a random training and test set is used for every trial, there is a possibility that the images that are more difficult to classify were on the test set while the easier ones were on the training set. In other words, the classifier was not trained well to handle more complicated inputs. There is also a possibility that the opposite situation will happen, i.e. the images that are easier to classify were on the test set while the difficult ones were on the training set. This can be seen on the results for the train set, the random data has a higher average compared to the fixed data.

Based on the results, it shows that the ELM classifier is robust since it was able to classify with high accuracy regardless how the datasets were generated. For the random data, the classifier could correctly classify all data. The classifier could even handle complex data with little impact on the accuracy as can be seen in the results for the random data.

**Experiment II**

For experiment II, the number of hidden nodes was varied according to the number of samples. According to the interpolation theorem [9], the maximum number of hidden nodes required is not larger than the training samples. The maximum number of hidden nodes used for this experiment is equal to the number of training samples. Since the dataset is composed of 600 images, 420 images were used as training samples. The variation in the number of hidden nodes used was N, 0.75N, 0.5N, 0.25N where N is the number of samples. Table 6.3 shows the variation in the number of hidden nodes.

**Table 6.3:** Varying the number of hidden nodes

|  |  |
| --- | --- |
| Set | Number of Hidden Nodes |
| A | 105 |
| B | 210 |
| C | 315 |
| D | 420 |

The fixed data used in Experiment I is also used in this experiment. Both the train and test set were used in testing the efficiency of the classifier. Table 6.4 and 6.5 shows the results of the test and train set respectively.

**Table 6.4:** Results of varying the number of hidden nodes for the test set

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | A | B | C | D |
| Mean | 97.593% | 96.796% | 94.667% | 87.463% |
| Median | 97.778% | 96.944% | 94.444% | 88.056% |
| Minimum | 95% | 94.444% | 92.222% | 72.222% |
| Maximum | 99.444% | 98.889% | 97.222% | 93.889% |
| STDEV | 1.266 | 1.069 | 1.12 | 4.494 |
| Variance | 1.604 | 1.143 | 1.439 | 20.193 |

**Table 6.5:** Results of varying the number of hidden nodes for the train set

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | A | B | C | D |
| Mean | 98.865% | 99.556% | 99.714% | 99.952% |
| Median | 99.048% | 99.524% | 99.762% | 100% |
| Minimum | 98.333% | 99.048% | 99.286% | 99.762% |
| Maximum | 99.524% | 99.762% | 100% | 100% |
| STDEV | 0.399 | 0.205 | 0.170 | 0.097 |
| Variance | 0.159 | 0.042 | 0.029 | 0.009 |

**Figure 6.2:** Results trials for varying the number of hidden nodes for the test set

**Figure 6.3:** Results for varying the number of hidden nodes for the train set

As shown in Figure 6.2 and Table 6.4, for the test set, as the number of hidden nodes is increased, the accuracy of the classifier tends to decrease. This is because as the number of hidden nodes approaches the number of training samples, the ELM becomes a singular least-squares problem and gives an unstable solution. The proof of having unstable solution can be depicted in the results when the number of hidden nodes is 420. The standard deviation is at highest and the accuracy could go as low as 72.22222%. The highest accuracy for this set was attained when the number of hidden nodes is 105, correctly classifying 179 out of 180 images achieving a success rate of 99.44444%.

As shown in Figure 6.3 and Table 6.5, for the train set, as the number of hidden nodes is increased, the accuracy of the classifier increases accordingly. This can be contrasted to the results obtained in the test set. The reason behind this is that as the number of hidden nodes approaches the number of training samples, the matrix **H** becomes square and invertible. Thus, according to [9], SLFNs can approximate these training samples with zero error. In fact, for this set, when the number of hidden nodes is 420, the maximum accuracy of the classifier is 100%.

**Experiment III**

The regularization coefficient (C) can be any positive number. For this experiment, the value of the regularization coefficient (C) is varied from 1 to 50. The standard number of hidden nodes was used in this experiment. The fixed dataset used in experiment I was also used in this experiment but only the test set was used in testing the classifier, i.e. for every regularization coefficient (C), the train set was used for training but only the test set was used in testing the accuracy of the classifier. This experiment was carried out to see if the introduction of the regularization coefficient will improve the performance of ELM in tomato classification. Table 6.6 shows the results of varying the values of the regularization coefficient (C). To avoid cluttering of data, only the values of the regularization coefficient with the higher average are displayed in Figure 6.4.

**Table 6.6:** Results of varying the values of the regularization coefficient

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | No C | 1 | 10 | 20 | 30 | 40 | 50 |
| Mean | 97.593% | 98.259% | 98.13% | 98.074% | 98.056% | 98.148% | 98.278% |
| Median | 97.778% | 98.333% | 98.333% | 98.333% | 98.333% | 98.333% | 98.333% |
| Min | 95% | 93.889% | 95.556% | 94.444% | 93.889% | 95.556% | 96.667% |
| Max | 99.444% | 100% | 99.444% | 100% | 99.444% | 100% | 99.444% |
| STDEV | 1.266 | 1.21 | 0.905 | 1.165 | 1.08 | 0.993 | 0.915 |
| Variance | 1.604 | 1.463 | 0.819 | 1.357 | 1.165 | 0.986 | 0.838 |

**Figure 6.4:** Results for varying the values of the regularization coefficient (C)

As shown in Table 6.6, the results for all values of C have greater accuracy compared to the result obtained without the introduction of C. This suggests that the introduction of the regularization coefficient (C) further improves the classification capability of ELM. However, as shown in Table 6.6 and Figure 6.4, an increase in the value of C does not necessarily mean an increase in the accuracy. In fact, 100% accuracy was achieved when the value of C is 1, 20, and 40. The results also show that the introduction of a regularization coefficient makes the classifier more stable as can be seen on the values of the STDEV.

**Experiment IV**

For the last experiment, the number of input features was varied. Table 6.7 shows the variation in the combination of the input features. The fixed data used in Experiment I is also used in this experiment. The standard ELM configuration was used in this experiment.

**Table 6.7:** Varying the input features

|  |  |
| --- | --- |
| Set | Color Feature (s) |
| A | R-G |
| B | H, R-G |
| C | L\*, a\* |
| D | L\*, a\*, H, R-G |
| E | R, G, R-G, H, a |

As shown in Table 6.7, each set has a different network structure since the input features correspond to the number of input nodes. The result of varying the input features is shown in Table 6.8.

**Table 6.8:** Results of varying the input features

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E |
| Mean | 96.037% | 94.444% | 96.741% | 97.593% | 95.833% |
| Median | 96.111% | 94.722% | 96.667% | 97.778% | 96.111% |
| Minimum | 94.444% | 92.222% | 95% | 95% | 91.111% |
| Maximum | 96.667% | 96.667% | 98.333% | 99.444% | 98.889% |
| STDEV | 0.68 | 1.384 | 0.711 | 1.266 | 2.02 |
| Variance | 0.463 | 1.916 | 0.505 | 1.604 | 4.082 |

**Figure 6.5:** Results for varying the input features

As shown in Figure 6.5 and Table 6.8, the results show that set D, containing the proposed input features performed best amongst all the setups. In fact, it can correctly classify 179 out of 180 images achieving an accuracy of 99.44444%. The results also show that the use of more input features does not necessarily mean an increase in the accuracy. In fact, set A containing only one input feature performed better than set B that contains two input features and set E that contains five input features.

For comparison, the result of this study is compared to the work of Opeña [15]. The results are not compared to other researches since each research used a different dataset.

**Table 6.9:** Comparison to TotoBee [15]

|  |  |  |
| --- | --- | --- |
|  | Current Research | TotoBee [15] |
| Accuracy | 97.593% | 98.19 % |
| Training Time (sec) | 1.89 | 153.46 |

As shown in Table 6.9, the accuracy of TotoBee [15] is larger than that of this research but only with a small difference. On the other hand, the training time of this research is smaller than that of TotoBee [15]. Time is a necessity in industrial applications. The results of this research offer faster training time with a small decrease in accuracy.

**Chapter 7**

# Conclusion and Future Works

# Extreme Learning Machines (ELM) as an emergent technology has attracted a lot of attention from researchers. It has been applied to various problems and it has shown great results. In this study, ELM was implemented to solve the tomato classification problem. Experimental results show that ELM can be an efficient method in the classification of tomatoes.

# In the first experiment, it has been shown that the ELM classifier is robust. It can handle complex data with little impact on the accuracy. In fact, for the random data, it correctly classified all tomato images.

# In the second experiment, it has been shown that the number of hidden nodes can affect the performance of ELM. As the number of hidden nodes is increased, the accuracy for the train set increases while the accuracy for the test set decreases.

# In the third experiment, the introduction of a regularization coefficient (C) further improves the generalization capability of the ELM. However, based on the results, an increase in the value of C does not necessarily mean an increase in the accuracy. It has also been shown that the introduction of a regularization coefficient makes the classifier more stable.

# In the last experiment, it has been shown that the input features have a significant effect on the performance of the ELM. However, an increase in the number of input features does not guarantee a good result. Set A containing only one input feature has better performance than set B and set C containing two and five input features respectively. Overall, set D containing the proposed input features performed best amongst all the setups. It correctly classified 179 out of 180 images achieving an accuracy of 99.44444%.

# Future works on this study could include the extension of this study to real-time tomato classification, i.e. the use of machines to pick tomatoes and directly classify their maturity stage. One could also make use of the Online-Sequential ELM (OS-ELM) which eliminates the need to retrain the whole dataset if the dataset is updated. Interested researchers could even apply the ELM to other vegetables or fruits especially those grown widely in the Philippines such as bananas.

**References**

[1] Asadollahi, H., Kamarposhty, M., Teymoori, M.: Classification and Evaluation of Tomato Images Using Several Classifier. In Proceedings of the 2009 International Association of Computer Science and Information Technology – Spring Conference (IACSIT-SC '09). (2009) 471 – 474

[2] Binary Images. From, http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL\_COPIES/OWENS/LECT2/node3.html (Accessed: March 2013)

[3] Color Models. From, http://software.intel.com/sites/products/documentation/hpc/ipp/ippi/ippi\_ch6/ch6\_color\_models.html (Accessed: March 2013)

[4] Extreme Learning Machines. From, http://www.ntu.edu.sg/home/egbhuang/ (Accessed: March 2013)

[5] Fojlaley M., Moghadam, P., Nia, S.: Tomato Classification and Sorting with machine vision using SVM, MLP, and LVQ. International Journal of Agriculture and Crop Sciences. (2012) 1083 – 1088

[6] Guide to Ripening Stages. From, http://www.lagorio.com/assets/pdf/lagorio-tomato-guide.pdf (Accessed: March 2013)

[7] How to Crop Images, From, http://digital-photography-school.com/cropping-for-impact (Accessed: March 2013)

[8] Huang, G.B., Ding, X., Zhou, H.: Optimization method based extreme learning machine for classification. Neurocomputing. (2010) 155 – 163

[9] Huang, G.B., Wang, D.H., and Lan, Y.: Extreme Learning Machines: A Survey. International Journal Machine Learning & Cybernetics. (2011) 2:107 – 122

[10] Huang, G.B., Zhou, H., Ding, X., Zhang, R.: Extreme Learning Machine for Regression and Multiclass Classification. IEEE Transactions on Systems, Man, and Cybernetics, Part B 42 (2). (2012) 513 – 529

[11] Iraji, M., Tosinia, A.: Classification Tomatoes on Machine Vision with Fuzzy the Mamdani Inference, Adaptive Neuro Fuzzy Inference System Based (Anfis-Sugeno), vol. 5 (11). (2011) 846 – 853

[12] Lopez Camelo, A., Gomez, P.: Comparison of color indexes for tomato ripening. Horticultura Brasileira, vol. 22, n.3. (2004) 534 – 537

[13] Minhas, R., Mohammed, A., Wu, J.: A fast recognition framework based on extreme learning machine using hybrid object information. Neurocomputing. (2010) 1831 – 1839

[14] Neural Networks. From, http://www.doc.ic.ac.uk/~nd/surprise\_96/journal/vol4/cs11/report.html (Accessed: March 2013)

[15] Opeña, H. J.: ToToBee: Tomato Maturity using Artificial Bee Colony Algorithm and Artificial Networks (Unpublished) (2013).

[16] Otsu Thresholding. From, http://www.labbookpages.co.uk/software/imgProc/otsuThreshold.html (Accessed: March 2013)

[17] Quick image scaling algorithms. From, http://www.compuphase.com/graphic/scale.htm (Accessed: March 2013)

[18] Russ, J.: The Image Processing Handbook, Sixth Edition*.* CRC Press (2011)

[19] Sargent, S., Moretti, C.: Tomato. From, http://www.ba.ars.usda.gov/hb66/138tomato.pdf (Accessed: March 2013)

[20] The RGB (CMY) Color Model. From, http://dba.med.sc.edu/price/irf/Adobe\_tg/models/rgbcmy.html (Accessed: March 2013)

[21] The SmartFresh Quality System for Tomatoes. From, http://www.agrofresh.com/smartfresh/tomatoes.html (Accessed: March 2013)

[22] Three algorithms for converting color to grayscale. From, http://www.johndcook.com/blog/2009/08/24/algorithms-convert-color-grayscale/ (Accessed: March 2013)

[23] Tomato History – The history of tomatoes as food. From, http://homecooking.about.com/od/foodhistory/a/tomatohistory.htm (Accessed: March 2013)

[24] Understanding digital images. From, http://www.reading.ac.uk/internal/using-images/UnderstandingDigitalImages/img-understandhome.aspx (Accessed: March 2013)

[25] Van de Poel, B., Bulens, I., Hertog, M., Van Gastel, L., De Proft, M., Nicolai, B., Geeraerd, A.: Model-based classification of tomato fruit development and ripening related to physiological maturity. Postharvest Biology and Technology, vol. 67. (2012) 59 – 67

[26] Wang, X., Mao, H., Han, X., Yin, J.: Vision-based judgment of tomato maturity under growth conditions. African Journal of Biotechnology, vol. 10 (18). (2011) 3613 – 3623

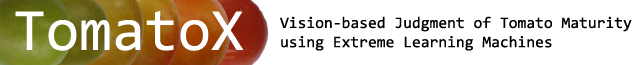
**APPENDIX**

**USER’S MANUAL**

The program entitled TomatoX is composed of seven (7) sections namely:

* ELM Training page
* Individual Classification page
* Batch Classification page
* Feature Extraction page
* Generate Data page
* About page
* Help page

A splash screen is displayed upon running the program as shown below. The Home page is set to the ELM Training page by default.

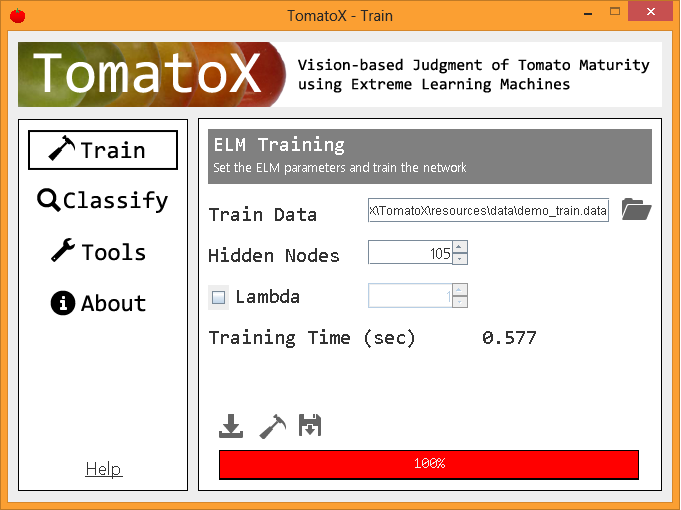
****

**Buttons**

|  |  |  |
| --- | --- | --- |
| Icon | Name | Description |
| D:\My Academics\Undergraduate Research\TomatoX\TomatoX\images\browse.png | Browse | Browse for a directory, test/train data, TomatoX classifier |
| D:\My Academics\Undergraduate Research\TomatoX\TomatoX\images\load.png | Load | Loads the test/train data |
| D:\My Academics\Undergraduate Research\TomatoX\TomatoX\images\train.png | Train | Trains the ELM network using the user-specified ELM parameters |
| D:\My Academics\Undergraduate Research\TomatoX\TomatoX\images\save.png | Save | Saves the trained ELM as a TomatoX Classifier |
| D:\My Academics\Undergraduate Research\TomatoX\TomatoX\images\classify.png | Classify | Classify an individual or a set of tomato |
| D:\My Academics\Undergraduate Research\TomatoX\TomatoX\images\view.png | View Table | Views the result of the batch classification as a table |
| D:\My Academics\Undergraduate Research\TomatoX\TomatoX\images\extract.png | Extract | Extract the color features from a tomato image |
| D:\My Academics\Undergraduate Research\TomatoX\TomatoX\images\show.png | View Process | Views the process of extracting the color features |
| D:\My Academics\Undergraduate Research\TomatoX\TomatoX\images\random.png | Generate | Generates a train and test data using the specified dataset |

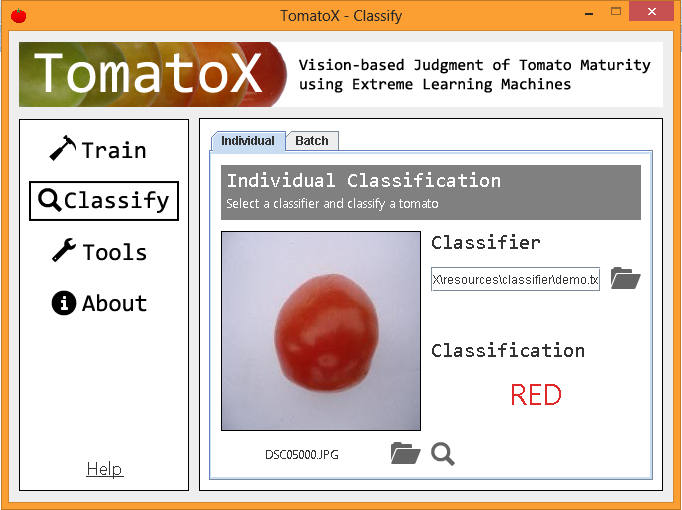
**ELM Train page**

This page allows the user to train the ELM network. The user specifies the train data and the ELM parameters such as the number of Hidden Nodes and the Lamda. It also displays the time it takes to train the ELM Network and it allows the user to save the trained ELM.

****

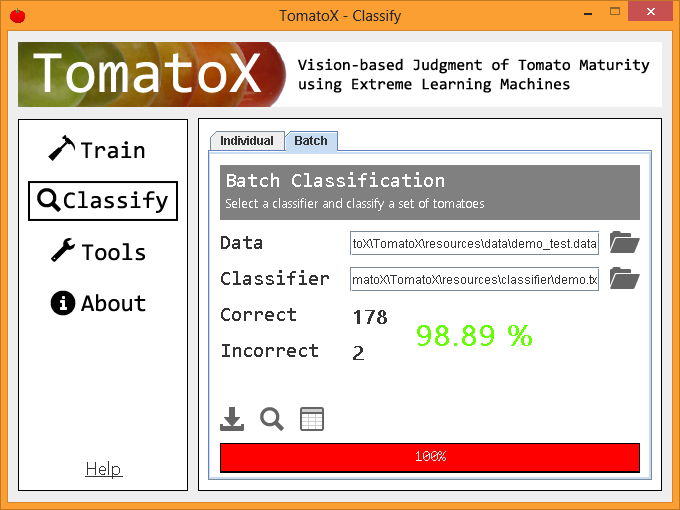
**Individual Classification page**

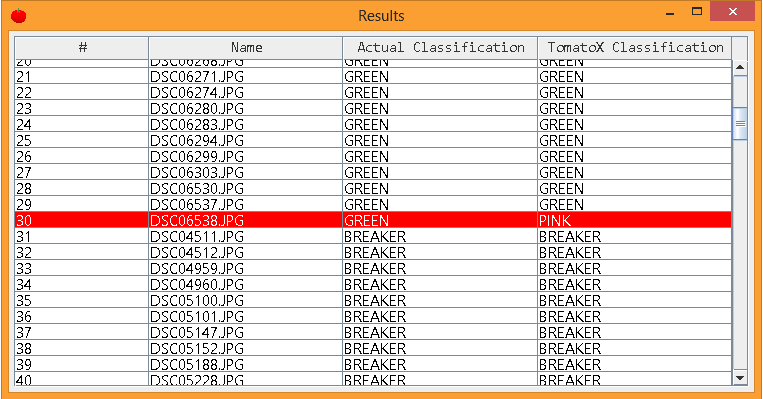
This page allows the user to classify an individual tomato. The user inputs the TomatoX Classifier and the tomato image. It displays the corresponding maturity stage of the input tomato image.

****

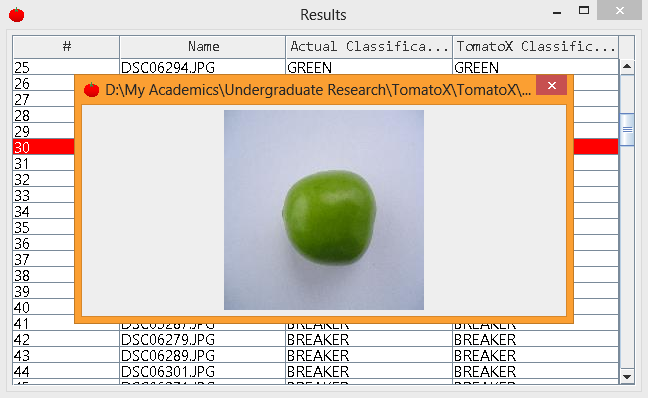
**Batch Classification page**

This page allows the user to classify a set of tomato. The user specifies the data containing the location of the tomatoes and a TomatoX Classifier. It displays the number of correct and incorrect classification. Also, it displays the accuracy of the TomatoX Classifier in percentage.



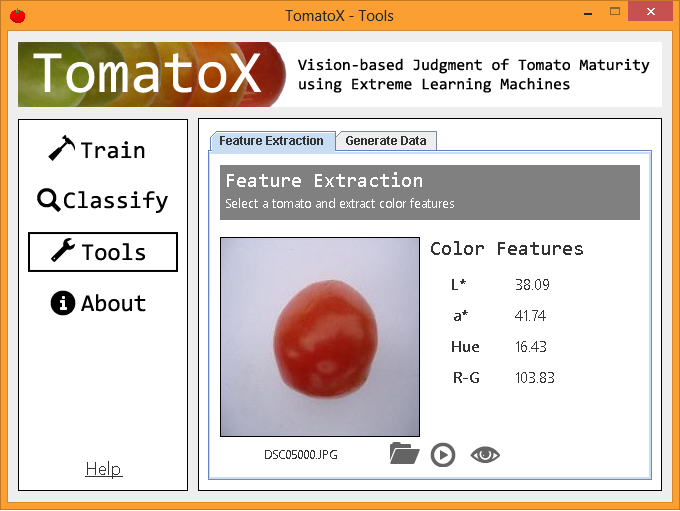
By clicking the View Table button, a frame containing a table showing the result of the classification is displayed as shown below. Incorrect result is highlighted in red.

By clicking on the name of the tomato image, a dialog containing the tomato image is displayed as shown below.

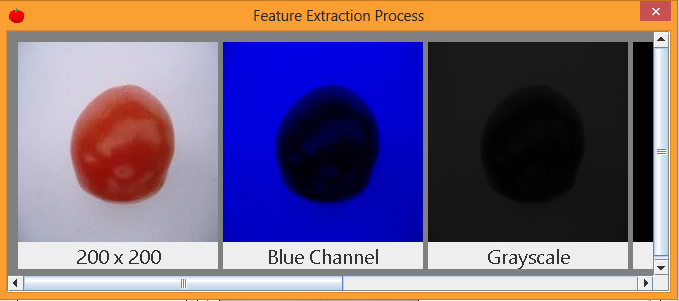


**Feature Extraction page**

This page allows the user to extract the color features from the input tomato image. The extracted features are displayed. It also allows the user to view the process of extracting the features.

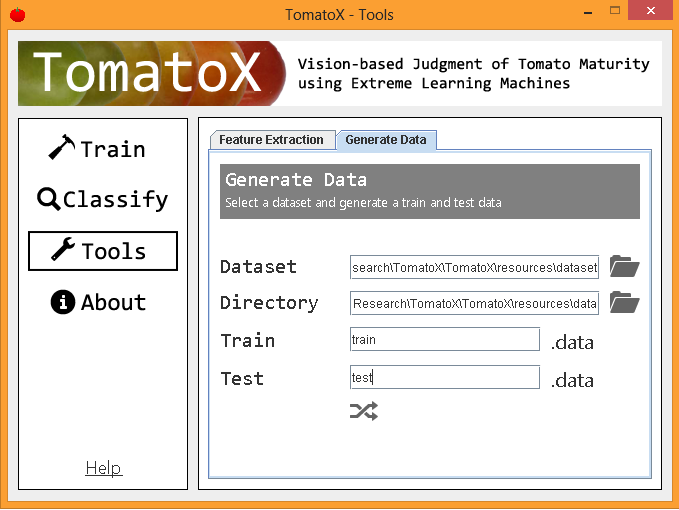
****

By clicking the View Process button, a dialog showing the process is displayed as shown below.



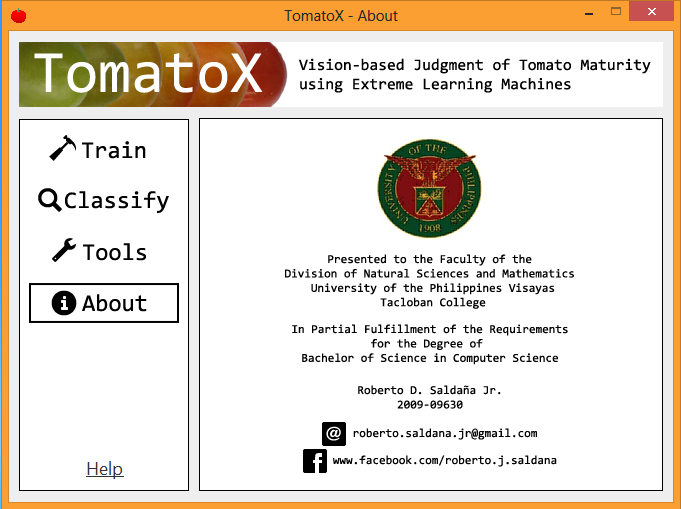
**Generate Data page**

This page allows the user to generate a train and test data from a given dataset. The user specifies the dataset and the directory in which the data will be saved.



**About page**

This page displays a brief overview of the program and the some information about author.



**Help page**

This page links to this user’s manual.