IMPLEMENTATION OF HUE SATURATION INTENSITY (HSI) COLOR SPACE TRANSFORMATION ALGORITHM WITH RED, GREEN, BLUE (RGB) COLOR BRIGHTNESS IN ASSESSING TOMATO FRUIT MATURITY

Budi Yanto[1], Maria Angela Kartawidjaja1[2], Ronald Sukwadi1[3] Marsellinus Bachtiar[4]
[1]Department of Informatics Engineering, Pasir Pengaraian University, Indonesia
[2,3,4]Department of Electrical Engineering, Atma Jaya Katholic University Indonesia

Email: *[1]budiyantost@gmail.com

Abstract: Tomatoes, as a type of vegetable or fruit, are often susceptible to damage, making handling them a complex challenge. Distinguishing between fresh and damaged tomatoes is very important, considering the significant impact on nutritional value and economic aspects. Traditional approaches via visual inspection have proven to be less efficient and inconsistent in their detection accuracy. To overcome this challenge, the use of images is a vital solution for distinguishing ripe, half-ripe and unripe tomatoes. In this context, HSI (Hue, Saturation, Intensity) calculations are applied to measure RGB color and room transformations. Images are extracted in jpg format, saved as training data, and this method is implemented using the Python programming language and GUI interface design in MATLAB. The research results show the HSI value for each class, with the ripe tomato class having an average hue of 0.0051 – 0.026, saturation 0.1862 – 0.3291, and intensity 0.0975 – 0.7522. Half-ripe tomatoes have hue 0.0208 – 0.0848, saturation 0.1346 – 0.5746, and intensity 0.1056 – 0.4714, while immature tomatoes have hue 0.0174 – 0.0689, saturation 0.0474 – 0.2072, and intensity 0.0595 – 0.3203. The integration of the HSI algorithm steps with the RGB color space provides an additional dimension to color analysis, which has the potential to increase the accuracy of tomato ripeness detection.

1. INTRODUCTION

Tomato (Solanum tomato) has a significant role as an agricultural commodity with high economic value and is the main consumption throughout the world[1]. The level of ripeness of tomato fruit is a crucial determining factor in determining the quality of the harvest and consumer satisfaction[2]. Unfortunately, conventional methods for assessing tomato fruit ripeness tend to be subjective and prone to uncertainty [3]. Therefore, a more objective and accurate approach is needed to provide more detailed information regarding the level of fruit maturity [4][5] .

a. Immature b. Half Ripe c. Ripe

Figure 1. Image of a tomato

The application of Digital Image Processing Technology is a promising solution to overcome the limitations of conventional methods in determining fruit ripeness [5][6]. The Hue Saturation Intensity (HSI) algorithm with the Red, Green, Blue (RGB) color space is the main focus of this research[7] [8]. This algorithm allows image transformation from RGB to HSI color space, providing richer information about color nuances, saturation and intensity in the image [9][10]. The advantage of this algorithm lies in its ability to provide in-depth analysis of color changes associated with tomato fruit ripeness[11]. The RGB to HIS transformation is described in a formula [12][13].

$$\theta = cos^{-1}\{\frac{\frac{1}{2}[(R-G)+(R-B)]}{[(R-G)^2+(R-B)(G-B)]^{\frac{1}{2}}}\} \quad (1)$$

$$H(Hue) = \begin{cases} \theta & if\ B \leq G \\ 360-\theta & if\ B>G \end{cases} \quad (2)$$

$$S(Saturation) = 1 - \frac{3}{R+G+B}[min(R,G,B)] \quad (3)$$

The normalization process involves transforming the Hue value, ranging from 0 to 360, to a normalized scale of 0 to 1 through the equation $H = H/360$ [14] [15] . Subsequently, the acquired saturation value is applied to distinguish foreground objects from the background using the thresholding technique [16].

Aim of this research is to test the effectiveness of the HSI algorithm with RGB color space in detecting ripeness of tomatoes, while developing technology-based applications to support the implementation of this algorithm [17][18]. By combining digital image analysis capabilities and programming technology, it is hoped that this research can make a positive contribution to increasing efficiency and accuracy in fruit ripeness detection [19].

Choosing of the programming language used in implementing this algorithm is in Python [20]. This decision was based on Python's flexibility and ease of use, as well as broad community support in the fields of computer science and image processing [21]. In addition, the results of the algorithm implemented in Python will be integrated with the MATLAB platform [22]. MATLAB was chosen because of its advantages in matrix manipulation, image analysis, and its ability as a complete development environment [23].

Benefits of this research will make a major contribution to the agricultural sector, especially in managing and improving the quality of tomato harvests. It is hoped that the applications resulting from this research can provide practical solutions that can be used by farmers and stakeholders in the agricultural sector. More broadly, this research can provide new understanding of how digital image processing technology and technology-based applications can be applied to improve food security and production efficiency.

2. RESEARCH METHOD

The dataset accessed consists of a number of tomato images downloaded from the open dataset site kaggle.com [24]. These images are used as the basis for extracting the color characteristics of tomatoes. This dataset includes 150 images of tomatoes, consisting of 50 images of ripe tomatoes, 50 images of half-ripe tomatoes, and 50 images of unripe tomatoes, which were selected manually. The images were then divided into two groups, namely 40 training samples and 10 test samples for each level of maturity (ripe, half-cooked and raw). The division of images into training and test sample groups is done manually based on their maturity level. Next, these images undergo transformation into the HSI (Hue, Saturation, Intensity) color space [25]. Training samples from these images will be used in implementing the maturity detection model, while test samples will be used as a tool to test the performance of the model and can be seen from the flow diagram below.

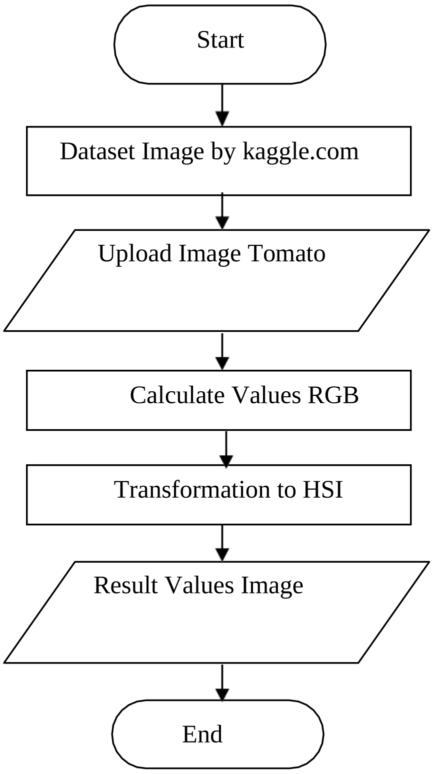

Figure 2. Research Flow Diagram

Steps to implement color space transformation from RGB to HIS on-tomato fruit images. The process begins with the use of an input dataset which contains a number of images of tomatoes with ripe, semi-ripe and unripe levels. This dataset is further divided into two groups, namely training images and testing images. Each group of images then undergoes a transformation process from RGB to HIS color space. The training image group is used to apply the maturity detection model, while the testing image group is used to test the model to obtain accuracy values in detecting the level of maturity in tomatoes

3. RESULTS AND DISCUSSIONS

Steps and process for the RGB color space with the application of algorithms using the Python language with Matlab software are as follows below.

1. GUI display of the application

```
%GUI
function varargout = Main(varargin)

gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @Main_OpeningFcn, ...
                   'gui_OutputFcn',  @Main_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
function Main_OpeningFcn(hObject, eventdata, handles, varargin)

handles.output = hObject;

guidata(hObject, handles);

function varargout = Main_OutputFcn(hObject, eventdata, handles)

varargout{1} = handles.output;
```

Figure 3. GUI display

2. Image input button and coding for image input. Images only use jpg format.

```
function pushbutton1_Callback(hObject, eventdata, handles) %Tombol input image

[filename,pathname] = uigetfile('*.jpg'); %format file hanya jpg
Img = imread(fullfile(pathname,filename));
handles.I = Img;
guidata(hObject,handles)
axes(handles.axes1)
imshow(Img) %menampilkan gambar yang telah dipilih
title(filename); %menampilkan nama gambar yang dipilih

function pushbutton2_Callback(hObject, eventdata, handles)
```

Figure 4. Input format *jpg

3. Extraction feature button, separates RGB values, and looks for HIS values

```
function pushbutton3_Callback(hObject, eventdata, handles) %Tombol extraction feature

% pisah rgb
Img = handles.I;
%cari nilai HSI
RGB     = im2double(Img);
Red     = RGB(:,:,1);
Green   = RGB(:,:,2);
Blue    = RGB(:,:,3);
```

Figure 5. Feature Extraction

4. Find the hue value by looking for the top and bottom values first. Then look for the theta value using the formula that has been entered into the coding, namely cos (top/bottom). And if the blue value >= green, then H = 360-theta. However, if blue <= green, then H = theta

```
%Hue
atas=1/2*((Red-Green)+(Red-Blue));
bawah=((Red-Green).^2+((Red-Blue).*(Green-Blue))).^0.5;
teta = acosd(atas./(bawah));
if Blue >= Green
    H = 360 - teta;
else
    H = teta;
end

H = H/360;
[r c] = size(H);
for i=1 : r
    for j=1 : c
        z = H(i,j);
        z(isnan(z)) = 0; %isnan adalah is not none artinya jika bukan angka dia akan memberi 0
        H(i,j) = z;
    end
end
```

Figure 6. Hue value

5. After that, look for the S and I values by entering the existing formula into the coding

```
%S
S=1-(3./(sum(RGB,3))).*min(RGB,[],3);
[r c] = size(S);
for i=1 : r
    for j=1 : c
        z = S(i,j);
        z(isnan(z)) = 0;
        S(i,j) = z;
    end
end
%I
I=(Red+Green+Blue)/3;
```

Figure 7. Value of S and I

6. Calculate the Mean RGBHSI, Var RGBHSI, and Range RGBHSI values from the data that has been entered

```
MeanR = mean2(Red);
MeanG = mean2(Green);
MeanB = mean2(Blue);
MeanH = mean2(H);
MeanS = mean2(S);
MeanI = mean2(I);
VarRed = var(Red(:)); VarGreen = var(Green(:)); VarBlue = var(Blue(:));
VarH = var(H(:)); VarS = var(S(:)); VarI = var(I(:));
RangeR = ((max(max(Red)))-(min(min(Red))));
RangeG = ((max(max(Green)))-(min(min(Green))));
RangeB = ((max(max(Blue)))-(min(min(Blue))));
RangeH = ((max(max(H)))-(min(min(H))));
RangeS = ((max(max(S)))-(min(min(S))));
RangeI = ((max(max(I)))-(min(min(I))));
data = get(handles.uitable2,'Data');
data{1,1} = num2str(MeanR);
data{2,1} = num2str(MeanG);
data{3,1} = num2str(MeanB);
data{4,1} = num2str(MeanH);
data{5,1} = num2str(MeanS);
data{6,1} = num2str(MeanI);

data{1,2} = num2str(VarRed);
data{2,2} = num2str(VarGreen);
data{3,2} = num2str(VarBlue);
data{4,2} = num2str(VarH);
data{5,2} = num2str(VarS);
data{6,2} = num2str(VarI);

data{1,3} = num2str(RangeR);
data{2,3} = num2str(RangeG);
data{3,3} = num2str(RangeB);
data{4,3} = num2str(RangeH);
data{5,3} = num2str(RangeS);
data{6,3} = num2str(RangeI);
```

Figure 8. Means value

7. Then classify the ripeness of the tomatoes

```
%pengklasifikasian kematangan tomat
set(handles.uitable2,'Data',data,'ForegroundColor',[0 0 0])
trainingl = xlsread('Data Training');
group = trainingl(:,25);
training = [trainingl(:,1) trainingl(:,2) trainingl(:,3) trainingl(:,4) trainingl(:,5) trainingl(:,6)
trainingl(:,7) trainingl(:,8) trainingl(:,9) trainingl(:,10) trainingl(:,11) trainingl(:,12) trainingl(:,13)
trainingl(:,14) trainingl(:,15) trainingl(:,16) trainingl(:,17) trainingl(:,18)];
Z=[MeanR MeanG MeanB MeanH MeanS MeanI VarRed VarGreen VarBlue VarH VarS VarI RangeR RangeG RangeB RangeH
RangeS RangeI];
hasill=knnclassify(Z,training,group); %metode hitung data

if hasill==1
    x='MATURE';
elseif hasill==2
    x='HALF-MATURE';
elseif hasill==3
    x='IMMATURE';
end
set(handles.edit2,'string',x);
```

Figure 9. Training Classification

8. Added training data button

```
function pushbutton4_Callback(hObject, eventdata, handles) %tombol training data matang
```

Figure 10. Training Process

9. Inserting ripe tomato training data into the specified folder

```
image_folder = 'D:\Tubes PCD\Training\mature';
filenames = dir(fullfile(image_folder, '*.jpg'));
total_images = numel(filenames);
Z1=[];
for n = 1:total_images
  full_name= fullfile(image_folder, filenames(n).name);
  Img = imread(full_name);
```

Figure 11. Input Training to Folder

10. Looking for HSI values from ripe tomato training data

```
%cari nilai HSI
RGB     = im2double(Img);
Red     = RGB(:,:,1);
Green   = RGB(:,:,2);
Blue    = RGB(:,:,3);
```

Figure 12. Search for HSI training to RGB values

11. Find the hue value from the ripe tomato training data using the same formula when determining tomato ripeness.

```
%Hue
atas=1/2*((Red-Green)+(Red-Blue));
bawah=((Red-Green).^2+((Red-Blue).*(Green-Blue))).^0.5;
teta = acosd(atas./(bawah));
if Blue >= Green
    H = 360 - teta;
else
    H = teta;
end

H = H/360;
[r c] = size(H);
for i=1 : r
    for j=1 : c
        z = H(i,j);
        z(isnan(z)) = 0;
        H(i,j) = z;
    end
end
```

Figure 13. Hue Value Search

12. Find the value of S and I from the training data of ripe tomatoes

```
%S
S=1-(3./(sum(RGB,3))).*min(RGB,[],3);
[r c] = size(S);
for i=1 : r
    for j=1 : c
        z = S(i,j);
        z(isnan(z)) = 0;
        S(i,j) = z;
    end
end
%I
I=(Red+Green+Blue)/3;
```

Figure 14. Search for S and I Values

13. Find the Mean RGBHSI, Var RGBHSI, and Range RGBHSI values

```
MeanR = mean2(Red);
MeanG = mean2(Green);
MeanB = mean2(Blue);
MeanH = mean2(H);
MeanS = mean2(S);
MeanI = mean2(I);
VarRed = var(Red(:));
VarGreen = var(Green(:));
VarBlue = var(Blue(:));
VarH = var(H(:));
VarS = var(S(:));
VarI = var(I(:));
RangeR = ((max(max(Red)))-(min(min(Red))));
RangeG = ((max(max(Green)))-(min(min(Green))));
RangeB = ((max(max(Blue)))-(min(min(Blue))));
RangeH = ((max(max(H)))-(min(min(H))));
RangeS = ((max(max(S)))-(min(min(S))));
RangeI = ((max(max(I)))-(min(min(I))));

sdR = std2(Red);
sdG = std2(Green);
sdB = std2(Blue);
sdH = std2(H);
sdS = std2(S);
sdI = std2(I);
```

Figure 15. RGB Means Value

14. Save the ripe tomato training data to variable z and will be stored in variable z1

```
% menyimpan semua data training yang sudah dihitung ke variabel z
Z=[MeanR MeanG MeanB MeanH MeanS MeanI VarRed VarGreen VarBlue VarH VarS VarI RangeR RangeG RangeB RangeH
    RangeS RangeI sdR sdG sdB sdH sdS sdI 1];
Z1=[Z1;Z]; %Z1 menyimpan keseluruhan data
end
```

Figure 16. Training results data is saved

The stages of the application that are adapted to the algorithm that has been applied in determining the level of fruit ripeness by transforming the RGB to HIS color space can be seen from the steps below.

1. First Stage Application Menu

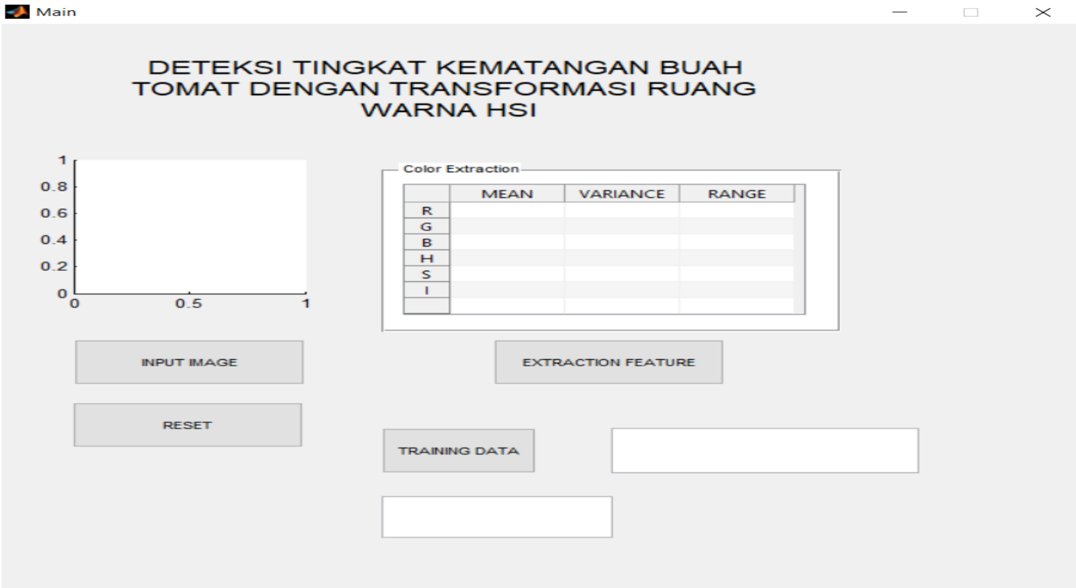

Figure 17. Application Main Menu

In the initial stage, this application describes the main menu of the application, in the process of determining fruit ripeness, namely input image to determine the image to be processed, and then features are extracted to determine the RGB HIS value, with training data in *jpg format.

2. The second stage is the upload and reset application menu

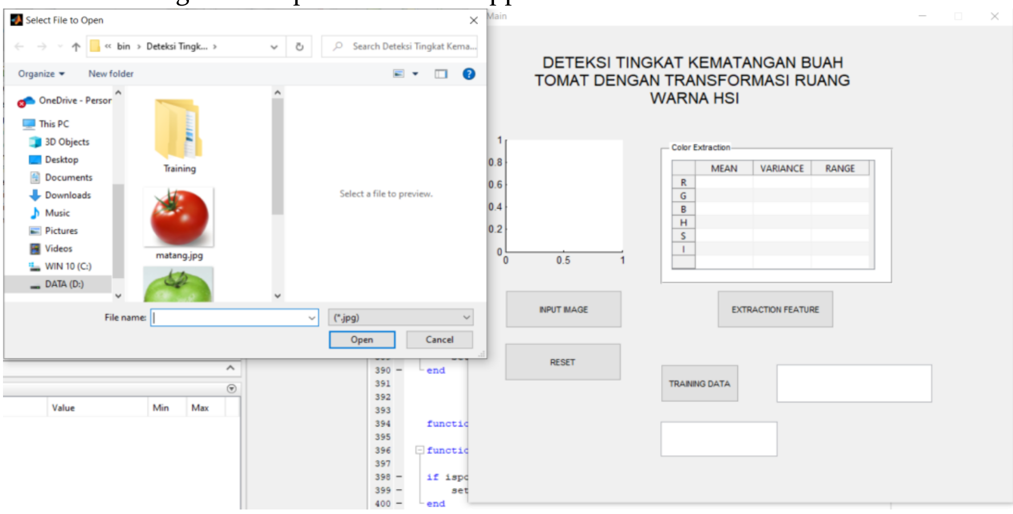

Figure 18. Display of the upload image button

3. The third stage of the ripe tomato data display menu

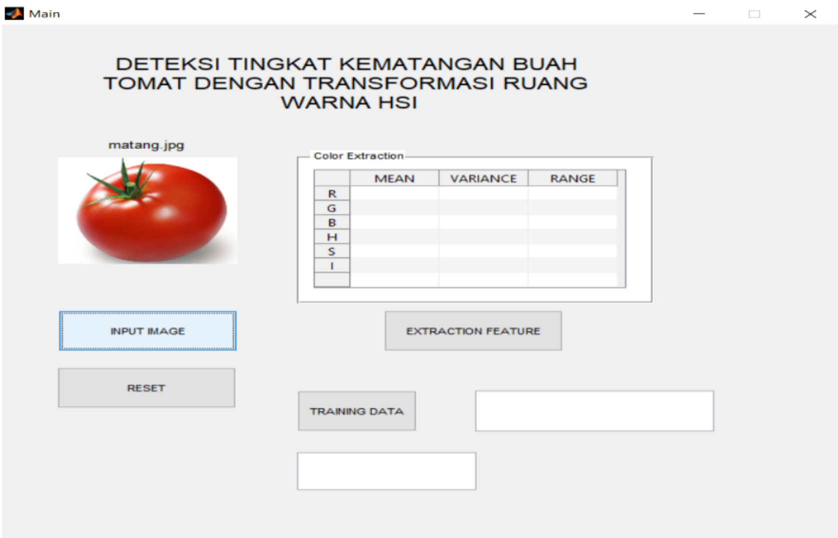

Figure 19. Ripe Tomato image data display

Then select the tomato whose ripeness you want to detect and click open. Then the selected image will appear along with its file name.

4. The fourth stage of the ripe tomato extraction process

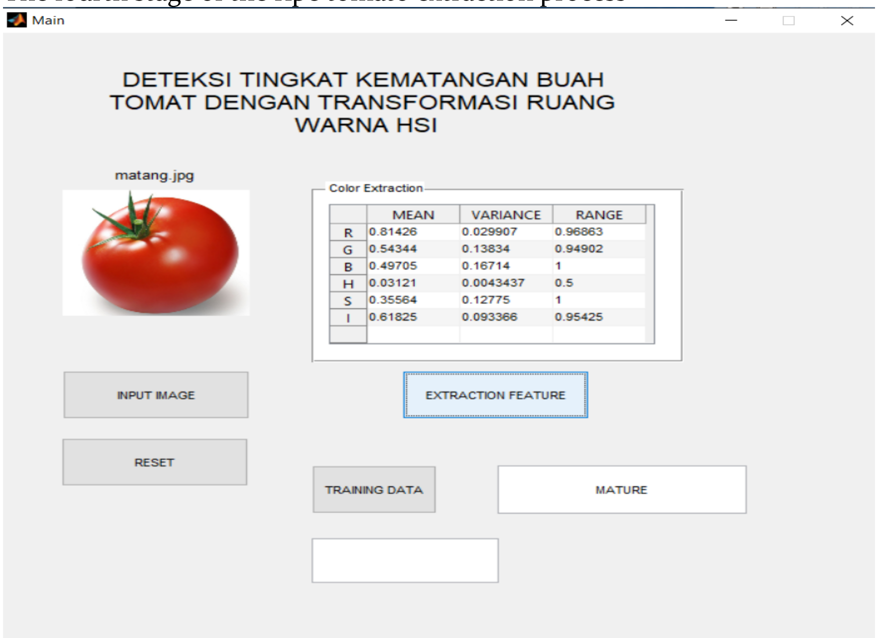

Figure 20. Ripe Tomato Extraction Process

After that, click extraction feature so that the color extraction value will appear in the table with the mean, var and range values of RGBHSI. Apart from that, the maturity level of the tomatoes will also appear in the column to the right of the training data button in the form of mature, half-mature, and immature.

5. The fifth stage resets or deletes the tomato image

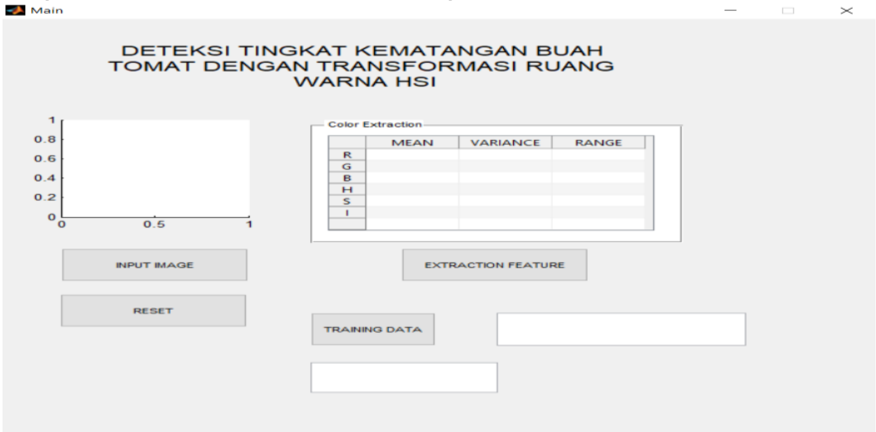

Figure 21. Reset process

6. Clicking the training data button to enter the values from the selected image into the training data file in Excel format. The data will automatically be entered into the Excel training data file

Picture 22. Half Ripe Tomato Training

7. Test with half ripe tomatoes

Figure 23. Extraction results of half-ripe tomatoes

8. Testing with unripe tomatoes

Figure 24. Extraction results of immature tomatoes

Results of the application training above, it can be concluded that the range values for ripe tomatoes cover the range H = 19.00936 - 32.572994, S = 0.52890784 - 0.9267963, and I = 0.13458732 - 0.41211584. The range values for semi-ripe tomatoes lie at H = 37.176495 - 58.759083, S = 0.6161851 - 0.96461624, and I = 0.36128944 - 0.54107976. For immature tomatoes, the H value ranges between 34.39732 - 82.86747, S = 0.37820813 - 0.9124262, and I = 0.21342424 - 0.61382484.

4. CONCLUSION

Conclusions obtained from several sample tests that have been tested with RGB color space values and the HIS algorithm in implementation using Python language and Matlab software as a graphical user interface.

1. The search for pixel values has a significant impact in this research, where the process of decomposing RGB color images into HIS (Hue, Intensity, Saturation) color space plays a key role in fruit ripeness analysis. The research results show the HSI value for each class, with the ripe tomato class having an average hue of 0.0051 – 0.026, saturation 0.1862 – 0.3291, and intensity 0.0975 – 0.7522.
2. The use of global colors in RGB images stacked in three layers can be susceptible to high lighting intensity. This weakness can affect test results, where excessive light absorption on objects can reduce the average RGB value.
3. Tomatoes can be detected automatically using a computer via the MATLAB R2019b application with the RGB method which is transformed to HIS. The integration of HSI algorithm steps with the RGB color space provides an additional dimension to color analysis, which has the potential to increase the accuracy of tomato ripeness detection

REFERENCES

[1] C. Wibowo, R. Wicaksono, P. Haryanti, D. M. Irawan, S. B. Sulistyo, and A. Fatoni, "Application of Starch-based Edible Coating on Tomato and Its Effect during Storage," in *IOP Conference Series: Earth and Environmental Science*, 2023, vol. 1155, no. 1. doi: 10.1088/1755-1315/1155/1/012014.

[2] K. L. Kamminga, A. L. Koppel, D. A. Herbert, and T. P. Kuhar, "Biology and management of the green stink bug," *J. Integr. Pest Manag.*, vol. 3, no. 3, 2012, doi: 10.1603/IPM12006.

[3] R. Paraguison-Alili, R. M. Dela Cruz, E. S. Romero, H. M. Cruz, C. O. Padilla, and F. L. Porciuncula,

"Molecular tools in monitoring the outset of tomato leaf curl virus (ToLCV) Transmission in the Philippines," *Curr. Plant Biol.*, vol. 24, 2020, doi: 10.1016/j.cpb.2020.100169.

[4] R. Pratama *et al.*, "Tomato Fruit Detection Detection Based on Color Features Using His Color Space Transformation Method," *J. Inform. dan Komputer) p-ISSN*, vol. 2, no. 2, 2019.

[5] D. Ayuningtyas, E. Suryani, and W. Wiharto, "Identification of Tomato Maturity Based on HIS Color Space Using the K-Nearest Neighbour Method," 2021. doi: 10.1109/ICAICST53116.2021.9497843.

[6] A. S. Hayat, M. Syukri, and S. Sakir, "Identification of Ripe Levels in Plantain Fruits Using Hue Saturation Intensity (HSI) Color Space Transformation Method," *Tekper J. Teknol. dan Manaj. Ind. Pertan.*, vol. 2, no. 1, 2021, doi: 10.33772/tekper.v2i1.17291.

[7] A. I. Thoriq, M. H. Zuhri, P. Purwanto, P. Pujiono, and H. A. Santoso, "Classification of Banana Maturity Levels Based on Skin Image with HSI Color Space Transformation Features Using the K-NN Method," *J. Dev. Res.*, vol. 6, no. 1, 2022, doi: 10.28926/jdr.v6i1.200.

[8] B. Yanto, J. Jufri, A. Lubis, B. H. Hayadi, and E. Armita, NST, "Klarifikasi Kematangan Buah Nanas Dengan Ruang Warna Hue Saturation Intensity (Hsi)," *INOVTEK Polbeng - Seri Inform.*, vol. 6, no. 1, p. 135, 2021, doi: 10.35314/isi.v6i1.1882.

[9] A. Wibowo, D. M. C. Hermanto, K. I. Lestari, and H. Wijoyo, "Deteksi Kematangan Buah Jambu Kristal Berdasarkan Fitur Warna Menggunakan Metode Transformasi Ruang Warna Hsv (Hue Saturation Value) Dan K-Nearest Neighbor," *INCODING J. Informatics Comput. Sci. Eng.*, vol. 1, no. 2, 2021, doi: 10.34007/incoding.v2i1.131.

[10] F. Aprilliani, D. Atmiasih, and A. Ristiono, "THE EVALUATION OF AVOCADO (Persea americana Mill.) MATURITY LEVEL USING IMAGE PROCESSING TECHNOLOGY," *J. Penelit. Pascapanen Pertan.*, vol. 18, no. 1, 2021, doi: 10.21082/jpasca.v18n1.2021.1-8.

[11] F. Putra and T. Rizaldi, "Application of Color Sensor in the Determination of Tomato Fruit Ripeness (Solanum Lycopersicum, L) in Gravitation Type Fruit Sorting Tool (Gravitation Type)," *Indones. J. Agric. Res.*, vol. 2, no. 1, 2019, doi: 10.32734/injar.v2i1.862.

[12] Y. Li *et al.*, "Method for wheatear counting and yield predicting based on image of wheatear population in field," *Nongye Gongcheng Xuebao/Transactions Chinese Soc. Agric. Eng.*, vol. 34, no. 21, 2018, doi: 10.11975/j.issn.1002-6819.2018.21.022.

[13] X. Yang, D. Zhu, R. Yang, X. Zuo, W. Xie, and Z. Fu, "Method for extracting UAV RGB image information based on matching point cloud and HSI color component," *Nongye Gongcheng Xuebao/Transactions Chinese Soc. Agric. Eng.*, vol. 37, no. 22, 2021, doi: 10.11975/j.issn.1002-6819.2021.22.034.

[14] M. Kamiyama and A. Taguchi, "Color conversion formula with saturation correction from HSI color space to RGB color space," *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, vol. E104A, no. 7, 2021, doi: 10.1587/transfun.2020EAL2087.

[15] B. Yanto, J. Jufri, A. Lubis, B. H. Hayadi, and E. Armita, NST, "KLARIFIKASI KEMATANGAN BUAH NANAS DENGAN RUANG WARNA HUE SATURATION INTENSITY (HSI)," *INOVTEK Polbeng - Seri Inform.*, vol. 6, no. 1, 2021, doi: 10.35314/isi.v6i1.1882.

[16] Z. Zhang and J. Fan, "Circular Histogram Thresholding Method Based on S-component Exponential Weighted H-component," *Jisuanji Gongcheng/Computer Eng.*, vol. 48, no. 10, 2022, doi: 10.19678/j.issn.1000-3428.0062899.

[17] C. Dewi, A. Santoso, I. Indriati, N. A. Dewi, and Y. K. Arbawa, "Evaluasi Performasi Ruang Warna pada Klasifikasi Diabetic Retinophaty Menggunakan Convolution Neural Network," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 8, no. 3, 2021, doi: 10.25126/jtiik.2021834459.

[18] B. Yanto, E. Rouza, L. Fimawahib, B. H. Hayadi, and R. R. Pratama, "Penerapan Algoritma Deep Learning Convolutional Neural Network Dalam Menentukan Kematangan Buah Jeruk Manis Berdasarkan Citra Red Green Blue (RGB)," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 10, no. 1, 2023, doi: 10.25126/jtiik.20231015695.

[19] R. S. Raghavendra *et al.*, "Deep Learning Based Dual Channel Banana Grading System Using Convolution Neural Network," *J. Food Qual.*, vol. 2022, 2022, doi: 10.1155/2022/6050284.

[20] M. Fahmi Wibawa, M. A. Rahman, and A. W. Widodo, "Penerapan Ruang Warna HSV dan Ekstraksi Fitur Tekstur Local Binary Pattern untuk Tingkat Kematangan Sangrai Biji Kopi," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 5, no. 7, 2021.

[21] Y. Hendrawan, A. Amini, D. M. Maharani, and S. M. Sutan, "Intelligent non-invasive sensing method in identifying coconut (coco nucifera var. ebunea) ripeness using computer vision and artificial neural network," *Pertanika J. Sci. Technol.*, vol. 27, no. 3, 2019.

[22] T. H. Nasution, M. Rumansa, and H. Lukman Adlin, "Designing the quality of coffee bean detection application using Hue Saturation Intensity," in *IOP Conference Series: Materials Science and Engineering*, 2019, vol. 648, no. 1. doi: 10.1088/1757-899X/648/1/012036.

[23] S. Aini, I. G. P. S. Wijaya, and G. W. Wiriasto, "Klasifikasi Kesegaran Buah Apel Berdasarkan Ekstraksi Fitur Warna Dan Tekstur Menggunakan Gray Level Co-Occurrence Matrix Dan Algoritma K-Nearest Neighbor," *J. Tek. Elektro*, vol. 16, 2018.

[24] V. Sudha and T. R. Ganeshbabu, "A convolutional neural network classifier VGG-19 architecture for

lesion detection and grading in diabetic retinopathy based on deep learning," *Comput. Mater. Contin.*, vol. 66, no. 1, 2021, doi: 10.32604/cmc.2020.012008.

[25] S. Zhi, Y. Cui, J. Deng, and W. Du, "An FPGA-based simple RGB-HSI space conversion algorithm for hardware image processing," *IEEE Access*, vol. 8, 2020, doi: 10.1109/ACCESS.2020.3026189.