Journal of Information Systems and Informatics
Vol. 5, No. 4, December 2023 e-ISSN: 2656-4882 p-ISSN: 2656-5935
DOI: 10.51519/journalisi.v5i4.613 Published By DRPM-UBD

Classification of Tomato Ripeness Based on Convolutional Neural Network Methods

Nur Azizah Ayunda[1], Emy Haryatmi[2], Tri Agus Riyadi[3]

[1,2]Master of Electrical Engineering, Universitas Gunadarma, Jakarta, Indonesia
[3]Informatics Departement, Universitas Gunadarma, Jakarta, Indonesia
Email: [1]nurazizahayunda14@gmail.com, [2]emy_h@staff.gunadarma.ac.id,
[3]ta_riyadi@staff.gunadarma.ac.id

Abstract

Sorting system for tomato is one of the important things to deploy to achieve better quality of tomato. Nowadays, many sorting system is done manually and this could spend a lot of time and become inefficient. One method can be implemented in the sorting system by using Convolutional Neural Network (CNN) method to classify the ripeness of tomatoes. The objective of this research is to classify the ripeness of tomatoes based on the color of tomatoes. There are three categories of color level such as green for raw tomato, turning for half-ripe tomato and red for ripe tomato. Research methodology of this research is data collection, data pre-processing and image maintenance, CNN model, and training data. The image used in this research are 1148 images. These images were taken manually using smartphone camera in outdoor environment. These images were used to build CNN model. The results of this research show that by testing 10 images of tomatoes achieved raw tomatoes close to 90%, ripe tomatoes close to 90% and half-ripe tomatoes close to 80%. Based on the results, CNN can be used as a good alternative in image classification tasks.

Keywords: CNN model, Classification, Tomato Ripeness, Image Processing

1. INTRODUCTION

Agriculture plays an important role in the country's economy. It has several agricultural products including mangoes, bananas, papayas, durians, salak, tomatoes, and others[1]. Tomato is a fruit that is widely consumed in the world and one of the fruits that has a certain level of maturity in a short period of time[2]. The external features of tomatoes are widely used to assess their ripeness, which is judged by color, size and shape. The maturity of tomatoes is usually determined by the color of the surface area because it is very important in assessing the quality or ripeness of tomatoes, and most buyers often choose tomatoes based on their color[3].

p-ISSN: 2656-5935 http://journal-isi.org/index.php/isi e-ISSN: 2656-4882

Tomatoes are one of the fruits that are very sensitive in the planting process. Therefore, the quality of tomatoes needs to be maintained consistently to increase the selling value of tomatoes by paying attention to the level of tomato ripeness[4]. Previously, manual determination of the ripeness of tomatoes had its drawbacks. The process takes a relatively long time, requires a lot of labor, and can cause inconsistencies in determining the ripeness of tomatoes[5]. Determination of the ripeness of tomatoes can be done by using automatic detection system. Digital image can be used for automatic detection system. A digital image is a representation, likeness, or imitation of an object. Image output of data recording systems can be optical in the form of photos, digital which can be directly stored on a storage medium[6].

Automatic detection system of the ripeness of tomatoes is used in tomatoes sorting system to achieve better quality of tomatoes. Nowadays, manual method is used to sorting the ripeness of tomatoes. Tomatoes are selected manually one by one according to the level of maturity from color, shape or size. This method spends a lot of time and inefficient. The solution to this problem is to build a tomato sorting system automatically by implementing Convolutional Neural Network (CNN) to classify the maturity of tomatoes.

CNN, a type of Artificial Intelligence, uses Deep Learning algorithms that can take image input and can classify an object. CNN's main task in Image Classification is to receive image input and follow the meaning of the image[3]. The system can recognize the quality of tomatoes according to three categories: raw, half-ripe and ripe[7]. Using tomato images as a control system is not an easy challenge. This is due to the presence of tomatoes taken with various backgrounds, changing lighting conditions and different colors of tomatoes[8]. The maturity of tomatoes by color was also defined in[5] by using CNN. They used red, green, and yellow color to differentiate the maturity of tomatoes automatically.

Many researchers have done the research regarding implementation of CNN to classify or detect an object. Research on maturity with different objects was conducted by Raymond Erz Saragih and W. R. Emanuel. The research was done using CNN to classify the ripeness of the banana. In this research, banana's ripeness was divided into four classess such as unripe/green, yellowish-green, mid-ripen and overripe. They used two models, namely MobileNet V2 and NASNetMobile. The higher accuracy was 96.18% achieved by using MobileNet V2. The faster execution was also achieved by MobileNet V2 than NASNetMobile[9].

Another research using CNN method was conducted by Chrisno R. Kotta, Debby Paseru, and Michael Sumampouw to detect diseases in tomato leaf image. This research was using Phyton and Google Colab and implemented in Android

p-ISSN: 2656-5935 http://journal-isi.org/index.php/isi e-ISSN: 2656-4882

Application. The accuracy was 94% when testing on images from the gallery and 80% when testing images directly from the camera[10].

Another research using CNN was also done by Yanto et al and the object was sweet orange. They classified two different sweet orange such as good and rotten. The accuracy rate was 97.5184% by using 100 datasets of orange[11]. Another study using CNN to detect the ripeness of fruit was done by Josua and Said[12]. They used 792 images of citrus fruit and classify them into 2 categories such as raw and ripe. The accuracy rate was 86.59% after using 40 epochs.

Another research using CNN was also done by Yuhandri et al. They used this method to detect mask in real time which used by people during covid-19. They used 2000 images from github.com and kaggle.com. The accuracy rate was 99.05%[13]. TiaraSari and Emy were also using CNN to detect dry corn kernels. They used 20 images for testing and 80 images for training datasets. The accuracy rate was around 80% - 100% by using 7 convolutional layers[14]. Novi Sudiati also used CNN to detect kind of spices. She used garlic, chilies, cinnamon, turmeric, cloves, ginger, pepper, basil, lemongrass, and coriander as objects. Based on testing process, CNN could identify spices from features and patterns[15]. CNN can also be used to detect and count the number of people in room as describe in the research of Indana et al[16]. They used faces and heads images to detect and count the people.

Based on the description above, the purpose of the study is to implement CNN method to classify tomato ripeness based on tomato skin color. The color level of tomato ripeness consists of unripe tomatoes of green color, semi-ripe tomatoes of turning color, and ripe tomatoes of red color. So, it is necessary to build a system using CNN mainly to classify tomato ripeness, with three levels of tomato ripeness that can be distinguished by the color of the tomato. Thus, tomato color become an important indicator in determining the level of maturity and quality of tomato tomatoes[17]. It could facilitate and improve the performance of farmers during preharvest or post-harvest in choosing the level of tomato maturity.

2. METHODS

In this study, a tomato ripeness classification system is designed to determine accuracy using digital image processing methods. Figure 1 shows a block diagram of the system designed in this study. In general, a structured system block diagram as shown in Figure 1 consists of tomato data collection stage, pre-processing and image maintenance stage with sizing adjustment stages, data cropping and augmentation, CNN model, training, and testing [4], [18]

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

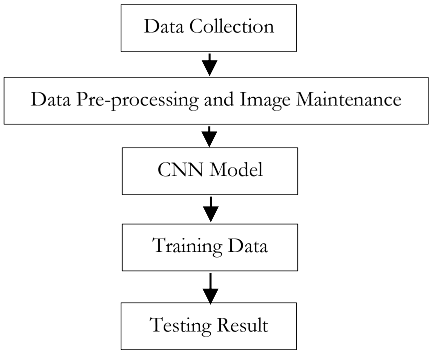


Figure 1. General System Block Diagram

2.1 Data Collection

In image data collection, images were taken by placing tomatoes on white paper to get images with a white background because it equalized the situation of each tomato. Tomatoes image was taken in outdoor environment with sufficient lightning from the sun between 13.00-14.00 using camera in the smartphone. The resulting image was clearly visible. The specification of the smartphone was Vivo 21S with 50 MP camera. Distance from tomatoes and smartphone was 20 cm. The number of photos collected was 1148 images. All images that had been taken from smartphone cameras were put into each folder with separate folders of raw tomatoes, half-ripe tomatoes and ripe tomatoes. Tomato images were taken from various sides in order to get more image data from all sides. This makes it easier for researchers to determine the classes of tomatoes.

2.2 Data Pre-processing and Image Maintenance

In this pre-process stage, all data pre-process was prepared to undergo the training process. the data would be resized, cropping and augmentation data[3]. The Pre-processing and target steps can be seen in figure 2. The result of pre-process stage can be seen in figure 3. The first step was image resize. Images resulted from collecting data was in 250x250 pixel. These images were resized into 150x150 pixel in order to fit into first convolutional layer and other layer used in this research. The second step was image cropping. Image cropping was used to focus on the tomatoes rather than the background of the image. The image was cropped into 15x15 pixel to focus on the tomato in the image. The third step was augmentation. The augmentation was done to recognize tomato image in any position such as rescale, shear_range, zoom_range and horizontal_flip. The images were rescaled into 1/255. The shear_range of the image was set into 0.2. All images would oblique distortion up to 20% from the original dimension in random direction.

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

Zoom_range of the image was set into 0.2. All images would enlarge up to 20%. Horizontal_flip was used to flip half of the image horizontally at random.

Figure 2. General System Block Diagram

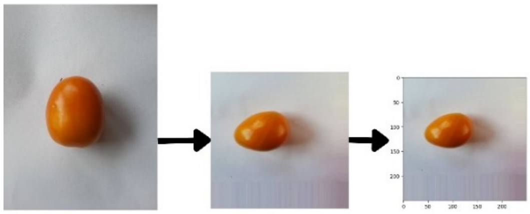


Figure 3. Examples of Resize, Cropping and Augmentation Results from the Image of Half-Ripe Tomatoes

2.3. CNN Model

The architecture of CNN can be found in figure 4. There are two main process such as future learning and classification. Future learning consists of convolution layer, ReLu and Max Pooling. Classification consists of Flutten, fully connected and Softmax.

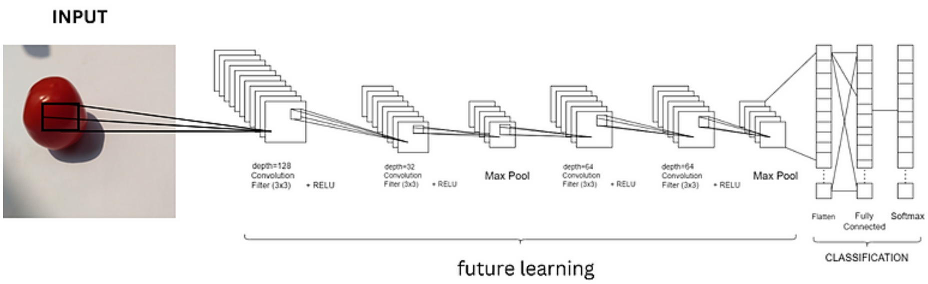


Figure 4. CNN Architecture

The model above contained several different types of layers, including the convolution layer (Conv2D) as image convolution proses with various carnels, layer pooling (MaxPooling2D) as a subsampling process where matrix max pooling can be adjusted as needed, the flatten layer (Flatten), and the dense layer (Dense) with many and predetermined characteristics. The convolution process was carried out 4 times indicated by the number of convolution layers used.

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

Generally, 2 – 3 layers were sufficient to get a classification model. However, in this study, a deeper or more network was used to train the model and see how the model performed.

While the activation function used was 'relu', it would make the training process faster. The kernel or filter size used for each convolution layer was 3x3. While the pooling size used was 2x2.

Table 1. Summary of CNN Model

Layer (type)	Output Shape	Param #
conv2d_4 (Conv2D)	(None, 150, 150, 128)	3584
activation_6 (Activation)	(None, 150, 150, 128)	0
conv2d_5 (Conv2D)	(None, 148, 148, 32)	36896
activation_7 (Activation)	(None, 148, 148, 32)	0
max_pooling2d_2 (MaxPooling2D)	(None, 74, 74, 32)	0
dropout_5 (Dropout)	(None, 74, 74, 32)	0
conv2d_6 (Conv2D)	(None, 74, 74, 64)	18496
activation_8 (Activation)	(None, 74, 74, 64)	0
conv2d_7 (Conv2D)	(None, 72, 72, 64)	36928
activation_9 (Activation)	(None, 72, 72, 64)	0
max_pooling2d_3 (MaxPooling2D)	(None, 36, 36, 64)	0
dropout_6 (Dropout)	(None, 36, 36, 64)	0
flatten_1 (Flatten)	(None, 82944)	0
dense_6 (Dense)	(None, 512)	42467840
activation_10 (Activation)	(None, 512)	0
dropout_7 (Dropout)	(None, 512)	0
dense_7 (Dense)	(None, 3)	1539
activation_11 (Activation)	(None, 3)	0

Total params: 42,565,283
Trainable params: 42,565,283
Non-trainable params: 0

The number of parameters trained in this model was 42,565,283 parameters. The image size in the last convolution process before entering the fully connected layer was 150x150 pixels. While the number of filters used was 128. Then reshape so that there were 82944 neurons that would enter the fully connected layer. The number of neurons in the hidden layer used was 512 neurons. The fully connected layer also added a dropout process when conducting training, until finally 3 categories were classified.

Table 2. CNN Model

No	Name	Size	Parameters
0	Input	250 x 250	0
1	conv2d (Conv2D)	150 x 150 x 128	3584
2	conv2d_1(Conv2D)	148 x 148 x 32	36896
3	max_pooling2d(MaxPooling2D)	74 x 74 x 32	0

p-ISSN: 2656-5935 http://journal-isi.org/index.php/isi e-ISSN: 2656-4882

No	Name	Size	Parameters
4	conv2d_2(Conv2D)	74 x 74 x 64	18496
5	conv2d_3(Conv2D)	72 x 72 x 64	36928
6	max_pooling2d_1(MaxPooling2D)	36 x 36 x 64	0
7	flatten (Flatten)	82944(36*36*64)	0
8	dense (Dense)	512	(82944*512)+512 =42467840
9	dropout_2 (Dropout)	512	0
10	Dense_1 (Dense)	3	(3*512)+3 = 1539

Table 2 was the calculation of inputs into convolutions. The number 82944 in the flatten layer was a number obtained from the multiplication of the last maxpooling dimension, which was 36 x 36 x 64 = 82944 and in the dense layer 512 was a number that showed the number of neurons used. The parameter of 42467840 was obtained from (82944 x 512) +512 = 42467840. In the dense layer, it showed the 3 image categories used, namely raw, half-cooked, and cooked. Therefore, the parameters obtained were 3 x 512 + 3 = 1539. The total parameters obtained from the models created were 42,565,283.

2.4 Training Data

In the training process, it had a total of 1148 data, the dataset split process was data train and data test. The comparison of training data and testing data was 80:20. It consisted of 918 training data and 115 of testing data. The testing data was 115 data which consisted of 35 mature class images, 40 raw class images, and 40 half-baked class images. The training data was 918 data which consisted of 340 mature class images, 318 raw class images and 260 half-baked class images.

In the split process, it would call the directory dataset containing raw tomato datasets, half-ripe tomatoes and ripe tomatoes. If the training data was able to learn the data well, there was a chance that the training data would also be able to learn new data, namely testing data. But on the other hand, if the training data was not able to learn the data well, then the training data would not be able to learn the testing data, where this condition would cause an InvalidArgumentError. In machine learning there were terms epoch and batch size when training. In this training, epoch = 5, batch size = 16 and step per epoch = 66 were used.

The value of loss and accuracy of training data can be found in figure 5. The result of epoch 1/5 loss is 0.3646 which means 36%. The accuracy is 0.8732 which means 87%. The val_loss is 0.5048 which means 50%. The val_accuracy is 0.7641 which means 76%. The result of epoch 4/5 loss is 0.1023 which means 10%. The accuracy is 0.9672 which means 96%. The val_loss is 0.0959 which means 9%. The val_accuracy is 0.9789 which means 97%. There are 5/5 Epoch used in this research because it already meets the condition of accuracy. The experiment was

p-ISSN: 2656-5935 http://journal-isi.org/index.php/isi e-ISSN: 2656-4882

also done for 80 Epoch, but it would stop in 28/28 Epoch because already meet the condition of accuracy.

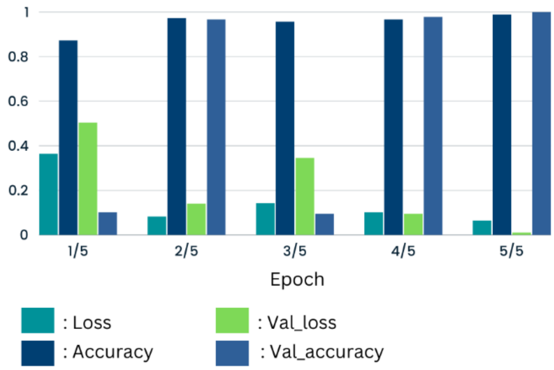


Figure 5. Loss and Accuracy of Training Data

3. RESULTS AND DISCUSSION

3.1 Data Testing dan Analysis

The accuracy and loss with model plot could be seen in figure 6 and Figure 7. There were graphs of accuracy and loss models of data train and data validation.

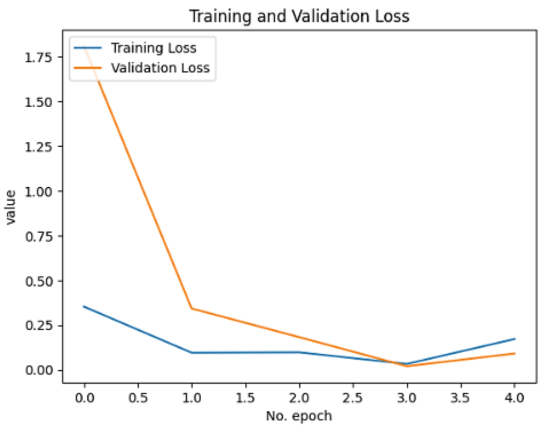


Figure 6. Graphic Training Loss

Figure 6 shows the loss associated with all possibilities produced by the model. Training data and validation was tested to predict the lowest error target. The loss value resulted from the model in training loss is 0.1386. The loss value resulted from the model in validation loss is 0.0939. These two values are the lowest value and it showed that it is good for the loss function.

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

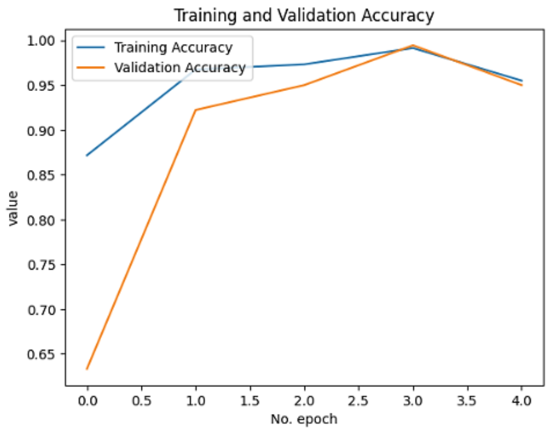


Figure 7. Graphic Training Accuracy

Figure 7 shows the accuracy of training and validation. It predicts true for all data. Training data and validation are tested to classify into correct class. The highest accuracy value is 0.9474 for training data and 0.0939 for testing data. It shows that this model can classify correctly.

Based on the both parameter such as loss and accuracy shows that the model can classified the type of tomatoes correctly. As epoch value increasing, loss value is close to zero or less than one and accuracy value continues increasing. This value shows good result of loss and accuracy. Low value in graphic training loss indicates minimum errors. High value in graphic training accuracy can read and differentiate the model classification in each class.

3.2 Confusion Matrix

The prediction results from the model on testing data showed that the results of predictions of mature conditions were classified into mature as presented in Figure 8. This mean that the classification of the corrected image as mature as 386 and missing data from mature inputs was classified as half-baked as many as 41 data. Predictions of raw conditions were classified into raw; this means that the classification of the image was correct as raw 385 data and missing data from raw inputs was classified as half-baked as much as 1 data. Predictions of half-baked conditions were classified as half-raw; this means that the classification of the image was correct as half-baked 302 data and missing data from half-raw inputs was classified as mature as 33 data. Then, the classification report because of classification process can be found in Figure 9.

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

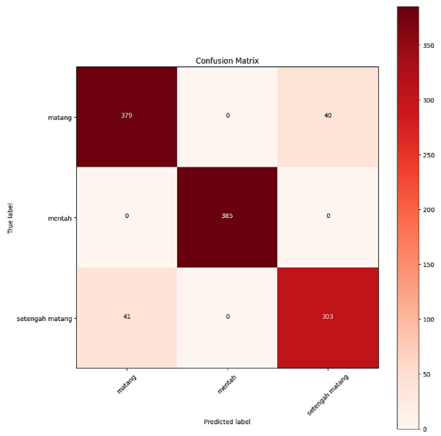


Figure 8. Confusion Matrix

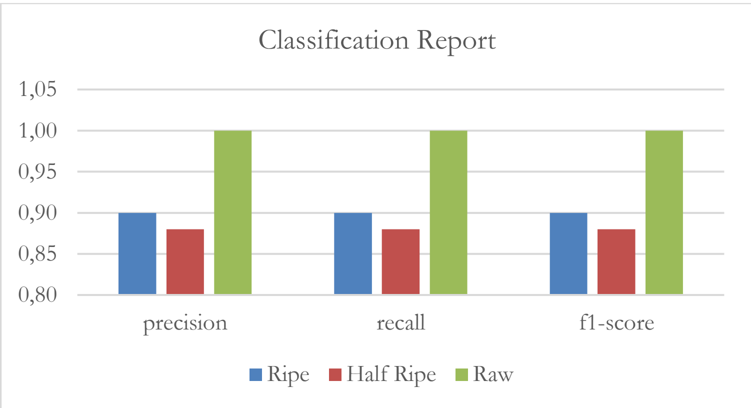


Figure 9. Classification Report of Tomatoes Ripeness

The classification report shows that the value of precision, recall and f1-score for ripe tomatoes are 0.90, 0.90 and 0.90 respectively. The value of precision, recall and f1-score for half ripe tomatoes are 0.88, 0.88 and 0.88 respectively. The value of precision, recall and f1-score for raw tomatoes are 1, 1, and 1 respectively. The calculation process was based on the values mentioned earlier. The accuracy, precision, recall and f1-score in percentage value could be calculated using a formula in the following process:

1) Accuracy

Accuracy was the presentation of the radio's prediction of the overall correct data. In this study, the accuracy of the prediction ratio was calculated correctly by the

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

system to the overall mature, raw, and half-baked data. The calculation was shown i Equation 1 [19].

$$\text{Accuracy } \% = \frac{TP+TN}{TP+TN+FP+FN} \text{ x } 100\% \tag{1}$$

$$\text{Accuracy} = \frac{386+385+302}{386+385+302+41+1+33} \text{ x } 100\%$$

$$\text{Accuracy} = \frac{1073}{1148} \text{ x } 100\%$$

$$\text{Accuracy} = 93\%$$

Where: TP = True positive, TN = True negative, FP = False positive, and FN = False negative. The accuracy of the classification report to present the entire data got 93% results. It proved in observations on Learning Outcomes produced accuracy above 90%.

2) Precision
This study presented several prediction opportunities into mature classes that correspond to reality, namely all mature data in the dataset. The calculation was shown in Equation 2 [20].

$$\text{Precision } \% = \frac{TP}{TP+FP} \text{ x } 100\% \tag{2}$$

$$\text{Precision} = \frac{386}{386+41} \text{ x } 100\%$$

$$\text{Precision} = 90\%$$

Evident in observations on Learning Outcomes, this precision in ripe, high-precision recognition showed how many of the tomatoes that tested positive were ripe.

3) Recall
In this study, recall was presented in the form of mature opportunities that were predicted carefully by the system. The calculation was shown in Equation 3[20].

$$\text{Recall } \% = \frac{TP}{TP+FN} \text{ x } 100\% \tag{3}$$

$$\text{Recall} = \frac{386}{386+33} \text{ x } 100\%$$

$$\text{Recall} = 92\%$$

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

The recall classification report showed the system a mature chance that it predicted was 92%. Evident in observations on Learning Outcomes, high recall ensures that many ripe tomatoes could be identified by the model.

4) F1 – Score

To calculate F1 – Score, the calculation was shown in Equation 4[19].

$$F1 - Score\ \% = 2 \text{ x } \frac{\text{Precision . Recall}}{\text{Precison+Recall}} \text{ x } 100\% \tag{4}$$

$$F1 - Score = 2 \text{ x } \frac{0.90 \text{ x } 0.92}{0.90+0.92} \text{ x } 100\%$$

$$F1 - Score = 2 \text{ x } \frac{0.828}{1.82} \text{ x } 100\%$$

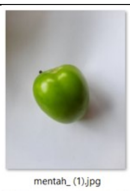
$$F1 - Score = 91\%$$

F1-Score on the performance of ripe tomatoes in classifying data was from both positive and negative classes. A high F1-score indicated that it had a good balance between precision and recall in predicting the class.

3.3 Observations on Learning Outcomes

Observational data was obtained from testing learning outcome programs. The learning results program was tested by manually inputting images with uploads contained in the dataset trial program. Test results based on models were generated by Convolutional Neural Network. The first experiment was the image of unripe tomatoes that had green on all surfaces of tomatoes. The images used in this experiment was 10 images of raw tomatoes.

Table 3. Test Results of Raw Tomatoes

No	Image Upload	Types of Maturity	Machine Learning Classification	Accuracy (%)
1		Raw	Raw	100
2		Raw	Raw	99

p-ISSN: 2656-5935 http://journal-isi.org/index.php/isi e-ISSN: 2656-4882

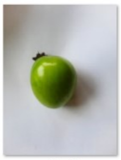
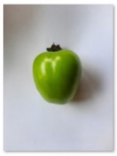
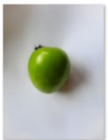
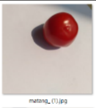
No	Image Upload	Types of Maturity	Machine Learning Classification	Accuracy (%)
3	 mentah_ (3).jpg	Raw	Raw	100
4	 mentah_ (4).jpg	Raw	Raw	100
5	 mentah_ (5).jpg	Raw	Raw	99
6	 mentah_ (6).jpg	Raw	Raw	100
7	 mentah_ (7).jpg	Raw	Raw	100
8	 mentah_ (8).jpg	Raw	Raw	100
9	 mentah_ (9).jpg	Raw	Raw	100
10	 mentah_ (10).jpg	Raw	Raw	100

Table 3 showed that the image size and shooting position affect the accuracy value obtained. Raw images had the least accuracy at 99%. Raw images with size had the highest accuracy value of 100%. Because raw images had only one color.

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

The second experiment was the image of ripe tomatoes that had red on all surfaces. The images used in this experiment was 10 images of ripe tomatoes.

Table 4. Test Results of Ripe Tomatoes

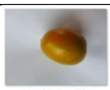
No	Image Upload	Types of Maturity	Machine Learning Classification	Accuracy (%)
1		Ripe	Ripe	99
2		Ripe	Ripe	99
3		Ripe	Ripe	99
4		Ripe	Ripe	99
5		Ripe	Ripe	99
6		Ripe	Ripe	99
7		Ripe	Ripe	99
8		Ripe	Ripe	99

Journal of Information Systems and Informatics
Vol. 5, No. 4, December 2023
p-ISSN: 2656-5935 http://journal-isi.org/index.php/isi e-ISSN: 2656-4882

No	Image Upload	Types of Maturity	Machine Learning Classification	Accuracy (%)
9	 matang_(9).jpg	Ripe	Ripe	99
10	 matang_(10).jpg	Ripe	Ripe	99

Table 4 showed that image size and shooting position affect the accuracy value obtained. Mature imagery had an accuracy of 99%. A mature image that had stable accuracy because it only had one color, namely red. The third experiment was the image of a half-ripe tomato that had a slight green color with a slight pink tint located on the surface of the tomato. The images used in this experiment was 10 images of half-ripe tomatoes.

Table 5. Test Results of Half-Ripe Tomatoes

No	Image Upload	Types of Maturity	Machine Learning Classification	Accuracy (%)
1	 setengah matang_(1).jpg	Half-ripe	Half-ripe	99
2	 setengah matang_(2).jpg	Half-ripe	Half-ripe	100
3	 setengah matang_(3).jpg	Half-ripe	Half-ripe	99
4	 setengah matang_(4).jpg	Half-ripe	Half-ripe	99
5	 setengah matang_(5).jpg	Half-ripe	Half-ripe	99

Journal of Information Systems and Informatics
Vol. 5, No. 4, December 2023

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

No	Image Upload	Types of Maturity	Machine Learning Classification	Accuracy (%)
6	 setengah matang_ (6).jpg	Half-ripe	Half-ripe	99
7	 setengah matang_ (7).jpg	Half-ripe	Half-ripe	99
8	 setengah matang_ (8).jpg	Half-ripe	Half-ripe	99
9	 setengah matang_ (9).jpg	Half-ripe	Half-ripe	99
10	 setengah matang_ (10).jpg	Half-ripe	Half-ripe	99

Table 5 showed that shooting with different lighting conditions affects the accuracy value obtained. Half-baked images had the least accuracy at 99% by size. Half-baked images with size had the highest accuracy value of 100%. The accuracy from three experiments was closed to 90% for different color of tomatoes such as ripe tomatoes, raw tomatoes, and half-ripe tomatoes. This proved that system was able to detect the ripeness of tomatoes from smartphone camera.

4. CONCLUSION

The classification of tomato ripeness using the Convolutional Neural Network model had been done successfully. The images of tomatoes were taken from smartphone camera in outdoor environment with sufficient lightning from the sun between 13.00-14.00. Distance between tomatoes and smartphone camera was 20 cm. Vivo 21S smartphone was used in this research to take the images of tomatoes and it was equipped with 50 MP camera. 1148 images were collected and stored into folder in the computer. In the testing model, 10 tomatoes images were used

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

for each experiment such as raw, half-ripe and ripe were. The testing model was classified successfully and the accuracy was closed to 90%. For future research, other method can be implemented to classify tomato ripeness such as multi-SVM and Learning Vector Quantization (LVQ) algorithm.

REFERENCES

[1] J. Pardede, B. Sitohang, S. Akbar, and M. L. Khodra, "Implementation of Transfer Learning Using VGG16 on Fruit Ripeness Detection," *International Journal of Intelligent Systems and Applications*, vol. 13, no. 2, pp. 52–61, Apr. 2021, doi: 10.5815/ijisa.2021.02.04.

[2] Sanjaya Suwanto, "Penerapan Learning Vector Quantization Pada Pengelompokan Tingkat Kematangan Buah Tomat Berdasarkan Warna Buah," *Jurnal CoreIT*, vol. 5, pp. 49–55, Dec. 2019.

[3] T. Muhamad Hafiez *et al.*, "Optimasi Klasifikasi Gambar Varietas Jenis Tomat Dengan Data Augmentation dan Convolutional Neural Network," *Smart Comp*, vol. 11, no. 2, pp. 175–186, 2022.

[4] R. Soekarta, N. Nurdjan, and A. Syah, "Klasifikasi Penyakit Tanaman Tomat Menggunakan Metode Convolutional Neural Network (CNN)," *Insect (Informatics and Security): Jurnal Teknik Informatika*, vol. 8, no. 2, pp. 143–151, Mar. 2023, doi: 10.33506/insect.v8i2.2356.

[5] P. Das and J. P. S. Yadav, "Automated Tomato Maturity Grading System using CNN," in *2020 International Conference on Smart Electronics and Communication (ICOSEC)*, IEEE, Sep. 2020, pp. 136–142. doi: 10.1109/ICOSEC49089.2020.9215451.

[6] S. T. E. Putri and A. Fahrurozi, "Pendeteksian Objek Pada Citra Hewan Karnivora Dan Herbivora Menggunakan Faster R-CNN," *Jurnal Ilmiah Informatika Komputer*, vol. 27, no. 1, pp. 32–42, Apr. 2022, doi: 10.35760/ik.2022.v27i1.5858.

[7] A. Helsaputra, A. Luhur Prasasti, and R. R. Septiawan, "Implementasi Deep Learning Untuk Prediksi Tingkat Kematangan Dan Bobot Buah Pepaya," *e-Proceeding of Engineering*, vol. 8, no. 6, p. 11993, 2021.

[8] L. Zu, Y. Zhao, J. Liu, F. Su, Y. Zhang, and P. Liu, "Detection and Segmentation of Mature Green Tomatoes Based on Mask R-CNN with Automatic Image Acquisition Approach," *Sensors*, vol. 21, no. 23, p. 7842, Nov. 2021, doi: 10.3390/s21237842.

[9] R. E. Saragih and A. W. R. Emanuel, "Banana Ripeness Classification Based on Deep Learning using Convolutional Neural Network," in *2021 3rd East Indonesia Conference on Computer and Information Technology (EIConCIT)*, IEEE, Apr. 2021, pp. 85–89. doi: 10.1109/EIConCIT50028.2021.9431928.

[10] C. R. Kotta, D. Paseru, M. Sumampouw, T. Informatika, U. Katolik De La Salle Manado, and K. I. Kombos Manado -, "Implementasi Metode

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

Convolutional Neural Network untuk Mendeteksi Penyakit pada Citra Daun Tomat," 2022.

[11] B. Yanto *et al.*, "Klasifikasi Tekstur Kematangan Buah Jeruk Manis Berdasarkan Tingkat Kecerahan Warna dengan Metode Deep Learning Convolutional Neural Network," *JURNAL INOVTEK POLBENG - SERI INFORMATIKA*, vol. 6, no. 2, pp. 259–268, 2021.

[12] J. Christian and S. I. Al Idrus, "Introduction to Citrus Fruit Ripens Using the Deep Learning Convolutional Neural Network (CNN) Learning Method," *Asian Journal of Applied Education (AJAE)*, vol. 2, no. 3, pp. 459–470, Jul. 2023, doi: 10.55927/ajae.v2i3.5003.

[13] Yuhandri, Musli Yanto, and Eka Naufaldi Novri, "Application of Object Mask Detection Using the Convolution Neural Network (CNN)," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 7, no. 4, pp. 922–929, Aug. 2023, doi: 10.29207/resti.v7i4.5059.

[14] A. TiaraSari and E. Haryatmi, "Penerapan Convolutional Neural Network Deep Learning dalam Pendeteksian Citra Biji Jagung Kering," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 5, no. 2, pp. 265–271, Apr. 2021, doi: 10.29207/resti.v5i2.3040.

[15] N. Sudiati and T. Informasi, "Metode Convolutional Neural Network (CNN) untuk deteksi jenis rempah-rempah," *JURNAL CYBER AREA*, vol. 3, no. 2, p. 1, 2023.

[16] I. Nihayatul Husna, M. Ulum, A. Kurniawan Saputro, D. Tri Laksono, and D. Neipa Purnamasari, "Rancang Bangun Sistem Deteksi Dan Perhitungan Jumlah Orang Menggunakan Metode Convolutional Neural Network (CNN)," *Seminar Nasional Fortei Regional*, vol. 7, no. Vol. 5 No. 1 (2022): SinarFe7-5 2022, 2022.

[17] N. Astrianda, "Klasifikasi Kematangan Buah Tomat Dengan Variasi Model Warna Menggunakan Support Vector Machine," *VOCATECH: Vocational Education and Technology Journal*, vol. 1, no. 2, pp. 45–52, Apr. 2020, doi: 10.38038/vocatech.v1i2.27.

[18] U. Ungkawa and G. Al Hakim, "Klasifikasi Warna pada Kematangan Buah Kopi Kuning menggunakan Metode CNN Inception V3," *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, vol. 11, no. 3, p. 731, Jul. 2023, doi: 10.26760/elkomika.v11i3.731.

[19] T. H. Mohamad, Y. Chen, Z. Chaudhry, and C. Nataraj, "Gear Fault Detection Using Recurrence Quantification Analysis and Support Vector Machine," *Journal of Software Engineering and Applications*, vol. 11, no. 05, pp. 181–203, 2018, doi: 10.4236/jsea.2018.115012.

[20] Syaiful Imron, E. I. Setiawan, Joan Santoso, and Mauridhi Hery Purnomo, "Aspect Based Sentiment Analysis Marketplace Product Reviews Using BERT, LSTM, and CNN," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 7, no. 3, pp. 586–591, Jun. 2023, doi: 10.29207/resti.v7i3.4751.