Available online at website: https://jurnal.iaii.or.id/index.php/RESTI

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 9 No. 4 (2025) 955 - 962 e-ISSN: 2580-0760

A New Triple-Weighted K-Nearest Neighbor Algorithm for Tomato Maturity Classification

Lidya Ningsih[1]*, Arif Mudi Priyatno[2], Addini Yusmar[3]
[1]Faculty of Informatic, Telkom University, Bandung, Indonesia
[2]Department of Digital Business, Faculty of Economics and Business, Universitas Pahlawan Tuanku Tambusai, Kampar, Indonesia
[3]Fakulty of Islamic Economics and Business, Mahmud Yunus State Islamic University of Batusangkar, Tanah Datar, Indonesia

[1]telulidyaningsih@telkomuniversity.ac.id, [2]arifmudi@universitaspahlawan.ac.id, [3]addiniy20@gmail.com

Abstract

As climatic products, tomatoes are highly sensitive to harvesting and processing. The sorting of tomatoes can be significantly improved by utilizing Hue Saturation Value (HSV) color features that are classified using neighboring algorithms, such as K-Nearest Neighbor (KNN), Weighted K-Nearest Neighbor (W-KNN), and DW-KNN. However, the DW-KNN algorithm does not consider the relative relationship between the farthest, nearest, and surrounding neighbors, which may impact the classification accuracy, particularly in datasets with uneven distributions. This study proposes a Triple Weighted K-Nearest Neighbor (TW-KNN) algorithm for tomato image classification. This algorithm effectively handles the problem of sensitivity and outliers in the data distribution and considers the relationship between neighboring distances. The classification data consisted of 400 tomato images with five maturity levels divided into training and testing sets using k-fold cross-validation. Tests were conducted using several variations of parameter k, namely 4, 6, 9, and 15, to evaluate the classification performance. The results show that the proposed TW-KNN algorithm consistently outperforms other methods by producing better classification results. This is demonstrated by an accuracy rate of 95.52% across different values of k. The superior performance of the TW-KNN highlights its ability to provide robust and stable classification results compared to conventional KNN variants. This finding indicates that the TW-KNN is more effective in consistently classifying tomato fruits, making it a promising approach for automated fruit sorting applications.

Keywords: DW-KNN; HSV; KNN; TW-KNN; W-KNN

How to Cite: L. N. Lidya, A. M. P. Arif, and A. Y. dini, "A New Triple-Weighted K-Nearest Neighbor Algorithm for Tomato Maturity Classification", *J. RESTI (Rekayasa Sist. Teknol. Inf.)* , vol. 9, no. 4, pp. 955 - 962, Aug. 2025.
Permalink/DOI: https://doi.org/10.29207/resti.v9i4.6441

Received: March 4, 2025
Accepted: July 23, 2025
Available Online: August 29, 2025

This is an open-access article under the CC BY 4.0 License
Published by Ikatan Ahli Informatika Indonesia

1. Introduction

Tomatoes are the most widely consumed horticultural commodity and the fourth most desirable commercial vegetable [1], [2]. In Indonesia, tomato production has increased from year to year. According to the Badan Pusat Statistik (BPS), in 2018 tomato production was 976,772 tons, then increased to 1,020,331 tons in 2019, in 2020 production jumped to 1,084,993 tons, and again increased to 1,107,575 tons in 2021[3]. Because the demand for tomatoes is very high, so the process of sorting tomatoes is very important to maintain the quality of tomatoes in accordance with consumer demand standards [4]. However, the assessment of tomato maturity is still performed manually, which reduces accuracy due to the subjectivity of human evaluation [5], [6].

The level of maturity of tomato fruit is influenced by color, size, and shape factors, with color being the main factor in assessing fruit maturity [4]. In addition, color also affects consumer preferences when buying tomatoes [7]. Along with advances in information technology, computers can identify fruit based on color characteristics. Classification of tomato fruit maturity levels produces varying levels of accuracy, depending on the model and algorithm used [8], [9]. According to the research conducted by [10], the classification of tomato maturity levels using the HSV algorithm and the KNN algorithm has an accuracy of 91.25%. The use of the HSV model stems from its natural geometric

representation [10]. The HSV model has a weakness in the RGB color space transformation process. The transformation process results in a loss of information, so the results obtained are less effective. This research proposes increasing information in the HSV color space by giving weight to each color channel in the HSV model. The color channel in the HSV model is multiplied by α (alfa), β (beta), and **γ** (gamma) [11].

The KNN algorithm is a simple, effective, intuitive, and competitive classification algorithm. However, this algorithm treats the data distribution among all neighbors equally, even though some neighbors may have higher relevance, resulting in more significant information not being fully utilized. This makes the KNN algorithm dependent on the amount of training data. In addition, the KNN algorithm relies on calculating distances between data points and considers unnecessary dataset features [12]. On the other hand, this algorithm is capable of adapting to several modifications in different algorithmic aspects, such as parameter optimization, enhancement of distance calculations, weighting the nearest data points, and pruning the training dataset [12]. Dudani proposed the W-KNN algorithm to address the sensitivity issue in the KNN algorithm. The technique used in the W-KNN algorithm is to assign a large weight to the nearest neighbors and a small weight to the farthest neighbors according to the distance from the data distribution [13]. W-KNN has a lower sensitivity level than KNN, but the robustness of W-KNN is still affected by the presence of outliers, especially in small sample sizes [14]. The research conducted by [15] demonstrates this through experiments on five datasets, where the W-KNN algorithm outperformed the KNN algorithm in three datasets regarding classification accuracy, with accuracy percentages of 73.20%, 75%, and 88.90%.Due to sensitivity problems with the W-KNN algorithm, research [13] created a new method called DW-KNN, which uses a double-weight approach for KNN. The weight of each nearest neighbor in the W-KNN rule is replaced with a more appropriate double weight. This method reduces the weight on all nearest neighbors, except for the first and farthest neighbors k. This technique is more optimal, especially on datasets with imbalanced class distributions and small sample sizes [16]. In the study [16], the DW-KNN algorithm significantly outperformed the KNN method, achieving an accuracy rate of 99.01%. However, DW-KNN does not consider the relative relationships of the farthest, nearest, and surrounding neighbors.Based on the described problem, this research proposes the TW-KNN algorithm for the classification of tomato fruit images. This algorithm addresses sensitivity issues, outliers in data distribution, and considers the relationship between neighbor distances.

2. Methods

The classification process for the maturity level of tomato fruit can be seen in Figure 1 by utilizing the HSV color feature, where each color channel is weighted to minimize loss of information during the transformation from RGB to HSV color space. Furthermore, the classification process uses KNN, W-KNN, DW-KNN, and TW-KNN algorithms to analyze the effect of weighting on each algorithm and determine the best algorithm based on the accuracy level of the obtained tomato fruit maturity classification. The classification process for the maturity level of tomatoes consists of training and testing to see the level of learning and the level of success of the model used.



Figure 1. Tomato Fruit Image

Before the classification process is carried out, the tomato image first goes through a preprocessing stage. This stage eliminates irrelevant information, such as the background, as shown in Figure 2. This process focuses the analysis only on the object of research.

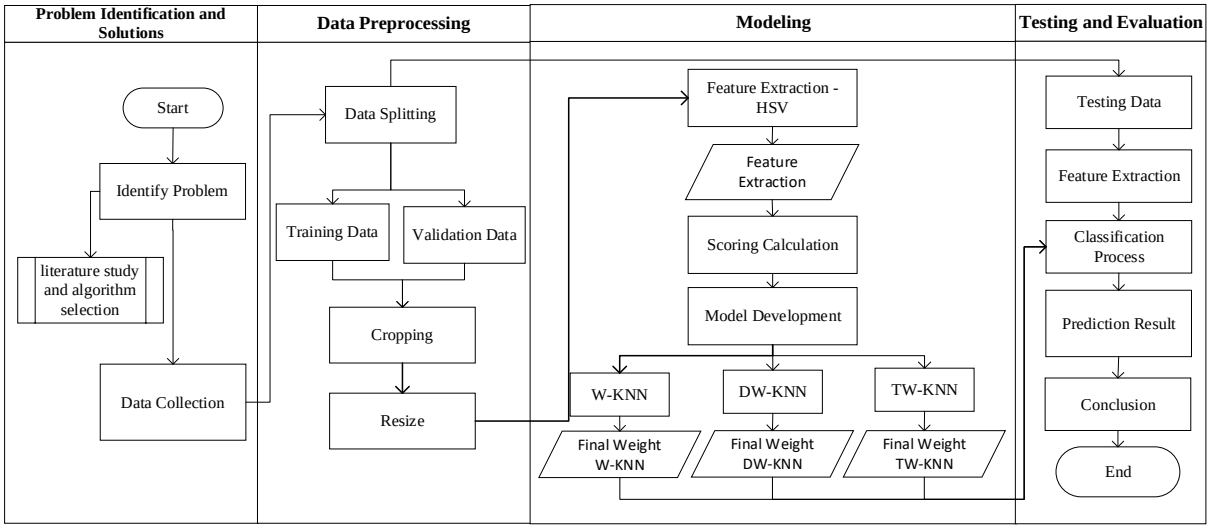


Figure 1 Research Stages

Next, a resize process is carried out to change the image pixel size according to the research's desired size. The pixel size used in this study is 400x400 pixels, as shown in Figure 3.

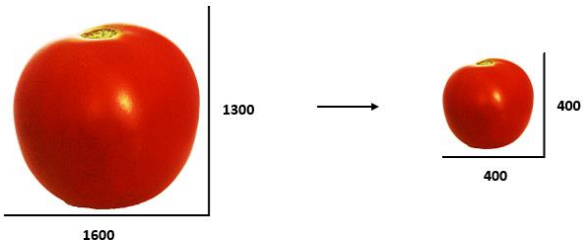


Figure 2. Resize Process

2.1 Image Data

This study classifies tomatoes' maturity level, as shown in Figure 4.

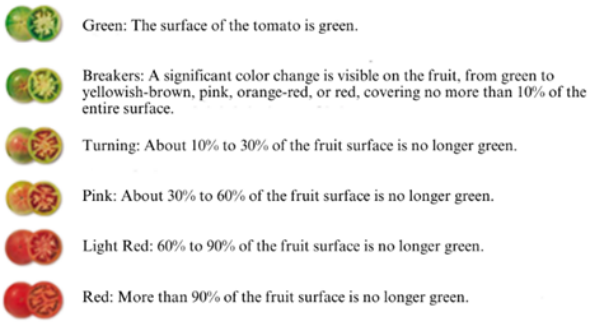


Figure 3. Maturity of Tomato Fruit

The maturity level of the breakers in Figure 4 and Figure 5 is combined with the green class because the breakers class is predominantly dark green, and only 10% contain brownish yellow, pink, or red on the entire surface [3].

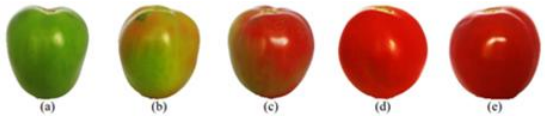


Figure 4. Maturity of Tomato Fruit (a) green, (b) turning, (c) pink, (d) light red, dan (e) red [12]

The data for this study came from a study of the maturity level of tomatoes [10]. The study utilized plum tomatoes and acquired images using a 24.3-megapixel DSLR camera. The resulting image was saved in PNG format.

Tomato image data collection uses a white background, with the image object in the middle. This study used 400 tomato images from the initial 100 tomatoes listed in Table 1. Each tomato produced four different types of images.

Table 1. Tomato Image Data Distribution

No	Attribute	Number of Images
1	Green	20
2	Turning	20
3	Pink	20
4	Light Red	20
5	Red	20

2.2 Hue Saturation Value (HSV)

The HSV algorithm is more intuitive than RGB. This is because the HSV color space component is in the range of angles 0^0 to 360^0 [10]. The HSV color space can be converted using Equations 1, 2, and 3

$$H = arctan\left\{\frac{\sqrt{3}(G-B)}{(R-G)+(R-B)}\right\} \quad (1)$$

$$S = 1 - \frac{\min(R,G,B)}{V} \quad (2)$$

$$V = \frac{R+G+B}{3} \quad (3)$$

However, if the saturation is 0, it is impossible to represent the hue warrant's channel. It is necessary to do the RGB normalization process according to Equations 4, 5, and 6 [10].

$$r = \frac{R}{R+G+B} \quad (4)$$

$$g = \frac{G}{R+G+B} \quad (5)$$

$$b = \frac{B}{R+G+B} \quad (6)$$

After the RGB normalization process, the RGB to HSV conversion process is carried out using Equations 7, 8, 9 and 10.

$$v = \max(r, g, b) \quad (7)$$

$$s = \begin{cases} 0 \\ v \end{cases} \quad (8)$$

$$H = \begin{cases} 0 & Jika\ s = 0 \\ 60 \times (-g - b) & jika\ v = r \\ 60 \times \left[2 + \frac{b-r}{s \times v}\right] & jika\ v = g \\ 60 \times \left[4 + \frac{r-g}{s \times v}\right] & jika\ v = b \end{cases} \quad (9)$$

$$H = H + 360 \qquad Jika\ H < 0 \quad (10)$$

The RGB color space has values (0,5, 0,5, and 0,25) while after the conversion process to the HSV color space, it become (30^0, √3/4, and 0,5) [17]. The result of converting the RGB color space to HSV has a loss of information, so the results obtained are less effective [11].

This study proposes the HSV method by applying weighting using the parameters alpha (α), beta (β), and gamma ($\boldsymbol{\gamma}$). This process involves multiplying the channel hue by the parameter α, saturation by the parameter β, and value by the parameter $\boldsymbol{\gamma}$, where the total weight of α, β, and δ is one. This approach minimizes information loss when converting RGB color features to HSV. This merging process becomes the output of the HSV color space for the classification process.

2.3 K-Nearest Neighbour (KNN)

The KNN algorithm is a supervised learning algorithm that can be used for classification and regression problems. This algorithm applies the principle of proximity based on the k-nearest neighbors [18]. The

KNN algorithm is known for its simplicity, ease of understanding, and ability to handle data noise. However, KNN has limitations, such as slow execution time and sensitivity to large dataset sizes [19]. Due to these limitations, modifications have been made to improve the performance of the KNN algorithm [19].

Several modifications to the KNN algorithm have been developed to overcome its limitations. One example is class contribution and feature weighting (DCT-KNN), which integrates class contribution and weighting, but this algorithm requires complex computational steps, resulting in long processing times [19].

2.4 *Weighted K-Nearest Neighbor (WKNN)*

The W-KNN algorithm was first introduced by Dudani as a development of the KNN algorithm by applying the weighted voting rule. This development was carried out by giving greater weight to closer neighbors and smaller weight to more distant neighbors using the distance-weighted function. In this algorithm, each member in the set of k closest neighbors is weighted based on the distance function to the object. The weight w_i for the closest neighbor i of the query x' is defined according to the following Equation 11 [13].

$$w_j = \begin{cases} \frac{d_k^{NN} - d_j^{NN}}{d_k^{NN} - d_1^{NN}}, & d_k^{NN} \neq d_1^{NN} \\ 1, & d_k^{NN} = d_1^{NN} \end{cases} \quad (11)$$

The classification result is then determined by the class with the highest total weight among the closest neighbors, which is calculated using majority weighted voting in Equation 12.

$$C_{jmax} = \underset{c_j}{\arg mac} \sum_{x_i^{NN} \in T'} w_i \times I(C = C_i^{NN}) \quad (12)$$

Thus, the classification process is more influenced by neighbors closer to the query, not just considering the total number of neighbors.

Based on Equation $w_j = \frac{d_k^{NN} - d_j^{NN}}{d_k^{NN} - d_1^{NN}}, \quad d_k^{NN} \neq d_1^{NN}$ 1, $d_k^{NN} = d1NN$ (11, the closest neighbor is given a weight of 1, the farthest neighbor is given a weight of 0, and the weight for other neighbors is determined by linear scaling based on the interval between them.

2.5 Dual Weighted K-Nearest Neighbour (DW-KNN)

The W-KNN algorithm performs better than other algorithms that use the *weighted voting* technique. However, this algorithm still faces limitations in handling sensitivity when selecting neighbor k values. This is due to *outliers*, which become more significant when the training sample size is relatively small. In addition, both the KNN and W-KNN algorithms show weaknesses in sensitivity to the selection of k values in conditions of uneven class distribution in the data set [13].

The DW-KNN algorithm designs a simple and effective classification to reduce sensitivity issues when selecting neighbors k. The DW-KNN algorithm utilizes distance-weighted and uniform-weighted voting functions, assigning varying weights to the nearest k neighbors based on distance and ranking. DW-KNN uses Equation 13 [13].

$$w_j = \begin{cases} \frac{d_k^{NN} - d_j^{NN}}{d_k^{NN} - d_1^{NN}} \times \frac{d_k^{NN} + d_1^{NN}}{d_k^{NN} + d_j^{NN}} & d_k^{NN} \neq d_1^{NN} \\ 1, & d_k^{NN} = d_1^{NN} \end{cases} \quad (13)$$

The classification results can be determined in the same way as the W-KNN algorithm (Equation $C_{jmax} = \underset{c_j}{\arg mac} \sum_{x_i^{NN} \in T'} w_i \times I(C = C_i^{NN})$ (12).

The DW-KNN algorithm uses a dual-weighting technique for each nearest neighbor consisting of two parts (Equation 13). The first part adopts the same weighting technique as the W-KNN algorithm, and the second part is a new weight specifically developed by the DW-KNN. Based on research by Gou (2019) these two scales are designed using a distance-based weighting scheme [13]. Although the DW-KNN algorithm is designed to handle sensitivity and *outlier issues* in data, it does not consider the relative relationship between neighbors.

2.6 Triple Weighted K-Nearest Neighbour (TW-KNN)

Although the DW-KNN algorithm is designed to overcome the sensitivity problem in the k parameter, it still requires determining the optimal k value, especially in data sets that are highly complex. In addition, it still has the problem of outliers against the weight of the nearest neighbors. DW-KNN performance also depends on data distribution. Extremely unbalanced class conditions can cause the conditional probability estimation of the class to be less accurate.

The proposed method uses a third weight to overcome the impact of other relevant variables, such as the relationship between neighbor distance and neighbor influence, to improve classification accuracy. This third weight is designed to strengthen the contribution of truly representative neighbors in the prediction process. In addition, by introducing a third weight, the TW-KNN algorithm can increase resistance to outliers, especially when the outliers are close to the query and cannot be eliminated by the previous two weights.

$$w_j = \begin{cases} \frac{d_k^{NN} - d_j^{NN}}{d_k^{NN} - d_1^{NN}} \times \frac{d_k^{NN} + d_1^{NN}}{d_k^{NN} + d_j^{NN}} \times \frac{d_k^{NN} \times d_j^{NN}}{d_k^{NN} + d_1^{NN}} & d_k^{NN} \neq d_1^{NN} \\ 1, & d_k^{NN} = d_1^{NN} \end{cases} \quad (14)$$

Add a third weight (Equation 14) to consider the relationship between the distance of the farthest neighbor d_k and neighbor j (d_j), and relate it to the distance of the nearest neighbor d_1.

Due to the addition of these three weights, this algorithm explicitly considers the relationship between neighbors through a calculation involving the product between the distance of the farthest neighbor and the *j* neighbor, then divided by the sum of the distances between the farthest and closest neighbors. This

approach adds a dimension between neighbors in the weighting, allowing a combination of elements from the closest and furthest neighbors to adjust the contribution of each neighbor to the classification result.

This third weighting also plays a role in perfecting the neighbor effect by considering their relative relationships, both to the closest and furthest neighbors. Thus, the weight given does not only depend on a single distance but also on a combination of distances that reflect the distribution and relationships between neighbors in the feature space. This approach allows the algorithm to be more adaptive in capturing complex classification patterns by taking into account the local structure of the data more holistically, reflecting the distribution and interrelationships between neighbors in the feature space.

This approach allows the algorithm to be more adaptive in capturing complex classification patterns by thoroughly considering the data's local structure.

3. Results and Discussions

The classification of tomatoes' maturity level uses several classification algorithms, including KNN, WK-NN, DW-KNN, and TW-KNN, with experimental testing carried out, as shown in Table 2.

Table 2. Testing Parameters

No.	Eksperimental Testing	Description
1.	Dividing training data and test data	The research used 400 tomato image datasets, divided into learning processes and algorithm testing. The division of training data in this study used K-fold cross-validation (1, 3, 5).
2.	Parameter K	The parameter is used to determine the distance of neighbors in the KNN algorithm. The value of the k parameter used in this study was 4, 6, 9, and 15.

The classification of the maturity level of tomato fruit is based on the HSV color space according to Table 3. The HSV color feature is weighted by multiplying each HSV color *channel* with *alfa* (α), *beta* (β), and *gamma* ($\boldsymbol{\gamma}$). The values of *alfa* (α) used are 0.4, *beta* (β) 0,4, and *gamma* ($\boldsymbol{\gamma}$) 0,2 to obtain the results of the HSV color channel feature as shown in Table 4.

Table 3. HSV Color Feature

No	Hue	Saturation	Value	Label
0	65.9150447	0.6423492	62.0744951	*Green*
80	9.30616341	0.7141841	89.9297132	*Light*
169	28.5177570	0.6212918	80.1957157	*Pink*
240	8.4573603	0.7014836	64.8743480	*Red*
320	39.1796522	0.6587762	69.2718775	*Turning*

Table 4. HSV feature performed Weighting

No	Hue	Saturation	Value	Label
0	0.3493425	0.1417481	0.0467967	Green
80	0.0631366	0.2667544	0.1734055	Light
160	0.1602675	0.1051040	0.1291621	Pink
240	0.0588451	0.2446531	0.0595228	Red
320	0.2141724	0.1703341	0.0795106	*Turning*

The results of the classification of the proposed method in this study will be compared with the KNN, W-KNN, and DW-KNN algorithms to see the best performance based on the algorithm's success rate in classifying tomato images. The success rate was obtained from the accuracy of each algorithm. After obtaining the results of HSV feature extraction, the next step is to test using tomato image data with KNN, W-KNN, DW-KNN, and TW-KNN algorithms. The results of the KNN algorithm tests are listed in Table 5.

Table 5.KNN Algorithm Test Results

K-Fold	K	Accuracy	Recall	Precision
1	4	91,04%	91,04%	90,83%
	6	92,54%	92,63%	93,38%
	9	86,57%	86,57%	87,80%
	15	88,06%	88,06%	88,45%
3	4	89,55%	89,55%	89,16%
	6	83,58%	83,58%	84,08%
	9	85,07%	85,07%	84,82%
	15	85,07%	85,07%	86,89%
5	4	80,30%	80,30%	82,22%
	6	87,88%	87,88%	88,60%
	9	89,40%	89,40%	89,81%
	15	89,39%	89,39%	90,16%

Table 5 shows the results of testing the KNN algorithm with the highest accuracy of 92.54%. However, the KNN algorithm achieves a significant accuracy of over 80%.

Table 6. W-KNN Algorithm Test Results

k-fold	K	Accuracy	Recall	Precision
1	4	85,07%	85,07%	86,70%
	6	89,55%	89,55%	89,95%
	9	91,04%	91,04%	91,14%
	15	89,55%	89,55%	89,20%
3	4	88,06%	88,06%	89,91%
	6	83,58%	83,58%	85,24%
	9	88,06%	88,06%	87,80%
	15	92,54%	92,54%	93,80%
5	4	89,40%	89,40%	89,90%
	6	84,85%	84,85%	85,29%
	9	83,33%	83,33%	84,90%
	15	84,85%	84,85%	86,06%

Table 6 shows the accuracy of the W-KNN algorithm. The algorithm achieved the highest accuracy result of 92.54%.

Table 7. DW-KNN Algorithm Test Results

k-fold	K	Accuracy	Recall	Precision
1	4	89,55%	89,55%	90,83%
	6	89,55%	89,55%	90,38%
	9	91,04%	91,04%	88,03%
	15	86,57%	86,57%	86,41%
3	4	88,06%	88,06%	88,93%
	6	86,57%	86,57%	86,19%
	9	83,58%	83,58%	84,53%
	15	85,07%	85,07%	84,93%
5	4	80,30%	80,30%	79,94%
	6	86,36%	86,36%	86,25%
	9	89,39%	89,39%	91,10%
	15	84,85%	84,85%	84,45%

Table 7 shows the test result of the DW-KNN algorithm. The algorithm's highest accuracy is 91.04%.

According to Table 8, the proposed method obtains better test results than the KNN, W-KNN, and DW-

KNN algorithms. The TW-KNN algorithm obtains the highest accuracy result, 95.52%. This study's test results use a *confusion matrix* to calculate each algorithm's accuracy, precision, and recall.

Table 8. TW-KNN Algorithm Test Results

k-fold	k	Accuracy	Recall	Precision
1	4	95,52%	95,52%	95,79%
	6	91,04%	91,04%	93,49%
	9	89,55%	89,55%	90,56%
	15	94,03%	94,03%	94,34%
3	4	83,58%	83,58%	86,87%
	6	91,04%	91,04%	91,39%
	9	86,57%	86,57%	87,37%
	15	85,07%	85,07%	85,74%
5	4	87,88%	87,88%	90,72%
	6	89,39%	89,39%	90,96%
	9	83,33%	83,33%	86,92%
	15	89,39%	89,39%	89,61%

The graph in Figure 5 compares the accuracy results of KNN and TW-KNN, where TW-KNN is more stable and tends to have higher accuracy than KNN. KNN shows more significant fluctuations, while TW-KNN is more consistent in maintaining accuracy, making it superior for data classification that requires high stability and accuracy.

Figure 6 shows a comparison chart of the W-KNN and TW-KNN algorithm test results. The TW-KNN algorithm is more stable in classifying tomato images, while the W-KNN algorithm experiences significant fluctuations. Although W-KNN has several points with higher accuracy, its instability may indicate the presence of *outliers that* affect the model's performance. TW-KNN excels at maintaining a balance between accuracy and stability, making it a more reliable choice for classification with more consistent results.

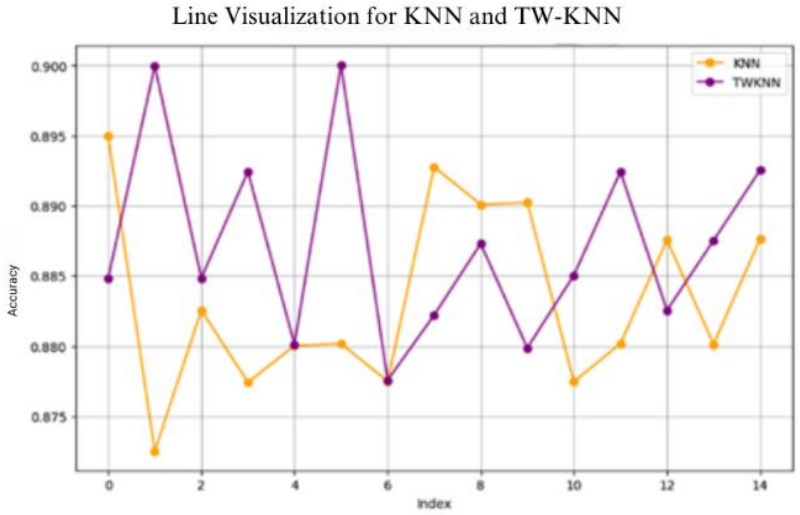


Figure 5. Comparison of KNN and TW-KNN Algorithms

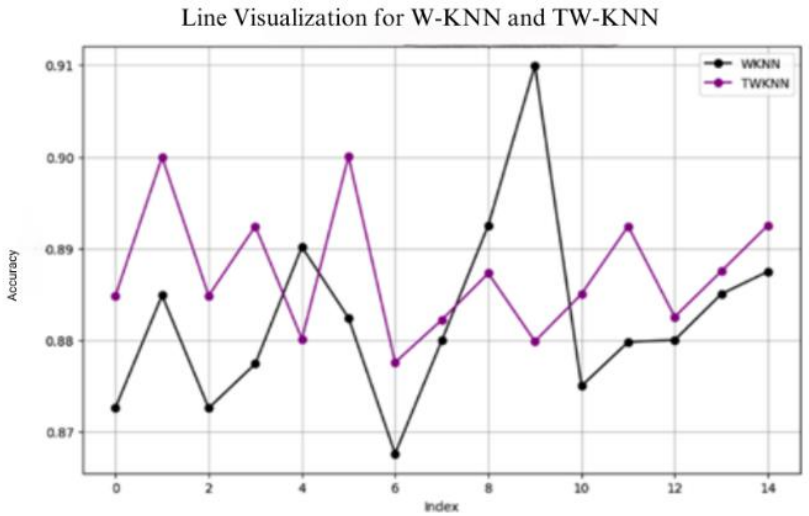


Figure 6. Comparison of W-KNN and TW-KNN Algorithms

Figure 7 compares the accuracy results of the DW-KNN and TW-KNN algorithms for various values of K. Overall, the TW-KNN algorithm shows superior performance, winning 10 out of 15 K trials. Meanwhile,

the DW-KNN algorithm only managed to obtain higher accuracy in 4 trials, with one balanced condition at K = 7. However, there is an advantage for DW-KNN at K = 9, where its accuracy is higher than TW-KNN. This shows that although TW-KNN is generally more stable, there are certain conditions where the DW-KNN method can produce better performance. These results confirm that the addition of the third weight proposed in TW-KNN has a significant effect on accuracy stability, although DW-KNN can still provide superior results at certain values of K.

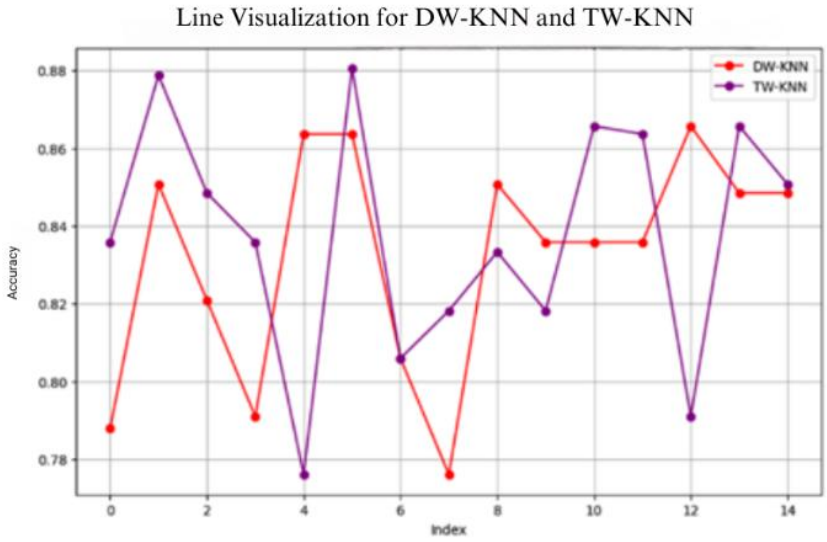


Figure 7. Comparison of DW-KNN and TW-KNN Algorithms

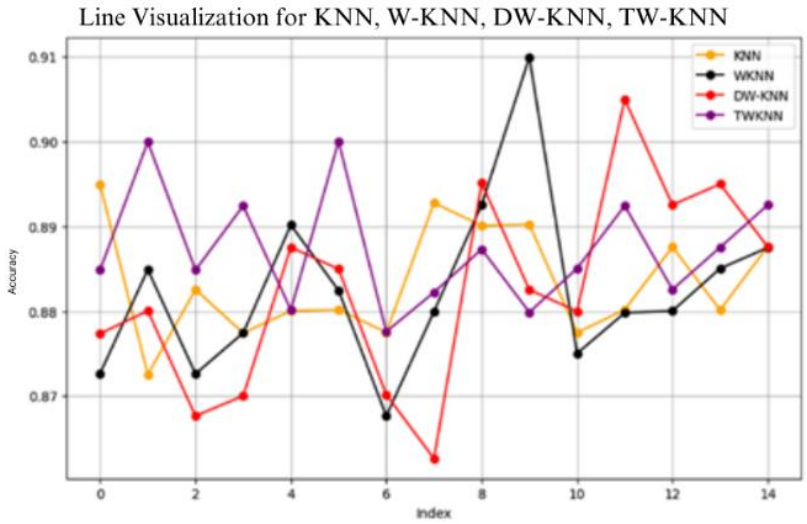


Figure 8. Comparison of KNN, W-KNN, DW-KNN, TW-KNN Algorithms

Figure 8 shows the accuracy of the four algorithms, with the TW-KNN algorithm showing a more stable performance than the other algorithms. Although the DW-KNN and W-KNN algorithms achieve higher accuracy at some points, both experience significant fluctuations. This shows that adding weights to the KNN algorithm yields good results in terms of the sensitivity of the optimal k-value and handling of outliers. The TW-KNN algorithm has a more consistent accuracy pattern that shows resistance to data variation, selection of optimal K values, and minimal influence of outliers.

4. Conclusions

Research the classification of tomato images with five maturity levels: Green, Turning, Pink, Light Red, and Red. The data in this study consists of 400 tomato images extracted using the HSV color feature. Each channel of the color feature is multiplied by the values of alpha, beta, and gamma. The results of the HSV color feature are classified using the KNN, W-KNN, DW-KNN, and TW-KNN algorithms. The tomato image is classified through training and testing based on k-fold cross-validation. The proposed algorithm obtained the best accuracy result, which was 95.52% with k=4 and fold of 1. The accuracy result shows the algorithm's superiority in the classification stability level, and

future work can enhance TW-KNN by incorporating fuzzy weighting for adaptive data processing and a hybrid model to improve accuracy and classification stability.

References

[1] Y. P. Huang, T. H. Wang, and H. Basanta, "Using Fuzzy Mask R-CNN Model to Automatically Identify Tomato Ripeness," *IEEE Access*, vol. 8, pp. 207672–207682, 2020, doi: 10.1109/ACCESS.2020.3038184.

[2] R. G. de Luna, E. P. Dadios, A. A. Bandala, and R. R. P. Vicerra, "Tomato growth stage monitoring for smart farm using deep transfer learning with machine learning-based maturity grading," *Agrivita*, vol. 42, no. 1, pp. 24–36, 2020, doi: 10.17503/agrivita.v42i1.2499.

[3] W. Rahmatika, Wasito, B. S. Wibawa, T. Handayani, and N. Fitriyah, "Potensi Pupuk Kotoran Kelinci dan Pupuk Anorganik pada Tanaman Tomat (Lycopersicum esculentum Mill.)," *Innofarm J. Inov. Pertan.*, vol. 26, no. 1, pp. 61–68, 2024.

[4] H. S. Mputu, A. Abdel-Mawgood, A. Shimada, and M. S. Sayed, "Tomato Quality Classification Based on Transfer Learning Feature Extraction and Machine Learning Algorithm Classifiers," *IEEE Access*, vol. 12, no. December 2023, pp. 8283–8295, 2024, doi: 10.1109/ACCESS.2024.3352745.

[5] Ifmalinda, Andasuryani, and I. Rasinta, "Classification of tomato (Lycoersicon Esculentum Miil) ripeness levels based on HSV color using digital image processing," *IOP Conf. Ser. Earth Environ. Sci.*, vol. 1116, no. 1, 2022, doi: 10.1088/1755-1315/1116/1/012062.

[6] H. Mubarok, S. Murni, and M. M. Santoni, "Penerapan Algoritma K-Nearest Neighbor untuk Klasifikasi Tingkat Kematangan Buah Tomat Berdasarkan Fitur Warna," *Semin. Nas. Mhs. Ilmu Komput. dan Apl. Jakarta-Indonesia*, no. April, pp. 773–782, 2021.

[7] P. Das and J. P. S. Yadav, "Automated Tomato Maturity Grading System using CNN," *Proc. - Int. Conf. Smart Electron. Commun. ICOSEC 2020*, no. Icosec, pp. 136–142, 2020, doi: 10.1109/ICOSEC49089.2020.9215451.

[8] S. A. Mutha, A. M. Shah, and M. Z. Ahmed, "Maturity Detection of Tomatoes Using Deep Learning," *SN Comput. Sci.*, vol. 2, no. 6, pp. 1–7, 2021, doi: 10.1007/s42979-021-00837-9.

[9] N. Astrianda, "Klasifikasi Kematangan Buah Tomat Dengan Variasi Model Warna Menggunakan Support Vector Machine," *VOCATECH Vocat. Educ. Technol. J.*, vol. 1, no. 2, pp. 45–52, 2020, doi: 10.38038/vocatech.v1i2.27.

[10] L. Ningsih and P. Cholidhazia, "Classification Of Tomato Maturity Levels Based on RGB And HSV Colors Using KNN Algorithm," *RIGGS J. Artif. Intell. Digit. Bus.*, vol. 1, no. 1, pp. 25–30, 2022, doi: 10.31004/riggs.v1i1.10.

[11] A. M. Priyatno, "Spammer Detection Based on Account, Tweet, and Community Activity on Twitter," *J. Ilmu Komput. dan Inf.*, vol. 13, no. 2, pp. 97–107, 2020, doi: 10.21609/jiki.v13i2.871.

[12] S. Uddin, I. Haque, H. Lu, M. A. Moni, and E. Gide, "Comparative performance analysis of K-nearest neighbour (KNN) algorithm and its different variants for disease prediction," *Sci. Rep.*, vol. 12, no. 1, pp. 1–11, 2022, doi: 10.1038/s41598-022-10358-x.

[13] P. Tamrakar and S. P. Syed Ibrahim, "Lazy Learning Associative Classification with WkNN and DWkNN Algorithm," *ITM Web Conf.*, vol. 37, p. 01023, 2021, doi: 10.1051/itmconf/20213701023.

[14] G. Chen, X. Guo, K. Liu, X. Li, and J. Yang, "RWKNN: A Modified WKNN Algorithm Specific for the Indoor Localization Problem," *IEEE Sens. J.*, vol. 22, no. 7, pp. 7258–7266, 2022, doi: 10.1109/JSEN.2022.3155902.

[15] F. Tarakci and A. Ozkan, "Comparison of classification performance of kNN and WKNN algorithms," *Selcuk Univ. J. Eng. Sci.*, vol. 20, no. 02, pp. 32–37, 2021, [Online]. Available: http://sujes.selcuk.edu.tr/sujes.

[16] S. Ji, R. Li, S. Shen, B. Li, B. Zhou, and Z. Wang, "Heartbeat Classification Based on Multifeature Combination and Stacking-DWKNN Algorithm," *J. Healthc. Eng.*, vol. 2021, 2021, doi: 10.1155/2021/8811837.

[17] E. K. Hassan and J. H. Saud, "HSV Color Model and Logical Filter for Human Skin Detection," *AIP Conf. Proc.*, vol. 2457, no. February, 2023, doi: 10.1063/5.0120025.

[18] L. Xiong and Y. Yao, "Study on an adaptive thermal comfort model with K-nearest-neighbors (KNN) algorithm," *Build. Environ.*, vol. 202, no. June, p. 108026, 2021, doi: 10.1016/j.buildenv.2021.108026.

[19] A. Almomany, W. R. Ayyad, and A. Jarrah, "Optimized implementation of an improved KNN classification algorithm using Intel FPGA platform: Covid-19 case study," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 6, pp. 3815–3827, 2022, doi: 10.1016/j.jksuci.2022.04.006.