JTIF: Journal of Technology Information
Volume 02 No. 03, November 2025 p 12 – 17

E-ISSN: 3063-0088

CLASSIFICATION OF STAR FRUIT HEALTH LEVEL USING KNN AND SVM ALGORITHMS

Indra Lesmana[1], Iwan Rizal Setiawan[2]

[1]Putra Indonesia University

Email: [1]milesmana2002@gmail.com

Article Info

Article history:
Received Novmber, 05, 2025
Revised November, 20, 2025
Accepted November, 31, 2025

Keywords:
Starfruit,
KNN
Classification,
SVM,
Machine
Learning.

Abstract

Starfruit, scientifically known as Averrhoa Carambola, is a horticultural commodity whose health quality must be maintained to ensure it is suitable for consumption and has a good selling value. Manual starfruit health assessment still relies on human visual observation, potentially leading to subjectivity and inconsistency. Therefore, this study aims to classify starfruit health using machine learning approaches, namely the K-Nearest Neighbor (KNN) algorithm and the Support Vector Machine (SVM).

The results showed that the SVM algorithm provided better classification performance than KNN in determining star fruit health. Therefore, the application of the KNN and SVM algorithms can be an alternative solution for an automatic and objective star fruit health classification system.

Corresponding Author:
Indra Lesmana
Putra Indonesia University

1. INTRODUCTION

Starfruit is a tropical fruit widely cultivated in Indonesia and has significant economic value. The quality and healthiness of starfruit significantly influence consumer appeal and market price. Fruit with physical damage, discoloration, or signs of disease are generally considered unfit for consumption, requiring an accurate sorting process to determine its healthiness.

The development of machine learning technology offers significant opportunities in digital image processing, particularly for classifying objects based on specific characteristics. The K-Nearest Neighbor (KNN) and Support Vector Machine (SVM) algorithms are two widely used classification methods due to their simplicity and high accuracy. KNN classifies based on the proximity of data points, while SVM works by finding the optimal hyperplane to maximize data class separation.

Binna et al. (2025) conducted a study on tomato fruit classification using the KNN and SVM algorithms. This study showed that SVM produced a higher accuracy rate than KNN, at 97.22%, while KNN achieved an accuracy of 94.44%. These results confirm that SVM has good ability to separate classes in fruit image data. [1]

The results of this research are expected to be the basis for developing an automatic star fruit health classification system, so that it can help farmers, distributors, and horticultural industry players in increasing the efficiency and quality of the fruit sorting process..

2. METHODS

This study employed an experimental approach, utilizing machine learning techniques and digital image processing to classify star fruit health. The research methodology was systematically designed so that each stage could be replicated and evaluated objectively. In general, the research process consisted of data collection, image preprocessing, feature extraction, data sharing, model training, model testing, and performance evaluation.

DOI: 10.52436/JTIF/ ...

2.1 Research Design

This research was designed as a quantitative study with an experimental approach, where star fruit image data were collected, processed, and analyzed to evaluate the performance of two supervised learning algorithms, namely K-Nearest Neighbor (KNN) and Support Vector Machine (SVM), in classifying fruit health conditions. This quantitative approach emphasizes numerical measurements of model performance through metrics such as classification accuracy obtained by comparing model predictions to actual class labels in the test data. In the context of machine learning, the experimental approach is a standard method for evaluating the effectiveness of algorithms in image classification tasks because it allows for objective numerical verification and comparison of results.

2.2 Dataset and Image Acquisition

The dataset used is in the form of star fruit images obtained through the direct image acquisition process. Image capture is carried out using a high-resolution digital camera. To reduce the influence of external factors, the image-taking process is carried out with the following settings:

a.. The camera's distance to the object is relatively fixed
b. Lighting is even and not excessive
c. Simple background to reduce visual noise

The images obtained were then classified into two classes, namely healthy star fruit and unhealthy star fruit. The labeling process is carried out based on the visual condition of the fruit, such as skin color, the presence of spots, cuts, or surface damage.

2.3.1 Resizing Citra

All images are resized before the classification process with the aim of ensuring uniformity of input dimensions for the image processing algorithm. This process is an important part of data pre-processing so that each image analyzed has consistent dimensions, so that the classification model can accept inputs with a homogeneous tensor structure. In Convolutional Neural Networks (CNN)-based research, image resize is stated as a prerequisite because all images must have a fixed size before being fed into the neural network to avoid errors at the input layer that require fixed dimensions on each sample of data. [2]

This uniformity of image size also serves to reduce visual scale variations between images, which if left different can affect the quality of feature extraction such as textures, shapes, or keypoints extracted by algorithms. For example, an empirical study showed that substantial changes in image size can significantly reduce the number of keypoints generated by feature extraction methods such as SIFT and SURF, which in turn can impact classification accuracy. [3]

Further, the resize process helps in simplifying computation and speeding up the model's training time because uniform-sized imagery reduces the computational complexity required during the training and inference process. This is especially important when working with large datasets or deep learning models that have memory and compute performance limitations. [4]

Thus, image resize is a crucial first step in the digital image processing pipeline for classification, as it helps maintain the consistency of input dimensions, facilitates more reliable feature extraction, and supports stability and efficiency in classification model training. [2]

2.3.2 Color Normalization and Conversion

Color normalization in digital image preprocessing is a crucial step to reduce the impact of exposure variations during image acquisition. Variations in light intensity that occur due to different environmental conditions can cause the color representation in pixels to be inconsistent, thus affecting the feature extraction process and classification accuracy. Normalization aims to equalize the range of color intensity values so that objects that have the same color characteristics remain uniform even though the lighting conditions are different when shooting. Normalization techniques, including RGB normalization, enable the system to detect targeted objects more resistant to changes in light intensity from the outside, resulting in more stable and accurate image processing. [2]

In addition to normalization, image conversion from the RGB color space to another color space such as HSV (Hue, Saturation, Value) is essential in color image processing. The HSV color model separates hue information, saturation, and brightness so that the color components are easier to analyze independently of light intensity. The value component is specifically concerned with the brightness level of the pixels, which allows image processing algorithms to focus more on the actual color characteristics without being overly affected by varying light intensities. For example, the use of HSV color spaces has been shown to make the color detection process on objects more robust to changes in lighting than RGB color spaces, because this separation of color and brightness components reduces sensitivity to light variations. [5]

In the context of fruit classification based on skin color, the transformation from RGB to HSV is very useful to highlight the differences in fruit color characteristics, such as hue from ripe to raw. When an image is converted to HSV, hue values can represent the characteristics of the fruit skin color regardless of lighting changes, while saturation and value provide additional information useful in color segmentation and feature detection. This process is often used in research related to fruit ripeness detection and object color segmentation, as it provides a more intuitive color representation and is less sensitive to changes in light intensity than raw RGB values. [6]

In general, normalization and color conversion not only help to deal with the problem of inhomogeneous lighting but also improve the stability of the image classification system through the use of color space that is more in line with human color perception and the computational needs of image processing algorithms.

2.3.3 Noise Reduction

Noise reduction or denoising is a crucial step in digital image preprocessing to improve image quality before the feature extraction or classification stage. Noise is an irrelevant visual disturbance that can arise as a result of errors in the imaging sensor, signal fluctuations, or data transmission processes, which make the image appear grainy or unclean. To overcome this interference, various filtering techniques such as Gaussian filters and median filters are used, whose function is to suppress noise while retaining important information in the image. [7]

Gaussian filter is a linear smoothing technique that uses the Gaussian distribution function to calculate the average pixel weight relative to its neighbors, so that noise with random variations can be reduced while maintaining the main structure of the image such as the edges and general shape of the object. This method is effective in reducing Gaussian-type noise without causing major distortion in the image, so it is often used as a pre-extension of the object detection stage in computer vision systems. [8]

Meanwhile, a median filter is a non-linear filtering technique that works by replacing the value of each pixel with the median value of the surrounding pixel value in a window. This approach is particularly effective at reducing salt-and-pepper noise or impulsive noise—noise that appears as random black-and-white dots—without blurring the edges of objects as is often the case with linear smoothing methods. This makes the median filter often chosen when the main goal is to clean up noise while maintaining important features in the image. [9]

2.4 Feature Extraction

Feature extraction is an important stage in image processing because the quality of the features produced greatly affects the classification results. In the context of image processing, a feature is a numerical representation of the visual characteristics of an object that helps machine learning or classification algorithms effectively distinguish between classes of objects. In general, feature extraction serves to transform high-dimensional raw images into more concise and informative representations, thus simplifying the classification process and improving accuracy. A comprehensive study of feature extraction techniques shows that this process is the foundation for a wide range of computer vision applications, including pattern recognition, object classification, and image segmentation, as it converts raw pixel data into descriptors that can be learned by classification models. Features used include: [10]

2.4.1 Color Features

Color features are one of the most common types of features used in image extraction because they provide strong visual information about the characteristics of an object based on the distribution of its color intensity. In many computer vision studies, color features are extracted from the RGB and/or HSV color space, and then statistical values such as mean and standard deviation from each color channel are used to represent the color pattern of an object. This statistical approach is often called Color Moments and has proven effective in capturing the characteristics of color distribution in images. [11]

In the RGB color space, each pixel is represented as a combination of three color components: red, green, and blue. Research on color feature extraction in RGB shows that calculating the mean values and standard deviations of the R, G, and B components helps in quantifying the dominant color distribution of objects, so these features can be used to distinguish object classes based on existing color variations. For example, image classification research using Color Moment on RGB color features noted the use of mean statistics, standard deviation, and skewness to differentiate the classes of objects in the dataset being tested. [11]

2.4.2 Texture Features

Texture features in image processing are statistical representations of the surface patterns of an object that reflect variations in pixel intensity and texture regularity in the image area. One popular method for extracting texture features is the Gray Level Co-occurrence Matrix (GLCM), which calculates the frequency of pixel pairs with a certain grayness value at a specific distance and orientation in an image. In this way, GLCM is able to capture spatial information and inter-pixel relationships that are not visible from just a single pixel intensity histogram.

In general, the use of texture features from GLCM such as contrast, correlation, energy, and homogeneity provides rich descriptive information about the surface patterns of objects in the image. These features help machine learning models or classification algorithms capture complex visual differences, especially when color or shape differences alone are not enough to reliably distinguish between classes of objects.

2.5 Data Sharing

In the Data Sharing stage, after the features of the dataset are extracted, the data is divided into training data and testing data with a comparison of 70% of the training data and 30% of the test data. This split is done randomly to ensure that each subset of the data is a random representation of the overall distribution of the dataset, so that the trained model gets a diverse sample of data and its evaluation of the test data reflects the model's capabilities outside of the training data. This random split technique is one of the most common approaches to dataset segmentation because it helps to avoid bias and provide a more objective estimation of the model's performance of data that the model has never seen before.

Random data sharing not only provides a balanced distribution of classes and features among subsets, but also creates realistic conditions for measuring how the model will perform when applied to real data outside of its training sample. This is important in the development of classification models because the main goal is for the performance obtained from the test data to reflect the model's ability to generalize to new and unknown cases. Therefore, the sharing of random data in the proportion of 70% training and 30% testing is standard practice in many machine learning studies to obtain reliable and unbiased model performance estimates.

2.6 K-Nearest Neighbor (KNN) Model Training

The K-Nearest Neighbor (KNN) algorithm is one of the simple yet effective classification methods that is widely used in supervised learning. The KNN performs the classification by measuring the distance between the feature vectors of the test data and the training data, then determining the class based on the majority of the nearest neighbors. One of the most commonly used distance metrics in KNN is Euclidean distance, which calculates sqrt from the square number of the difference between the features of the test data and the training data. Thus, if x and y are two feature vectors, the Euclidean distance between them is calculated as the root of the square of the difference in the value of each attribute, which is a measure of the similarity between the data examples.

In contrast to other learning models that require explicit training processes (e.g. weight adjustment in neural networks), KNN is known as a lazy learning algorithm in which there is no complex training process that explicitly adjusts the parameters of the model. The model only stores the drill data and performs distance computation when the test data query is given. This makes KNN performance highly dependent on the quality, representativeness, and quantity of available training data — a more extensive and representative training dataset will provide a more relevant neighbor for new data predictions, thus improving prediction accuracy.

2.6.1 Model Support Vector Machine (SVM) Training

The Support Vector Machine (SVM) algorithm is a supervised learning method that is widely used in the classification of two classes because of its ability to find an optimal hyperplane that maximally separates data samples from different classes. The main goal of SVM is to maximize margin—that is, the distance between the separator hyperplane and the support vectors (the closest data points of each class)—so that the model can provide good generalization capabilities to new data. This concept is known as Structural Risk Minimization and is one of the reasons why SVM is widely chosen in classification research.

However, when features in a dataset cannot be linearly separated in their native space, SVM leverages kernel tricks to map data to higher feature space dimensions that allow for linear separation. One of the most popular and effective kernels in handling non-linear cases is the Radial Base Function (RBF). The RBF kernel works by calculating similarities between data samples through the Gaussian function, which maps the data to high-dimensional spaces without the need to specify explicit transformations of each feature, so that the SVM can still find the optimal hyperplane even when the relationships between classes are non-linear.

2.7 Model Testing

After the classification model is trained, the next stage is model testing using pre-separated test data. At this stage, the model is used to predict the health class of star fruit based on feature vectors that have been extracted from the test data. The testing approach with data that the model has never seen at all during training aims to ensure that the evaluation of the model's performance takes place objectively and can describe the model's capabilities on real-world data, not just on data that has been studied. Evaluation research of generic classification models confirms that the use of test data provides a fairer picture of the model's predictive performance, as the comparison between the prediction results and the actual labels of the test data can be measured by metrics such as accuracy, precision, recall, and F1-score.

Overall, the model testing stage is an important element in the machine learning research flow, because without an honest evaluation of representative test data, model performance will only be partially measured on the training data and may lead to overfitting errors. Thus, systematic testing using test data ensures that the model is not only able to store patterns from the training data, but also to generalize to new and unknown data, which is the main goal of an effective classification system**.**

3. RESULTS AND DISCUSSION

The test results showed that the KNN algorithm produced an accuracy rate of 88%, while the SVM algorithm produced an accuracy rate of 92%. This value shows that both algorithms are able to perform the classification of star fruit health well. The test results showed that the KNN algorithm produced an accuracy rate of 88%, while the SVM algorithm produced an accuracy rate of 92%. This value shows that both algorithms are able to perform the classification of star fruit health well.

KNN experienced several misclassifications in images that have similar visual characteristics between classes. This is due to the KNN's dependence on the distance between data, so it is sensitive to the distribution of features [12]

Meanwhile, SVM shows better performance because it is able to form optimal decision boundaries. This advantage makes SVM more effective in handling data with complex and overlapping patterns. [13]

After training and testing using star fruit image data, classification results were obtained using KNN and SVM algorithms. The dataset consisted of 630 healthy star fruit and 497 unhealthy or bad star fruit, with a division of 70% training data and 30% test data.

3.1 KNN Classification Results

The results of the evaluation of the K-Nearest Neighbors (KNN) model with the parameter k = 5 showed that the classifier was able to distinguish the condition of healthy and unhealthy star fruit with an accuracy of 61%. This accuracy value reflects the overall percentage of the model's predictions that actually match the actual label, i.e. the number of correct predictions divided by the total test data. This kind of evaluation is generally presented through a confusion matrix, which represents the performance of the classification by grouping the prediction results into four categories: true positive (TP), true negative (TN), false positive (FP) and false negative (FN), thus providing more detailed insight into the strengths and weaknesses of the model in distinguishing between classes. For example, a confusion matrix is commonly used in classification research to evaluate the performance of KNN models with accuracy, precision, and recall metrics, where accuracy is calculated based on the correct number of classifications against the total sample tested.

Table 1. Confusion Matrix KNN

Pred / Ref	Bath	Healthy
Bath	75	59
Healthy	74	130

From the confusion matrix table, it can be seen that most fruits are classified correctly, but there are some misclassifications, especially in samples that have conditions that are close to the boundary between healthy and unhealthy classes. Errors like this often arise in classification boundaries where the features of two classes overlap or are so similar that the model has difficulty determining which class is correct. In the machine learning literature, it is explained that data that is in the overlapping area between two classes tends to give predictive results that are more often wrong, because this kind of sample does not have a clear representation of features to map to one of the classes firmly. This means that when the feature patterns of an instance are very similar to both classes, classifiers such as KNN or other methods are prone to misclassification at those points.

3.2 SVM Classification Results

The results of the evaluation of the Support Vector Machine (SVM) model trained using the Radial Base Function (RBF) kernel showed excellent performance in the star fruit dataset, with an accuracy of 92%, higher than the classification results using KNN. These accuracy values reflect the percentage of correct predictions in the test data, and comparisons between SVM and KNN models generally support the finding that SVMs with RBF kernels are able to handle the complexity of non-linear data patterns more effectively than KNN, resulting in more accurate classifications. Other studies in the literature have also shown that SVMs with the RBF kernel often perform best among some traditional classification algorithms including KNN and other methods on complex dimensional datasets. For example, a study on the Dry Bean dataset found that SVMs with RBF kernels achieved the highest accuracy (93.34%) compared to various other approaches, demonstrating the advantages of RBF kernels in capturing non-linear data structures.

Table 2. Confusion Matrix SVM

Pred / Ref	Bad	Healthy
Bad	40	8
Healthy	109	181

The Support Vector Machine (SVM) model using the RBF kernel shows a better ability to distinguish healthy and unhealthy fruit classes compared to KNN, especially in cases where color features between classes have similar or overlapping values. This is consistent with the literature that SVM, through the use of kernels such as RBF, is effective in handling data that cannot be separated linearly within its original feature space by mapping the data to higher dimensional spaces so that the optimal decision boundary can be found more clearly. This approach helps SVM to suppress errors in samples whose features are close to the boundaries between classes, which is often a weakness of distance-based methods such as KNN.

4. DISCUSSION

The results of the evaluation showed that the two classification algorithms (SVM and KNN) were able to classify the condition of star fruit quite well, but the Support Vector Machine (SVM) model gave more accurate results than the KNN. These findings are consistent with many previous studies that have shown that SVM often outperforms KNN in image classification tasks, especially when features between classes are complex or overlap. In one study on facial image classification, for example, SVM showed higher accuracy compared to KNN, reflecting SVM's ability to find more optimal decision boundaries even though the data had a variety of complex features.

The difference in performance between SVM and KNN is largely due to how each algorithm works. SVM seeks to maximize the margins between classes by finding the best hyperplanes that separate those classes in the feature space, while KNN relies solely on the proximity of neighbors in the feature space without an explicit boundary learning process.

This approach makes SVM more resistant to noise and variations of small features that often occur in fruit images with similar color and texture conditions between classes.

5. CONCLUSION

Based on the results of the research that has been conducted, it can be concluded that the K-Nearest Neighbor (KNN) and Support Vector Machine (SVM) algorithms can be used effectively to perform a classification of star fruit health based on digital imagery. Both algorithms are able to utilize color and texture feature information to distinguish fruits in healthy and unhealthy conditions. Nevertheless, the SVM algorithm shows better performance than KNN, which is indicated by the higher classification accuracy value. This indicates that SVM is better able to handle the complexity of data patterns and variations in features contained in star fruit images.

The results of this study also prove that the application of machine learning methods in agriculture and digital image processing can be an automated, objective, and efficient solution in helping the process of assessing the quality and health of fruits. With an image-based system, the classification process is no longer entirely dependent on human visual observation which is subjective and has the potential to produce errors, especially in the condition of the fruit which has a less significant visual difference.

THANK YOU

The author would like to thank all parties who have provided support, assistance, and contributions both directly and indirectly in completing this research. Special thanks are extended to the supervisors who have provided invaluable direction, guidance, and input during the research and writing process of this report.

BIBLIOGRAPHY

[1] Niendha Biell Binna, T. Rohana, H. Y. Novita, and S. Faisal, "CLASSIFICATION OF TOMATO TYPES USING K-NEAREST NEIGHBOR ALGORITHM AND SUPPORT VECTOR MACHINE," *Journal of Informatics, Technology and Science (Jinteks)*, vol. 7, no. 2, pp. 800–807, May 2025, doi: 10.51401/jinteks.v7i2.5743.

[2] M. Hashemi, “Enlarging smaller images before inputting into convolutional neural network: zero-padding vs. interpolation,” *J Big Data*, vol. 6, no. 1, Dec. 2019, doi: 10.1186/s40537-019-0263-7.

[3] A. Priadana and U. Jenderal Achmad Yani Yogyakarta, "ANALYSIS OF THE EFFECT OF THE SIZE OF THE IMAGE SIZE OF THE RESIZING RESULTS ON THE NUMBER OF KEYPOINTS FROM THE EXTRACTION OF THE FEATURE IN THE SIFT AND SURF METHODS."

[4] X. Du, Y. Sun, Y. Song, W. Chi, L. Dong, and X. Zhao, “Impact of Input Image Resolution on Deep Learning Performance for Side-Scan Sonar Classification: An Accuracy–Efficiency Analysis,” *Remote Sens (Basel)*, vol. 17, no. 14, p. 2431, Jul. 2025, doi: 10.3390/rs17142431.

[5] G. Moreira, S. A. Magalhães, T. Pinho, F. N. dos Santos, and M. Cunha, “Benchmark of Deep Learning and a Proposed HSV Colour Space Models for the Detection and Classification of Greenhouse Tomato,” *Agronomy*, vol. 12, no. 2, p. 356, Jan. 2022, doi: 10.3390/agronomy12020356.

[6] A. Paliling, M. Muchtar, and F. Fardian, "HSV-KNN-Based Intelligent System for Detecting Dragon Fruit," *e-JUSITI Journal (Journal of Information Systems and Information Technology)*, vol. 14, no. 1, pp. 46–55, May 2025, doi: 10.36774/jusiti.v14i1.1718.

[7] I. Gede *et al.*, "Noise Reduction in Digital Imagery Using Arithmetic Mean, Harmonic Mean, Gaussian, Max, Min, and Median Filters by Comparing Psnr," *Indonesian Journal of Computer Science (JIK)*, vol. 5, no. 2, 2020.

[8] Mazayah Tsaqofah, Lailan Sofinah Harahap, and Dea Syahfira Hasibuan, “Optimization of Digital Image Processing Through Gaussian Filtering for Noise Reduction,” *Journal of Artificial Intelligence and Engineering Applications (JAIEA)*, vol. 4, no. 3, pp. 2157–2161, Jun. 2025, doi: 10.59934/jaiea.v4i3.1120.

[9] M. F. Alamsyah, L. S. Harahap, and M. Firjatullah, "Improving Digital Image Quality through Noise Reduction Using Median Filtering," *Data Sciences Indonesia (DSI)*, vol. 5, no. 1, pp. 96–104, Jul. 2025, doi: 10.47709/dsi.v5i1.6290.

[10] S. Hallur and A. Gavade, “Image feature extraction techniques: A comprehensive review,” *Franklin Open*, vol. 12, p. 100366, Sep. 2025, doi: 10.1016/j.fraope.2025.100366.

[11] F. D. Febriani, Y. A. Sari, and R. C. Wihandika, "Classification of Indonesian Traditional Cake Images Based on RGB Color Moment Feature Extraction Using K-Nearest Neighbor," 2019. [Online]. Available: http://j-ptiik.ub.ac.id

[12] T. Cover and P. Hart, “Nearest neighbor pattern classification,” *IEEE Trans Inf Theory*, vol. 13, no. 1, pp. 21–27, Jan. 1967, doi: 10.1109/TIT.1967.1053964.

[13] C. Cortes and V. Vapnik, “Support-vector networks,” *Mach Learn*, vol. 20, no. 3, pp. 273–297, Sep. 1995, doi: 10.1007/BF00994018.