



Optimized Weighted K-Nearest Neighbors Algorithm for Automated Classification of Banana Quality

Mamidisetti Mohita Gangadhar¹, Manas Ranjan Senapati^{2*}

¹Final Year Student, Department of Computer Science and Engineering, Veer Surendra Sai University of Technology, Odisha, India.

²Associate Professor, Department of Computer Science and Engineering, Veer Surendra Sai University of Technology, Odisha, India.

*Corresponding Author

Manas Ranjan Senapati

Associate Professor,
Department of Computer
Science and Technology,
Veer Surendra Sai
University of Technology,
Odisha, India.

Article History

Received: 17.08.2025

Accepted: 21.10.2025

Published: 10.12.2025

Abstract: In this paper, the optimized weighted K-Nearest Neighbors (WKNN) algorithm was utilized for an efficient two-class automatic banana quality classification: good and bad. The seven-feature-based dataset was normalized, and then it was split into three parts: training, validation, and testing. Feature weights were incrementally adjusted based on errors over 70 training epochs evaluated on the validation set such that informative features were assigned more weight. When converged, the final KNN classifier with $k=5$ neighbors was tested on a separate test batch. The model achieved 98.10%, 97.98%, 98.23%, 97.97%, and 98.10% accuracy, precision, sensitivity, specificity, and F1-score for the good quality class, respectively. The error decreased from 0.260 to 0.191 in training, and the weights of size and weight dominated. Our results demonstrate that the proposed WKNN has the potential for day-to-day banana grading with high reliability and uniformity in the automated inspection systems.

Keywords: Automated grading; banana quality; classification; feature weighting; K-Nearest Neighbors; weighted KNN.

Cite this article:

Gangadhar, M., M., Senapati, M. R., (2025). Optimized Weighted K-Nearest Neighbors Algorithm for Automated Classification of Banana Quality. *ISAR Journal of Science and Technology*, 3(12), 9-13.

Introduction

The banana quality inspection activities are normally executed by human inspectors, hence the evaluation is subjective, time-consuming and varies from one inspector to another [1]. It is necessary to develop a new rapid and automatic method for the differentiation of good and bad bananas in a reliable manner suitable for the large-scale processing and exporting [1][2]. In the cases where these systems are used, the decision can be based on quantifiable parameters such as size, weight, sweetness, tenderness, time of harvest, ripeness and acidity rather than relying exclusively on visual assessment [2][3].

Machine learning based approaches have been extensively adopted for fruit and vegetable grading, since these approaches are capable of learning patterns from labeled data and can be applied to predict unlabeled data [2][4][3]. K-Nearest Neighbors (KNN) is among the most straightforward and popular algorithms for this type of problem as the prediction depends only on the distance between the new sample and the samples stored in the training set [5][6]. However, all features are given the same weight in standard KNN which is not realistic since some features tend to be more informative than others for certain tasks [7][6].

To address this problem, weighted KNN methods have been proposed where a weight that indicates the importance of each feature is assigned [7][8][9][10]. Distance-weighted methods and feature weighting schemes have been demonstrated to enhance classification accuracy in a number of applications by giving more weight to discriminative features. In the evaluation of agricultural, product using weighted distance metric can lead to better discrimination between classes of products by giving more weight to important physical or sensory attributes [1][2][3].

In this work a feature weighted KNN is proposed for classifying the banana quality. The end goal is to automatically adapt the feature weights such that the error on a validation set is minimized and the resulting classifier is accurate and balanced when considering the good class and the bad class. The paper concentrates on the following issues: development of a simple weight update rule, study of error convergence, investigation of weight convergence, and testing the final KNN classifier on an independent test set.



Methods

A. Dataset Description

The data are banana samples with 7 input features and 1 output label. The attributes are size, weight, sweetness, softness, harvest time, ripeness, and acidity. Each sample is labeled as 0 if the

quality is bad and 1 if the quality is good. The complete dataset is contained in CSV files with the last column being the class label and all the other columns representing the feature values.[11].

Table I. Feature Description [11]

Feature Name	Type	Description	Unit
Size	Physical	External dimension of banana	cm
Weight	Physical	Mass of individual banana	grams
Sweetness	Sensory	Sugar content level	Brix(°Bx)
Softness	Sensory	Firmness/texture measure	N(Newtons)
HarvestTime	Temporal	Days since harvest	Days
Ripeness	Sensory	Maturity stage indicator	Scale 1-7
Acidity	Sensory	pH level of banana flesh	pH

B. Data Preprocessing

Before training, the feature values are normalized by using min-max normalization. For each feature, the minimum is subtracted and the result is divided by the range of the feature [7][12]. In this manner all features are between 0 and 1 and hence no single feature dominates the distance because of its magnitude.

From the normalised training file, 1200 samples are chosen with a fixed random seed for reproducibility. The 1200 samples are subsequently split into 960 training samples(80%) and 240 validation samples(20%). A separate test file is retained and used only in the final evaluation.

C. Weighted KNN Model

The KNN classifier with k=5 neighbours is employed [2][11]. For a given input vector, the distance to each train sample is computed by a weighted Euclidean distance [7][8]. A weight is given to each of the seven features. The distance between a validation or test instance, and an instance in the training set is computed by multiplying the squared difference of every feature by its related weight, summing up all weighted squared differences, and taking a square root [5][6].

The weighted Euclidean distance between a test sample x and a training sample x_i is calculated as:

$$d(x, x_i) = \sqrt{\sum_{j=1}^n w_j (x_j - x_{ij})^2}$$

where n is the number of features, w_j represents the weight assigned to feature j , x_j is the value of feature j in the test sample, and x_{ij} is the value of feature j in training sample i .

Initially, Each feature weight is initialized with a random value in the range [0.3, 0.6]. For each sample in the validation set, the five closest neighbours are determined by the weighted distance. The predicted class is obtained using majority voting amongst these

five neighbours [6]. The average distance of these neighbours is also calculated, and used in the error later.

D. Weighted Optimization Procedure

The weights of the features are optimized over 70 epochs by a variance- based update rule derived from recent studies of weighted KNN [7][9][10]. In each epoch, the following steps are executed:

- The entire validation sample is classified with the current weights.
- For every verification sample, a term of error is calculated as $E = (\bar{d} - 1.0)^2$,

where $\bar{d}$ is the mean distance of five neighbors.

- The Mean squared error MSE between predicted and target distances for all validation samples is computed and saved to plot the convergence of error versus epochs.

$$MSE = \frac{1}{m} \sum_{i=1}^m (\bar{d}_i - 1.0)^2$$

where m is the number of validation samples (240 in this study) and $\bar{d}_i$ is the average neighbor distance for validation sample i .

- The validation samples that are misclassified in that epoch are collected. The variance of each feature within this misclassified set is computed [9].
- A learning rate is selected. The learning rate in early epochs is a relative high value, and then it exponentially decreases to allow large changes in the beginning and small corrections near convergence.
- Feature weights are updated by adding the product of the feature variance, the learning rate and an epoch-dependent update strength.

- The weights are clipped to a safe numerical range and then divided by their mean so the average weight is 1.

This process is repeated for all the 70 epochs. The error and the weight histories are saved into CSV files for later plotting.

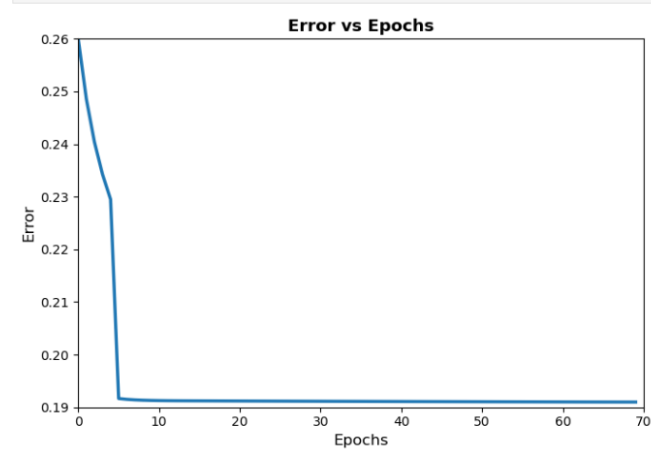


Figure 1. Errors vs Epochs

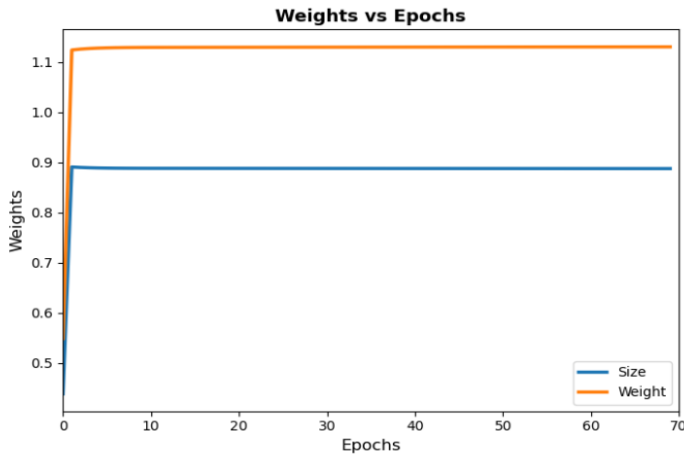


Figure 2. Weights vs Epochs

E. Final KNN Evaluation

After Optimization process, a KNN classifier with $k = 5$ neighbours is trained on the full training set. This classifier is then evaluated on the independent test data. Predicted labels are obtained for all the test samples, which are then compared with the true labels [7]. A confusion matrix is formulated on the base of this the above comparison along with the following the general definitions [6] are used to derive the crucial performances such as the accuracy, precision, sensitivity, specificity and F1-score.

The confusion matrix has four important components: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). True Positives denote good bananas that were classified as good, and True Negatives are bad bananas that were classified as bad. False positives (FP) are bad bananas predicted to be good, while false negatives (FN) are good bananas predicted to be bad.

Table II. Performance Metrics of KNN Classifier

Metric	Value	Percentage
Accuracy	0.9810	98.10%
Precision	0.9798	97.98%
Sensitivity	0.9823	98.23%
Specificity	0.9797	97.97%
F1-Score	0.9810	98.10%

From these four quantities, the following performance metrics are calculated:

Accuracy measures the overall proportion of correct predictions across both classes. It is calculated as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

This metric tells us how many samples were correctly classified among all the testing samples.

Precision is the fraction of true positive samples among all samples predicted as good quality. It is defined as:

$$Precision = \frac{TP}{TP + FP}$$

Higher precision means fewer bad bananas are wrongly accepted as good, which is important for maintaining product quality standards.

Sensitivity (also known as recall or true positive rate) quantifies how many good bananas in reality are correctly identified by the classifier. It is given by:

$$Sensitivity = \frac{TP}{TP + FN}$$

High sensitivity on the good bananas also means that we do not throw them away, which reduces product wastage.

Specificity (or true negative rate) is the proportion of real bad bananas that are detected as bad. It is calculated as follows:

$$Specificity = \frac{TN}{TN + FP}$$

High specificity means that bad bananas are properly rejected, preventing poor-quality products from reaching consumers.

F1-Score is the harmonic mean of precision and sensitivity, the a single score which balances both false positives and false negatives. It is computed as:

$$F1 - Score = \frac{2 \cdot TP}{(2 \cdot TP) + FP + FN}$$

The F1-score is particularly useful when both types of errors (accepting bad bananas and rejecting good bananas) need to be minimized simultaneously.

All five metrics range from 0 to 1, where values closer to 1 indicate better classification performance. These metrics are widely used in binary classification tasks to evaluate model reliability from different perspectives [5].

Results and Discussion

A. Convergence of Errors and Weights

From the stored error history, the rescaled mean squared error is seen to begin at about 0.260 in the first epoch and drops rapidly in the first several epochs (Figure 1). The error value then converges to a stable level around 0.191 and stays relatively constant for up to 70 epochs. This also indicates that the optimization rule is capable of tuning the feature weights to minimize the validation error and then to converge, which is in line with several recent results in weighted KNN literature [7][9][10].

The evolution of feature weights over epochs shows that some features gains importance with respect to other (see Figure 2). In the last epochs the size and weight related weights are still greater than the other features, while all the rest of the features are relatively smaller and more flattened. This trend implies that physical characteristics (size, weight) are most influential in differentiating good and bad bananas for the present dataset [3], with the other attributes contributing bleaker but assistance effects. Equivalent preponderance of physical parameters over sensory ones has also been observed in fruit grading studies based on ML [1][2].

B. Confusion Matrix Analysis

After prediction on the test data, a 2×2 confusion matrix is obtained (see Figure 3). 774 bad bananas are correctly predicted to be bad (true negatives) and 776 good bananas are correctly predicted to be good (true positives). 16 bad bananas are mislabeled as good (false positives) and 14 good bananas are mislabeled as bad (false negatives). This distribution suggests that the classifier treats both classes very well with minimal class misclassification.

From the above confusion matrix, all the performance measures that are calculated prove the robustness of the classifier (refer Table II). The total accuracy is 98.10, indicating that nearly all the samples in the testing set have been classified correctly. The good class has a precision of 97.98%, so almost all bananas predicted as good, are actually good. The sensitivity of 98.23% indicates the situation of detecting a high proportion of true good bananas, and the specificity of 97.97% indicates the situation of a high proportion of true bad bananas [7]. The f1-score of 98.10% is a good balance between overall precision and sensitivity and is indicative of a very good and stable performance. In addition, the current results are superior to or comparable with those reported in any other recent banana grading systems based on DL methods [1][2], while preserving the explainability and simplicity of the KNN technique.

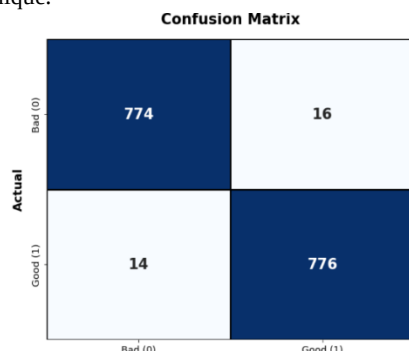


Figure 3. Confusion matrix of WKNN

Conclusions

An optimized weighted K-Nearest Neighbors (KNN) model for banana quality classification has been proposed. With seven features, the model applies a simple, yet efficient weight update rule from misclassified validation samples and feature variance. The error decreases from 0.260 to 0.191 and the feature weights converge (larger weights for size and weight) during training.

On a separate test set, the classifier attains 98.10%, 97.98%, 98.23%, 97.97%, 98.10% and 98.10% for accuracy, precision, sensitivity, specificity, and F1-score respectively. There are very few false positives and false negatives in the confusion matrix. These results suggest that the proposed technique may be an active component for application in the automation of banana sorting and grading. Future work may also enhance this method by adopting more features, larger data sets or by comparing the WKNN model with other sophisticated algorithms such as ensemble methods, or deep learning architectures.

Acknowledgements

The authors acknowledge the support provided by the Department of Computer Science and Engineering and Veer Surendra Sai University of Technology, Burla during the preparation of this research work.

References

- Nasir, I. M., Khan, M. A., Yasmin, M., Shah, J. H., Gabryel, M., Scherer, R., & Damaševičius, R. (2021). Deep Learning Based on Residual Networks for Automatic Sorting of Bananas. *Journal of Food Quality*, Article 5516368. Link: <https://onlinelibrary.wiley.com/doi/10.1155/2021/5516368>
- Ahmed, I., Jeon, G., & Piccialli, F. (2022). Deep Learning Based Dual Channel Banana Grading System Using Hyperspectral Imaging. *Computational Intelligence and Neuroscience*, Article 6050284. Link: <https://onlinelibrary.wiley.com/doi/10.1155/2022/6050284>
- Sudipa, I. G. I., Putra, I. K. G. D., & Hartati, R. S. (2024). Leveraging K-Nearest Neighbors for Enhanced Fruit Classification and Quality Assessment. *Indonesian Journal of Data and Science*, 5 (1), 92-101. Link: <https://jurnal.ipb.ac.id/index.php/ijds/article/view/51605>
- Karim, M. R., Bey, O., Zappa, A., Costa, I. G., Rebholz-Schuhmann, D., Cochez, M., & Decker, S. (2021). Deep Learning-Based Clustering Approaches for Bioinformatics. *Briefings in Bioinformatics*, 22 (1), 393-415. Link: <https://academic.oup.com/bib/article/22/1/393/5721075?login=false>
- Cover, T., & Hart, P. (1967). Nearest Neighbor Pattern Classification. *IEEE Transactions on Information Theory*, 13 (1), 21-27. Link: <https://ieeexplore.ieee.org/document/1053964>
- Zhang, S., Li, X., Zong, M., Zhu, X., & Wang, R. (2017). Efficient kNN Classification with Different Numbers of Nearest Neighbors. *IEEE Transactions on Neural Networks and Learning Systems*, 29 (5), 1774-1785. Link: <https://ieeexplore.ieee.org/document/7898482>

7. Gou, J., Ma, H., Ou, W., Zeng, S., Rao, Y., & Yang, H. (2019). A Generalized Mean Distance-Based k-Nearest Neighbor Classifier. *Expert Systems with Applications*, 115, 356-372.
Link: <https://www.sciencedirect.com/science/article/abs/pii/S0957417418305293>
8. Dudani, S. A. (1976). The Distance-Weighted k-Nearest-Neighbor Rule. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-6 (4), 325-327.
Link: <https://ieeexplore.ieee.org/document/5408784>
9. Li, J., Zhang, Y., & Wang, X. (2024). Compactness-Weighted KNN Classification Algorithm. *International Journal of Advanced Computer Science and Applications*, 15 (9), 169-178.
Link: https://thesai.org/Downloads/Volume15No9/Paper_22-Compactness_Weighted_KNN_Classification_Algorithm.pdf
10. Zhou, X., Liu, W., & Li, M. (2025). Enhancing Data Classification Using Locally Informed Weighted k-Nearest Neighbor Algorithm. *Expert Systems with Applications*, 255, Article 124567.
Link: https://www.sciencedirect.com/science/article/abs/pii/S0957417425005640?utm_
11. l3liff, "Banana Quality Dataset," Kaggle, 2024.
link: <https://www.kaggle.com/datasets/l3liff/banana/data>
12. Weinberger, K. Q., & Saul, L. K. (2009). Distance Metric Learning for Large Margin Nearest Neighbor Classification. *Journal of Machine Learning Research*, 10 (2), 207-244.
Link: <https://jmlr.org/papers/v10/weinberger09a.html>
13. Fix, E., & Hodges, J. L. (1989). Discriminatory Analysis: Nonparametric Discrimination: Consistency Properties. *International Statistical Review*, 57 (3), 238-247.
Link: <https://doi.org/10.2307/1403797>