**Accredited Ranking SINTA 2**
Decree of the Director General of Higher Education, Research, and Technology, No: 158/E/KPT/2021
Validity period from Volume 5 Number 2 of 2021 to Volume 10 Number 1 of 2026

Published online on: **http://jurnal.iaii.or.id**

**JURNAL RESTI**
**(Rekayasa Sistem dan Teknologi Informasi)**
Vol. 7 No. 6 (2023) 1456 - 1463 ISSN Media Electronic: 2580-0760

# Date Fruit Classification using K-Nearest Neighbor with Principal Component Analysis and Binary Particle Swarm Optimization

Wikky Fawwaz Al Maki[1], Khaidir Mauladan[2], Indra Bayu Muktyas[3]
[1,2]School of Computing, Telkom University, Bandung, Indonesia
[3]STKIP Surya, Tangerang, Indonesia
[1]wikkyfawwaz@telkomuniversity.ac.id, [2]khaidir@student.telkomuniversity.ac.id, indrabayu.muktyas@stkipsurya.ac.id

***Abstract***

*Various cultivars of date fruit distributed throughout exhibit diverse complexity and unique attributes, including color, flavor, shape, and texture. These distinctive characteristics and appearance occasionally may lack variability in date fruits, as various kinds of date fruit may have subtle differences in color, shape, and texture. To overcome the difficulty of sorting and classifying multiple types of date fruit, a classification model was developed to categorize date fruit based on their visual appearances and digital characteristics. This study proposes a classification system that categorizes date fruit into five distinct types. The system achieves this by extracting features related to date fruit images' color, shape, and texture. Specifically, color moments, HOG descriptors, and circularity are used for feature extraction. The resulting high-quality training data is then used to train a K-Nearest Neighbor (KNN) classifier. Considering the parameters applied in developing the proposed classification model is essential. Therefore, the proposed KNN model will be optimized by Principal Component Analysis (PCA) and Binary Particle Swarm Optimization (BPSO). PCA is employed for dimensionality reduction, whereas BPSO is implemented to discover the optimal neighbors. The experimental results demonstrated that the classification model achieved an accuracy of 93.85%, a considerable improvement of 12% over barebone KNN.*

*Keywords: histogram of orientation gradients; principal component analysis; k-nearest neighbor; binary particle swarm optimization*

## 1. Introduction

Date fruit, scientifically designated Phoenix Dactylifera, is a nutritious and luscious food enjoyed by people in many portions of the world and has a long history of cultivation dating back to historical civilizations. Date fruit is one of the most global producers of fruits, with over 14 million in 2020, with the top product coming from the Middle East Country [1] with various kinds of date fruit, each with its distinctive characteristics. With different varieties of date fruit spreading around the globe, however, not limited to Ajwa, Galaxy, Nabtat Ali, Shaishe, Sugaey, Rutab, Meneifi, Medjool, and Sokari, each type has its distinctive texture, taste, color, and shape. However, several challenges must be overcome while categorizing date fruits. An obstacle lies in the visual diversity exhibited by date fruits., as several kinds of date fruit may have subtle differences in color, shape, and appearance. Overcoming this challenge is crucial to satisfy market demands, provide quality control, promote breeding research initiatives, and adhere to trade restrictions. The date fruit business may also boost productivity and encourage growth and innovation in the international market by simplifying the categorization process. This challenge can be resolved by adopting deep learning approaches specializing in subtle digital data changes. As proposed in [2], by utilizing the VGG-16 to develop an image classification model, the study could classify three different kinds of date fruits. These results demonstrated how automated date fruit sorting may benefit from a deep learning algorithm. The developed model was tested and assessed on three varieties of date fruit. Consequently, it has not been proven to perform well with additional classes and traditional machine learning algorithms. Further research is required to understand its capabilities and potential limitations fully.

In image classification, different features in an image could potentially be relevant to influence the model's performance. A study [3] stated that color and shape characteristics may be utilized to recognize fruits not limited to apples and can lessen the model's accuracy. Additionally, it is claimed that color characteristics can differentiate fruits with distinctive colors, such as date

Accepted: 01-10-2023 | Received in revised: 25-12-2023 | Published: 30-12-2023

fruits. The texture feature is an additional attribute used in object classification. The creases on the surface of date fruit tend to be irregular to human perception. Therefore, image descriptors can be utilized to capture the texture patterns of an image. In [4], the researchers successfully implemented the Pyramid Histogram of Orientation Gradients and SVM to classify three big cat species based on their fur pattern. The proposed model achieved an accuracy of 91.07%. The "curse of dimensionality" will occur throughout the training process as a result of the combination of various aspects from color, shape, and texture, which results in high dimensional features [5], [6]. Principal Component Analysis (PCA) is expected to resolve this issue by transforming the high-dimensional features into a smaller dimensional set of features.

Date fruit classification is a supervised learning problem because the training data are labeled and used to train the model. Supervised learning implies predicting a class label for an image based on features and patterns in the image. KNN is used in this study to solve the supervised learning problem. One hyperparameter of the KNN algorithm is the value $k$, which specifies the number of neighbors used to generate predictions. The value of $k$ can be sensitive as it can considerably impact the model's performance. If the value of $k$ is diminutive, the model may be overly sensitive to noise and potentially generate deficient new predictions. In contrast to the $k$ value being overly vast, the model may be less sensitive and produce overly smooth predictions that cause classes to blend in. In [4] and [7], the authors proposed an image classification method by implementing barebone KNN. However, the proposed models provided inadequate results since the accuracies are below 75%. Therefore, this paper proposes an optimization technique to improve KNN as an image classification method.

One of the KNN hyperparameters that can be optimized is the number of neighbors. A population-based optimization approach is esteemed to be efficient. Also, it has a minimal computational workload. Particle swarm optimization (PSO) is a common and widely used approach for optimization problems. However, PSO cannot generate a discrete value for KNN to use as the number of neighbors. Binary PSO was introduced in 1995 to solve discrete problems [8]. An example of BPSO implementation is a study for breast cancer prediction based on images with two dissimilar datasets that used BPSO as a feature selection technique [9]. With an overall improvement of 4-8% and 1-4% for both datasets, the cited study's usage of BPSO as a strategy to potentially enhance the performance of the respective predictive model is demonstrated to be successful.

This study aims to construct a profound classification model that can distinguish distinctive kinds of date fruit. The model will be optimized to improve the performance of the model. This study's structure is brief as follows: The suggested study technique, which comprises image preprocessing, image augmentation, and image segmentation, is described in Section II. The processed images then employed feature extraction and dimensionality reduction for the extracted features. The extracted features are then used to feed the classification model and employ optimization techniques for performance evaluation. Section III discusses the outcomes of this study and analyzes it. Section IV concludes the findings of this study.

## 2. Research Method

The system in this study was built using an approach that consists of multiple steps. The system begins dataset preprocessing for effectiveness by reducing image resolution and segmenting images, then moves on to feature extraction, producing data features that will be improved with dimensionality reduction. The data features of the dimensionality-reduced data are then divided into train and test data, which are fed to the KNN classifier to create a classification model that will be optimized using BPSO. The performance of the model is subsequently assessed based on its accuracy, recall, precision, and F1 score. The process of the above sequences is presented in Figure 1.

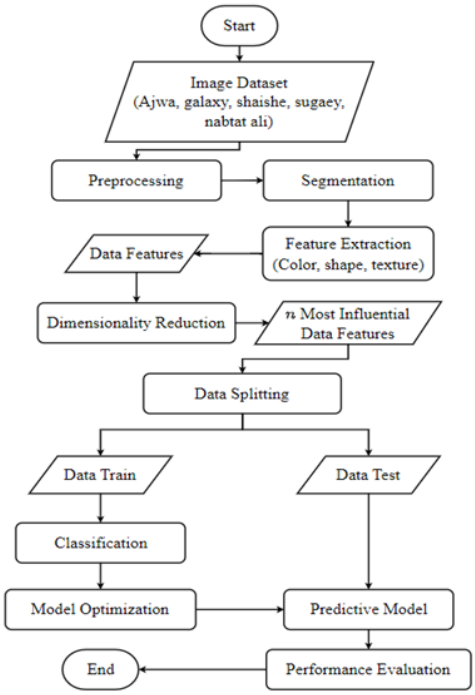


Figure 1. Proposed system architecture

### 2.1. Dataset

The date image dataset is acquired from www.kaggle.com [10]. The dataset is still considered to be a recent upload from 2021. Each date fruit class

DOI: https://doi.org/10.29207/resti.v7i6.4839

comprises 130 images of date fruits in a regulated environment. Each image in the dataset is 3456 pixels wide and 2304 pixels high except Ajwa, which has a size of 5184x3456, and every image has a distinguishable white background.

In this study, each class comprised 130 images from the dataset, comprising 650 images. The images were meticulously selected to represent the diversity of characteristics within each class and ensure that the training employed a comprehensive data set. Using a large set of images enables us to develop the model accurately and assess its performance on various date fruit types. The appearance of each date fruit class is shown in Figure 2.

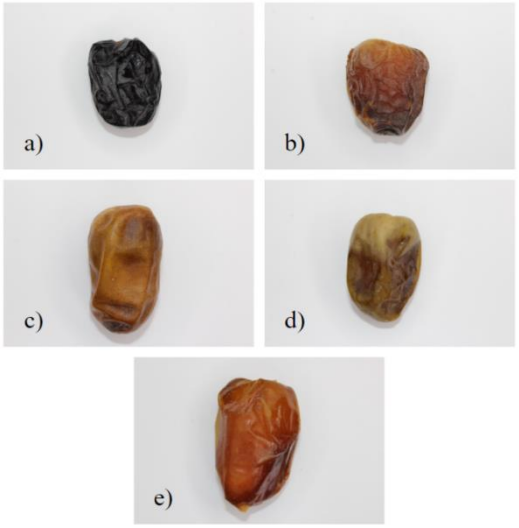


Figure 2. Dataset sample: a) Ajwa, b) Galaxy, c) Nabtat Ali, d) Shaishe, e) Sugaey

## 2.2. Preprocessing

A strategy to enhance the efficiency of image classification algorithms is to downscale or resize the image dataset. This strategy may minimize the computational time and decrease the computational workload necessary for data preprocessing. Nevertheless, balancing the trade-off of computing efficiency and the potential information loss is crucial. In this connection, information loss can occur when the resolution of the images is decreased. A lower dimension of 128x128 pixels is determined to minimize computational time and workload.

## 2.3. Segmentation

In the usual events, images obtained using an image acquisition device invariably comprise a background posterior to the target of interest. This phenomenon can lead to uncertain feature extraction, and when it comes to fruit classification using machine learning, segmented photos can impact the model's classification accuracy. Image segmentation extracts the target of interest in the image required for classification to be identified and analyzed while further measured to feed the classification model [11].

In this study, GrabCut is utilized as one of the image segmentation techniques. Using GrabCut, an initial bounding box is selected around the target of interest. The chosen image inside the bounding box will be considered the foreground, while the image outside the selected region will be regarded as the background [12]. GrabCut then determines whether each pixel is a part of the background or foreground using the Mincut/Maxflow algorithm.

The GrabCut technique allows users to provide input to accurately segment an image by specifying regions that are definitely foreground, definitely background, potentially foreground, or potentially background. An example of GrabCut usage is shown in Figure 3.

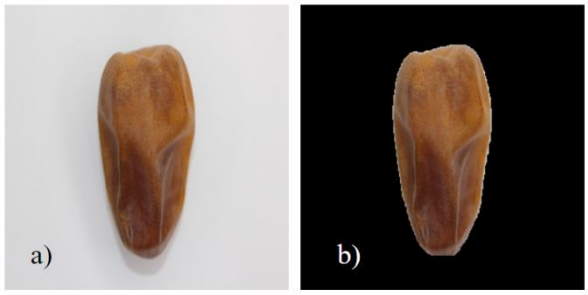


Figure 3. a) original image, b) Segmented image

## 2.4. Feature Extraction

Feature extraction extracts digital data from images, in which the extracted data is then identified and further processed. From a single snapshot of a date fruit, a substantial amount of digital information can be collected, encompassing many aspects such as shape, color, and texture. Shapes possess a more significant portion in classifying distinct objects. However, as date fruits are predominantly spherical, circularity is proposed. Color features, typically present in color channel, have been observed by implementing many methods. Several approaches use the color channel values of each pixel, namely the mean, standard deviation, and skewness. These measures are commonly referred to as color moments. In addition, date fruits possess distinctive textures, typically increasing, which can serve as a feature. In this study, HOG is employed to obtain textures of the image dataset.

### 2.4.1. Color Moments

Color moment is a highly effective and widely employed technique to represent the color features of an image, particularly in image processing. Color moment is a favored approach for object recognition and image classification because it briefly explains the color content and is resistant to fluctuations in the image. Color moments are calculated using RGB and HSV

DOI: https://doi.org/10.29207/resti.v7i6.4839

color spaces. The calculation of color moment is performed by using the color channel's skewness and mean. Variance is further measured to represent each color channel's distribution [13]. The mathematical model of mean, variance, and skewness of color moments are as equations 1, 2, and 3, respectively [13].

$$E_i = \frac{1}{N}\sum_{j=1}^{N} p_{i,j} \quad (1)$$

$$\sigma_i = \sqrt{\frac{1}{N}\sum_{j=1}^{N}\left(p_{i,j} - E_i\right)^2} \quad (2)$$

$$\varsigma_i = \sqrt[3]{\frac{1}{N}\sum_{j=1}^{N}\left(p_{i,j} - E_i\right)^3} \quad (3)$$

$p_{i,j}$ represents the current pixel coordinate horizontally and vertically, and the total number of pixels is denoted with $N$.

2.4.2. Circularity

Circularity measures how much a form deviates from a perfect circle. It is a quantitative measure that assesses the degree to which the boundary of an object resembles a perfect circle. It is expressed as a value between 0 and 1, where a form is considered an ideal circle if it has a circularity value of 1. Equation 4 represents the circularity model.

$$C = 4\mu A/perimeter^2 \quad (4)$$

$A$ represents the object's area and $perimeter$ denotes the distance between the object's edge to its center point related to its area [14].

2.4.3. Histogram of Oriented Gradients (HOG)

HOG was initially proposed as an approach for human recognition [15]. This feature descriptor preserves track of the frequency of gradient occurrences, orienting oneself within a detection window. In image processing, using gradients is valuable in terms of features as it can define corners and edges. HOG has three main processes: calculating gradients, orientation binning, and block normalization.

Gradient calculation: the definition of a gradient is the image's first-degree derivatives in both the vertical and horizontal dimensions. The gradient magnitudes and angles are then determined using these gradients as represented in equations 5 and 6 [15].

$$G = \sqrt{(g_x)^2 + \left(g_y\right)^2} \quad (5)$$

$$\theta = \tan^{-1}\left(\frac{g_x}{g_y}\right) \quad (6)$$

$g_x$ and $g_y$ are gradients in X and Y directions.

Orientation binning: after the gradient is calculated, the histogram for each linked block of the image is computed after the image has been split into blocks. Each pixel in a block has its gradient magnitude separated into multiple orientation bins based on the gradient angle. The orientation binning process is illustrated in Figure 4 [16].

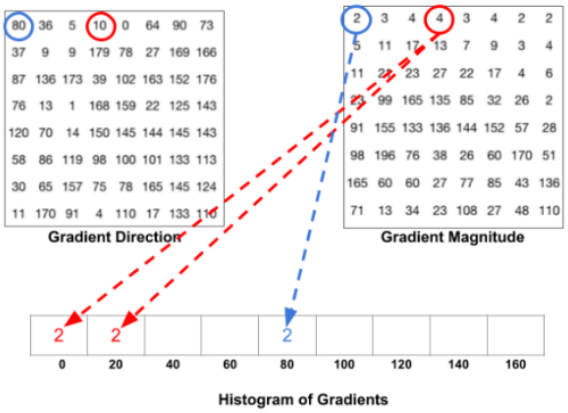


Figure 4. Illustration of orientation binning

In block normalization, adjacent blocks are grouped. Each group is normalized using $L_2$-normalization. The normalized block group outputs a descriptor in histograms [15].

2.5. Dimensionality Reduction

Using the dimensionality-reduction approach, principal component analysis (PCA) aims to decrease computational workload by acquiring a lower-dimensional subspace from high-dimensional data [17]. By measuring the variance of the data, PCA transforms linear data by creating Principal Components (PCs), which are new features. The first component has the most significant variance and the most significant impact on the model, while the following elements have progressively lower variances. [6]. The summarized flowchart of PCA can be observed in Figure 5.

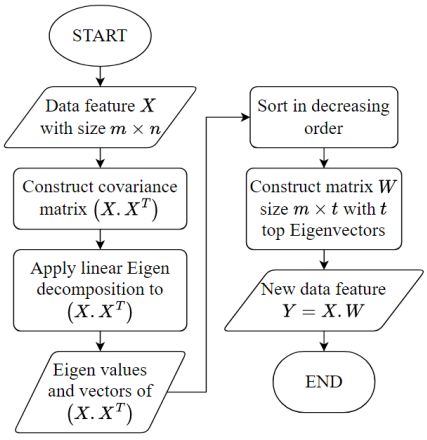


Figure 5. PCA Flowchart [6]

2.6. Classification

In this study, the classification step is conducted by implementing a supervised learning approach of KKN.

DOI: https://doi.org/10.29207/resti.v7i6.4839

KNN classifies data based on how similar the instances are. The process involves classifying a test image. The classification is performed by comparing it to a database of images (training data) and choosing the most common class among the test image's closest neighbors in the feature space [18]. KNN works on the fundamental premise that the image most similar to a provided image feature space is its 'nearest neighbor.' A hyperparameter that the user can specify is the number of closest neighbors or $k$. The test image's class is then determined by a vote of its $k$ closest neighbors as observed in Figure 6.

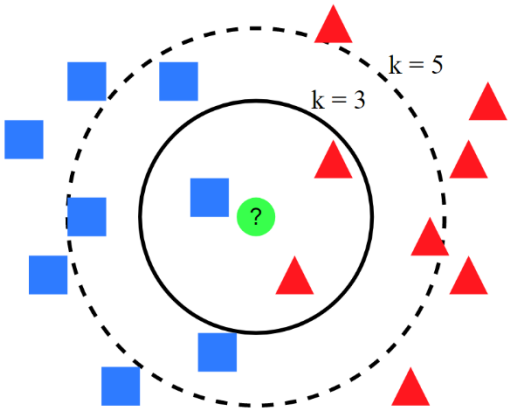


Figure 6. KNN illustration

KNN uses a distance metric to calculate the closest neighbors to the current agent, namely the Euclidian distance, Manhattan distance, or Minkowski distance, to assess the image similarity. Equations 7, 8, and 9 show the metric distance models [19].

$$d(p,q) = \sqrt{\sum_{i=1}^{N}(q_i - p_i)^2} \quad (7)$$

$$d(p,q) = \sum_{i=1}^{N}|q_i - p_i| \quad (8)$$

$$d(p,q) = \sqrt[j]{\sum_{i=1}^{N}(|q_i - p_i|)^j} \quad (9)$$

$p$ and $q$ are neighbors that the distance is calculated and $N$ represents the total amount of data.

### 2.7. Optimization

Optimizing the KNN classifier can be implemented to resolve various optimization problems on different portions of the model contingent on the objective of the optimization. BPSO can be exploited to amend factors influencing the KNN model's performance, such as the optimal number of neighbors. BPSO is designed to resolve binary or discrete problems, deeming it suitable for KNN optimization. In this study, BPSO, as a variant implementation of PSO, will be modified to resolve the problem. A swarm in PSO consists of a collection of particles, each represented by a feature vector, within a multi-dimensional search space [20]. With every subsequent iteration, the position of every randomly initialized particle or agent advances closer to the ideal position with the best fitness [21]. The particle's 'velocity' is modified to produce this movement by considering its position and current velocity, each individual's best position, and the best overall position of the rest of the initialized particles (global best).

BPSO implements a similar structural implementation as a regular PSO. The sole differences are the particle's position representation and how the particle's velocity influences its position. With the length of the input binary that represents the maximum value of $k$, BPSO as $k$ value optimization (BPSO-kv) initializes its particle position with a random continuous value between 0 and 1, in which values over 0.5 are treated as bit 1 and otherwise as bit 0 [22]. The particle's initial velocity is further determined with a random value within the range of (-1, 1) with input binary length. BPSO as $k$ value optimization can be summarized as shown in Figure 7. The change in velocity utilizes the formula shown in Equation 10, and updating each particle's position utilizes Equation 11 [21].

$$V_i(t+1) = wV_i(t) + c_1 r_1\left(S_{ibest} - S_i(t)\right) + c_2 r_2(S_{gbest} - S_i(t)) \quad (10)$$

$$S_i(t+1) = S_i(t) + V_i(t+1) \quad (11)$$

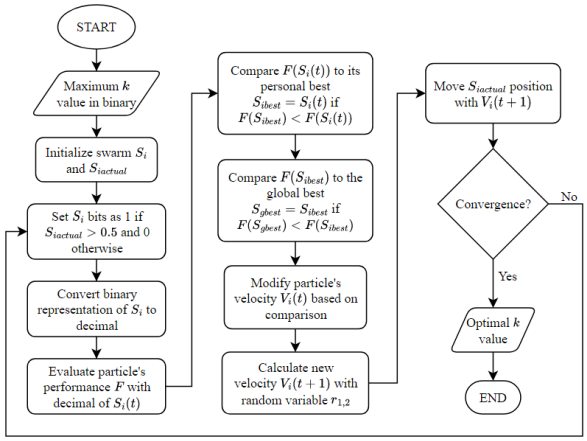


Figure 7. BPSO-kv Flowchart

BPSO as $k$ value optimization (BPSO-kv) implements a set length of binary values representing the maximum k value used in KNN and converts the binary position in $S_i$ to decimal value for a discrete number of neighbors.

### 2.8. Performance Evaluation

In the performance evaluation stage, several metrics are considered to determine whether the resulting model is satisfactory. A confusion matrix obtains accuracy rate, prediction precision, recall, and the F1-score.

Accuracy estimates how competently a model can predict the correct output for a given input. It is calculated as the sum of accurate predictions forecasted

DOI: https://doi.org/10.29207/resti.v7i6.4839

by the model divided by all predictions conducted, as shown in Equation 12.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (12)$$

Precision measures the proportion of correctly predicted instances in a class out of all the model's predictions for a class. Precision is calculated with equation 13.

$$Precision = \frac{TP}{TP+FP} \quad (13)$$

The recall is the probability of the prediction that is correctly forecasted. Recall is measured using equation 14.

$$Recall = \frac{TP}{TP+FN} \quad (14)$$

The F1-score is the calculated average between the model's accuracy and recall score. The F1 score measures the model's overall performance, with a higher score signifying more excellent performance. F1-score is calculated with equation 15.

$$F1 - score = 2 * \frac{Precision*Recall}{Precision+Recall} \quad (15)$$

## 3. Results and Discussions

Five distinctive date fruit classes—Ajwa, Galaxy, Shaishe, Nabtat Ali, and Sugaey—comprise the dataset employed in this study. Initially, each image is resized to 128x128 pixels to reduce the computational time. A total of 650 samples shall be utilized in the modeling phase of the dataset. The image instances are subsequently divided into two distinct groups, namely data training and testing, with a ratio of 4:1, respectively. Additionally, the data is split equally for each class. The data train consists of 520 images, whereas the data test consists of 26 images per class, resulting in a total of 130 images used in the training phase.

It is ensured that the initial images in the training and testing datasets are within the RGB color space. The process of extracting a color characteristic consists of two steps, the foremost within the RGB color space and the subsequent in the HSV color space. RGB and HSV are computed for every pixel in each color channel employing color moments. Color moments calculate color distributions in an image. Several types of color moment metrics can be calculated, including mean, variance, and skewness. The mean represents the arithmetic average of the color intensities in the image, while the variance measures the spread of the colors around the mean. On the other hand, skewness quantifies the degree of asymmetry in the color distribution. Consequently, there are 3x3x2 data features overall, with mean, variance, and skewness for each channel of RGB and HSV.

The image's shape feature is quantified by applying a binary thresholding technique to the images. This involved setting a threshold value and converting the pixel intensities in the image to either black or white, contingent on whether they were above or below the threshold. Additionally, image segmentation made binary thresholding easier and shape assessment more accurate. The resulting binary images were then utilized to calculate the area of the target interest in the images, which allowed the generation of a circular score using the circularity formula, which indicates how circular the object is. The circularity score was calculated within an inclusive range of 0 to 1, with a score of 1 indicating a perfectly circular object. Furthermore, the texture characteristics of the image are extracted by employing HOG.

HOG partitions the image into smaller regions and calculates the histogram of gradient orientation for each region. The gradient orientation histogram characterizes the distribution of intensity gradients within the region and can be used to identify the dominant texture patterns in the image. This experiment applied eight orientations or bins as the hyperparameter for HOG. In this connection, the size of each block is 3x3 cells, whereas the size of each cell is 7x7 pixels. This schema generated a histogram descriptor of the image with a size of (1, 18432). By considering the shape, color, and texture characteristics, the outcome is a dense data feature.

A dense data feature can cause image classification to perform inferiorly. This is due to several common reasons in classification problems, such as the curse of dimensionality, noise, and overfitting. The curse of dimensionality refers to the phenomenon where an increase in the number of data features necessitates a more significant amount of data to maintain a balanced relationship between the features and the target variable. In this connection, increasing data features may introduce noise to the training data, negatively impacting the model's performance. Reducing the data characteristic might reduce the noise level in the data. Therefore, dimensionality reduction is implemented to combat the mentioned problems. Overfitting is caused by oversized data features that make the model overly complex and highly specialized to the training data. Hence, the model may perform poorly on unseen data.

In this research, PCA is strictly used for dimensionality reduction with a total of 18451 features reduced to a set of significantly fewer features, and to create a neutral standpoint, comparing PCA with multiple k values is not essential. From Table 1, PCA performs insignificantly across numerous trials, albeit with similar performances exclusively. The top 50 out of 18,451 features or components will be chosen as they are deemed sufficient for decent classification

DOI: https://doi.org/10.29207/resti.v7i6.4839

performance and temper the problems mentioned before, which are then used for further training.

Table 1. PCA comparisons for K = 7

| Component | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|
| 10 | 81.54% | 80.99% | 81.54% | 81.02% |
| 20 | 83.85% | 84.14% | 83.85% | 83.97% |
| 30 | 79.23% | 78.62% | 79.23% | 78.69% |
| 40 | 83.08% | 84.95% | 80.08% | 82.48% |
| 50 | 85.38% | 84.98% | 85.38% | 85.06% |
| 60 | 83.85% | 83.94% | 83.85% | 83.69% |

BPSO as $k$ value optimization works similarly to how regular PSO is designed. BPSO initializes the swarm with a set number of particles and designates each actual position with random values between 0 and 1. Additionally, each particle is designated a random velocity.

The values of actual positions are set as bit 1 if above 0.5 and bit 0 otherwise and set as the binary representation of the particle's position. This is rooted in the binary implementation of PSO. Then, every particle evaluates its position's $k$ value, converted from binary to decimal for KNN performance,and then seeks a new position by analyzing the alteration in the particle's velocity. This swarm explores the provided search space of potential $k$ values until a set number of iterations. For KNN optimization employing BPSO, the hyperparameters of BPSO are as follows; 100 particles, 50 iterations, $c1, c2 = 2,$, and an inertia weight of 0.5.

In this experiment, the KNN with only PCA and not optimized by BPSO is compared with the barebone KNN. For comparison matters, the $k$ value for KNN without utilizing BPSO are 3, 5, and 7. Finally, a confusion matrix assesses the model's performance by its accuracy rate, F1 score, recall, and prediction precision.

From Tables 2, 3, and 4, the accuracy of the barebone KNN algorithm classification model using K = 3, 5, and 7 as the hyperparameter attained 81.85%, 84.62%, and 85.38%, respectively. KNN classifier with reduced feature dimensions using PCA, with 3, 5, and 7 neighbors, resulted in a lower accuracy performance of 81.54%, 83.08%, and 79.23%. This occurred because PCA significantly reduced the number of features required to feed the KNN model, granted that with a considerable feature reduction, only a maximum of 6.15% decrease in accuracy, as shown in Table 4.

Table 2. Performance comparison for K = 3

| Model | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|
| KNN | 81.85% | 81.35% | 81.54% | 81.43% |
| KNN+PCA | 81.54% | 81.15% | 81.54% | 81.24% |

Table 3. Performance comparison for K = 5

| Model | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|
| KNN | 84.62% | 84.64% | 84.62% | 84.53% |
| KNN+PCA | 83.08% | 83.07% | 83.08% | 81.89% |

Table 4. Performance comparison for K = 7

| Model | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|
| KNN | 85.38% | 85.51% | 85.38% | 85.07% |
| KNN+PCA | 79.23% | 79.42% | 79.23% | 79.11% |

Table  shows that the optimized KNN model by implementing both PCA and BPSO improved overall performance compared to KNN without PCA and BPSO. BPSO generated 20 neighbors as the optimal number for the KNN model, producing an accuracy of 93.85%. Also, the proposed model's precision and recall performance are on par with accuracy. This signifies the proposed model's high capability to predict which class of date fruit is correct and the probability of correctly classifying a date fruit's class across all five date fruit types. Furthermore, since the proposed model's precision and recall are on par with accuracy, the F1-score is additionally on par with the accuracy.

Table 5. Proposed optimized model performance for K = 20

| Model | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|
| KNN | 82.31% | 82.23% | 82.31% | 81.83% |
| KNN+PCA | 81.54% | 82.33% | 81.54% | 81.54% |
| KNN+PCA +BPSO-kv | 93.85% | 93.84% | 93.85% | 93.69% |

The experiment results demonstrate that employing BPSO to optimize the $k$ value for KNN yields the best performance with an optimal number of neighbors. Using the BPSO algorithm, we found novel and frequently ignored solutions in this study. By limiting the search space to a maximum k value of 127 (1111111 in binary), the algorithm's exploratory nature allowed it to uncover original solutions while effectively managing computational resources quickly. As a result of the particles' exploratory nature, every particle was compelled to follow the leading best particle. This strategy encourages exploration while the computational costs of thorough solution exploration are reduced.

## 4. Conclusion

This study's findings show that BPSO successfully maximizes the number of neighbors utilized in the K-Nearest Neighbor classifier for image classification. KNN can produce a promising predictive model with an accuracy of 93.85% by using BPSO to discover the optimal $k$ value. Furthermore, employing PCA for dimensionality reduction proves that with fewer features, it can perform similarly while decreasing the computational cost. The experimental results imply that BPSO may also benefit or enhance KNN performance in other occurrences. Additionally, this study emphasizes the importance of correctly choosing the $k$ value in KNN, as it may significantly impact the model's performance. Our findings additionally show that color, shape, and texture features can be used to obtain a decent model to classify date fruits with an accuracy above 80% without dimensionality reduction.

DOI: https://doi.org/10.29207/resti.v7i6.4839

A classification model with more date fruit types will be investigated for future work by including data features while adding additional date fruit classes for a larger dataset.

## References

[1] Food and A. O. of the United Nations, "Food and Agriculture Commodities Production 2020." Accessed: May 18, 2022. [Online]. Available: https://www.fao.org/faostat/en/#rankings/countries_by_commodity/

[2] A. Nasiri, A. Taheri-Garavand, and Y.-D. Zhang, "Image-based deep learning automated sorting of date fruit," *Postharvest Biol. Technol.*, vol. 153, pp. 133–141, Jul. 2019, doi: 10.1016/j.postharvbio.2019.04.003.

[3] X. Liu, D. Zhao, W. Jia, W. Ji, and Y. Sun, "A Detection Method for Apple Fruits Based on Color and Shape Features," *IEEE Access*, vol. 7, pp. 67923–67933, 2019, doi: 10.1109/ACCESS.2019.2918313.

[4] Fernanda Januar Pratama, Wikky Fawwaz Al Maki, and Febryanti Sthevanie, "Big Cats Classification Based on Body Covering," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 5, pp. 984–991, Oct. 2021, doi: 10.29207/resti.v5i5.3328.

[5] A. Wang, W. Zhang, and X. Wei, "A review on weed detection using ground-based machine vision and image processing techniques," *Comput. Electron. Agric.*, vol. 158, pp. 226–240, Mar. 2019, doi: 10.1016/j.compag.2019.02.005.

[6] F. Anowar, S. Sadaoui, and B. Selim, "Conceptual and empirical comparison of dimensionality reduction algorithms (PCA, KPCA, LDA, MDS, SVD, LLE, ISOMAP, LE, ICA, t-SNE)," *Computer Science Review*, vol. 40. Elsevier Ireland Ltd, May 01, 2021. doi: 10.1016/j.cosrev.2021.100378.

[7] T. Tri Saputra Sibarani and C. Author, "Analysis K-Nearest Neighbors (KNN) in Identifying Tuberculosis Disease (Tb) By Utilizing Hog Feature Extraction," *Int. Comput. Sci. Inf. Technol. JournalISSN*, vol. 1, no. 1, pp. 33–38, 2020.

[8] L. Kumar and K. K. Bharti, "An improved BPSO algorithm for feature selection," in *Lecture Notes in Electrical Engineering*, Springer Verlag, 2019, pp. 505–513. doi: 10.1007/978-981-13-2685-1_48.

[9] A. K. Mishra, P. Roy, and S. Bandyopadhyay, "Binary Particle Swarm Optimization Based Feature Selection (BPSO-FS) for Improving Breast Cancer Prediction," 2021, pp. 373–384. doi: 10.1007/978-981-15-4992-2_35.

[10] W. S. N. Alhamdan and J. M. Howe, "Date Fruit Image Dataset in Controlled Environment." Accessed: May 27, 2022. [Online]. Available: https://www.kaggle.com/datasets/wadhasnalhamdan/date-fruit-image-dataset-in-controlled-environment

[11] Z. Wang, E. Wang, and Y. Zhu, "Image segmentation evaluation: a survey of methods," *Artif. Intell. Rev.*, vol. 53, no. 8, pp. 5637–5674, Dec. 2020, doi: 10.1007/s10462-020-09830-9.

[12] A. Bhargava and A. Bansal, "Classification and Grading of Multiple Varieties of Apple Fruit," *Food Anal. Methods*, vol. 14, no. 7, pp. 1359–1368, Jul. 2021, doi: 10.1007/s12161-021-01970-0.

[13] X. Zenggang, T. Zhiwen, C. Xiaowen, Z. Xue-min, Z. Kaibin, and Y. Conghuan, "Research on Image Retrieval Algorithm Based on Combination of Color and Shape Features," *J. Signal Process. Syst.*, vol. 93, no. 2–3, pp. 139–146, Mar. 2021, doi: 10.1007/s11265-019-01508-y.

[14] P. U. Riswana, "Extract Circular Object by tracing Region Boundary and using Circularity Measure," *Int. Res. J. Eng. Technol.*, 2019, [Online]. Available: www.irjet.net

[15] W. Zhou, S. Gao, L. Zhang, and X. Lou, "Histogram of Oriented Gradients Feature Extraction from Raw Bayer Pattern Images," *IEEE Trans. Circuits Syst. II Express Briefs*, vol. 67, no. 5, pp. 946–950, May 2020, doi: 10.1109/TCSII.2020.2980557.

[16] Rismiyati and H. A. Wibawa, "Snake Fruit Classification by Using Histogram of Oriented Gradient Feature and Extreme Learning Machine," in *2019 3rd International Conference on Informatics and Computational Sciences (ICICoS)*, IEEE, Oct. 2019, pp. 1–5. doi: 10.1109/ICICoS48119.2019.8982528.

[17] G. T. Reddy *et al.*, "Analysis of Dimensionality Reduction Techniques on Big Data," *IEEE Access*, vol. 8, pp. 54776–54788, 2020, doi: 10.1109/ACCESS.2020.2980942.

[18] E. Hossain, M. F. Hossain, and M. A. Rahaman, "A Color and Texture Based Approach for the Detection and Classification of Plant Leaf Disease Using KNN Classifier," in *2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, IEEE, Feb. 2019, pp. 1–6. doi: 10.1109/ECACE.2019.8679247.

[19] Haviluddin *et al.*, "A Performance Comparison of Euclidean, Manhattan and Minkowski Distances in K-Means Clustering," in *2020 6th International Conference on Science in Information Technology: Embracing Industry 4.0: Towards Innovation in Disaster Management, ICSITech 2020*, Institute of Electrical and Electronics Engineers Inc., Oct. 2020, pp. 184–188. doi: 10.1109/ICSITech49800.2020.9392053.

[20] R. K. Chaurasiya, M. I. Khan, D. Karanjgaokar, and B. K. Prasanna, "BPSO-Based Feature Selection for Precise Class Labeling of Diabetic Retinopathy Images," 2020, pp. 253–264. doi: 10.1007/978-981-13-8196-6_24.

[21] V. P. Kour and S. Arora, "Particle Swarm Optimization Based Support Vector Machine (P-SVM) for the Segmentation and Classification of Plants," *IEEE Access*, vol. 7, pp. 29374–29385, 2019, doi: 10.1109/ACCESS.2019.2901900.

[22] Mojtaba Ahmadieh Khanesar, Mohammad Teshnehlab, and Mahdi Aliyari Shoorehdeli, "A novel binary particle swarm optimization," in *2007 Mediterranean Conference on Control & Automation*, IEEE, Jun. 2007, pp. 1–6. doi: 10.1109/MED.2007.4433821.

DOI: https://doi.org/10.29207/resti.v7i6.4839