

RESEARCH ARTICLE

A Two-Stage Feature Selection Approach for Fruit Recognition Using Camera Images With Various Machine Learning Classifiers

TRI TRAN MINH HUYNH¹, TUAN MINH LE¹, LONG TON THAT¹, LY VAN TRAN²,
AND SON VU TRUONG DAO^{2,3}

¹School of Electrical Engineering, International University, Vietnam National University Ho Chi Minh City, HCMC, Ho Chi Minh City 70000, Vietnam

²School of Industrial Engineering and Management, International University, Vietnam National University Ho Chi Minh City, HCMC, Ho Chi Minh City 70000, Vietnam

³School of Science, Engineering and Technology, RMIT University Vietnam, Ho Chi Minh City 70000, Vietnam

Corresponding author: Son Vu Truong Dao (dvtson@hcmiu.edu.vn; son.daovutruong@rmit.edu.vn)

This work was supported by International University, VNU-HCM under Grant T2021-02-IEM.

ABSTRACT Fruit and vegetable identification and classification system is always necessary and advantageous for the agriculture business, the food processing sector, as well as the convenience shops and supermarkets where these products are sold. Therefore, it is necessary to build an effective automated tool to meet the needs of the market by boosting the outcome, in order to improve economic efficiency. In this paper, a two-stage model is proposed to recognize fruits using camera images. We employed a Densnet121 to get the features from the fruits dataset in the first module. In the second stage, we utilize a feature subset selection method to choose the most significant features for recognizing fruits from the images of the fruits. In this study, Adaptive particle - Grey Wolf Optimization (APGWO) has been applied for choosing the most pertinent features. The final subset feature has been used for recognizing fruits using several machine learning classifiers, namely K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF), and Multilayer Perceptron (MLP). The proposed research's experimental results are highly effective; the training time of proposed models is reduced to over 50%, and the classification accuracy reaches 99%.

INDEX TERMS Wrapper, feature selection, adaptive particle grey wolf optimization, feature extraction, fruits recognition.

I. INTRODUCTION

Fruit is very important to our body. It contains various vitamins and minerals and is recommended to be eaten after meals [1]. Due to these reasons, the development in the food industry is escalating to increase. Technology, in Parallel, is also developing to meet the required demand so that automation increases and operating costs of factories are significantly reduced. With the development of CNN, it is easier and easier to classify types of pests, diagnose diseases, and determine the maturity of food. Especially during the covid pandemic, human resources are decreasing, and the demand for food is high, leading to automation in the factory.

The associate editor coordinating the review of this manuscript and approving it for publication was Pasquale De Meo.

With the advancement in the fields of Image Processing, Computer Vision, and Machine learning, Automation in the Food Industry is no longer a problem when a robot with cameras can replace human at certain processing steps. Deep learning method, to be more specific, deep convolutional neural network is one of the leading approaches in vision-based detection, recognition, and classification for the food industry.

There are several approaches which applied deep learning methods to classify fruits. To be more specific, the owner of a fruit360 dataset developed a Convolutional Neural Network (CNN) with four convolution layers interleaved with four max-pooling layers. The output of these layers is the input of two fully connected layers to generate 256 outputs with the ReLu activation function. The last layer is

the softmax loss layer with 256 inputs. The output of this layer is equal to the number of classes [2]. Their work gained a very high result with 96.3% accuracy in 60 classes. Because of the diversity of data with a large number of samples, this is one of the pioneers in the development of fruit recognition techniques. In addition, Kausar et al. proposed Pure-CNN and used a fruit360 dataset with 81 types of fruits to test and achieved 98.88% accuracy [3]. Up to now, many researchers have modified and developed the CNN models to gain not only better accuracy but also reduce the training time. Since the late 20th century, there are many successful pre-trained CNN models such as LeNet, AlexNet, VGG, etc. Tomar applied the fruit360 dataset with seven transfer learning models, namely InceptionV3, ResNet50, VGG16, MobileNet, xception, DenseNet and InceptionResNet [4]. After comparing the experimental results of these models, they concluded that with 120 classes of the fruit360 dataset, the best model is VGG16, which has 99% and 95% training and testing accuracy, respectively. However, in this research, a total of 131 classes of fruits had been taken into account. Our approach is used to develop a new algorithm that can solve these current issues by combining multiple methods in Artificial Intelligence and computer vision to receive high accuracy but reduce costs. First, we extract the features of the Fruit360 dataset from Densenet121. Then we apply Adaptive Particle Grey Wolf Optimizer to choose the suitable features. These selected features will be trained by well-known models applied in fruits recognition:

A. SUPPORT VECTOR MACHINE (SVM)

a linear model for regression and classification problems. SVM can be used for linear or nonlinear problems, it is useful for many practical applications. SVM creates a line or a hyperplane to divide the data into classes [5]. It has been shown to be an effective approach for high-dimensional space problems. This method is typically utilized in sentiment analysis and classification because of how effectively it handles computing on enormous datasets.

B. K-NEAREST NEIGHBORS (K-NN)

A well-known statistical technique for pattern recognition [6]. The KNN algorithm is simple and effective. The training set is not required for the classifier, and there is no training time complexity. The quantity of documents in the training set directly relates to the computational complexity of KNN classification. Other approaches for dividing the set of samples that have more intersections or overlaps in the class domain are less effective than this method.

C. MULTILAYER PERCEPTRON (MLP)

Multiple nodes are grouped in layers in the MLP classifier. This creates a directed graph with input, hidden, and output layers, all of which are fully connected to each other. This classifier features a straightforward architecture and a straightforward method, making it one of the most well-known neural network models [7], [8], [9].

D. DECISION TREE (DT)

The process of categorizing instances according to features is represented by the decision tree model. Its main advantages are that this model is readable and the classification procedure is fast. When learning, it uses the training data to build a decision tree model according to the principle of minimizing the loss function [10].

E. RANDOM FOREST (RF)

RF is a commonly used ensemble learning algorithm, and the base classifier is a decision tree. Each tree selects randomly variables and observations for classifier construction, and voting is used to determine the outcome. Generally, $\log N$ features are selected for each tree (N is the number of features). If all features are selected for each tree, the random forest at this time can be regarded as a bagging algorithm [11]. This approach has been frequently used to choose the model parameter in landslide susceptibility mapping [12], [13].

The following and preceding sequences are used to organize this research. The literature review is presented in Section II. In addition, Section III has a detailed explanation of the proposed method. Finally, the final work and experimental results are demonstrated in section IV.

II. LITERATURE REVIEWS

A. FEATURE EXTRACTION (FE)

The feature extraction approach is to extract features from the dataset to increase the performance of the learned model. This also helps the classifier to increase the training speed due to reducing the unappropriated features.

The input of the classifier for image classification and segmentation is usually the significant feature of each image. There are three main types of features that can be extracted from an image, shape-based features (area, perimeter, shape index), intensity-based features (statistical parameters such as mean, variance, median), and texture-based features (contrast, correlation, entropy, inertia [14]). In this study, we will use the mean intensity of the intensity-based features.

B. FEATURE SELECTION (FS)

1) AN OVERVIEW OF FEATURE SELECTION

In the feature space, a feature extraction technique may create a few redundant and irrelevant features. It's critical to get rid of any redundant or irrelevant feature properties; it not only improves overall classification accuracy but also reduces computing overheads [15]. The feature selection method is usually used to do this.

The feature selection method is a machine learning procedure to select features to solve a high dimensional-ity problem. To make subsequent analysis easier, we select a number of important features and normally disregard noisy and irrelevant data in terms of redundancy and relevancy. These feature subsets were categorized by Yu et al. [16] into four categories: noisy and irrelevant, redundant and weakly relevant, forcefully relevant, and non-redundant.

Furthermore, filter and wrapper methods can be used to implement a variety of approaches, including search strategies, models, feature evaluation, and feature quality measures. The number of characteristics and the size of the hypothesis spaces are proportionate. As the number of features increased, so did the size of the searching space. If an original dataset contains M features with a binary class label, the search space contains 2^{2^M} Combination. The operating mechanism of the function choice process is depicted in Figure 1. It could be found that there are five essential additives of the process including dataset, selected subset feature, evaluation, selection criteria, and validation.

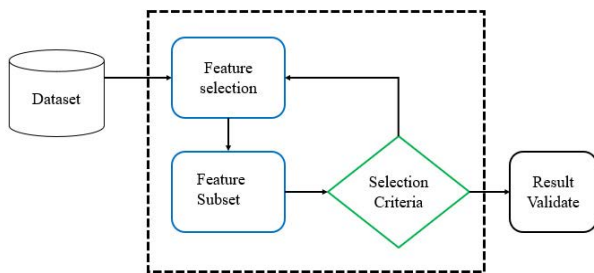


FIGURE 1. Feature subset selection process.

There are three types of feature selection strategies: wrapper, filter, and embedded methods.

A filter method is a selection approach related to the relationship between the objective variable and the explanatory variable that is unrelated to the machine learning method. The correlation coefficient score and the chi-squared test are all examples of filtering processes [17], [18]

The performance of the wrapper approach is determined by the classifier. Try to put a subset of selected features into an algorithm and see how well it performs compared to the prior model before deciding whether or not to include more features [19], [20], [21]

Finally, the embedded technique selects features using hybrid learning methods and ensemble learning. It works similarly to the Wrapper Method; this technique also chooses variables in the machine learning model at the same time. Moreover, the feature quantity is calculated concurrently with model learning, without the need to combine model learning, making it simple to calculate.

The following figure describes a flow chart of technique categorization to solve feature subset selection and explains the way arrived at metaheuristic algorithms.

2) METAHEURISTIC ALGORITHMS

Metaheuristic algorithms are optimization methods that aim to find the best solution to a problem. These are derivative-free techniques that are flexible, simple, and avoid local optima. Because of producing random solutions in their optimization process, this method has a stochastic behavior [22]. The major feature of metaheuristic algorithms is the extraordinary ability to prevent algorithms from converging prematurely. The algorithm has been used to solve a range

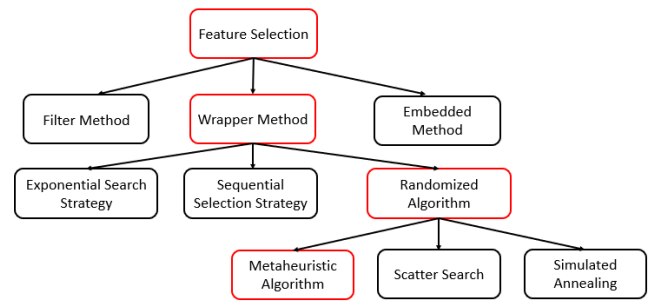


FIGURE 2. An overview of feature selection techniques.

of engineering and science problems with great success, including those in electrical engineering, industrial fields, civil engineering, communication, data mining and so on.

The two primary groups of metaheuristic algorithms are as follows.

a: SINGLE SOLUTION-BASED METAHEURISTIC ALGORITHMS

During the iteration step, the optimization process is modified after applying a single solution. It may lead to local optima trapping and a lack of exploration of the search space

b: POPULATION (MULTIPLE) SOLUTION-BASED METAHEURISTIC ALGORITHMS

Start by generating a population of solutions and then optimizing them. Multiple solutions help each other and have a large search space exploration; thus, the algorithms are useful for avoiding local optima. These algorithms have been created to solve a variety of problems. The metaheuristic algorithms can be classified into four types: swarm intelligence-based, evolution-based, human-related algorithms and physics-based.

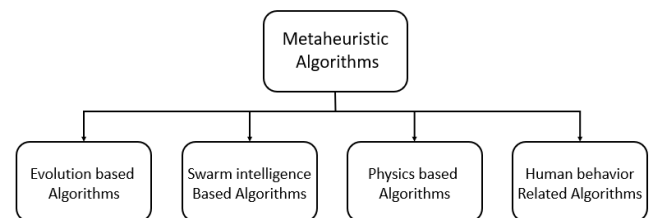


FIGURE 3. Classification of feature selection methods.

c: EVOLUTION-BASED ALGORITHMS

this algorithm starts with a population of randomly generated solutions and is based on the evolution of nature. The best results are combined in these types of algorithms to produce new individuals. There are many popular algorithms such as Genetic algorithm (GA), differential evolution, tabu search, genetic programming, and evolution strategy [22], [23].

d: SWARM INTELLIGENCE-BASED ALGORITHMS

based on the behaviours of animals and birds. Particle Swarm Optimization (PSO) is based on the behaviour of birds, which fly over the search space to find the ideal position [24]. Swarm intelligence techniques include honeybee swarm optimization algorithm, ant colony optimization, monkey optimization, and others.

e: PHYSICS-BASED ALGORITHMS

based on the laws of physics that govern the universe. Physics-based algorithms include Harmony search, simulated annealing, and others [25].

f: HUMAN BEHAVIOUR-RELATED ALGORITHMS

solely based on the behaviour of humans. Every human being has a unique method of carrying out tasks that has an impact on their overall performance. It encourages researchers to work on algorithms [26].

3) HYBRID METAHEURISTIC ALGORITHMS

In order to develop a new and better algorithm, hybrid metaheuristic algorithms combine many different metaheuristic algorithms. Several methodologies have been developed to choose the best relevant features for solving the feature selection problems. The improvement of algorithms helps in avoiding premature convergence, local optima trapping, and efficient search space exploration, as well as in making good exploitation. A greater balance between an algorithm’s exploitation and exploration quality is also achieved by the improved algorithms, further enabling them to produce the optimal solution.

A competitive binary Grey Wolf Optimization (CBGWO) which is based on Mirjalili’s Grey Wolf optimizer was offered by the author to enhance BGWO’s performance and further classify EMG signals [27], [28], [29]. Other wrapper function selections have been proposed to discover the best features, including binary particle swarm optimization (BPSO) [30] and binary differential evolution (BDE) [31]. To initialize the BGWO population, the author introduced the Genetic Algorithms (GA) in [32]. Aside from that, Faris et al. [33] present an overview of GWO research.

III. METHODOLOGY

In this research, we have considered a fruit image database to classify fruits. The process of our approach is shown in Figure 4. At the beginning of our proposed model, we applied Densnet121 for extracting features. Due to the high dimensionality of the feature set, a two-stage feature selection technique is used to pick the most important features. In this stage, the resultant feature set is passed through the Adaptive Particle Grey Wolf Optimization method for the selection, which is based on previous works [34] and [35]. Then, this optimized feature subset is trained by various machine learning models to classify the type of fruits and compare it to

other algorithms without applying the FS method in order to determine which models achieve the best performance.

In this paper, some machine learning models, namely SVM, KNN, MLP, RF, and DT are implemented for evaluating the performance of the system using the feature subset selection method. The detailed working approach for feature extraction using Densnet121 and the two-stage feature selection technique for determining the most effective feature set is shown in the next subsection.

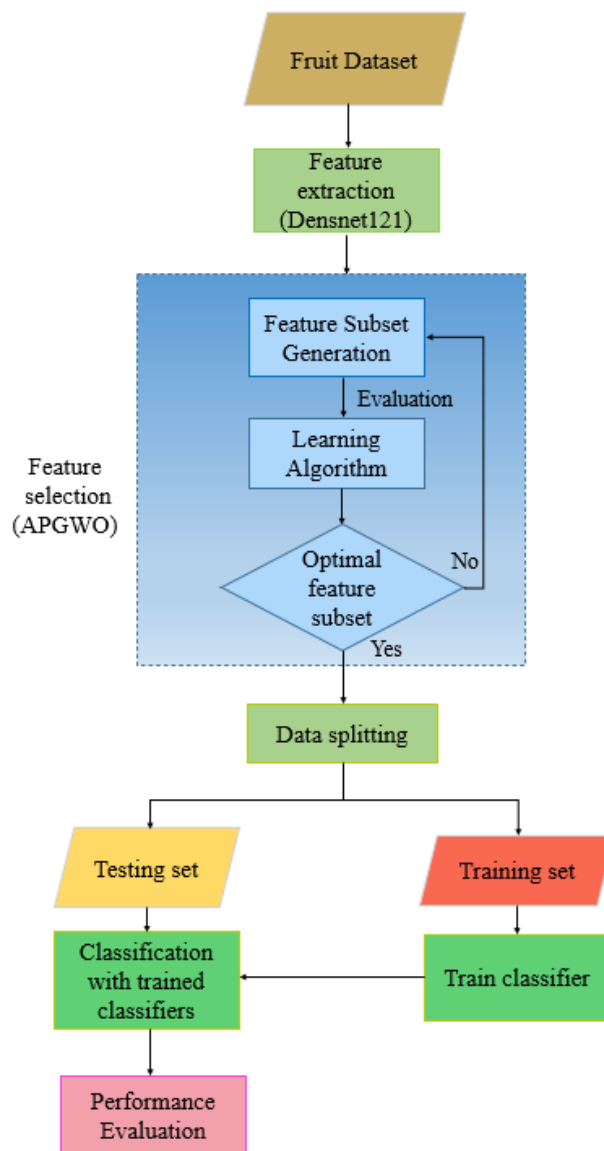


FIGURE 4. The process of our proposed method.

A. DATASET

The fruits dataset is one of the largest datasets up to now. It contains 131 types of fruits with a total of 90,380 pictures [2]. Each fruit was placed in a low-speed rotating system (3rpm) and was recorded for 20 seconds. The white background was the colour of a sheet of paper so that the owner

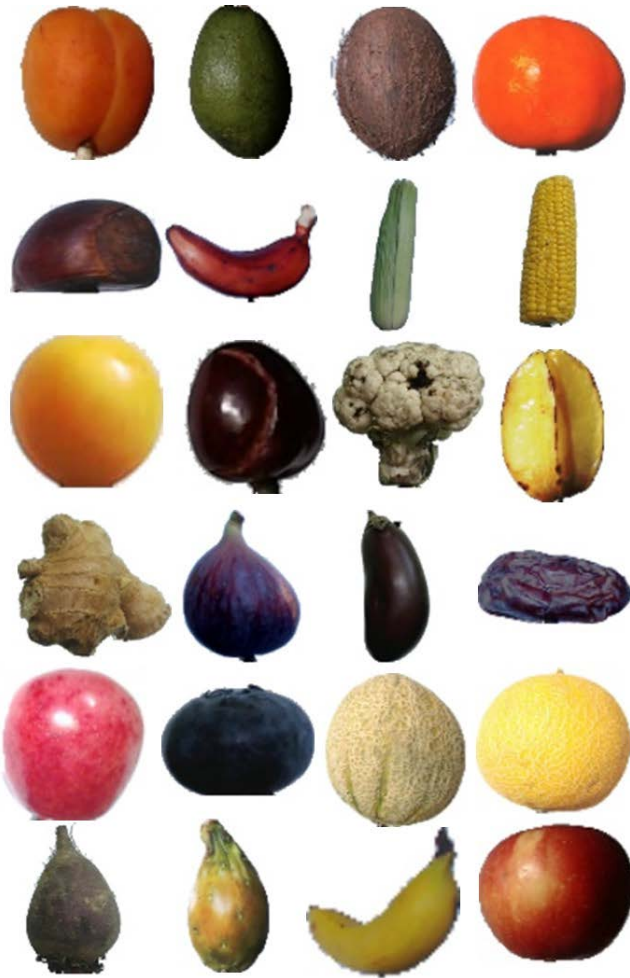


FIGURE 5. Some examples of fruit taken from the Fruit360 dataset.

of this dataset could extract the fruit out of the background. In this paper, we divided the dataset into train and test sets. Figure 5 below shows some examples of fruits, which are taken from the Fruit360 dataset.

B. RESEARCH METHODOLOGY

1) DENSENET121

The entire dataset was used as input for the feature extraction. In this step, two pre-trained models which are DenseNet121 were applied. For the DenseNet121, we extract the features from the 4th dense block. This output contains 1024 features, and each has a dimension of 7 × 7 as shown in Figure 6. Next, all of these features are the input of the features selection method APGWO. The selected features are then taken to the input of machine learning models for the fruit classification.

2) GREY WOLF OPTIMIZATION (GWO)

Individual-to-group communication is referred to as swarm intelligence. The use of herd intelligence has a wide range of applications in research and industry. Herd intelligence research can aid in the management of complicated systems.

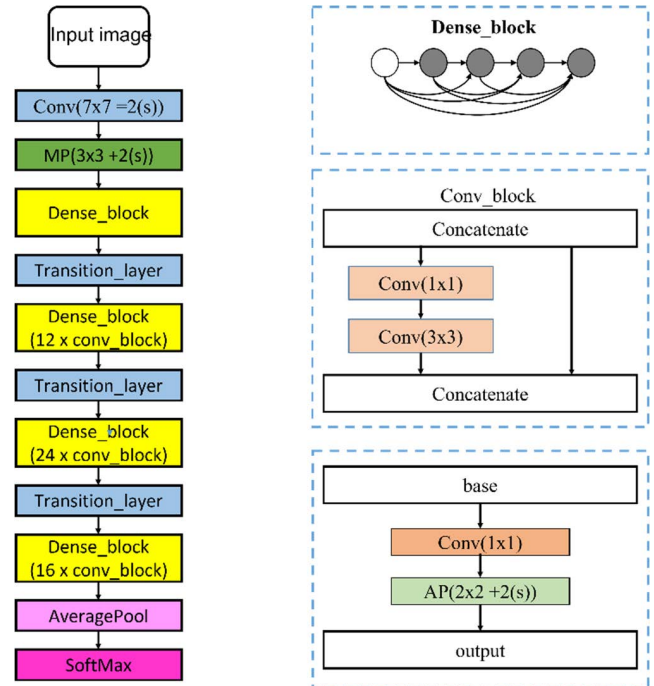


FIGURE 6. Densenet121 architecture.

In order to solve the problem, numerous metaheuristic methods have been developed [22], [23]. In this section, GWO was used to reduce the size of the feature subset. GWO simulates the way how the wolf searches for food and avoids enemies in order to survive. Mirjalili et al. [29], were the first to introduce GWO. Alpha denotes that the leader makes the decision about where to sleep, when to hunt, and when to wake up. The beta is the second level in the pack. These wolves in herds were led by alphas, although they also commanded another wolf. Omega is the lowest rank in the pack. Those wolves are weak and rely on the other wolves. Although delta wolf relies on alpha wolf and beta wolf, they are more responsible than omega ones. These wolves are in charge of monitoring territorial boundaries and issuing warnings inside in the danger case, protecting, ensuring the safety of the herd, and caring for the weak and ill wolves.

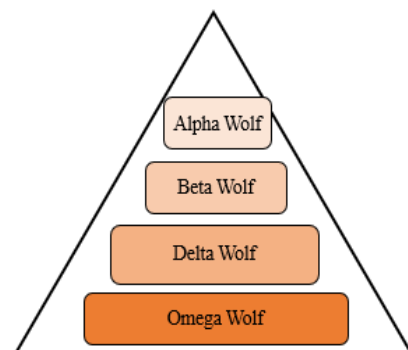


FIGURE 7. The hierarchy of the grey wolf.

The following diagram depicts how an instance in GWO updates its position based on alpha, beta, and delta.

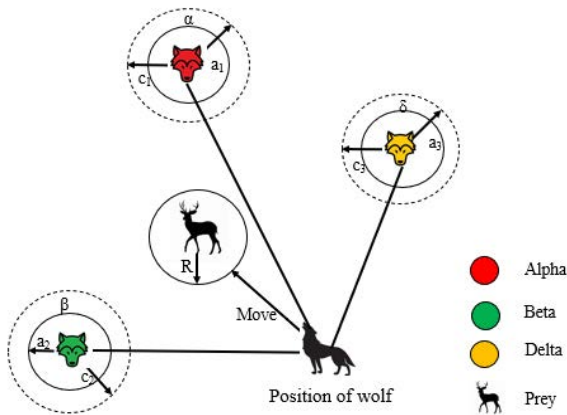


FIGURE 8. Position updating in GWO.

The optimum answer is considered alpha, beta, and delta, correspondingly, for developing the model. GWO’s encircling prey step is shown below:

$$\vec{D} = \left| \vec{C} \cdot \vec{X}_p(t) - \vec{X}(t) \right| \quad (1)$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A} \cdot \vec{D} \quad (2)$$

where t denotes the current iteration, \vec{A} and \vec{C} denote coefficient vectors, \vec{X} denotes a position of a grey wolf vector, \vec{X}_p implies the prey’s position vector. In equations (8) and (9), the coefficient is shown:

$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a} \quad (3)$$

$$\vec{C} = 2\vec{r}_2 \quad (4)$$

where \vec{a} are decreased in the range of $[2,0]$, \vec{r}_1, \vec{r}_2 are random vectors in the range of $[0, 1]$.

These are the steps of hunting. The final position of the wolf $\vec{X}(t+1)$ is indicated as follows:

$$\vec{D}_\alpha = \left| \vec{C}_1 \cdot \vec{X}_\alpha - \vec{X} \right| \quad (5)$$

$$\vec{D}_\beta = \left| \vec{C}_2 \cdot \vec{X}_\beta - \vec{X} \right| \quad (6)$$

$$\vec{D}_\delta = \left| \vec{C}_3 \cdot \vec{X}_\delta - \vec{X} \right| \quad (7)$$

$$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1 \cdot \vec{D}_\alpha \quad (8)$$

$$\vec{X}_2 = \vec{X}_\beta - \vec{A}_2 \cdot \vec{D}_\beta \quad (9)$$

$$\vec{X}_3 = \vec{X}_\delta - \vec{A}_3 \cdot \vec{D}_\delta \quad (10)$$

The grey wolf’s position has described in the following:

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \quad (11)$$

As previously stated, when the prey stops moving, the wolves will attack to finish the hunting. Note that \vec{a} decrease, \vec{A} also decreases. \vec{A} is a value that is random between $-2a$ and $2a$, with a decreasing \vec{a} from 2 to 0 for each iteration. The wolf attacks the prey $|A| < 1$. For monitoring the

exploitation/ exploration trade-off, parameter \vec{a} has been updated, the parameter \vec{a} is adjusted between 2 and 0 in equation (12):

$$\vec{a} = 2 - t * \frac{2}{Maxiter} \quad (12)$$

where t denotes the number of iterations and $Maxiter$ denotes the total number of iterations.

The subset of features in the solution representation is revealed as the first step in resolving the challenge of GWO’s feature subset selection. The location of the solution can be a “1” or a “0.” If the value is 0, the feature is not selected; however, if the value is 1, the feature is chosen.

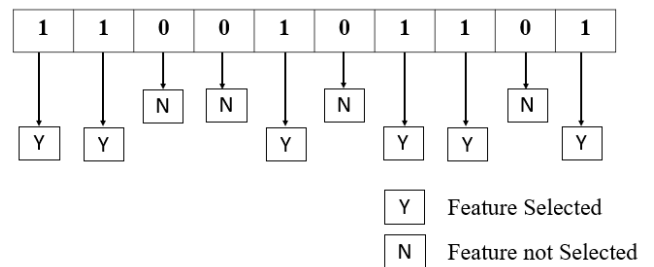


FIGURE 9. An example solution of a feature subset selection.

As described by Emary et al. [28], the GWO technique needs to be updated for binary feature selection. The following equation is a description of the BGWO2 model:

$$x_d^{t+1} = \begin{cases} 1 & \text{if } sigmoid\left(\frac{x_1 + x_2 + x_3}{3}\right) \geq rand \\ 0 & \text{otherwise} \end{cases} \quad (13)$$

3) ADAPTIVE PARTICLE GREY WOLF OPTIMIZATION (APGWO)

Similarly, Eberhart and Kennedy present a theory of Particle Swam Optimization [24] regarding the herd’s prey-hunting strategy in an environment where each animal in the pack is aware of its location in relation to the food and which position is closest to it. To find the nearest way to food, each member of the herd has to follow their leaders who are closest to the prey. The authors present a PSO method that can be used to adapt to this circumstance and solve optimization problems. The two factors that characterize each component of the PSO are the element’s current position - x - and velocity - v . At the same time, the fitness function calculates a fitness value for each part. At the time of departure, the position of each element is stated at random. Each aspect is affected by two parameters of position: the best one is pBest and the other that is held in the whole flock is gBest. The PSO’s elements will navigate the issue space by following the features that are currently the best. After each step, the velocity and position of each component are determined by the equations below:

$$v_k^{t+1} = w * v_k^t + c_1^t * rand * (pbest_k^t - x_k^t) + c_2^t * rand * (gBest - x_k^t) \quad (14)$$

$$x_k^{t+1} = x_k^t + v_k^t \tag{15}$$

The values of c_1 and c_2 are usually specified as constants in PSO to balance the exploration phases, most likely to $c_1 = c_2 = 1$ or $c_1 = c_2 = 2$. In each iteration, a formula is used to change the acceleration coefficients. Equations (16) and (17) contain the new coefficients:

$$c_1^t = 1.2 - \frac{f(x_k^t)}{f(gBest)} \tag{16}$$

$$c_2^t = 0.5 + \frac{f(x_k^t)}{f(gBest)} \tag{17}$$

where:

- $f(gBest)$: The global best fitness of the swarm
- k : The coefficients and the fitness
- t : Iteration

The values of 1.2 and 0.5 have also been discovered through empirical studies. The following is an example of the inertia formula:

Algorithm 1 The pseudocode of APGWO [36]

```

Initialize the particle population
Initialize parameters
Define t = 0
while (t < Max iteration)
  for each particle with position xp
    calculate the fitness value f(xp)
    if fitness f(xp) is better than fitness (pbestp)then
      pbestp ← xp
    endif
    if best f(pbestp) is better than gbest then
      gbest* ← pbestp
    endif
  end for
  update w by using (26)
  for each particle with position xp
    update c1, c2 by (24), (25)
    calculate velocity of each particle by (27)
    update position of each particle by (28)
  end for
  if rand (0,1) < probab
    run GWO
    Xp = position of the best wolf
  endif
  t = t + 1
end while
return gbest

```

$$w_t = (maxIter - t) * \frac{wMax - wMin}{maxIter} + wMin \tag{18}$$

The sigmoid function is shown in equation (19):

$$v_{ij}^t(t) = sig(v_{ij}(t)) = \frac{1}{1 + e^{-v_{ij}(t)}} \tag{19}$$

The equation (20) shows the improvement of the particle's position:

$$x_{ij}(t + 1) = \begin{cases} 1, & \text{if } r_{ij} < sig(v_{ij}(t + 1)) \\ 0, & \text{otherwise.} \end{cases} \tag{20}$$

where ij parameter has a value in the range of 0 to 1.

In the PSO main loop, a few GWO iterations reflect the chance of mutation that will result in a hybrid variety. The likelihood of a mutation in our case study is set to 0.1. The inner loop is only triggered a few times because this value is small, which ensures that the quality of the solution is unaffected.

The pseudo-code of APGWO is shown in the algorithm below:

For the APGWO solution, it is an array that has a size of $1 \times n$ (n represents for all features) and contains binary numbers; 1 for the selected feature and 0 for the rejected ones. The following criteria have been established for the wrapper algorithms: 50 iterations for the PSO loop, 20 search agents for the nested GWO loop, and 20 search agents for the PSO main loop. According to [37], initial weight and final weight are denoted by w_max and w_min , respectively. The value of 0.9 is set as w_max and 0.2 for the w_min . The method aims to reduce the fitness function that is as follows:

$$Minimize \alpha \times E_t + (1 - \alpha) \times \frac{S}{L} \tag{21}$$

where E_t is the validation set's error rate, $\alpha = 0.9$, S and L are the number of chosen features, and the total number of features, respectively. Attempt is made to simultaneously optimize this fitness function and reduce the number of selected features to improve validation accuracy. The optimizer concentrates more on increasing validation accuracy when the α value is high.

The following figure represents the flow chart of APGWO algorithms:

TABLE 1. parameter setting of APGWO.

ID	Parameter	Value
1	Max iteration	20
2	Agents	10
3	dim	Number of data
4	beta	0.01
5	alpha	1 - beta
6	wMax	0.9
7	wMin	0.2

IV. EXPERIMENTAL RESULTS

This section discusses and compares the result of using feature selection methods Adaptive Particle - Grey Wolf Optimization to optimize several machine learning models. The system's outcomes and performance evaluation are presented in the following subsections.

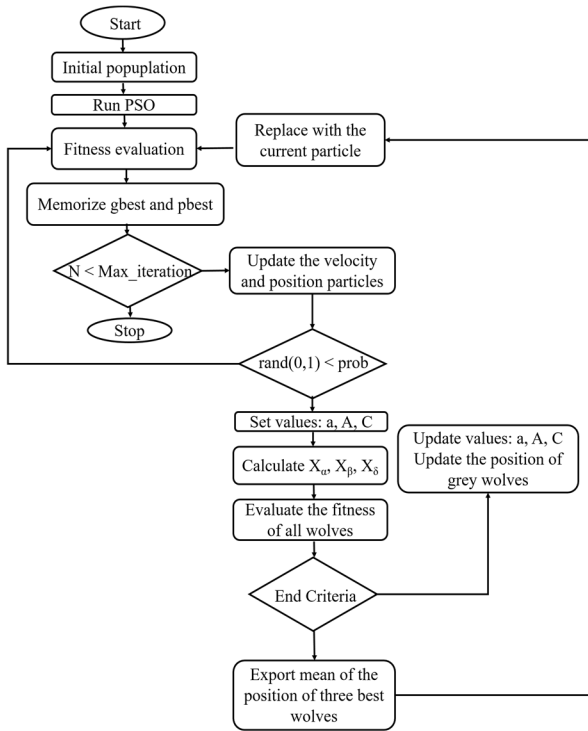


FIGURE 10. Flowchart of APGWO [38].

The dataset is separated into 80:20 in this system, where 80% is used to train, and 20% is used to test the accuracy of these algorithms

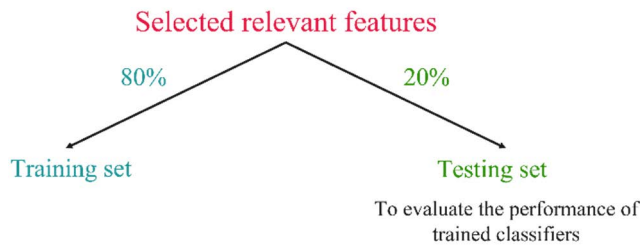


FIGURE 11. The ratio between training and testing set.

A. FEATURE EXTRACTION

Each of the extracted image features is treated as a numerical matrix with the dimension of $m \times n$. Next, the average value of each matrix is calculated by the following formula.

Assume that a matrix $S(m \times n)$ has the form

$$S = \begin{bmatrix} a_{11} & \cdots & a_{m1} \\ \vdots & \ddots & \vdots \\ a_{1n} & \cdots & a_{mn} \end{bmatrix} \quad (22)$$

The mean of S is given by:

$$mean = \frac{1}{m \times n} \sum_{i=1}^m \sum_{j=1}^n a_{ij} \quad (23)$$

The mean value of the feature image represents one feature. Therefore, there are 90380×1024 , and 90380×512 mean feature values for Densenet121 and VGG16, respectively.

B. RESULTS OF USING MODELS WITH ALL FEATURES

Five machine learning models, namely KNN, SVM, DT, RF, and MLP, were applied to train all available features. The following figure shows the accuracy comparison between algorithms. KNN, SVM, and MLP reach the highest performance with 99.99%, as shown in the figure below.

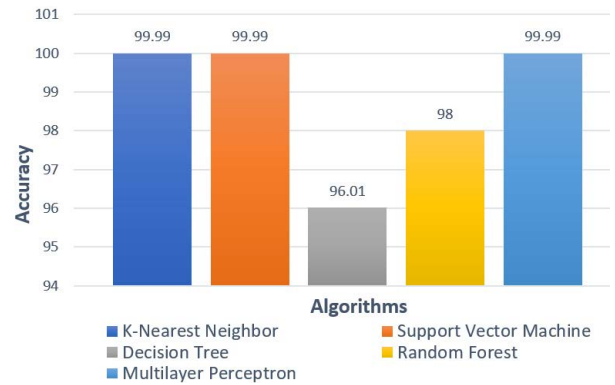


FIGURE 12. The accuracy of all machine learning algorithms using APGWO with all available features.

C. RESULTS OF USING MODELS WITH SELECTED FEATURES

After utilizing the APGWO method, 508 from 1024 features of Densenet121 are selected. Thus the feature space is reduced by 50.4%. Figure 13 demonstrates the number of all available features and the selected features after using the feature subset selection method.

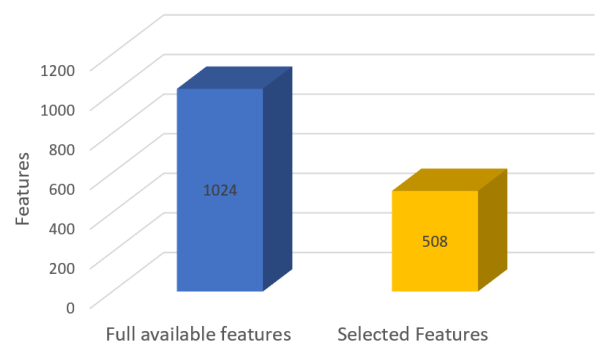


FIGURE 13. The comparison between the number of full features and selected features after using APGWO.

These selected features become the input of several machine learning models, and 50 epochs have been considered in this step. The following figure shows the proposed approach returned the highest accuracy of KNN, SVM, and MLP with 99.99%. While the classification accuracy of the SVM and DT also achieve high performance with 97.46% and 91.09%, respectively.

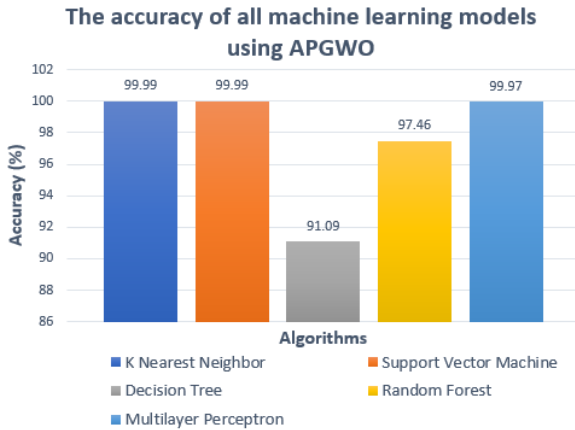


FIGURE 14. The machine learning models' accuracy.

D. PERFORMANCE

After using a wrapper feature selection method to reduce the feature space of the fruits dataset, we apply various models to classify the difference between fruits. The following table presents the accuracy performance of models between full available features and selected features.

With full features, there is the highest position of three machine learning models, namely KNN, SVM, and MLP with 99.99%. Meanwhile, it is obvious that the proportion of Decision Tree accounts for the lowest with around 98%. In this system, APGWO is applied to enhance the performance of machine learning models. It can be seen that 508 features out of 1024 of the original are selected for the first time, and the accuracy of KNN and MLP are higher than the other models, which is indicated that there are un-useful features in the fruits dataset. RF and MLP are two models that remain the high percentage with 97.46% and 99.97%, respectively. While DT only achieves 91.04%.

The experimental results obtained and shown in the table below describe machine learning algorithms combined with the APGWO method that outperform other models without applying the feature subset selection method. These experimental results also shows that our model maintains high accuracy.

Moreover, we run our method several times in order to achieve higher experimental results and enhance our proposed model. It is clear that KNN and SVM perform most accurately, reaching the first and second positions, respectively. As a result, we can observe that KNN and RF are two models that perform well. To save time and get the best results, we recommend future studies to employ both models.

The figure below shows the five-time trials of our proposed models. Five trials are performed in this research to validate the stability of machine learning models, with the initial results from the proposed method shown previously corresponding to Trial 1. It can be seen that the accuracy score remains the same across five trials.

The Table 3 describes the training time of all machine learning models with all features and selected with the

TABLE 2. Accuracy performance of machine learning algorithms with full features and selected features.

Feature Models	Full set	508	528	533	518	535
KNN	99.9	100	100	100	100	100
SVM	99.9	100	100	100	100	100
DT	96	91	91.4	91	91.2	91.1
RF	98	97	97.5	97.5	97.5	97.5
MLP	99.9	99.9	99.9	99.9	99.9	99.9

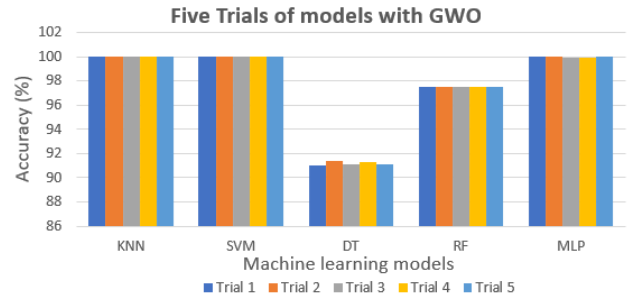


FIGURE 15. Several trials of models using APGWO.

feature selection method - APGWO. In addition, the accuracy between our approach and current model are listed in Table 4. According to both two tables, our proposed method reduces the training time; it yielded higher scores in 51.3% less time than the MLP model with full features. It can be seen that other models with APGWO methods also have a training time lower than models with full features. Moreover, our approach is outperformed than other current method. To be more specific, with 131 classes, our model achieves 99.9% compared to Pure-CNN (81 classes), CNN (60 classes), and VGG16 (120 classes) which are 98.8%, 96.3%, and 95% accuracy, respectively.

TABLE 3. the training time comparison between models.

	Full	APGWO	Percentage reduction
KNN	9:56:22	4:49:03	51.3%
SVM	0:24:14	0:09:44	59.8%
DT	0:04:47	0:03:20	50.8%
RF	0:00:31	0:00:28	11.3%
MLP	0:02:45	0:00:41	75%

TABLE 4. The comparison between proposed approach and others model.

	Proposed approach (131 classes)	CNN [2] (60 classes)	Pure-CNN[3] (81 classes)	VGG16[4] (130 classes)
MLP	99.9%	96.3%	98.8%	95%

Along with the achievements already mentioned above, our model also has the following limitations. The Grey Wolf Optimization (GWO) algorithm's disadvantages include a slow rate of convergence and a propensity for local optimums on specific problems. In future work, we will utilize Improved Grey Wolf Optimization (IGWO) to improve our approach. This method will add disturbance to the wolf to active search capabilities, and dynamic weights are applied to the wolf's position to prevent it from losing diversity and settling into a local optimum. This strategy could enhance the accuracy of our model.

V. CONCLUSION

This paper has proposed a method for classifying various types of fruits in order to support in manufacturing. The scientific community has been working to effectively classify various fruits by leveraging the capabilities of ML or DL. In this study, we proposed a model using Densnet121 as feature extractor and a feature selection serves as a mechanism to choose the most relevant features for the classification from the image of the fruits. Adaptive Particle Grey Wolf Optimization (APGWO) is used to enhance a feature subset selection method. With 508 selected features from the original dataset, we have applied various models for training such as K-Nearest Neighbour (KNN), Random Forest (RF), Multilayer Perceptron (MLP), Support Vector Machine (SVM), and Decision Tree (DT). The performance of the proposed methods has been achieved with high accuracy. The training and execution time are significantly reduced when irrelevant input attributes are removed. In future work, we may apply the dimensionality reduction for multiple models before they go through the FA step. In addition, the MLP may be automatically tuned by adjusting the number of hidden layers and hidden nodes, the activation functions, or the parameter of the feature selection method for improving the performance. By combining the proposed method with volume/mass estimation method [39], [40], and [41] the completed system in fruit/vegetable recognition and sort will be developed in the upcoming research.

APPENDIX

TABLE 5. Glossary.

ACRONYMS	
ML	Machine Learning
AI	Artificial Intelligence
KNN	K Nearest Neighbor
SVM	Support Vector Machine
DT	Decision Tree
RF	Random Forest Classifier
FE	Feature Extraction
FS	Feature Selection
APGWO	Adaptive Particle Grey Wolf Optimization
GWO	Grey Wolf Optimization

REFERENCES

- [1] *Health Benefits of Fruits and Vegetables—PubMed*. Accessed: Jun. 6, 2022. [Online]. Available: <https://pubmed.ncbi.nlm.nih.gov/22797986/>
- [2] H. Mureşan and M. Oltean, "Fruit recognition from images using deep learning," Jan. 2021, *arXiv:1712.00580*.
- [3] A. Kausar, M. Sharif, J. Park, and D. R. Shin, "Pure-CNN: A framework for fruit images classification," in *Proc. Int. Conf. Comput. Sci. Comput. Intell. (CSCI)*, Dec. 2018, pp. 404–408, doi: [10.1109/CSCI46756.2018.00082](https://doi.org/10.1109/CSCI46756.2018.00082).
- [4] H. Tomar, "Multi-class image classification of fruits and vegetables using transfer learning techniques," Dublin Bus. School, Dublin, Ireland, Tech. Rep. 4252, 2020. Accessed: Jun. 6, 2022. [Online]. Available: <https://esource.dbs.ie/handle/10788/4252>
- [5] D. A. Pisner and D. M. Schnyer, "Support vector machine," in *Machine Learning*, A. Mechelli and S. Vieira, Eds. New York, NY, USA: Academic, 2020, ch. 6, pp. 101–121, doi: [10.1016/B978-0-12-815739-8.00006-7](https://doi.org/10.1016/B978-0-12-815739-8.00006-7).
- [6] J. M. Keller, M. R. Gray, and J. A. Givens, "A fuzzy K-nearest neighbor algorithm," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. SMC-15, no. 4, pp. 580–585, Jul./Aug. 1985, doi: [10.1109/TSMC.1985.6313426](https://doi.org/10.1109/TSMC.1985.6313426).
- [7] M. Jahangir, H. Afzal, M. Ahmed, K. Khurshid, M. F. Amjad, R. Nawaz, and H. Abbas, "Auto-MeDiSine: An auto-tunable medical decision support engine using an automated class outlier detection method and AutoMLP," *Neural Comput. Appl.*, vol. 32, no. 7, pp. 2621–2633, Apr. 2020. Accessed: Jun. 10, 2022. [Online]. Available: <https://www.springerprofessional.de/en/auto-medisine-an-auto-tunable-medical-decision-support-engine-us/16630282>
- [8] M. Jahangir, H. Afzal, M. Ahmed, K. Khurshid, and R. Nawaz, "An expert system for diabetes prediction using auto tuned multi-layer perceptron," in *Proc. Intell. Syst. Conf. (IntelliSys)*, Sep. 2017, pp. 722–728, doi: [10.1109/IntelliSys.2017.8324209](https://doi.org/10.1109/IntelliSys.2017.8324209).
- [9] M. T. Le, M. T. Vo, L. Mai, and S. V. T. Dao, "Predicting heart failure using deep neural network," in *Proc. Int. Conf. Adv. Technol. Commun. (ATC)*, Oct. 2020, pp. 221–225, doi: [10.1109/ATC50776.2020.9255445](https://doi.org/10.1109/ATC50776.2020.9255445).
- [10] J. R. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, Mar. 1986, doi: [10.1007/BF00116251](https://doi.org/10.1007/BF00116251).
- [11] Z. Mašćet and A. Subasi, "Congestive heart failure detection using random forest classifier," *Comput. Methods Programs Biomed.*, vol. 130, pp. 54–64, Jul. 2016, doi: [10.1016/j.cmpb.2016.03.020](https://doi.org/10.1016/j.cmpb.2016.03.020).
- [12] M. M. Kabir, M. Shahjahan, and K. Murase, "A new local search based hybrid genetic algorithm for feature selection," *Neurocomputing*, vol. 74, no. 17, pp. 2914–2928, Oct. 2011, doi: [10.1016/j.neucom.2011.03.034](https://doi.org/10.1016/j.neucom.2011.03.034).
- [13] M. Mafarja, I. Aljarah, A. A. Heidari, H. Faris, P. Fournier-Viger, X. Li, and S. Mirjalili, "Binary dragonfly optimization for feature selection using time-varying transfer functions," *Knowl.-Based Syst.*, vol. 161, pp. 185–204, Dec. 2018, doi: [10.1016/j.knosys.2018.08.003](https://doi.org/10.1016/j.knosys.2018.08.003).
- [14] D. Selvaraj. (2013). *A Review on Tissue Segmentation and Feature Extraction of MRI Brain Images*. Accessed: Jun. 6, 2022. [Online]. Available: <https://www.semanticscholar.org/paper/A-Review-on-Tissue-Segmentation-and-Feature-of-MRI-Selvaraj/fabfa96897dcb59ad9f04b5ff92bd15e1bd159ef>
- [15] L. A. Belanche and F. F. González, "Review and evaluation of feature selection algorithms in synthetic problems," Jan. 2011, *arXiv:1101.2320*.
- [16] L. Yu and H. Liu, "Efficient feature selection via analysis of relevance and redundancy," *J. Mach. Learn. Res.*, vol. 5, no. 10, pp. 1205–1224, 2004.
- [17] G. Uysal. (2012). *A novel probabilistic feature selection method for text classification | PDF | artificial neural network | statistical classification*. Scribd. Accessed: Jun. 10, 2022. [Online]. Available: <https://www.scribd.com/document/386131380/Uysal-Gunal-2012-A-Novel-Probabilistic-Feature-Selection-Method-for-Text-Classification>
- [18] Z. Xu, I. King, M. R.-T. Lyu, and R. Jin, "Discriminative semi-supervised feature selection via manifold regularization," *IEEE Trans. Neural Netw.*, vol. 21, no. 7, pp. 1033–1047, Jul. 2010, doi: [10.1109/TNN.2010.2047114](https://doi.org/10.1109/TNN.2010.2047114).
- [19] K. Yan and D. Zhang, "Feature selection and analysis on correlated gas sensor data with recursive feature elimination," *Sens. Actuators B, Chem.*, vol. 212, pp. 353–363, Jun. 2015, doi: [10.1016/j.snb.2015.02.025](https://doi.org/10.1016/j.snb.2015.02.025).
- [20] A. Jain and D. Zongker, "Feature selection: Evaluation, application, and small sample performance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 2, pp. 153–158, Feb. 1997, doi: [10.1109/34.574797](https://doi.org/10.1109/34.574797).

- [21] A. Jović, K. Brkić, and N. Bogunović, "A review of feature selection methods with applications," in *Proc. 38th Int. Conv. Inf. Commun. Technol., Electron. Microelectron. (MIPRO)*, May 2015, pp. 1200–1205, doi: [10.1109/MIPRO.2015.7160458](https://doi.org/10.1109/MIPRO.2015.7160458).
- [22] P. Kuila and P. K. Jana, "A novel differential evolution based clustering algorithm for wireless sensor networks," *Appl. Soft Comput.*, vol. 25, pp. 414–425, Dec. 2014, doi: [10.1016/j.asoc.2014.08.064](https://doi.org/10.1016/j.asoc.2014.08.064).
- [23] H.-G. Beyer and H.-P. Schwefel, "Evolution strategies—A comprehensive introduction," *Natural Comput.*, vol. 1, no. 1, pp. 3–52, Mar. 2002, doi: [10.1023/A:1015059928466](https://doi.org/10.1023/A:1015059928466).
- [24] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw. (ICNN)*, vol. 4, 1995, pp. 1942–1948, doi: [10.1109/ICNN.1995.488968](https://doi.org/10.1109/ICNN.1995.488968).
- [25] Z. W. Geem, J. H. Kim, and G. V. Loganathan, "A new heuristic optimization algorithm: Harmony search," *J. Simul.*, vol. 76, no. 2, pp. 60–68, Feb. 2001, doi: [10.1177/003754970107600201](https://doi.org/10.1177/003754970107600201).
- [26] R. V. Rao, V. J. Savsani, and D. P. Vakharia, "Teaching–learning–based optimization: An optimization method for continuous non-linear large scale problems," *Inf. Sci.*, vol. 183, no. 1, pp. 1–15, Jan. 2012, doi: [10.1016/j.ins.2011.08.006](https://doi.org/10.1016/j.ins.2011.08.006).
- [27] J. Too, A. Abdullah, N. M. Saad, N. M. Ali, and W. Tee, "A new competitive binary grey wolf optimizer to solve the feature selection problem in EMG signals classification," *Computers*, vol. 7, no. 4, p. 58, Nov. 2018, doi: [10.3390/computers7040058](https://doi.org/10.3390/computers7040058).
- [28] E. Emary, H. M. Zawbaa, and A. E. Hassanien, "Binary grey wolf optimization approaches for feature selection," *Neurocomputing*, vol. 172, pp. 371–381, Jan. 2016, doi: [10.1016/j.neucom.2015.06.083](https://doi.org/10.1016/j.neucom.2015.06.083).
- [29] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014, doi: [10.1016/j.advengsoft.2013.12.007](https://doi.org/10.1016/j.advengsoft.2013.12.007).
- [30] L.-Y. Chuang, H.-W. Chang, C.-J. Tu, and C.-H. Yang, "Improved binary PSO for feature selection using gene expression data," *Comput. Biol. Chem.*, vol. 32, no. 1, pp. 29–37, Feb. 2008, doi: [10.1016/j.compbiolchem.2007.09.005](https://doi.org/10.1016/j.compbiolchem.2007.09.005).
- [31] X. He, Q. Zhang, N. Sun, and Y. Dong, "Feature selection with discrete binary differential evolution," in *Proc. Int. Conf. Artif. Intell. Comput. Intell.*, vol. 4, Oct. 2009, pp. 327–330, doi: [10.1109/AICI.2009.438](https://doi.org/10.1109/AICI.2009.438).
- [32] Q. Li, H. Chen, H. Huang, X. Zhao, Z. Cai, C. Tong, W. Liu, and X. Tian, "An enhanced grey wolf optimization based feature selection wrapped kernel extreme learning machine for medical diagnosis," *Comput. Math. Methods Med.*, vol. 2017, Jan. 2017, Art. no. e9512741, doi: [10.1155/2017/9512741](https://doi.org/10.1155/2017/9512741).
- [33] H. Faris, I. Aljarah, M. A. Al-Betar, and S. Mirjalili, "Grey wolf optimizer: A review of recent variants and applications," *Neural Comput. Appl.*, vol. 30, no. 2, pp. 413–435, Jul. 2018, doi: [10.1007/s00521-017-3272-5](https://doi.org/10.1007/s00521-017-3272-5).
- [34] T. M. Le, L. V. Tran, and S. V. T. Dao, "A feature selection approach for fall detection using various machine learning classifiers," *IEEE Access*, vol. 9, pp. 115895–115908, 2021, doi: [10.1109/ACCESS.2021.3105581](https://doi.org/10.1109/ACCESS.2021.3105581).
- [35] T. M. Le, T. M. Vo, T. N. Pham, and S. V. T. Dao, "A novel wrapper–based feature selection for early diabetes prediction enhanced with a metaheuristic," *IEEE Access*, vol. 9, pp. 7869–7884, 2021, doi: [10.1109/ACCESS.2020.3047942](https://doi.org/10.1109/ACCESS.2020.3047942).
- [36] M. R. Girgis, A. S. Ghiduk, and E. H. Abd-Elkawy, "Automatic data flow test paths generation using the genetical swarm optimization technique," *Int. J. Comput. Appl.*, vol. 116, no. 22, pp. 25–33, Apr. 2015.
- [37] D. N. Jeyakumar, T. Jayabarathi, and T. Raghunathan, "Particle swarm optimization for various types of economic dispatch problems," *Int. J. Electr. Power Energy Syst.*, vol. 28, no. 1, pp. 36–42, Jan. 2006, doi: [10.1016/j.ijepes.2005.09.004](https://doi.org/10.1016/j.ijepes.2005.09.004).
- [38] T. M. Le, T. N. Pham, and S. V. T. Dao, "A novel wrapper-based feature selection for heart failure prediction using an adaptive particle swarm grey wolf optimization," in *Enhanced Telemedicine e-Health: Advanced IoT Enabled Soft Computing Framework*, G. Marques, A. K. Bhoi, I. de la Torre Díez, and B. Garcia-Zapirain, Eds. Cham, Switzerland: Springer, 2021, pp. 315–336, doi: [10.1007/978-3-030-70111-6_15](https://doi.org/10.1007/978-3-030-70111-6_15).
- [39] T. Huynh, L. Tran, and S. Dao, "Real-time size and mass estimation of slender axi-symmetric fruit/vegetable using a single top view image," *Sensors*, vol. 20, no. 18, p. 5406, Sep. 2020, doi: [10.3390/s20185406](https://doi.org/10.3390/s20185406).
- [40] T. T. M. Huynh, L. Tonthat, and S. V. T. Dao, "A vision-based method to estimate volume and mass of fruit/vegetable: Case study of sweet potato," *Int. J. Food Properties*, vol. 25, no. 1, pp. 717–732, Dec. 2022, doi: [10.1080/10942912.2022.2057528](https://doi.org/10.1080/10942912.2022.2057528).
- [41] T. Huynh and S. Dao, "Highly efficient fruit mass and size estimation using only top view images," *Proceedings*, vol. 42, no. 1, p. 57, 2019, doi: [10.3390/ccsa-6-06588](https://doi.org/10.3390/ccsa-6-06588).



TRI TRAN MINH HUYNH received the B.Eng. degree in electrical engineering and the M.Sc. degree in electronics engineering, in 2018 and 2021, respectively. He is currently a Laboratory Specialist at the School of Electrical Engineering, International University, Vietnam National University, Vietnam. His research interests include embedded systems, machine vision, and deep learning.



TUAN MINH LE received the bachelor's degree in electrical engineering and the M.Sc. degree in electronics engineering from International University, Vietnam National University, Vietnam, in 2018 and 2022, respectively. He is currently a Laboratory Technician. His research interests include embedded systems, the Internet of Things, and machine learning.



LONG TON THAT received the B.Eng., M.Sc., and Ph.D. degrees from The University of Manchester, U.K., in 2007, 2008, and 2012, respectively. Since 2014, he has been a Lecturer with the Department of Automation and Control Engineering, International University (IU), Vietnam National University Ho Chi Minh City (VNU-HCMC). His research interests include computational intelligence, nonlinear control theory in biological systems, motor control, and biomedical imaging.



LY VAN TRAN received the B.Eng. degree in mechanical engineering, in 1996, and the M.Eng. degree in manufacturing systems engineering, in 2000. From 2000 to 2019, he was in various industries with a wide range of roles from operations to senior management. His research interests include operation research and product development.



SON VU TRUONG DAO received the B.Eng. degree in aeronautical engineering, the M.Sc. degree in manufacturing systems and technology, and the Ph.D. degree in sensors technology, in 2004, 2005, and 2010, respectively. From 2006 to 2008, he was with Hylax Ltd., Singapore, designing high-power laser systems. From 2010 to 2012, he was a Postdoctoral Fellow at Kindai University, Japan. From 2012 to 2015, he worked as a Senior Scientist at Ritsumeikan University, Japan. His research interests include development of advanced imaging sensors and systems and applied artificial intelligence.

• • •