

Article

An Automated Hyperparameter Tuning Recurrent Neural Network Model for Fruit Classification

Kathiresan Shankar ¹, Sachin Kumar ^{1,*}, Ashit Kumar Dutta ², Ahmed Alkhayyat ³,
Anwar Ja'afar Mohamad Jawad ⁴, Ali Hashim Abbas ⁵ and Yousif K. Yousif ⁶

¹ Big Data and Machine Learning Lab, South Ural State University, 454080 Chelyabinsk, Russia; drkshankar@ieee.org

² Department of Computer Science and Information System, College of Applied Sciences, AlMaarefa University, Riyadh 11597, Saudi Arabia; adotta@mcst.edu.sa

³ College of Technical Engineering, The Islamic University, Najaf 61001, Iraq; ahmedalkhayyat85@iunajaf.edu.iq

⁴ Department of Computer Techniques Engineering, Al-Rafidain University College, Baghdad 10064, Iraq; anwar.jawad@ruc.edu.iq

⁵ College of Information Technology, Imam Ja'afar Al-Sadiq University, Al-Muthanna 66002, Iraq; alsalamy1987@gmail.com

⁶ Department of Computer Technical Engineering, Al-Hadba University College, Mosul 41001, Iraq; yousif.ameen@hcu.edu.iq

* Correspondence: sachinagnihotri16@gmail.com

Abstract: Automated fruit classification is a stimulating problem in the fruit growing and retail industrial chain as it assists fruit growers and supermarket owners to recognize variety of fruits and the status of the container or stock to increase business profit and production efficacy. As a result, intelligent systems using machine learning and computer vision approaches were explored for ripeness grading, fruit defect categorization, and identification over the last few years. Recently, deep learning (DL) methods for classifying fruits led to promising performance that effectively extracts the feature and carries out an end-to-end image classification. This paper introduces an Automated Fruit Classification using Hyperparameter Optimized Deep Transfer Learning (AFC-HPODTL) model. The presented AFC-HPODTL model employs contrast enhancement as a pre-processing step which helps to enhance the quality of images. For feature extraction, the Adam optimizer with deep transfer learning-based DenseNet169 model is used in which the Adam optimizer fine-tunes the initial values of the DenseNet169 model. Moreover, a recurrent neural network (RNN) model is utilized for the identification and classification of fruits. At last, the Aquila optimization algorithm (AOA) is exploited for optimal hyperparameter tuning of the RNN model in such a way that the classification performance gets improved. The design of Adam optimizer and AOA-based hyperparameter optimizers for DenseNet and RNN models show the novelty of the work. The performance validation of the presented AFC-HPODTL model is carried out utilizing a benchmark dataset and the outcomes report the promising performance over its recent state-of-the-art approaches.

Keywords: fruit classification; deep transfer learning; recurrent neural network; Adam optimizer; hyperparameter tuning

MSC: 68T45



Citation: Shankar, K.; Kumar, S.; Dutta, A.K.; Alkhayyat, A.; Jawad, A.J.M.; Abbas, A.H.; Yousif, Y.K. An Automated Hyperparameter Tuning Recurrent Neural Network Model for Fruit Classification. *Mathematics* **2022**, *10*, 2358. <https://doi.org/10.3390/math10132358>

Academic Editor: Jianping Gou

Received: 23 May 2022

Accepted: 28 June 2022

Published: 5 July 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Automatic fruit classification is an intriguing challenge in the growth of fruit and retailing industrial chain since it is helpful for the fruit producers and supermarkets to discover various fruits and their condition from the containers or stock with a view to improvising manufacturing effectiveness and revenue of the business [1]. Thus, intelligent systems making use of machine learning (ML) approaches and computer vision (CV) have been applied to fruit defect recognition, ripeness grading, and classification in the last decade [2].

In automated fruit classification, two main methods, one conventional CV-related methodologies and the other one deep learning (DL)-related methodologies, were investigated. The conventional CV-oriented methodologies initially derive the low-level features, after which they execute image classification through the conventional ML approaches, while the DL-related techniques derive the features efficiently and execute an endwise image classification [3]. In the conventional image processing and CV approaches, imagery features, such as shape, texture, and color, were utilized as input unit for fruit classification.

Previously, fruit processing and choosing depended on artificial techniques, leading to a huge volume of waste of labor [4]. Nonetheless, the above-mentioned techniques require costly devices (various kinds of sensors) and professional operators, and their comprehensive preciseness is typically less than 85% [5]. With the speedy advancement of 4G communication and extensive familiarity with several mobile Internet gadgets, individuals have created a large number of videos, sounds, images, and other data, and image identification technology has slowly matured [6].

Image-related fruit recognition has gained the interest of authors because of its inexpensive gadgets and extraordinary performances [7]. At the same time, it is needed to design automated tools capable of handling unplanned scenarios such as accidental mixing of fresh products, fruit placement in unusual packaging, different lighting conditions or spider webs on the lens, etc. Such situations may also cause uncertainty in the model results. The intelligent recognition of fruit might be utilized not only from the picking stages of the prior fruit but also in the processing and picking phase in the next stage [8]. Fruit identification technology depending on DL could substantially enhance the execution of fruit identification and comprises a positive impact on fostering the advancement of smart agriculture. In comparison with artificial features and conventional ML combination techniques, DL may derive features automatically, and contains superior outcomes that slowly emerged as the general methodology of smart recognition [9]. Particularly, convolutional neural network (CNN) is one of the vital DL models utilized for image processing. It is a type of artificial neural network (ANN) which utilizes convolution operation in at least one of the layers. Recently, CNNs have received significant attention on the image classification process. Specifically, in the agricultural sector, CNN-based approaches have been utilized for fruit classification and fruit detection [10].

This paper introduces an Automated Fruit Classification using Hyperparameter Optimized Deep Transfer Learning (AFC-HPODTL) model. The presented AFC-HPODTL model employs contrast enhancement as a pre-processing step which helps to improve the quality of the image. Next is the Adam optimizer with deep transfer learning-based DenseNet169 model. Moreover, the Aquila optimization algorithm (AOA) with recurrent neural network (RNN) model is utilized for the identification and classification of fruits. The performance validation of the presented AFC-HPODTL model is carried out using a benchmark dataset and examines the results under different aspects. In summary, the contribution of the paper is as follows:

- An intelligent AFC-HPODTL model comprising of pre-processing, Adam with DenseNet169-based feature extraction, RNN classification, and AOA-based hyperparameter tuning is presented. To the best of our knowledge, the AFC-HPODTL model has never been presented in the literature.
- Hyperparameter tuning of the DenseNet169 and RNN models takes place using Adam optimizer and AOA techniques respectively, which in turn considerably enhances the fruit classification performance shows the novelty of the work.
- The performance of the proposed AFC-HPODTL model is validated on two open databases and the results demonstrate the better performance over other DL models.

The rest of the paper is organized as follows. Section 2 offers a detailed literature review of existing fruit classification models. Next, Section 3 introduces the proposed AFC-HPODTL model and Section 4 provides the experimental result analysis. Finally, Section 5 concludes the study.

2. Related Works

In [11], the authors suggest an effective structure for fruit classification with the help of DL. Most importantly, the structure depends on two distinct DL architectures. One is a proposed light model of six CNN layers, and the other is a fine-tuned visual geometry group-16 pretrained DL method. Rojas-Aranda et al. [12] provide an image classification technique, based on lightweight CNN, for the purpose of fastening the checking procedure in the shops. A novel images dataset has presented three types of fruits, without or with plastic bags. These input units are the RGB histogram, the RGB centroid acquired from K-means clustering, and single RGB color. In [13], the researchers suggested a new fruit classification method that uses Long Short-Term Memory (LSTM), RNN structures, and CNN features. Type-II fuzzy advancement was further utilized as a preprocessing device for advancing the images. Furthermore, TLBO-MCET was used to tune the hyperparameters of the suggested method.

In [14], the researchers advanced a hybrid DL-related fruit image classification structure called attention-related densely connected convolution network with convolution auto-encoder (CAE-ADN), that employs a CAE for pretraining the images and leverages an attention-related DenseNet for extracting the image features. In the opening portion of the structure, an unsupervised technique with a group of images is applied to pretrain the greedy layer-wised CAE. In the next portion of the structure, the supervised ADN with the ground truth is applied. The structure's last portion performs an estimation of the classes of fruits. Kumari and Gomathy [15] recommend a classical method that utilizes texture features and color for fruit classification. The conventional fruit classification technique is reliable upon manual function on the basis of visual ability. The classification can be performed with the help of the Support Vector Machine (SVM) classifier depending on co-occurrence and statistical features extracted from the wavelet transform.

In [16], a 13-layer CNN was devised. Three categories of data augmentation methods are employed: noise injection, image rotation, and Gamma correction. The researchers made a comparison of average pooling and max pooling. The stochastic gradient descent with momentum is utilized for training the CNN with a minibatch size of 128. In [17], a fruit image classification technique based on lightweight neural network MobileNetV2 and transfer learning (TL) method is employed for recognizing fruit images. They leveraged MobileNetV2 network pretrained by ImageNet dataset as a base system after replacing the topmost layer of the base system with a Softmax classifier and conventional convolution layer. They applied dropout to the newly added conv2d simultaneously for diminishing overfitting. The pretrained MobileNetV2 is utilized for extracting features and the Softmax classifier is utilized for classifying features.

The researchers in [18] provide an extensive review of the hyperparameter tuning of CNN models by the use of nature-inspired algorithms. It provides an overview of various CNN approaches utilized for image classification, segmentation, and styling. Next, in [19], the mathematical relationship between four hyperparameters, namely learning rate, batch size, dropout rate, and convolution kernel size were investigated in detail. A generalized multi-parameter mathematical correlation approach was derived, showing that the hyperparameters play a vital part in the efficiency of the NN models. Guo et al. [20] introduced a distributed particle swarm optimization (DPSO) algorithm for hyperparameter tuning of the CNN models. On comparing with the complex, with manual designs based on historical experience and personal preference, the DPSO algorithm effectually chooses the hyperparameters of the CNN model. In addition, the DPSO algorithm has shown significant improvement over the conventional PSO algorithm.

Several fruit classification models exist in the literature. Despite the development of the ML and DL models in previous works, it is still necessary to boost the fruit classification performance. Due to the continual deepening of the model, the number of parameters of DL models gets increased rapidly, resulting in model overfitting. Moreover, different hyperparameters have a significant impact on the efficiency of the CNN model. Particularly, the hyperparameters such as epoch count, batch size, and learning rate selection are

important to achieve effective results. As the trial and error method for hyperparameter tuning is a tiresome and erroneous process, metaheuristic algorithms can be applied. Therefore, in this work, we employ the Adam optimizer and AOA algorithm for the parameter selection of the DenseNet169 and RNN models, respectively.

3. The Proposed Model

In this study, a new AFC-HPODTL model was developed for the automatic identification and classification of fruits. The presented AFC-HPODTL model comprises a series of processes namely pre-processing, DenseNet169 feature extraction, Adam optimizer, RNN classification, and AOA hyperparameter optimization. Figure 1 illustrates the overall process of the AFC-HPODTL algorithm.

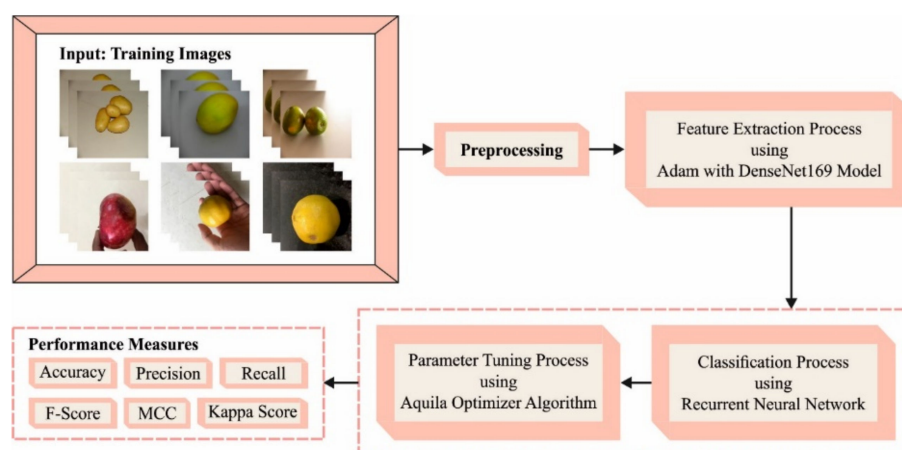


Figure 1. Overall process of AFC-HPODTL approach.

3.1. Contrast Enhancement

Initially, the presented AFC-HPODTL model employs contrast enhancement as a pre-processing step which helps for improving the quality of the image. CLAHE is different from AHE as it gets care of over-amplification of contrasts. CLAHE functions on smaller areas from the image named tiles, before the total images. The adjacent tiles are then integrated utilizing bilinear interpolation for removing the artificial boundary. This technique is executed for improving the contrast of images.

3.2. Feature Extraction

To extract feature vectors from the pre-processed fruit images, the DenseNet169 model is employed. The CNN structures have two bases, namely the convolution and classification bases. The convolution base contains three important kinds of layers, namely the convolution, activation, and pooling layers [21]. These kinds of layers were utilized for discovering the fundamental features of input images that are named feature maps (FM). The FM was obtained by applying convolutional procedures to input images or prior features utilizing linear filtering, and integration of a bias term. Afterward, the passing of this FM was achieved with a nonlinear activation function such as Sigmoid and ReLU. Conversely, the classification base comprised the dense layer integrated with activation layer for converting the FMs to 1D vector for expediting the classifier task utilizing several neurons. Generally, more than one dropout layer is used along with the classification base to minimize the overfitting which encounters CNN structures and enhances its generalized nature. Adding a dropout layer to the classification base establishes a novel hyperparameter named dropout rate. Usually, the dropout rate is fixed in the range of 0.1–0.9.

DenseNet is the most novel addition to the NNs utilized for detection of visual objects. DenseNet169 is a process of the DenseNet group [22]. The DenseNet group is planned for the executing image classifier. DenseNet169 is superior to the rest of the DenseNet group. Typically, in DenseNet, every image is being trained. An ImageNet image database,

however, can be trained by the method and stored and tested by loading our saved method rather than ImageNet. At this point, the results of the earlier layer is obtained concatenated with the future layer DenseNet. DenseNet has been shown to reduce the accuracy from a higher level NN that is produced by vanishing gradients, while there is a longer path that exists amongst the input as well as output layers and the data obtained vanish even before attaining its target. The DenseNet goes to type of typical network. Based on the new stats, a convolution layer is more effective and accurate when it can be shorter and linked among layers nearby, the input and closer output. At this point, the DenseNet was employed for connecting all the layers that are in feed-forward fashion. Generally, a classical convolution network has L layers. Moreover, L linking exists among the L layer. That represents one link among all the layers and their following layers.

It takes $L(L + 1)/2$ direct connections from the networks. For all the layers as input, every presiding layer is utilized. In order to input every following layer, their FM is being utilized. Several benefits can be obtained from DenseNet. It can decrease the vanishing gradient problems. The feature propagation is strengthened, feature reprocess is encouraged, and it decreases the number of parameters. The presented structure is estimated on extremely competitive image detection benchmark ImageNet and it also utilizes the saved and load function. The combination of layers was feasible as defined by when there is an entire similarity from the FM dimensional at the time of concatenation or addition. DenseNet is divided into DenseBlocks with a different number of filters, but within the blocks, the dimensional is similar. The Batch normalization (BN) was executed by utilizing down-sampling with transition layer. That is assumed to be a vital stage by the CNN. Based on the improvement of dimensional of the channels, the number among the DenseBlocks of filter variations, The rate of growth is represented by K . It plays an important role in generalizing l^{th} layer. The count of data which is further required from all the layers are being measured by:

$$k_l = k_0 + k * (l - 1) \quad (1)$$

Here, the Adam optimizer fine-tunes the initial values of the DenseNet169 model. We employ ADAM, which is an optimization approach, as a substitute for traditional stochastic gradient descent algorithm for updating the network weight in training dataset [23]. This is utilized for performing optimization. ADAM is derived from adagrad and it is the more adaptable technique. ADAGRAD and momentum are collectively called ADAM.

Variables $w^{(t)}$ and $L^{(t)}$, where index t specifies the present trainable iteration, the parameter update in ADAM is shown as follows:

$$m_w^{(t+1)} \leftarrow \beta_1 m_w^{(t)} + (1 - \beta_1) \nabla_w L^{(t)} \quad (2)$$

$$v_w^{(t+1)} \leftarrow \beta_2 v_w^{(t)} + (1 - \beta_2) (\nabla_w L^{(t)})^2 \quad (3)$$

$$\hat{m}_w = \frac{m_w^{(t+1)}}{1 - (\beta_1)^{(t+1)}} \quad (4)$$

$$\hat{v}_w = \frac{v_w^{(t+1)}}{1 - (\beta_2)^{(t+1)}} \quad (5)$$

$$w^{t+1} \leftarrow w^t - \eta \frac{\hat{m}}{\sqrt{\hat{v}_w} + \epsilon} \quad (6)$$

From the expression, β_1 and β_2 denotes gradient forgetting factor and second moment of gradient. ϵ is a small scalar utilized for preventing division by 0.

3.3. Fruit Classification

In the final stage, the RNN model is utilized for the identification and classification of fruits. The presented technique makes use of the LSTM model, which is a special kind

of RNN. In the RNN, the neurons are interconnected with one another through a directed cycle [24]. The RNN model processes the data sequentially since it utilizes internal memory for processing a series of inputs or words. RNN implements a similar task to all the elements since the output is dependent on each preceding node input and remembered data. Figure 2 depicts the structure of RNN. For additional processing, Equation (7) characterizes typical RNN structure where h_t indicates the novel state at time t , f_w denotes a function with w variable, $h_t - 1$ represents an older state (preceding state), and x_t signifies input vector at t time.

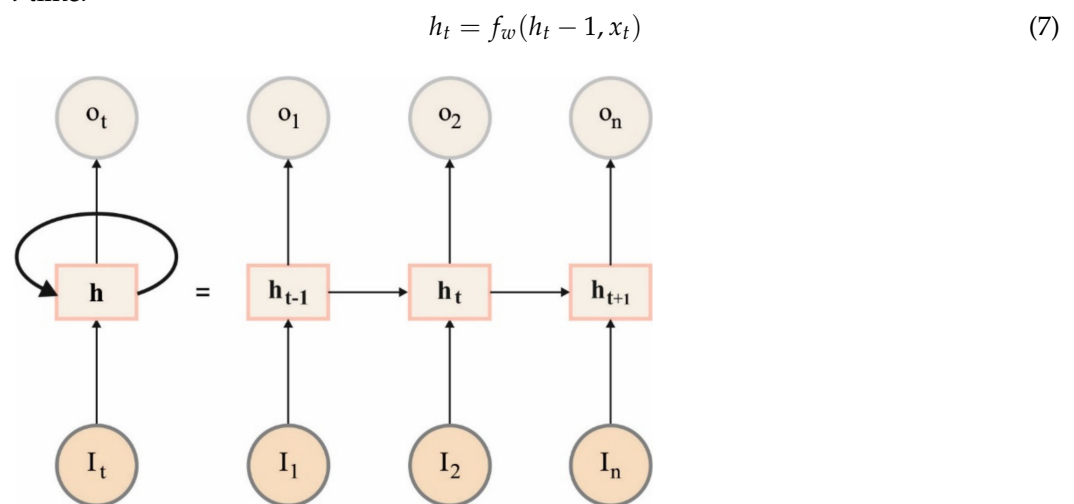


Figure 2. RNN architecture.

We alternate the Equations (7) and (8) viz., utilized to assign weight.

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t) \quad (8)$$

Given that, the activation function is denoted as $\tanh$, the weight of hidden state is represented by w_h , and the input vector can be signified as x_t . The exploding vanishing or gradient problems are generated while learning of gradient model is back-propagated by using the network. A special kind of RNN model called LSTM is utilized for handling the gradient vanishing problems. The LSTM saves long-term dependency by utilizing three diverse gates in an efficient manner. The LSTM gate is explained in the following expression.

$$\text{Input Gate } In_t = \sigma(W_{in} \cdot [hs_t - 1], x_t + b_{in}) \quad (9)$$

$$\text{Memory Cell } C_t = \tanh(W_c \cdot [hs_t - 1], x_t + b_e) \quad (10)$$

$$\text{Forget Gate } f_t = \sigma(W_f \cdot [hs_t - 1], x_t + b_f) \quad (11)$$

$$\text{Output Gate } f_o = \sigma(W_o \cdot [hs_t - 1], x_t + b_o) \quad (12)$$

From the formula, b characterizes the bias vector, W is utilized for weighted, and x_t indicates the input vector at t time, whereas, i , f , and c represent input, forget, cell memory, and output gates.

3.4. Hyperparameter Tuning

In this study, the AOA is exploited to tune the hyperparameters of the RNN model such as learning rate, number of hidden layers, weight initialization, and decay rate. The AOA algorithm is a new modern swarm intelligence approach [25]. There are four hunting strategies of Aquila; for dissimilar types of prey, Aquila might flexibly change hunting strategy for diverse prey and later use their fast speed combined with claws and sturdy feet to attack the prey. The summary of mathematical expression is demonstrated in the following steps.

Step 1: Extended exploration (X_1): higher soar using vertical stoop

Here, the Aquila flies higher above the ground level and widely explores the searching space, later a vertical dive is taken when the Aquila defines the prey region. Such behavior can be mathematically expressed as follows:

$$X_1(t+1) = X_{best}(t) \times \left(1 - \frac{t}{T}\right) + (X_M(t) - X_{best}(t) \times r_1) \quad (13)$$

$$X_M(t) = \frac{1}{N} \sum_{i=1}^N X_i(t) \quad (14)$$

From the equation, $X_{best}(t)$ signifies the optimally obtained location, and $X(t)$ represents the average location of each Aquila in present iteration. t and T indicate the existing iteration and the maximal amount of iterations, correspondingly N denotes the population size, and r_1 refers to an arbitrary integer that lies within the range of zero and one.

Step 2: Narrowed exploration (X_2): contour flight with shorter glide attack

This is a popular hunting methodology for Aquila. It applies short gliding to attack the prey, afterward descending within the designated area and flying around the prey. The updated location is given in the following:

$$X_2(t+1) = X_{best}(t) \times LP(D) + X_R(t) + (y - x) \times r_2 \quad (15)$$

In Equation (15), $X_R(t)$ refers to an arbitrary location of Aquila, D indicates the dimension size, and r_2 represent an arbitrary integer lies in the range of $[0, 1]$. $LP(D)$ signifies Levy flight function that is given in the following:

$$LP(D) = s \times \frac{u \times \sigma}{|v|^{\frac{1}{\beta}}} \quad (16)$$

$$\sigma = \left(\frac{\Gamma(1 + \beta) \times \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) \times \beta \times 2^{\left(\frac{\beta-1}{2}\right)}} \right) \quad (17)$$

From the expression, s and β are constant values equivalent to 0.01 and 1.5, correspondingly, and u and v stand for arbitrary numbers lying within a range $[0, 1]$. y and x represent the spiral shape in the search space that is computed in the following:

$$\begin{cases} x = r \times \sin(\theta) \\ y = r \times \cos(\theta) \\ r = r_3 + 0.00565 \times D_1 \\ \theta = -\omega \times D_1 + \frac{3 \times \pi}{2} \end{cases} \quad (18)$$

In Equation (18), r_3 is the number of search cycles within the interval of 1 and 20, D_1 is comprised of integer numbers from 1 to D dimensional size, then ω is equivalent to 0.005.

Step 3: Expanded exploitation (X_3): lower flight with a slower descent attacks

Here, once the prey region is commonly identified, the Aquila vertically descends to execute an initial attack. AOA uses the designated region to get closer and attack the prey. This behavior can be mathematically modeled by the following equation:

$$X_3(t+1) = (X_{best}(t) - X_M(t)) \times \alpha - r_4 + ((UB - LB) \times r_5 + LB) \times \delta \quad (19)$$

In Equation (19), $X_{best}(t)$ represents the optimally attained location, and $X_M(t)$ indicates the average value of present position. α and δ signify the exploitation fine-tuning parameter set as 0.1, UB and LB denotes the upper and lower limits, and r_4 and r_5 refers to arbitrary value lies in the interval of $(0, 1)$.

Step 4: Narrowed exploitation (X_4): grabbing and walking prey

Here, the Aquila chase the prey with regard to escape trajectory and later attack the prey on the ground. The arithmetical expression of the behavior is given below:

$$\begin{cases} X_4(t+1) = QF \times X_{best}(t) - (G_1 \times X(t) \times r_6) - G_2 \times LF(D) + r_7 \times G_1 \\ QP(t) = t^{\frac{2 \times rand() - 1}{(1-T)^2}} \\ G_1 = 2 \times r_8 - 1 \\ G_2 = 2 \times (1 - \frac{t}{T}) \end{cases} \quad (20)$$

In Equation (20), $X(t)$ indicates the present location, and $QF(t)$ characterizes the quality function value that balances the searching strategy. G_1 represents the movement parameter of Aquila during tracking prey, which is an arbitrary integer lying within the range of $[-1, 1]$. G_2 signifies the flight slope while chasing prey that linearly reduces from 2 to 0. r_6 , r_7 , and r_8 are arbitrary numbers that lie within $[0, 1]$.

The AOA system computes a fitness function (FF) for achieving higher classifier efficiency. It defines the positive integer for demonstrating a better performance of candidate outcomes. During this case, the minimized classifier error rate can be assumed as FF provided in Equation (21).

$$\begin{aligned} fitness(x_i) &= Classifier\ Error\ Rate(x_i) \\ &= \frac{\text{number of misclassified fruit images}}{\text{Total number of fruit images}} * 100 \end{aligned} \quad (21)$$

4. Performance Validation

The experimental validation of the AFC-HPODTL model was tested using two datasets, namely dataset 1 [26] and dataset 2 [27]. The proposed AFC-HPODTL model is simulated using Python 3.6.5 tool on a PC i5-8600k, GeForce 1050Ti 4GB, 16GB RAM, 250GB SSD, and 1TB HDD. The parameter settings of the DenseNet model are given as follows: dropout: 0.5, batch size: 5, epoch count: 50, and activation: ReLU.

4.1. Result Analysis on Dataset 1

Dataset 1 (D1) is an openly accessible fruit and vegetable dataset that comprises 15 classes as shown in Table 1. All the classes involve at least 75 images, resultant from 2633 images in total. These images are gathered at a resolution of 1024×768 pixels on distinct dates and times. The dataset was freely accessible in [26]. A few sample images from dataset 1 are showcased in Figure 3.

Table 1. Dataset-1 details.

Labels	Name	No. of Instances
C1	Agata potato	75
C2	Asterix potato	75
C3	Cashew	75
C4	Diamond peach	75
C5	Fuji apple	75
C6	Granny smith apple	75
C7	Honeydew melon	75
C8	Kiwi	75
C9	Nectarine	75
C10	Onion	75
C11	Orange	75
C12	Plum	75
C13	Spanish pear	75
C14	Tahiti lime	75
C15	Watermelon	75
Total No. of Instances		1125

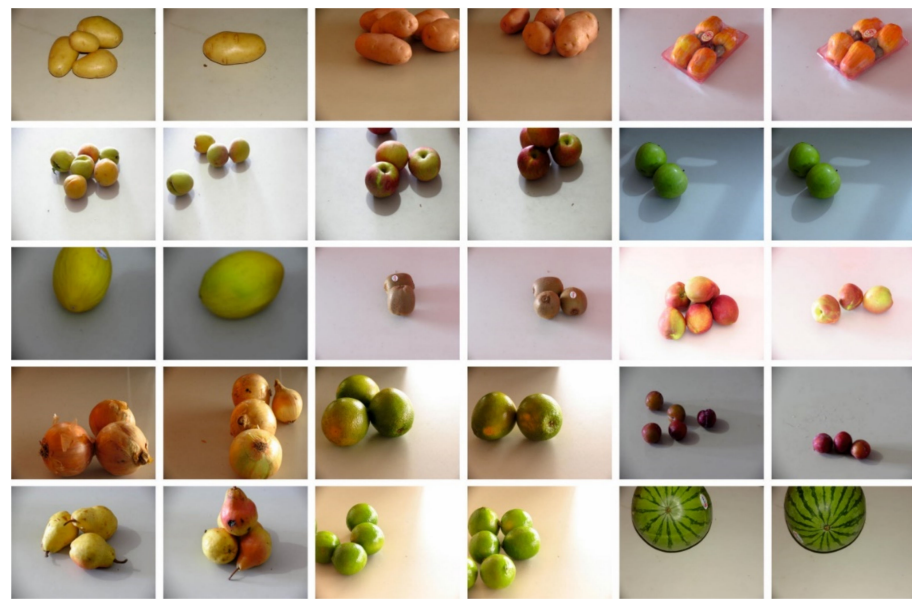


Figure 3. Sample images from dataset 1.

Figure 4 demonstrates a set of confusion matrices created by the AFC-HPODTL model on the test dataset 1. The figure indicates that the AFC-HPODTL model has effectually categorized the images into 15 fruit classes under all datasets.

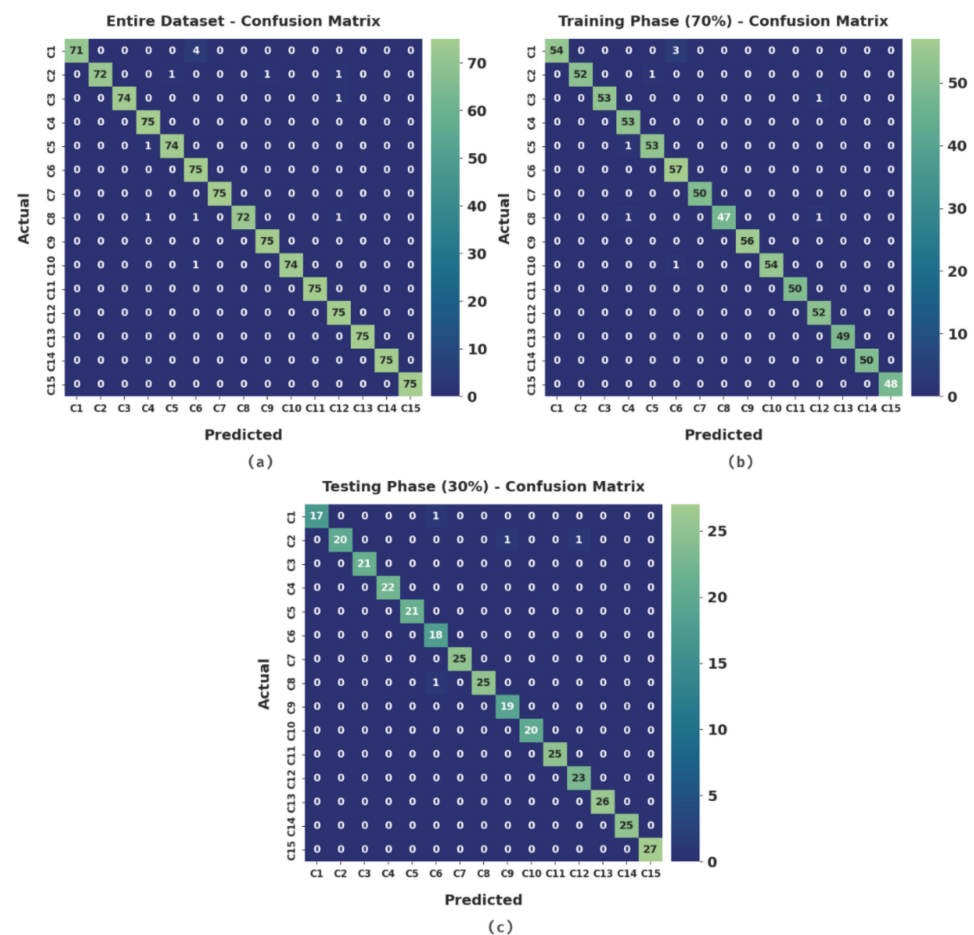


Figure 4. Confusion matrices of AFC-HPODTL approach on dataset1: (a) entire dataset, (b) 70% of TR data, and (c) 30% of TS data.

Table 2 reports the overall fruit classification results of the AFC-HPODTL model obtained under dataset 1. The results indicate that the AFC-HPODTL model obtained effective classification results on all datasets. For instance, with the entire dataset, the AFC-HPODTL model classified 15 classes with average $accu_y$, $prec_n$, $reca_l$, F_{score} , MCC, and kappa score of 99.85%, 98.90%, 98.84%, 98.85%, 98.78%, and 98.76% respectively. Afterward, with 70% of TR data, the AFC-HPODTL approach classified 15 classes with average $accu_y$, $prec_n$, $reca_l$, F_{score} , MCC, and kappa score of 99.85%, 98.95%, 98.88%, 98.80%, 98.83%, and 98.77%, correspondingly. Similarly, with 30% of the TS dataset, the AFC-HPODTL algorithm classified 15 classes with average $accu_y$, $prec_n$, $reca_l$, F_{score} , MCC, and kappa score of 99.84%, 98.72%, 98.77%, 98.70%, 98.64%, and 98.73%, correspondingly.

Table 2. Result analysis of AFC-HPODTL approach under various measures on dataset 1.

Labels	Accuracy	Precision	Recall	F-Score	MCC	Kappa Score
Entire Dataset						
C1	99.64	100.00	94.67	97.26	97.11	-
C2	99.73	100.00	96.00	97.96	97.84	-
C3	99.91	100.00	98.67	99.33	99.28	-
C4	99.82	97.40	100.00	98.68	98.60	-
C5	99.82	98.67	98.67	98.67	98.57	-
C6	99.47	92.59	100.00	96.15	95.95	-
C7	100.00	100.00	100.00	100.00	100.00	-
C8	99.73	100.00	96.00	97.96	97.84	-
C9	99.91	98.68	100.00	99.34	99.29	-
C10	99.91	100.00	98.67	99.33	99.28	-
C11	100.00	100.00	100.00	100.00	100.00	-
C12	99.73	96.15	100.00	98.04	97.92	-
C13	100.00	100.00	100.00	100.00	100.00	-
C14	100.00	100.00	100.00	100.00	100.00	-
C15	100.00	100.00	100.00	100.00	100.00	-
Average	99.85	98.90	98.84	98.85	98.78	98.76
Training Phase (70%)						
C1	99.62	100.00	94.74	97.30	97.13	-
C2	99.87	100.00	98.11	99.05	98.98	-
C3	99.87	100.00	98.15	99.07	99.00	-
C4	99.75	96.36	100.00	98.15	98.03	-
C5	99.75	98.15	98.15	98.15	98.01	-
C6	99.49	93.44	100.00	96.61	96.40	-
C7	100.00	100.00	100.00	100.00	100.00	-
C8	99.75	100.00	95.92	97.92	97.81	-
C9	100.00	100.00	100.00	100.00	100.00	-
C10	99.87	100.00	98.18	99.08	99.02	-
C11	100.00	100.00	100.00	100.00	100.00	-
C12	99.75	96.30	100.00	98.11	98.00	-
C13	100.00	100.00	100.00	100.00	100.00	-
C14	100.00	100.00	100.00	100.00	100.00	-
C15	100.00	100.00	100.00	100.00	100.00	-
Average	99.85	98.95	98.88	98.90	98.83	98.77

Table 2. Cont.

Labels	Accuracy	Precision	Recall	F-Score	MCC	Kappa Score
Testing Phase (30%)						
C1	99.70	100.00	94.44	97.14	97.03	-
C2	99.41	100.00	90.91	95.24	95.05	-
C3	100.00	100.00	100.00	100.00	100.00	-
C4	100.00	100.00	100.00	100.00	100.00	-
C5	100.00	100.00	100.00	100.00	100.00	-
C6	99.41	90.00	100.00	94.74	94.57	-
C7	100.00	100.00	100.00	100.00	100.00	-
C8	99.70	100.00	96.15	98.04	97.90	-
C9	99.70	95.00	100.00	97.44	97.32	-
C10	100.00	100.00	100.00	100.00	100.00	-
C11	100.00	100.00	100.00	100.00	100.00	-
C12	99.70	95.83	100.00	97.87	97.74	-
C13	100.00	100.00	100.00	100.00	100.00	-
C14	100.00	100.00	100.00	100.00	100.00	-
C15	100.00	100.00	100.00	100.00	100.00	-
Average	99.84	98.72	98.77	98.70	98.64	98.73

The training accuracy (TA) and validation accuracy (VA) attained by the AFC-HPODTL approach on dataset 1 is demonstrated in Figure 5. The experimental outcome shows that the AFC-HPODTL methodology gained maximal values of TA and VA. Specifically, the VA seemed to be higher than the TA.

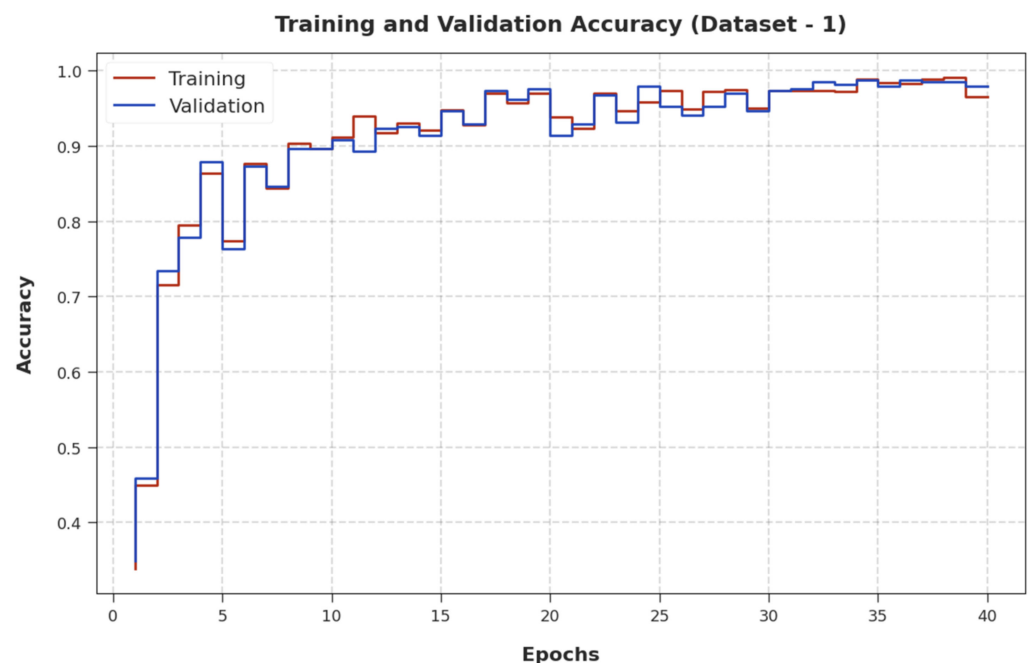


Figure 5. TA and VA analysis of AFC-HPODTL approach under dataset 1.

The training loss (TL) and validation loss (VL) achieved by the AFC-HPODTL system on dataset 1 are established in Figure 6. The experimental outcome inferred that the AFC-HPODTL approach achieved the lowest values of TL and VL. Specifically, the VL seemed to be lower than TL.



Figure 6. TL and VL analysis of AFC-HPODTL approach under dataset 1.

Table 3 and Figure 7 provide a comprehensive comparison study of the AFC-HPODTL model with existing models [28] on dataset 1. The results show that the NASNetMobile and MobileNetV1 models showed worse fruit classification results. Following, the Inception v3 model gained a slightly increased classification outcome. Then, the DenseNet121, VGG-16, and MobileNetV2 models reported moderately closer classification results. However, the AFC-HPODTL model gained maximum performance with $accu_y$, $prec_n$, $reca_l$, $F1_{score}$, and kappa score of 99.84%, 98.72%, 98.77%, 98.70%, and 98.73%, respectively.

Table 3. Comparative analysis of AFC-HPODTL approach with existing algorithms on dataset 1 [28].

Methods	Accuracy	Precision	Recall	F1-Score	Kappa-Score
DenseNet121	95.10	93.74	95.03	94.22	94.47
NASNetMobile	85.98	88.58	87.24	85.96	85.95
VGG-16	95.44	94.69	94.40	94.42	94.91
MobileNetV1	86.83	88.11	86.45	85.87	86.40
InceptionV3	90.35	90.71	89.45	89.32	88.55
MobileNetV2	96.07	95.44	95.75	95.79	95.22
AFC-HPODTL	99.84	98.72	98.77	98.70	98.73

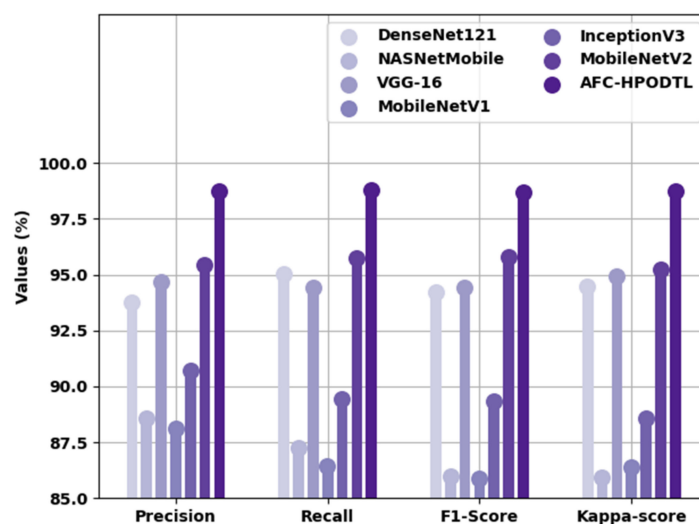


Figure 7. Comparative analysis of AFC-HPODTL approach under dataset 1.

4.2. Result Analysis on Dataset 2

Dataset 2 (D2) is an Indian fruit dataset that involves 12 classes as illustrated in Table 4. This is a balanced dataset, whereas all the classes have 1000 images, resulting from 12,000 images in total. All the images were obtained with various lighting, angles, and background conditions. The dataset is openly accessible in [27]. A few sample images from dataset 2 are demonstrated in Figure 8.

Table 4. Dataset 2 details.

Labels	Name	No. of Instances
C1	Bad apple	1000
C2	Good apple	1000
C3	Bad banana	1000
C4	Good banana	1000
C5	Bad guava	1000
C6	Good guava	1000
C7	Bad lime	1000
C8	Good lime	1000
C9	Bad orange	1000
C10	Good orange	1000
C11	Bad pomegranate	1000
C12	Good pomegranate	1000
Total No. of Instances		12,000

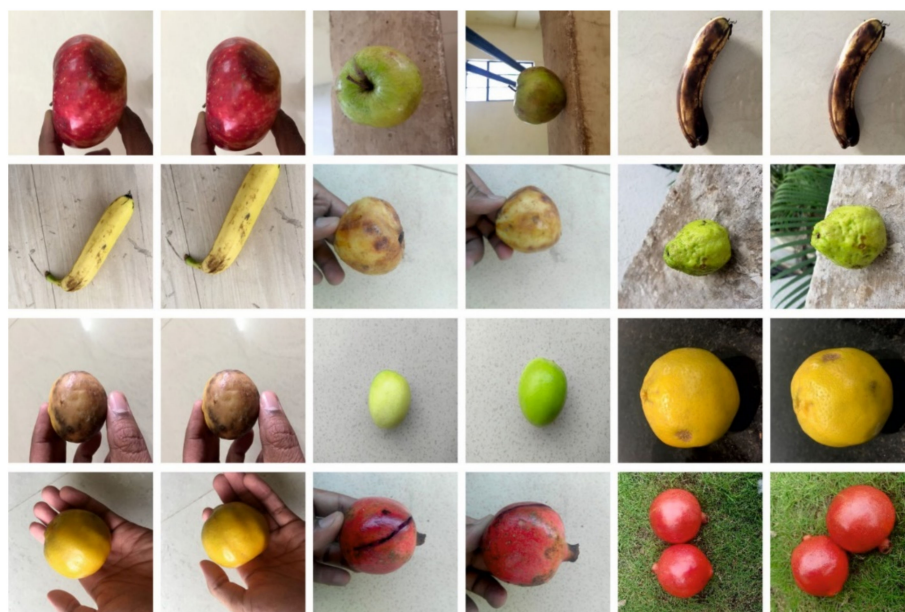


Figure 8. Sample images from dataset 2.

Figure 9 depicts a set of confusion matrices created by the AFC-HPODTL approach on the test dataset 2. The figure shows that the AFC-HPODTL algorithm effectively categorized the images into 12 fruit classes in all datasets.

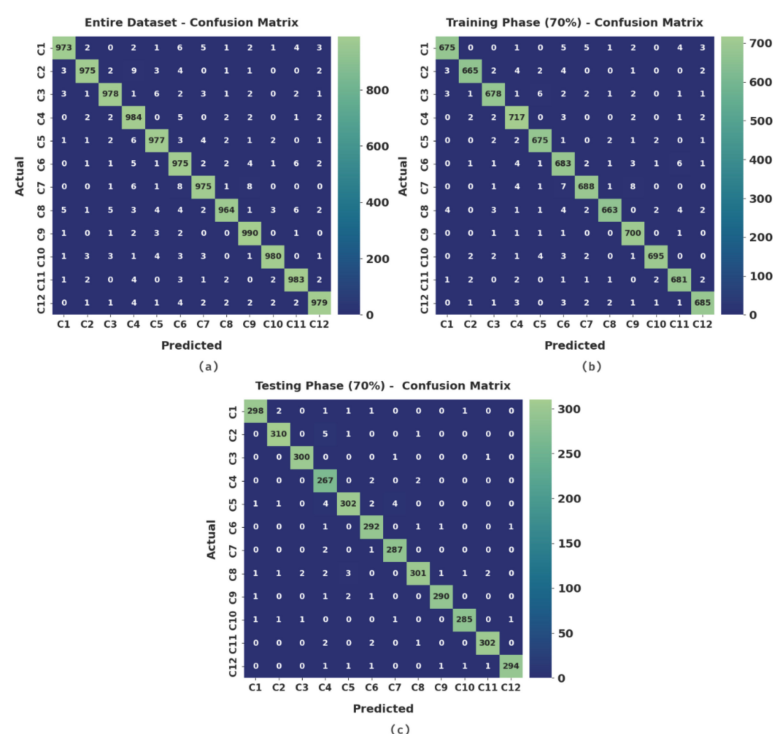


Figure 9. Confusion matrices of AFC-HPODTL approach on dataset 2: (a) entire dataset, (b) 70% of TR data, and (c) 30% of TS data.

Table 5 demonstrates the overall fruit classification outcomes of the AFC-HPODTL approach obtained under dataset 2. The outcomes show that the AFC-HPODTL model obtained effectual classification outcomes on all datasets. For instance, with entire dataset, the AFC-HPODTL approach classified 15 classes with average $accu_y$, $prec_n$, $reca_l$, F_{score} , MCC, and kappa score of 99.63%, 97.79%, 97.78%, 97.78%, 97.58%, and 97.57%, correspondingly. Next, with 70% of TR data, the AFC-HPODTL algorithm classified 15 classes with average $accu_y$, $prec_n$, $reca_l$, F_{score} , MCC, and kappa score of 99.61%, 97.70%, 97.68%, 97.68%, 97.47%, and 97.47%, correspondingly. Similarly, with 30% of the TS dataset, the AFC-HPODTL methodology classified 15 classes with average $accu_y$, $prec_n$, $reca_l$, F_{score} , MCC, and kappa score of 99.67%, 97.99%, 98.02%, 98%, 97.82%, and 97.82%, correspondingly.

Table 5. Result analysis of AFC-HPODTL approach under various measures on dataset 2.

Labels	Accuracy	Precision	Recall	F-Score	MCC	Kappa Score
Entire Dataset						
C1	99.65	98.48	97.30	97.89	97.70	-
C2	99.68	98.58	97.50	98.04	97.86	-
C3	99.67	98.19	97.80	98.00	97.81	-
C4	99.51	95.81	98.40	97.09	96.83	-
C5	99.61	97.60	97.70	97.65	97.44	-
C6	99.42	95.68	97.50	96.58	96.27	-
C7	99.61	97.79	97.50	97.65	97.43	-
C8	99.58	98.57	96.40	97.47	97.25	-
C9	99.72	97.63	99.00	98.31	98.16	-
C10	99.74	98.89	98.00	98.44	98.30	-
C11	99.68	97.81	98.30	98.05	97.88	-
C12	99.69	98.39	97.90	98.15	97.98	-
Average	99.63	97.79	97.78	97.78	97.58	97.57

Table 5. Cont.

Labels	Accuracy	Precision	Recall	F-Score	MCC	Kappa Score
Training Phase (70%)						
C1	99.62	98.40	96.98	97.68	97.48	-
C2	99.68	98.66	97.36	98.01	97.84	-
C3	99.58	97.84	97.13	97.48	97.26	-
C4	99.57	96.76	98.35	97.55	97.32	-
C5	99.68	97.68	98.40	98.04	97.86	-
C6	99.35	95.26	97.02	96.13	95.78	-
C7	99.55	97.73	96.90	97.31	97.07	-
C8	99.62	98.66	96.65	97.64	97.44	-
C9	99.69	97.09	99.29	98.18	98.01	-
C10	99.73	98.86	97.89	98.37	98.22	-
C11	99.64	97.42	98.27	97.84	97.65	-
C12	99.65	98.00	97.86	97.93	97.74	-
Average	99.61	97.70	97.68	97.68	97.47	97.47
Testing Phase (30%)						
C1	99.72	98.68	98.03	98.35	98.20	-
C2	99.67	98.41	97.79	98.10	97.92	-
C3	99.86	99.01	99.34	99.17	99.10	-
C4	99.36	93.36	98.52	95.87	95.57	-
C5	99.44	97.42	96.18	96.79	96.49	-
C6	99.61	96.69	98.65	97.66	97.45	-
C7	99.75	97.95	98.97	98.46	98.32	-
C8	99.50	98.37	95.86	97.10	96.83	-
C9	99.78	98.98	98.31	98.64	98.52	-
C10	99.78	98.96	98.28	98.62	98.50	-
C11	99.75	98.69	98.37	98.53	98.40	-
C12	99.78	99.32	98.00	98.66	98.54	-
Average	99.67	97.99	98.02	98.00	97.82	97.82

The TA and VA attained by the AFC-HPODTL approach on dataset 2 are demonstrated in Figure 10. The experimental outcome shows that the AFC-HPODTL methodology gained maximal values of TA and VA. Specifically, the VA appeared superior to the TA.

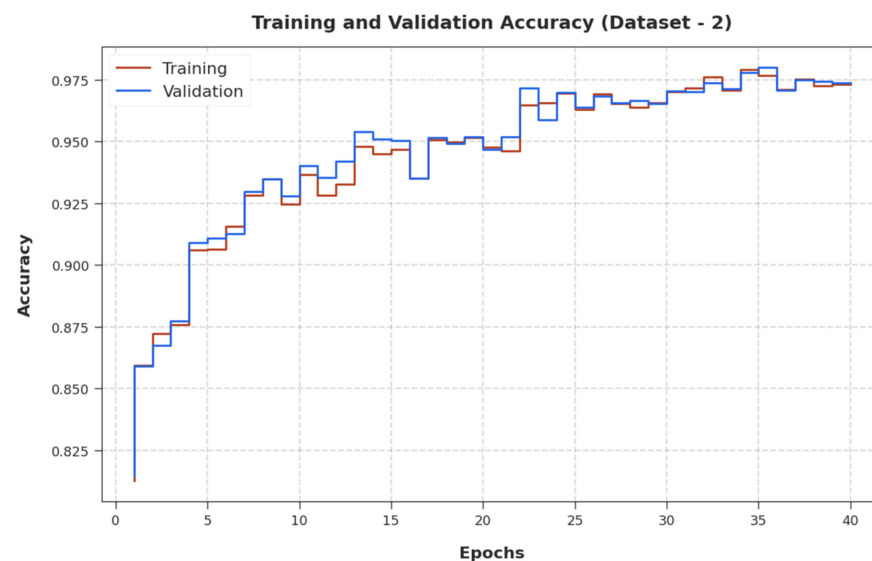


Figure 10. TA and VA analysis of AFC-HPODTL approach under dataset 2.

The TL and VL achieved by the AFC-HPODTL system on dataset 2 are established in Figure 11. The experimental outcome exposed that the AFC-HPODTL approach achieved the lowest values of TL and VL. Specifically, the VL seemed to be lesser than TL.



Figure 11. TL and VL analysis of AFC-HPODTL approach under dataset 2.

Table 6 and Figure 12 illustrate a comprehensive comparison analysis of the AFC-HPODTL algorithm with existing approaches on dataset 2 [28]. The outcomes demonstrate that the NASNetMobile and MobileNetV1 techniques demonstrated worse fruit classification results. The Inception v3 model gained somewhat superior classification outcomes. Likewise, the DenseNet121, VGG-16, and MobileNetV2 approaches reported moderately closer classification results. Eventually, the AFC-HPODTL system showed a higher performance with $accu_y$, $prec_n$, $recal$, $F1_{score}$, and kappa score of 99.67%, 97.99%, 98.02%, 98%, and 97.82%, correspondingly.

Table 6. Comparative analysis of AFC-HPODTL approach with existing algorithms on dataset 2 [28].

Methods	Accuracy	Precision	Recall	F1-Score	Kappa-Score
DenseNet121	95.56	95.89	96.41	95.64	96.04
NASNetMobile	93.89	93.40	94.07	93.03	93.04
VGG-16	95.74	97.02	95.74	96.79	96.51
MobileNetV1	86.29	87.80	87.13	87.06	85.46
InceptionV3	94.91	95.17	96.19	95.44	95.90
MobileNetV2	96.20	96.46	96.89	96.98	96.03
AFC-HPODTL	99.67	97.99	98.02	98.00	97.82

From the detailed results and discussion, it is apparent that the AFC-HPODTL model accomplished maximum fruit classification results over the other models.

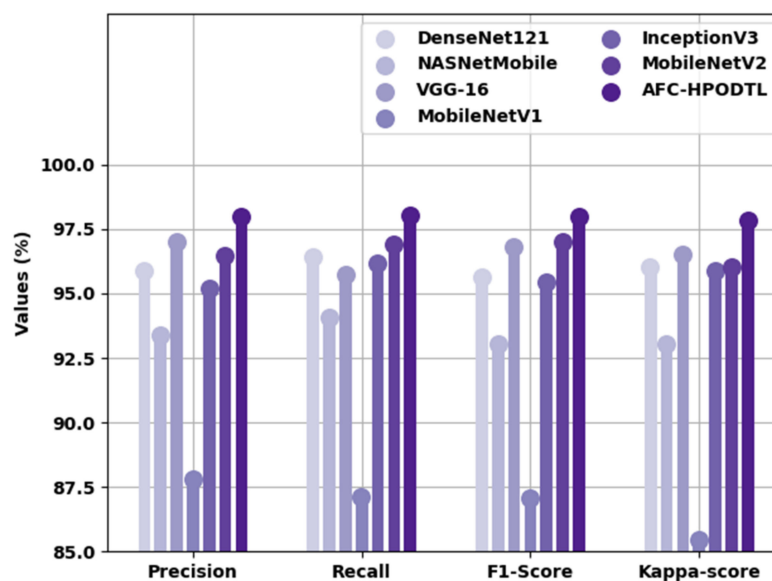


Figure 12. Comparative analysis of AFC-HPODTL approach on dataset 2.

5. Conclusions

In this study, a new AFC-HPODTL model was developed for the automatic identification and classification of fruits. The presented AFC-HPODTL model comprises a series of processes, namely pre-processing, DenseNet169 feature extraction, Adam optimizer, RNN classification, and AOA hyperparameter optimization. For feature extraction, the Adam optimizer with deep transfer learning-based DenseNet169 model is used and the AOA-RNN model is utilized for the classification of fruits. The performance validation of the presented AFC-HPODTL model was carried out using a benchmark dataset and the results reported promising performance over recent state-of-the-art approaches with maximum accuracy of 99.84% and 99.67% on datasets 1 and 2, respectively. The results demonstrated that the presented model has the effectual ability to classify fruits in real time. As a part of future scope, hybrid DL models can be integrated into the AFC-HPODTL model for enhanced classification performance. In addition, the presented model can be extended to the examination of fruit quality assessment in future. Moreover, the computational complexity of the proposed model can be examined in our future work.

Author Contributions: Conceptualization, S.K.; data curation, K.S.; formal analysis, K.S.; funding acquisition, A.A.; investigation, S.K.; methodology, K.S. and S.K.; project administration, A.K.D. and K.S.; resources, A.J.M.J. and Y.K.Y.; validation, A.H.A.; visualization, Y.K.Y.; writing—original draft, S.K. and K.S.; writing—review and editing, S.K. All authors have read and agreed to the published version of the manuscript.

Funding: The work is supported by the Ministry of Science and Higher Education of the Russian Federation (Government Order FENU-2020-0022).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Altaheri, H.; Alsulaiman, M.; Muhammad, G. Date fruit classification for robotic harvesting in a natural environment using deep learning. *IEEE Access* **2019**, *7*, 117115–117133. [\[CrossRef\]](#)
2. Chen, X.; Zhou, G.; Chen, A.; Pu, L.; Chen, W. The fruit classification algorithm based on the multi-optimization convolutional neural network. *Multimed. Tools Appl.* **2021**, *80*, 11313–11330. [\[CrossRef\]](#)

3. Khan, R.; Debnath, R. Multi class fruit classification using efficient object detection and recognition techniques. *Int. J. Image Graph. Signal Process.* **2019**, *11*, 1. [\[CrossRef\]](#)
4. Abdusalomov, A.; Mukhiddinov, M.; Djuraev, O.; Khamdamov, U.; Whangbo, T.K. Automatic Salient Object Extraction Based on Locally Adaptive Thresholding to Generate Tactile Graphics. *Appl. Sci.* **2020**, *10*, 3350. [\[CrossRef\]](#)
5. Yoon, H.; Kim, B.H.; Mukhriddin, M.; Cho, J. Salient region extraction based on global contrast enhancement and saliency cut for image information recognition of the visually impaired. *KSII Trans. Internet Inf. Syst. (TIIS)* **2018**, *12*, 2287–2312.
6. Nasir, I.M.; Bibi, A.; Shah, J.H.; Khan, M.A.; Sharif, M.; Iqbal, K.; Nam, Y.; Kadry, S. Deep learning-based classification of fruit diseases: An application for precision agriculture. *CMC-Comput. Mater. Contin.* **2021**, *66*, 1949–1962. [\[CrossRef\]](#)
7. Naranjo-Torres, J.; Mora, M.; Hernández-García, R.; Barrientos, R.J.; Fredes, C.; Valenzuela, A. A review of convolutional neural network applied to fruit image processing. *Appl. Sci.* **2020**, *10*, 3443. [\[CrossRef\]](#)
8. Macanhã, P.A.; Eler, D.M.; Garcia, R.E.; Junior, W.E.M. Handwritten feature descriptor methods applied to fruit classification. In *Information Technology-New Generations*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 699–705.
9. Siddiqi, R. Fruit-classification model resilience under adversarial attack. *SN Appl. Sci.* **2022**, *4*, 1–22. [\[CrossRef\]](#)
10. Ukwuoma, C.C.; Zhiguang, Q.; Bin Heyat, M.B.; Ali, L.; Almaspoor, Z.; Monday, H.N. Recent Advancements in Fruit Detection and Classification Using Deep Learning Techniques. *Math. Probl. Eng.* **2022**, *2022*, 9210947. [\[CrossRef\]](#)
11. Hossain, M.S.; Al-Hammadi, M.; Muhammad, G. Automatic fruit classification using deep learning for industrial applications. *IEEE Trans. Ind. Inform.* **2018**, *15*, 1027–1034. [\[CrossRef\]](#)
12. Rojas-Aranda, J.L.; Nunez-Varela, J.I.; Cuevas-Tello, J.C.; Rangel-Ramirez, G. Fruit classification for retail stores using deep learning. In *Mexican Conference on Pattern Recognition*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 3–13.
13. Gill, H.S.; Khehra, B.S. Hybrid classifier model for fruit classification. *Multimed. Tools Appl.* **2021**, *80*, 27495–27530. [\[CrossRef\]](#)
14. Xue, G.; Liu, S.; Ma, Y. A hybrid deep learning-based fruit classification using attention model and convolution autoencoder. *Complex Intell. Syst.* **2020**, 1–11. [\[CrossRef\]](#)
15. Kumari, R.S.S.; Gomathy, V. Fruit classification using statistical features in svm classifier. In Proceedings of the 2018 4th International Conference on Electrical Energy Systems (ICEES), Chennai, India, 7–9 February 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 526–529.
16. Zhang, Y.D.; Dong, Z.; Chen, X.; Jia, W.; Du, S.; Muhammad, K.; Wang, S.H. Image based fruit category classification by 13-layer deep convolutional neural network and data augmentation. *Multimed. Tools Appl.* **2019**, *78*, 3613–3632. [\[CrossRef\]](#)
17. Xiang, Q.; Wang, X.; Li, R.; Zhang, G.; Lai, J.; Hu, Q. Fruit image classification based on Mobilenetv2 with transfer learning technique. In Proceedings of the 3rd International Conference on Computer Science and Application Engineering, Sanya, China, 22–24 October 2019; pp. 1–7.
18. Mohakud, R.; Dash, R. Survey on hyperparameter optimization using nature-inspired algorithm of deep convolution neural network. In *Intelligent and Cloud Computing*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 737–744.
19. Shen, M.; Yang, J.; Li, S.; Zhang, A.; Bai, Q. Nonlinear Hyperparameter Optimization of a Neural Network in Image Processing for Micromachines. *Micromachines* **2021**, *12*, 1504. [\[CrossRef\]](#)
20. Guo, Y.; Li, J.Y.; Zhan, Z.H. Efficient hyperparameter optimization for convolution neural networks in deep learning: A distributed particle swarm optimization approach. *Cybern. Syst.* **2020**, *52*, 36–57. [\[CrossRef\]](#)
21. Ezzat, D.; Ella, H.A. GSA-DenseNet121-COVID-19: A hybrid deep learning architecture for the diagnosis of COVID-19 disease based on gravitational search optimization algorithm. *arXiv* **2020**, arXiv:2004.05084.
22. Lodhi, B.; Kang, J. Multipath-DenseNet: A Supervised ensemble architecture of densely connected convolutional networks. *Inf. Sci.* **2019**, *482*, 63–72. [\[CrossRef\]](#)
23. Soydaner, D. A comparison of optimization algorithms for deep learning. *Int. J. Pattern Recognit. Artif. Intell.* **2020**, *34*, 2052013. [\[CrossRef\]](#)
24. Rehman, A.U.; Malik, A.K.; Raza, B.; Ali, W. A hybrid CNN-LSTM model for improving accuracy of movie reviews sentiment analysis. *Multimed. Tools Appl.* **2019**, *78*, 26597–26613. [\[CrossRef\]](#)
25. Abualigah, L.; Yousri, D.; Abd Elaziz, M.; Ewees, A.A.; Al-Qaness, M.A.; Gandomi, A.H. Aquila optimizer: A novel meta-heuristic optimization algorithm. *Comput. Ind. Eng.* **2021**, *157*, 107250. [\[CrossRef\]](#)
26. Rocha, A.; Hauagge, D.C.; Wainer, J.; Goldenstein, S. Automatic fruit and vegetable classification from images. *Comput. Electron. Agriculture* **2010**, *70*, 96–104. [\[CrossRef\]](#)
27. Meshram, V.; Thanomliang, K.; Ruangkan, S.; Chumchu, P.; Patil, K. FruitsGB: Top Indian fruits with quality. *IEEE Dataport* **2020**. [\[CrossRef\]](#)
28. Shahi, T.B.; Sitaula, C.; Neupane, A.; Guo, W. Fruit classification using attention-based MobileNetV2 for industrial applications. *PLoS ONE* **2022**, *17*, e0264586. [\[CrossRef\]](#)