FMDB *Transactions on Sustainable Computer Letters*

Automated Fruit Identification Using Modified AlexNet Feature Extraction-Based FSSATM Classifier

M. Arunadevi Thirumalraj[1,*], B. Rajalakshmi[2], B. Santosh Kumar[3], S. Gopikha[4], Amarilys González García[5]

[1]Department of Computer Science and Engineering, Karunya Institute of Technology and Science, Coimbatore, Tamil Nadu, India.
[1]Department of Computer Science and Business Management, Saranathan College of Engineering, Tiruchirappalli, Tamil Nadu, India.
[2,3]Department of Computer Science and Engineering, New Horizon College of Engineering, Bengaluru, Karnataka, India.
[4]Department of Information Technology, St. Joseph's College of Engineering, Chennai, Tamil Nadu, India.
[5]Department of Research and Development, Placental Histotherapy Center, Havana, Cuba.
aruna.devi96@gmail.com[1], dr_rajalakshmi_imprint@yahoo.com[2], skumars1803@gmail.com[3], gopikha.in@gmail.com[4], agonzalez@miyares-cao.cu[5]

Abstract: Automating fruit detection is a continuous challenge due to its complexity. Because fruit varieties and subtypes may vary by geography, manually classifying fruits can be challenging. The Fruit-360 dataset was categorised using convolutional neural network-based techniques (e.g., VGG16, Inception V3, MobileNet, and ResNet-18) in several recent publications. Unfortunately, the 131 fruit classifications are not comprehensive enough to be of much service. Furthermore, the computational efficiency of these models was poor. With 90,483 sample images and 131 fruit categories, our innovative, comprehensive, and reliable study can recognise and predict them. A modified AlexNet-based strategy, combined with an effective classifier, was employed to bridge the research gap effectively. The upgraded AlexNet uses the Golden Jackal Optimisation Algorithm (GJOA) to determine the optimal feature extraction technique tuning after processing the input images. Moreover, the Fruit Shift Self-Attention Transform Mechanism (FSSATM) serves as the final classifier. This transform mechanism combines spatial position encoding (SPE) with a spatial feature extraction module (SFE) to increase the transformer's accuracy.

Keywords: Golden Jackal Optimisation Algorithm; Fruit Shift Self Attention; Transform Mechanism; Modified AlexNet; Automated Fruit Identification; Spatial Feature; Extraction Module; Spatial Position Encoding.

Received on: 13/01/2025, **Revised on:** 23/03/2025, **Accepted on:** 06/05/2025, **Published on:** 22/11/2025

Journal Homepage: https://www.fmdbpub.com/user/journals/details/FTSCL

DOI: https://doi.org/10.69888/FTSCL.2025.000486

Cite as: M. A. Thirumalraj, B. Rajalakshmi, B. S. Kumar, S. Gopikha, and A. G. García, "Automated Fruit Identification Using Modified AlexNet Feature Extraction-Based FSSATM Classifier," *FMDB Transactions on Sustainable Computer Letters*, vol. 3, no. 4, pp. 198–212, 2025.

Copyright © 2025 M. A. Thirumalraj *et al.*, licensed to Fernando Martins De Bulhão (FMDB) Publishing Company. This is an open access article distributed under CC BY-NC-SA 4.0, which allows unlimited use, distribution, and reproduction in any medium with proper attribution.

1. Introduction

We should be very concerned about the meals we eat, given the phenomenal rise in the current population. Nutritionists promote fruits as a rich source of nutrients, and most individuals incorporate them into their regular diets [1]. Over the years, several

*Corresponding author.

approaches to fruit identification using computer vision technology have been developed. The goal of these methods is to classify and differentiate between different kinds of fruits in a picture library [2]. Both academics and industry professionals agree that fruit classification is a difficult and divisive topic. Grocery store employees may quickly determine the price of a specific fruit, for instance, by classifying it [3]. Additionally, nutritional recommendations are beneficial because they guide customers in selecting appropriate foods to meet their health and nutritional needs [4]. Automated fruit packaging is a common practice in most food processing plants. Because different regions within the same nation have distinct fruit varieties and subtypes, the laborious process of manually classifying fruits remains an ongoing challenge. This huge difference is based on the necessary components found in fruits, which vary by population and location [5]. The use of artificial intelligence is rapidly expanding across all facets of society, and the food and agriculture sectors are no exception. Among the various fields that have found applications for AI are medicine, teaching, farming, and many more [6].

Artificial Intelligence (AI) has found several applications in healthcare, including the diagnosis of skin cancer, the identification of various anatomical objects, the prediction of neurodevelopmental abnormalities in children, and mental health [7]. The world's population is growing, the climate is changing, and humans have created other environmental risks, all of which threaten agriculture and may ultimately lead to increased food demand [8]. In this regard, it appears that computer vision-driven Agtech businesses and artificial intelligence (AI) are saviours, as they expedite various procedures, including harvesting, quality control, picking and packaging, sorting, and grading [9]. Fruits are particularly vulnerable because of their fragility and rapid spoilage. Improper and delayed fruit grading, categorisation, and identification by unskilled personnel result in the loss of 30–35% of the collected fruits [10]. Classifying fruits is the most important and challenging part of buying and selling fruit. Anyone involved in the fruit trade needs to be well-versed in the many kinds of fruit to set fair prices. Therefore, it's important to know how to identify various fruit kinds [11]. Marketing and dataset analysis are just a few of the many fields that have found success with AI and ML techniques [12]. Consequently, several researchers have been interested in applying proven methods to automated fruit categorisation, driven by rapid advances in machine learning, especially over the last decade [13].

Form, size, texture, and colour are among the external quality descriptors researchers often use in their studies. Most of the proposed classifiers either failed to identify any fruit at all accurately or could identify only a specific type of fruit [14]. We now have a plethora of tools for sorting, identifying, and grading seeds, fruits, and vegetables. Various fruit classes have prompted the proposal of distinct categorisation schemes. Identifying and categorising fruit illnesses was the focus of several researchers [15]. The previous model was based on the VGG19 architecture. When classifying illnesses in fruits, their model achieved nearly 99% accuracy. In this study, FSSATM is used for classification, while modified AlexNet is used for feature extraction. Afterwards, a more efficient noise-removal technique, the IBFTF algorithm, is developed. To improve classification accuracy, GJOA is used to fine-tune the proposed models.

2. Related Works

Using Augment Yolov3, Karthikeyan et al. [16] established a new YOLOAPPLE system for classifying apples into three categories: normal, damaged, and red delicious. To achieve better outcomes in the next iteration, consider grabbing Apple's backdrop. To maintain feature loss preferences during training, they enrich Yolov3 with additional spatial functions. Yolov3 is enhanced by incorporating a backbone and utilising a feature pyramid network before the object detector to add spatial pyramid pooling features. Ultimately, the fully linked layer will determine whether an apple is normal, damaged, or a Red Delicious. Comparing the Augment Yolov3 model to the traditional Yolov3 and Yolov4 deep learning models, the former achieves a mean average precision of 90.13%, while the latter enables a multi-class detection and identification system. To improve localisation and achieve exact multi-item detection, experimental results were obtained using a newly constructed object recognition model trained on a dataset. To determine the potential harvest of Citrus unshiu fruit, Kwon et al. [17] investigated the optimal height for UAV photography. Based on the regular diameter of C. unshiu fruit (46.7 mm), we found that a resolution of about 5 pixels/cm is required for meaningful calculation of fruit size. We obtained these photos from five different sources. Furthermore, we found that when comparing photos with and without histogram equalisation, the fruit count estimate was significantly higher with the latter. Normal image estimates for photos taken at 30 m height are 73, 55, and 88 fruits, respectively. Nevertheless, the image estimations of 88, 71, and 105 were histogram equalised. There are a total of 141 fruits, 88 fruits, and 124 fruits.

The estimated value was comparable to that of histogram equalisation when using a Vegetation Index like IPCA, although there was a discrepancy between the I1 estimate and the actual yields. For future studies on uncrewed aerial vehicle (UAV) applications in citrus fruit yield, our results provide a valuable database for reference. In this way, the system can produce reliable findings, and using flying stages, such as UAVs, can be a step towards implementing this type of system across ever-greater territories at a reasonable cost. In their study, Raihen and Akter [18] employed a variety of ML and DL techniques, including: logistic regression, XGBoost, LightGBM, Random Forest, Decision Tree, K-Nearest Neighbour, Support Vector Machine (SVM), and Artificial Neural Network (ANN). Traditional measures are employed to evaluate the study's effectiveness. Of the fourteen models, two use the caret, H2O, neuralnet, and keras packages; the other, LightGBM, achieves

an accuracy of 90.30%, while AdaBoost achieves 98.40%. Both models also have ROC curve scores around 90%. A high-density genetic map of the F2 population was developed by Shu et al. [19]. It encompassed linkage groups and included 1,347 bin markers. The F2 population's trait segregation study reveals that a single locus controls the colour of both immature and mature fruits. The locus controlling the colour of immature fruits was found to be tightly linked to bin markers 19 on chromosome 1 and 849 on chromosome 6. It has been suggested that the inactive shikimate kinase-like two gene could be a potential regulator of immature fruit colour, and that the capsanthin-capsorubin synthase gene could be responsible for the yellow hue in HNUCC16 pepper fruits, based on the conversion of the two bin markers into dCAPS markers. In summary, the results provide new insights into how colour develops and offer a tool for molecular breeding and genetic enhancement of pepper fruit colour.

Patel and Patil [20] proposed an integrated grading system and an intelligent system for automated detection and categorisation of banana fruit sickness. The proposed system uses deep learning models, machine learning algorithms, and computer vision techniques to identify and grade illnesses accurately. Using image processing methods, the system collects crucial information from pictures of banana fruits, which are then fed into a trained classification model. To classify bananas into several disease groups, the classification model employs state-of-the-art algorithms. The complex grading system also takes into account the size, colour, and texture of the sick fruit, among other factors, to determine its severity and quality. High disease-detection accuracy and accurate banana grading are two key outcomes of the experiments, demonstrating that the proposed strategy is effective. Banana growers and other agricultural stakeholders may save time and money with an automated device that controls diseases in plantations. In citrus fruit identification algorithms, Lin et al. [21] address issues of low detection accuracy and frequent missed detections, particularly under occlusion conditions. It presents AG-YOLO, a network that combines contextual information through attentiveness. Using YOLO leverages its ability to gather comprehensive contextual information from neighbouring scenes. Furthermore, it incorporates a Global Context Fusion Module (GCFM) that enhances the model's ability to recognise obstructed targets by allowing local and global information to interact and fuse via self-attention. To analyse AG-YOLO's performance, an independent dataset was compiled containing over 8,000 outdoor photos. A subset of 957 photos that met the requirements for occlusion scenarios involving citrus fruits was selected after a careful screening procedure.

This dataset covers a wide range of complex situations, including occlusion, extreme occlusion, overlap, and extreme overlap. On this dataset, AG-YOLO performed exceptionally well, achieving P-values of 90.6%, mAP@50 of 83.2%, and mAP@50:95 of 60.3%. The effectiveness of AG-YOLO is confirmed by these measures, which outperform the current popular object identification algorithms. By successfully addressing occlusion detection, AG-YOLO achieved a frame rate of 34.22 FPS without compromising detection accuracy. Notably, both speed and accuracy are preserved at 34.22 FPS, demonstrating a considerably faster performance. This is especially true while dealing with the intricacies of occlusion difficulties. When it comes to object detection, AG-YOLO clearly outperforms previous models in efficiently handling severe occlusions, achieving high localisation accuracy, low missed-detection rates, and fast detection speed. This highlights its role as a dependable and effective solution to the challenge of handling heavy occlusions in object recognition. To identify several mango diseases and differentiate them from healthy specimens, Reddy et al. [22] employed an image classification technique. The preprocessing phase consists of two main steps: background removal and contrast enhancement. Histogram equalisation is a technique for improving picture contrast. Using instance segmentation, a crucial procedure, is the next step after the preprocessing stage. A Convolutional Recurrent Neural Network (CNN_FOA) Optimiser is fed the collected radiomic properties. The CNN FOA is used for categorising mango photos. Experimental verification and validation have shown that the projected perfect crops provide optimal results with a 97% accuracy rate.

To identify when olive fruits of different cultivars are ripe in an orchard setting, Zhu et al. [23] suggest a new method called Olive-EfficientDet. For more accurate fruit maturity stage classification, Olive-EfficientDet uses a convolutional block attention module (CBAM) that is logically incorporated into the backbone network. When it comes to occlusion and overlap of olive fruits, the upgraded system is built to fuse semantic linkages and position information across multiple layers fully. The experimental findings demonstrated that the suggested Olive-EfficientDet offers a reliable method for determining when olive fruits are ripe in orchard settings. For olive varieties 'Frantoio,' 'Ezhi 8′,' 'Leccino,' and 'Picholine,' the mean average precision (mAP) of fruit maturity detection was 94.60%, 93.50%, 93.75%, and 96.05%, respectively. The average detection time per picture was 337 ms, and the model size was a mere 32.4 MB. Furthermore, the Olive-EfficientDet demonstrates remarkable flexibility in handling complex lighting, occlusion, and overlapping in difficult, uncontrolled orchard settings. Using Olive-EfficientDet and other cutting-edge technologies to detect ripeness, researchers conducted comparative trials. In a comparison of four cultivars, Olive-EfficientDet outperformed SSD, EfficientDet, YOLOv3, and Faster R-CNN in mAP for detecting ripe olive fruits. With its impressive model size and speed, Olive-EfficientDet achieved the highest mAP for detecting ripe olive fruits in orchard settings. This work can serve as a technical basis for olive harvesting robots to detect when fruits are ripe. It has been addressed by Vinisha and Bod [24] in the development of an innovative tumour detection system that relies on UNets trained on fruit flies (TFFbU). Trypetidae fruit flies were also more fit after using the UNet pooling module. The best results have usually come from there. The initial step in training the system was to use the datasets typically sourced from the internet.

Consequently, the training mistakes are removed in the TFFbU's main layer before data cleaning. Then, the UNet dense layer is employed for tumour detection and segmentation.

Finally, the constructed TFFbU is tested and validated by running the proposed model in MATLAB. Several metrics, including recall, accuracy, precision, Dice coefficient, and Jaccard index, are used to evaluate the model. The novel TFFbU model being planned can also segment and forecast different tumour types. By incorporating a loss function into the U-Net decoder, Li et al. [25] propose a canopy labelling method well-suited to the U-Net and a lightweight segmentation network. This approach significantly decreases the computational complexity needed for large-scale canopy segmentation. Datasets collected from two separate lychee orchards over two seasons were used to verify the practicality and efficacy of the proposed strategy. Compared to the basic U-Net model, the enhanced U-Net achieved a higher average recognition rate of 90.98% and a lower floating-point operations per second (FLOPs) of 50.86%. Since it does not require repetitive sampling of the same region, the suggested model is more efficient than prior YOLACT-based instance segmentation approaches. It also outperformed popular semantic segmentation models, such as Deeplabv3+ and ResNet50-U-Net, under identical experimental conditions. With the number of sampled tiny pictures decreased from 194 to 78 in the same region, total efficiency improved by 148%, yielding superior segmentation results. To facilitate precise orchard management, the suggested approach can be used to extract and locate the crown of a lychee tree.

3. Proposed System

3.1. The Fruit-360 Dataset

With 67,692 images in the training set, besides 22,688 in the test set, Fruit-360 has a total of 90,483 fruit photographs [26]. There are 131 distinct fruit kinds in the collection, and each fruit has a single fruit picture. The dimension of these photos is 100 × 100 pixels. The number of photographs in the training and test sets varies slightly across fruit types; nonetheless, it is common to have around images per fruit variety. A twenty-second video of fruit being gently spun by a motor is used to obtain these photos, and the frames/images are extracted from that movie. To set the stage for the capture, a blank piece of white paper is utilised. Then, a dedicated algorithm gets to work removing the fruit's backdrop. Because the backdrop might be affected by changing light intensity, it must be eliminated.

3.2. Pre-Processing

Due to external environmental influences, the fruit dataset images often exhibit low contrast and irregular brightness. While increasing contrast can make objects more visible, it can also amplify noise, blur edges, and produce indistinct features, all of which can reduce the accuracy of fruit detection. An image improvement technique built on the IBFTF algorithm was offered as a solution to this problem. This technique enhances visual effects and adds richness to images, which is important for further recognition studies. The model combined concepts of picture enhancement and image denoising using a wavelet transform to address the problems mentioned above successfully. First, a wavelet decomposition is used to obtain the noisy image's LF and HF coefficients. The Retinex image improvement algorithm with improved bilateral filtering strengthens the LF coefficients, while an improved threshold function method de-noises the HF coefficients. The processed LF and HF coefficients are then subjected to an inverse wavelet transformation to produce the rebuilt visual. To improve the technique, which successfully addresses the previously identified issues. The precise activities of the algorithm in this study are as follows:

- The low- and high-frequency components of the noise image are computed via wavelet decomposition.
- The enhanced image-enhancing method handles the LF coefficients
- The improved threshold function method handles the HF coefficient
- The reconstructed image is obtained through wavelet rebuilding of both LF and HF coefficients
- The rebuilt image is processed through a piecewise linear alteration, yielding the enhanced image

Algorithm 1 outlines the procedure depicted in the above steps.

Algorithm 1: IBFTF image augmentation
Input: Image $S(x, y)$
Rotate the noisy image into LF $W_{ø}$ and HF W_{φ}^{i} coefficients
The image is launched $R(x, y)$
$W_{ø}$ uses heightened bilateral filtering $I_D(i, j)$ dispensation
$W(i, j, k, l), W(i, j, k, l)$ sets the limit P, and the unique bilateral filtering window size is $2P + 1$
The heightened function $\omega_{J,K}$ is used to estimate the HF wavelet coefficient in three parts
Process $W_{ø}$ using $f(i, j)$ $\text{three} - \text{segment piecewise linear transformation}$

$W_{ø}$ and W_{φ}^{i} are reconstructed using 2D discrete wavelet $f(x, y)$ Output reconstructed image

3.3. Feature Extraction using Modified AlexNet Model

Currently, one of the hotspots in fruit recognition is AlexNet, the most used convolutional neural network. Due to limitations in AlexNet, achieving an accurate diagnosis is quite challenging. The large variance, nonlinearity, and nonstationarity make the input pictures difficult. As a result, internal covariate shifting occurs, leading to differences in the input distributions of the AlexNet layers. This can make it extremely difficult and time-consuming to achieve parameter training precision, which requires an appropriate setup. The FC layer in a conventional AlexNet is in the last three layers: fc6, fc7, and fc8. An FC is made up of several interconnected layers [27]. A problem with AlexNet's FC layer is that it has too many trainable parameters. The following outlines the process for determining the training settings for the FC layers. There are two different kinds of FC layers in AlexNet. While the FC layers that follow (fc7 and fc8) are connected to other FC layers, the initial FC layer is connected to the last convolutional layer. Every scenario is examined independently.

Case 1: An FC (fc6), the subsequent equations can obtain the layer's sum of limits associated with a conveyor:

$$P_{cf} = W_{cf} + B_{cf} \quad (1)$$

$$B_{cf} = F \quad (2)$$

$$W_{cf} = F \times N \times O^2 \quad (3)$$

Where:

P_{cf} = number of parameters; W_{cf} = is linked to a conv layer; B_{cf} = How many biases are present in a conv-linked FC layer, where O is the size of the output picture from the preceding layer and N is the number of kernels used in that layer. F represents the FC layer's neuron count. F=4096, N=256, and O=6 make up AlexNet's initial FC layer (fc6). Therefore,

$$W_{cf} = 4096 \times 256 \times 6^2 = 37{,}748{,}736 \quad (4)$$

$$B_{cf} = 4096$$

$$P_{cf} = W_{cf} + B_{cf} = 37{,}748{,}736 + 4096 = 37{,}752{,}832 \quad (5)$$

Case 2: If you want to know how many parameters are associated with an FC layer, you can use these equations.

$$P_{ff} = B_{ff} + W_{ff} \quad (6)$$

$$B_{ff} = F \quad (7)$$

$$W_{ff} = F_{-1} \times F \quad (8)$$

Where:

P_{cf} = sum of limits; W_{cf}= The sum of weights in the layer that accompanies an FC layer; B_{ff} = The sum of layers that is linked to an FC layer; F = The sum of neurons in the FC layer; F_{-1} = The sum of neurons in the layer just before the FC layer. In the second FC layer (fc7) of AlexNet, F is 4096, and F_{-1}= 4096. Therefore,

$$B_{ff} = F = 4096$$

$$W_{ff1} = F_{-1} \times F \times 4096 \times 4096 = 16{,}777{,}216$$

$$P_{ff1} = B_{ff} + W_{ff} = 4096 + 16{,}777{,}216 = 16{,}781{,}312$$

In the last FC layer (fc8) of F_{-1} = 4096. Therefore,
$B_{ff} = F = 1000$

$W_{ff} = F_{-1} \times F = 4096 \times 1000 = 4{,}096{,}000$

$P_{ff2} = B_{ff} + W_{ff} = 1000 + 4{,}096{,}000 = 4{,}097{,}000$

The total number of parameters in AlexNet is the sum of the parameter limits in its three FC layers.

$P_{total} = P_{cf} + P_{ff1} + P_{ff2}$

$= 37{,}752{,}832 + 16{,}781{,}312 + 4{,}097{,}000 = 58{,}631{,}144$

Upon computation, Table 1 shows 62,378,344 limits in AlexNet, with 58,631,144 training parameters originating from the final three FC layers, indicating a noteworthy percentage. Nevertheless, the overabundance of training parameters in the FC layer of AlexNet leads to overfitting, thereby increasing the model's training and testing times.

Table 1: Sums of the AlexNet perfect

Parameters	Layer Name
34,944	conv1
614,656	conv2
885,120	conv3
1,327,488	conv4
884,992	conv5
37,752,832	fc6
16,781,312	fc7
4,097,000	fc8

By examining the shortcomings of the conventional AlexNet model, this study modified the model's structure, as shown in Table 1. The updated AlexNet model is depicted in Figure 1.

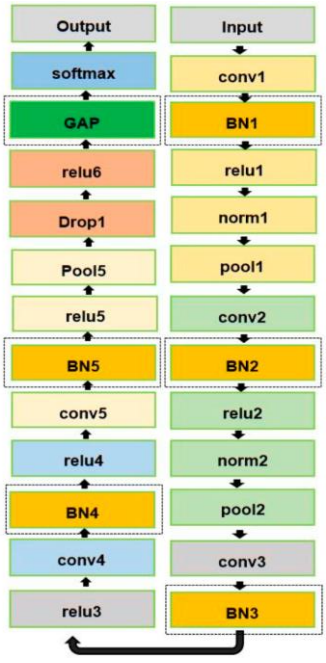

Figure 1: Modified AlexNet model

First, the GAP serves as AlexNet's fully connected layer, reducing the overall limit on training and testing time while also preventing overfitting. Second, to prevent internal covariate shift, the classic AlexNet uses a BN layer. The idea behind BN is really simple. To maintain consistent means and variances during CNN training in mini-batch mode, BN normalises the layer activations. It improves accuracy and training time while producing high-quality parameter training. The speed and training

speed of AlexNet can be significantly increased by selecting optimal hyperparameters during the CNN development process. The main hyperparameters influencing the CNN model's performance are the optimiser, activation kernels, and pooling kernels. This model employs the GJOA optimisation approach, allowing for adaptive modification of the learning rate.

3.3.1. Golden Jackal Optimisation Algorithm for Fine-Tuning

Ibrahim et al. [8] developed a programme that imitates the natural hunting patterns of golden jackals. Typically hunt together. The jackal's three phases of hunting are: (1) seeking out and approaching the prey; (2) encircling and stopping moving; and (3) lunging in the direction of the prey. Equation (9) generates a randomly distributed collection of prey site matrices during the initialisation phase:

$$\begin{bmatrix} Y_{1,1} & \cdots & Y_{1,j} & \cdots & Y_{1,n} \\ Y_{2,1} & \ddots & Y_{2,j} & \iddots & Y_{2,n} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ Y_{N-1,1} & \iddots & Y_{N-1,j} & \ddots & Y_{N-1,n} \\ Y_{N,1} & \cdots & Y_{N,j} & \cdots & Y_{N,n} \end{bmatrix} \quad (9)$$

Where n stands for dimensions and N for the number of prey populations, the following is the golden jackal's hunt mathematical model. ($|E| > 1$):

$$Y_1(t) = Y_M(t) - E.\,|Y_M(t) - rl.\,Prey(t) \quad (10)$$

$$Y_2(t) = Y_{FM}(t) - E.\,|Y_{FM}(t) - rl.\,Prey(t) \quad (11)$$

Where t is the present repetition, $Y_M(t)$ indicates jackal, $Y_{FM}(t)$ designates the site of the female; besides, Prey(t) is the site of the prey. $Y_1(t)$ and $Y_2(t)$ Are the female and male golden jackals' most recent locations known? E, or the prey's avoiding energy, is computed as follows.:

$$E = E_1.\,E_0 \quad (12)$$

$$E_1 = c_1.\left(1 - (t/T)\right) \quad (13)$$

where E_0 is a random sum in the range [–1, 1], representing the prey's initial energy; T characterises the maximum sum of repetitions; c1 is the default continuous set to 1.5; and E_1 energy. In Equations (10) and (11), $|Y_M(t) - rl \cdot Prey(t)|$ designates the distance between the golden jackal and prey; besides, "rl" is the vector of random statistics intended by the Levy flight function.

$$rl = 0.05.\,LF(y) \quad (14)$$

$$LF(y) = 0.01 \times \frac{(\mu \times \sigma)}{\left(\left|v^{\left(\frac{1}{\beta}\right)}\right|\right)} \quad \sigma = \left\{\frac{\Gamma(1+\beta)\times\sin(\pi\beta/2)}{\Gamma\left(\frac{1+\beta}{2}\right)\times\beta\times(2^{\beta-1})}\right\}^{1/\beta} \quad (15)$$

Where u and v are accidental standards in (0, 1), and b is the evasion, set to 1.5.

$$Y(t+1) = \frac{Y_1(t)+Y_2(t)}{2} \quad (16)$$

Where $Y(t+1)$ is the prey's current location as determined by the jackals. The escaping energy is reduced when the golden jackals harass their prey. The golden jackals encircling and consuming their victim are represented mathematically as follows. ($|E| \leq 1$):

$$Y_1(t) = Y_M(t) - E.\,|rl.\,Y_M(t) - Prey(t)| \quad (17)$$

$$Y_2(t) = Y_{FM}(t) - E.\,|rl.\,Y_{FM}(t) - rl.\,Prey(t)| \quad (18)$$

Algorithm 1: Golden Jackal Optimization
Inputs: The population size N and the maximum number of iterations T

Outputs: The location of prey and its fitness valueInitialize the random prey population Yi (i $= 1, 2, \ldots, N$)
Calculate the fitness values of prey
$Y1 = $ best prey individual (Male Jackal Position)
$Y2 = $ second $-$ best prey individual (Female Jackal Position)
Update the evading energy "E" using Equations (12) and (14)
If ($|E| \leq 1$) (Exploration phase)
Update the prey position using Equations (10), (11), and (16)
Update the prey position using Equations (16), (17), and (18)
end for
$t = t + 1$
end while
return Y1

3.4. Classification using FSSATM

Here, we present the suggested spectral-swin with enhanced spatial extraction (SSFE), which is broken down into four parts: the architecture, the SFE module, the SPE module, and the spectral unit.

3.4.1. Overall Construction

In this work, we develop a novel transformer-based technique for fruit categorisation called SSWT. The two main components of SSWT—the spectral swin module and the spatial feature extraction module (SFE)—enable it to solve fruit classification problems. The model receives a patch of features as input. First, the data is sent to SFE, whose convolutional layers and spatial attention module extract the initial spatial features. Subsequently, the data is compressed and sent to the module. To provide spatial structure to the data, a spatial location encoding is inserted before each s-swin transformer layer. Using linear layers produces the final classification results.

3.4.2. Spatial Feature Extraction Segment

To compensate for the transformer's shortcomings, we developed a spatial feature module to process spatial data and local characteristics. The first half utilises convolutional layers for feature extraction and batch normalisation to prevent overfitting; this is the preliminary phase of the process. Second, there's a spatial attention mechanism that should help the model pick out the most relevant data points. For the input patch cube $I \in R^{H \times W \times C}$, where $H \times W$ is the sum of bands. Each pixel space in I consists of C spectral dimensions, and forms a one-hot class vector $S = [s1, s2, s3, \cdots, sn] \in R^{1 \times 1 \times n}$, where n is the sum of classes. Firstly, the spatial features of fruit images are originally extracted, and the formula is exposed as shadows:

$$X = \mathrm{GELU}(\mathrm{BN}(\mathrm{Conv}(I))) \tag{19}$$

Where $\mathrm{Conv}(\cdot)$ represents the convolution layer. $\mathrm{BN}(\cdot)$ characterizes normalization. $\mathrm{GELU}(\cdot)$ signifies a function. The layer is exposed below:

$$\mathrm{Conv}(I) = \prod_{j=0}^{J} \left(I * W_j^{r1 \times r2} + b_j\right) \tag{20}$$

Where I is the input, J is the sum of kernels, $W_j^{r1 \times r2}$ is the jth kernel with the size of $r1 \times r2$, and b_j is the jth bias. || symbolises a chain; besides $*$, it is a convolution process. After that, spatial attention may help the model identify key locations in the data. In the case of a first-level feature map $X \in R^{H' \times W' \times C}$($H' \times W'$ is the spatial size of X), the procedure of SA is exposed in the subsequent formula:

$$S_M = \mathrm{MaxPooling}(X) \tag{21}$$

$$S_A = \mathrm{AvgPooling}(X) \tag{22}$$

$$X_{SA} = \sigma\left(\mathrm{Conv}\left(\mathrm{Concat}(S_M, S_A)\right)\right) \otimes X \tag{23}$$

Worldwide average pooling in the channel direction is referred to as Average Pooling, while worldwide maximum pooling is referred to as Maximum Pooling. "Concat" means to concatenate in the direction of the channel. So, s stands for the activation function. "⊏" means to multiply elements one by one. In Figure 2, we can see the SA structure.

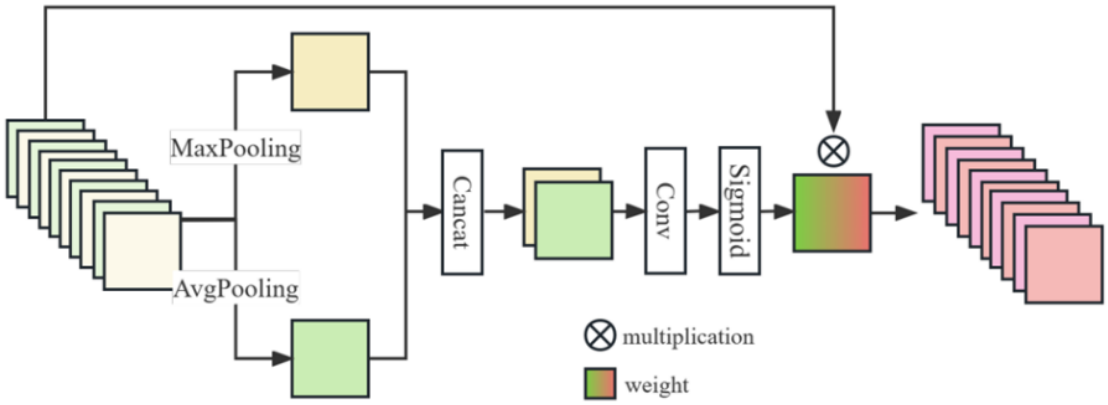

Figure 2: The assemblage of spatial care in SFE

3.4.3. Spatial Site Encoding

The input fruit pictures are transferred data, which may compromise the model's structure. A spatial position is inserted before each modifier module to specify the relative spatial locations between pixels and to preserve samples. A patch of an area serves as the input for the fruit classification algorithm; the only thing it targets for classification is the label of the centre pixel. The relevance of nearby pixels tends to diminish with increasing distance from the centre, although they can still contribute spatial information for categorising the centre pixel. Such a centrally crucial position encoding is to be learned by SPE. The definition of a patch's pixel locations is as follows:

$$\mathrm{pos}(x_i, y_i) = |x_i - x_c| + |y_i - y_c| + 1 \tag{24}$$

where (x_c, y_c) denotes the organisation of central importance to be classified. (x_i, y_i) shows where additional pixels in the dataset are located. There is a unique and crucial pixel in the middle, and the remaining pixels have varying location encodings based on their distance from the centre. The data is incorporated with learnable position encodings, allowing it to describe the spatial structure in fruit photos flexibly:

$$Y = X + \mathrm{spe}(P) \tag{25}$$

Where X is the fruit image data, P characterises the site matrix-based rendering as in Equation (25). $\mathrm{spe}(\cdot)$ is an array that may be learned; to obtain the final spatial position encoding, it uses the site matrix as a subscript. The last step is to add the location encoding to the data.

3.4.4. Spectral Swin-Transformer Segment

Transformer can handle lengthy dependencies well, but it can't extract local features. Our concept utilises a window-based multi-head architecture, inspired by Swin-Transformer. The input cannot split the window in space like Swin-T can, since it is a patch, which is often tiny in three-dimensional size. A spectral-shift window, known as spectral-window multi-head, was created for MSA, leveraging the rich data in the spectral dimension. Information may be shared between neighbouring windows via window shifting and MSA within windows, thereby enhancing local feature capture. You may use the following formula to express MSA:

$$Z = \mathrm{Attn}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_K}}\right)V \tag{26}$$

$$\psi = \mathrm{Concat}(Z_1, Z_2, \ldots, Z_h)W \tag{27}$$

The input matrices, known as queries, keys, and values, are translated into the matrices Q, K, and V. D_K denotes the dimension of K. Q and K are used to determine the attention scores. W stands for the output mapping matrix, h is the MSA head number, and y is the MSA output. It is assumed that the input size is HHW×C, where C is the sum of spectral bands and HHW is the space size. Since the size of every window is fixed to C/4, each window is split equally. Following division, the sizes of each window are [C/4, C/4, C/4, C/4]. Next, MSA is performed for each window. The window is then pushed in the spectral direction

by half a window. At each window is $[C/8, C/4, C/4, C/4, C/8]$ in size. MSA is performed again in every window. Thus, the S-W-MSA procedure through m windows is

$$Y^{(m)} = \left[\psi\left(y^{(1)}\right)\oplus\psi\left(y^{(2)}\right)\oplus,..,\oplus\psi\left(y^{(m)}\right)\right] \quad (28)$$

where $\oplus$ resources concat, $y^{(i)}$ is the statistics of the i-th window. Except for the window design, the remaining elements of the S-SwinT module—MLP and layer normalisation connections—remain unchanged compared to SwinT. The formulas shown below are in shadows:

$$\widehat{Y}^{l} = S - W - MSA\left(LN\left(Y^{l-1}\right)\right) + Y^{l-1} \quad (29)$$

$$Y^{l} = MLP\left(LN\left(\widehat{Y}^{l}\right)\right) + \widehat{Y}^{l} \quad (30)$$

$$\widehat{Y}^{l+1} = S - SW - MSA\left(LN\left(Y^{l}\right)\right) + Y^{l} \quad (31)$$

$$Y^{l+1} = MLP\left(LN\left(\widehat{Y}^{l+1}\right)\right) + \widehat{Y}^{l+1} \quad (32)$$

4. Results and Discussion

The deep learning framework PyTorch was used with an NVIDIA Tesla V100 featuring 32 GB of video RAM. Table 2 lists the simulation parameters.

Table 2: Experiment situation

Parameter Values Improvement	Experimental Environment Configuration
Intel(R) Xeon(R) Gold 6371C CPU@2.60 GHz	CPU
NVIDIA Tesla V1000 GPU32 G	GPU
32 G	RAM
100 G	Magnetic disk
PyTorch	Deep learning framework
Windows 100(64-bits)	Operating Scheme
Python 3.7.1CUDA10.1	Others

4.1. Validation of Feature Extraction Models

Tables 3 and 4 explain the experimental analysis of the proposed feature extraction model based on 70%-30% and 80%-20%.

Table 3: Validation analysis of proposed feature extraction on 70%-30%

Module	Precision	Recall	F1	Accuracy (%)
LeNet	0.8298	0.8508	0.8401	84.06
ResNet	0.8679	0.8648	0.8663	86.12
VGGNet	0.9011	0.8883	0.8947	89.78
AlexNet	0.9279	0.9109	0.9193	92.71
MAlexNet-GJO	0.9467	0.9337	0.9402	93.82

Table 3 above represents the Validation Analysis of projected feature extraction at a 70%-30% ratio. In the investigation of the LeNet module, the precision was 0.8298, the recall was 0.8508, the F1 score was 0.8401, and the accuracy was 84.06. Then, the ResNet module achieved a precision of 0.8679, a recall of 0.8648, an F1-score of 0.8663, and accuracy of 86.12%. Then, the VGGNet module achieved a precision of 0.9011, recall of 0.8883, F1-score of 0.8947, and accuracy of 89.78%.

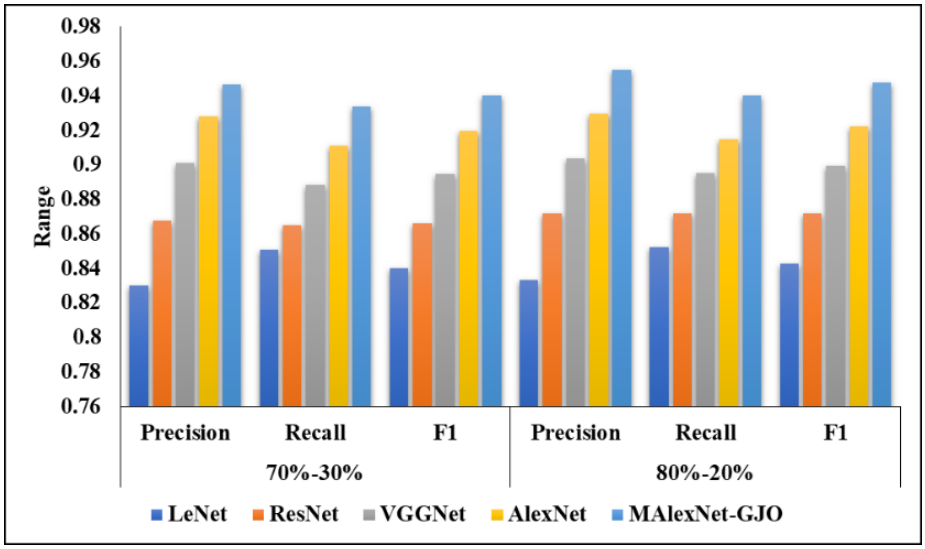

Figure 3: Visual representation of the proposed feature extraction model

Then, the AlexNet module achieved a precision of 0.9279, a recall of 0.9109, an F1-score of 0.9193, and accuracy of 92.71%. Then, the MAlexNet-GJO module achieved a precision of 0.9467, a recall of 0.9337, an F1-score of 0.9402, and accuracy of 93.82%. Figure 3 presents the graphical description of the analysis on feature extraction models.

Table 4: Validation analysis of proposed feature extraction on 80%-20%

Module	Precision	Recall	F1	Accuracy (%)
LeNet	0.8333	0.8525	0.8427	84.25
ResNet	0.8718	0.8718	0.8718	86.87
VGGNet	0.9038	0.8952	0.8995	89.56
AlexNet	0.9295	0.9148	0.9220	92.06
MAlexNet-GJO	0.9551	0.9400	0.9475	94.44

In Table 4 above, the Validation Investigation of the projected feature extraction is presented for an 80%-20% split. In the investigation of the LeNet module, the precision was 0.8333, the recall was 0.8525, the F1-score was 0.8427, and the accuracy was 84.25.

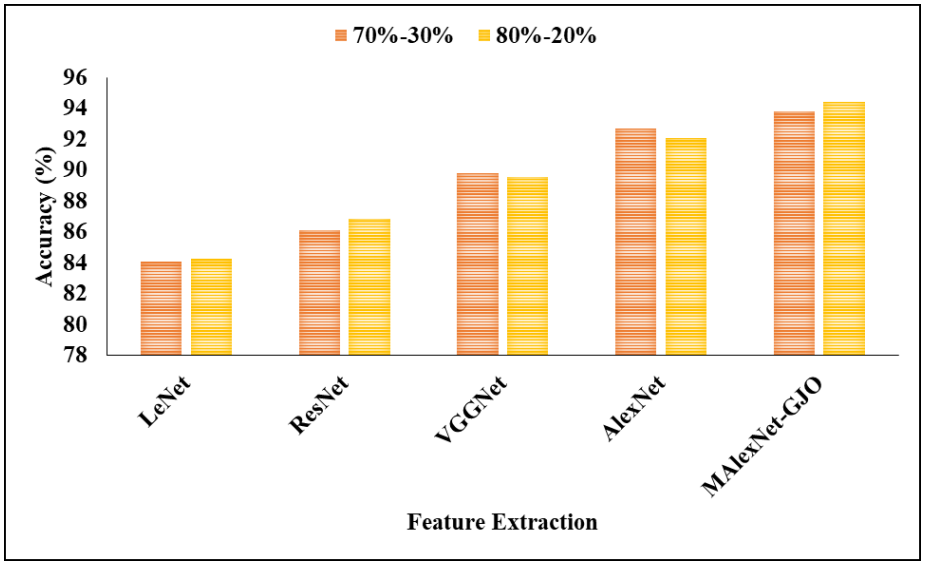

Figure 4: Graphical representation of the proposed model in terms of accuracy

Then, the ResNet module achieved precision, recall, F1-score, and accuracy of 0.8718, 0.8718, 0.8718, and 86.87%, respectively. Then, the VGGNet module achieved a precision of 0.9038, a recall of 0.8952, an F1-score of 0.8995, and accuracy of 89.56%. Then, the AlexNet module achieved a precision of 0.9295, a recall of 0.9148, an F1-score of 0.9220, and accuracy of 92.06%. Then, the MAlexNet-GJO module achieved a precision of 0.9551, a recall of 0.9400, an F1-score of 0.9475, and accuracy of 94.44%. Figure 4 presents a graphical representation of the feature extraction models' accuracy.

4.2. Verification of Proposed Classifier Model

Tables 5 and 6 present the validation results for the proposed classifier across various training-to-testing ratios.

Table 5: Validation of the proposed model for 70%-30%

Module	Precision	Recall	F1	Accuracy (%)
Multi-ScaleAlexNet	0.9163	0.9159	0.9134	91.96
TFFbU	0.8572	0.8565	0.8568	85.49
Olive-EfficientDet	0.9224	0.9281	0.9264	86.62
Self-Attention	0.9369	0.9360	0.9364	93.59
FSSATM	0.9551	0.9400	0.9475	94.44

In Table 5 above, it is characterised that the Authentication of the proposed model is 70%-30%. In the analysis of the multi-ScaleAlexNet module, the precision was 0.9163, the recall was 0.9159, the F1 score was 0.9134, and the accuracy was 91.96. Then, the TFFbU module achieved a precision of 0.8572, an F1-score of 0.8565, and an accuracy of 85.49%. Then, the Olive-EfficientDet module achieved a precision of 0.9224, recall of 0.9281, F1-score of 0.9264, and accuracy of 86.62%. Then, the Self-Attention module achieved a precision of 0.9369, recall of 0.9360, F1-score of 0.9364, and accuracy of 93.59%. Then, the FSSATM module achieved a precision of 0.9551, a recall of 0.9400, an F1-score of 0.9475, and an accuracy of 94.44%. The accuracy of the proposed classifier is given in Figure 5.

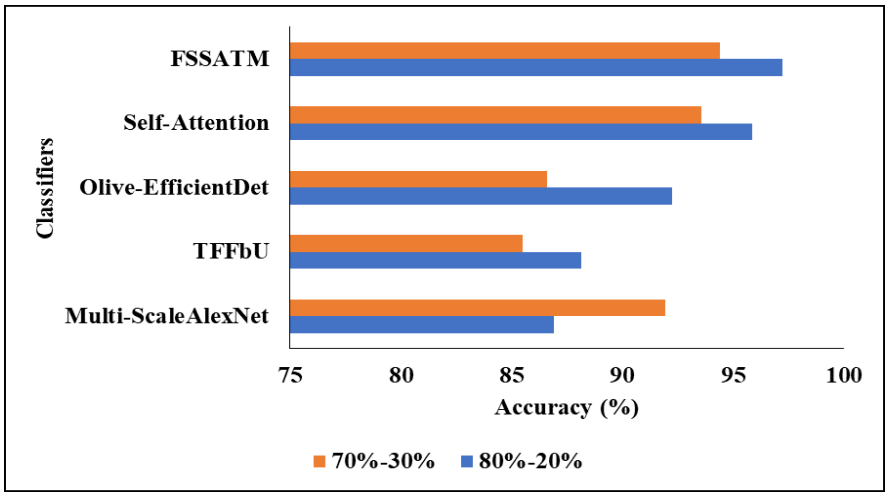

Figure 5: Accuracy analysis of the proposed classifier

In Table 6, the overhead represents the Experimentation of the projected model for an 80%-20% split. In the investigation of the multi-scale AlexNet module, the precision was 0.8639, the recall was 0.8747, the F1-score was 0.8693, and the accuracy was 86.91%. Then, the TFFbU module achieved a precision of 0.8925, a recall of 0.8714, an F1-score of 0.8818, and an accuracy of 88.16%.

Table 6: Experimentation of the proposed model for 80%-20%

Module	Precision	Recall	F1	Accuracy
Multi-ScaleAlexNet	0.8639	0.8747	0.8693	86.91
TFFbU	0.8925	0.8714	0.8818	88.16
Olive-EfficientDet	0.9216	0.9309	0.9262	92.27

Self-Attention	0.9595	0.9513	0.9554	95.86
FSSATM	0.9756	0.9715	0.9735	97.24

Then, the Olive-EfficientDet module achieved a precision of 0.9216, recall of 0.9309, F1-score of 0.9262, and accuracy of 92.27%. Then, the Self-Attention module achieved a precision of 0.9595, an F1-score of 0.9513, a recall of 0.9554, and an accuracy of 95.86%. Then, the FSSATM module achieved a precision of 0.9756, a recall of 0.9715, an F1-score of 0.9735, and an accuracy of 97.24%. Figure 6 presents a graphical representation of the proposed classifier for various training-to-testing data ratios.

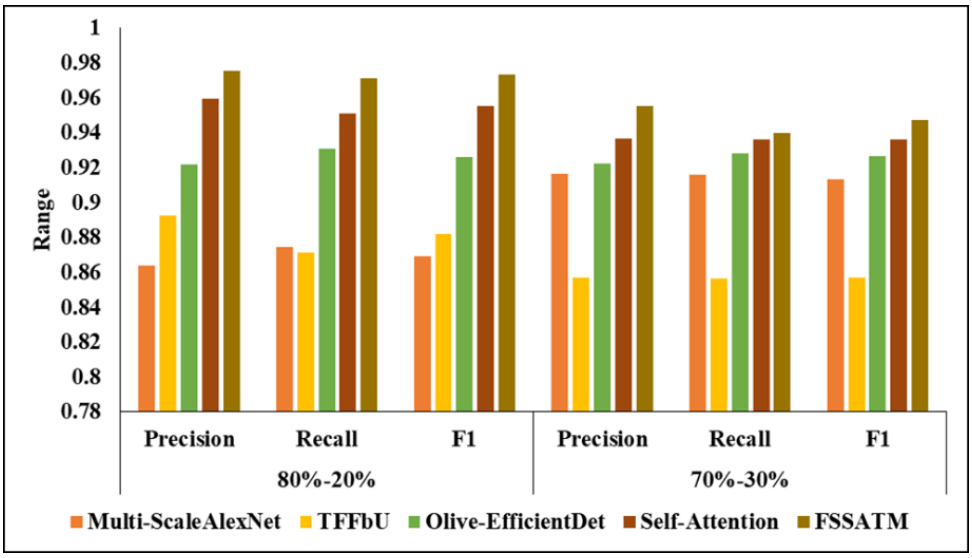

Figure 6: Visual analysis of the proposed model for different ratios

5. Conclusion

Numerous academics have attempted to utilise learning approaches to identify fruits in the Fruit-360 dataset, which comprises 90,483 sample photos and 131 fruit classifications. However, none of the earlier efforts focused on managing the entire set of 131 fruit classes and their corresponding fruit pictures. Consequently, this study presents a unique and effective attempt to identify all photos in the Fruit-360 dataset using a feature extraction and classification technique based on deep learning. Nine feature descriptors were employed to evaluate the performance of the updated AlexNet algorithm in image-based classification, with GJOA utilised for fine-tuning feature extraction. Thus, this study effort presents a modified version of the AlexNet technique that is both resilient and thorough. The model employs shifting windows as a self-attentive method to compensate for its inability to acquire local contextual data during categorisation. The learning curve and the confusion matrix were used to evaluate the performance of the tested algorithm. Here, it can be said that the suggested algorithms achieved better results than any other procedures for the given job. Consequently, the findings provide strong evidence that the proposed approach is more efficient and accurate than CNN-based methods for multiple-class image classification. Furthermore, the system demonstrated its ability to process the whole Fruit-360 dataset with reduced processing resources. The suggested feature extraction classifiers are suitable for real-time applications, as inferred from the findings, as well as economical scheme implementations. One major drawback of the suggested technique is that, depending on the dataset, it may require a different structure (e.g., a different number of levels and total inputs) to achieve greater accuracy. Consequently, a general framework for image-based categorisation issues should be implemented in future efforts.

Acknowledgment: The authors from Karunya Institute of Technology and Science, Saranathan College of Engineering, New Horizon College of Engineering, St. Joseph's College of Engineering, and the Placental Histotherapy Center gratefully acknowledge the support and resources provided by their respective institutions.

Data Availability Statement: The datasets generated and analyzed during this study are accessible upon reasonable request from the corresponding author.

Funding Statement: The authors received no financial support for the research, authorship, or publication of this manuscript.

Conflicts of Interest Statement: The authors declare that they have no conflicts of interest relevant to this work.

Ethics and Consent Statement: This study was conducted in accordance with established ethical standards. Informed consent was obtained from all participants, and all authors affirm adherence to ethical research practices.

References

1. T. B. Shahi, C. Sitaula, A. Neupane, and W. Guo, "Fruit classification using attention-based MobileNetV2 for industrial applications," *PLOS ONE*, vol. 17, no. 2, p. e0264586, 2022.
2. C. C. Ukwuoma, Q. Zhiguang, M. B. Bin Heyat, L. Ali, and Z. Almaspoor, "Recent advancements in fruit detection and classification using deep learning techniques," *Mathematical Problems in Engineering*, vol. 2022, no. 56, pp. 1–29, 2022.
3. N. E. Mimma, S. Ahmed, T. Rahman, and R. Khan, "Fruits classification and detection application using deep learning," *Scientific Programming*, vol. 2022, no. 1, p. 4194874, 2022.
4. H. S. Gill, G. Murugesan, B. S. Khehra, G. S. Sajja, G. Gupta, and A. Bhatt, "Fruit recognition from images using deep learning applications," *Multimedia Tools and Applications*, vol. 81, no. 23, pp. 33269–33290, 2022.
5. N. Ismail and O. A. Malik, "Real-time visual inspection system for grading fruits using computer vision and deep learning techniques," *Information Processing in Agriculture*, vol. 9, no. 1, pp. 24–37, 2022.
6. K. M. Albarrak, Y. Gulzar, Y. Hamid, A. Mehmood, and A. B. Soomro, "A deep learning-based model for date fruit classification," *Sustainability*, vol. 14, no. 10, p. 6339, 2022.
7. N. Aherwadi and U. Mittal, "Fruit quality identification using image processing, machine learning, and deep learning: A review," *Advances and Applications in Mathematical Sciences*, vol. 21, no. 5, pp. 2645–2660, 2022.
8. N. M. Ibrahim, D. G. I. Gabr, A. U. Rahman, S. Dash, and A. Nayyar, "A deep learning approach to intelligent fruit identification and family classification," *Multimedia Tools and Applications*, vol. 81, no. 19, pp. 27783–27798, 2022.
9. A. Majid, M. A. Khan, M. Alhaisoni, U. Tariq, N. Hussain, Y. Nam, and S. Kadry, "An integrated deep learning framework for fruits diseases classification," *Computers, Materials and Continua*, vol. 71, no. 1, pp. 1387-1402, 2022.
10. D. Hussain, I. Hussain, M. Ismail, A. Alabrah, S. S. Ullah, and H. M. Alaghbari, "A simple and efficient deep learning-based framework for automatic fruit recognition," *Computational Intelligence and Neuroscience*, vol. 2022, no. 1, p. 6538117, 2022.
11. N. Aherwadi, U. Mittal, J. Singla, N. Z. Jhanjhi, A. Yassine, and M. S. Hossain, "Prediction of fruit maturity, quality, and its life using deep learning algorithms," *Electronics*, vol. 11, no. 24, p. 4100, 2022.
12. L. G. Fahad, S. F. Tahir, U. Rasheed, H. Saqib, M. Hassan, and H. Alquhayz, "Fruits and vegetables freshness categorization using deep learning," *Computers, Materials and Continua*, vol. 71, no. 3, pp. 5083-5098, 2022.
13. H. S. Gill and B. S. Khehra, "An integrated approach using CNN-RNN-LSTM for classification of fruit images," *Materials Today: Proceedings*, vol. 51, no. 1, pp. 591–595, 2022.
14. M. Mukhiddinov, A. Muminov, and J. Cho, "Improved classification approach for fruits and vegetables freshness based on deep learning," *Sensors*, vol. 22, no. 21, p. 8192, 2022.
15. R. Verma and A. K. Verma, "Fruit classification using deep convolutional neural network and transfer learning," *in Proc. Int. Conf. on Emerging Technologies in Computer Engineering*, Jaipur, India, 2022.
16. M. Karthikeyan, T. S. Subashini, R. Srinivasan, C. Santhanakrishnan, and A. Ahilan, "YOLOAPPLE: Augment Yolov3 deep learning algorithm for apple fruit quality detection," *Signal, Image and Video Processing*, vol. 18, no. 1, pp. 119–128, 2024.
17. S. H. Kwon, K. B. Ku, A. T. Le, G. D. Han, Y. Park, J. Kim, T. T. Tuan, Y. S. Chung, and S. Mansoor, "Enhancing citrus fruit yield investigations through flight height optimization with UAV imaging," *Scientific Reports*, vol. 14, no. 1, p. 322, 2024.
18. M. N. Raihen and S. Akter, "Prediction modeling using deep learning for the classification of grape-type dried fruits," *International Journal of Mathematics and Computer in Engineering*, vol. 2, no. 1, pp. 1-12, 2024.
19. H. Shu, C. He, M. A. Mumtaz, Y. Hao, Y. Zhou, W. Jin, J. Zhu, W. Bao, S. Cheng, G. Zhu, and Z. Wang, "Fine mapping and identification of candidate genes for fruit color in pepper (Capsicum chinense)," *Scientia Horticulturae*, vol. 310, no. 2, p. 111724, 2023.
20. H. B. Patel and N. J. Patil, "An intelligent grading system for automated identification and classification of banana fruit diseases using deep neural network," *International Journal of Computing and Digital Systems*, vol. 15, no. 1, pp. 761–773, 2024.
21. Y. Lin, Z. Huang, Y. Liang, Y. Liu, and W. Jiang, "AG-YOLO: A rapid citrus fruit detection algorithm with global context fusion," *Agriculture*, vol. 14, no. 1, p. 114, 2024.
22. V. K. Reddy, A. Suhasini and V. V. S. S. S. Balaram, "Detection and classification of disease from mango fruit using convolutional recurrent neural network with metaheuristic optimizer," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 12, no. 9, pp. 321–334, 2024.

23. X. Zhu, F. Chen, X. Zhang, Y. Zheng, X. Peng, and C. Chen, “Detection the maturity of multi-cultivar olive fruit in orchard environments based on Olive-EfficientDet,” *Scientia Horticulturae*, vol. 324, no. 1, p. 112607, 2024.
24. A. Vinisha and R. Boda, “A novel framework for brain tumor segmentation using neuro Trypetidae fruit fly-based UNet,” *International Journal of Intelligent Systems and Applications in Engineering*, vol. 12, no. 1, pp. 783–796, 2024.
25. Z. Li, X. Deng, Y. Lan, C. Liu, and J. Qing, “Fruit tree canopy segmentation from UAV orthophoto maps based on a lightweight improved U-Net,” *Computers and Electronics in Agriculture*, vol. 217, no. 2, p. 108538, 2024.
26. H. Mureșan and M. Oltean, “Fruit recognition from images using deep learning,” *Acta Universitatis Sapientiae, Informatica*, vol. 10, no. 1, pp. 26–42, 2018.
27. A. Thirumalraj, V. Asha, and B. P. Kavin, “An improved Hunter-Prey optimizer-based DenseNet model for classification of hyper-spectral images,” *in AI and IoT-Based Technologies for Precision Medicine, A. Khang, Ed*., *IGI Global*, Hershey, Pennsylvania, United States of America, 2023.