Automated Fruit Identification using Modified AlexNet Feature Extraction based FSSATM Classifier

Mrs Arunadevi Thirumalraj
K.Ramakrishnan College of Technology
B. Rajalakshmi
New Horizon College of Engineering
B Santosh Kumar
New Horizon College of Engineering
S. Stephe
stephes.ece@krce.ac.in
K Ramakrishnan College of Engineering

Research Article

Keywords: Golden jackal optimization algorithm, Fruit Shift Self Attention Transform Mechanism, Modified AlexNet, Automated fruit identification, Spatial feature extraction module

Posted Date: March 13th, 2024

DOI: https://doi.org/10.21203/rs.3.rs-4074664/v1

Additional Declarations: No competing interests reported.

Automated Fruit Identification using Modified AlexNet Feature Extraction based FSSATM Classifier

[1]Mrs Arunadevi Thirumalraj, [2]Dr. B. Rajalakshmi, [3]Mr. Santosh Kumar B, [4]Dr S. Stephe.

[1]Department of Computer Science Engineering, K. Ramakrishnan College of Technology, Trichy.

[2,3]New Horizon College of Engineering, Bengaluru, Karnataka, India.

[4]Assistant Professor, Department of Electronics and Communication Engineering, K. Ramakrishnan College of Engineering, Trichy.

Mail Id: [1]aruna.devi96@gmail.com, [2]dr_rajalakshmi_imprint@yahoo.com, [3]skumars1803@gmail.com, [4]stephes.ece@krce.ac.in.

Abstract

Because fruits are complex, automating their identification is a constant challenge. Manual fruit categorisation is a difficult task since fruit types and subtypes are often location-dependent. A sum of recent publications has classified the Fruit-360 dataset using methods based on Convolutional Neural Networks (e.g., VGG16, Inception V3, MobileNet, and ResNet18). Unfortunately, out of all 131 fruit classifications, none of them are extensive enough to be used. Furthermore, these models did not have the optimum computational efficiency. Here we propose a new, robust, and all-encompassing research that identifies and predicts the whole Fruit-360 dataset, which consists of 90,483 sample photos and 131 fruit classifications. The research gap was successfully filled using an algorithm that is based on the Modified AlexNet with an efficient classifier. The input photos are processed by the modified AlexNet, which uses the Golden jackal optimisation algorithm (GJOA) to choose the best tuning of the feature extraction technique. Lastly, the classifier employed is Fruit Shift Self Attention Transform Mechanism (FSSATM). This transform mechanism is aimed to improve the transformer's accuracy and comprises a spatial feature extraction module (SFE) besides spatial position encoding (SPE). Iterations and a confusion matrix were used to validate the algorithm. The outcomes prove that the suggested tactic yields a relative accuracy of 98%. Furthermore, state-of-the-art procedures for the drive were located in the literature and compared to the built system. By comparing the results, it is clear that the newly created algorithm is capable of efficiently processing the whole Fruit-360 dataset.

Keywords: Golden jackal optimization algorithm; Fruit Shift Self Attention Transform Mechanism; Modified AlexNet; Automated fruit identification; Spatial feature extraction module.

Introduction

We should be very worried about the meals we eat because of the present population rate's phenomenal rise. Nutritionists promote fruits as a great source of nutrients, and most individuals include them in their regular diets [1]. There have been a number of approaches to fruit identification using computer vision technology throughout the years. 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 swiftly determine the price of a certain fruit, for instance, by determining its class [3]. In addition, nutritional recommendations are helpful since they guide customers choose the right food types to meet their health and nutrition needs [4]. Automated packaging of fruits is a common practice in most food processing plants. Because different regions of the same nation have different fruit kinds and subtypes, the laborious process of manually classifying fruits is 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 there are other environmental risks that humans have created, all of which threaten agriculture and might ultimately cause food demand to rise [8]. In this regard, it would appear that the computer vision-driven Agtech business and artificial intelligence (AI) are saviours, since they expedite a number of 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 results in the loss of 30–35% of 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 fruit kinds in order to set fair prices. Therefore, it's important to know how to identify various fruit kinds [11].

Marketing, dataset analysis are just a few of the many fields that have found success with the use of AI and ML techniques [12]. Consequently, several researchers have been interested in applying proven methods to automated fruit categorisation because to the fast advancements in learning, especially in the last 10 years [13]. Form, size, texture, and colour are some of the external quality descriptors that researchers often used in their studies. Most of the proposed classifiers either failed to accurately identify any fruit at all or were only able to identify a certain 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 it came to illness classification for fruits, they said that their suggested model achieved an accuracy of almost 99%.

In this study, FSSATM is employed for classification, whereas modified AlexNet is employed for feature extraction. Afterwards, a more efficient technique for noise removal is developed: the IBFTF algorithm. To increase the classification accuracy, GJOA is utilised to fine-tune the suggested models. The following is the framework for the remainder of the paper: Second, the relevant literature is reviewed; third, the technique is discussed; fourth, the results are analysed; and to end, Section 5 accomplishes the study.

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. In order to get better outcomes in the following iteration, grab cut the apple's backdrop. To keep feature loss preferences intact during training, they enrich Yolov3 with additional spatial function. Yolov3 is enhanced by including the backbone and by utilising the feature pyramid network prior to the object detector to add spatial pyramid pooling features. In the end, the fully linked layer will determine if an apple is normal, damaged, or red delicious. Comparing the Augment Yolov3 model to the traditional Yolov3, Yolov4 deep learning models, the former obtains a mean average precision of 90.13% while the latter allows for a multi-class detection and identification system. In order to improve the localisation process and achieve exact multi-item detection, the experimental results were derived using a freshly constructed object recognition perfect that was trained on dataset.

In order to determine the potential harvest of Citrous unshiu fruit, Kwon et al., [17] looked into the best 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 m. Furthermore, we discovered that when comparing photos with and without histogram equalisation, fruit count estimate was much improved 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 estimate 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 unmanned aerial vehicle (UAV) field studies on citrous fruit yield, our results offer a useful database. In this way, the system is able to produce reliable findings, and using flying stages like UAVs can be a step towards implementing this type of perfect 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 used to assess the effectiveness of the study. Out of the fourteen models, two use the caret, H2O, neuralnet, and keras packages; the other, LightGBM, has an accuracy of 90.30%, while the other, AdaBoost, achieves 98.40%. Both models also have ROC curve scores around 90%.

A high-density genetic map of the F2 populace 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 that controls the colour of immature fruits was found to be tightly linked to bin markers 19 on chromosome 1 and 849 on chromosome 6, respectively. It has been suggested that the inactive shikimate kinase-like 2 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, according to the conversion of the two bin markers into dCAPS markers. In sum, the results provide fresh information on how colour develops and provide a tool for molecular breeding and genetic enhancement of pepper fruit colour.

An integrated grading system and an intelligent system for automated banana fruit sickness detection and categorisation have been proposed by Patel & Patil [20]. The proposed system uses deep learning models, machine learning algorithms, and computer vision techniques to accurately identify and grade illnesses. 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 accuracy in disease detection and accurate banana grading are two outcomes of the experiments that show the suggested strategy is effective. Banana growers and other agricultural stakeholders may save time and money with an automated device that controls diseases in plantations.

When it comes to citrous fruit identification algorithms, Lin et al. [21] tackles the problems of poor detection accuracy and frequent missed detections, especially in occlusion conditions. It presents AG-YOLO, a network that combines contextual information through attentiveness. Using YOLO takes use of its capability 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 through self-attention processes. For the goal of analysing AG-YOLO's performance, an independent dataset was collected that had more than 8,000 outdoor photos. A subset of 957 photos that fit the requirements for occlusion scenarios of citrous fruits was obtained after a careful screening procedure. Covering a wide variety of complicated situations, this dataset contains examples of occlusion, extreme occlusion, overlap, and extreme overlap. On this dataset, AG-YOLO performed exceptionally well, with a P-value of 90.6%, a mAP@50 of 83.2%, and a 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 tackling the problem of occlusion detection, AG-YOLO was able to reach a frame rate of 34.22 FPS without sacrificing detection accuracy. Impressively, the speed and accuracy are both preserved at 34.22 FPS, demonstrating a considerably quicker 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 terms of efficient handling of severe occlusions, as well as high localisation accuracy, low missed detection rates, and fast detection speed. This emphasises its function as a dependable and effective answer to the problem of dealing with heavy occlusions in object recognition.

In order to identify several mango diseases and differentiate them from healthy specimens, Suhasini et al., [22] used an image classification technique. Two main steps make up the preprocessing phase: removing the background and improving the contrast. 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) Optimizer is fed the collected radiomic properties. The CNN FOA is employed for the purpose of categorising the mango photos. Experimental verification and validation have shown that the projected perfect crops optimal results with a 97% accuracy rate.

In order 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 olive fruits, the upgraded is built to fully fuse semantic linkages and position information from multiple layers. The experimental findings demonstrated that the suggested Olive-EfficientDet offers a reliable way to determine 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 when faced with complicated lighting, occlusion, and overlap in difficult and uncontrolled orchard settings. Using Olive-EfficientDet and other cutting-edge technologies for detecting when fruit is ripe, researchers ran comparative trials. In a comparison of four different cultivars, Olive-EfficientDet outperformed SSD, EfficientDet, YOLOv3, and Faster RCNN in terms of mAP for detecting when olive fruits are ripe. With its impressive model size and speed, Olive-EfficientDet achieved the highest mAP for detecting when olive fruits are ripe in orchard settings. This work can serve as a technical basis for olive harvesting robots to detect when fruits are ripe, and it has been addressed by Vinisha and Bod [24].with the purpose of developing 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 typically used datasets sourced from the internet. Consequently, the training mistakes are removed in the TFFbU's main layer prior to 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. A number of metrics, including recall, accuracy, precision, Dice, and Jaccard, are used to estimate the model. The novel TFFbU model that is being planned can also segment and forecast different types of tumours.

By incorporating a loss function into the U-Net decoder, Li et al. [25] suggest a canopy labelling method that is well-suited to 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 the course of two seasons were used to verify the practicality and efficacy of the suggested strategy. Compared to the basic model U-Net, the enhanced U-Net had a higher average recognition rate of 90.98%

and a reduced quantity of floating-point operations per second (FLOPs) of 50.86%. Since it does not need repetitive sampling of the same region, the suggested model is more efficient than prior instance segmentation approaches based on YOLACT. It also beat popular semantic segmentation models like Deeplabv3 + and ResNet50-U-Net under the identical experimental conditions. With a drop in the number of sampled tiny pictures from 194 to 78 for the same region, total efficiency was improved by 148% and superior segmentation results were achieved. To help with precise orchard management, plants, the suggested approach may be utilised to extract and find 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 amount of photographs in the training set and the test set varies slightly among fruit types; nonetheless, it is common to have around images supplied for each 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 the changing light intensity, it has to be eliminated.

3.2. Pre-processing

Because of external environmental influences, fruit dataset images frequently have low contrast and irregular brightness. While increasing contrast can make objects more visible, it can also magnify noise in the image, blur edges, and produce indistinct features, all of which can lower 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 study. The model combined the concepts of picture enhancement and image de-noising using the wavelet transform approach in order to successfully handle the aforementioned problems. 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 transform to produce the rebuilt visual. In order to improve technique is utilised, which successfully addresses the issues that were previously identified. The precise activities of the algorithm in this study are as shadows:

1. The low and high frequencies of the noise image are computed through wavelet decomposition;
2. The enhanced image enhancing method handles the LF coefficients;
3. The improved threshold function method handles the HF coefficient;
4. The reconstructed image is obtained through wavelet rebuilding of both LF and HF coefficients;

5. The rebuilt image is processed through a piecewise linear alteration, yielding the enhanced image.

Algorithm 1 depicts the procedure outline portrayed in the above steps.

```
Algorithm 1: IBFTF image augmentation
Input: Image $S(x, y)$
  1. Rot the noisy image into LF $W_{\emptyset}$ and HF $W_{\varphi}^{i}$ coefficients
  2. The image is launched $R(x, y)$
  3. $W_{\emptyset}$ uses heightened bilateral filtering $I_D(i, j)$ dispensation
  4. $W(i, j, k, l), W(i, j, k, l)$ sets the limit $P$, and the unique bilateral filtering window size
     is $2P + 1$
  5. The heightened function $\omega_{J,K}$
     $is\ used\ to\ estimate\ the\ HF\ wavelet\ coefficient\ in\ three\ parts$
  6. Process $W_{\emptyset}$ using $f(i, j)\ three - segment\ piecewise\ linear\ transformation$
  7. $W_{\emptyset}$ and $W_{\varphi}^{i}\ are\ reconstructed\ using\ 2D\ discrete\ wavelet\ f(x, y)$
  8. $Output\ reconstructed\ image$
```

3.3. Feature Extraction using Modified AlexNet Model

Currently, one of the hotspots in fruit recognition is AlexNet, the most used convolutional neural network. Owing to some limitations in AlexNet, it is quite difficult to obtain good outcomes in fruitful diagnosis. The large variance, nonlinear, and nonstationary properties make the input pictures difficult. As a result, internal covariate shifting takes place and the input distributions of the AlexNet layers vary from one another. This can make it extremely difficult and time-consuming to achieve precision in parameter training, which calls for appropriate setup. The FC layer in a conventional AlexNet is found in the last three levels, or fc6, fc7, and fc8. An FC is made up of several interconnected layers [27].

An issue with AlexNet's FC layer is that it has an excessive number of training parameters. The following is the process for figuring out the FC layers' training settings. There are two different kinds of FC layers in AlexNet. While the FC layers that come after (fc7 and fc8) are connected to other FC layers, the initial FC to the last conv layer. Every scenario is examined independently.

Case 1: An FC (fc6) layer's sum of limits associated to a conv layer can be intended by the subsequent equations:

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

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

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

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 layer that is accompanying to an FC layer; B_{ff} = The sum of layer that is linked to an FC layer; F = The sum of neurons in the FC layer; F_{-1} = The sum of neurons in the just before 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 limits in each of 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 displays 62,378,344 limits in AlexNet, with 58,631,144 training parameters originating from the final three FC layers of AlexNet, indicating a noteworthy percentage. Nevertheless, the overabundance of training parameters in the FC layer of AlexNet results in overfitting and lengthens the training and testing times of the model. By examining the shortcomings of the conventional AlexNet model, this study changed the model's structure. The updated model of AlexNet is depicted.Table 1. Sum of limits 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

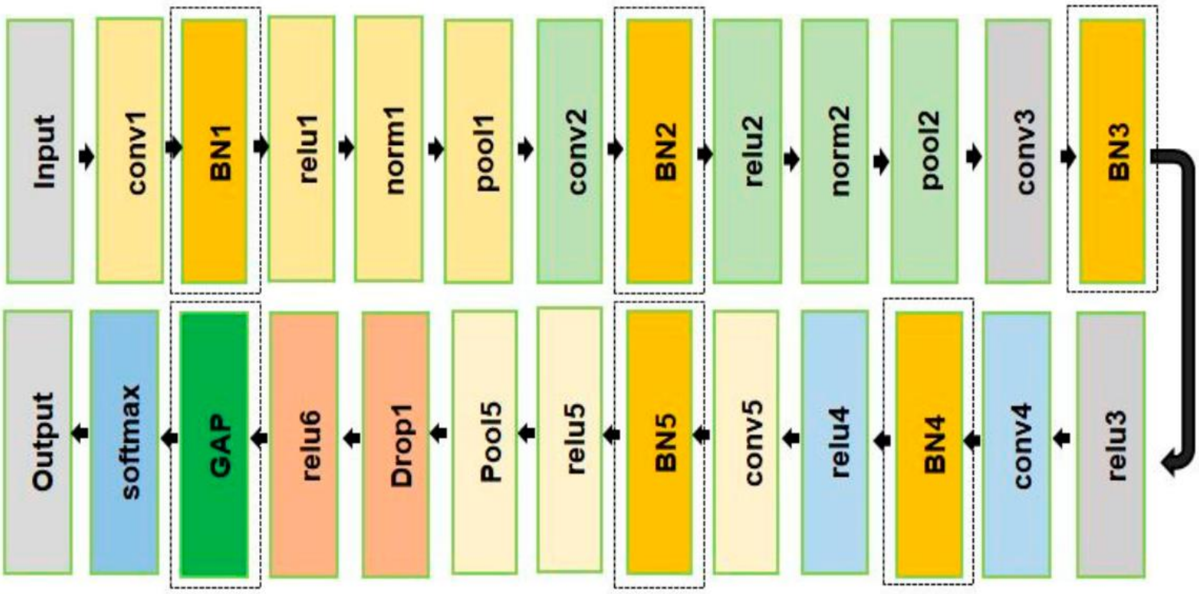

Figure 1. Modified AlexNet model.

First, the GAP takes the role of AlexNet's fully connected layer, so reducing the overall sum of limits, training, and testing period while also preventing overfitting. Second, to stop this internal covariate shifting, the classic AlexNet adopts the BN layer. The idea behind BN is really simple. In order to maintain consistent means and variances during CNN training in micro batch mode, BN applies the normalisation transform to the layer activations. It expedites the accuracy and training time while producing good parameter training. The improved AlexNet speed, and training speed may all be considerably increased by choosing the best hyper-parameters for the model throughout the CNN perfect development process. The main hyper-parameters influencing the CNN model's performance are the optimizer, activation kernels, besides pooling kernels. This model employs the GJOA optimisation approach, and it allows for adaptive modification of the learning rate.

3.3.1. Golden Jackal Optimization Algorithm for fine-tuning

Muhammad Ansari developed programme that imitates the natural hunting patterns of golden jackals. A typically hunt together. The jackal's three phases of hunting are: (1) seeking out and approaching the prey; (2) encircling besides stops moving; besides (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 prey. $Y_1(t)$ and $Y_2(t)$ are the female and male golden jackals' most recent locations. 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 characterizes 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 among the golden jackal and prey besides "rl" is the vector of random statistics intended by the Levy flight function.

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

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

where u and v are accidental standards in (0, 1) besides b is the evasion continuous 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 is represented mathematically as follows. (|E| ≤ 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
$Inputs$: $The\ population\ size\ N\ and\ maximum\ sum\ of\ iterations\ T$

```
$Outputs: The\ location\ of\ prey\ and\ its\ fitness\ value$
$Calculate\ the\ fitness\ values\ of\ prey$
$Y1 = best\ prey\ individual\ (Male\ Jackal\ Position)$
$Y2 = second\ best\ prey\ individual\ (Female\ Jackal\ Position)$
$Update\ the\ evading\ energy\ “E”\ using\ Equations\ (12)\ and\ (14)$
$If\ (|E| \leq 1)\ (Exploration\ phase)$
$Update\ the\ prey\ position\ using\ Equations\ (10), (11), and\ (16)$
$Update\ the\ prey\ position\ using\ Equations\ (16), (17), and\ (18)$
$end\ for$
$t = t + 1$
$end\ while$
$return\ Y1$
```

3.4. Classification using FSSATM

Here, we will present the suggested spectral-swin with enhanced spatial extraction (SSFE), breaking it down into four parts: the architecture as a whole, 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)—are what allow it to solve fruit classification problems. The model receives a patch of features as input. First, the data is sent into SFE, whose convolution layers and spatial attention module extracts the first spatial features. Subsequently, the data is compressed and sent into module. To provide spatial structure to the data, a spatial location encoding is inserted before each s-swin transformer layer. By using linear layers, the final classification results are produced.

3.4.2. Spatial Feature Extraction Segment

To make up for transformer's shortcomings, we built a spatial feature module to process spatial data and local characteristics. The first half uses convolutional layers for feature extraction and batch normalisation to avoid 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 $I \in R^{\mathrm{H \times W \times C}}$, where $H \times W$ is the sum of bands. Each pixel space in I contains of C spectral dimensions besides 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, besides the formula is exposed as shadows:

$$X = GELU(BN(Conv(I))) \quad (19)$$

where $Conv(\cdot)$ represents the convolution layer. $BN(\cdot)$ characterizes normalization. $GELU(\cdot)$ signifies function. The layer is exposed below:

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

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. || symbolizes chain, besides $*$ is convolution process.

After that, a spatial attention may help the model identify key locations in the data. In Figure 2, we can see the SA structure. In the case of a first-level feature map $X \in R^{H\prime\times W\prime\times C}(H' \times W'$ is the spatial size of X), the procedure of SA is exposed in the subsequent formula:

$$S_M = MaxPooling(X) \quad (21)$$

$$S_A = AvgPooling(X) \quad (22)$$

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

Worldwide average pooling in the channel direction is called AvgPooling, while worldwide maximum pooling is called MaxPooling. "Concat" means to concatenate in the direction of the channel. So, s stands for the activation function. "⊏" means to multiply elements one by one.

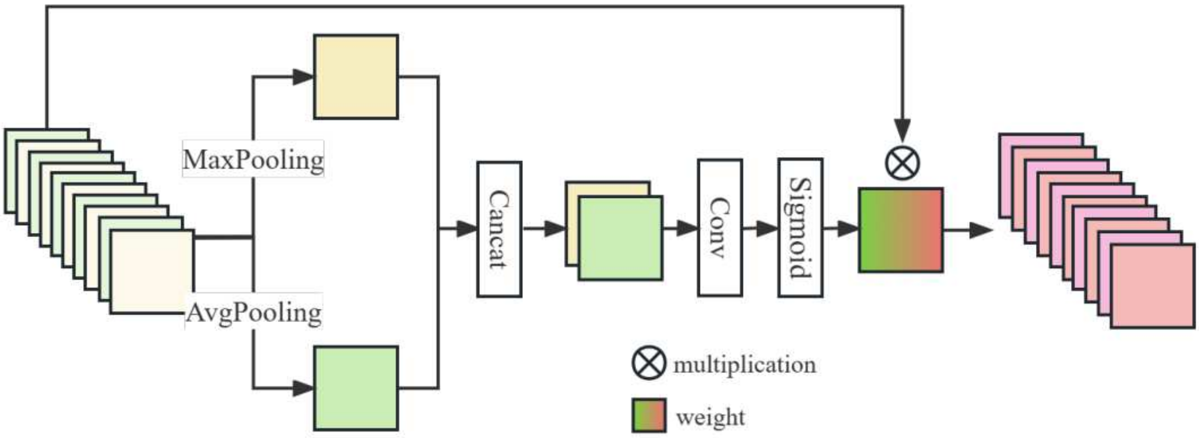

Figure 2. The assemblage of the spatial care in SFE.

3.4.3. Spatial Site Encoding

The input fruit pictures are transferred data, which may compromise the 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 the nearby pixels tends to diminish with increasing distance from the centre, although they can still contribute spatial information for the centre pixel's categorisation. Such a centrally crucial position encoding is to be learned by SPE. The definition of a patch's pixel locations is as follows:

$$pos(x_i, y_i) = |x_i - x_c| + |y_i - y_c| + 1 \quad (24)$$

where (x_c, y_c) denotes the organize of central 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 how far out from the centre they are. The data is incorporated with the learnable position encoding so that it may flexibly describe the spatial structure in fruit photos.:

$$Y = X + spe(P) \quad (25)$$

where X is the fruit image data, besides P characterizes the site matrix built rendering to Equation (25). $spe(\cdot)$ 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 uses window-based multi-head), which is 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 window of spectral shift known as spectral window multi-head swas created for MSA in consideration of the rich data in the spectral dimension. Information may be shared between neighbouring windows by window shifting and MSA inside windows, which can enhance local feature capture. You may use the following formula to express MSA.:

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

$$\psi = Concat(Z_1, Z_2, \dots, Z_h)W \quad (27)$$

The input matrices known as queries, keys, besides values are translated into the matrices Q, K, and V. K's dimension is denoted by d_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, the each window is split equally. Following division, the sizes of each window are [C/4, C/4, C/4, C/4]. Next, MSA is carried out for every 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 carried out once again in every window. Thus, the S-W-MSA procedure through m windows is

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

where $\oplus$ resources concat, $y^{(\mathrm{i})}$ is the statistics of the i-th window.

With the exception of the window design, the remaining elements of the S-SwinT module—MLP, layer normalisation connections—remain unchanged when compared to SwinT. The formulas shown below are as shadows:

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

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

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

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

4. Results and Discussion

The deep learning framework, PyTorch, was used together with an NVIDIA Tesla V100 32 G of video RAM. Table 2 lists the parameters used in the simulation.

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

Table 3 and 4 explains the experimental analysis of 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

In above Table 3 represent that the Validation Analysis of projected feature extraction on 70%-30%. In the investigation of LeNet module, attained the precision rate as 0.8298 and recall of 0.8508 and f1-score as 0.8401 and accuracy as 84.06 respectively. Then the ResNet module, attained the precision rate as 0.8679 and recall of 0.8648 and f1-score as 0.8663 and accuracy as 86.12 respectively. Then the VGGNet module, attained the precision rate as 0.9011 and recall of 0.8883 and f1-score as 0.8947 and accuracy as 89.78 respectively. Then the AlexNet module, attained the precision rate as0.9279 and recall of 0.9109 and f1-score as 0.9193 and accuracy as 92.71 respectively. Then the MAlexNet-GJO module, attained the precision rate as 0.9467 and recall of 0.9337 and f1-score as 0.9402 and accuracy as 93.82 respectively.

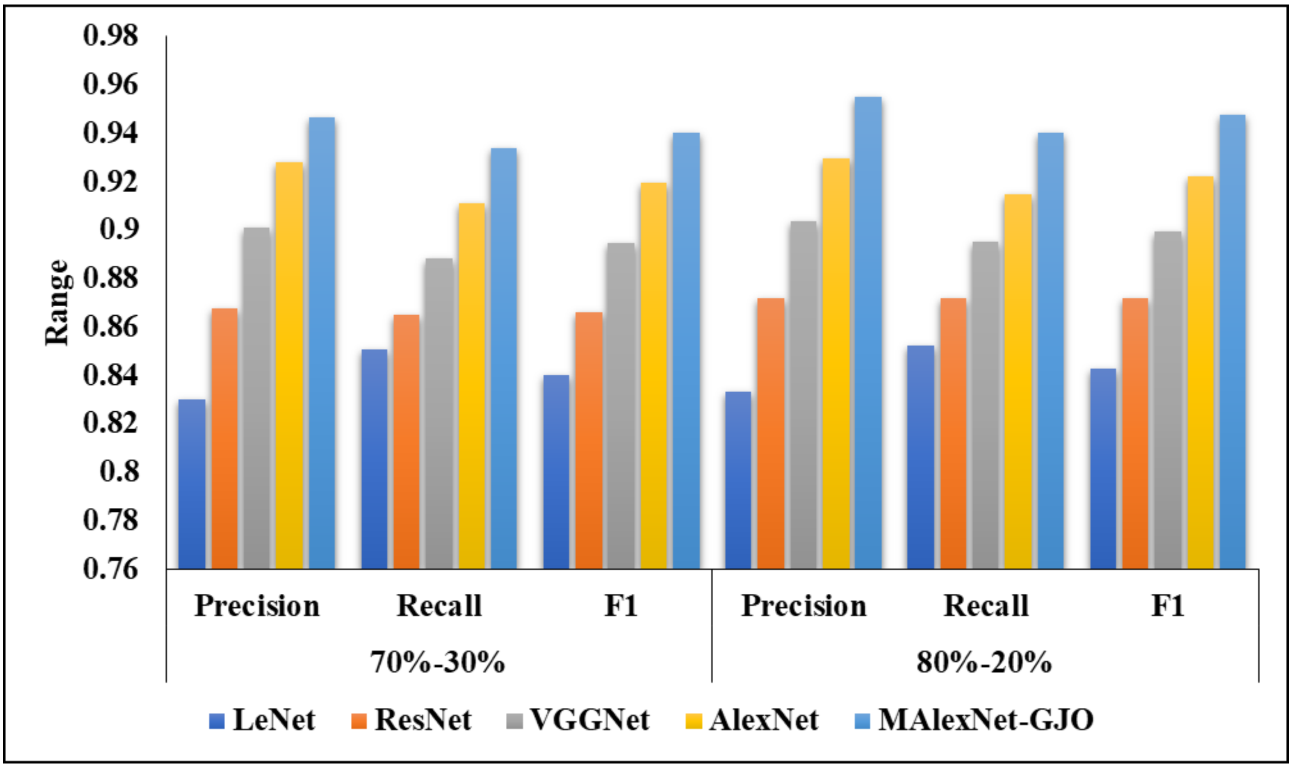

Figure 3: Visual Representation of proposed Feature extraction model

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 above Table 4 characterise that the Validation Investigation of projected feature extraction on 80%-20%. In the investigation of LeNet module, attained the precision rate as 0.8333 and recall of 0.8525 and f1-score as 0.8427 and accuracy as 84.25 respectively. Then the ResNet module, attained the precision rate as 0.8718 and recall of 0.8718 and f1-score as 0.8718 and accuracy as 86.87 respectively. Then the VGGNet module, attained the precision rate as 0.9038 and recall of 0.8952 and f1-score as 0.8995 and accuracy as 89.56 respectively. Then the AlexNet module, attained the precision rate as 0.9295 and recall of 0.9148 and f1-score as 0.9220 and accuracy as 92.06 respectively. Then the MAlexNet-GJO module, attained the precision rate as 0.9551 and recall of 0.9400 and f1-score as 0.9475 and accuracy as 94.44 respectively.

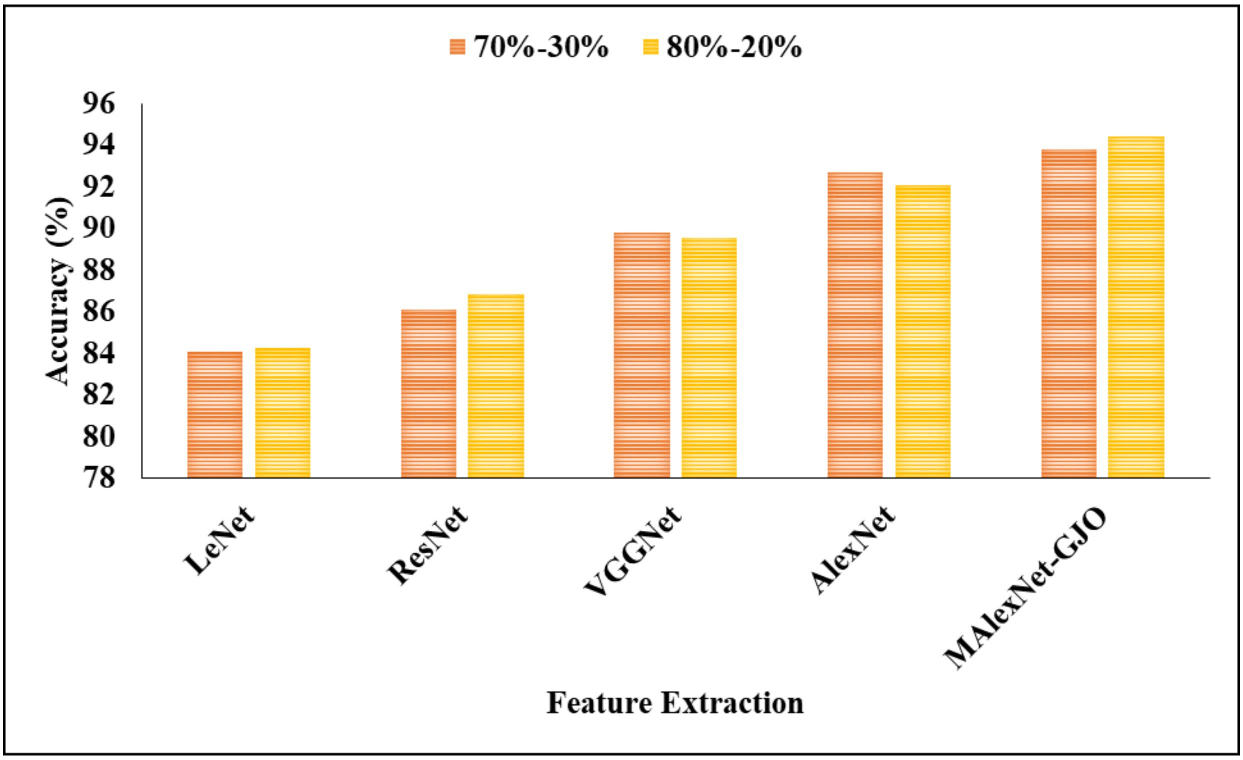

Figure 4: Graphical Representation of proposed model in terms of Accuracy

4.2. Verification of Proposed Classifier model

Table 5 and 6 mentions the validation analysis of proposed classifier for different training ratio and testing ratio.

Table 5: Validation of 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 above Table 5 characterise that the Authentication of proposed model for 70%-30%. In the analysis of multi-ScaleAlexNet module, accomplished the precision rate as 0.9163 and recall of 0.9159 and f1-score as 0.9134 and accuracy as 91.96 respectively. Then the TFFbU module, attained the precision rate as 0.8572 and f1-score as 0.8565 and accuracy as 85.49 respectively. Then the Olive-EfficientDet module, attained the precision rate as 0.9224 and recall of 0.9281 and f1-score as 0.9264 and accuracy as 86.62 respectively. Then the Self-Attention module, attained the precision rate as 0.9369 and recall of 0.9360 and f1-score as 0.9364 and accuracy as 93.59 respectively. Then the FSSATM module, attained the precision rate as 0.9551 and recall of 0.9400 and f1-score as 0.9475 and accuracy as 94.44 respectively.

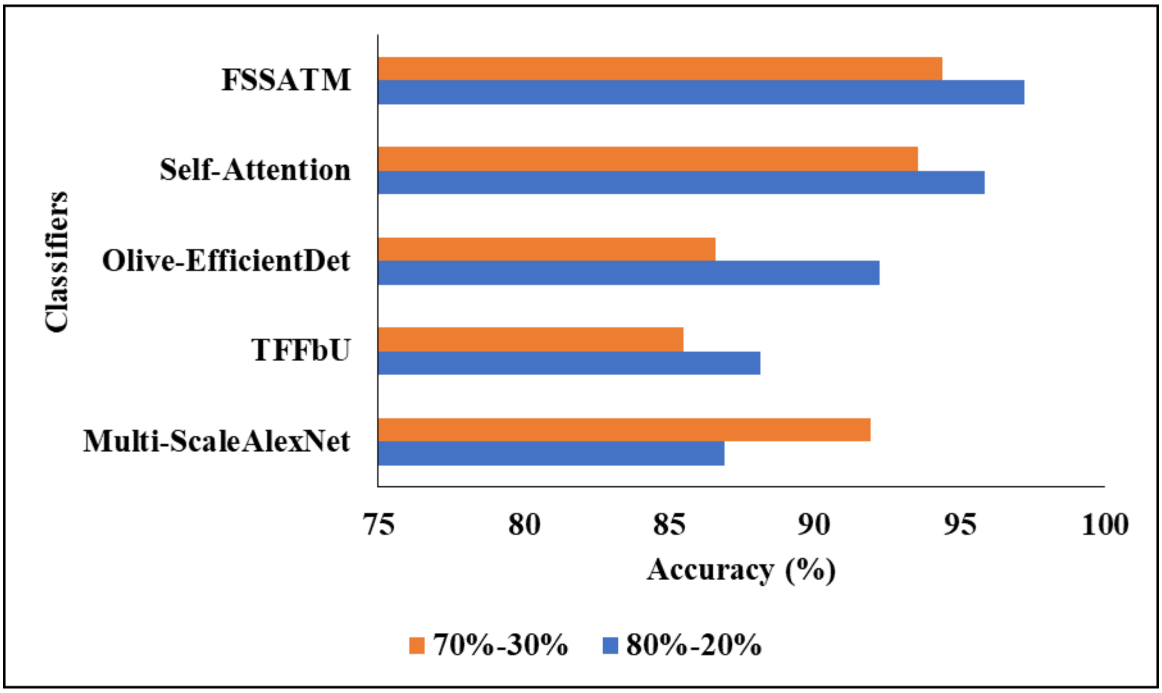

Figure 5: Accuracy Analysis of Proposed Classifier

Table 6: Experimentation of 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

In overhead Table 6 represent that the Experimentation of Projected model for 80%-20%. In the investigation of multi-ScaleAlexNet module, attained the precision rate as 0.8639 and recall of 0.8747 and f1-score as 0.8693 and accuracy as 86.91 respectively. Then the TFFbU module, attained the precision rate as 0.8925 and recall of 0.8714 and f1-score as 0.8818 and accuracy as 88.16 respectively. Then the Olive-EfficientDet module, attained the precision rate as 0.9216 and recall of 0.9309 and f1-score as 0.9262 and accuracy as 92.27 respectively. Then the Self-Attention module, attained the precision rate as 0.9595 and f1-score as 0.9513 and recall of 0.9554 and accuracy as 95.86 respectively. Then the FSSATM module, attained the precision rate as 0.9756 and recall of 0.9715 and f1-score as 0.9735 and accuracy as 97.24 respectively.

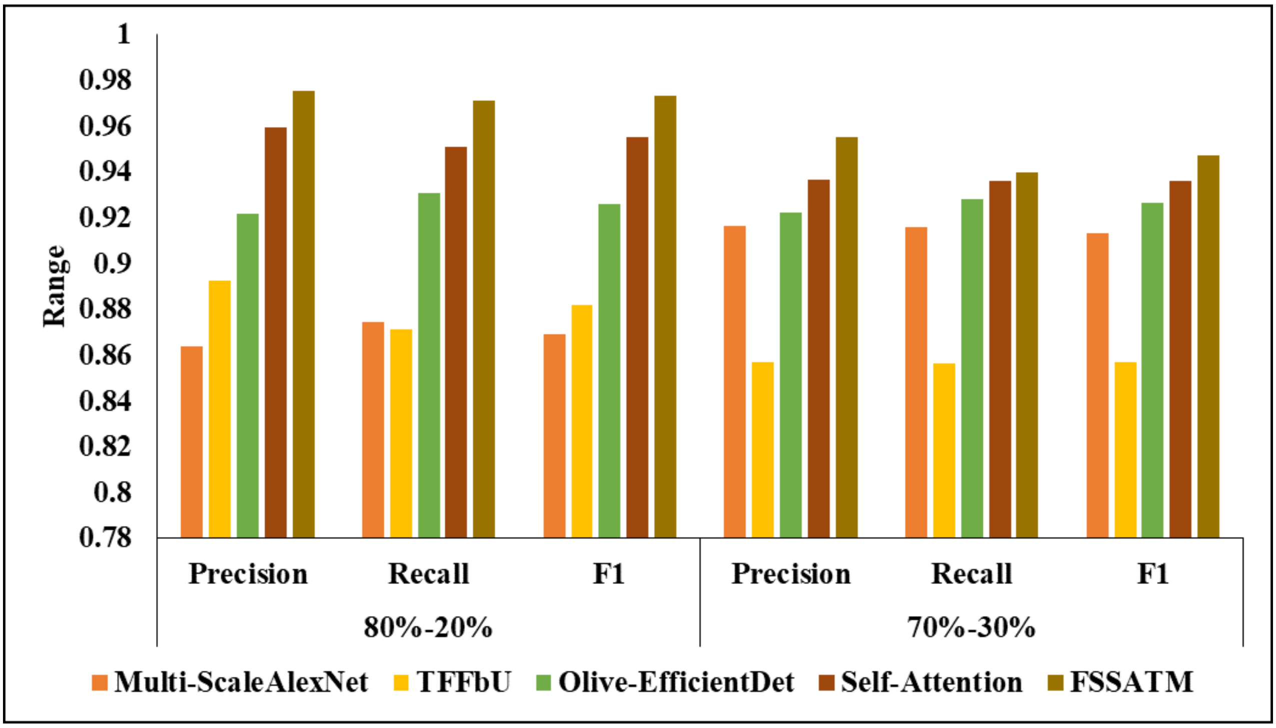

Figure 6: Visual Analysis of proposed model for different ratio

5. Conclusion

Numerous academics attempted to use learning approaches to identify fruits in the Fruit-360 dataset, which has 90,483 sample photos and 131 fruit classifications. But none of the earlier efforts concentrated on managing the entire set of 131 fruit classes and their associated fruit pictures. Consequently, this study paper 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 used to evaluate the updated AlexNet algorithm's performance in image-based classification, with GJOA being used to fine-tune 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 incapacity to acquire local contextual data during categorisation. The learning curve and the confusion matrix were used to assess the tested algorithm's performance. Here, it can be said that, for the given job, the suggested algorithms achieved healthier than any other procedures. Consequently, the findings offer strong proof that the suggested approach is more efficient and accurate in comparison to CNN-based techniques in handling multiple class picture classification issues. Furthermore, the system demonstrated its ability to process the whole Fruit-360 dataset with reduced processing resources. The suggested feature extraction classifiers are appropriate for real-time besides economical scheme implementations, it may be inferred based on the findings. One major drawback of the suggested technique is that, depending on the dataset, it can require a different structure (e.g., a different number of levels and total inputs) in order to achieve greater accuracy. Consequently, a general framework for image-based categorisation issues should be implemented in future efforts.

Funding

No Funding

Conflict of interest

The author declares no conflict of interest

Data availability statement

The data that support the findings of this study are available on request from the corresponding author.

References

[1] Shahi, T. B., Sitaula, C., Neupane, A., & Guo, W. (2022). Fruit classification using attention-based MobileNetV2 for industrial applications. Plos one, 17(2), e0264586.

[2] Ukwuoma, C. C., Zhiguang, Q., Bin Heyat, M. B., Ali, L., Almaspoor, Z., & Monday, H. N. (2022). Recent advancements in fruit detection and classification using deep learning techniques. Mathematical Problems in Engineering, 2022, 1-29.

[3] Mimma, N. E., Ahmed, S., Rahman, T., & Khan, R. (2022). Fruits Classification and Detection Application Using Deep Learning. Scientific Programming, 2022.

[4] Gill, H. S., Murugesan, G., Khehra, B. S., Sajja, G. S., Gupta, G., & Bhatt, A. (2022). Fruit recognition from images using deep learning applications. Multimedia Tools and Applications, 81(23), 33269-33290.

[5] Ismail, N., & Malik, O. A. (2022). Real-time visual inspection system for grading fruits using computer vision and deep learning techniques. Information Processing in Agriculture, 9(1), 24-37.

[6] Albarrak, K., Gulzar, Y., Hamid, Y., Mehmood, A., & Soomro, A. B. (2022). A deep learning-based model for date fruit classification. Sustainability, 14(10), 6339.

[7] Aherwadi, N. A. G. N. A. T. H., & Mittal, U. S. H. A. (2022). Fruit quality identification using image processing, machine learning, and deep learning: A review. Adv. Appl. Math. Sci, 21, 2645-2660.

[8] Ibrahim, N. M., Gabr, D. G. I., Rahman, A. U., Dash, S., & Nayyar, A. (2022). A deep learning approach to intelligent fruit identification and family classification. Multimedia Tools and Applications, 81(19), 27783-27798.

[9] Majid, A., Khan, M. A., Alhaisoni, M., Tariq, U., Hussain, N., Nam, Y., & Kadry, S. (2022). An Integrated Deep Learning Framework for Fruits Diseases Classification. Computers, Materials & Continua, 71(1).

[10] Hussain, D., Hussain, I., Ismail, M., Alabrah, A., Ullah, S. S., & Alaghbari, H. M. (2022). A simple and efficient deep learning-based framework for automatic fruit recognition. Computational Intelligence and Neuroscience, 2022.

[11] Aherwadi, N., Mittal, U., Singla, J., Jhanjhi, N. Z., Yassine, A., & Hossain, M. S. (2022). Prediction of fruit maturity, quality, and its life using deep learning algorithms. Electronics, 11(24), 4100.

[12] Fahad, L. G., Tahir, S. F., Rasheed, U., Saqib, H., Hassan, M., & Alquhayz, H. (2022). Fruits and Vegetables Freshness Categorization Using Deep Learning. Computers, Materials & Continua, 71(3).

[13] Gill, H. S., & Khehra, B. S. (2022). An integrated approach using CNN-RNN-LSTM for classification of fruit images. Materials Today: Proceedings, 51, 591-595.

[14] Mukhiddinov, M., Muminov, A., & Cho, J. (2022). Improved classification approach for fruits and vegetables freshness based on deep learning. Sensors, 22(21), 8192.

[15] Verma, R., & Verma, A. K. (2022, February). Fruit classification using deep convolutional neural network and transfer learning. In International Conference on Emerging Technologies in Computer Engineering (pp. 290-301). Cham: Springer International Publishing.

[16] Karthikeyan, M., Subashini, T. S., Srinivasan, R., Santhanakrishnan, C., & Ahilan, A. (2024). YOLOAPPLE: Augment Yolov3 deep learning algorithm for apple fruit quality detection. Signal, Image and Video Processing, 18(1), 119-128.

[17] Kwon, S. H., Ku, K. B., Le, A. T., Han, G. D., Park, Y., Kim, J., ... & Mansoor, S. (2024). Enhancing citrus fruit yield investigations through flight height optimization with UAV imaging. Scientific Reports, 14(1), 322.

[18] Raihen, M. N., & Akter, S. (2024). Prediction modeling using deep learning for the classification of grape-type dried fruits. International Journal of Mathematics and Computer in Engineering.

[19] Shu, H., He, C., Mumtaz, M. A., Hao, Y., Zhou, Y., Jin, W., ... & Wang, Z. (2023). Fine mapping and identification of candidate genes for fruit color in pepper (Capsicum chinense). Scientia Horticulturae, 310, 111724.

[20] Patel, H. B., & Patil, N. J. (2024). An Intelligent Grading System for Automated Identification and Classification of Banana Fruit Diseases Using Deep Neural Network. International Journal of Computing and Digital Systems, 15(1), 761-773.

[21] Lin, Y., Huang, Z., Liang, Y., Liu, Y., & Jiang, W. (2024). AG-YOLO: A Rapid Citrus Fruit Detection Algorithm with Global Context Fusion. Agriculture, 14(1), 114.

[22] Suhasini, A., & Balaram, V. V. S. S. S. (2024). Detection and Classification of Disease from Mango fruit using Convolutional Recurrent Neural Network with Metaheruistic Optimizer. International Journal of Intelligent Systems and Applications in Engineering, 12(9s), 321-334.

[23] Zhu, X., Chen, F., Zhang, X., Zheng, Y., Peng, X., & Chen, C. (2024). Detection the maturity of multi-cultivar olive fruit in orchard environments based on Olive-EfficientDet. Scientia Horticulturae, 324, 112607.

[24] Vinisha, A., & Boda, R. (2024). A Novel Framework for Brain Tumor Segmentation using Neuro Trypetidae Fruit Fly-Based UNet. International Journal of Intelligent Systems and Applications in Engineering, 12(1s), 783-796.

[25] Li, Z., Deng, X., Lan, Y., Liu, C., & Qing, J. (2024). Fruit tree canopy segmentation from UAV orthophoto maps based on a lightweight improved U-Net. Computers and Electronics in Agriculture, 217, 108538.

[26] Mure¸san, H.; Oltean, M. Fruit recognition from images using deep learning. Acta Univ. Sapientiae Inform. 2018, 10, 26–42.

[27] Thirumalraj, A., Asha, V., & Kavin, B. P. (2023). An Improved Hunter-Prey Optimizer-Based DenseNet Model for Classification of Hyper-Spectral Images. In A. Khang (Ed.), AI and IoT-Based Technologies for Precision Medicine (pp. 76-96). IGI Global. https://doi.org/10.4018/979-8-3693-0876-9.ch005.