

## Research Article

# Fruits Classification and Detection Application Using Deep Learning

**Nur-E-Aznin Mimma, Sumon Ahmed, Tahsin Rahman, and Riasat Khan** 

*Department of Electrical and Computer Engineering, North South University, Dhaka, Bangladesh*

Correspondence should be addressed to Riasat Khan; [riasat.khan@northsouth.edu](mailto:riasat.khan@northsouth.edu)

Received 10 June 2022; Revised 4 October 2022; Accepted 4 November 2022; Published 17 November 2022

Academic Editor: Zhiri Tang

Copyright © 2022 Nur-E-Aznin Mimma et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Computer vision and image processing techniques are considered efficient tools for classifying various types of fruits and vegetables. In this paper, automated fruit classification and detection systems have been developed using deep learning algorithms. In this work, we used two datasets of colored fruit images. The first FIDS-30 dataset of 971 images with 30 distinct classes of fruits is publicly available. A major contribution of this work is to present a private dataset containing 761 images with eight categories of fruits, which have been collected and annotated by ourselves. In our work, the YOLOv3 deep learning object detection algorithm have been used for individual fruit detection across multiple classes, and ResNet50 and VGG16 techniques have been utilized for the final classification for the recognition of a single category of fruit in images. Next, we implemented the automatic fruit classification models with flask for the web framework. We got 86% and 85% accuracies from the public dataset with ResNet50 and VGG16, respectively. We achieved 99% accuracy with ResNet50 and 98% accuracy with the VGG16 model on the custom dataset. The domain adaptation approach is used in this work so that the proposed deep learning-based prediction model can cope with real-world problems of diverse domains. Finally, an Android smartphone application has been developed to classify and detect fruits with the camera in real-time. All the images uploaded from the Android device are automatically sent and consequently analyzed on the web, and finally, the processed data and results are returned to the smartphone. The custom dataset and implementation codes will be available after the manuscript has been accepted. The custom dataset can be found at: [https://github.com/SumonAhmed334/dataset\\_fruit](https://github.com/SumonAhmed334/dataset_fruit).

## 1. Introduction

Fruits contain vitamins and dietary fibers, a vital source of the human diet [1]. Different types of more than 2,000 fruits are found worldwide, but most people are familiar with only 10% of them. According to fruit production statistics, million metric tons of fruits were produced worldwide in 2021, from which the largest producing countries are China, India, and Brazil [2]. The advanced agricultural fruit recognition system with a simple camera or sensor will play an excellent role for farmers and general people [3]. In this modern era of technological advancements, fruit classification and recognition systems can be used for kids' educational purposes, which interest them greatly [4]. The latest advanced computer vision technology with the utilization of

deep neural networks can be used for object discovery and semantic picture division [5].

Thanks to computer vision and advanced image processing techniques, it has been successfully employed in various works for automatic fruit recognition and classification. Some of the notable works on automatic fruit recognition and classification have been briefly described in the following paragraphs.

Most of these automatic fruit classification works used a wide range of deep learning-based neural network frameworks. For instance, in a recent work [6], the authors designed an automatic model to recognize vegetables by image processing and computer vision approaches. The authors compare 24 different types of vegetables in the dataset by using 3,924 pictures. First, the authors trained the

data, and then, preprocessed the images by resizing and normalizing. Next, they implemented the convolutional neural network (CNN) [7] learning model, which trains the data with a batch size of 16 and 100 numbers of epochs. Finally, the convolution neural network approach developed on Keras recognized the vegetables employing color and texture characteristics and achieved approximately 95.50% accuracy. In [8], the authors utilized MLPNeural, fluctuating logic, significant analysis of components, and neural networks as additional techniques for automated skin deformity identification of citrus fruits. Next, the accuracy of the proposed system has been evaluated in terms of its categorization of the decision-making approach. This work achieved defect detection accuracies of 100 percent, 91 percent, 89 percent, and 83 percent, with the decision tree, principal parts analysis, fuzzy logic, and MLPNeural approaches, respectively. Joseph et al. proposed an automatic fruit classification system using a convolutional neural network framework [9]. This study uses a diverse dataset of Indian fruits of more than 130 classes. Finally, the authors claimed accuracy of approximately 95% for the proposed TensorFlow-based CNN model. Ponce and his colleagues [10] focused on the automatic classification of olive. For the experiment, 2,800 olive fruits of seven different varieties were photographed. This article used six different convolutional neural network architectures, but finally, Inception-ResNetV2 produced the best results. This model achieves a 95.91 percent accuracy rate for diverse categories of olive classification. In [11], the authors introduced computerized classification to classify fruits. They worked with a particular dataset containing four different types of fruits, all subjected to image processing algorithms. BPNN, SVM, and CNN classifiers are utilized in this paper, with CNN providing the most remarkable accuracy of 90% compared to other classifiers. A novel fruit categorization approach based on fruit pictures and deep learning is presented in [12]. The authors created a six-layer CNN to recognize nine different types of fruits. Three types of datasets have been combined in this work, viz. one is a bespoke dataset and the other two are public datasets. According to the findings of the experiments, the proposed CNN method with six layers performed well in classification, with an accuracy of approximately 91.50%, which performed better than the state-of-the-art classification approaches.

In some of the articles, advanced image processing and machine learning techniques have been used. Seng and his co-author [13] built an automatic fruits recognition system consisting of various images processing frameworks, i.e., input selection, color, shape, and size computing, and classification or recognition. The authors used 50 collected images for this recognition system in this paper. Next, 36 images were used to develop and train using the KNN model. Finally, the authors utilized 14 fruit images to evaluate the proposed framework. They have used a custom-made dataset that they have created on their own. Lastly, this model shows nearly 90% accuracies in computing the geometrical properties of various fruit categories. Shiv and Anand [3] built a fruit classification system using image processing techniques.

The Gaussian Naive Bayes algorithm has been developed utilizing the Python platform environment. The findings of the various types of apples, i.e., Granny Smith, Braeburn, Golden Delicious and Cripps Pink, and other fruits, for example, mandarin, lemon, and orange, indicated that the projected average accuracy values for training and test datasets were 100% and 73%, respectively.

Few automatic plant and fruit classification works have been extended to different applications, e.g., disease detection, smart farming, and so on. Abayomi-Alli et al. [14] used MobileNetV2 to detect cassava leaf disease and an open-source dataset. This deep learning model achieved 97.7% accuracy and 96.7% *F1* score. Almadhor et al. [15] initiated plant disease recognition using machine learning techniques and complex sensors. The authors implemented image segmentation and feature extraction by applying image processing approaches. Oyewola et al. [16] identified cassava mosaic viruses in plants to enhance the production of crops. A custom convolutional recurrent neural network is applied to an open-source dataset of more than 5,600 images with five distinct classes.

After reviewing the above papers, we can confirm that computer vision and sophisticated deep learning approaches have been successfully applied for automatic fruit classification and detection. However, only a few studies have used deep learning models in a website framework or smartphone app for real-time classification. Most of the models are not tested to determine whether they can be implemented in diverse domains.

This paper implements an automatic classification and detection of fruits using neural network approaches. This system can be used for children for educational purposes, industries and supermarkets and for people to know different fruits for learning purposes. The primary contributions of this work are described as follows:

- (i) Automatic fruit detection and classification system have been developed by using two datasets, i.e., open-source FIDS-30 of 30 classes and collected custom dataset of eight categories of fruits.
- (ii) For classification, we used VGG16 and ResNet50 neural network models. YOLOv3 and YOLOv7, deep learning frameworks, have been employed to detect multiple fruits in the image.
- (iii) The domain adaptation technique is applied so that the proposed deep learning-based fruit classification model can cope with real-world problems of diverse domains. Using this method, different sets of images of various fruits were used to train and test the proposed model.
- (iv) The web framework of the proposed automatic fruit classification and detection system is created with the help of a flask.
- (v) A Python-based API has been utilized to develop an Android smartphone application, which uses the phone's camera for instantaneous detection and recognition of fruits.

The primary contributions of this work are to create a custom dataset of eight species of fruits and apply YOLOv7, deep learning model, and domain adaptation technique.

In this paragraph, the organization of this manuscript has been discussed. Section 2 discusses the materials and methods of the proposed automatic fruit classification system with appropriate tables, figures, and flowcharts. The actual results of the research have been shown in Section 3. Finally, Section 4 provides some recommendations for the paper's future improvement.

## 2. Materials and Models

In this section, the dataset details, brief discussions about the employed deep learning models and the performance of the system will be discussed.

*2.1. Datasets.* In this work, we used two types of datasets of various categories of fruits. One is the public dataset named FIDS-30 [17], and the pictures of this dataset are collected from Internet sources. This open-source consists of 30 classes of fruits, as illustrated in Table 1. There are approximately 30 to 40 images in each category with a considerable variation of added noises. A significant contribution of this work is to present a dataset of various fruits images that we had obtained and captured by smartphone cameras. In this second dataset, we work with eight classes of fruits, which are primarily available in Bangladesh. As illustrated in Table 2, this custom dataset contains 761 images of apples, coconuts, grapes, limes, oranges, tomatoes, bananas, and guavas. Figure 1 demonstrates some sample images of the used open-source FIDS-30 dataset. Next, in the data splitting part, we assigned 80% of the images for the training, 10% for the testing part, and 10% for the validation part. Next, traditional data augmentation techniques, e.g., projection, rotation, scaling, changing brightness and contrast, etc., are applied to both datasets to enhance the efficiency of the deep learning techniques.

In Figure 2, demonstrates some sample images of the acquired custom dataset we captured with the mobile camera.

*2.2. Model Descriptions.* This section describes the deep learning approaches that have been employed in this work for automatic fruit detection of multiple classes and classification of single category. In this paper, YOLOv3 deep convolutional neural network framework has been used for fruit detection. The convolutional neural networks are a variation of profound networks, which consequently learn basic edge shapes from crude information and recognize the complicated conditions inside each picture through include extraction [13]. The convolutional neural networks incorporate different layers compared to the human visual framework. Among them, the convolutional layers, by and large, have filters with the parts of  $11 \times 11$ ,  $9 \times 9$ ,  $7 \times 7$ ,  $5 \times 5$ , or  $3 \times 3$ . The channel fits load through preparing and learning, while the loads can separate elements, only like camera channels.

*2.2.1. YOLOv3 and YOLOv7.* The YOLO technique is faster than the faster RCNN approach because it can directly predict the bounding box and compare class likelihood through a single neural architecture. This strategy can help to mitigate the effects of lighting variance, overlap, and occlusion [17]. YOLOv3 and YOLOv7 utilize Darknet53 and PyTorch, respectively, as the component extraction organization and utilization of the FPN construction to accomplish the combination of various scale highlights and different scale expectations [20]. The utilization of various scale expectations causes YOLOv3 to identify tiny targets better. The most advanced single-stage object detection framework, YOLOv7 [21] offers faster inference with efficient model scaling. This model preserves multiple batches of weights for parts of the entire architecture. Hence, YOLOv3 and YOLOv7 have been used as the framework to detect fruits in this paper.

In the YOLO calculation, the first pictures are first resized to the information size, utilizing a scale pyramid structure like the feature pyramid network. For instance, YOLOv3 will anticipate the three dimensions of the element guide of  $13 \times 13$ ,  $26 \times 26$ , and  $52 \times 52$  and go through multiple times to move highlights between 2 contiguous scales. In each expectation scale, each framework cell will anticipate three jumping boxes with the assistance of 3 anchor boxes. The bounding box shape could increase detection performance when focusing on a particular activity [17]. The architecture of the proposed YOLOv3 model is depicted in Figure 3. In Figure 4, an example of the detection operation for individual fruit detection of multiple classes of fruits of the proposed YOLOv3 framework is illustrated.

Next, ResNet50, and VGG16 techniques are utilized for the final classification.

*2.2.2. ResNet50.* ResNet50 is a convolutional neural organization that is 50 layers profound. This convolutional neural network comprises 5 phases, each with a convolution and identity block. Every convolution and personality block consists of three layers. More than 23 million trainable parameters are available in the ResNet50 framework. The [9] ResNet50's top layers are not frozen, which they learn through backpropagation. However, the rest of the ResNet50's layers remain frozen [22]. This operation is performed by building a ResNet50 model with pretrained parameters (weights) from the ImageNet dataset [23]. The preprepared organization can group photos into 1,000 item classes.

The architecture of the used ResNet50 model is demonstrated in Figure 5, which is a 50-layer deep residual learning framework with five steps. This framework is devised for large-scale classification of fruits in the natural environment. The cross-entropy loss function for the ResNet50 model is expressed as follows:

$$\text{Loss} = -\left(\frac{1}{n}\right) \sum_{i=0}^n \log P(N), \quad (1)$$

where  $n$  is the output size of the classification network.

TABLE 1: Class names and numbers of dataset-I.

Class names	Number of samples
Acerolas	24
Apples	38
Apricots	30
Avocados	26
Bananas	42
Blackberries	37
Blueberries	32
Cantaloupes	31
Cherries	33
Coconuts	26
Figs	26
Grapefruits	31
Grapes	38
Guava	33
Kiwifruit	36
Mangos	34
Olives	23
Oranges	35
Passion fruit	22
Peaches	27
Pears	32
Pineapples	34
Plums	31
Pomegranates	30
Raspberries	39
Strawberries	46
Tomatoes	46
Watermelons	39
Lemons	29
Limes	29

TABLE 2: Class names and numbers of dataset-II.

Class names	Number of samples
Apple	200
Orange	100
Guava	60
Lime	50
Tomato	95
Coconut	50
Banana	150
Grapes	30

2.2.3. *VGG16*. The main contribution of VGG16 is to demonstrate that, under certain circumstances, increasing the network’s depth can improve performance [24]. This convolutional neural network model is vast, and it has around 138 million (approx.) boundaries. The model accomplishes 92.70% top-5 test precision in ImageNet, which is a dataset of more than 14 million pictures consists of approximately 1,000 classes. Accordingly, the framework has learned rich element portrayals for many pictures. The organization has a picture input size of  $224 \times 224$ . Next, we add another top layer comprising a seriously huge, completely associated layer with satisfactory regularization as a dropout. Consequently, the pictures were downsized to  $224 \times 224$  pixels in size to meet the input requirements of the pretrained VGG16 [25]. Since there are four yield classes, the

yield layer has four hubs. We then, at that point, freeze the VGG16 model, i.e., we do not prepare any of the loads; we use it as it is. VGG16 has 16 convolutional layers, as depicted in Figure 6, and is particularly appealing due to its very homogeneous and uniform architecture.

Next, the proposed automatic fruit detection and classification systems have been deployed into a website framework and Android smartphone application, which has been discussed in the subsequent sections.

2.3. *Web Framework*. In this work, the flask framework has been used to create a web application of the proposed fruit classification system. Python-based flask is an easy-to-use and flexible web framework [26]. This module is named a microframework because it does not require specialized instruments or libraries. It has no dataset deliberation layer, structure approval layer, or other parts where prior external libraries provide typical functions. However, this flexible module supports integrations that can add application features as if they were made inside of flask. Figure 7 illustrates how the proposed classification models have been deployed to a web and smartphone applications using the flask framework.

Finally, Figure 8 shows the complete fruit detection and recognition system deployed in a website framework. Here are two categories: the first one is called classification, and the second one is called detection. When a user selects the classification category, consequently, he needs to upload the image of a single fruit. After that, the name will be shown as the output on the final page of the mobile application. For detection, a user has to drop a photo of multiple fruits, and then, the annotations and name will be returned as output. It is worth mentioning that the user can upload images from the mobile phone’s memory or can capture images instantaneously by using its camera.

2.4. *Android Application*. We have created an Android smartphone application with which we can easily detect and classify fruits using a mobile camera in real time. Consequently, after building the fruit detection and classification model, an API was created. We could directly load this model into the app, but we will serve it via API to decrease the final application’s requirements. Second, this system can be utilized in other applications such as websites, telegram bots, slack bots, and so on. Developing an API in Python is simple, and we have used the flask REST framework in this research. It is a flexible microframework that can handle any web service with just a few lines of code. We have constructed an API endpoint for this application, which accepts all model parameters as input and returns the final forecast. The result from the API, the fruit images, and the user data will be stored in the database. This work used a Firebase nonrelational database for the app management tasks. Figure 9 illustrates the working sequences in developing the proposed Android application of the fruit classification system.

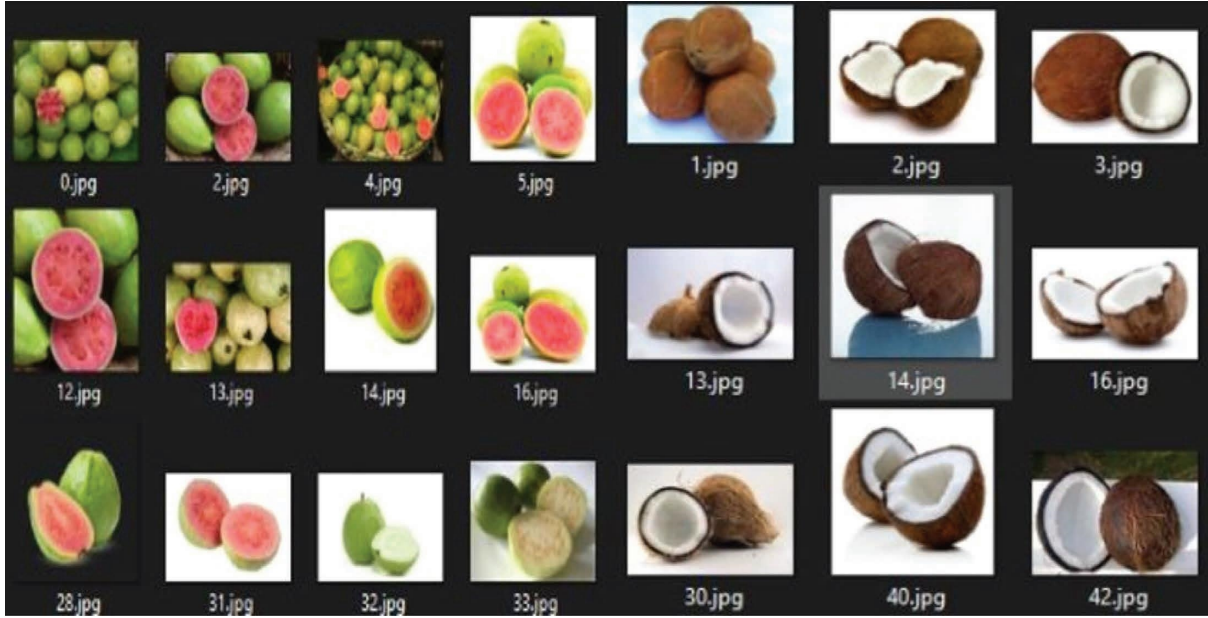


FIGURE 1: Sample images of various fruits of FIDS-30 dataset-I.



FIGURE 2: Sample images of different fruits of the custom dataset-II.

### 3. Results and Discussion

In this section, the results of the proposed automatic fruit classification system are discussed. In this work, an online framework, make sense, has been used to label the images and export them into corresponding XML files. The labeled images are preprocessed in the Roboflow tool by performing conventional preprocessing techniques, e.g., auto-orientation, object isolation (crop and extract bounding boxes into individual images), and resizing. The data augmentation process has been done to create new training examples for the model to learn from by generating augmented versions of each image in the training set [14]. In this work, bounding box level augmentation instead of picture level has been executed as the bounding box accomplishes better accuracy. Random horizontal and vertical flips,  $\pm 90^\circ$  rotation, normalization, and brightness adjustment are implemented. The

data augmentation techniques make the model insensitive to camera orientation, generalize better to various contexts, and reduce overfitting and underfitting problems. Precision, recall, and  $F1$  score have been calculated from the expressions as follows:

$$\text{Precision} = \frac{TP}{(TP + FP)}, \quad (2)$$

$$\text{Recall} = \frac{TP}{(TP + FN)}, \quad (3)$$

$$F1 = \frac{(2 \times \text{Precision} \times \text{Recall})}{(\text{Precision} + \text{Recall})}. \quad (4)$$

In (2) and (3), TP, FN, and FP are shortenings for True Positive (promising discoveries), False Negative (missed

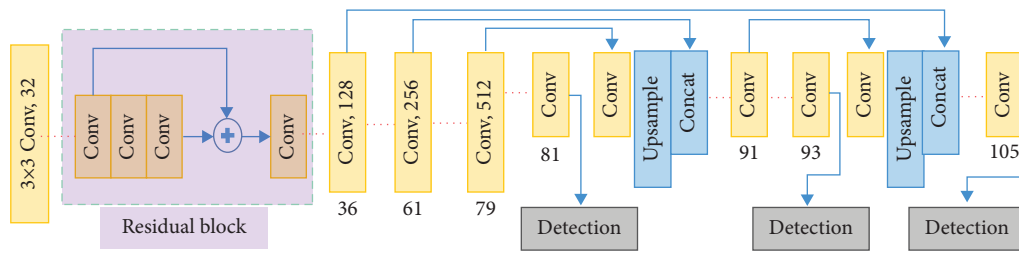


FIGURE 3: Architecture of the YOLOv3 model.

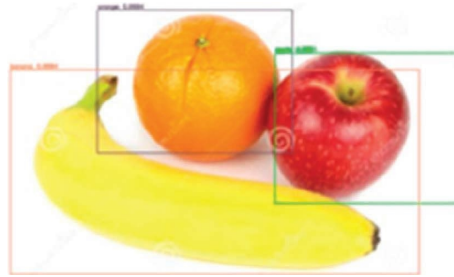


FIGURE 4: Demonstration of fruit detection of multiple classes of fruits of the implemented YOLOv3 model.

identifications), and False Positive (inaccurate findings), respectively. *F1* score was led as a compromise among recall and precision to show the exhaustive presentation of the prepared models, which has been expressed in (4).

**3.1. Results for the FIDS-30 Open-Source Dataset.** Tables 3 and 4 demonstrate the *F1* score for the public FIDS-30 dataset in the ResNet50 and VGG16 models, respectively. According to these tables, the mean *F1* score for the VGG16 and ResNet50 frameworks for the FIDS-30 dataset are 0.878 and 0.843, respectively.

Next, Figures 10 and 11 illustrate the performance of the ResNet50 and VGG16 classification algorithms, respectively, which have been summarized using confusion matrices.

Figures 12 and 13 illustrate the training and validation behaviors versus epochs of the VGG16 model with the FIDS-30 dataset. We have used VGG16 as our final classification model because this technique gives us more accuracy than ResNet50.

In this work, the YOLOv7 deep learning model has been implemented in the Roboflow environment and PyTorch framework. Hyperparameters to train the YOLOv7 model have been illustrated in Table 5. The dataset has been trained for ten epochs and 16 batch sizes and the training stops if the validation generalization loss does not improve. The YOLOv7 model achieved 96.1% accuracy, 0.93 and 0.89 precision and recall, respectively, for the FID-30 dataset of 30 fruit categories. Testing accuracy and precision with the increment of epochs for the YOLOv7 model are demonstrated in Figure 14.

**3.2. Results for the Custom Dataset.** In this section, the proposed models' performance has been evaluated using the custom dataset obtained by the authors of the manuscript. The proposed VGG16 and ResNet50 convolutional neural

networks have attained 99% and 98% accuracy, respectively, on the test data after the end of the tenth epoch. The confusion matrix in Figure 15 demonstrates how the models performed on the test dataset for eight classes of fruits. Figure 16 shows the behavior of training and validation accuracies concerning epoch numbers for the VGG16 technique. According to Figure 17, the training and validation losses of the VGG16 framework reduce significantly with the change of epochs. Interestingly, the results for the ResNet50 model for the custom dataset have not been reported, as the VGG16 model performed slightly better than the ResNet50 technique.

**3.3. Results for the Domain Adaptation Technique.** Next, the domain adaptation technique is applied to substantiate the efficiency of the proposed fruit detection and classification system. According to this approach, the deep learning model was initially trained on source images and evaluated on a dataset from a different source [27]. Various cross-domain adaptation experiments have been performed to evaluate the performance of the proposed fruit classification system. First, the proposed models were trained with the custom dataset and tested with the open-source FIDS-30 dataset. According to Table 6, in this case, the ResNet50 and VGG16 techniques exhibit testing accuracies of 85% and 86%, respectively. Next, the proposed models were trained with the FIDS-30 public dataset and tested with the custom dataset. For this case, the ResNet50 and VGG16 techniques exhibit testing accuracies of 98% and 99%, respectively, with the custom dataset.

**3.4. Android Development.** In this work, we have designed a simple Android smartphone application with a user-friendly interface and named it Fruit Holmes. We used

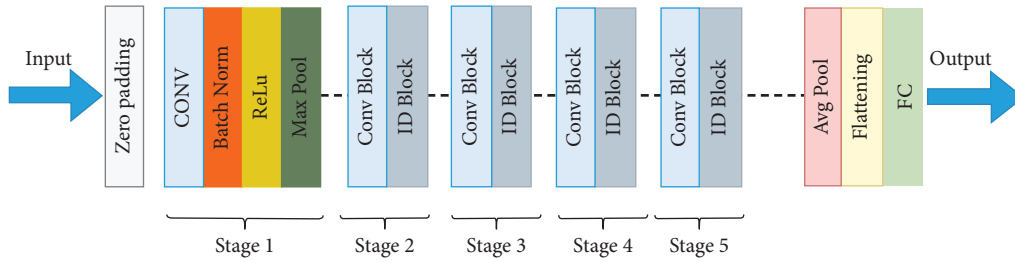


FIGURE 5: Architecture of the used ResNet50 architecture.

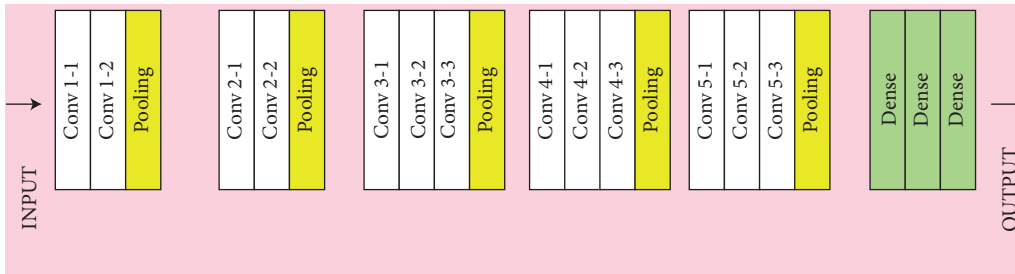


FIGURE 6: Architecture of the employed VGG16 framework.

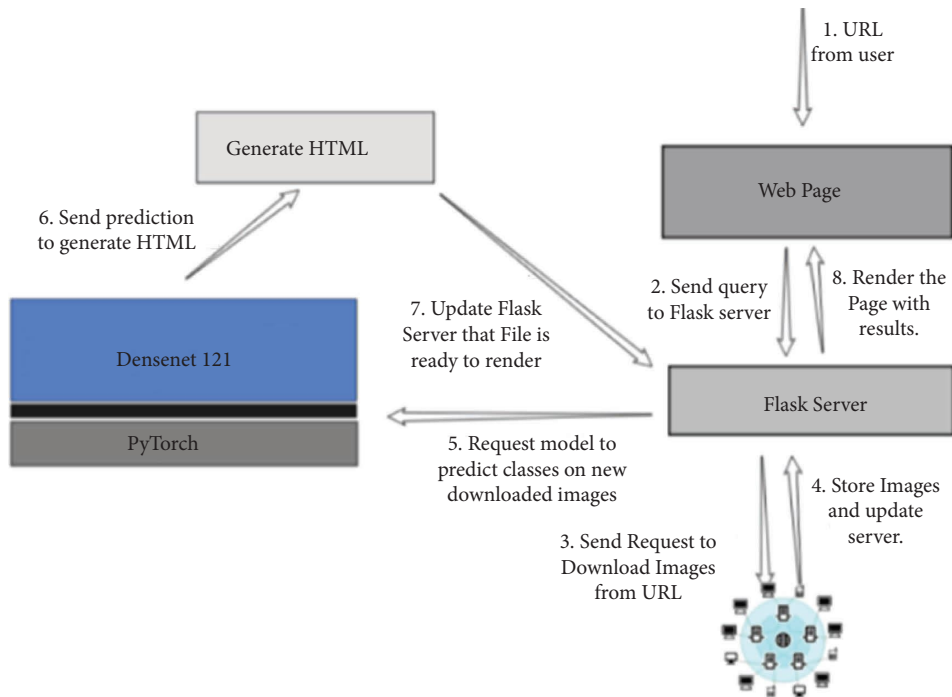


FIGURE 7: Deployment of the flask to create website and smartphone applications.

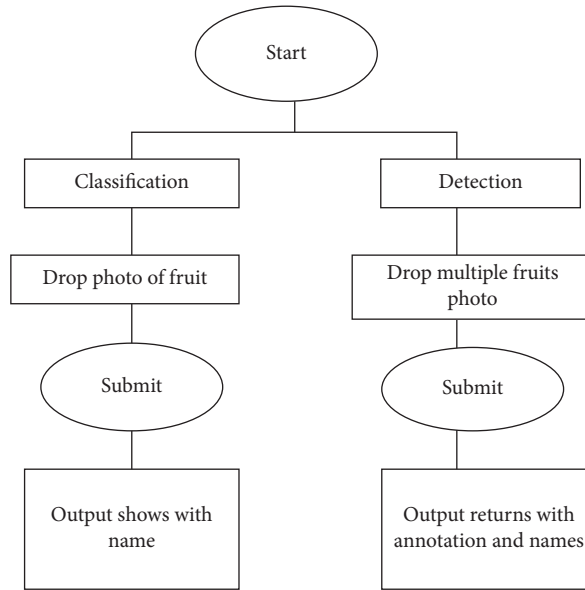


FIGURE 8: Work flowchart of the implementation of the proposed system in a web application.

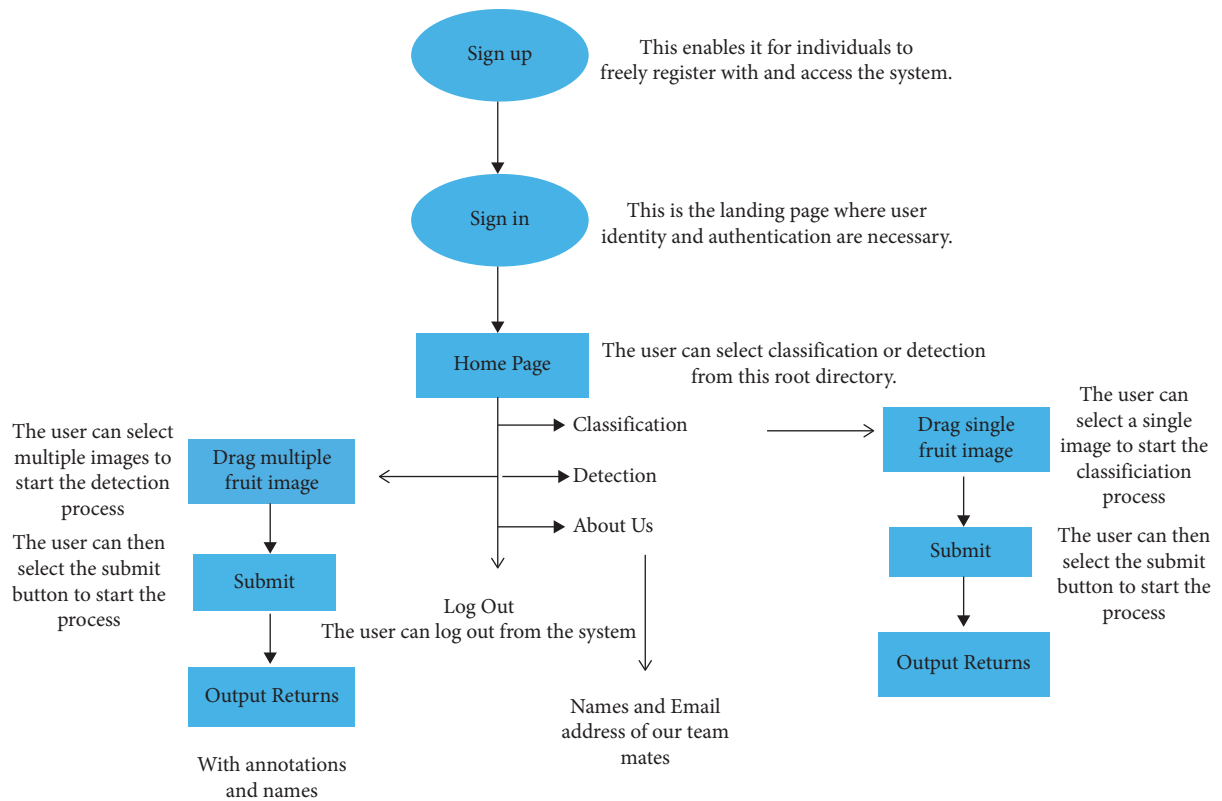


FIGURE 9: Working sequences in developing the proposed Android application of the fruit classification system.



TABLE 3: F1 score for all 30 classes in the ResNet50 model for the FIDS-30 dataset.

Class names	F1 scores
Acerolas	0.80
Apples	0.60
Apricots	0.50
Avocados	1
Bananas	0.89
Blackberries	1
Blueberries	0.80
Cantaloupes	0.80
Cherries	1
Coconuts	1
Figs	0
Grapefruits	1
Grapes	0.86
Guava	0.67
Kiwifruit	1
Mangos	1
Olives	1
Oranges	0.80
Passion fruit	1
Peaches	0.75
Pears	1
Pineapples	1
Plums	0.86
Pomegranates	1
Raspberries	0.80
Strawberries	0.80
Tomatoes	0.80
Watermelons	0.89
Lemons	0.86
Limes	1

TABLE 4: F1 score for all 30 classes in the VGG model for the FIDS-30 dataset.

Class names	F1 scores
Acerolas	0.57
Apples	1
Apricots	0.40
Avocados	0.89
Bananas	0.89
Blackberries	0.89
Blueberries	1
Cantaloupes	0.89
Cherries	0.86
Coconuts	1
Figs	0.89
Grapefruits	1
Grapes	0.80
Guava	0.57
Kiwifruit	1
Mangos	0.67
Olives	1
Oranges	0.67
Passion fruit	0.67
Peaches	0.86
Pears	1
Pineapples	0.75
Plums	0.86
Pomegranates	1
Raspberries	0.57
Strawberries	1

TABLE 4: Continued.

Class names	F1 scores
Tomatoes	1
Watermelons	1
Lemons	0.92
Limes	0.67

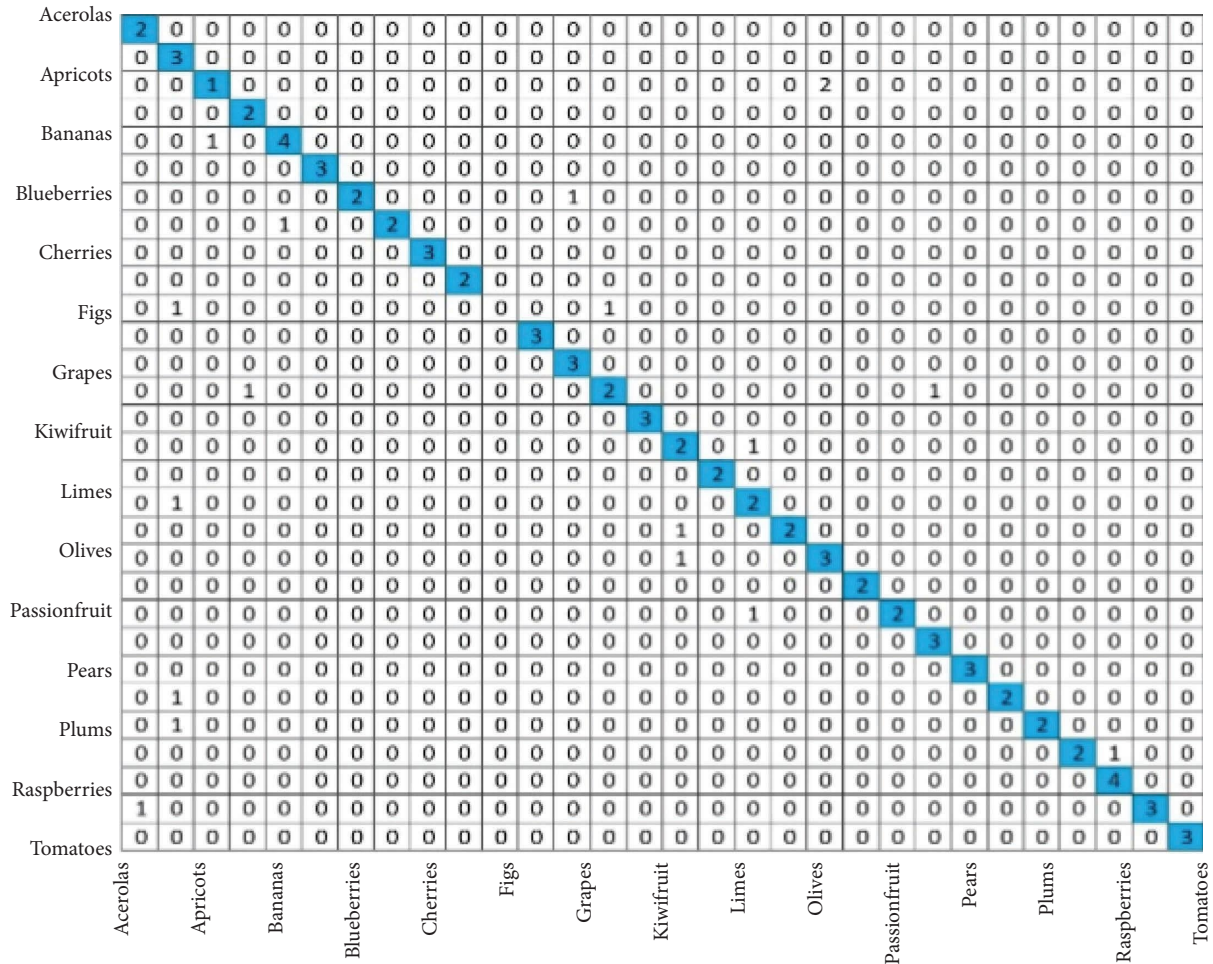


FIGURE 10: Confusion matrix of ResNet50 model for the FIDS-30 dataset.

Android Studio to create a remote Android app to implement several libraries. We used the VGG16 technique for the proposed Android application because it gives us more accurate results than other models. Next, we used a flask framework to connect the models. Flask assists us in creating the APK link. After copying the URL into the app-specific blank box, the app and models are connected. Finally, we add text-based and verbal speech

descriptions of the detected fruits to this app using the necessary libraries. After the output fruit has been detected, there will be an option “say.” After clicking that button, the designed mobile application will start reading the name and descriptions of the specific fruit. Figure 18 shows some example images of the proposed application in real-time for fruit classification (left) and detection (right).

Acerolas	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	
Apricots	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bananas	0	0	1	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Blueberries	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Cherries	1	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Figs	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Grapes	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Kiwifruit	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Limes	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Olives	0	0	0	0	0	0	0	0	0	0	0	1	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Passionfruit	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	
Pears	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	
Plums	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	
Raspberries	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	
Tomatoes	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	
	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0
	Acerolas	Apricots	Bananas	Blueberries	Cherries	Figs	Grapes	Kiwifruit	Limes	Olives	Passionfruit	Pears	Plums	Raspberries	Tomatoes															

FIGURE 11: Confusion matrix of VGG16 for the FIDS-30 dataset.

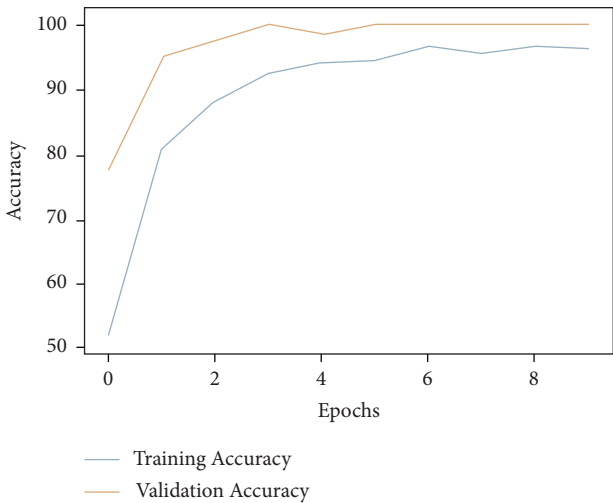


FIGURE 12: Training and validation accuracies vs. epoch graphs of the VGG16 network for FIDS-30 dataset.

Table 7 illustrates the comparison of the proposed fruit detection and classification system with other similar works on the FIDS-30 dataset. According to this table, the

proposed automatic fruit classification and detection system outperformed most of the works in terms of classification accuracy and other metrics.

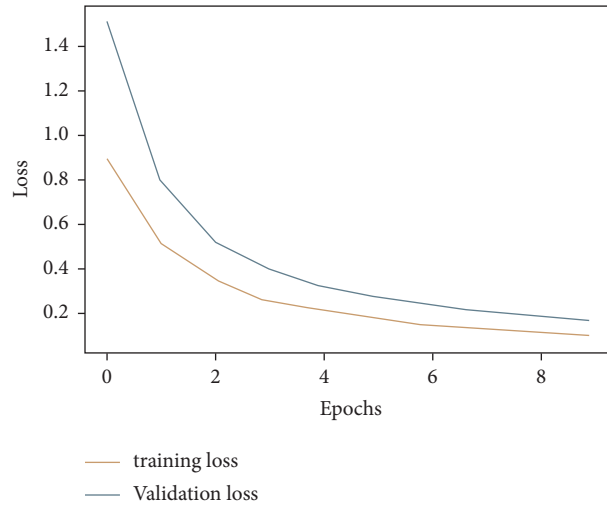


FIGURE 13: Training and validation loss vs. epochs of the VGG16 model for FIDS-30 dataset.

TABLE 5: Hyperparameters to train the YOLOv7 model.

Parameters	Values
Number of epochs	10
Batch size	16
Auto-anchor check	Disable
Optimizer	Adam
Early stopping	Enable
Dropout	0.25

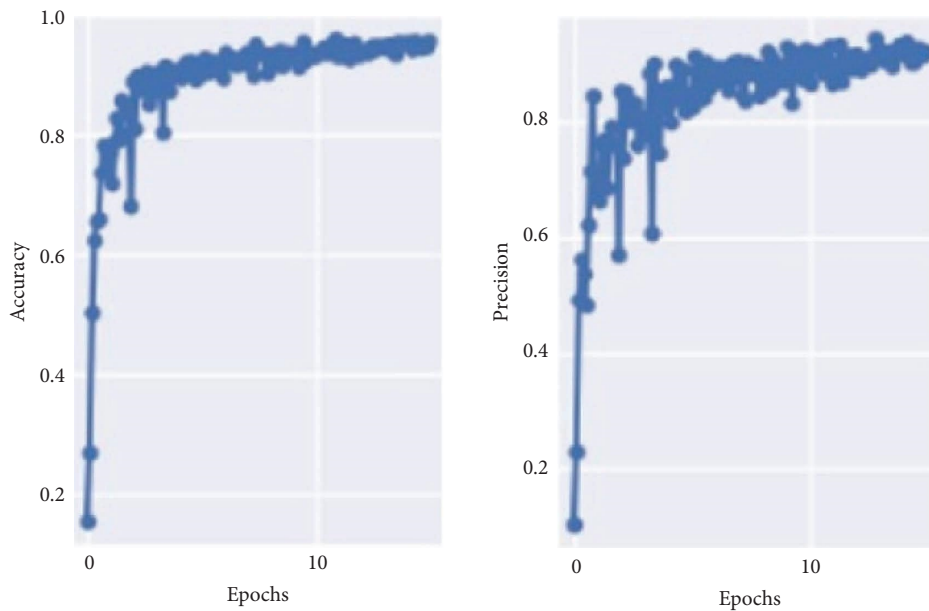


FIGURE 14: Accuracy and precision vs. epochs for the FIDS-30 dataset in YOLOv7.

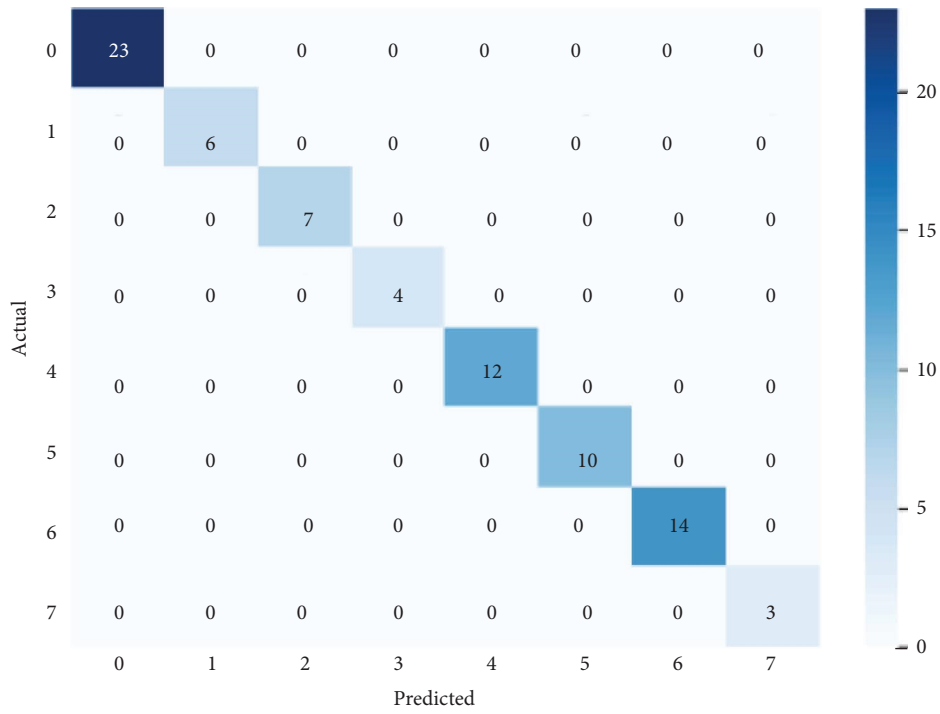


FIGURE 15: Confusion matrix of VGG16 technique for the custom dataset.

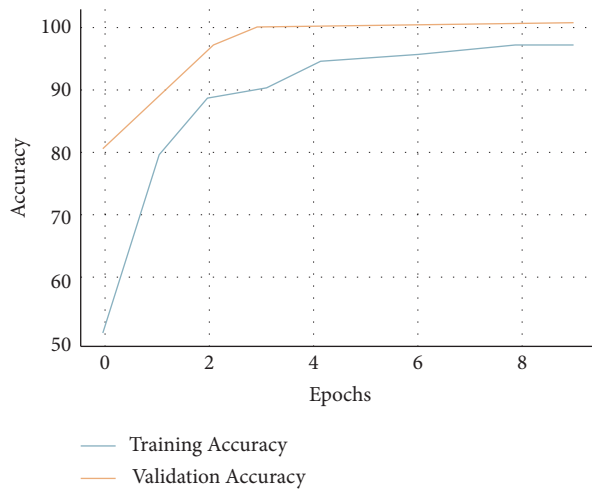


FIGURE 16: Training accuracy vs. validation accuracy graph for custom dataset in VGG16 model.

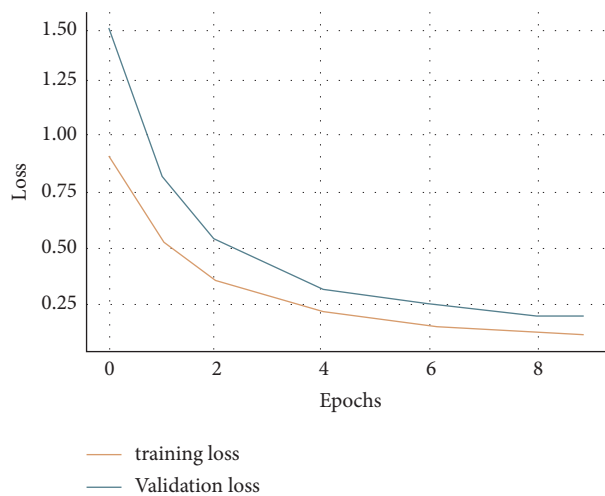


FIGURE 17: Training loss and validation loss graph for custom dataset in VGG16.

TABLE 6: Gained results from domain adaptation experiments.

Training datasets	ResNet50	VGG16
Custom dataset	Test accuracy -85% (FIDS-30)	Test accuracy-86% (FIDS-30)
FIDS-30 public dataset	Test accuracy -98% (custom data)	Test accuracy-99% (custom data)

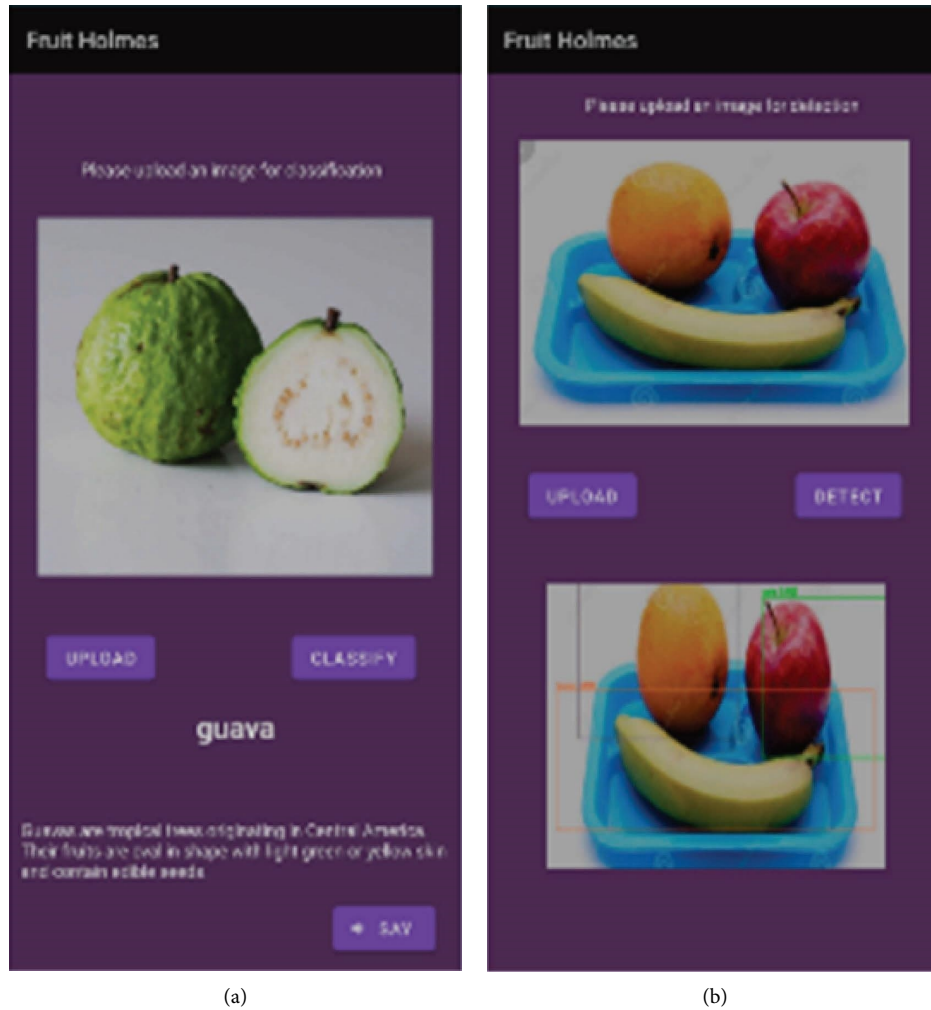


FIGURE 18: Examples of the sample work of the designed smartphone application interface.

TABLE 7: Comparison of the proposed system with other existing works on FIDS-30 dataset.

References	Networks	Accuracies (%)	Other metrics
[28]	ResNet50	89.16	<i>F1</i> score: 0.89
[29]	MobileNetv3	89.7	
[30]	Inceptionv3	89.7	
This work	YOLOv7	96.1	Precision: 0.93 recall: 0.89

## 4. Conclusions

This work has developed an automated fruit classification and detection system using deep learning techniques. Two datasets have been utilized in this paper, i.e., an open-source FIDS-30 dataset of 30 classes and a custom private dataset of 8 categories of fruits. After applying the conventional augmentation techniques, the deep learning models are applied. The YOLOv3 approach is applied for the detection of multiple fruits in images. VGG16 and ResNet50 models are employed for the automatic fruit classification. Finally, the deep learning models have been deployed into a website and Android smartphone application using the flask framework, API, and Android Studio. The proposed system is expected to use in industries and supermarkets and as an educational tool for children.

In the future, more advanced deep learning models with parameter tuning can be used to improve detection and classification accuracy and reduce memory usage. A larger dataset with more species and diverse pictures of fruits and synthetic images can be used. We would like to convert this proposed system to identify the defective and damaged fruits in the future. It would be an excellent invention for our farmers to do smart farming in this modern era.

## Data Availability

The data used to support the findings of this study are included in the article.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## References

- [1] D. Pem and R. Jeewon, "Fruit and vegetable intake: benefits and progress of nutrition education interventions- narrative review article," *Iranian Journal of Public Health*, vol. 44, no. 10, pp. 1309–1321, 2015.
- [2] R. C. Fierascu, E. Sieniawska, A. Ortan, I. Fierascu, and J. Xiao, "Fruits by-products - a source of valuable active principles. a short review," *Frontiers in Bioengineering and Biotechnology*, vol. 8, pp. 319–328, 2020.
- [3] S. R. Dubey and A. S. Jalal, "Species and variety detection of fruits and vegetables from images," *International Journal of Applied Pattern Recognition*, vol. 1, pp. 108–126, 2013.
- [4] R. A. A. Al-Fallujah, "Color, shape, and texture based fruit recognition system," *International Journal of Advanced Research in Computer Engineering & Technology*, vol. 5, pp. 2108–2112, 2016.
- [5] S. Arivazhagan, S. Newlin, N. Selva, and G. Lakshmanan, "Fruit recognition using color and texture features," *Journal of Emerging Trends in Computing and Information Sciences*, vol. 1, no. 2, pp. 90–94, 2010.
- [6] P. S. Dutch and K. Jayasimha, "Intra class vegetable recognition system using deep learning," in *Proceedings of the International Conference on Intelligent Computing and Control Systems (ICICCS)*, pp. 602–606, Maisamaguda, India, June 2020.
- [7] A. Nasiri, A. Taheri-Garavand, and Y. D. Zhang, "Image-based deep learning automated sorting of date fruit," *Post-harvest Biology and Technology*, vol. 153, pp. 133–141, 2019.
- [8] F. López-García, G. Andreu-García, J. Blasco, N. Aleixos, and J. M. Valiente, "Automatic detection of skin defects in citrus fruits using a multivariate image analysis approach," *Computers and Electronics in Agriculture*, vol. 71, no. 2, pp. 189–197, 2010.
- [9] J. L. Joseph, V. A. Kumar, and S. P. Mathew, *Fruit Classification Using Deep Learning*, vol. 756, Springer, Singapore.
- [10] J. M. Ponce, A. Aquino, and J. M. Andújar, "Olive-fruit variety classification by means of image processing and convolutional neural networks," *IEEE Access*, vol. 7, pp. 147629–147641, 2019.
- [11] R. Yamparala, R. Challa, V. Kantharao, and P. S. R. Krishna, "Computerized classification of fruits using convolution neural network," in *Proceedings of the International Conference on Smart Structures and Systems (ICSSS)*, Chennai, India, July 2020.
- [12] S. Lu, Z. Lu, S. Aok, and L. Graham, "Fruit classification based on six layer convolutional neural network," in *Proceedings of the International Conference on Digital Signal Processing (DSP)*, Shanghai, China, November 2018.
- [13] W. C. Seng and S. H. Mirisae, "A new method for fruits recognition system," in *Proceedings of the International Conference on Electrical Engineering and Informatics*, pp. 130–134, Selangor, Malaysia, August 2009.
- [14] O. O. Abayomi-Alli, R. Damaševičius, S. Misra, and R. Maskeliunas, "Cassava disease recognition from low-quality images using enhanced data augmentation model and deep learning," *Expert Systems*, vol. 38, 2021.
- [15] A. Almadhor, H. T. Rauf, M. I. U. Lali, R. Damasevicius, B. Alouffi, and A. Alharbi, "AI-driven framework for recognition of guava plant diseases through machine learning from DSLR camera sensor based high resolution imagery," *Sensors*, vol. 21, no. 11, pp. 3830–3919, 2021.
- [16] D. O. Oyewola, E. G. Dada, S. Misra, and R. Damaševičius, "Detecting cassava mosaic disease using a deep residual convolutional neural network with distinct block processing," *PeerJ Computer Science*, vol. 7, pp. 3522–e415, 2021.
- [17] H. Muresan and M. Oltean, "Fruit recognition from images using deep learning," Technical Report, Babes-Bolyai University, Cluj-Napoca, Romania, 2017.
- [18] S. K. Behera, A. K. Rath, and P. K. Sethy, "Fruit recognition using support vector machine based on deep features," *Karbala International Journal of Modern Science*, vol. 6, no. 2, pp. 235–245, 2020.
- [19] L. G. Nouaze, J. C. T. Mbouembe, and J. H. Kim, "Yolo-Tomato: a robust algorithm for tomato detection based on YOLOv3," *Sensors*, vol. 20, 2020.
- [20] Q. Xiang, X. Wang, R. Li, G. Zhang, J. Lai, and Q. Hu, "Fruit image classification based on MobileNetV2 with transfer learning technique," in *Proceedings of the International Conference on Computer Science and Application Engineering*, pp. 1–7, Sanya, China, October 2019.
- [21] C. Wang, A. Bochkovskiy, and H. M. Liao, "YOLOv7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," pp. 1–16, 2022, <https://arxiv.org/abs/2207.02696>.
- [22] A. S. B. Reddy and D. S. Juliet, "Transfer learning with ResNet-50 for malaria cell-image classification," in *Proceedings of the International Conference on Communication and Signal Processing (ICCSP)*, pp. 945–949, Melmaruvathur, India, April 2019.

- [23] Q. A. Al-Haija and A. Adebajo, "Breast cancer diagnosis in histopathological images using ResNet-50 convolutional neural network," in *Proceedings of the International Iot, Electronics And Mechatronics Conference (IEMTRONICS)*, pp. 1–7, Toronto, Canada, September 2020.
- [24] H. Yang, J. Ni, J. Gao, Z. Han, and T. Luan, "A novel method for peanut variety identification and classification by Improved VGG16," *Scientific Reports*, vol. 11, no. 1, Article ID 15756, 2021.
- [25] A. K. Rangarajan and R. Purushothaman, "Disease classification in eggplant using pre-trained VGG16 and MSVM," *Scientific Reports*, vol. 10, pp. 1–11, 2020.
- [26] M. R. Mufid, A. Basofi, M. U. H. Al Rasyid, I. F. Rochimansyah, and A. Rokhim, "Design an MVC model using Python for flask framework development," in *Proceedings of the International Electronics Symposium (IES)*, pp. 214–219, Surabaya, Indonesia, September 2019.
- [27] B. Chen and S. Tan, "Feature transfer: unsupervised domain adaptation for cross-domain deepfake detection," *Security and Communication Networks*, vol. 2021, Article ID 9942754, 8 pages, 2021.
- [28] K. Munir, A. I. Umar, and W. Yousaf, "Automatic fruits classification system based on deep neural network," *NUST Journal of Engineering Sciences*, vol. 13, pp. 37–44, 2020.
- [29] P. Jain, P. Chawla, M. Masud, S. Mahajan, and A. Kant Pandit, "Automated identification algorithm using CNN for computer vision in smart refrigerators," *Computers, Materials & Continua*, vol. 71, no. 2, pp. 3337–3353, 2022.
- [30] P. Jain and P. Chawla, "Smart module design for refrigerators based on inception-V3 CNN architecture," in *Proceedings of the International Conference on Electronics and Sustainable Communication Systems*, pp. 1852–1859, Coimbatore, India, August 2021.