Hindawi
Mathematical Problems in Engineering
Volume 2022, Article ID 9210947, 29 pages
https://doi.org/10.1155/2022/9210947

Research Article

Recent Advancements in Fruit Detection and Classification Using Deep Learning Techniques

Chiagoziem C. Ukwuoma,[1] Qin Zhiguang,[1] Md Belal Bin Heyat,[2,3] Liaqat Ali,[4] Zahra Almaspoor,[5] and Happy N. Monday[6]

[1] *School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu 611731, Sichuan, China*
[2] *IoT Research Center, College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, Guangdong, China*
[3] *Department of Science and Engineering, Novel Global Community Education Foundation, Hebersham, NSW, Australia*
[4] *Department of Electrical Engineering, University of Science and Technology Bannu, Bannu, Pakistan*
[5] *Department of Statistics, Yazd University, Yazd 89175-741, Iran*
[6] *School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, Sichuan, China*

Correspondence should be addressed to Zahra Almaspoor; z.almaspoor@stu.yazd.ac.ir

Received 31 October 2021; Revised 15 December 2021; Accepted 28 December 2021; Published 31 January 2022

Academic Editor: Dost Muhammad Khan

Recent advances in computer vision have allowed broad applications in every area of life, and agriculture is not left out. For the agri-food industry, the use of advanced technology is essential. Owing to deep learning's capability to learn robust features from images, it has witnessed enormous application in several fields. Fruit detection and classification remains challenging due to the form, color, and texture of different fruit species. While studying the impact of computer vision on fruit detection and classification, we pointed out that till 2018 many conventional machine learning methods were utilized while a few methods exploited the application of deep learning methods for fruit detection and classification. This has prompted us to pursue an extensive study on surveying and implementing deep learning models for fruit detection and classification. In this article, we intensively discussed the datasets used by many scholars, the practical descriptors, the model's implementation, and the challenges of using deep learning to detect and categorize fruits. Lastly, we summarized the results of different deep learning methods applied in previous studies for the purpose of fruit detection and classification. This review covers the study of recently published articles that utilized deep learning models for fruit identification and classification. Additionally, we also implemented from scratch a deep learning model for fruit classification using the popular dataset "Fruit 360" to make it easier for beginner researchers in the field of agriculture to understand the role of deep learning in the agriculture domain.

1. Introduction

tImage classification is a very active research direction in many areas and plays a very important role. Image recognition serves various uses including video analysis, face recognition, image classification, etc. Deep learning (DL) is a subdomain of machine learning (ML) that has shown excellent results in image identification [1]. DL utilizes the multi-layer structure to process image characteristics which significantly increase image recognition efficiency [2]. In other words, the application of image recognition and DL is becoming a concept within the field of logistics and supply chain. For instance, image recognition can better facilitate logistics and transportation and solve the errors in many fully automated transport vehicles due to large-scale track identification errors [3]. Another application of DL is the classification of fruits. DL can effectively extract image characteristics and then introduce classification.

Current computer vision (CV) developments have shown outstanding results in many areas of life. Fruit detection and classification has demonstrated to be a complex and a challenging task. For some economic sectors, both for wholesale and retail markets, research in fruit processing is very important including the processing industry. These factors have motivated researchers who developed various methods to process fruits automatically, either to identify them or to estimate their quality efficiently. Over the past few years, agricultural industries such as food processing, marketing, packaging, and fruit classification have become more focused research direction. Processing and sorting of unique crop plants such as orange, cherry, apple, mango, and citrus are labor and time intensive due to a number of varieties of same fruit, for instance, more than 7,000 apple varieties are produced worldwide (http://usapple.org). Automation can thus minimize labor costs and quickly increase productivity. In early research, scientists suggested different methods from CV in order to manually extract features from fruit and ML for classifying the CV features. Through CV algorithms, color, form, size, and texture characteristics of fruit are used for classification algorithms [4–6]. Most of them utilized preprocessing or feature extraction through CV in conjunction with different classifiers. However, most evolved classifiers are not robust for all fruit types which results in higher misclassification rates. For quality evaluation and robotic harvesting, fruit detection and classification has seen some implementation of DL methods, but these methods are having limited classes and small datasets. Literature analyses of new methods to classify fruits were published in 2017 by Liu et al. [7]. They conducted a search on the recent published work in the area of fruit detection and classification. They selected, among others, eleven publications that were significant. Among the selected publications, just 4 works discuss DL or other conventional ML methods [8–11]. This demonstrates that while convolutional neural networks (CNNs) were of great interest at that time, still many studies on fruit detection and classification did not utilize CNNs. They also suggested that DL models should be more commonly used, especially CNNs, because they have shown state-of-the-art performance on image classification in biomedical and health informatics.

In the area of object detection and image recognition, CNN has become a highly important model for study. CNN's ability to extract attributes automatically from an input image made it more robust to use. In CNN, the image can be fed into the network directly in contrast to conventional CV-based feature extraction algorithms, and thus it eliminates preprocessing and extraction processes. Convolutional layers (CLs), pooling layers (PLs), and fully connected (FC) layers are the three layers that make a classic CNN. After winning the ImageNet award, CNN gained much attention [12]. The various CNN models developed by many scholars by varying width and depth of layers were then discussed in [13–15]. Detection and classification of fruits is a relatively complex problem due to the great variety of intraclass forms, colors, and textures. These limitations have led to a shortage of automated fruit classification systems for multiple classes. A more complex information system of fruit automated detection and classification may be useful in picking right fruit with the correct nutrition. It can support children and people with visual impairments and develop self-checking supermarkets. We identify fruit classification tasks as class determination by their particular type in order to define the study areas of our review. Fruit detection, on the other hand, is geared towards automatic harvesting. Based on DL's high level of attention over recent years and contrary to current surveys, we present a thorough review of the use of DL in the processing of fruit images, particularly in areas of classification and detection. This paper offers a comparative survey of existing fruit detection and classification approaches. The contributions of the study are as follows:

(i) Due to the novelty of DL application in the studied area, to the best of our knowledge, we present the first thoroughly investigative study on the application of DL in the field of detection and classification of fruits from 2017 to date.

(ii) We define and discuss the architectures used in recent research and the publicly accessible datasets for fruit detection and classification.

(iii) Two main areas in the agri-food industry related to the classification of fruits and fruit detection were deeply investigated. We summarized the major aspects, properties, and results of the work carried out by numerous researchers.

(iv) Aiming to give a better understanding of how DL (CNN) models are implemented, we present a theoretical background on CNNs. Additionally, for more elaboration of the topic, we also conducted an experiment on fruit classification using the popular dataset, namely, Fruit 360.

(v) To further illustrate deployment of transfer learning for the purpose of fruit detection and classification, we deployed the ResNet-50 architecture via transfer learning and compared it with the CNN model developed and trained from scratch. We utilized the newly introduced one-policy learning rate.

(vi) Lastly, the application of DL recently has progressed sharply and witnessed improvement in the area of object detection and classification. We also explored the newly proposed concept and thus also deployed an adversarial attack mechanism to the trained model in order to illustrate the need for application of attention mechanism and adversarial defense mechanism in the detection and classification of fruit models.

This paper is organized as follows. Section 2 talks about the background of the study, i.e., it summarizes the role of CV methods applied in the domain of fruit detection and classification. Section 3 introduces an overview of the DL, i.e., CNN model and the prerequisite to implementing a CNN model. Section 4 talks about fruit detection and classification as well as various DL models applied in the field. We discuss about the utilized datasets by different authors in Section 5. In Section 6, we discuss evaluation metrics while Section 7 illustrates our experimental analysis on fruit classification based on a DL model using the popular Fruit 360 dataset. We further illustrate the use of transfer learning in the fruit detection and classification field and compare the result with the CNN models trained and developed from scratch. Section 8 highlights the future needs in the design of fruit detection and classification algorithms, and finally, we conclude our study in Section 9.

2. Background

DL is described as research direction in ML for the creation of complex models that solve complex tasks. DL models have been developed using supervised learning techniques and based on artificial neural networks. The fact that these models are proven capable of understanding the problem that needs to be resolved in the hierarchy of concepts is an essential component of DL [16]. CV is one of the areas in which DL achieved state-of-the-art performance. More precisely, on problems like optical character recognition [17], object detection [18], semantic segmentation [19], and image classification [20], DL models were found to achieve the best results. Thus, presently, most of the practical solutions for these concerns are DL based.

In current history, a number of papers have shown tremendous progress in modelling frameworks with DL for image recognition. With the CNN, the authors in [13] used DL for the identification of fruit objects and achieved an outstanding degree of both learning and recognition. In [21], the studies evaluated two CNN architectures, namely, MobileNet and Inception, as the classifiers of ten classes of fruit. They claimed that MobileNet was propagating images with about the same accuracy significantly faster [22]. However, clementines and kiwifruits were difficult to predict which may be due to the option of training and testing a number of images taken with a video camera mounted to the conceptual retail market systems and simultaneously obtained from ImageNet. Abdul Hamid et al. [23] introduced a comparative analysis between the bag of features (BoF), AlexNet, and a CNN for fruit recognition. The results showed that all three methods were very reliable, but the most rapidly available prediction for recognition was the CNN technique. In turn, two deep neural networks, with excellent results in the precise fruit classification of both bases, were proposed and tested in [24] utilizing simple and more complex datasets. A number of computational experiments have been presented in [25] to train various CNN architectures to detect fruit. For a similar reason in [14], a 13-layer CNN was proposed.

There is significant evidence of attempts to develop an automated fruit detection and classification system. Fruit identification faces a variety of challenges due to its irregular form, size, and varying color. A lot of research has been undertaken to define strategies for coping with these problems. Virtually every physical component of fruits is known to be feasibly classifiable. ML methods have been investigated and continuously developed for this role, both by the empirical network and neural network (NN) [9, 13, 14]. In this respect, most attempts are made to combine image analysis as feature details with ML framework for classification/recognition [26–29]. These efforts identify and represent a physical feature in a vision-based machine representation called the description of features or feature description. These characteristics are then given to converge on a qualitative result as the input to the classification algorithm. Numerous techniques for the description and classification of features have been studied, but substantial overhaul and enhancement are required to achieve an efficient classification. A thorough reconsideration for all associated questions of features, sensors, and classification algorithms is important for an effective system of detection and classification for fruit.

3. Overview of the Convolutional Neural Network (CNN)

The key DL architectures for image classification are the convolutional neural networks (CNNs) [30–35]. We note that the use of CNN for recognition of fruit has increased dramatically over the last three years (2018 to 2021) and has generated excellent results through either new models or pretrained transfer-learning networks. CNNs are kinds of artificial neural networks that operate in at least one of their layers with convolution [36]. CNNs have been seen as a competitive tool for image classification in several fields since 2012, when Krizhevsky et al. [12] won the ImageNet Competition (ILSVRC) [37]. As an effective tool for image classification in many areas, CNNs have gained great popularity. Particularly, in agriculture, fruit classification [14, 38, 39] and fruit detection [40, 41] applied CNN-based approaches. Evolution of CNN is traced back to multi-layer neural network that was first proposed by LeCun et al. [42] in 1998. Multi-layer perceptron is a regularized version of CNN. The multi-layer perceptron typically means that all networks are completely connected, i.e., each neuron in one layer is linked to the next layer. Unlike CNNs, convolution operations are used at least in one of their layers [13]. Figure 1 shows the schematic diagram of basic CNN architecture, whereas mathematically, we can define the structure of the convolution neural network operation from layer to layer as

$$x^1 \longrightarrow \left(w^1\right) \longrightarrow x^2 \longrightarrow \ \ldots \ \longrightarrow x^{L-1} \longrightarrow \left(w^{L-1}\right) \longrightarrow x^L \longrightarrow \left(w^L\right) \longrightarrow z, \tag{1}$$

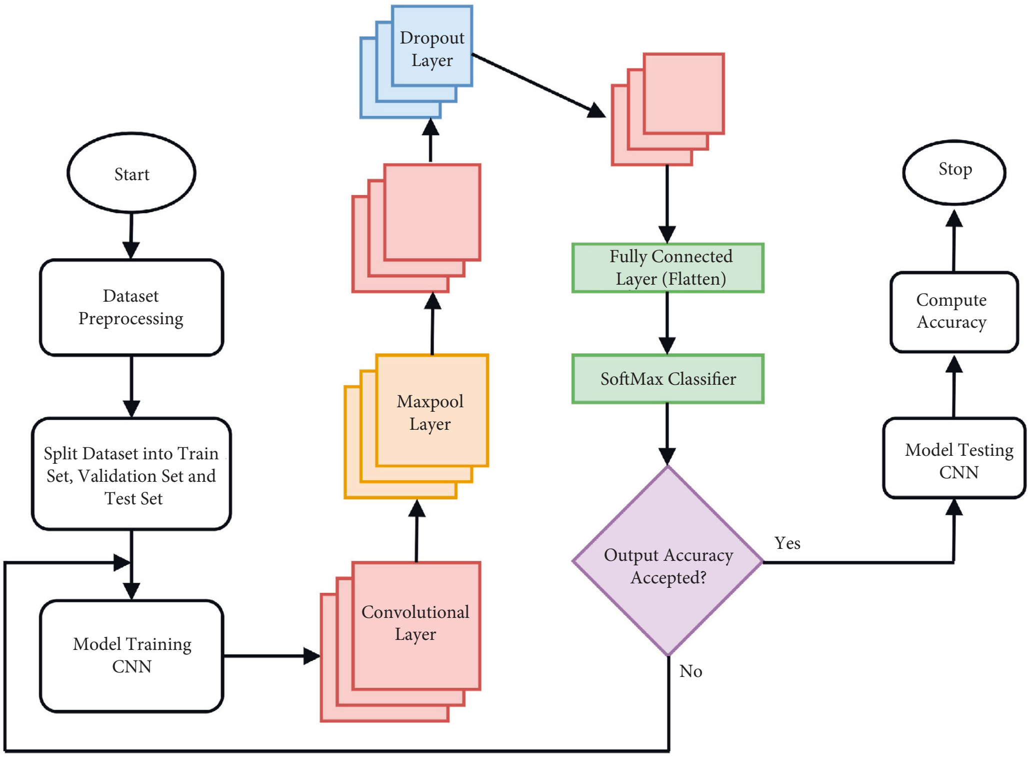


Figure 1: Schematic diagram of basic CNN architecture for fruit detection and classification.

where x^1 denotes the input, usually an image of order 3 tensor $n_{H,} n_{W,} n_C$ (image height, image width, and channel (usually RGB)). It goes into the first layer which is denoted as (w^1) where w^1 represents the parameters involved in the preprocessing is. x^2 denotes the output of the first layer after convolution and acts as an input to the subsequent layer. When converting the input image, we make use of the filters f, and it must be the same size with the channel size of the image, and thus the dimension of the filter is represented as $di\ m(filter) = (f, f, n_c)$. Mathematically, for a given image denoted as I and filter denoted as K, we have

$$conv(I,K)_{x,y} = \sum_{i=1}^{nH}\sum_{j=1}^{nw}\sum_{k=1}^{nc} K_{i,j,k} I_{x+i-1,y+j-1,k}. \tag{2}$$

After convolution, (2) becomes

$$\begin{aligned}\text{dimdim}\ (conv(I,K)) &= \left(\left\lfloor \frac{n_H + 2p - f}{s} + 1 \right\rfloor, \left\lfloor \frac{n_W + 2p - f}{s} + 1 \right\rfloor\right); s < 0 \\ &= (n_H + 2p - f, n_w + 2p - f); s = 0,\end{aligned} \tag{3}$$

where $\lfloor x \rfloor$ denotes x floor function and p and s denote padding and stride, respectively. Convolution is followed by pooling which can either be average pooling or max pooling. When pooling is done to the convolution layer, (3) becomes

$$\begin{aligned}\text{dimdim}\ (pooling(image)) &= \left(\left\lfloor \frac{n_H + 2p - f}{s} + 1 \right\rfloor, \left\lfloor \frac{n_W + 2p - f}{s} + 1 \right\rfloor, n_C\right); s > 0 \\ &= (n_H + 2p - f, n_w + 2p - f, n_c); s = 0.\end{aligned} \tag{4}$$

A fully connected layer takes in a vector $a^{[i-1]}$ and returns a vector $a^{[i]}$. Let us consider the jth one in an ith layer, and the full connected layer can be denoted mathematically as

$$z_j^{[i]} = \sum_{l=1}^{n_{i-1}} w_{j,l}^{[i]} a_l^{[i-1]} + b_j^{[i]} \longrightarrow a_j^{[i]} = \varphi^{[i]}\left(z_j^{[i]}\right), \tag{5}$$

where $b_j^{[i]}$ denotes bias and $\varphi^{[i]}$ denotes the activation function. The input to the fully connected layer might be the output of a convolution or pooling layer with the dimension $n_H^{[i-1]}, n_W^{[i-1]}, n_C^{[i-1]}$, and thus there is need for flattening into a 1D vector $n_H^{[i-1]} * n_W^{[i-1]} * n_C^{[i-1]}$:

$$n_{i-1} = n_H^{[i-1]}, n_W^{[i-1]}, n_C^{[i-1]}. \tag{6}$$

In (1), suppose we are tackling a classification task having C classes. The general procedure is to output x^L as a $C\ D$ vector in which the posterior probability x^1 is obtained from ith class. Transform x^{L-1} into $(L-1)th$ using SoftMax sets x^L as the probability mass function. For the loss calculation, suppose the ground truth image is denoted as t and is the corresponding value for the input x^1, and we calculate the difference between the predicted value x^L and the target t which can be represented mathematically as

$$z = \frac{1}{2}\left\| t - x^L \right\|^2. \tag{7}$$

The CNN architecture contains several phases comprising the parts below.

3.1. Convolution Layer.

The key construction block used in CNN is the convolutional layer (ConV layer). It comprises a series of filters K, which is learned (i.e., kernels) and which are almost always square and each filter has a width and a height. These filters are tiny (in spatial dimensions) but extend to the maximum depth of the volume [43].

3.2. Activation Function (AF).

An activation layer accepts the input volume and applies the AF given [44]. Since an element-specific activation function is applied, the activation layer output is always the same as the input dimension. We use a non-linear activation layer after each ConV layer in a CNN. We have numerous activation functions such as the sigmoid function, tanh function (hyperbolic tangent), ReLU (rectified linear unit function), ELU, or any other Leaky ReLU variants. The sigmoid activation function is defined as

$$\frac{1}{1 + e^{-x}}. \tag{8}$$

It is a non-linear function that outputs multiple neurons when passed through a sigmoid function as the activation function becomes non-linear. It ranges from 0 to 1 with a shape. The tanh activation function is a mathematically twisted version of the sigmoid function and performs better than the sigmoid function. The value ranges from −1 to +1, and it is mathematically defined as

$$\frac{e^x - e^{-x}}{e^x - e^{-x}}\ or\ 2 * sigmoi\ d(2x) - 1. \tag{9}$$

The ReLU (rectified linear unit function) is currently the most widely used activation layer which outputs x if x is positive and otherwise zero. It is mathematically illustrated as

$$A(x) = x\ if\ x \geq 0,\ otherwise\ 0. \tag{10}$$

The ReLU activation function is computational costly compared to sigmoid and tanh activation function. The improved version of the ReLU activation function is the Leaky ReLU function which, instead of defining the ReLU function 0 for x less than 0, defines a smaller linear component. Mathematically, we define the Leaky ReLU activation function as

$$f(x) = x\ \ if\ x \geq 0\ otherwise\ \alpha x. \tag{11}$$

3.3. Pooling or Subsampling Layer.

It decreases the number and spatial size of convolutional outputs by cutting back on network parameters. It helps us control overfitting. Subsampling layers operate individually for all input depths utilizing either the max pooling usually carried out in the center of the CNN-architectural structure to minimize the size of the input or the average pooling usually used as the last network layer in the process (such as ResNet, GoogleNet, and Squeeze Net) where FC layers are to be avoided.

3.4. Fully Connected Layers (FC Layers).

FC layers are often placed at the end of the network. FC layers make use of the outcomes of the process of convolution and pooling for the classification of image into a class (i.e., label). In FC layers, neurons are completely connected to the previous layer, i.e., the activation layer as default for feedforward neural networks. Typically, the SoftMax function is used for a multiclass classification task, where each probability values falls between [0, 1] and their overall amount equal to 1. Finally, each neuron determines one specific label value.

3.5. Dropout.

Dropout is basically a type of regularization which aims to prevent overfitting, perhaps at the detriment of training accuracy, by increasing test accuracy [45, 46]. The purpose we use dropout is because we wish to minimize overfitting by altering the network architecture directly during training. Random dropouts make sure that no single node in the network "activates" when presented with a specific pattern. (12) denotes the mathematical representation of a dropout:t

$$E\left[\frac{\partial E_D}{\partial w_i}\right] = \frac{\partial E_N}{\partial w_i} + wipi(1 - pi)I_i^2, \tag{12}$$

where ∂ denotes the dropout rate $(\partial = 1\ with\ probabilit\ y\ p\ an\ d\ 0\ otherwise)$.

3.6. Hyperparameters.

Hyperparameters are parameters that are very crucial in deploying a neural network. They are specific values used in controlling the learning process of the network. Every task is of different category, and the approach of solving it differs; thus, hyperparameters help us to refine our model to attend to the specific task at hand. Tuning hyperparameters of deep neural network is a very difficult task as it slows to train because of different parameters to configure [47]. We will briefly describe the common and widely employed hyperparameters by researchers in training a neural network.

3.6.1. Learning Rate. It is the pace at which the weights of the neural network change between iterations. Large swings in the weights may result from a high learning rate, and we may never reach their ideal values. A low learning rate is desirable, but it will take more iterations for the model to converge. Currently, the use of adaptive learning rate or momentum-based method helps researchers to start with a fixed learning rate, gradually decreasing the learning rate to optima. Using momentum, we calculate the learning rate in terms of decay rate:

$$n_{n+1} = \frac{\eta_n}{1 + d_n}, \tag{13}$$

where η denotes learning rate, d denotes decay parameters, and n denotes the iteration step. Calculating the learning rate in terms of schedule, we calculate the decay application:

$$\eta_n = n_0 d^{[1+n/r]}, \tag{14}$$

where η_n denotes learning rate at n iteration, n_0 denotes the initial learning rate, and d denotes the rate at which it drops (d = 0.5 means that 50% of dropout) while r denotes how often the dropout is applied (10 indicates that at every 10 epochs, the learning rate should drop). Exponential learning scheduler is calculated as

$$\eta_n = n_0 e^{-dn}, \tag{15}$$

where d denotes the decay parameter. Using the adaptive learning rate, one can choose from the different adaptive gradient descent algorithms such as Adam, AdaDelta, AdaGrad, and RMSprop which are embedded into the deep learning libraries (Keras and PyTorch).

3.6.2. Number of Epochs. This indicates the number of times the whole training set is passed into the model. When training from scratch, it is advised to train for a long epoch to allow the network to learn very well, while during training, using a pretrained model, we use small epoch since the model is a pretrained model meaning that it is initially with a larger dataset.

3.6.3. Batch Size. Batch size basically explains the number of subsamples to be fed into the model after which an update on the parameter happens.

3.6.4. Momentum. This is usually to help us know the direction of the next iteration with the knowledge of the previous iteration. Basically, we use values between the range of 0.5 and 0.9, and it is to avoid oscillation.

3.7. Prerequisite to Implementing a CNN Model. Before implementing a convolution neural network, the following steps are mandatory.

3.7.1. Defining Your Work Objective. It is very mandatory to know the objective of the work before deciding if CNN will be the best model for the work or not.

3.7.2. Defining the CNN Architecture. Having known your objective, defining the architecture to use is the next step. This involves the description of the number of layers, as well as the size and number of filters for each layer.

3.7.3. Loss Function. This indicates the difference between the specified ground truth labels and the network outputs. The mean squared error function is usually used. Therefore, losses must be reduced in order to find and optimize the participation of every weight. Currently, researchers widely make use of Adam in place of the gradient descent algorithm as a part derivative of the loss function for the reduction procedure.

3.7.4. Training Dataset. The available data are generally divided into three subsets: a training set to train the network, the validation set to evaluate the model during the training process, and the testing set to evaluate the final trained model. Many CNN frameworks require all training details to be in the same form (i.e., dimensions). Preprocessing data is therefore the first step in normalizing the data before the training process.

4. Fruit Detection and Classification

Description of recognition (detection and classification) could also be interpreted in various ways:

(i) The identification of a fruit (differentiating a fruit and an object, e.g., a leaf and a background).

(ii) Classification of the fruit species (e.g., orange and tangelo).

(iii) Recognition of a number of species of fruit (e.g., Crimson Snow Apple from Granny Smith Apple).

The right approach of finding the right species and the right selection of fruits is to recognize the nature of the problem presented in this article. Classification of fruit, due to the enormous numbers of varieties [48], is a relatively complex problem. In species and varieties, significant variations in appearance occur including irregular forms, colors, and textures. In addition, images narrowly cover the camera's lightning landscape, distance, and angle which all contribute to blurred images. The part or entire occlusion of the object is another concern. This weakness led to the absence in real-life implementations of multi-class automated fruit classification systems [48]. Several investigations are conducted to identify and classify fruits with various objectives and applications. Figure 2 illustrates the implementation steps of object detection and classification which

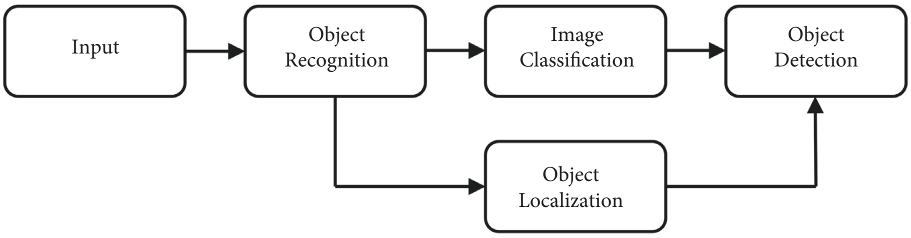


Figure 2: Basic implementation steps of object detection and classification.

is the same procedure for fruit detection and classification. In this survey, we analyzed in depth the various deep learning models applied to identification and classification of fruits.

4.1. Preprocessing and Segmentation. Preprocessing and segmentation is a vital step in the field of classification and detection. Since fruits vary in shape, size, color, and texture, preprocessing is the first and most important thing to do in the task of fruit detection and classification. During the preprocessing stage, images taken are preprocessed to remove the background noise, thereby extracting the fruit image, as shown in Figure 3. After that, most researchers convert the image into a grey image from RGB before converting to binary image. Since the introduction of DL, feature extraction has been the widely used preprocessing step after dataset acquisition. Techniques such as FCH, MI, and so on are used to extract the fruit features (shape, color, and size) before converting them into vector features.

If you want to find foreground items in images with stationary backgrounds, you can use image segmentation. Objects of interest in a scene are often segmented using techniques such as background subtraction. It' is possible to think about background subtraction as a way for separating two objects in an image. Segmentation techniques are based on Otsu, lesions, ROILS, or edges to differentiate the ROIs from the background [49–52]. After that, a fusion process is carried out to integrate the feature vectors to a final vector. Since segmentation is a must for object detection, we can divide the used architecture into two, namely, the fully convolutional networks and the Mask R-CNN [53, 54]. The fully convolutional neural network can be seen in the work where conventional FCN was employed, where U-Net was employed, and where SigNet was used [55–58]. Mask R-CNN was used in [54, 59, 60]. This is a region-based type of segmentation and the most widely used. First, the ROI (region of interest) is extracted from the fruit dataset, followed by the removal of the segmented image background. The classification of detection network is then applied to the segmentation image to achieve the detection task. Since we are not much concerned about the segmentation aspect, we summarize the preprocessing and segmentation process of fruit detection and classification diagrammatically.

4.2. Deep Learning Models Applied to Fruit Detection. In several real-life applications, fruit identification systems were implemented in store checkout, where the device might very well use scanner tags instead of manual ones. Furthermore, it can be used as aids for the blind. The identification of different fruit species is a repeated activity in supermarkets, where the cashier must identify each type of product that defines its cost. The right solution to this problem is to provide a system of fruit identification that automates price labelling and measurement. While several researchers addressed the fruit detection question, as seen in [61–64], the problem of developing a fast and reliable fruit detection system still persists as recorded in [65]. This is because the look of fruits in field settings, including color, type, scale, texture, and reflection properties, is highly variable. Deep neural networks have recently made important advances in the classification and identification of objects. There are two steps in the state-of-the-art PASCAL-VOC detection system [66]. The first phase of the pipeline implements the regional proposed method for extracting areas of interest from an image, such as selective search and edge box, and then feeds them to a deeper neural classification network [67]. Despite the pipeline's excellent recording achievement, real-time implementation cannot be realized as a result of the pipeline's high computational cost. This problem is overcome by integrating a deep convolutional network for classification with the object proposal network, generally called region proposal network (RPN) [68–70], so that the system can simultaneously predict and classify object boundaries at each location. The parameters of the two networks are shared, which result in much higher ratings and are thus optimized for robotic purposes. We will discuss in detail the different DL models applied in solving fruit detection problems as summarized in Table 1.

4.2.1. Faster Region-Based Convolutional Neural Network (Faster RB + CNN). This model was adopted by Sa et al. [71]. The purpose was to establish a neural network which would be used to harvest fruits from autonomous robots. The network model employs transfer learning using ImageNet and two input image types: RGB and NIR (near infrared). There were two ways to merge the RGB and NIR input images: early fusion and late fusion. To begin with, 3 RGB channels and 1 NIR channel are needed for the early fusion. Late fusion utilized 2 explicitly trained models, which are paired with predictions from both models and results summed. Chen et al. [40] likewise employed a faster region-based convolutional neural network for fruit detection in orchards and compared the performance against other architectures (VGG and ZFNet). There are five convolutional

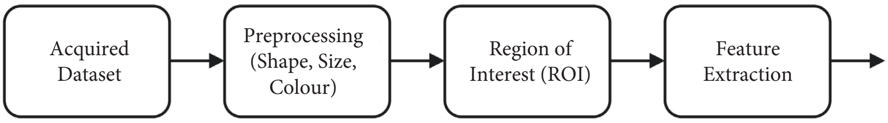


Figure 3: Preprocessing and segmentation steps for fruit detection and classification.

Table 1: Summary of the deep learning models applied to fruit detection.

Ref/year	DL model	Dataset	Dataset partition	Accuracy
[71] 2016	Faster R-CNN	TL + Field Farm	82% Train, 18% Test	0.83 F1-s
[41] 2017	Faster R-CNN	Orchard	2268 Train, 482 Test	>0.9 F1-s
[72] 2017	IN-ResNet	Personalized	24000 Train, 2400 Test	91% - 93
[41] 2017	VGG-16	Orchard	2268 Train, 482 Test	95%
[73] 2018	CNN	Kiwifruit	70% Train, 30% Test	89.29%
[74] 2019	YOLO V3	PT + WGISD	—	—
[75] 2019	DAN	Fruit 360	70% Train, 30% Test	91%
[76] 2019	Faster R-CNN + Iv2	Cherries	60% Train, 20% Val, 20% Test	85%
[77] 2019	E-Net	Fruit 360	80% Train, 20% Test	93.7%
[78] 2019	SS-CNN	Apple/Pears Orchard	—	+90%
[79] 2019	M-YOLO	PT + Mango Orchard	1300 Train, 130 Validation, 300 Test	0.97 F1-s
[80] 2019	M-Net	Mango Orchard	1300 Train, 130 Validation, 300 Test	73.6%
[81] 2019	M-RCNN + RetinaNet + FPN	Strawberry Dataset	2000 Train, 100 Test	95.78%
[82] 2019	Faster R-CNN + VGG-16	Kiwifruits	70% Train, 30%	Test -
[82] 2019	MMF MKV2	Kiwifruits	70% Train, 30% Test	±90%
[83] 2019	MVGG-16	Guava	80% Train, 20% Test	98.3%
[84] 2019	MVGG-16	Date Fruit	80% Train, 20% Test	98.59%
[83] 2019	MGNet	Guava	80% Train, 20% Test	94.8%
[84] 2019	AlexNet	Date Fruit	80% Train, 20% Test	99.01%, 97.01%
[85] 2019	ResNet	Strawberry	80% Train, 20% Test	94%
[86] 2019	MR-CNN + RNet-101	Orange	60% Train, 20% Validation, 20% Test	97.53%
[87] 2020	YOLO V3	PT + WGISD	Pretrained + 300 Train, 60	Test 97.3%
[88] 2020	YOLO V2	Mango + WGISD	300 Train, 60 Test	96.1%
[87] 2020	YOLO V2	Mango + WGISD	300 Train, 60 Test	95.6%
[89] 2020	YOLO V4	Banana Orchard	835 Train, 209 Validation, 120 Test	99.29%
[90] 2020	IM-R-CNN	Apple	368 Train, 120 Test	97.31%PR
[88] 2020	M-YOLOV3	Mango Orchard	1300 Train, 130 Validation, 300 Test	94% F1-s
[91] 2020	YOLO V4 + U-Net	Litchi Fruits	—	100%
[17] 2020	RetinaNet-FPN V4	Strawberry	80% Train, 20% Test	—

TL = transfer learning, F1-s: F1-score, PR: precision rate, PT: pretrained, WGISD: Wine Grape Instance Segmentation Dataset, and MKV2: Microsoft Kinect V2.

layers in the ZF network and 13 deeper layers in the VGG-16 network. There is a set of convolutional layers from which a 3-channel input image is propagated, from the region of interest. Each box is spread across fully connected layers that offer back its class likelihood and retract a finer boundary box around each object. The ground truth of the input image is used during training in the RPN and the R-CNN layers. A class specific detection level is added to the performance during testing and a non-maximum detection threshold is applied to avoid overlap.

4.2.2. Modified Inception-ResNet (MI-ResNet). This model is based on deep simulated learning. Rahnemoonfar and Sheppard [72] used this model in counting of fruits. The objective was to create a simulated deep CNN for yield estimation. The model was employed to solve various problems facing CV algorithms for counting fruit for estimating production, such as the varying degree of fruit overlap, lighting and foliage occlusion, shadow fruit and the variance in size, and so on. The overhead for object detection and localization is minimized by this model. The models utilize synthetic data for training and are evaluated on real data and consist of, in addition to the updated Inception ResNet-A, several convolution and pooling layers. Concatenation was done to the divided three parallel layers of the updated Module Inception-ResNet-A. Following the modified Inception-ResNet-A layer, a reduction modified inception module was employed to minimize the image size simultaneously and increase the number of filters. Another set of Inception-ResNet-A layer follows immediately accompanied by 3 × 3 average pooling before the final fully connected layer was added.

4.2.3. VGG-16. VGG-16 which is a 13-layer convolution model was compared by Bargoti and Underwood [41] against a faster region-based convolutional neural network for deep fruit detection in orchards. The performance from the convolution layers is a map with high dimensions, sampled by 16 due to the steps of the layers in the pooling. In the local function map areas, a layer of box regression and a box classification layer are distributed to two entirely related siblings.

4.2.4. You Only Look Once (YOLO V3). This model has been used by Katarzyna and Pawel [74] to classify fruit variety into uncertainty retail conditions. To detect the fruit of the entire image, a vision-based deep CNN was utilized, and then a CNN approach was used for classifying the fruit of images with a single entity (apple). In order to create apple region of interests (RoIs) from the original images, the YOLO architecture V3 was used [92–94]. The key difference between this architecture and the previous version is that it allows detection on three different scales and is thus ideal for smaller objects. Object characteristics like the pyramid network are derived from these scales. In the first step, YOLO divides the image input into a grid of $S \times S$, where S is the magnitude. Only one entity with boundary boxes is expected for each cell. The network uses logistic regression to predict an objectless score for each border box. The score criteria weeded out bad predictions. The end result is a box whose top and lower right corners can be explained. Shi et al. [95] also used the YOLO V3 network in an attribution-based pruning method for real-time mango detection. The YOLOV3 was pruned to fit into their work. On the other hand, Liang et al. [91] employed the YOLO V3 network as a visual identification technique for litchi fruits and fruiting stems that ripen at night. In this study, litchi fruits are identified in the night time on the basis of YOLO V3, and in accordance with the bounding boxes, the region of interest (RoI) was determined in the fruiting trunks. Finally, the fruit stalk is segmented one after another on the basis of U-Net for detecting litchi fruit and fruit stalks at night. Deep neural networks and three-dimensional association were utilized by Santos et al. [87] to identify, segment, and track grapes using YOLO V3. It was determined whether or not the YOLO V3 could correctly identify fruits from segments of the Embrapa WGISD.

4.2.5. Deep Alex Networks (DANs). This model was proposed by Divya Shree et al. [75] for detection of fruits from images and display of its nutritional value. AlexNet is, as we know, is a CNN image recognition network. The job is to identify the given entry in one of the classes. The network of Alex has 8 layers. We have the convolutional layers as the first five layers and the fully connected layers at the last three layers including activation layer (ReLU) and maximum pooling layers. These input images are presumed to be of 227 dimensions to 3 with a filter number of 96 of dimensions $11 \times 11 \times 3$ with 4 as the stride length. Convolutional layer extracts the functionality, and fully connected layers are natural neural networks.

4.2.6. Faster R-CNN Meta-Architecture + Inception V2 (Faster R-CNN + IV2). Villacrés and Auat Cheein [76] went ahead to employ Faster R-CNN in a more unique and modified way for the purpose of cherry detection and characterization. The use of Faster R-CNN architecture was adopted including InceptionV2 as the feature extractor to detect the cherries that are found within the bounding box [96]. In the preprocessing stage, the input images were subdivided after which the convolutional feature map was generated by the feature extractor which was used in two modules. The first module also known as the regional proposal network (RPN) uses a sliding convolutional network over the feature map locating at each point an anchor box. After that, identical fully connected layers were used to determine the proposed regions, i.e., the bounding box coordinates and the probability of it belonging to either a class or background of the input. The second modules utilized the proposed region by cropping the feature map as well as generating the RoI. After that, the generated RoI is passed through a pooling layer to a fully connected layer for the estimation of the probability and restructuring the coordinates of the bounding boxes.

4.2.7. You Only Look Once (YOLO V2). YOLO V2 was used by Xiong et al. [88] for the visual detection of green mangos in orchards with the aid of an unmanned aerial vehicle (UAV). The YOLO V2 consists of 19 convolutional layers and five maximum pooling layers and achieves greater detection accuracy while preserving the detection pace of the YOLO. We can additionally see the use of YOLO V2 in the work of Santos et al. [87] for fruit recognition system. It was used to identify, segment, and track grapes, among other things. For the assessment of fruit identification, they used the Embrapa Wine Grape Instance Segmentation Dataset (WGISD).

4.2.8. EfficientNet (E-Net). We noticed the use of the EfficientNet model in the work of Thi Phuong Chung and Van Tai [77] for fruit recognition system. It makes use of pre-trained convolution neural networks for performing image related functions in structure of a base network. EfficientNet originally performs a grid search for the base search in order to decide the relationships between the unique scaling dimensions of the network while looking at each model dimension and availability of computational resources.

4.2.9. You Look Only Once (YOLO V4). Fu et al. [89] utilized the YOLO V4 model to quickly and accurately recognize banana fruits in large backdrop orchards. The CSPDarknet53 modules were utilized as the backbone of the model. It consists of five CSP (cross, stage, and partial) connections with eleven convolutional layers with batch normization and Mish as activation function (11 CBM). In simple terms, the CBM is a normal convolution that employs batch normalization and the use of activation function of Mish. It is worthy to note that CBM module and CBL models are same except that both use a different activation function. The CBM uses the Mish, whereas CBL uses the Leaky ReLU activation function.

4.2.10. Single-Shot CNN (SS-CNN). A single-shot CNN was employed by Bresilla et al. [78] for a real-time fruit detection within a tree. A modification of the YOLO V2 model was employed here. The modification was seen at the grid search method. They changed the standard model input grid by removing some layers of the model and came up with a new model that utilized only 11 layers, dual grid size, and additional two new blocks.

4.2.11. Improved Mask R-CNN (IM-R-CNN). For the task of detection and segmentation of overlapped fruits with apple harvesting robot, Jia et al. [90] introduced the use of optimized Mask R-CNN. They replaced the original backbone network structure of the Mask R-CNN with a combination of the ResNet and DenseNet network structure for the feature extraction. Reason for the replacement was to increase feature reusability and transitivity by the usage of less parameters, and they still obtained an excellent performance. The resulting feature map generated by this backbone network was fed into the RPN as input to generate the region proposal with the idea that the input is for each feature map accordingly. Lastly, the full convolution network generates the mask which shows the region where the apple is found. Mask R-CNN was also employed by Santos et al. [87] for grape detection, segmentation, and tracking using the Embrapa Wine Grape Instance Segmentation Dataset (WGISD). Yu et al. [81] also modified the Mask R-CNN for the task of fruit detection for strawberry harvesting robot in non-structural environment. The feature extractor was built on top of the ResNet-50 backbone architecture and the feature pyramid network (FPN). Sequel to the modelling of the region proposal for each feature map, they trained the region proposal network end to end after which the generation of the ripe fruit mask images was done, and a visual localization method was carried out for the strawberry picking points.

4.2.12. Mango YOLO (M-YOLO). A mango-based model was proposed by Jia et al. [90] for a task of real-time fruit detection and orchard fruit load estimation. This is the integration of the YOLO V3 and YOLO V2 (tiny) networks which form the benchmarking of "Mango Yolo." The architecture offered an acceptable processing speed for real-time operations with just 33 Mango Yolo layers compared to 107 YOLO V3 layers.

4.2.13. Mango Net (M-Net). Based on semantic segmentation, Kestur et al. [80] developed this CNN model for the task of mango detection and counting via an open orchard. The activation layer (ReLU) follows all convolution operations with the exception of a penultimate convolution layer which is followed by activation layer (sigmoid). The output is then sampled for the last convolution layer. In all convolutional layers, a 3×3 stride of 1 filter is used and 1-pixel padding is presented. Filter count and convolution layer thickness are both fixed for a particular block in a multi-filter convolution process. Each block except block 4 is subject to maximum pooling. The kernel of the maximum pooling is 2×2 with a stride of 2. In blocks 1, 2, 3, and 4, the number of filters is 64, 128, 256, and 1, respectively. To obtain the final pixel classification result, block 4 output is upsampled to the size of the image input. All convolution layers are activated by ReLU function with the exception of the penultimate layer. A semantic segmented image is formed in the last layer. A pixel-sensitive function map is the semantic segmented image and was used for mango detection and counting.

4.2.14. RetinaNet + Feature Pyramid Network (RetinaNet-FPN). Kirk et al. [97] proposed this model for a rapid and robust outdoor fruit detection system combining bio-inspired features with one-stage DL networks. The RGB data and the CIELab data are depth-specifically packed to form a 6-dimensional tensor input to the grid, where the dimension of input D is enhanced by a stride of 2 at the convolutional layer. Then, from the four blocks of the 3×3 convolution layer and activation layer (ReLU) twice repeated by increasing the number of the input channel D comes a four-feature map generated by the ResNet-18 feature extractor. The latter three maps are then used for 5 multi-scale map generation in the pyramid network.

4.2.15. Faster R-CNN Network + VGG-16 (Faster R-CNN + VGG-16). Liu et al. [82] employed this method in the classification of kiwifruit detection. They pretrained the original convolutional layers of the VGG-16 network with ImageNet dataset and fine-tuned them with the RGB and NIR images of kiwifruit training dataset denoted accordingly as RGB-only and NIR-only.

4.2.16. Multi-Modality Fusion (MMF). Liu et al. [82] employed this method in the classification of kiwifruit detection. To receive and fuse the aligned RGB and NIR images, two modified networks have been created. One was VGG-16, which was simultaneously sponsored by RGB and NIR images. The other included two VGG-16 pretrained networks, each of which was assisted by RGB and NIR image and then connected to the feature map. They were, respectively, referred to as image-fusion and feature-fusion modes. The image-fusion mode changed the structure of the VGG-16 network's input layer from three to six channels (RGB image as well as the NIR image has 3 channels). The VGG-16 network was changed and adapted to simultaneously receive RGB and NIR data. The RGB and NIR images were input individually into two VGG-16 networks in the feature-fusion mode before being merged on the feature map. To accomplish this, the feature maps of the RGB and NIR images are then combined with the concatenation layer. The two VGG-16 networks were initialized with the RGB-only and NIR-only parameters, respectively, and fine-tuned as the original VGG-16 network.

4.2.17. CNN. This model was adopted for the purpose of image recognition of multi-cluster kiwifruit in field by Fu et al. [98]. The input layer is followed by three convolution layers, namely, Conv1, Conv3, and Conv5, before a subsampling layer and the finally output layer. By using a batch normalization (BN) process, the CNN architecture was optimized. Added BN layers were at the 1, 3, and 5 convolutional layers of the initial LeNet. The scale of all kernels was 5 × 5; the samples of all layers were 2 × 2. 6, 16, and 120 represent the Conv1, Conv3, and Conv5 feature maps, respectively, with the activation function ReLU.

4.2.18. Modified VGG-16 (MVGG-16). Lin et al. [83] employed the modification of VGG-16 architecture for the purpose of guava detection and pose estimation. The modification is seen at removal of the dense prediction map of VGG-16 using the state-of-the-art FCN model [53]. In simple terms, the FCN model was used to rewrite the fully connected layers of VGG-16 into the fully convolutional layers. Yuesheng et al. [99] employed this architecture alongside GoogleNet. We also saw the use of VGG-16 in a modified way for harvesting date fruits in [84]. The model uses a smooth architecture that uses filter dimension of 3 × 3 with a stride of 1 for the convolutional layers and a dimension of 2 × 2 with stride of 2 in all layers. The 13 convolutional layers which made up the architecture with 3 fully connected layers are grouped into 5 blocks while the maximum pooling layers are used to link the adjoining blocks. The number of filters in the convolutional layer increases by a multiplication of 2 after each maximum pooling. ReLU is used as the activation function in all the layers, and the fully connected layer architecture of AlexNet was employed.

4.2.19. Modified GoogleNet (MGNet). GoogleNet modification was seen in the work of Lin et al. [83] for the purpose of guava detection and pose estimation alongside the modification of VGG-16 architecture. They swapped the FCN model into the fully connected layers of GoogleNet into the fully convolutional layers.

4.2.20. AlexNet (ANet). AlexNet architecture was employed for date fruit classification by Altaheri et al. [84] for robotic harvesting. The network consists of 5 convolutional layers of which the first two convolutional layers are followed by a local response nomination and a maximum pooling layer while the fifth layer of convolution is followed only by a maximum pooling layer and three fully connected layers. The first layer is of size 11 × 11 with a stride of 4 and 96 filters, and the second layer is of size 3 × 3 with a stride of 1 and 256 kernel filters, whereas the third to fifth convolutional layers are of size 384 × 384 with kernel size 3 × 3 with 256 filters. After the first convolutional layer, the stride is set to 1 for all other convolutional layers and a ReLU (rectified linear unit) is set as the activation function.

4.2.21. ResNet (RNet). Ge et al. [85] utilized the ResNet architecture for strawberry detection. Its uses the basic CNN to extract features from the input images. These feature maps pass through RPN (region proposal network) to generate the RoI (region of interest) potential bounding boxes and then the RoI is aligned.

4.2.22. Mask R-CNN + ResNet-101(M-RCNN + ResNet). Mask R-CNN which is an extended version of Faster R-CNN comprises two stages, namely, the region proposal network (RPN) and the feature extraction stage. Chen et al. [40] employed the ResNet-101 in an augmented way with the Mask R-CNN to generate a binary object mask for each region of interest.

4.3. Deep Learning Models Applied to Fruit Classification. The system of image classification plays a very important role in many fields. Recognition of images and DL are rising so fast and they help more and more fields. Classification of fruit is a complicated problem because of all the variations. When it comes to classification, there are usually two issues: (i) the grading by fruit of different types (for example, differentiating between apples and oranges) and (ii) the grading of same fruit varieties (e.g., to differentiate among apple varieties such as red delicious, honey crisp, golden delicious, gala, and so on). But even with the first type of issue, it remains difficult to achieve precise classification because of form, color, maturity variations, and so on. Another concern is the precision of the classification DL models. Table 2 summarizes the different DL models applied to fruit classification. We discussed different researchers' DL models used in fruit classification below.

4.3.1. Convolutional Autoencoder-Attention-Based DenseNet (CAE-ADN). This model was proposed by Xue et al. [109] for hybrid deep learning-based fruit classification. This model pretrained images with a convolution autoencoder and extracted image features using an attention-based DenseNet. In the first part of the system, the greedy layerwise CAE is pretrained with an unsupervised method with an image set. Initializing a variety of weights and ADN biases using a CAE structure, the supervised ADN with the ground truth is applied to the second part of the system. The last section of the system includes a forecast of the fruit group.

4.3.2. Simplified CNN Architecture (9-Layer CNN). Katarzyna and Pawel [74] proposed the use of a 9-layer CNN termed as simplified CNN for the task of fruit variety classification in uncertainty conditions of retail sales. This is an architecture of a deep neural network with 9 layers. The first layer is a 150 × 150 × 3-pixel input layer, which was resized to an image of 320 × 258 × 3 pixels. The next four stages comprise two trails with maximum pooling layers that have no padding using the receptive field (convolutional kernel) of 3 × 3. The layers have feature maps of 32 and 64, respectively. Non-linear ReLU (rectifier linear unit) is used as the activation layer in the convolution layers. The maximum pooling strategy was used in the third and fifth layers to minimize dimensionality and simultaneously catch the features in the subregions binned [116]. In the final step, the fully connected layers were used to identify the fruit on the preceding dropout layer. 64 ReLU fully connected neurons were provided in the 8th layer. The final layer of the classifier consists of 6 SoftMax neurons, corresponding to the 6 different types of apples. The

Table 2: Summary of the deep learning models applied to fruit classification.

Ref/year	DL model	Dataset	Dataset partition	Accuracy
[100] 2015	CNN	Personalized Dataset, UEC-FOOD100	—	80.8% SF, 60.9% MF
[101] 2017	Modified VGG	Personalized Dataset	80% Train, 20% Validation	95.6%
[102] 2017	MCNN	ImageNet	—	74% WDA 90% DA
[14] 2017	13-layer CNN	Veg Fruit Dataset	63000 Train, 1800 Test	94.94%
[103] 2018	PCNN + GAP	Fruit 360	80% Train, 20% Test	98.88%
[103] 2018	CNN FC-L	Fruit 360	80% Train, 20% Test	97.41%
[103] 2018	CNN FC-L	Dropout Fruit 360	80% Train, 20% Test	97.87%
[104] 2018	MAlexNet	Personalized ImageNet	80% Train, 20% Test	92.1%
[105] 2018	DCNN	Personalized	30082 Train, 7520 Validation, 6804 Test	90%
[74] 2018	6-layer CNN	Personalized	900 Train, 900 Test	91.44%
[106] 2018	8-layer CNN	VegFru	50% Train, 50% Validation, 50% Test	96.67%
[74] 2019	9-layer CNN	COCO apple class	70% Train, 15% Validation, 15% Test	99.78%
[107] 2019	LW models	TL, Fruit 360	80% Train, 20% Test	98.7%
[108] 2019	DCNN models	Fruit 360	80% Train, 20% Test	99.6%
[24] 2019	VGG-16 + GAP	SPD, Personalized	85% Train, 5% Validation, 15% Test	99.49%
[24] 2019	LA	SPD, Personalized	85% Train, 5% Validation, 15% Test	99.75%, 96.75
[39] 2019	M-GNet	Hyperspectral Images	2000 Train, 700 Validation, 125 Test	88.15% PRGB 85.93% LC 92.23% CK
[109] 2020	CAE-AND	Fruit 26, Fruit 15	85,260 Train, 38,952 Test	95.86%, 93.78%
[110] 2020	InterFruit	InterFruit	70% Train, 30% Test	92.74%
[111] 2020	VGGNet	—	—	—
[112] 2020	CNN SL	Orange Fruit	60% Train, 20% Validation, 20% Test	—
[109] 2020	ResNet-500	Fruit 26, Fruit 15	80% Train, 20% Test	93.59%, 91.44%
[109] 2020	DenseNet-169	Fruit 26 Fruit 15	80% Train, 20% Test	93.87%, 91.46%
[113] 2020	Deep CNN	Cheery	—	99.4%
[114] 2020	MobileNetv2	A O B	—	95% PB 93% WPB
[115] 2020	EDLS Fruits	Fresh, Fruit-360, Rotten for Classification	—	—

DL: deep learning, TL: transfer learning, SPD: Supermarket Produce Dataset, PT = pretrained dataset, PB: plastic bags, WPB: without plastic bags, A O B: apples, oranges, and bananas, WDA: without data augmentation, DA = data augmentation, SF: single food, MF: multi-food, PRGB: with pseudo-RGB images, LC: with linear combinations, and CK: with convolutional kernels.

Adam algorithm with cross-entropy was used as a loss function to train the CNN model [92].

4.3.3. Pure Convolutional Neural Network (PCNN + GAP). PCNN is a classification framework employed for the task of fruit image classification and was proposed by Nuske et al. [61]. The architecture is simplified with a minimum number of parameters, which are specified as input, convolutional layers, strides, ReLU, GAP layer, and SoftMax. The PCNN has 7 layers, and some of them are progressively followed with stride. In addition, the authors employed the recently established global average pooling (GAP) layer which has been shown to be highly successful to minimize overfitting and take the average of entire characteristic maps. The convolutional layer was used to extract the feature maps from input images by linear convolutional filter accompanied by non-linear functions (ReLU, sigmoid, tanh, and so on). Afterwards, extracted feature maps are sent into another layer which is followed by a stride. Downsampling is done in PCNN by using a convolutional layer accompanied with stride.

4.3.4. Convolutional Neural Network + Fully Connected Layer (CNN + FC Layer). This model was used to make comparison with the PCNN + GAP model. It is worth to note that it is deployed for the task of fruit image classification. It is a classical CNN which contains six convolutional layers and three maximum pooling layers for dimension reduction along with stride 2 and a fully connected layer.

4.3.5. Convolutional Neural Network + Fully Connected Layer + Dropout (CNN + FC Layer + Dropout). This is a fruit image classification network that added dropout layers to a CNN and fully connected network to avoid overfitting by Kausar et al. [103] and was used to make comparison with pure CNN. It is a classical CNN which contains six convolutional layers and three maximum pooling layers for dimension reduction along with stride 2 and a fully connected layer.

4.3.6. InterFruit. This is a DL network proposed by Liu [110] for classifying fruit from images. InterFruit is a stack architecture integrating AlexNet component, ResNet component, Inception component, and 3 fully connected layers with no need for extracting color and texture features. The last fully connected layer, in particular, played a role in measuring and generating results of various fruits (classifier). The Adam optimizer was used to eliminate errors and cross-entropy loss was used as a cost function.

4.3.7. Lightweight Models (LW Models). Lightweight model proposed here is the integration of MobileNet V2 and ShuffleNet V2 [117, 118]. This model was proposed by Bac et al. [63] and is composed of two main parts: a feature extractor (MobileNet V2 and ShuffleNet V2) for its speed and accuracy and the output layer. A global average pooling follows immediately after the feature extractor. The output is normalized using the SoftMax function [16]. For network optimization, the Adam optimizer was more correctly used to adjust the stochastic gradient descent [92]. AMSGrad is used to boost the optimizer [119].

4.3.8. Deep Convolutional Neural Network Model (DCNN). Alzubaidi et al. [108] used this model for multi-class classification of fruits. In the beginning, the model begins with two conventional convolutional layers of 3 × 3, 5 × 5, to decrease input size. In order to accelerate training processes and prevent gradient problems, each convolutional layer is accompanied by batch normalization and rectified linear unit layers. Four blocks of parallel convolutional layers were used to extract the characteristics from the conventional convolutional layers. There are four convolutional layers operating in parallel on the first block, followed by the output of four convolutional layers and conventional convolutional layers in the first concatenation layer using residual connections. Overlaying problems were avoided by using three fully connected layers having two dropout layers in-between. In the end, 118 fruit groups were classified using SoftMax.

4.3.9. CNN-Based Architecture (CNN-BA). We saw the use of this CNN model in classifying recognized fruits and validating the hidden layers' accuracies in the work. Two convolutional layers after the input layer are followed by max pooling layer and two fully connected layers, respectively. The input layer receives 30,000 neurons as the input data, followed by the first convolutional layer with 64 filters and a kernel size of 3 × 3 pixels and the activation function of rectified linear unit (ReLU). The second convolutional layer receives the same number of filters and kernel size as that of the first convolutional layer. A LeCun uniform kernel initializer is used for initialization of the weights along with the 2 convolutional layers. ReLU is used to increase the efficiency of all convolutional layers and fully connected layers as an activation function.

4.3.10. VGGNet: A Variant of Convolutional Neural Network. This model intends to train a VGGNet to recognize more than one label in a single instance of image sample. The input layer accepts a fixed 96 × 96 RGB image size. Then, a stack of convolution layers will process the image. The convolutional layers have a 3 × 3 size each of multiple filters, whereas the last convolutional layer has a 1 × 1 size filter which is the linear transformation of the input data accompanied by a non-linear convolutional layer. The maximum pooling layers follow immediately after the convolutional layers which have a set of 2 × 2 filters with stride of 2, whereas the stride is set to 1 in the convolution layer. The space padding is set so that after the convolution step, the space resolution remains. A sigmoid activation layer is used for the training. 3 fully connected layers are set after the layers of the normal VGGNet architecture.

4.3.11. CNN Architecture with Several Different Layers (CNN SL). This is a CNN architecture with several layers, including four convolution layers, two maximum pooling layers with kernel size of 3 × 3, dropout layer, flatten layer, and dense layer. Asriny et al. [112] employed this model for the task of orange fruit image classification. The hidden layer contains 256 nodes, and they compared the use of both tanh and ReLU activation functions to ascertain the best model performance.

4.3.12. Residual Networks (ResNet-500). ResNet was first proposed by He et al. [120]. This model was used as a baseline in comparison to CAE-AND. The model has the same parameter number as that of CAE-AND.

4.3.13. Dense Convolutional Network (DenseNet-169). DenseNet was first employed in the work of Li et al. [121]. Many researchers have adopted this model for different tasks, where it was used as a baseline in comparison to CAE-AND for the task of fruit classification. A 16-output-channel convolution was performed first on the input images before entering the first dense block.

A one-pixel zero padding was done on every side of the inputs of the convolutional layers with 3 × 3 kernel size for the benefit of size fitting the feature map. The transition layer used in-between two contiguous dense blocks is a 1 × 1 convolution followed by an average pooling of size 2 × 2. After the final dense block's global average pooling is completed, a SoftMax classifier is used. The three dense blocks' feature map is of size 32 × 32, 16 × 16, and 8 × 8, respectively. All the layer-to-layer connection was done in a feedforward style.

4.3.14. Deep CNN Based on Hybrid Pooling Approach (Deep CNN). Momeny et al. [113] utilized this model for an accurate classification of cherry fruit. This model comprises 4 convolutional layers which use different kernels for the input image convolving and 3 hybrid maximum pooling layers for the network parameter reduction. It accepts RBG images of fixed size of 64 × 64 as input. The maximum pool layers come immediately after the convolutional layers, whereas ReLU was used as the activation function. Batch normalization was implemented between the convolution layer and the ReLU layer to speed up the training as well as reduce the sensitivity of the network initialization.

4.3.15. VGG-16-Based Architecture. VGG-16 is a deep CNN model applied in the industrial sector by Hossain et al. [24] for automatic classification of fruits. It consists of 5 convolutional layers of size 3 × 3 with same kernel size within each layer and a maximum pooling layer. The kernel increases from 64 in the first layer to 512 in the last layer. The number of learnable layers in the network is 16 in total.

4.3.16. Light Architecture (LA). This was seen in the work for automatic fruit classification using DL for industrial applications. This model involves three steps: preprocessing of the data, data feature extraction, and classification. The images were cropped in the preprocessing step to create an image with the same height and width, which are both equal to the smallest in the original image. All images were then resized to the regular size of 64 × 64 × 3, which is the input form of the first layer of the model. The feature transformation step consists of 3 convolutional blocks of kernel size 3 × 3 followed by max pooling layers of kernel size 2 × 2 before a fully connected layer. The transformation of the features was carried out through repeated convolution and max pooling operations. The last step is the classification step where the 2D feature maps from the previous layers are flattened, i.e., transforming the 2D feature into a 1D feature vector, and then fed into the output layer which is fully connected layer with same number of classes for classification. Activation function used is SoftMax, and dropout layers were implemented after each pooling layer.

4.3.17. Google Inception V3 Model + Simple CNN (GIM-CNN). This setting was used for the classification of fruits and vegetables with its nutrients. The Google Inception V3 model was used to process the images for the CNN. GAP (global average pooling) was applied to the dataset (reshaped to size 299 × 299 × 3) for averaging the features of the total images input. To avoid overfitting, dropout of size 0.5 was applied with stochastic gradient descent to achieve a better accuracy. They implemented and trained it for 32 epochs with 3 defined callbacks to record the progress in a log file.

4.3.18. 13-Layer Deep Convolutional Neural Network. This model was employed for the task of classification of fruit category. The preprocessed fruit image is inputted directly by the input layer. The convolutional layers performs a 2D convolution for 3D input and filter. Convolution is followed by the activation function ReLU which accepts the feature map neurons, whereas the output of the activation function is replaced by the maximum pooling layer. The SoftMax layer utilized the SoftMax function, whereas the fully connected layer multiplies the input weight and then adds a bias vector.

4.3.19. Modified DCNN. This is a two-way DL neural network employed in the work for automatic fruit recognition for future class of fruit. The constructed network consists of 150 × 150 × 3 RGB as input layer, four hidden layers, i.e., three convolutional pooling layers with kernel size of 3 × 3 and strid of 1 pixel for feature map extraction followed by a max pooling layer having size of 2 × 2. The hidden layers are followed by a fully connected layer with 64 ReLU. Next to the fully connected layer, the network has an output layer having 15 SoftMax neurons, each for one fruit class. The activation function used in this model is ReLU. In summary, the first phase of this network is the convolutional neural network with maximum pooling and the second phase is the fully connected layer. Hussain et al. [105] went further to employ this architecture for the purpose of automatic fruit recognition for commercial source trace system with changes in the network parameters.

4.3.20. MobileNetV2. To speed up the checkout process, Rojas-Aranda et al. [114] utilized an image classification lightweight CNN-based model to classify fruits in retail stores.

4.3.21. Modified GoogleNet (M-GNet). This is a modified version of the GoogleNet architecture. Steinbrener et al. [39] employed this architecture for hyperspectral fruit and vegetable classification. The main part of the architecture is the concatenation of the information from the different parts of the spatial correlation by the inception module. The architecture having a 9-inception module also consists of simple convolutional and fully connected layers. The modification of this architecture from the original GoogleNet comes in the kernel size and the addition of two auxiliary classifiers situated within the network at lower layers to increase the learning rate of the layers. The kernel size used here is of 1 × 1, 3 × 3, and 5 × 5 size concatenated to extract image features.

4.3.22. Modified CNN (MCNN). CNN is a widely used architecture for fruit classification. Nordin et al. [38] used a 4-layer architecture with the exception of the final layer and input layer which is not regarded as a layer. The 4 layers are made up of 3 convolutional-pooling layers and 1 fully connected layer. The first convolutional layer has a kernel size of 7 × 7 with a stride of 1 followed by maximum pooling of size 2 × 2. The second and third convolutional layers use a kernel size of 5 × 5 and 3 × 3 each. The first three layers use the ReLU activation function while the last layers have 10 SoftMax neurons corresponding to the 10 categories of fruit. The network was trained with the stochastic gradient descent with a cross-entropy cost function. They applied dropout to the third and the fully connected layer of size 0.25 and 0.5, respectively.

4.3.23. Modified AlexNet (MAlexNet). Zhu et al. [104] employed the use of AlexNet in a modified way for the classification of vegetables. They utilized the ImageNet dataset. The modification comes from the replacement of the traditional sigmoid function and the tanh function with the ReLU activation function to speed up the model training.

4.3.24. Six-Layer CNN (6-Layer CNN). This is a normal convolutional neural network setup with 4 Conv2d layers and two fully connected layers employed by Katarzyna and Pawel [74] to classify fruits.

4.3.25. Eight-Layer CNN (8-Layer CNN). This a modified deep CNN with a PReLu (parametric rectified linear unit) as the activation function instead of the plain ReLU and a dropout layer positioned before each fully connected layer. For the avoidance of overfitting, Wang and Chen [106] applied data augmentation for the purpose of fruit category classification.

4.3.26. Embedded Deep Learning Support (EDLS). This is an integration of DL models for the purpose of fruit recognition and classification on embedded systems as seen in the work of Unal et al. [115]. The different arrangements were the use of Adam optimizer and batch size of 16 in two convolutional layers of size 3 × 3 with 2 filters and 4 filters, respectively, applied to the first layer and second layer. The second arrangement takes the same shape as that of the first one but with an RMSProp optimizer. Using the same configuration in the second experiment, they changed the batch size to 32 and trained it only on the Fruit 360 dataset. The 3rd arrangement is also similar to the first and second arrangements but with the optimizer changed to stochastic gradient descent, batch size of 32, and 4 filters applied to both layers. Likewise, using the same configuration for the third arrangement, the authors changed the batch size to 16 and optimizer to SGD and applied two filters at the first layer and four filters at the second layer.

4.3.27. Modified VGG. This is a model that goes deeper than the usual CNN for a high classification and recognition rate. Zeng et al. [101] employed this model for the purpose of fruit and vegetable classification system. The input layer receives 224 × 224 RGB images and subtracts the mean RGB value for each pixel. The filter is of size 3 × 3 and a stride of 1. The maximum pooling is of size 2 × 2 with stride size of 2. The two fully connected layers have 4096 channels each, and the final output layer is 26, i.e., 26 classes of fruits.

4.3.28. CNN. Zhang et al. [100] employed the CNN for the purpose of fruit image recognition. The input RGB image is of size 128 × 128, with 30 kernel size of 11 × 11 and a sparse connectivity of the kernels from the second to the fourth layer. Each of the 120 4 × 4 kernels in the fifth convolutional layer is used. The output of the final fully connected layer is input into a 100-way SoftMax function with 1000 neurons for the dataset's 100 classes of dataset labels.

5. Benchmark Datasets

Sampled images consisting of real-world information are referred to as datasets, and the term data acquisition is the method of digital collection of such images. For obtaining a good classifier, a high-quality dataset is important. The most difficult detection task is the absence of sufficient labelled samples. During the course of our research, we found that most researchers, especially for object detection, deal with real-time identification of fruits mostly in orchards. Each researcher utilized his/her own dataset. We will briefly discuss some of the datasets deployed by researchers for the classification of fruits. We have used various datasets but will focus more on the dataset that was made available by the authors online. Table 3 summarizes the datasets recorded in our reviewed papers that were not made available for other authors to use in their work. We will briefly discuss the dataset made available by the authors within our reviewed publications.

5.1. Orchard Data. The orchard data comprise three types of fruit: mangos, almonds, and apples. They were acquired in an orchard during the day time in Victoria and Queensland, Australia. A sensor was mounted on an all-round research ground vehicle. The vehicle went through numerous lines of the orchards capturing image data from the trees. The total number of the dataset is 3232 with the training set = 2268, testing set = 482, and validation set = 482. For the number of images in each class, the mango class has 1154, 270, and 270 images for the training, testing, and validation sets while the

Table 3: Benchmark datasets.

Name	No. of classes	Total no.	Train set	Val set	Test set	Ref.
Orange Fruit	5 classes	1000	60%	20%	20%	[112]
Fruit 26	26 classes	124,212	85,260	—	20%	[109]
Fruit 15	15 classes	44,406	—	—	38,952	[109]
Field Farm	7 classes	122	100	—	22	[71]
Cherry	2 classes	14,380	—	—	—	[113]
Cherry	1 class	15,000	60%	20%	20%	[76]

almond class has 385, 100, and 100 images, respectively. The apple class contains 729, 112, and 112 images for the training, testing, and validation sets, respectively. The dataset is accessible online via http://data.acfr.usyd.edu.au/ag/treecrops/2016-multifruit.

5.2. Fruit 360. This is the most popular and widely used dataset for fruit classification by authors. It was first used by Muresan and Oltean [25]. Various researchers [75, 103, 108] also used this dataset in their work. The Fruit 360 dataset consists of 81,226 images of 120 classes of fruits which is divided into three sets: the training set which contains 60,498 image datasets, the test set which contains 20,622 image datasets, and lastly the validation dataset which contains 106 image datasets. The dataset was acquired by rotating a low (3 rpm) speed motor and recording (20 sec per each recording). Since the lighting conditions were not consistent, the backdrop of all photographs was transformed into white color because of the variety. All the images had a white background of 100/100 pixels. Sample of the Fruit 360 dataset is shown in Figure 4. It is accessible online via http://www.kaggle.com/moltean/fruits.

5.3. Supermarket Produce Dataset. This is a small public dataset that Rocha et al. [122] obtained at various times and dates back to couple of years. With a fixed resolution of 1024 to 768, all images were collected in JPG format. The image data collection includes 15 classes: Agata Potato class contains 201 images, Asterix Potato class contains 182 images, Cashew class contains 210 images, Diamond Peach class contains 211 images, Fuji Apple class contains 212 images, Granny Smith Apple class contains 155 images, Honeydew Melon class contains 145 images, Kiwi class contains 171 images, Nectarine class contains 247 images, Onion class contains 75 images, Orange class contains 103 images, Plum class contains 264 images, Tahiti Lime class contains 106 images, Watermelon class contains 192 images, and Williams Pear class contains 159 images. A total of 2,633 RGB images with color channel of 8 bits per pixel are in the set. Sample of the Supermarket Produce Dataset is shown in Figure 5. It is accessible online via http://www.vision.caltech.edu/ImageDatasets.

5.4. InterFruit. This contains 3,139 images making a total of 40 classes obtained from Baidu, JD.com, Google, and Taobao. The images were cropped to 300 × 300 pixels. 70% of each classes of fruit images were randomly set aside for training while the test set has the remaining 30%. This dataset was used by Liu [110]. Sample of the InterFruit dataset is shown in Figure 6. It is accessible via (password = 35d3) https://pan.baidu.com/s/19LywxsGuMC9laDiou03fxg.

6. Benchmarked Evaluation Indices

Various researchers employ different evaluation metrics based on the objective of the research. We describe the most widely and commonly used evaluation metrics in the field of object detection and classification: precision, recall, average precision, mean average precision, and the F1-score. Precision and recall are mathematically stated as [123–132]

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} * 100\%, \tag{16}$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} * 100\%, \tag{17}$$

where TP = true positive indicating that the predicted outcome corresponds to the actual outcome, FP = false positive indicating that the predicted outcome does not correspond to the actual outcome, and FN = false negative indicating that the predicted outcome does not correspond to the unrecognized outcome. The average precision is calculated mathematically as

$$P_{\text{Average}} = \sum_{j=1}^{N\,(\text{class})} \text{precision}\,(j) * \text{recall}\,(j) * 100\%, \tag{18}$$

where N (class) represents the number of all the classes in the dataset, and thus mAP is calculated mathematically as

$$mAP = \frac{P_{\text{Average}}}{N\,(\text{class})}. \tag{19}$$

Note. The higher the mAP value, the better the recognition accuracy of the framework and vice versa. The F1-score deals with the accuracy and recall value of the model. It is used to calculate the speed in the recognition rate. The more the F1-score value, the greater the speed in the recognition rate of the framework, and it is calculated mathematically as

$$\text{F1} - \text{score} = \frac{2\text{precision} * \text{recall}}{\text{precision} + \text{recall}} * 100\%. \tag{20}$$

7. Discussion

We present a comprehensive survey on the application of DL models in fruit detection and classification. We analyzed the

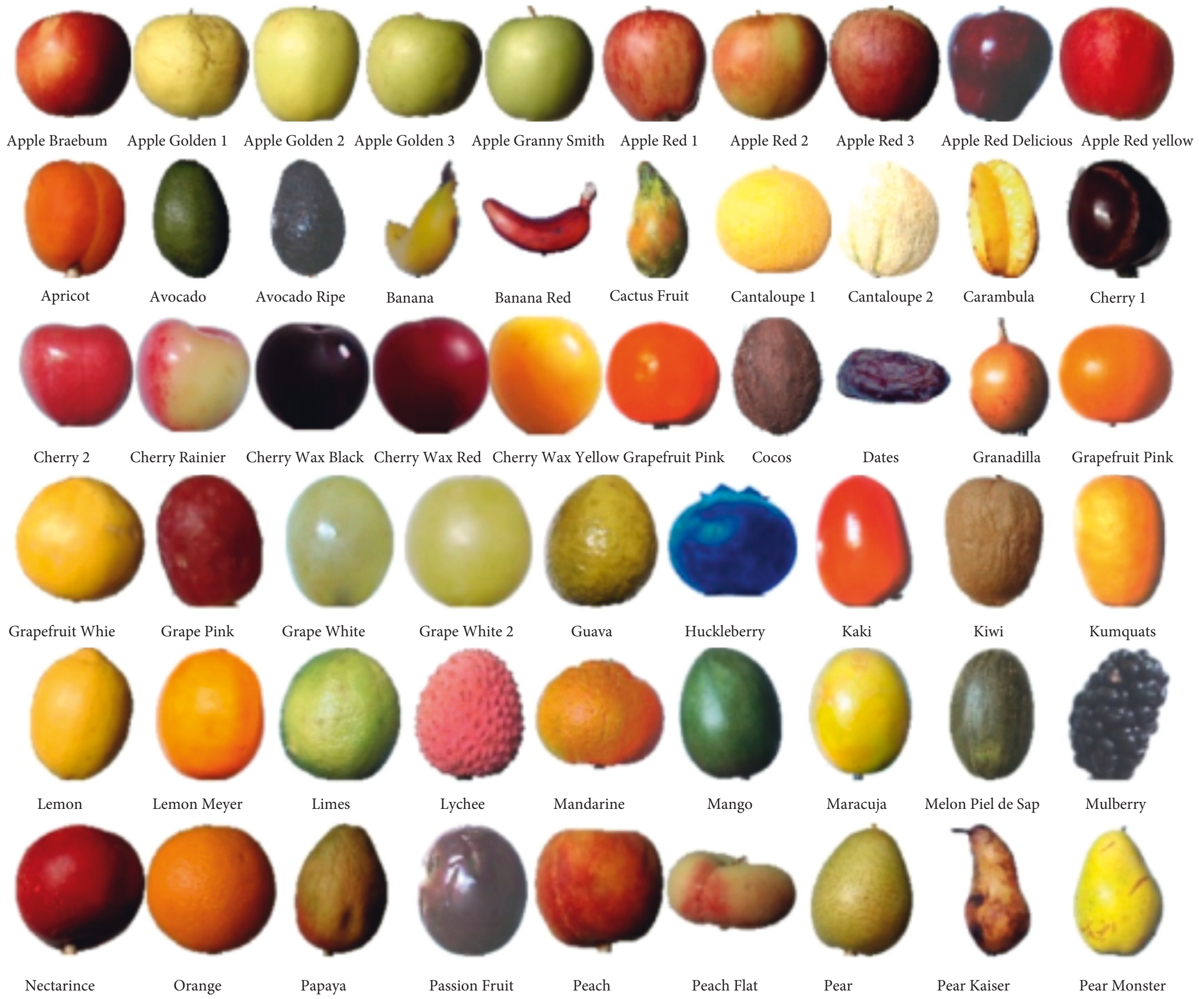
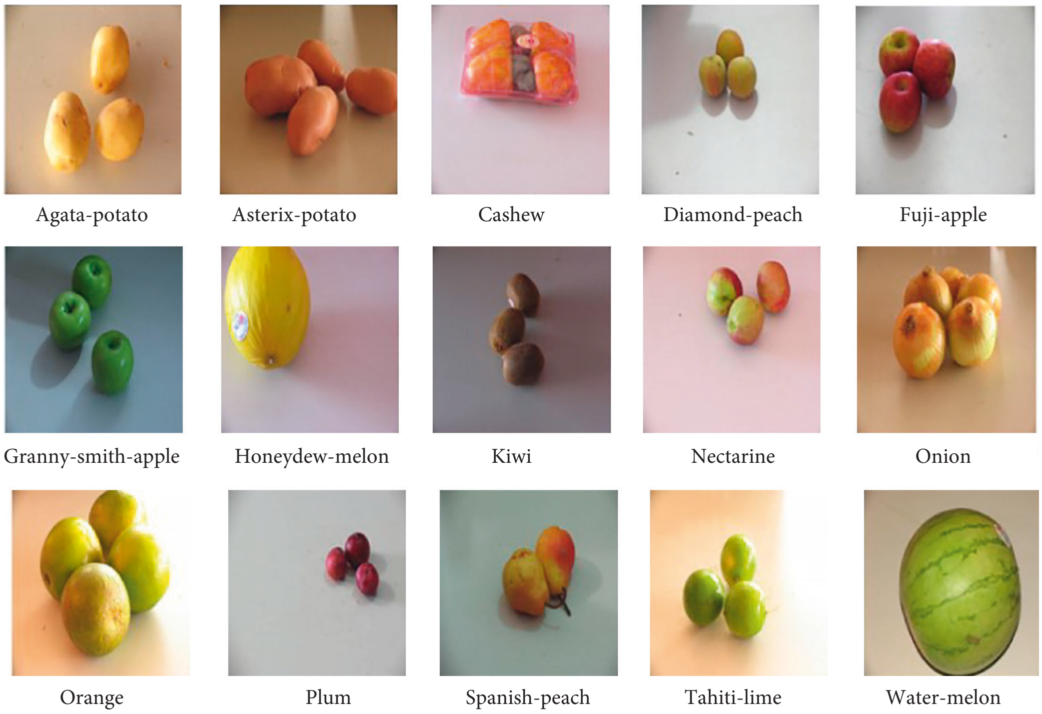


Figure 4: Sample of the Fruit 360 dataset.

different models employed, the identified datasets, and the up-to-date researchers' contribution in this aspect. We will discuss the key challenges facing fruit detection and classification and highlight our observation from the different works surveyed. First and foremost, we will discuss our finding regarding the various DL models employed by researchers on both fruit detection and classification tasks. We will partition the various architectures into two main types, namely, "From-Scratch-Training" and "Pretrained" architectures. From-Scratch-Training signifies that the authors built the architectures from scratch for a specific task while "Pretrained" signifies that authors used a well-known DL model such as YOLO, Faster CNN, Alex Net, among others. In addition to the "Pretrained" architectures, we can still break it down into two subdivisions, namely, "transfer learning" and "modified pretrained." "Transfer learning" means borrowing another architecture to fit into a new task especially when one has minimum dataset available for a specific task, whereas "modified pretrained" suggests the adjustment of the pretrained architecture to adapt to the new objective of the task. It is worth to note that the adjustment made to the modified pretrained depends on the kind of work to be performed. For instance, if the kind of task to be performed is more complex and a greater number of behaviors are to be studied, then the number of layers and filters must be increased.

Taking note from Figure 7, we noticed that the use of Trained-From-Scratch is highly employed in classification task compared to the detection task, whereas the use of modified pretrained model is highly used in the detection task against classification task. It is seen from the results of the various papers that DL model (CNN) achieved results above 95% noting that DL models are the best approach towards fruit classification against the other traditional methods. Unlike fruit classification, fruit detection tasks involve segmenting fruits in the orchard which makes the tasks more complex. It is required to design an accurate DL model architecture that can efficiently perform a semantic segmentation in the open-air images. Judging from Figure 6, we can agree that transfer learning (modified pretrained models) is the best method over others.

Faster region-based convolutional neural network (Fast-RCNN) is the most widely used architecture in terms of modifying a pretrained model for detection tasks. It is made

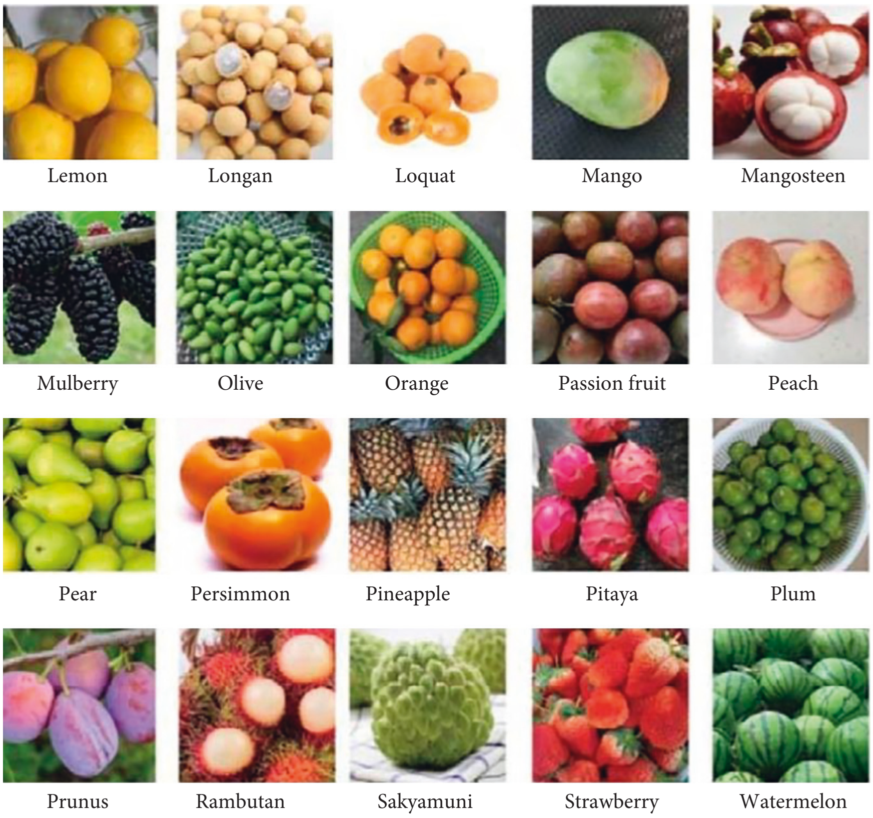


Agata-potato Asterix-potato Cashew Diamond-peach Fuji-apple

Granny-smith-apple Honeydew-melon Kiwi Nectarine Onion

Orange Plum Spanish-peach Tahiti-lime Water-melon

Figure 5: Sample of the Supermarket Produce Dataset.



Lemon Longan Loquat Mango Mangosteen

Mulberry Olive Orange Passion fruit Peach

Pear Persimmon Pineapple Pitaya Plum

Prunus Rambutan Sakyamuni Strawberry Watermelon

Figure 6: Sample of the InterFruit dataset.

up of a feature extractor model and a region proposal model. Most researchers find this model easier to employ by changing the feature extractor to their taste, thereby coming up with a modified architecture or new architecture. Masked-RCNN which follows the same principle as that of Faster-RCNN has seen much utilization for the purpose of

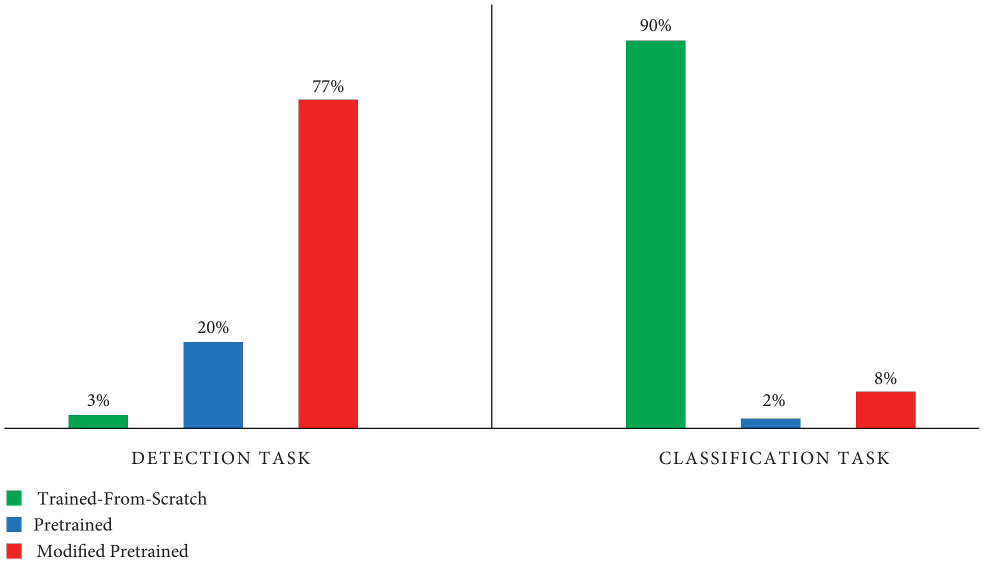


Figure 7: Graphical representation of DL model deployment in fruit detection and classification.

object detection. We noticed that most researchers employ the ResNet or the VGG as the feature extractor while deploying modified Faster-RCNN and Masked CNN. Currently in the world of object detection, the YOLO (You Only Look Once) has earned popularity due to its ability to detect tiny object of which various researchers utilized different versions for the purpose of fruit detection [133].

Moreover, sequel to the dataset employed by the researches, we figured out that dataset is the most challenging factor facing fruit detection and classification. Due to minimal availability of fruit dataset, fruit detection and classification has been a big issue in the world of AI. To train a DL classifier, large datasets are required, comprising a vast number of parameters to be tuned to control training convergence. The availability of large, multi-labelled, and well-annotated dataset repositories will eliminate the need for researchers to collect massive datasets in different real conditions and environments that would need the oversight of agricultural specialists for interpretation. The most widely used dataset for fruit classification is the "Fruit 360" dataset. It contains 120 classes, and each class has different categories of fruit and is split into three sets: training set contains 60,498 images, validation set has 106 images, and testing set has 20,622 images, and the dataset contains 81,226 images in total as illustrated in Figure 8. A major drawback of this dataset is that the images are small (100 × 100 pixels), which makes it difficult to differentiate between some fruits. The number of images in each class is not equal which will result in the accurate classification of classes with huge number of images compared to the classes with few datasets. Also, the images have no background, and thus it does not scale very well to real-world applications. Several researchers [113, 122, 134] tried to develop a more complex dataset for the same purpose of fruit classification. But the dataset does not have a sufficient number of images. We recommend to apply a data augmentation technique in issues like this. Data augmentation is a solution to solve the limitations of datasets of which generative adversarial network (GAN) is a typical example. It consists of an increase in data training, including rotations, translations, and mirrors, by carrying out a series of transformations. There are two phases in the GAN algorithm: a generative phase depicting the input image distribution and the discriminatory phase evaluating the likely output sample. Many GAN models have been developed as attempts to overcome the need for large-scale training datasets such as cycle consistent GAN, deep convolutional GAN, conditional GANs, autoregressive deep convolutional GAN, progressive growing GAN, and so on. Transfer learning (TL) is also recommended as a solution to minimal dataset issue as seen in the works of Sa et al. and Singh et al. [71, 134]. The method of TL is focused on a DL network that has been previously developed and updated to build a new mission. TL makes it possible for a CNN algorithm to obtain weights from a particular model already preworked on a broader dataset. In order to easily move the weights to the proposed new model and classes in the target dataset, the finishing layer is to be replaced with a new layer. Therefore, a pretrained network may be used to learn new trends from new data. It is also helpful if we do not have adequate knowledge to train the network. We therefore use a pretrained model for the task in a fitting dataset. The main concept is to freeze some network layers and change input and output layers normally. TL on the other hand will work for limited number of fruit categories as most of the datasets used in the TL have limited number of fruit categories.

In addition, as with fruit detection, we partition datasets in three aspects, namely, (i) images captured with deep sensors (RGB-D), which allow for a precise estimate of the distance from the fruit in robotic measurement; (ii) RGB images captured with a multi-camera system for a wider view and distance measurement; and (iii) combining NIR and LiDAR sensors with RGB cameras in multi-sensor systems. Attributes are the physical features of an object that can separate it from other objects. Fruit has many physical

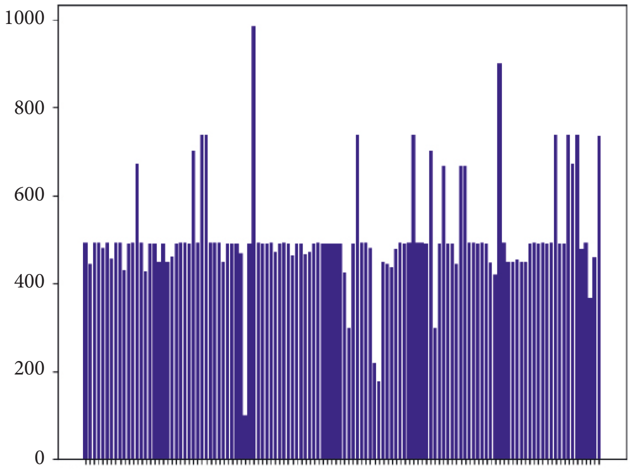


Figure 8: Illustration of classes of Fruit 360.

features such as form, texture, color, and dimension, which can be used to accurately distinguish them. Different classes of fruits are distinct. The variants of the interclass are essential improvements, e.g., color, form, and shape variations, while variations of the intraclass are usually even more subtle and impossible to discern, for instance, various types of mangos or apples have slight feature variations. The system will deal with inter and intraclass classification by an optimal range of functions. The other challenging domain is the computer-based feature representation. Significant research on the portrayal of characteristics has been published. Investigations have also shown that the successful classification of fruit or object usually cannot be deemed adequate for a single feature. We came up with the following summary of our discussion. In fruit classification studies, we found that no evaluation has been carried out with multiple types of fruit in the same image, limiting themselves to images with a single kind of fruit, either individually or grouped. Thus, the challenge is to design a CNN model for multi-detection and classification of different kinds of fruit at the same time.

(i) We present the possible solution to eliminate the scarcity of large dataset (data augmentation such as GAN and TL) especially for the purpose of fruit detection.

(ii) The use of TL in fruit detection basically supports limited number of fruits. We noted that most of the established TL models used mangos or oranges as dataset which literally means that most TL models are trained with few classes of fruits compared to the 131 classes of fruit in the Fruit 360 dataset.

(iii) There is a need for researchers to invest more on the development of fruit dataset that we be used for both object detection and classification. Deploying a fruit detection system in an agro-environment needs the detection system to be adequately trained with all classes of fruits.

(iv) The development of fruit detection models is expensive and time wasting as real-time data collection was done by the authors.

(v) Currently, Fruit 360 dataset is the largest dataset for the purpose of fruit classification. This is a good dataset in terms of numbers, but using it for training for real-time application will result in many misclassifications as the dataset is too small and does not have challenging factor (noise) in the background.

7.1. Technological Advantage of the Aforementioned Deep Learning Frameworks on Fruit Detection and Classification. Deep and complex image feature information is difficult to extract using the traditional methods as they only extract underlying features, whereas deep learning frameworks have demonstrated to be efficient in solving this issue by giving room for conducting unsupervised learning from the original input image, thus extracting multi-image feature information which comprises low, intermediate, and high-level sematic features of the input image. On the other hand, DL frameworks automatically learn features from large input image without manual manipulations as in cases for traditional frameworks, thus showing that DL frameworks have multiple layers that help in their autonomous learning tendency and feature extraction, extracting image features from the input image for detection and classification. The good thing about DL frameworks is the ability to improve the computational power to fit into the recent detection and classification needs such as the growth of training samples as well as the computational power.

7.2. Technological Disadvantage of the Aforementioned Deep Learning Frameworks on Fruit Detection and Classification. Some the frameworks which use network as feature extractor solely depend on other classifiers for the final classification results, which indicates that if the network outputs wrong features, the classification result will be very poor. The use of sliding window for classification requires the size selection to be accurate which can only get a rough position and slow down the speed of the sliding and traversal. Multi-task learning mostly has a very complex network structure, thereby requiring pixel-to-pixel labelling when trying to add the segmentation branches.

8. Future Work and Recommendation

Taking hint from the work of Zhang et al. [135] on plant leaf disease recognition, we saw that the authors constructed a three-channel CNN framework which accepts three color components, thereby reducing the known DL requirement of large dataset in training and achieving a high detection accuracy. Such idea should be adopted in designing a fruit detection and classification system, thereby overcoming the issues of the small availability of training datasets. Looking at the acquisition of dataset used, most researchers focused on the images generated in the visible range, whereas a lot of information is contained in the electromagnetic wave outside the visible range, and thus researchers should pay attention to the information such as multi-spectral, near infrared, and visible light during dataset acquisition.

Looking at the current trend in DL, in the field of fruit detection and classification, researchers widely adopt the supervised learning method, thereby requiring that the datasets for training must be large. Manually labelling of the datasets takes a lot of time and manpower, and thus researchers are recommended to venture into the unsupervised mode of learning. Deploying trained DL models on mobile platforms are unsuitable due to the large amount of memory needed during testing as well as the time consumption of the models. It is very critical for researchers to study how to reduce complexity as well as study fast executing models without affecting the result. Generally, hyperparameter selection has been a big problem in deep learning especially in new task. It affects the model result positively and negatively, thus having greater effect on the final training results.

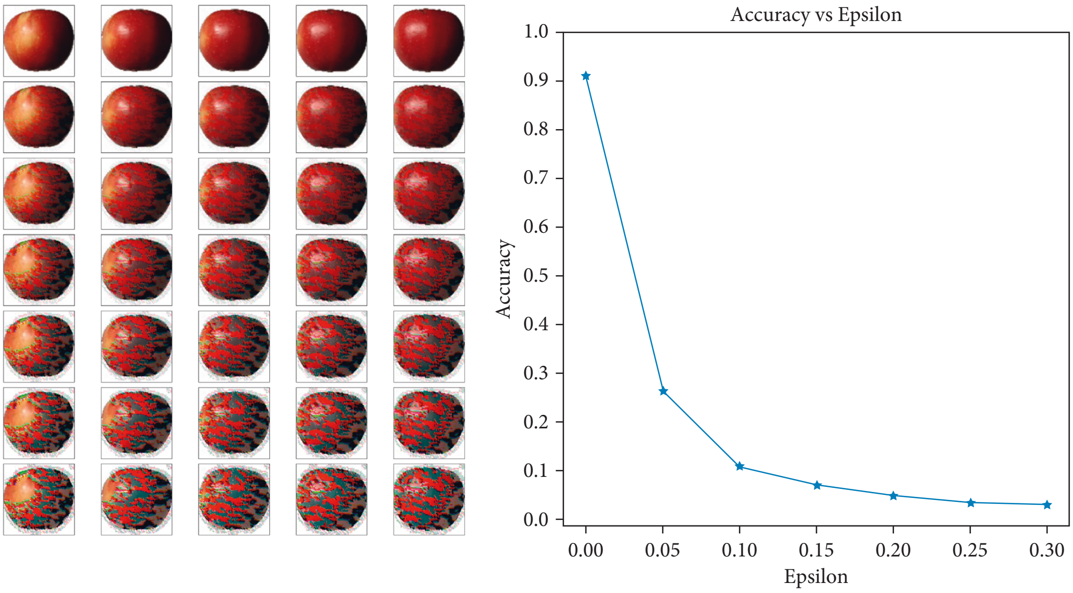
Attention mechanism (AM) has gained a lot of attention in computer vision fields such as object detection and classification [136]. There is need to incorporate AM into DL frameworks for the task of fruit detection and classification since AM has a high capacity to pick features. Knowing that CNN and RNN-based models are widely used in fruit detection and classification, we suggest the use of AM to extract the fruit features which are difficult to distinguish due to their size, shape, and color and enable them into another CNN for information fusion. The goal of AM which is to predict the weight vector for feature maps by model learning makes the integration one of the possible best solutions to tackle the issue of fruit detection and classification.

Recent article has shown the effect of adversarial attack in deep learning models. Even though DL models have shown great success at many complex tasks, they are not as intelligent as we might think. DL models have been found to be sensitive to what is known as adversarial example. The input in an adversarial example is fictitiously generated yet identical to the actual input in terms of perception. Trying to demonstrate the effects of the adversarial attack on fruit detection and classification, we employed an attack mechanism following the fast gradient sign attack (FGSM). We explain and harnessed adversarial examples which are most prevalent adversarial attacks in our experiments in Section 9. Figure 9(a) illustrates the pictorial view of the attacked images at different epsilon while Figure 9(b) shows the classification accuracy of the attacked images. We can vividly note that the more the epsilon rate is, the more the classification accuracy drops; thus, deep neural network is not able to correctly classify input when the input is being attacked, leading to misclassification. In the field of fruit detection and classification, such misdetection and classification can cause a lot of harm and damage to humans. In order to prevent that, we suggest for researchers to develop an adversarial defense network to withstand attacks during fruit detection and classification.

9. Experimental Analysis

In order to give basic understanding of what is meant by DL model to beginner researchers who do not necessarily have skills in computer science especially in the area of agriculture, we present the practical implementation of a simple DL model (CNN) for a fruit classification task. We used the PyTorch framework in our implementation. PyTorch is a famous open-source learning machine library based on torches that have a wrapper in C. It is an alternative to the open-source machine library, TensorFlow. It was primarily developed and published under the modified BSD license by the Facebook AI Research lab (FAIR). It stores data as a 3D-based storage system called tensor rather than vector, like TensorFlow. PyTorch supports many languages, but Python is extensively the most recognized language of all. PyTorch is a more flexible library for developing DL-models. However, Keras and Tensorflow provide built-in architectures with less flexibility. In DL-based research, PyTorch offers maximum versatility and speed.

We employed the Fruit 360 dataset in our experiment as it is currently the widely used dataset in fruit classification by researchers. We maintain the set value for the training and testing sets as originally done by the owner. Table 4 shows our model configurations. We set our hyperparameters (learning rate, no. of epochs, batch size, and dropout) to 0.001, 40, 15, and 0.5, respectively. We made use of the Adam optimizer which is an extension of the stochastic gradient descent. Adam optimizer has seen a wider adoption in CV for DL applications. Figure 10(a) indicates that after 15 epochs, our network achieved a validation accuracy of 100%. To avoid overfitting, we applied dropout (0.50) to the network from which we observed that the validation loss graph decreases alongside the training loss. Table 5 illustrates the details of the employed classifiers, whereas Table 4 shows the implementation details of the CNN architecture. Figure 10(a) shows the model training and validation loss. We plotted the model loss curves against the number of epochs. Figure 10(b) illustrates the training vs the validation accuracy of the model. With the help of the dropout, we were able to overcome overfitting. Our model achieved an overall classification accuracy of 95%. In order to compare the performance of our model which is from scratch implementation and to illustrate more on the application of deep



(a)

(b)

Figure 9: Illustration of classes of Fruit 360.

Table 4: Implementation details of the CNN classifier.

Name	Kernel	Stride	Padding	Input	Output	Param#	Activation
Conv2d-L_1	3 × 3	1 × 1	1 × 1	3 × 100 × 100	32 × 100 × 100	896	ReLU
Conv2d-L_2	3 × 3	1 × 1	1 × 1	32 × 100 × 100	64 × 100 × 100	18,496	ReLU
MaxP2d	2	2	0	64 × 100 × 100	64 × 50 × 50	—	—
Conv2d-L_3	3 × 3	1 × 1	1 × 1	64 × 64	128 × 50 × 50	73,856	ReLU
Conv2d-L_4	3 × 3	1 × 1	1 × 1	128 × 50 × 50	128 × 50 × 50	147, 584	ReLU
MaxP2d	2	2	0	128 × 50 × 50	128 × 25 × 25	—	—
Conv2d-L_5	3 × 3	1 × 1	1 × 1	128 × 25 × 25	256 × 25 × 25	295,168	ReLU
Conv2d-L_6	3 × 3	1 × 1	1 × 1	256 × 25 × 25	256 × 25 × 25	590,080	ReLU
MaxP2d	5	5	—	256 × 25 × 25	256 × 5 × 5	—	—
FC-L	—	—	15,000	6,400	1024	6,554,624	ReLU
FC-L	—	—	—	1,024	512	524,800	ReLU
FC-L	—	—	—	512	120	67,203	—

Param#: parameters, Act: activation, Conv2d-L_1, L_2, L_3, L_4, L_5, L_6: convolution 2 dimension layer 1, 2, 3, 4, 5, and 6, MaxP2d: maximum pooling 2 dimension, and FC-L: fully connected layer.

learning models to fruit classification, we employed through transfer learning the ResNet-50 architecture. We set the parameters as follows: batch size-15, epochs = 40, max lr = $1e-3$, grad clip = $1e-1$, weight decay = $1e-4$, and opt func = torch.optim.Adam.

It is very hard to find the correct learning rate as a comparatively high learning rate induces divergence during the training of deep learning models, whereas a relatively poor learning rate leads to a super-fast model. In addition, we used the "One Cycle Learning Rate Strategy" to slowly boost the learning rate to the highest rate set by the consumer before eventually dropping to a very low rate. After a load of exercise, this rate shift happens. Figure 10(c) shows the training loss vs epoch graph of the ResNet-50 architecture. We saw the effect of the "One Cycle Learning Rate Strategy" from epoch 8 to epoch 16, and the architecture tried to use both highest and lowest values to get the correct learning rate of which at epoch 30, we noticed the convergence by the graph. Increasing the number of epochs and more training explains the advantages of the once cycle learning rate strategy. ResNet-50 model's accuracy can be seen in Figure 10(d) where we plotted the accuracy vs number of epochs graph. The ResNet-50 model achieved 99% classification accuracy on the test set. We noticed a 4% increase from the CNN model in the classification accuracy.

We have successfully created and trained a deep learning model based on CNN and ResNet to classify images of fruits using the Fruit 360 dataset. Table 6 illustrates the classification report of the ResNet-50 architecture. We notice a misclassification at the Lemon, Physalis, and Grapefruit white class. From our result, we also saw that Adam optimization algorithm serves as a good replacement from the classical stochastic gradient descent. During the training process of both models, we saw that the complexity of a model should correspond to the complexity of the classification task to be done and amount of dataset employed,

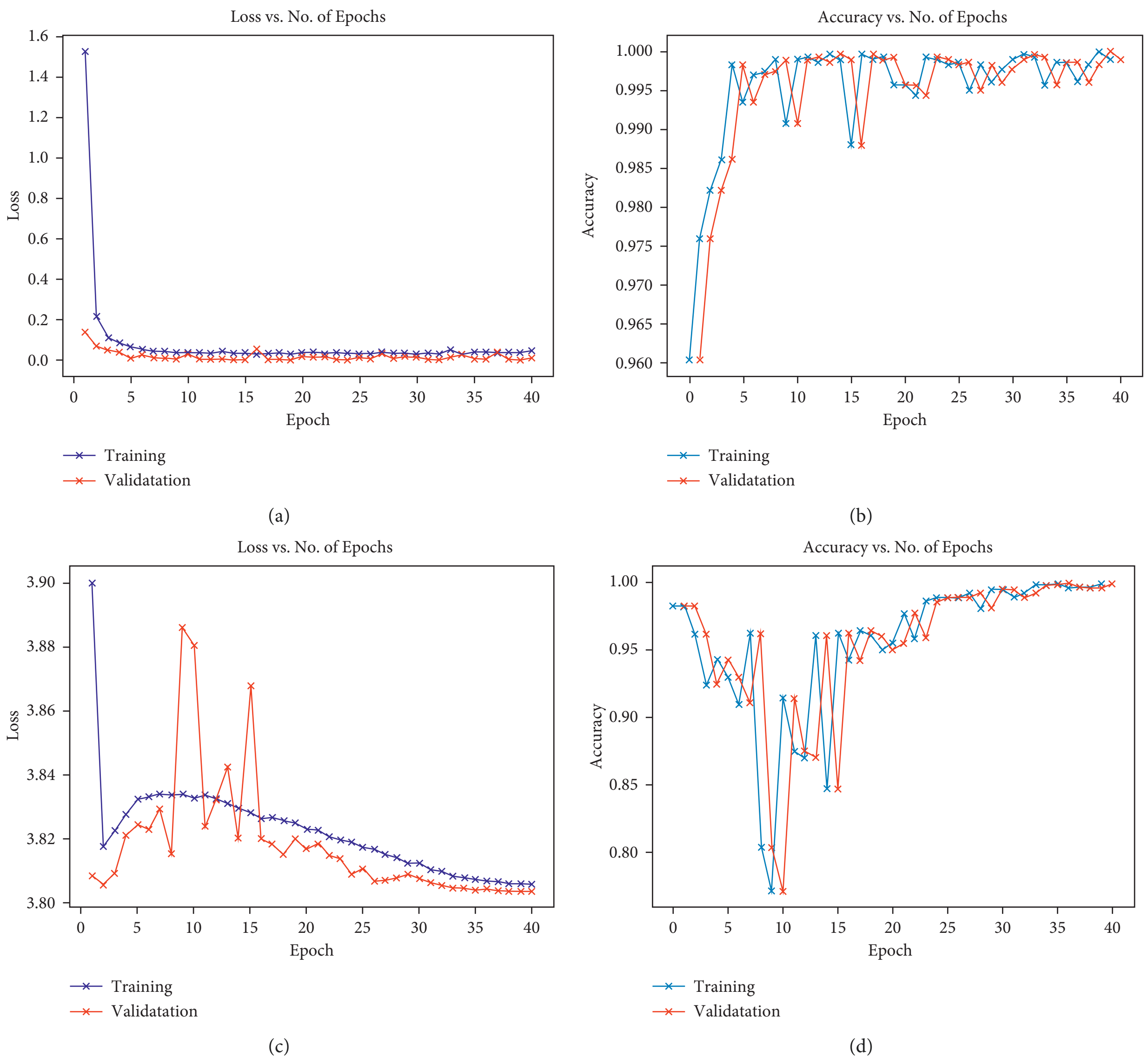


FIGURE 10: Experimental result from scratch CNN implementation vs the ResNet-50 architecture. (a) CNN model loss. (b) CNN model accuracy. (c) ResNet-50 model loss. (d) ResNet-50 model accuracy.

TABLE 5: Result comparisons of the deployed classifiers.

Model	Training loss	Training accuracy	Validation loss	Validation accuracy	Testing accuracy (%)
CNN	0.001	0.999	0.007	0.999	95
ResNet-50	3.803	0.995	3.803	0.996	99

TABLE 6: Classification report of ResNet-50 model.

Class name	Prec.	Recall	F1-s	Support	Class name	Prec.	Recall	F1-s	Support
Apple Braeburn	1.00	1.00	1.00	24	Cherry Wax Black	1.00	1.00	1.00	26
Apple Crimson Snow	1.00	1.00	1.00	23	Cherry Wax Red	1.00	1.00	1.00	22
Apple Golden 1	1.00	1.00	1.00	30	Cherry Wax Yellow	1.00	1.00	1.00	21
Apple Golden 2	1.00	1.00	1.00	20	Chestnut	1.00	1.00	1.00	16
Apple Golden 3	1.00	1.00	1.00	30	Clementine	1.00	1.00	1.00	21
Apple Granny Smith	1.00	1.00	1.00	17	Cocos	1.00	1.00	1.00	31
Apple Pink Lady	1.00	1.00	1.00	21	Dates	1.00	1.00	1.00	28
Apple Red 1	1.00	1.00	1.00	19	Eggplant	1.00	1.00	1.00	20

Table 6: Continued.

Class name	Prec.	Recall	F1-s	Support	Class name	Prec.	Recall	F1-s	Support
Apple Red 2	1.00	1.00	1.00	29	Ginger Root	1.00	1.00	1.00	8
Apple Red 3	1.00	1.00	1.00	20	Granadilla	1.00	1.00	1.00	26
Apple Red Delicious	1.00	1.00	1.00	28	Grape Blue	1.00	1.00	1.00	48
Apple Red Yellow 1	1.00	1.00	1.00	29	Grape Pink	1.00	1.00	1.00	23
Apple Red Yellow 2	1.00	1.00	1.00	36	Grape White	1.00	1.00	1.00	37
Apricot	1.00	1.00	1.00	26	Grape White 2	1.00	1.00	1.00	24
Avocado	1.00	1.00	1.00	23	Grape White 3	1.00	1.00	1.00	24
Avocado ripe	1.00	1.00	1.00	17	Grape White 4	1.00	1.00	1.00	30
Banana	1.00	1.00	1.00	25	Grapefruit Pink	1.00	1.00	1.00	21
Banana Lady Finger	1.00	1.00	1.00	21	Grapefruit White	1.00	0.57	0.73	21
Banana Red	1.00	1.00	1.00	27	Guava	1.00	1.00	1.00	19
Beetroot	1.00	1.00	1.00	23	Hazelnut	1.00	1.00	1.00	22
Blueberry	1.00	1.00	1.00	27	Huckleberry	1.00	1.00	1.00	31
Cactus fruit	1.00	1.00	1.00	23	Kaki	1.00	1.00	1.00	33
Cantaloupe 1	1.00	1.00	1.00	24	Kiwi	1.00	1.00	1.00	19
Cantaloupe 2	1.00	1.00	1.00	31	Kohlrabi	1.00	1.00	1.00	18
Carambola	1.00	1.00	1.00	20	Kumquats	1.00	1.00	1.00	26
Cauliflower	1.00	1.00	1.00	35	Lemon	0.71	1.00	0.83	22
Cherry 1	1.00	1.00	1.00	30	Lemon Meyer	1.00	1.00	1.00	30
Cherry 2	1.00	1.00	1.00	39	Limes	1.00	1.00	1.00	29
Cherry Rainier	1.00	1.00	1.00	40	Lychee	1.00	1.00	1.00	19
Mandarine	1.00	1.00	1.00	20	Plum	1.00	1.00	1.00	23
Mango	1.00	1.00	1.00	20	Plum 2	1.00	1.00	1.00	27
Mango Red	1.00	1.00	1.00	23	Plum 3	1.00	1.00	1.00	44
Mangostan	1.00	1.00	1.00	11	Pomegranate	1.00	1.00	1.00	27
Maracuja	1.00	1.00	1.00	22	Pomelo Sweetie	1.00	1.00	1.00	26
Melon Piel de Sapo	1.00	1.00	1.00	44	Potato Red	1.00	1.00	1.00	17
Mulberry	1.00	1.00	1.00	21	Potato Red Washed	1.00	1.00	1.00	29
Nectarine	1.00	1.00	1.00	18	Potato Sweet	1.00	1.00	1.00	22
Nectarine Flat	1.00	1.00	1.00	19	Potato White	1.00	1.00	1.00	32
Nut Forest	1.00	1.00	1.00	8	Quince	1.00	1.00	1.00	20
Nut Pecan	1.00	1.00	1.00	7	Rambutan	1.00	1.00	1.00	26
Onion Red	1.00	1.00	1.00	21	Raspberry	1.00	1.00	1.00	21
Onion Red Peeled	1.00	1.00	1.00	25	Redcurrant	1.00	1.00	1.00	24
Onion White	1.00	1.00	1.00	24	Salak	1.00	1.00	1.00	31
Orange	1.00	1.00	1.00	28	Strawberry	1.00	1.00	1.00	25
Papaya	1.00	1.00	1.00	24	Strawberry Wedge	1.00	1.00	1.00	30
Passion Fruit	1.00	1.00	1.00	17	Tamarillo	1.00	1.00	1.00	20
Peach	1.00	1.00	1.00	28	Tangelo	1.00	1.00	1.00	26
Peach 2	1.00	1.00	1.00	41	Tomato 1	1.00	1.00	1.00	32
Peach Flat	1.00	1.00	1.00	24	Tomato 2	1.00	1.00	1.00	37
Pear	1.00	1.00	1.00	28	Tomato 3	1.00	1.00	1.00	32
Pear Abate	1.00	1.00	1.00	23	Tomato 4	1.00	1.00	1.00	31
Pear Forelle	1.00	1.00	1.00	30	Tomato Cherry Red	1.00	1.00	1.00	23
Pear Kaiser	1.00	1.00	1.00	15	Tomato Maroon	1.00	1.00	1.00	19
Pear Monster	1.00	1.00	1.00	18	Tomato Yellow	1.00	1.00	1.00	15
Pear Red	1.00	1.00	1.00	33	Walnut	1.00	1.00	1.00	37
Pear Williams	1.00	1.00	1.00	26	Tamarillo	1.00	1.00	1.00	20
Pepino	1.00	1.00	1.00	25	Tangelo	1.00	1.00	1.00	26
Pepper Green	1.00	1.00	1.00	16	Tomato 1	1.00	1.00	1.00	32
Pepper Red	1.00	1.00	1.00	39	Tomato 2	1.00	1.00	1.00	37
Pepper Yellow	1.00	1.00	1.00	30	Tomato 3	1.00	1.00	1.00	32
Physalis	1.00	0.96	0.98	26	Tomato 4	1.00	1.00	1.00	31
Physalis with Husk	1.00	1.00	1.00	23	Tomato Cherry Red	1.00	1.00	1.00	23
Pineapple	1.00	1.00	1.00	27	Tomato Maroon	1.00	1.00	1.00	19
Pineapple Mini	1.00	1.00	1.00	27	Tomato Yellow	1.00	1.00	1.00	15
Pitahaya Red	1.00	1.00	1.00	25	Walnut	1.00	1.00	1.00	37
		Accuracy						1.00	3024
		Macro average				1.00	1.00	1.00	3024
		Weighted average				1.00	1.00	1.00	3024

Prec: precision; F1-s: F1-score.

and hence it causes overfitting. For further deep knowledge of deep learning models for fruit classification, we recommend the use of datasets that are very hard to classify.

10. Conclusion

In this study, we studied and analyzed various deep learning methods proposed by numerous researchers in the domain of fruit detection and classification. While studying different automated approaches for fruit detection and classification, we noticed that previous review papers focused on the application of computer vision techniques in the area. However, deep learning models were not given much attention despite their state-of-the-art performances on many image classification problems. To fill this gap, we conducted an up-to-date review of the recently published literature in the domain of fruit detection and classification that utilized deep learning models. Additionally, a detailed study was presented considering feature description, detection, and classification algorithms as well as different datasets for fruit detection and classification. Moreover, after critical analysis of the reviewed methods, open challenges in terms of datasets, feature representation, and classification algorithms were identified to overcome. Furthermore, to provide elaboration on the use of DL models in the field of agriculture, we also carried out experiments on CNN models. We hope that this survey will provide the basic concepts and applications of DL models in the domain of fruit detection and classification and will help the beginners working in this area.

Data Availability

The data used to support the findings of the study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

We are thankful to Prof. Wu, Prof. Naseem, Prof. Lai, Prof. Umar, Prof. Tumrani, Prof. Ansari, Prof. Alam, Prof. Kamal, Prof. Kamal, Prof. Baig, Dr. Ijaz, Dr. Nasir, and Dr. Guragai for the useful discussion of the study. We acknowledge the Key Laboratory of Network and Data Security, University of Electronic Science and Technology of China, and IoT Research Center, Shenzhen University, China, for providing better environment to carry out the study. We are also thankful to the Department of Statistics, Yazd University, Yazd, Iran, for the financial support.

References

[1] M. Pak and S. Kim, “A review of deep learning in image recognition,” in *Proceedings of the 2017 4th International Conference on Computer Applications and Information Processing Technology (CAIPT)*, pp. 1–3, IEEE, Kuta Bali, Indonesia, Aug. 2017.

[2] H. Zhai, “Research on image recognition based on deep learning technology,” in *Proceedings of the 2016 4th International Conference on Advanced Materials and Information Technology Processing (AMITP 2016)*, Guilin, China, September 2016.

[3] L. Jiang, Y. Fan, Q. Sheng, X. Feng, and W. Wang, “Research on path guidance of logistics transport vehicle based on image recognition and image processing in port area,” *EURASIP Journal on Image and Video Processing*, vol. 2018, no. 1, 2018.

[4] R. C. Harrell, D. C. Slaughter, and P. D. Adsit, “A fruit-tracking system for robotic harvesting,” *Machine Vision and Applications*, vol. 2, no. 2, pp. 69–80, 1989.

[5] W. C. Woo Chaw Seng and S. H. Mirisaee, “A new method for fruits recognition system,” in *Proceedings of the 2009 International Conference on Electrical Engineering and Informatics*, pp. 130–134, Bangi, Malaysia, Aug. 2009.

[6] J. Feng, L. Zeng, and L. He, “Apple fruit recognition algorithm based on multi-spectral dynamic image analysis,” *Sensors*, vol. 19, no. 4, p. 949, 2019.

[7] F. Liu, L. Snetkov, and D. Lima, “Summary on fruit identification methods: a literature review,” in *Proceedings of the 2017 3rd International Conference on Economics, Social Science, Arts, Education and Management Engineering (ESSAEME 2017)*, Atlantic press, AV Amsterdam, Netherlands, July 2017.

[8] G. Capizzi, G. L. Sciuto, C. Napoli, E. Tramontana, and M. Woźniak, “Automatic classification of fruit defects based on Co-occurrence matrix and neural networks,” *Annals of Computer Science and Information Systems*, vol. 5, pp. 861–867, 2015.

[9] Y. Zhang, S. Wang, G. Ji, and P. Phillips, “Fruit classification using computer vision and feedforward neural network,” *Journal of Food Engineering*, vol. 143, pp. 167–177, 2014.

[10] Y. Zhang, P. Phillips, S. Wang, G. Ji, J. Yang, and J. Wu, “Fruit classification by biogeography-based optimization and feedforward neural network,” *Expert Systems*, vol. 33, no. 3, pp. 239–253, 2016.

[11] S. Wang, Y. Zhang, G. Ji, J. Yang, J. Wu, and L. Wei, “Fruit classification by wavelet-entropy and feedforward neural network trained by fitness-scaled chaotic ABC and biogeography-based optimization,” *Entropy*, vol. 17, no. 12, pp. 5711–5728, 2015.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[13] Y. Sakai, T. Oda, M. Ikeda, and L. Barolli, “A vegetable category recognition system using deep neural network,” in *Proceedings of the 2016 10th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS)*, pp. 189–192, IEEE, Fukuoka, Japan, Jul. 2016.

[14] Y.-D. Zhang, Z. Dong, X. Chen et al., “Image based fruit category classification by 13-layer deep convolutional neural network and data augmentation,” *Multimedia Tools and Applications*, vol. 78, no. 3, pp. 3613–3632, 2019.

[15] Y. Bengio and Y. Lecun, “Convolutional networks for images, speech, and time-series,” *Handb. brain theory neural networks*, vol. 3, 1995.

[16] C. Tian, L. Fei, W. Zheng, Y. Xu, W. Zuo, and C.-W. Lin, “Deep learning on image denoising: an overview,” *Neural Networks*, vol. 131, pp. 251–275, 2020.

[17] C.-Y. Lee and S. Osindero, “Recursive recurrent nets with attention modeling for OCR in the wild,” in *Proceedings of*

the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Las Vegas, NV, USA, June 2016.

[18] W. Liu, D. Anguelov, D. Erhan et al., "SSD: single shot multibox detector," *Computer Vision - ECCV 2016*, vol. 9905, pp. 21–37, 2016.

[19] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, 2018.

[20] F. Chollet, "Xception: deep learning with depthwise separable convolutions," in *Proceedings of the - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, IEEE, Honolulu, HI, USA, Oct. 2016.

[21] F. Femling, A. Olsson, and F. Alonso-Fernandez, "Fruit and vegetable identification using machine learning for retail applications," in *Proceedings of the 2018 14th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, pp. 9–15, IEEE, Las Palmas de Gran Canaria, Spain, Nov. 2018.

[22] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: inverted residuals and linear bottlenecks," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Salt Lake City, UT, USA, June 2018.

[23] N. N. A. Abdul Hamid, R. A. Razali, and Z. Ibrahim, "Comparing bags of features, conventional convolutional neural network and AlexNet for fruit recognition," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 14, no. 1, p. 333, 2019.

[24] M. S. Hossain, M. Al-Hammadi, and G. Muhammad, "Automatic fruit classification using deep learning for industrial applications," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 2, pp. 1027–1034, 2019.

[25] H. Mureşan and M. Oltean, "Fruit recognition from images using deep learning," *Acta Universitatis Sapientiae, Informatica*, vol. 10, no. 1, pp. 26–42, 2018.

[26] H. Kuang, C. Liu, L. L. H. Chan, and H. Yan, "Multi-class fruit detection based on image region selection and improved object proposals," *Neurocomputing*, vol. 283, pp. 241–255, 2018.

[27] K. Yamamoto, W. Guo, Y. Yoshioka, and S. Ninomiya, "On plant detection of intact tomato fruits using image analysis and machine learning methods," *Sensors*, vol. 14, no. 7, pp. 12191–12206, 2014.

[28] M. Vogl, J.-Y. Kim, and S.-D. Kim, "A fruit recognition method via image conversion optimized through evolution strategy," in *Proceedings of the 2014 IEEE 17th International Conference on Computational Science and Engineering*, IEEE, Chengdu, China, Dec. 2014.

[29] E. Barnea, R. Mairon, and O. Ben-Shahar, "Colour-agnostic shape-based 3D fruit detection for crop harvesting robots," *Biosystems Engineering*, vol. 146, pp. 57–70, 2016.

[30] F. Khan, M. Khan, N. Iqbal et al., "Prediction of recombination spots using novel hybrid feature extraction method via deep learning approach," *Frontiers in Genetics*, vol. 11, 2020.

[31] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[32] B. Guragai, O. AlShorman, M. Masadeh, and M. B. B. Heyat, "A survey on deep learning classification algorithms for motor imagery," *2020 32nd International Conference on Microelectronics (ICM)*, pp. 1–4, 2020.

[33] F. Akhtar, M. B. Bin Heyat, J. P. Li, P. K. Patel, Rishipal, and B. Guragai, "Role of machine learning in human stress: a review," in *Proceedings of the 2020 17th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, pp. 170–174, AAAI, Chengdu, China, December 2020.

[34] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: a brief survey," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, 2017.

[35] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," *Computer Vision - ECCV 2014*, vol. 28, pp. 818–833, 2014.

[36] F. Amato, A. López, E. M. Peña-Méndez, P. Vaňhara, A. Hampl, and J. Havel, "Artificial neural networks in medical diagnosis," *Journal of Applied Biomedicine*, vol. 11, no. 2, pp. 47–58, 2013.

[37] O. Russakovsky, J. Deng, H. Su et al., "ImageNet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[38] M. J. Nordin, O. W. Xin, and N. Aziz, "Food image recognition for price calculation using convolutional neural network," in *Proceedings of the 2019 3rd International Conference on Digital Signal Processing - ICDSP 2019*, February 2019.

[39] J. Steinbrener, K. Posch, and R. Leitner, "Hyperspectral fruit and vegetable classification using convolutional neural networks," *Computers and Electronics in Agriculture*, vol. 162, pp. 364–372, 2019.

[40] S. W. Chen, S. S. Shivakumar, S. Dcunha et al., "Counting apples and oranges with deep learning: a data-driven approach," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 781–788, 2017.

[41] S. Bargoti and J. Underwood, "Deep fruit detection in orchards," in *Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA)*, June 2017.

[42] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[43] K. Uchida, M. Tanaka, and M. Okutomi, "Coupled convolution layer for convolutional neural network," *Neural Networks*, vol. 105, pp. 197–205, 2018.

[44] M. Goyal, R. Goyal, P. Venkatappa Reddy, and B. Lall, "Activation functions," *Deep Learning: Algorithms and Applications*, vol. 865, pp. 1–30, 2020.

[45] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014.

[46] H.-i. Lim, "A study on dropout techniques to reduce overfitting in deep neural networks," *Lecture Notes in Electrical Engineering*, vol. 716, pp. 133–139, 2021.

[47] N. M. Aszemi and P. D. D. Dominic, "Hyperparameter optimization in convolutional neural network using genetic algorithms," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 6, 2019.

[48] K. Hameed, D. Chai, and A. Rassau, "A comprehensive review of fruit and vegetable classification techniques," *Image and Vision Computing*, vol. 80, pp. 24–44, 2018.

[49] V. Singh, Varsha, and A. K. Misra, "Detection of unhealthy region of plant leaves using image processing and genetic algorithm," in *Proceedings of the 2015 International Conference on Advances in Computer Engineering and Applications*, pp. 1028–1032, IEEE, Ghaziabad, India, Mar. 2015.

[50] J. Jianwei Qin, T. F. Thomas F Burks, D. Dae Gwan Kim, and D. M. Duke M Bulanon, "Classification of citrus peel diseases using color texture feature analysis," in *Proceedings of the Food Processing Automation Conference*, AAAI, Washington, D.C., Columbia, United States, January 2008.

[51] S. Dhole and R. P. Shaikh, "Review of leaf unhealthy region detection using image processing techniques," *Bulletin of Electrical Engineering and Informatics*, vol. 5, no. 4, pp. 451–453, 2016.

[52] Z. Malik, S. Ziauddin, and A. Safi, "Detection and counting of on-tree citrus fruit for crop yield estimation," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 5, 2016.

[53] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 640–651, 2017.

[54] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 386–397, 2020.

[55] K. Lin, L. Gong, Y. Huang, C. Liu, and J. Pan, "Deep learning-based segmentation and quantification of cucumber powdery mildew using convolutional neural network," *Frontiers of Plant Science*, vol. 10, 2019.

[56] X.-f. Wang, Z. Wang, and S.-w. Zhang, "Segmenting crop disease leaf image by modified fully-convolutional networks," *Intelligent Computing Theories and Application*, vol. 11643, pp. 646–652, 2019.

[57] O. Ronneberger, P. Fischer, and T. Brox, "U-net: convolutional networks for biomedical image segmentation," *Lecture Notes in Computer Science*, in *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241, Springer, Switzerland AG, November 2015.

[58] M. Kerkech, A. Hafiane, and R. Canals, "Vine disease detection in UAV multispectral images using optimized image registration and deep learning segmentation approach," *Computers and Electronics in Agriculture*, vol. 174, p. 105446, 2020.

[59] E. L. Stewart, T. Wiesner-Hanks, N. Kaczmar et al., "Quantitative phenotyping of northern leaf blight in UAV images using deep learning," *Remote Sensing*, vol. 11, no. 19, p. 2209, 2019.

[60] Q. Wang, F. Qi, M. Sun, J. Qu, and J. Xue, "Identification of tomato disease types and detection of infected areas based on deep convolutional neural networks and object detection techniques," *Computational Intelligence and Neuroscience*, vol. 2019, pp. 1–15, 2019.

[61] S. Nuske, S. Achar, T. Bates, S. Narasimhan, and S. Singh, "Yield estimation in vineyards by visual grape detection," in *Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2352–2358, San Francisco, CA, USA, Sep. 2011.

[62] S. Nuske, K. Wilshusen, S. Achar, L. Yoder, S. Narasimhan, and S. Singh, "Automated visual yield estimation in vineyards," *Journal of Field Robotics*, vol. 31, no. 5, pp. 837–860, 2014.

[63] C. W. Bac, J. Hemming, and E. J. Van Henten, "Robust pixel-based classification of obstacles for robotic harvesting of sweet-pepper," *Computers and Electronics in Agriculture*, vol. 96, pp. 148–162, 2013.

[64] C. Hung, J. Nieto, Z. Taylor, J. Underwood, and S. Sukkarieh, "Orchard fruit segmentation using multi-spectral feature learning," in *Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, Tokyo, Japan, Nov. 2013.

[65] K. Kapach, E. Barnea, R. Mairon, Y. Edan, and O. B. Shahar, "Computer vision for fruit harvesting robots – state of the art and challenges ahead," *International Journal of Computational Vision and Robotics*, vol. 3, no. 1/2, p. 4, 2012.

[66] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: a retrospective," *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, 2015.

[67] C. L. Zitnick and P. Dollár, "Edge boxes: locating object proposals from edges," *Computer Vision - ECCV 2014*, vol. 8693, pp. 391–405, 2014.

[68] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[69] G. L. Hung, M. S. B. Sahimi, H. Samma, T. A. Almohamad, and B. Lahasan, "Faster R-CNN deep learning model for pedestrian detection from drone images," *SN Computer Science*, vol. 1, no. 2, 2020.

[70] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[71] I. Sa, Z. Ge, F. Dayoub, B. Upcroft, T. Perez, and C. McCool, "Deepfruits: a fruit detection system using deep neural networks," *Sensors*, vol. 16, no. 8, p. 1222, 2016.

[72] M. Rahnemoonfar and C. Sheppard, "Deep count: fruit counting based on deep simulated learning," *Sensors*, vol. 17, no. 4, p. 905, 2017.

[73] L. Fu, Y. Feng, T. Elkamil, Z. Liu, R. Li, and Y. Cui, "Image recognition method of multi-cluster kiwifruit in field based on convolutional neural networks," *Nongye Gongcheng Xuebao/Transactions Chinese Soc. Agric. Eng.*vol. 34, no. 2, pp. 205–211, 2018.

[74] R. Katarzyna and M. Paweł, "A vision-based method utilizing deep convolutional neural networks for fruit variety classification in uncertainty conditions of retail sales," *Applied Sciences*, vol. 9, no. 19, p. 3971, 2019.

[75] B. Divya Shree, R. Brunda, and N. Shobha Rani, "Fruit detection from images and displaying its nutrition value using deep Alex network," *Advances in Intelligent Systems and Computing*, vol. 898, pp. 599–608, 2019.

[76] J. F. Villacrés and F. Auat Cheein, "Detection and characterization of cherries: a deep learning usability case study in Chile," *Agronomy*, vol. 10, no. 6, p. 835, 2020.

[77] D. Thi Phuong Chung and D. Van Tai, "A fruits recognition system based on a modern deep learning technique," *Journal of Physics: Conference Series*, vol. 1327, no. 1, p. 012050, 2019.

[78] K. Bresilla, G. D. Perulli, A. Boini, B. Morandi, L. Corelli Grappadelli, and L. Manfrini, "Single-shot convolution neural networks for real-time fruit detection within the tree," *Frontiers of Plant Science*, vol. 10, 2019.

[79] A. Koirala, K. B. Walsh, Z. Wang, and C. McCarthy, "Deep learning for real-time fruit detection and orchard fruit load estimation: benchmarking of 'MangoYOLO'," *Precision Agriculture*, vol. 20, no. 6, pp. 1107–1135, 2019.

[80] R. Kestur, A. Meduri, and O. Narasipura, "MangoNet: a deep semantic segmentation architecture for a method to detect and count mangoes in an open orchard," *Engineering Applications of Artificial Intelligence*, vol. 77, pp. 59–69, 2019.

[81] Y. Yu, K. Zhang, L. Yang, and D. Zhang, "Fruit detection for strawberry harvesting robot in non-structural environment

based on Mask-RCNN," *Computers and Electronics in Agriculture*, vol. 163, p. 104846, 2019.

[82] Z. Liu, J. Wu, L. Fu et al., "Improved kiwifruit detection using pre-trained VGG16 with RGB and NIR information fusion," *IEEE Access*, vol. 8, pp. 2327–2336, 2020.

[83] G. Lin, Y. Tang, X. Zou, J. Xiong, and J. Li, "Guava detection and pose estimation using a low-cost RGB-D sensor in the field," *Sensors*, vol. 19, no. 2, p. 428, 2019.

[84] H. Altaheri, M. Alsulaiman, and G. Muhammad, "Date fruit classification for robotic harvesting in a natural environment using deep learning," *IEEE Access*, vol. 7, pp. 117115–117133, 2019.

[85] Y. Ge, Y. Xiong, and P. J. From, "Instance segmentation and localization of strawberries in farm conditions for automatic fruit harvesting," *IFAC-PapersOnLine*, vol. 52, no. 30, pp. 294–299, 2019.

[86] P. Ganesh, K. Volle, T. F. Burks, and S. S. Mehta, "Deep orange: mask R-CNN based orange detection and segmentation," *IFAC-PapersOnLine*, vol. 52, no. 30, pp. 70–75, 2019.

[87] T. T. Santos, L. L. de Souza, A. A. dos Santos, and S. Avila, "Grape detection, segmentation, and tracking using deep neural networks and three-dimensional association," *Computers and Electronics in Agriculture*, vol. 170, p. 105247, 2020.

[88] J. Xiong, Z. Liu, S. Chen et al., "Visual detection of green mangoes by an unmanned aerial vehicle in orchards based on a deep learning method," *Biosystems Engineering*, vol. 194, pp. 261–272, 2020.

[89] L. Fu, J. Duan, X. Zou et al., "Fast and accurate detection of banana fruits in complex background orchards," *IEEE Access*, vol. 8, pp. 196835–196846, 2020.

[90] W. Jia, Y. Tian, R. Luo, Z. Zhang, J. Lian, and Y. Zheng, "Detection and segmentation of overlapped fruits based on optimized mask R-CNN application in apple harvesting robot," *Computers and Electronics in Agriculture*, vol. 172, p. 105380, 2020.

[91] C. Liang, J. Xiong, Z. Zheng et al., "A visual detection method for nighttime litchi fruits and fruiting stems," *Computers and Electronics in Agriculture*, vol. 169, p. 105192, 2020.

[92] D. P. Kingma and J. L. Ba, "Adam: a method for stochastic optimization," 2015.

[93] J. Redmon and A. Farhadi, "YOLOv3: an incremental improvement," 2018.

[94] X. Kou, S. Liu, K. Cheng, and Y. Qian, "Development of a YOLO-V3-based model for detecting defects on steel strip surface," *Measurement*, vol. 182, p. 109454, 2021.

[95] R. Shi, T. Li, and Y. Yamaguchi, "An attribution-based pruning method for real-time mango detection with YOLO network," *Computers and Electronics in Agriculture*, vol. 169, p. 105214, 2020.

[96] Y. Ren, C. Zhu, and S. Xiao, "Object detection based on fast/faster RCNN employing fully convolutional architectures," *Mathematical Problems in Engineering*, vol. 2018, pp. 1–7, 2018.

[97] R. Kirk, G. Cielniak, and M. Mangan, "L∗a∗b∗Fruits: a rapid and robust outdoor fruit detection system combining bio-inspired features with one-stage deep learning networks," *Sensors*, vol. 20, no. 1, p. 275, 2020.

[98] L. Fu, Y. Feng, Y. Majeed et al., "Kiwifruit detection in field images using Faster R-CNN with ZFNet," *IFAC-PapersOnLine*, vol. 51, no. 17, pp. 45–50, 2018.

[99] F. Yuesheng, S. Jian, X. Fuxiang et al., "Circular fruit and vegetable classification based on optimized GoogLeNet," *IEEE Access*, vol. 9, pp. 113599–113611, 2021.

[100] W. Zhang, D. Zhao, W. Gong, Z. Li, Q. Lu, and S. Yang, "Food image recognition with convolutional neural networks," in *Proceedings of the 2015 IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom)*, IEEE, Beijing, China, Aug. 2015.

[101] G. Zeng, "Fruit and vegetables classification system using image saliency and convolutional neural network," in *Proceedings of the 2017 IEEE 3rd Information Technology and Mechatronics Engineering Conference (ITOEC)*, IEEE, Chongqing, China, Oct. 2017.

[102] G. Ciocca, P. Napoletano, and R. Schettini, "Food recognition: a new dataset, experiments, and results," *IEEE Journal of Biomedical and Health Informatics*, vol. 21, no. 3, pp. 588–598, 2017.

[103] A. Kausar, M. Sharif, J. Park, and D. R. Shin, "Pure-CNN: a framework for fruit images classification," in *Proceedings of the 2018 International Conference on Computational Science and Computational Intelligence (CSCI)*, IEEE, Las Vegas, NV, USA, Dec. 2018.

[104] L. Zhu, Z. Li, Z. Li, C. Li, J. Wu, and J. Yue, "High performance vegetable classification from images based on AlexNet deep learning model," *International Journal of Agricultural and Biological Engineering*, vol. 11, no. 4, pp. 190–196, 2018.

[105] I. Hussain, Q. He, and Z. Chen, "Automatic fruit recognition based on DCNN for commercial source Trace system," *International Journal on Computational Science & Applications*, vol. 8, no. 2/3, pp. 01–14, 2018.

[106] S.-H. Wang and Y. Chen, "Fruit category classification via an eight-layer convolutional neural network with parametric rectified linear unit and dropout technique," *Multimedia Tools and Applications*, vol. 79, no. 21-22, pp. 15117–15133, 2020.

[107] M. Teletin and L. Dobai, "Lightweight models for fruits recognition," in *Proceedings of the 2019 IEEE 13th International Symposium on Applied Computational Intelligence and Informatics (SACI)*, pp. 69–74, IEEE, Timisoara, Romania, May 2019.

[108] L. Alzubaidi, O. Al-Shamma, M. A. Fadhel, Z. M. Arkah, and F. H. Awad, "A deep convolutional neural network model for multi-class fruits classification," *Advances in Intelligent Systems and Computing*, vol. 1181, pp. 90–99, 2021.

[109] G. Xue, S. Liu, and Y. Ma, "A hybrid deep learning-based fruit classification using attention model and convolution autoencoder," *Complex & Intelligent Systems*, 2020.

[110] W. Liu, "Interfruit : deep learning network for classifying fruit images," *bioRxiv*, 2020.

[111] A. A. Lydia and F. S. Francis, "Multi-label classification using deep convolutional neural network," in *Proceedings of the 2020 International Conference on Innovative Trends in Information Technology (ICITIIT)*, pp. 1–6, IEEE, Kottayam, India, Feb. 2020.

[112] D. M. Asriny, S. Rani, and A. F. Hidayatullah, "Orange fruit images classification using convolutional neural networks," *IOP Conference Series: Materials Science and Engineering*, vol. 803, no. 1, p. 012020, 2020.

[113] M. Momeny, A. Jahanbakhshi, K. Jafarnezhad, and Y.-D. Zhang, "Accurate classification of cherry fruit using

deep CNN based on hybrid pooling approach," *Postharvest Biology and Technology*, vol. 166, p. 111204, 2020.

[114] J. L. Rojas-Aranda, J. I. Nunez-Varela, J. C. Cuevas-Tello, and G. Rangel-Ramirez, "Fruit classification for retail stores using deep learning," *Lecture Notes in Computer Science,Pattern Recognition*, vol. 12088, 2020.

[115] H. B. Unal, E. Vural, B. K. Savas, and Y. Becerikli, "Fruit recognition and classification with deep learning support on embedded system (fruitnet)," in *Proceedings of the 2020 Innovations in Intelligent Systems and Applications Conference (ASYU)*, pp. 1–5, IEEE, Istanbul, Turkey, Oct. 2020.

[116] R. Qian, Y. Yue, F. Coenen, and B. Zhang, "Traffic sign recognition with convolutional neural network based on max pooling positions," in *Proceedings of the 2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, pp. 578–582, IEEE, Changsha, China, Aug. 2016.

[117] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: inverted residuals and linear bottlenecks mark," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Salt Lake City, UT, USA, June 2018.

[118] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "ShuffleNet V2: practical guidelines for efficient CNN architecture design," *Computer Vision - ECCV 2018*, vol. 11218, pp. 122–138, 2018.

[119] T. Tan, S. Yin, K. Liu, and M. Wan, "On the convergence speed of AMSGRAD and beyond," in *Proceedings of the 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, IEEE, Portland, OR, USA, Nov. 2019.

[120] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Las Vegas, NV, USA, June 2016.

[121] W. Li, W. Tao, J. Qiu, X. Liu, X. Zhou, and Z. Pan, "Densely connected convolutional networks with attention LSTM for crowd flows prediction," *IEEE Access*, vol. 7, pp. 140488–140498, 2019.

[122] A. Rocha, D. C. Hauagge, J. Wainer, and S. Goldenstein, "Automatic fruit and vegetable classification from images," *Computers and Electronics in Agriculture*, vol. 70, no. 1, pp. 96–104, 2010.

[123] D. Lai, M. B. B. Heyat, F. I. Khan, and Y. Zhang, "Prognosis of sleep bruxism using power spectral density approach applied on EEG signal of both EMG1-EMG2 and ECG1-ECG2 channels," *IEEE Access*, vol. 7, pp. 82553–82562, 2019.

[124] M. B. B. Heyat, D. Lai, F. I. Khan, and Y. Zhang, "Sleep bruxism detection using decision tree method by the combination of C4-P4 and C4-A1 channels of scalp EEG," *IEEE Access*, vol. 7, pp. 102542–102553, 2019.

[125] R. Pal, M. B. Bin Heyat, Z. You et al., "Effect of maha mrityunjaya HYMN recitation on human brain for the analysis of single EEG channel C4-A1 using machine learning classifiers on yoga practitioner," in *Proceedings of the 2020 17th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, pp. 89–92, IEEE, Chengdu, China, Dec. 2020.

[126] M. B. Bin Heyat, F. Akhtar, A. Khan et al., "A novel hybrid machine learning classification for the detection of bruxism patients using physiological signals," *Applied Sciences*, vol. 10, no. 21, p. 7410, 2020.

[127] O. Alshorman et al., "Frontal lobe real-time EEG analysis using machine learning techniques for mental stress detection," *Journal of Integrative Neuroscience*, vol. 9, no. 1, pp. 141–145, 2021.

[128] C. Chola, M. B. B. Heyat, F. Akhtar et al., "IoT based intelligent computer-aided diagnosis and decision making system for health care," in *Proceedings of the 2021 International Conference on Information Technology (ICIT)*, pp. 184–189, IEEE, Amman, Jordan, Jul. 2021.

[129] L. Ali, Z. He, W. Cao, H. T. Rauf, Y. Imrana, and M. B. Bin Heyat, "MMDD-ensemble: a multimodal data-driven ensemble approach for Parkinson's disease detection," *Frontiers in Neuroscience*, vol. 15, pp. 1–11, 2021.

[130] M. B. B. Heyat, D. Lai, F. Akhtar et al., "Bruxism detection using single-channel C4-A1 on human sleep S2 stage recording," in *Intelligent Data Analysis Intelligent Data Analysis: From Data Gathering to Data Comprehension*, D. Gupta, S. Bhattacharyya, and A. Khanna, Eds., pp. 347–367, John Wiley & Sons, 1st ed. edition, 2020.

[131] H. Bb, F. Akhtar, A. Mehdi, S. Azad, S. Azad, and S. Azad, "Normalized power are used in the diagnosis of insomnia medical sleep syndrome through EMG1-EMG2 channel," *Austin Journal of Sleep Disorders*, vol. 4, no. 1, pp. 2–4, 2017.

[132] D. Lai, Y. Zhang, X. Zhang, Y. Su, and M. B. Bin Heyat, "An automated strategy for early risk identification of sudden cardiac death by using machine learning approach on measurable arrhythmic risk markers," *IEEE Access*, vol. 7, pp. 94701–94716, 2019.

[133] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Las Vegas, NV, USA, June 2016.

[134] H. Singh, N. Singh, and S. Dolui, "Effect of inter-fruit competition on development of physiological disorder â€œAril browningâ€ in pomegranate (Punica granatum L.)," *Journal of Applied and Natural Science*, vol. 8, no. 4, pp. 1835–1838, 2016.

[135] S. Zhang, W. Huang, and C. Zhang, "Three-channel convolutional neural networks for vegetable leaf disease recognition," *Cognitive Systems Research*, vol. 53, pp. 31–41, 2019.

[136] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu, "Recurrent models of visual attention," adv. Neural inf. Process. Syst.," 2014, http://arxiv.org/abs/1406.6247.