Available at www.sciencedirect.com

INFORMATION PROCESSING IN AGRICULTURE 9 (2022) 24–37

journal homepage: www.elsevier.com/locate/inpa

Real-time visual inspection system for grading fruits using computer vision and deep learning techniques

Nazrul Ismail, Owais A. Malik *

Digital Science, Faculty of Science, Universiti Brunei Darussalam, Jln Tungku Link, Gadong BE1410, Brunei Darussalam

A R T I C L E I N F O

Article history:
Received 18 May 2020
Received in revised form
21 December 2020
Accepted 24 January 2021
Available online 11 February 2021

Keywords:
Deep learning
Fruit classification
Computer vision
Real-time system
Raspberry Pi
Agriculture

A B S T R A C T

Traditional manual visual grading of fruits has been one of the important challenges faced by the agricultural industry due to its laborious nature as well as inconsistency in inspection and classification process. Automated defects detection using computer vision and machine learning has become a promising area of research with a high and direct impact on the domain of visual inspection. In this study, we propose an efficient and effective machine vision system based on the state-of-the-art deep learning techniques and stacking ensemble methods to offer a non-destructive and cost-effective solution for automating the visual inspection of fruits' freshness and appearance. We trained, tested and compared the performance of various deep learning models including ResNet, DenseNet, MobileNetV2, NASNet and EfficientNet to find the best model for the grading of fruits. The proposed system also provides a real time visual inspection using a low cost Raspberry Pi module with a camera and a touchscreen display for user interaction. The real time system efficiently segments multiple instances of the fruits from an image and then grades the individual objects (fruits) accurately. The system was trained and tested on two data sets (apples and bananas) and the average accuracy was found to be 99.2% and 98.6% using EfficientNet model for apples and bananas test sets, respectively. Additionally, a slight improvement in the recognition rate (0.03% for apples and 0.06% for bananas) was noted while applying the stacking ensemble deep learning methods. The performance of the developed system has been found higher than the existing methods applied to the same data sets previously. Further, during real-time testing on actual samples, the accuracy was found to be 96.7% for apples and 93.8% for bananas which indicates the efficacy of the developed system.

© 2021 China Agricultural University. Production and hosting by Elsevier B.V. on behalf of KeAi. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

1. Introduction

Agriculture industry has a vital role in the economy of many countries in the world. An important sector of agriculture business is the production and supply of the fresh fruits and vegetables to the vendors and markets. The growing demand for effective food production and quick and safe supply to the market has led to the development and use of var-

* Corresponding author at: Digital Science, Faculty of Science, Universiti Brunei Darussalam, Brunei Darussalam.
E-mail address: owais.malik@ubd.edu.bn (O.A. Malik).

Peer review under responsibility of China Agricultural University.
https://doi.org/10.1016/j.inpa.2021.01.005

2214-3173 © 2021 China Agricultural University. Production and hosting by Elsevier B.V. on behalf of KeAi.
This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

ious innovative technologies in this industry [1]. The technologies such as Internet of Things (IoT) based smart farming has been found useful in improving the quality of fruit and vegetable yields [2]. Moreover, the use of intelligent logistic by medium and large scale enterprises has reduced the time to sort, package and deliver the production to the market [3]. However, at the level of small scale agricultural enterprises and farmers, limited embracing of the new technologies has been noticed [1]. Two major challenges in adoption of these technologies by them are the overall increased cost and the requirement for learning specialized skills. Hence, there is a rising need to develop low-cost and easy-to-use solutions for these enterprises and farmers so that they can take more advantage of the new technologies. This study focuses on the automation of grading and sorting process which is one of the important phase of the fresh fruits' supply chain system. Since outer appearance reflects the freshness of a fruit and marks its selling point, it is one of the major criterion in grading the fruits. The tasks of grading and sorting of fruits according to their outer appearance and freshness are still laborious and challenging in nature at the level of small scale enterprises. The manual quality control highly depends on the trained humans to inspect the outer appearance of the fruits and then make a decision about their grading. Automating the grading and sorting of fruits is not only an essential step to deal with the inconsistency in classification but it also helps in reducing the labor cost and time spent in packaging, pricing and supplying these fruits to the markets or vendors. Thus, an alternative solution for low budget enterprises is to opt for a low-cost intelligent fruits grading system.

In recent past, efforts have been made to automate the fruits' classification problem based on their outer appearance or freshness by employing computer vision and machine learning techniques [3–9]. However, in most of these studies, grading has been done using the traditional approach of feature extraction and applying machine learning techniques. For example, in [4] the apples have been classified by using color, texture and shape feature descriptors, namely Global Color Histogram, Color Coherence Vector (CCV), Local Binary Patterns (LBP), Complete Local Binary Patterns (CLBP) and Zernike Moments (ZM). The extracted features were used individually as well as in combination to train and test the machine learning techniques with a highest accuracy of 95.9% using a combination of CCV, CLBP and ZM. Further, in [5] both unsupervised and supervised learning algorithms have been used for apples' grading. Initially, K-means clustering was used for segmentation of defected apples and in the next stage statistical, textural and geometric features were extracted from the refined defected regions. These features were used to train, test and compare the performance of three machine learning techniques namely Support Vector Machine (SVM), MLP and K-Nearest Neighbor (KNN). The results of this study showed highest grading results using SVM classifier with recognition rates of 92.5% for healthy and defected categories and 89.2% for three quality categories (in terms of ranks). In a recent research on measuring the ripeness quality on bananas, an artificial neural network (ANN) based framework has been proposed using different features like color, development of brown spots, and Tamura statistical texture [6]. The performance of the ANN model was compared with various other techniques including SVM, naive Bayes, KNN, decision tree, and discriminant analysis. The findings showed that the proposed system has the highest overall recognition rate 97.75%. Similar approaches have been followed for determining the ripeness and maturity level of other fruits including blueberry [7], oil palm fruit (Elaeis guineensis) [8] and oranges [9]. Though the performance of the machine learning models in these studies is quite good but these models are mainly based on the hand-craft feature extraction methods which are mostly time-consuming and dependent on the type of images (fruits) used for training and testing. Moreover, these models have been trained and tested only for small data sets which increases the risk of biased predictions.

An alternative approach to deal the above problems is to use deep learning techniques to develop a fruits' classification and grading system. The deep learning based models can extract the relevant features automatically and do not require manual intervention. This approach has been followed in some of the recent studies [10–16]. A grading system for apple data set has been developed in [10] using pre-trained Inceptionv3 deep learning model which can classify the apples into four grades with a top 5 accuracy of 90%. This system has used transfer learning approach using fruit 360 dataset and used it to grade self-collected data set of 150 apples. Similar approach has been followed in few other studies [11,12]. Deep learning has also been applied to find the lesion in the apple fruits [13]. The use of Cycle-Consistent Adversarial Network (CycleGAN) for augmentation along with deep learning classifiers has been proved successful (95.57% using YOLOV3-dense) in detecting the lesion from the apple images. A different approach has been adopted in [14] where 3-D surface meshes have been used to train and test the Convolutional Neural Networks (CNNs) for identification of bruised apples. The best predictive model has been reported with an accuracy of 97.67% in this study. The use of deep learning has also been explored for postharvest classification of Cavendish banana [15]. The model was trained using a self-designed Convolutional Neural Network (CNN) on four classes with a total of 1116 images and an average accuracy of 90% has been reported for test data. Recently, a deep learning based maturity classification system for papaya fruits has been proposed in [16]. Various models were applied in this study on a self-collected data set of 300 images and 100% accuracy has been reported on 30 test samples using VGG19 model. The results of these studies indicate the successful application of deep learning techniques in fruits' classification and grading. Although, the accuracy of the above proposed deep learning based systems was found between 90% and 100% in these studies, most of these models were trained and tested on small data sets. Moreover, most of the existing studies have developed and tested the models either only on off-line image data sets or have used expensive vision systems to capture the real-time images.

This work presents the application of current state-of-the-art deep learning models for developing an automated visual inspection system for fruits. Various deep learning have been applied and compared to build a robust system to be deployed for real time testing. In order to further improve the grading performance of the system, stacking ensemble of deep learning has been applied and new models have been generated. The developed system uses a low cost Raspberry Pi module with a camera and a touchscreen display for user interaction during real time testing.

Our contributions in this study are highlighted as follows:

- We have proposed and designed a low-cost machine vision system that is capable of automating the visual inspection of fruits in real-time.
- We have developed an end-to-end approach using computer vision and Convolutional Neural Network (CNN) models (individual and ensemble) in order to detect the visual defects in fruits for agriculture domain.
- The system can work both in offline and real time mode, and can accurately grade the multiple instances of the fruits in a given image or real time using efficient segmentation process.

The rest of the paper is organized as follows: Section 2 outlines the materials and methods of the proposed system. Section 3 presents the results and discussion of the developed model on test data sets and the actual samples. Finally, conclusions and future work are presented in Section 4.

2. Materials and methods

2.1. Proposed system

The overall flow of the proposed system is presented in Fig. 1. The system is composed of three components 1) a computer vision and deep learning based fruits' classifier, 2) a Raspberry Pi camera module for real-time image acquisition and testing and 3) a touch screen front-end with graphical user interface for displaying the real-time classification output. Various state-of-the-art deep learning techniques were initially trained and tested for the fruits' image data sets and then these off-line models for each technique were saved and deployed for real-time testing. The components of the system are explained in the subsequent sub-sections.

2.2. System hardware

For real-time testing of the deep learning model, a prototype system was developed. The hardware of the prototype system consists of two Raspberry Pi modules: a Google AIY Vision Kit (Raspberry Pi Zero accelerated with vision bonnet for neural network inference and a Pi Camera) and a Raspberry Pi module with a touchscreen to display the real-time feedback from the camera using our system as shown in Fig. 2. In addition, a lighting box was used to enhance the environment-lighting conditions thus allowing a controlled setting where an equal distribution of light is illuminated onto the test samples for accurate classification during real-time testing.

2.3. Classification model

In order to develop the real-time visual inspection system for grading the fruits, deep learning models were trained and tested using the existing data sets. The details of the data sets, image pre-processing and training/testing of the models are described below.

2.3.1. Data sets

Deep learning models were trained and tested for two data sets. The first data set consists of 8791 apple images where 4693 images are of defected apples and the remaining 3946 images are from healthy class [17]. The data set images have been taken from an Internal Feeding Worm (IFW) database of the Comprehensive Automation for Specialty Crops (CASC) research project with a prior consent from Purdue University. The data set contains images for four cultivars of apple: Fuji, Golden Delicious, Red Delicious, and York. Each cultivar has images from various stages of ripening of healthy and defected apples. The dataset includes two groups of data (2009 data set and 2010 data set) where both were collected at the Penn State University Fruit Research and Extension Center. The images of defected apples have visible dark spots on outer skin due to damage from internal feeding worms. The images of individual apples have been clipped to 120×120 pixels size for image processing tasks. The distribution of the healthy and defected classes for these cultivars is presented in Fig. 3. Since cultivars do not significantly affect the learning of the algorithm, we considered the grading as a binary classification problem by combining the cultivars into their respective external state (i.e. Healthy and Defected). Fig. 4 shows few samples from apples' data set.

The second data set consists of images of Egyptian banana species with ground truth labels for different ripening levels that has been curated by Mazen et al. [6]. This data set has 300 images of bananas which have been categorized into 4 different ripening levels such as unripe (green banana), yellowish green, mid-ripe and overripe. The data set was collected internally by the curators using a high resolution camera. The distribution of these classes is presented in Fig. 5. This data set has a class imbalance problem for yellowish green and overripe categories. The class imbalanced problem has been solved by using the data augmentation technique as explained in Section 2.3.4. Fig. 6 shows few samples from the bananas' data set.

2.3.2. Image pre-processing

In most vision systems, acquired images are often corrupted by undesirable high-frequency signals (random noise) which cause random variations in intensity, illumination and poor contrast. We employed Gaussian filter for image noise removal. Gaussian filter is a type of linear smoothing filters where the kernel has isotropic or circular symmetrical weights drawn from a normal distribution. In image processing, Gaussian filter is applied as two-dimensional zero-mean discrete function which results in uniform smoothing across the image in all directions (Fig. 7). The Gaussian kernel function applied to an image at pixel (x, y) is given by:

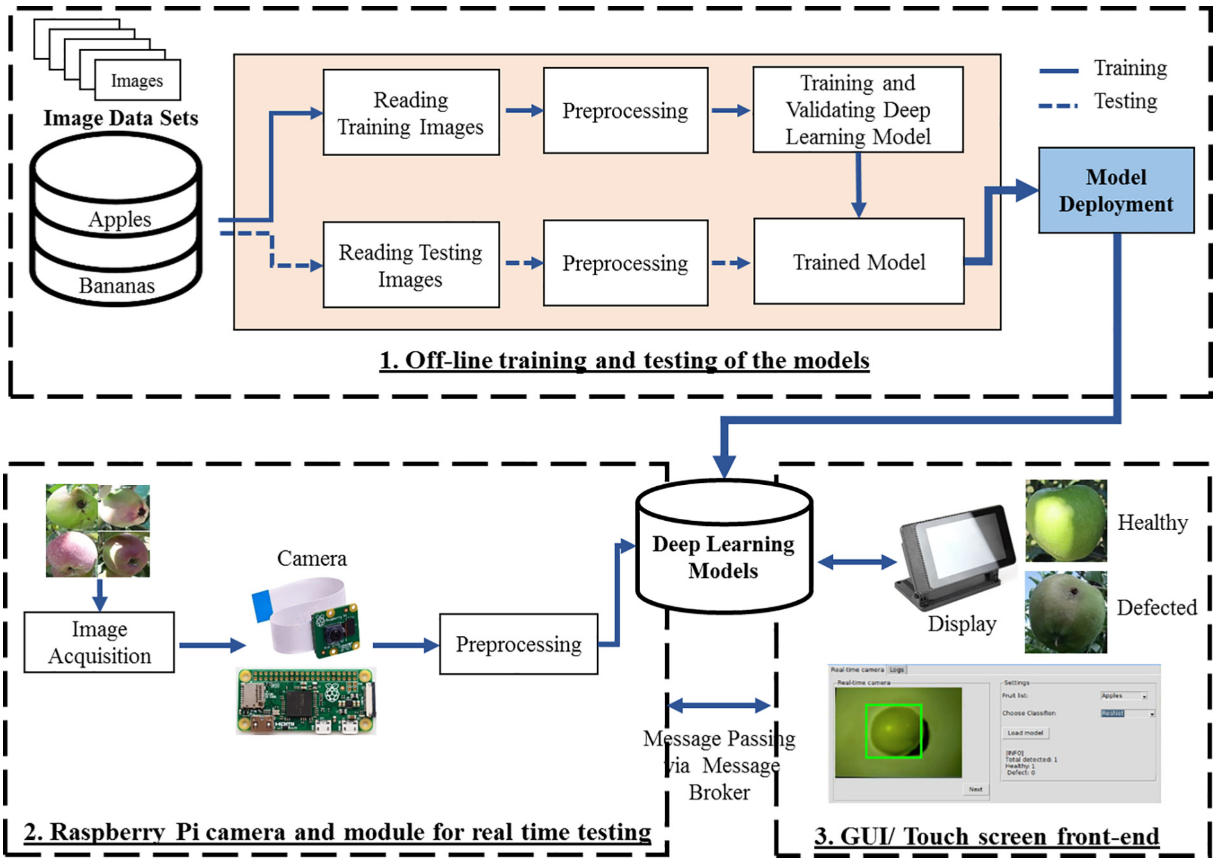

Fig. 1 – Proposed System.

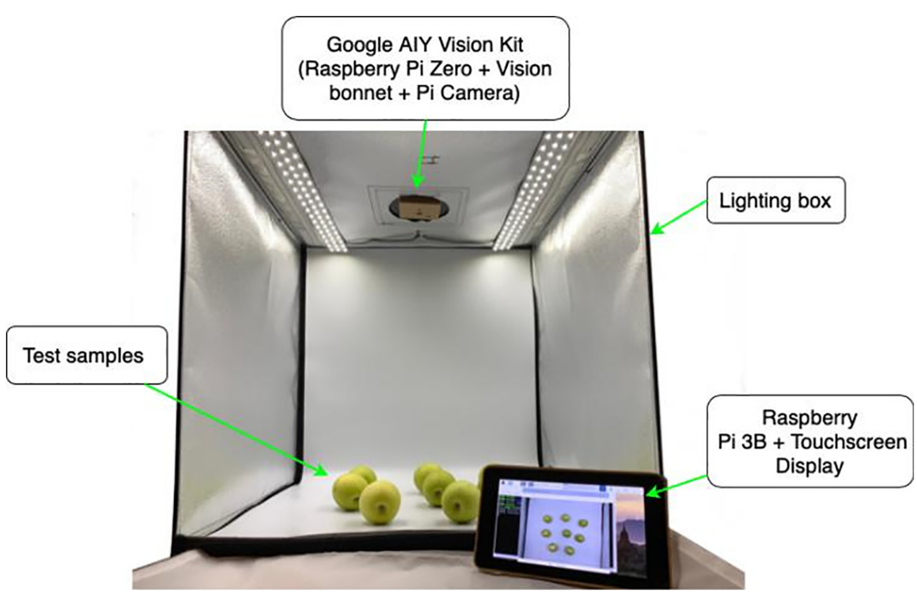

Fig. 2 – Hardware for the Proposed System.

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \tag{1}$$

The Gaussian Kernel applies smoothing by replacing pixel in the original image with the weighted average of neighborhood pixel such that the weight of the pixels that are distant to the central pixel decreases monotonically. The whole smoothing operation is done by a cross-correlation function onto the input images $f\,(x,\ y)$ and can be summarized as below:

$$f'(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} G(x,t) \otimes f(x-s, y-t) \tag{2}$$

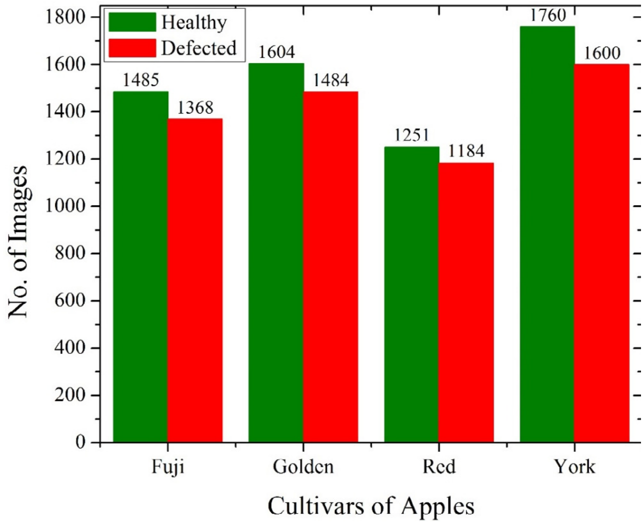

Fig. 3 – Class distribution of healthy and defected apples for each cultivar.

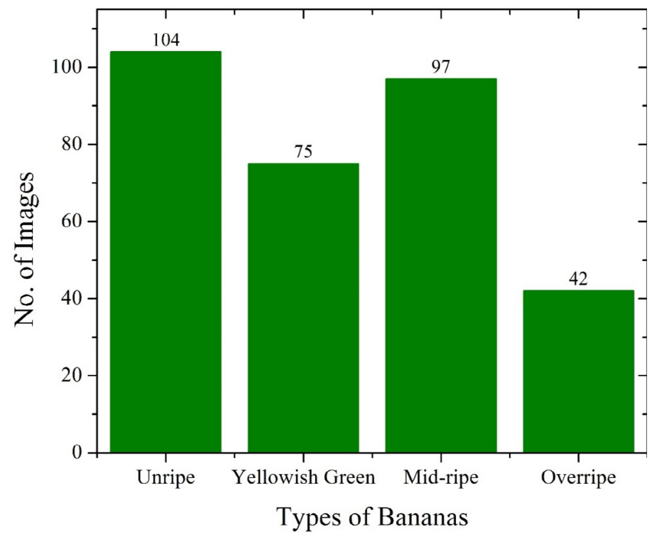

Fig. 5 – Class distribution of bananas' data set.

We applied a minimal Gaussian filter to both apples and banana data sets and set the value of 0.01 with a kernel size of 3 × 3. An example of the smoothing operation on apples' data set is shown in Fig. 8.

To eliminate any undistributed illumination across an image in the data set, we applied histogram equalization using contrast limited adaptive histogram equalization (CLAHE) which is a local histogram equalization method [18]. The fruit images were divided into grids and for each of these blocks, histograms were equalized. To avoid the noise amplification problem, contrast limiting threshold was applied. The function was implemented using OpenCV.

Fig. 4 – Samples from apples' data set.

Fig. 6 – Samples from bananas' data set.

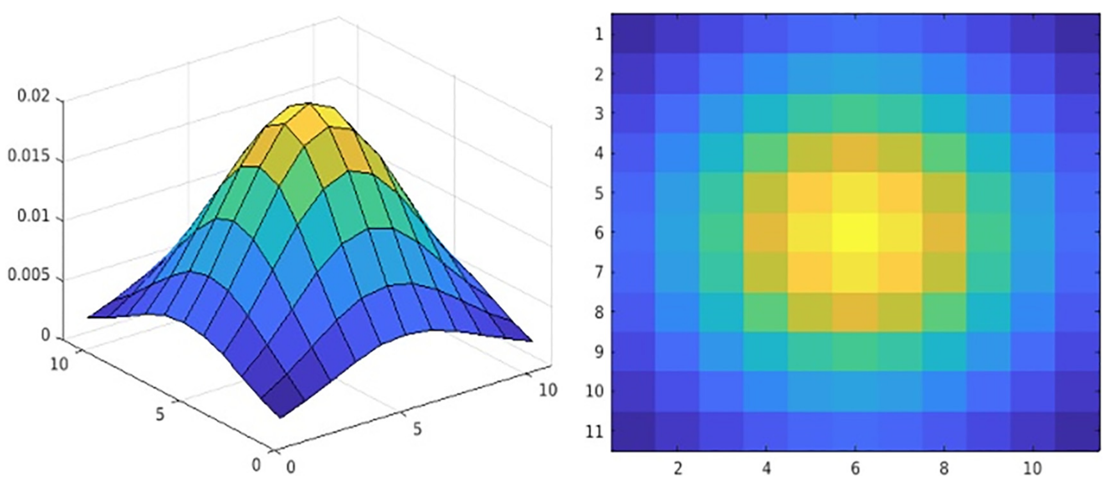

Fig. 7 – Visualization of Gaussian Kernel function of size 11 × 11, = 3, where the yellow region signify larger weights onto central pixel of the image with respect to kernel being convolved.

Fig. 8 – Gaussian smoothing applied on apples' data set.

Fig. 9 – Image enhancement using CLAHE on apples' dataset.

Fig. 9 shows a sample of image enhancement for apples' data set using CLAHE.

2.3.3. *Image segmentation*

Since the images in the data sets had different background than the background of the real-time system so we used two separate segmentation techniques for foreground extraction. Mean shift clustering [19] was used for pre-processing of the images for training the deep learning network while watershed segmentation [20] was used for pre-processing of the images during real-time testing of the developed model. These segmentation techniques are briefly described below.

2.3.3.1. *Mean shift clustering*. Mean shift algorithm seeks modes or local maxima in the feature vector space. The feature space can be in the form of pixels intensity, gradients or textures. The goal of the algorithm is to group features that belong together into the region (*attraction basin*) for which all data points trajectories lead to the same mode (peak) [19]. Mean shift finds attraction basin through multiple *restart gradient descent*, where it randomly selects the local maxima at any data point x_i of a density function *f(x)* and mean shift computes the gradient and takes an uphill step. We kept the bandwidth size in selecting neighborhood pixels according to the optimal size for each of the target data set that prevents over- and under- segmentation. Later, Otsu thresholding was performed to the quantized images in order to extract the region of interest (ROI). Fig. 10 depicts an example of the whole process of segmentation of apples' and bananas' dataset using this method. The first column corresponds to the input image, the second column is the mean shifted mask using $L^*u^*v^*$ color space and the third column is the segmented ROI after applying Otsu thresholding.

2.3.3.2. *Watershed segmentation*. It is a controlled marker based segmentation technique which utilizes mathematical morphology for separating the foreground and background information [20]. It transforms gray scale images as a topographical landscape where bright pixels are represented as surfaces with higher magnitude while dark pixels are represented as surface having lower magnitude. Later, the "ridges" or segmented lines are found by searching the "catchment basins" or the minima on the topographical landscape. This technique was used in combination with the mean shift clustering and thresholding to extract the fruits images captured through Raspberry Pi camera system inside the lighting box. Since multiple fruits were placed at a time inside the lighting box for grading, these individual fruits were first segmented by applying watershed segmentation and computing the distance mapping using the Euclidean Distance Transform (EDT). To define the local maxima (peaks), we set the minimum distance of at least 10 pixel between the foreground and background on the distance map. Later, we applied connected-component analysis to segment unique objects that are connected as shown in Fig. 11.

2.3.4. *Data augmentation*

Data augmentation was used to increase the generalization of the image classification model [21,22]. During the data augmentation phase, the images are usually either randomly translated by a few pixels or flipped horizontally. In this study, we adopted the optimal augmentation policy similar to the ImageNet [23] to our data sets. The optimal augmentation policy mostly performs color based operations such as Solarize, Posterize, Equalize, Color and AutoContrast [24].

2.3.5. *Deep learning models and training of networks*

2.3.5.1. *Network architectures*. In order to develop the fruits' grading model, the data sets were partitioned into training, validation and testing (80%, 10% and 10%). For apples' data set, we trained the models from scratch using ResNet50 [22], DenseNet121 [25], NASNet [26] and EfficientNet [21] architectures. The weights for these networks were initialized using the technique introduced by He et al. [27]. The images were resized to 224 × 224 pixels for ResNet50, DenseNet121, NASNet and the resolution was up-scaled for variants of EfficientNet B0-B2. Due to the small size of bananas' data set, transfer learning was performed using the pre-trained weights from the networks for apples data set (Fig. 12). The experimenting was done with two prominent transfer learning techniques namely fine-tuning new layers added to base networks and using the intermediate layers features as embedding to train a linear classifier [28]. To further improve the accuracy of the models, ensemble methods were used by

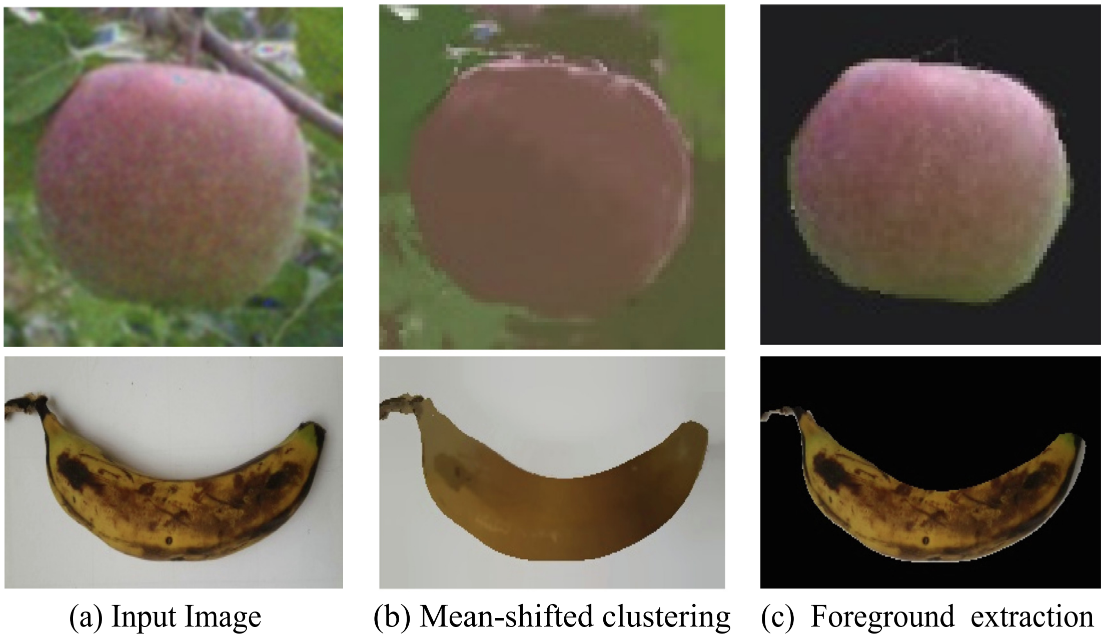

Fig. 10 – Segmentation of the region of interest.

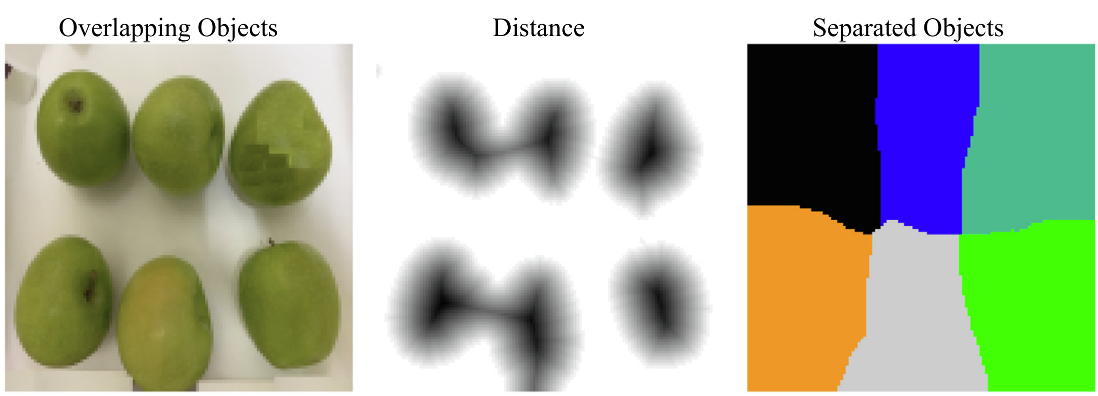

Fig. 11 – Watershed segmentation of multiple fruits inside the lighting box - middle image shows the Euclidean distance transform mapping between the ROI (apples) and the background, right image shows the segmented area of setting the pixel distance of 10 of each peak (local maxima).

stacking combinations of network architectures and applying Global average pooling layer to the stacked feature embedding before the final *softmax* layer for apples' data set and linear classifier for bananas' data set.

2.3.5.2. *Learning rate finder.* Following the super convergence approach for neural networks, learning rate range test (LR range test) was performed to find the optimal LR_{min} and LR_{max} to be used in Cyclical Learning rate (CLR) [29,30]. Losses for each mini-batches were plotted with respect to the LR range of $10^{-10} < LR < 1$, increasing exponentially on each mini-batches. LR range test serves as a guide on how well a network performs through a range of learning rates and to spot values that are effective to train the network. Stopping criteria for LR range test was to set at a condition when current loss is 4 times greater than the previous loss as seen in Fig. 13 at $LR > 10^{-2.5}$ regime. The "triangular" learning rate policy rule was opted for this study where learning rates oscillates in a triangular cycle from LR_{min} to LR_{max} in equal length of steps size.

2.3.5.3. *Hyperparameter tuning.* The Bayesian Optimization technique was used to find the optimal hyperparameters. Bayesian Optimization incorporates the Bayes theorem to solve a non-convex and global optimization problems such as the deep neural networks. We used Gaussian processes as our surrogate function and Expected Improvement (EI) algorithm was used to select the most promising hyperparameter settings as acquisition function. The AdamW and Stochastic Gradient Descent (SGD) were used as optimizer functions and the momentum and weight decay were tuned with these optimizer functions. The optimal value for MobileNetV2 was searched between 0.2 and 1.0 to control the depth of the network.

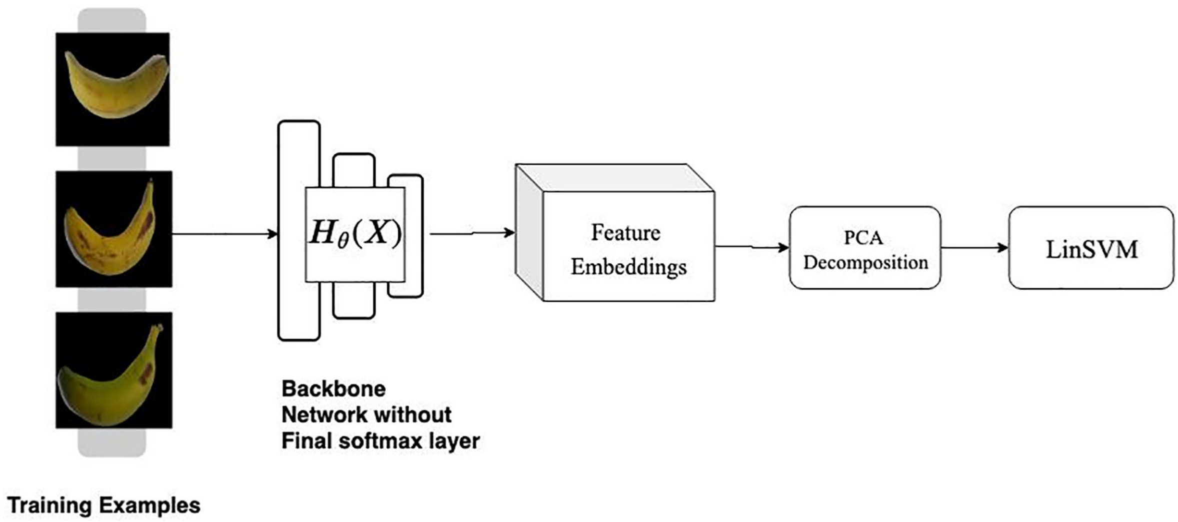

Fig. 12 – Transfer learning training pipeline for bananas' data set.

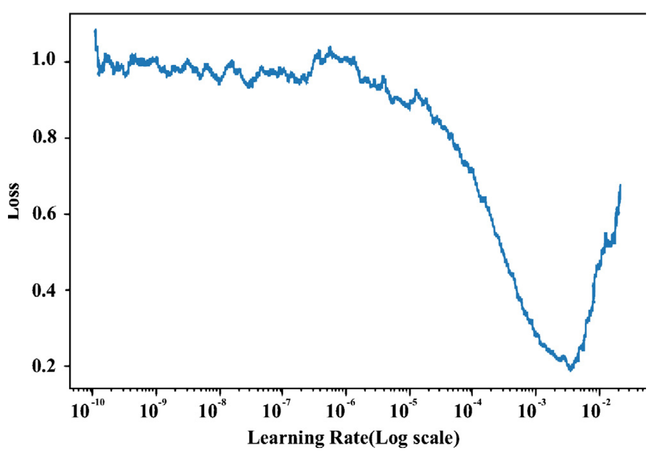

Fig. 13 – Exponential weighted moving averages plot of LR range test curve for EfficientNet trained on apples data set, here the optimal learning rate would be in-between 10^{-5} and 10^{-3} with the steepest descent on loss.

3. Results and discussion

The trained models were evaluated in two phases: initially the models were tested off-line on a hold-out test set (10% of the images from each data set) from both data sets and in the next phase the models were deployed to the real-time system where their performance was assessed in grading the actual assorted apples and bananas varying in species, colors and shapes. 10-fold cross-validation was used for evaluating the linear classifiers in the output layer and an average of 5 runs were executed for deep networks for apples data set due to resource intensive nature of the cross validation technique. We selected specificity (True Positive Rate), sensitivity (True Negative Rate), accuracy and Area under ROC Curve (AUC) metrics to evaluate our models and computed the average for all class labels presented in the target dataset. Sensitivity is defined as the ability of the prediction model to select the instance of a certain class from the dataset. It is the proportion of the actual positive classes which are correctly identified. On the other hand, specificity is defined as the proportion of the predicted negative classes which are correctly identified. The equations for these metrics are given below:

$$Sensitivity = \frac{\sum_{i=1}^{k} TP_i}{\sum_{i=1}^{k} (TP_i + FN_i)} \tag{3}$$

$$Specificity = \frac{\sum_{i=1}^{k} TN_i}{\sum_{i=1}^{k} (TN_i + FP_i)} \tag{4}$$

$$Accuracy = \frac{\sum_{i=1}^{k} \frac{TP_i + TN_i}{TP_i + TN_i + FP_i + FN_i}}{k} \tag{5}$$

where k represents total class labels. TP, TN, FP, and FN are the numbers of the true positives, true negatives, false positives and false negatives predictions for the considered class, respectively.

Using the above metrics, the comparison of the performance of different networks on hold-out test set of apples' and bananas' data sets is shown in Table 1 and Table 2, respectively. It can be observed from Table 1 that the best model to correctly classify the healthy and defected apples with spots on the surface is the EfficientNet-B2 (99.2% recognition rate). Table 2 shows a remarkable performance of models trained on banana dataset using transfer learning for different architectures namely DenseNet, NasNet and EfficientNet. Overall the model achieves satisfying accuracy in classifying the test examples from the hold out set into their ripeness stage. This is due to the prior features learnt by the models during training on apples datasets which greatly helped in features learning for the bananas' dataset using the transfer learning approach. Figs. 14 and 15 show the AUC bar plots for apples' and bananas' data sets, respectively which depict that the trained EfficientNet models have a high capability in distinguishing the different classes. Moreover, Fig. 16 shows an example of Class Activation Map (CAM) for EfficientNet at the final Convolution layer for apples' data set, depicting the discriminative region used by the network

Table 1 – Performance comparison of different networks for apples' test set.

Model	Specificity (%)	Sensitivity (%)	Accuracy (%)
ResNet-50 [22]	97.2 (±0.12)	97.2 (±0.12)	96.9 (±0.22)
MobileNetV2 [31]	98.0 (±0.22)	98.6 (±0.12)	98.5 (±0.22)
DenseNet-121 [25]	98.0 (±0.12)	98.6 (±0.12)	98.5 (±0.42)
NASNet-A [26]	98.0 (±0.12)	97.0 (±0.12)	98.2 (±0.12)
EfficientNet-B0 [21]	98.5 (±0.12)	99.1 (±0.12)	98.6 (±0.12)
EfficientNet-B1 [21]	98.7 (±0.12)	99.2 (±0.12)	98.7 (±0.12)
EfficientNet-B2 [21]	99.0 (±0.12)	100.0 (±0.22)	99.2 (±0.12)

Table 2 – Performance comparison of different networks for bananas' test set.

Model	Specificity (%)	Sensitivity (%)	Top-1 Accuracy (%)	Top-3 Accuracy (%)
ResNet-50 [22]	87.0 (±0.42)	86.2 (±0.42)	91.2 (±0.42)	98.2 (±0.42)
MobileNetV2 [31]	87.0 (±0.42)	86.2 (±0.42)	93.2 (±0.42)	98.2 (±0.42)
DenseNet-121 [25]	98.0 (±0.42)	98.6 (±0.42)	96.2 (±0.42)	98.6 (±0.42)
NASNet-A [26]	98.0 (±0.42)	97.1 (±0.42)	96.2 (±0.42)	98.6 (±0.42)
EfficientNet-B0 [21]	98.5 (±0.42)	97.1 (±0.42)	96.2 (±0.42)	99.0 (±0.42)
EfficientNet-B1 [21]	99.5 (±0.42)	97.1 (±0.42)	98.1 (±0.42)	99.0 (±0.42)
EfficientNet-B2 [21]	98.5 (±0.42)	98.1 (±0.42)	98.6 (±0.42)	99.0 (±0.42)

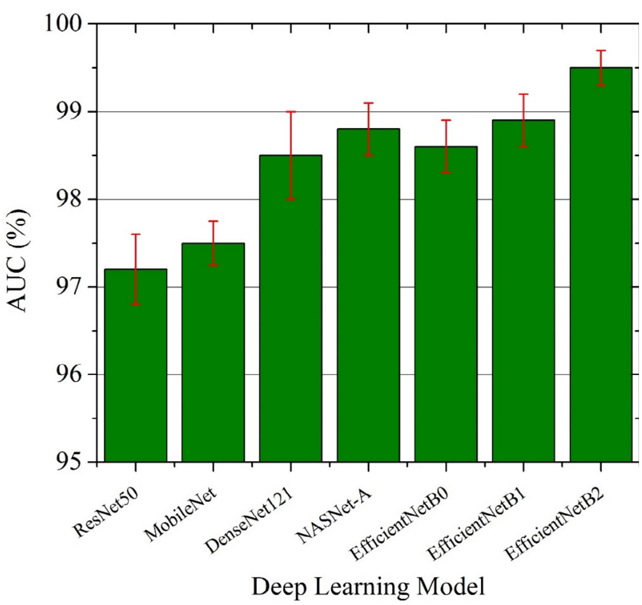

Fig. 14 – AUC bar plot for apples' test set.

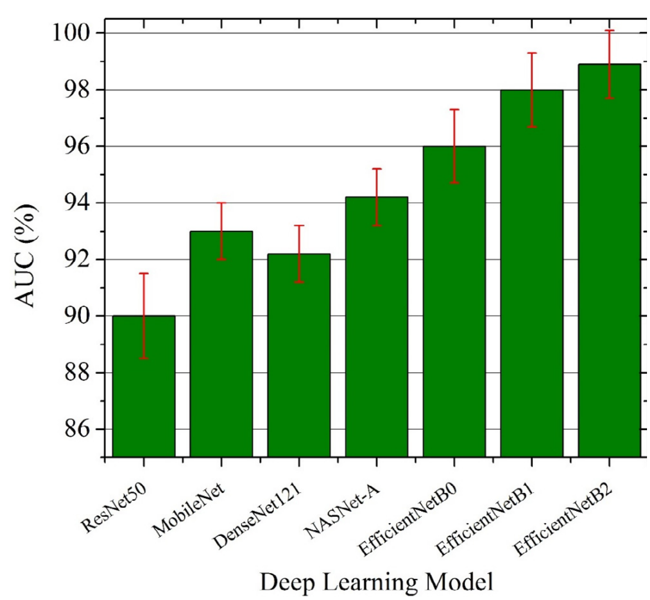

Fig. 15 – AUC bar plot for bananas' test set.

to classify the apples. The trained model exhibits no confusion in detecting defects with the stem end and calyx regions. The proposed end-to-end solution offers more robust approach compared to [6] and have fully emphasized to the regions where defects are present on the surface. The performance of the models was further improved by applying ensemble methods for both datasets which can be seen by stacking multiple features embedding of EfficientNetB0 + B1 + B2 with 99.5% and 98.9% recognition rate for apples' and bananas' dataset as shown in Table 3 and Table 4, respectively.

In order to further evaluate our trained models in real-time, we used a collection of 30 apples (consisting of variety of species of 15 healthy and 15 bruised ones having dark spots i.e. defected apples) and a collection of 32 bananas with equal distribution of assorted ripeness colors and shapes. All of the real-time testing was done using our proposed system as described Section 2. Figs. 17–19 show the performance evaluation for healthy and defected apples when tested using the trained deep learning models. The recognition rate of the state-of-the-art image classification models was found below 90% during the real-time evaluation (Fig. 19). While the performance of the EfficientNet (a recently developed deep learning network) and its variants was observed better than the performance of the previous state-of-the-art models during

Fig. 16 – Class Activation Map (CAM) for EfficientNet at the final Convolution layer for apples' data set.

Table 3 – Performance comparison of ensemble stacked networks for apples' test set.

Model	Specificity (%)	Sensitivity (%)	Accuracy (%)
ResNet50 + DenseNet-121 + MobileNetV2	98.2 (±0.22)	98.5 (±0.22)	97.2 (±0.22)
ResNet50 + DenseNet-121 + NASNet-A	98.0 (±0.22)	98.9 (±0.22)	98.5 (±0.23)
DenseNet-121 + NASNet-A + EfficientNetB0	98.2 (±0.22)	98.9 (±0.22)	98.6 (±0.23)
EfficientNetB0 + B1 + DenseNet-121	98.2 (±0.22)	99.0 (±0.22)	98.9 (±0.23)
EfficientNetB0 + B1 + B2	99.2 (±0.22)	99.6 (±0.22)	99.5 (±0.23)

Table 4 – Performance comparison of ensemble stacked networks for bananas' test set.

Model	Specificity (%)	Sensitivity (%)	Top-1 Accuracy (%)	Top-3 Accuracy (%)
ResNet50 + DenseNet-121 + MobileNetV2	87.0 (±0.42)	86.2 (±0.42)	93.2 (±0.42)	98.2 (±0.42)
ResNet50 + DenseNet-121 + NASNet-A	97.0 (±0.42)	98.2 (±0.42)	96.2 (±0.42)	98.2 (±0.42)
DenseNet-121 + NASNet-A + EfficientNetB0	98.0 (±0.42)	98.6 (±0.42)	97.2 (±0.42)	98.6 (±0.42)
EfficientNetB0 + B1 + DenseNet-121	99.5 (±0.42)	97.1 (±0.42)	98.1 (±0.42)	99,0 (±0.42)
EfficientNetB0 + B1 + B2	99.1 (±0.42)	98.5 (±0.42)	98.9 (±0.42)	99.2 (±0.42)

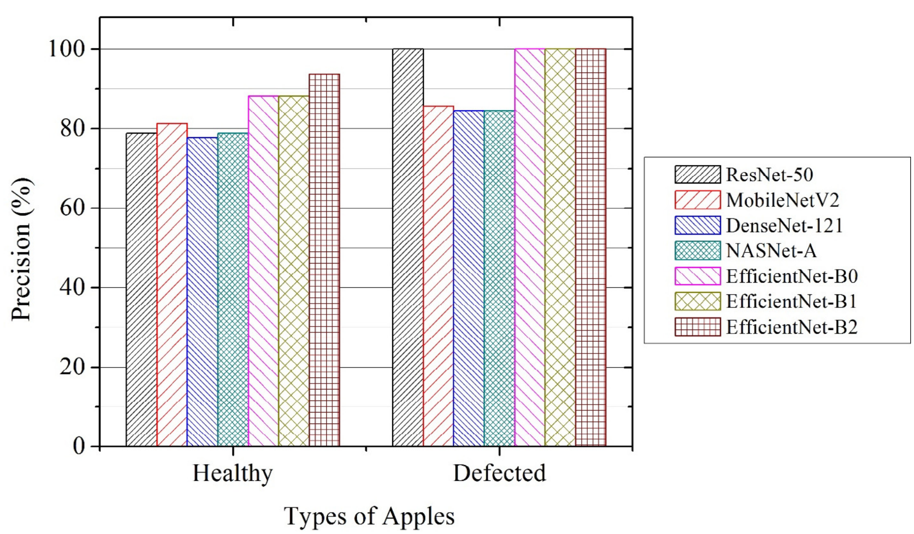

Fig. 17 – Real-time precision evaluation of various deep learning models for 30 test samples of apples.

real-time testing with a highest accuracy rate of 96.7% for EfficientNetB2 (Fig. 19).

The results presented in Fig. 18 indicate that most of the models are having a sensitivity of 100% for healthy class with the exception of MobileNetV2 and DenseNet121. Hence, the classifiers are able to correctly distinguish healthy apples from the defected classes. With the lack of variability in the data set where defected apples are only of the type CASC IFW apples, the generalization of the classifiers to distinguish defected apples were fairly low despite being trained on newer model and this is shown by the sensitivity value of 86.7% for EfficientNetB0, EfficientNetB1 and 93.3% for EfficientNetB2 as demonstrated in Fig. 18.

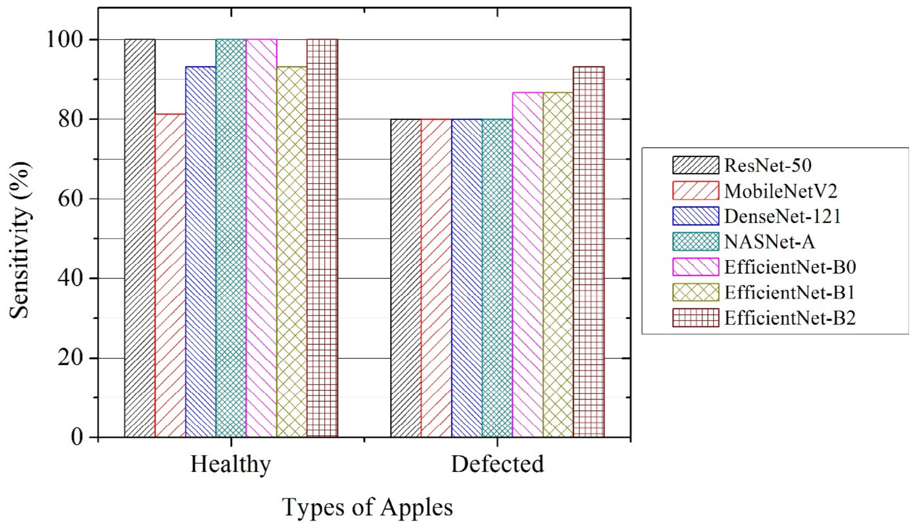

Fig. 18 – Real-time sensitivity evaluation of various deep learning models for 30 test samples of apples.

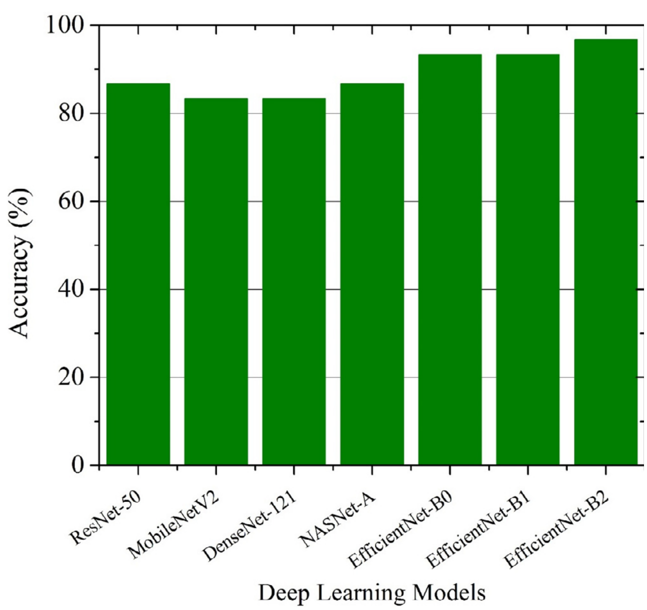

Fig. 19 – Real-time accuracy performance of various deep learning models for 30 test samples of apples.

Further, the trained deep learning models were applied to the collection of banana fruits and the performances of these models were compared for real-time testing as shown in Fig. 20 to Fig. 22. The recognition rate during the real-time testing was found around 80% for most of the classifiers with highest value of 93.8% for EfficientNetB2 (Fig. 22).

The results shown in the above figures indicate that most of the classifiers have a tendency to make confusion between yellowish green and the green class due to the subtle differences in the color. This pattern can be noted from the sensitivity values for MobileNetV2 and DenseNet121 in Fig. 21 (a sensitivity value of 62.5%). The highest sensitivity value for yellowish green class was found to be 87.5% for EfficientNet and its variants as shown on Fig. 21. Green and overripe classes were found to have high recognition rate due to their similarity with features from the CASC IFW apple data set where apples were mostly green, and the defected apples were mostly covered with dark spots. Due to the transfer learning, this similarity in features helped in detecting the ripeness of banana fruits more accurately for these two classes. Based on the overall experimented results, EfficientNet shows a satisfying accuracy for both CASC IFW apple and banana data sets. However, the performance of the best models also differ for offline test set and actual real-time testing samples due to the different data collection environment. Such performance difference is expected in computer vision based solutions because of the lighting effects.

4. Conclusions and future work

In this paper we proposed a deep learning based low-cost machine vision system for grading the fruits based on their outer appearance or freshness. Various state-of-the-art deep learning models and stacking ensemble deep learning methods were applied to two data sets of fruits. The results of this study show that EfficientNet CNN models and their stacked combinations have the highest accuracy in grading the test set and real samples as compared to the other deep learning models. Moreover, the application of deep learning models has been found more accurate in classification of fruits as compared to the results previously reported while applying traditional machine learning techniques using the feature extraction methods. However, there are certain limitations of the current study which require further investigations. In our proposed system, classification is solely based on the outer appearance of the fruit and it only uses single view of the fruits' images. In future, the system will be trained and tested using multi-view vision system for capturing image data sets. In addition, for real-time testing of the system, the lighting box will be setup over a conveyor

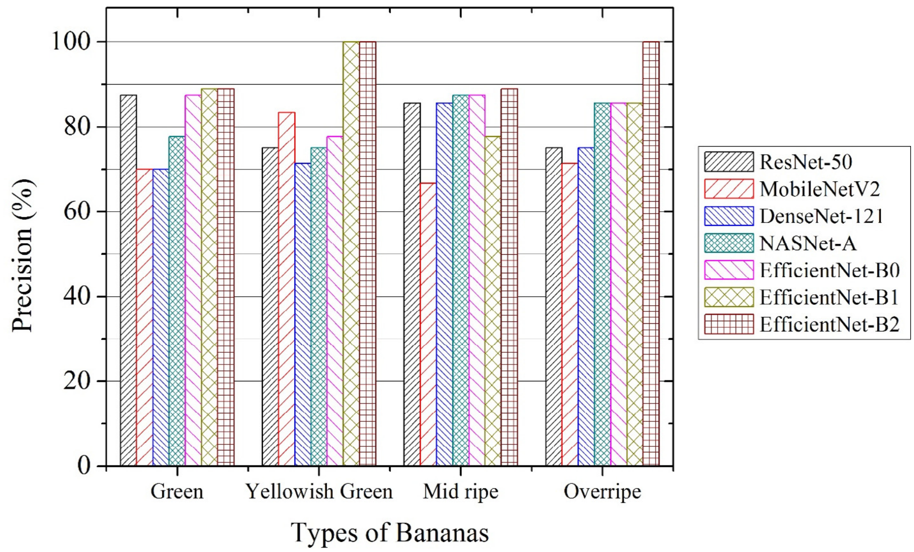

Fig. 20 – Real-time precision evaluation of various deep learning models for 32 test samples of bananas.

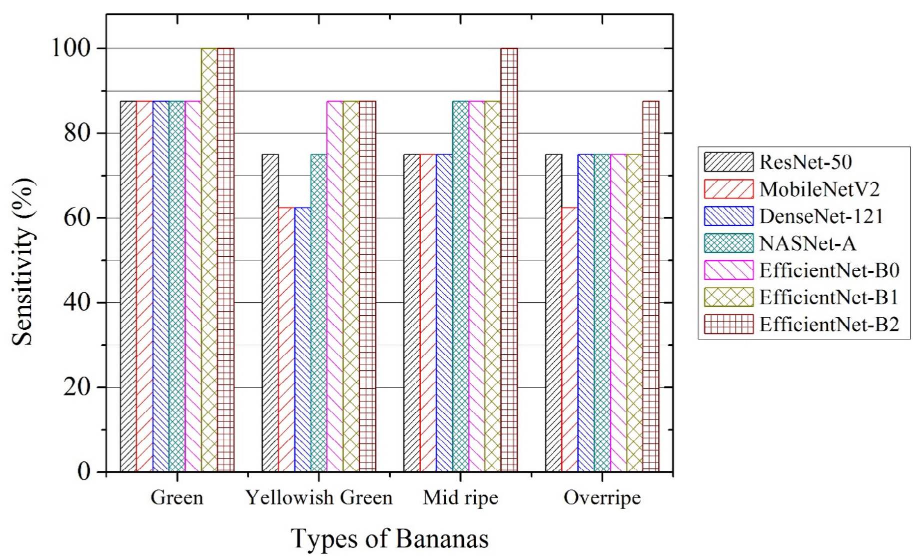

Fig. 21 – Real-time sensitivity evaluation of various deep learning models for 32 test samples of bananas.

belt with an environment where multiple camera systems will be installed to capture the images of the fruits from multi-dimensions.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

We would like to express our gratitude to Dr. Tanmay K. Prakash at Purdue University, who has granted us access to CASC IFW database.

REFERENCES

[1] Tripathi MK, Maktedar DD. A role of computer vision in fruits and vegetables among various horticulture products of agriculture fields: a survey. Inf Process Agric 2020;7:183–203.

[2] Yan B, Wu XH, Ye B, Zhang YW. Three-level supply chain coordination of fresh agricultural products in the Internet of Things. Ind Manag Data Syst 2017;117:1842–65.

[3] Nukala R, Panduru K, Shields A, Riordan D, Doody P, Walsh J. Internet of Things: a review from Farm to Fork. In: Proc. Proceedings of 27th Irish Signals and Systems Conference. Londonderry, UK; 2016. p. 1–6.

[4] Dubey SR, Jalal AS. Apple disease classification using color, texture and shape features from images. Signal, Image Video Process 2016;10:819–26.

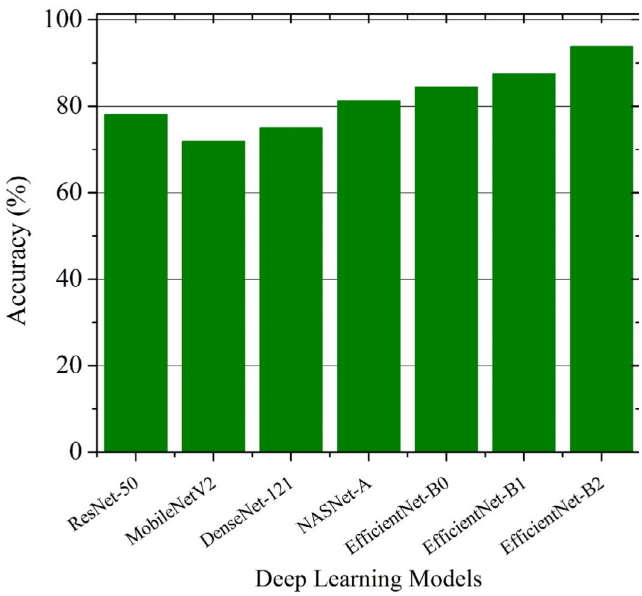

Fig. 22 – Real-time accuracy performance of various deep learning models for 32 test samples of bananas.

[5] Moallem P, Serajoddin A, Pourghassem H. Computer vision-based apple grading for golden delicious apples based on surface features. Inf Process Agric 2017;4:33–40.

[6] Mazen FMA, Nashat AA. Ripeness classification of bananas using an artificial neural network. Arab J Sci Eng 2019;44:6901–10.

[7] Tan K, Lee WS, Gan H, Wang S. Recognising blueberry fruit of different maturity using histogram oriented gradients and colour features in outdoor scenes. Biosyst Eng 2018;176:59–72.

[8] Silalahi DD, Reaño CE, Lansigan FP, Panopio RG, Bantayan NC. Using genetic algorithm neural network on near infrared spectral data for ripeness grading of oil palm (Elaeis guineensis Jacq.) fresh fruit. Inf Process Agric 2016;3:252–61.

[9] Sabzi S, Abbaspour-Gilandeh Y, García-Mateos G. A new approach for visual identification of orange varieties using neural networks and metaheuristic algorithms. Inf Process Agric 2018;5:162–72.

[10] Pande A, Munot M, Sreeemathy R, Bakare R V. An efficient approach to fruit classification and grading using deep convolutional neural network. In: Proc. Proceedings of IEEE 5th international conference for convergence in technology. Bombay, India; 2019. p. 1–7.

[11] Rojas-Aranda JL, Nunez-Varela JI, Cuevas-Tello JC, Rangel-Ramirez G. Fruit Classification for retail stores using deep learning. In: Proc. proceedings of pattern recognition 12th mexican conference. Morelia, Mexico; 2020. p. 3–13.

[12] Hossain MS, Al-Hammadi M, Muhammad G. Automatic fruit classification using deep learning for industrial applications. IEEE Trans Ind Informatics 2019;15:1027–34.

[13] Tian Y, Yang G, Wang Z, Li E, Liang Z. Detection of apple lesions in orchards based on deep learning methods of CycleGAN and YOLOV3-dense. J Sens 2019;2019:7630926.

[14] Hu Z, Tang J, Zhang P, Jiang J. Deep learning for the identification of bruised apples by fusing 3D deep features for apple grading systems. Mech Syst Signal Process 2020;145 106922.

[15] Ucat RC, Cruz JCD. Postharvest grading classification of Cavendish banana using deep learning and tensorflow. In: Proc. Proceedings of international symposium on multimedia and communication technology. Quezon City, Philippines; 2019. p. 1–6.

[16] Behera SK, Rath AK, Sethy PK. Maturity status classification of papaya fruits based on machine learning and transfer learning approach. Inf Process Agric 2020. In Press.

[17] Guiqin Li, German Holguin, Johnny Park, Brian Lehman, Larry Hull VJ. The CASC IFW Database; 2009.

[18] Pizer SM, Amburn EP, Austin JD, Cromartie R, Geselowitz A, Greer T, et al. Adaptive histogram equalization and its variations. Comput Vision, Graph Image Process 1987;39:355–68.

[19] Cheng Y. Mean shift, mode seeking, and clustering. IEEE Trans Pattern Anal Mach Intell 1995;17:790–9.

[20] Vincent L, Soille P. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. IEEE Trans Pattern Anal Mach Intell 1991;13:583–98.

[21] Tan M, Le Q V. EfficientNet: Rethinking model scaling for convolutional neural networks. In: Proc. proceedings of 36th international conference on machine learning. Long Beach CA, United States; 2019. p. 10691–700.

[22] He K, Zhang X, Ren S, Sun J. Deep Residual Learning for Image Recognition. In: Proc. proceedings of IEEE conference on computer vision and pattern recognition. Nevada, United States; 2016. p. 770–8.

[23] Deng J, Dong W, Socher R, Li L, Li K, Fei-Fei L. ImageNet: A large-scale hierarchical image database. In: Proc. proceedings of IEEE conference on computer vision and pattern recognition. Miami, FL, United Sates; 2009. p. 248–55.

[24] Cubuk ED, Zoph B, Mane D, Vasudevan V, Le Q V. Autoaugment: Learning augmentation strategies from data. In: Proc. proceedings of IEEE conference on computer vision and pattern recognition. California, United States; 2019. p. 113–23.

[25] Huang G, Liu Z, Maaten L Van Der, Weinberger KQ. Densely connected convolutional networks. In: Proc. proceedings of IEEE conference on computer vision and pattern recognition. Honolulu, United States; 2017. p. 2261–9.

[26] Zoph B, Vasudevan V, Shlens J, Le Q V. Learning transferable architectures for scalable image recognition. In: Proc. proceedings of IEEE conference on computer vision and pattern recognition, Utah, United States; 2018. p. 8697–710.

[27] He K, Zhang X, Ren S, Sun J. Delving deep into rectifiers: surpassing human-level performance on ImageNet classification. In: Proc. proceedings of IEEE international conference on computer vision. Santiago, Chile; 2015. p. 1026–34.

[28] Praveen Kumar J, Domnic S. Image based leaf segmentation and counting in rosette plants. Inf Process Agric 2019;6:233–46.

[29] Smith LN, Topin N. Super-convergence: very fast training of neural networks using large learning rates. Link: https://arxiv.org/abs/1708.07120. May 2018 / Nov. 2020.

[30] Smith LN. Cyclical learning rates for training neural networks. In: Proc. IEEE winter conference on applications of computer vision. Santa Rosa, United States; 2017. p. 464–72.

[31] Sandler M, Howard A, Zhu M, Zhmoginov A, Chen L. MobileNetV2: inverted residuals and linear bottlenecks. In: Proc. proceedings of 2018 IEEE conference on computer vision and pattern recognition. Utah, United States; 2018. p. 4510–20.