




Fruit Classification Using Deep Convolutional Neural Network and Transfer Learning

Rachna Verma¹  and Arvind Kumar Verma²

¹ Department of CSE, Faculty of Engineering, J.N.V. University, Jodhpur, Rajasthan, India

rachnaverma@jnvu.edu.in

² Department of PI, Faculty of Engineering, J.N.V. University, Jodhpur, Rajasthan, India

akverma.pi@jnvu.edu.in

Abstract. Identification and classification of fruits of different qualities are vital for fruit industries. Traditional techniques, such as visual inspection of fruits by handpicking, are time-consuming, tiresome, and error-prone. To automate the fruit inspection process and segregating them into different classes, computer vision and machine learning approaches has been applied and researched. For segregating fruits into different classes or qualities, Transfer Learning is one of the popular technique to build a fruit classifier. This paper evaluates the performance of VGG16, InceptionV3, Xception, ResNet152V2, and DenseNet by training and testing models using transfer learning. The experiment is conducted on two fruits datasets. Further, to improve the accuracy of fruit classification, the pre-trained model DenseNet is partially unfreezed and retrained. The results show that this model archives accuracy of 99.61% for fruit classification.

Keywords: CNN · Deep learning · Transfer learning · Fruits classification

1 Introduction

Indian agriculture industries play a significant role in the economy of India. India is primarily an agriculture-based economy and has tremendous opportunities in exporting fruits and vegetables. As per APEDA (Agricultural and Processed Food Products Export Development Authority), Ministry of Commerce and Industry, India [1], India stands second in producing fruits and vegetables globally, after China. During 2020–2021, India exported 956961.00 metric tons of fruits, which were worth Rs 5647crores. Fruits such as pomegranates, bananas, oranges, and mangoes are exported in large amounts from India to different countries.

Farmers produce fruits and vegetables and sell them either in local markets or to fruit industries. The fruit industries process them to segregate into different grades (quality of fruits) before packaging for export. The high quality of fruit is the primary requirement for exporting. Hence, accurately and efficiently selecting high-quality fruits is vital for the fruit industry.

Fruits sorting in major fruits industries in India is mainly done by handpicking and inspecting fruits manually, which is time-consuming, tiresome, and error-prone. Further,

rotten fruits need to be sorted and removed immediately to avoid spoiling other fruits. The quality of fruit is decided by many parameters, such as ripe, unripe, and rotten. Thus, there is a need to automate the fruit sorting process to reduce labor costs and accurately sort fruits into different qualities.

In literature, some works are reported to automate the fruit classification problem using various machine learning and computer vision techniques. The major steps used in fruits and vegetables classification are data acquisition, pre-processing, feature extraction, and classification [2]. Images are the primary input data in the fruit classification system, acquired using the camera, ultrasound, MRI, infrared, Lidar [2, 3]. The digital images of various fruits and vegetables of different qualities and grades are collected to create a training dataset.

The images acquired through sensors contain noises and distortions. In the pre-processing step, the noises and distortions are removed or minimized before the feature extraction process. Further, pre-processing enhances the image data and various features of the images, which is essential for obtaining discriminative features for classification. Sometimes segmentation is required as a pre-processing step that separates the foreground object of interest from the background. Segmentation techniques [3], such as thresholding, clustering are primarily used for segmentation. An improper segmentation degrades the classifier's performance.

After pre-processing, features such as texture and color, are extracted for further processing as these features play a vital role in discriminating one object from the other. These extracted features are used to train a fruit classifier. Various machine learning algorithms, such as KNN, SVM, ANN, deep learning networks, have been used in the literature [3–5] to design fruit quality classification algorithms.

Development in deep Convolutional Neural Networks (CNN) has proven to be very effective in image classification tasks [6, 7]. CNN is trained to identify and classify objects. Building and training a CNN from scratch requires a huge amount of data and time. Hence, transfer learning is used. Transfer learning is a way to transfer knowledge from a similar domain to a specific domain. It is used when the training data is limited [8]. In literature, transfer learning is also used fruit classification [9, 10]. Therefore, this work proposes a fruit classifier to automate the visual inspection of some Indian fruits using transfer learning. The major contributions are:

- i) Performance evaluation of various popular pre-trained networks for fruit image identification and classification.
- ii) Fine-tuning of Desenet121 by partially unfreezing a few layers of the convolutional network.

2 Related Work

Many research works on fruit and vegetable classification and their quality grading, based on color, texture and shape, have been reported using machine learning-based computer vision [2–5]. Four different machine learning classifiers, KNN, SVM, SRC, and ANN, have been used in [4] to classify fruits. A classification technique based on KNN, SVM, linear discriminates, and regression classifier is used by Singh and Singh

[11] to classify healthy and rotten apples. SVM classifier is used by Moallem et al. [12] to detect defects in apples. They used statistical, texture, and geometrical features and achieved 92.50% and 89.20% accuracy for healthy and defective apples. Long and Thinh [13] used external features, such as length, width, weight, and defect, to classify mango into three different grades: good, medium, and bad. The classifiers used by [13] are Random Forest, Linear discriminant analysis, KNN, and SVM. The random forest has outperformed with a precision of 98.1% compared with the other three models.

Developments in the deep convolutional neural network is found effective in image identification and classification tasks [6, 7]. In [14], authors used two models - a six layered CNN model and a customized VGG16 model for automatic fruit classification. A nine-layer CNN is proposed by [15] for fruit classification in uncertainty conditions. They used YOLO V3 to generate a bounding box around the apple in the original images. They reported an average accuracy level of 99.73%.

Full training of a deep CNN is very expensive in terms of time and resource. Hence, many researchers used the transfer learning approach, which uses weights of pre-trained generic models to build new specific models. Transfer learning is widely used in various computer vision tasks and helps efficiently fine-tune a generic pre-trained network for a specific purpose. Further, it has been found that transfer learning effectively transfers learned knowledge from the general domain to a specific domain even when data is limited. It not only saves training time but also reduces generalization error. Hence, many researchers in fruit and vegetable classification and grading have used transfer learning.

Zilong et al. [16] have investigated the effectiveness of various configurations of CNN to detect damaged apples. They also explored different fusion strategies and achieved an accuracy of 97.67%. They used VGG-19 and Inception-v3 CNN architecture for feature extraction. Vishal et al. [9] used DenseNet161, InceptionV3, and MobilenetV2 architectures to study the misclassification problem. Further, they proposed MNet architecture based on Inception-v3 to classify fruits into different classes and achieved an accuracy of 99.92%.

A real-time automatic visual inspection system for grading of apple and banana using pre-trained CNN architectures - ResNet, DenseNet, MobileNet, NASNet, and EfficientNet is studied by Nazrul and Malik [10]. They found EfficientNet to be the best model for fruit grading and achieved an accuracy of 99.2% and 98.6% on apple and banana test sets, respectively. Shih-Lun et al. [17] also used pre-trained networks AlexNet, VGG, and ResNet to classify Mangoes of different grades. A CNN architecture based on MobileNetV2 for the classification of fruits inside a plastic bag is proposed by Rojas Aranda et al. [18]. Besides fruit images, they also input additional features, such as color and centroid, to improve accuracy.

From the above, it can be concluded that fruit and vegetable classification and grading is an emerging area of research. There are many fruits and vegetables for which classifiers have to be developed. Further, there are many generic networks whose efficacy for fruit and vegetable classification has to be evaluated. In this paper, we have evaluated the performances of various networks.

3 Methodology

This work evaluates the performance of generic VGG16 [22], InceptionV3 [23], Xception [24], DenseNet [25] and ResNet152V2 [26] networks for fruit identification and classification. Further, we have attempted fine-tuning of Desenet121 by partially unfreezing a few layers of the convolutional network to enhance the accuracy of the fruit classifier, using two fruit image datasets [19, 20]. This section briefly describes the datasets, CNN networks, and transfer learning concepts used in the paper.

3.1 Datasets

In this paper, we have used two different fruit datasets for training fruit classifiers. The first dataset (dataset-1) is fruits fresh and rotten for classification [19]. It consists of three types of fruits - Apple, Banana, and Oranges, having 13599 healthy and rotten images; Of the total images, 7308, 2698 and 3593 images are used for training, testing and validation. The dataset consists of six classes: Fresh Apples, Bananas, and Oranges; and Rotten Apples, Bananas, and Oranges. The image sizes in dataset-1 are variable. Hence, they need to be properly resized before using them in a network. We have used resized images to $150 \times 150 \times 3$ in this paper.

The second dataset (dataset-2) consists of images of six types of Indian Fruits: Apples, Bananas, Guavas, Limes, Oranges, and Pomegranates. The dataset consists of 12000 healthy and rotten images. The dataset is available at [20]. The dataset is divided into twelve classes, with each class having 1000 images. The twelve fruit classes in the dataset are Fresh Apples, Fresh Bananas, Fresh Guavas, Fresh Limes, Fresh Oranges, Fresh Pomegranates, Rotten Apples, Rotten Bananas, Rotten Guavas, Rotten Limes, Rotten Oranges and Rotten Pomegranates. The images are colored RGB images of sizes $256 \times 256 \times 3$. To build the classifier, dataset-2 is partitioned into training, testing and validation in 70%, 15%, and 15%, respectively.

Pre-trained networks used in this study require all input images to be of some specific sizes during their training. For example, the Xception network requires the input image dimensions of size $299 \times 299 \times 3$ for its training. However, these networks accept input image dimensions to fall within a range. For example, the Xception network can take images from $299 \times 299 \times 3$ to $71 \times 71 \times 3$. The number of channels is fixed at 3. Similarly, DenseNet accepts the input image from $224 \times 224 \times 3$ to $32 \times 32 \times 3$. Therefore, we have used input images of size $224 \times 224 \times 3$ for consistent comparisons of results for all networks for dataset-2. Dataset-1 contains images of variable sizes. Therefore, we have used input images of size $150 \times 150 \times 3$ for training, testing, and validation of the classifier.

3.2 Convolution Neural Network

CNN has outperformed traditional technics for various computer vision tasks, including the identification and classification of images. A CNN is an artificial neural network specifically designed to process rectangular image pixel data and has produced superior results than other neural networks for image processing tasks. A typical CNN consists of two parts (see Fig. 1): A convolutional base and a classifier. A Convolutional base

consists of a stack of layers comprised of convolutional and pooling layers. The convolutional base is used to learn and extract various features. The classifier consists of a fully connected layers to classify images. The initial layers, i.e., the head layers of the convolutional base, learn the general features, whereas the features extracted by the tail layers are specific to the chosen dataset and task.

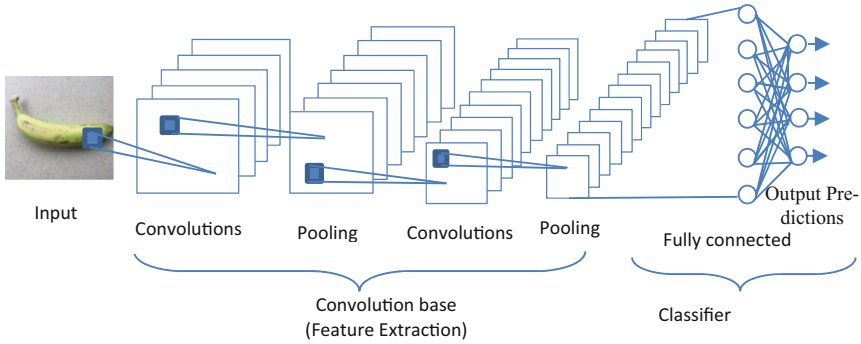


Fig. 1. Typical convolutional neural network architecture

3.3 Transfer Learning

Deep CNN networks consist of many layers in their convolutional base, sometimes more than 100 layers. Transfer learning is a way to build accurate deep CNN models efficiently [7]. Interested readers can refer to [8] for more details. In transfer learning, we use weights and parameters of pre-trained networks that have been trained earlier using generic datasets of similar types. In this way, the network does not have to learn from scratch and require a huge amount of training data. The advantage of transfer learning is that it is more efficient and requires less data than training the same network from scratch.

Various pre-trained networks are explored and compared to develop automatic fruit identification and classification systems [5, 9, 10]. The list of pre-trained models is large, and we have used a limited number of pre-trained networks ranging from classical to modern architectures. All the pre-trained networks used in this paper are trained on the ImageNet dataset of the ImageNet Large Scale Recognition Challenge (ILSVRC) [21] to classify 1000 generic classes, and their pre-trained weights are available. The pre-trained networks used in this paper for fruit classification are VGG16 [22], InceptionV3 [23], Xception [24], DenseNet [25] and ResNet152V2 [26].

3.4 Training of Networks

This paper adopts two strategies to train the above pre-trained networks using transfer learning to identify and classify fruits from dataset-1 and dataset-2. In strategy 1, we have used the frozen CNN (the base network) as a feature extractor, and the network is

fine-tuned by adding a new classifier to the base network. This strategy is implemented by removing the pre-trained network’s original classifier, and a new block of fully-connected layers are added on the top of the existing base network. The new block of fully connected layers consists of a dense layer of 1024 neurons (ReLU activation function), and a dropout layer to avoid overfitting. Finally, a dense softmax output layer of 6 or 12 neurons (depending on the dataset used) is added, (see Fig. 2). The probability of dropout is set to 0.2. We call this model as Model-I. Model-I is fine-tuned by freezing all the layers of the convolutional base network, using dataset-1 and dataset-2. Model-I is trained with both fruits datasets. In this strategy, during transfer learning, the original weights of the convolutional network are preserved.

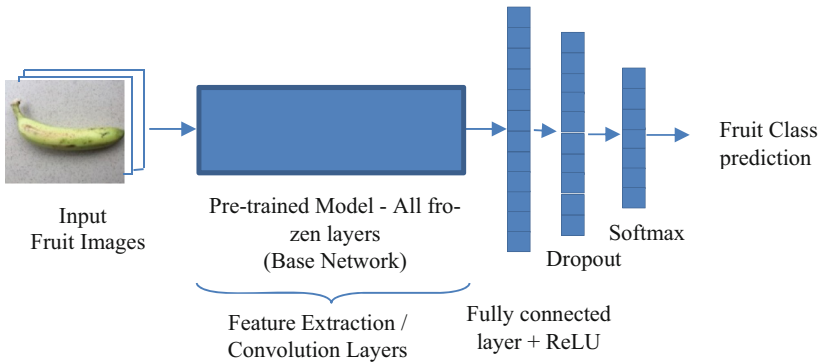


Fig. 2. Block diagram of Model-I

In the second strategy, the classifier remains the same as used in the first strategy. Here, a few layers of the convolutional base network are unfreezed for training (see Fig. 3). The unfreezed layers and a new block of fully-connected layers (shown in blue in Fig. 3) are retrained with dataset-2. We call it Model-II. During retraining, we have to be very careful with the learning rate that controls the weights of the pre-trained network. A small learning rate is preferred for pre-trained networks because a high learning rate may distort the CNN weights too early.

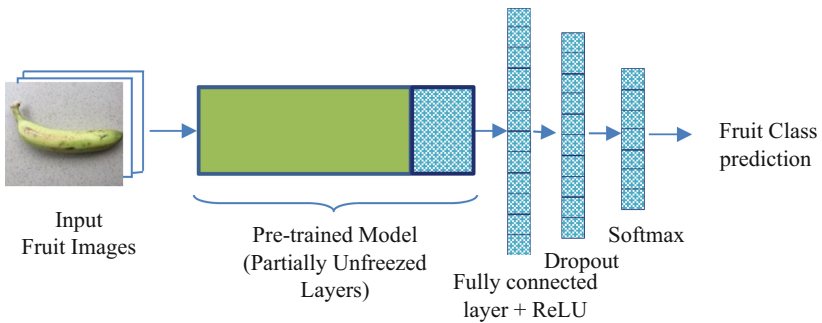


Fig. 3. Block diagram of Model-II

4 Results and Discussion

Our results for VGG16 [22], InceptionV3 [23], Xception [24], DenseNet [25] and ResNet152V2 [26] pre-trained networks (model-I and model-II), which are trained and evaluated with dataset1 and dataset-2, are reported in this section. These networks have been selected because these networks have achieved excellent performance on the Imagenet challenge [21]. All these pre-trained networks are freely available in the Keras API for use.

The experiments are performed using Google Colab with its GPU. We used Keras APIs in Python for building the fruit classifiers. Further, to reduce the training time, we used transfer learning for both datasets. Standard formulas are used to calculate accuracy, precision, recall, and F1-score [6].

For consistency in performance evaluation, the parameters used for both datasets are kept the same. The optimizer used is Adam, and the batch size = 32. The learning rate used for Model-I is $1e-3$. The remaining parameters of the Model-I are kept at their default values provided by the Keras API. The input image size of 150×150 is used for dataset-1 and 224×224 for dataset-2. The values obtained for the various statistical metrics are given in Table 1 for dataset-1 and Table 2 for dataset-2. The tables show that the best model, among the tested models, that correctly classifies the fruits to the correct classes is the DenseNet Model [25]. The accuracy of the DenseNet model for dataset-1 is 99.03% and 99.11% for dataset-2. Figure 4 shows the accuracy performance of various pre-trained networks evaluated using Model-I. In Table 2, we observed that Recall and Accuracy are the same for all pre-trained networks. The equal values for accuracy and recall are because dataset-2 is highly balanced and contains an equal number of images for each class, i.e., 1000 images per fruit class.

Table 1. Performance metrics of pre-trained networks using Model-I for dataset-1

| Model | Accuracy | Precision | Recall | F1-score |
|------------------|----------|-----------|--------|----------|
| VGG16 [22] | 96.62 | 96.75 | 96.50 | 96.59 |
| InceptionV3 [23] | 96.81 | 96.74 | 96.69 | 96.61 |
| Xception [24] | 96.62 | 96.94 | 96.52 | 96.69 |
| ResNet152V2 [26] | 97.14 | 97.13 | 97.08 | 97.10 |
| DenseNet [25] | 99.03 | 99.00 | 98.99 | 98.98 |

Table 2. Performance metrics of pre-trained networks using Model-I for dataset-2

| Model | Accuracy | Precision | Recall | F1-score |
|---------------------------------|----------|-----------|---------|----------|
| VGG16 [22] | 95.22 | 95.53 | 95.22 | 95.23 |
| InceptionV3 [23] | 97.05 | 97.14 | 97.05 | 97.02 |
| Xception [24] | 97.00 | 97.12 | 97.00 | 96.99 |
| ResNet152V2 [26] | 97.61 | 97.73 | 97.61 | 97.62 |
| DenseNet [25] | 99.11 | 99.13 | 99.11 | 99.11 |
| Partial unfreezed DenseNet [25] | 99.6111 | 99.6158 | 99.6111 | 99.6098 |

From Table 1 and Table 2, it can be observed that DenseNet is the best. The confusion matrices obtained for the DenseNet architecture for Model-I are shown in Fig. 5 (for dataset-1) and Fig. 6 (for dataset-2). The test set of dataset-1 contains 2672 images, and dataset 2 contains 1800 images (150 images for each class).

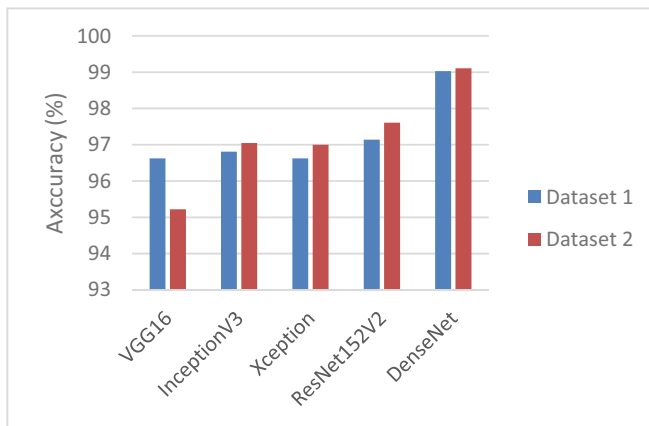


Fig. 4. Accuracy performance of various pre-trained networks using Model-I for dataset-1 and dataset-2.

To further increase the accuracy of the DenseNet model on fruit datasets, it is further evaluated using Model-II. As stated earlier, in Model-II, we partially unfroze the convolutional network of DenseNet and retrained the network (Fig. 3). The input image size for training DenseNet is 224×224 ; hence we trained and tested it only on dataset 2.

The convolutional base of DenseNet is a huge network and consists of five convolutional blocks, and each convolutional block consists of many sub-blocks of convolutional and pooling layers. We experimented with unfreezing a few sub-blocks of convolutional block-5. Convolutional block-5 consists of 16 sub-blocks. We experimented and evaluated the network's performance by unfreezing sub-blocks 16 to 12 of convolutional

block-5. We found that the performance of the network improved till unfreezing sub-block 13 but degraded when further unfreezed to sub-block-12. When we unfreeze a sub-block, all the downline sub-blocks from that block are unfreezed until the network's end. We started by unfreezing one subblock of conv5_block from the bottom at a time for partial retraining and evaluated the network.

We have used a learning rate of $1e-04$ with the optimizer Adam. Further, the early stopping technique is used to avoid overfitting the network. The evaluation metrics are reported in Table 2 (the last row). The accuracy of the DenseNet model evaluated using Model-II for dataset-2 is 99.61%, and its confusion matrix is shown in Fig. 7. From the confusion matrix (Fig. 7), we can see that the number of misclassified fruits is 7 out of 1800 fruits, and the only misclassified fruits are Fresh Apples and Rotten Oranges. All other fruits (fresh and rotten) are 100% correctly classified.

Since the proposed model, Model-II using DenseNet architecture, achieves an accuracy of 99.61%, therefore, DenseNet architecture when compared with other architectures considered in this paper can be applied in fruits industries for classification of fresh and rotten fruits.

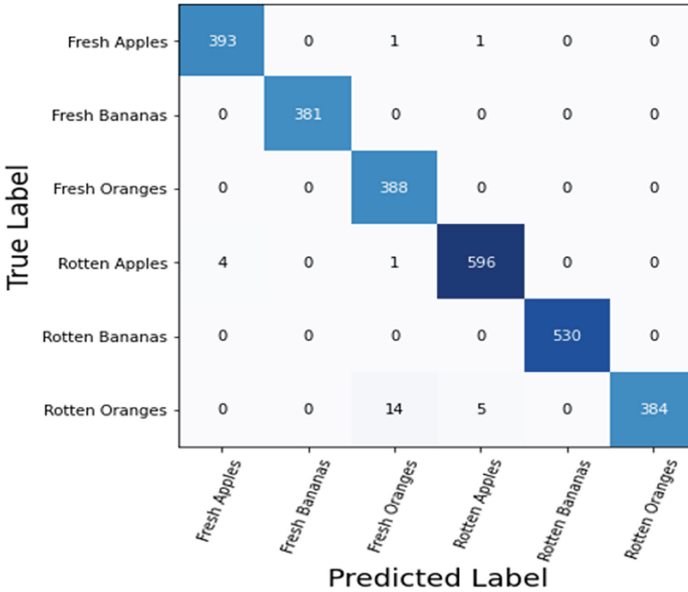


Fig. 5. Confusion matrix obtained for the DenseNet pre-trained network using Model-I for dataset-1

5 Conclusions

This paper proposes fruit classification systems based on generic image classifiers: VGG16, InceptionV3, Xception, DenseNet, and ResNet152V2. We have partially retrained these networks using transfer learning for two fruits datasets. We experimented with changing the classifiers and unfreezing a few sub-blocks of base convolutional networks of DenseNet network. We adopted two training strategies for transfer learning. In the first strategy, the base network is frozen, the old classifier is removed, and a new classifier is added and fine-tuned. Results show that the DenseNet model is superior than other considered networks on both datasets. The accuracy achieved is 99.03% (for dataset-1) and 99.11% (for dataset-2). We applied the second strategy to DenseNet to further improve its accuracy. In the second strategy, the base network of the DenseNet model is partially unfreezed and fine-tuned, which results in an accuracy of 99.61%. It can be concluded that an accurate fruit classification can be developed using these generic image classifiers. In a future study, these classifiers will be tested on other fruits and fruit datasets.

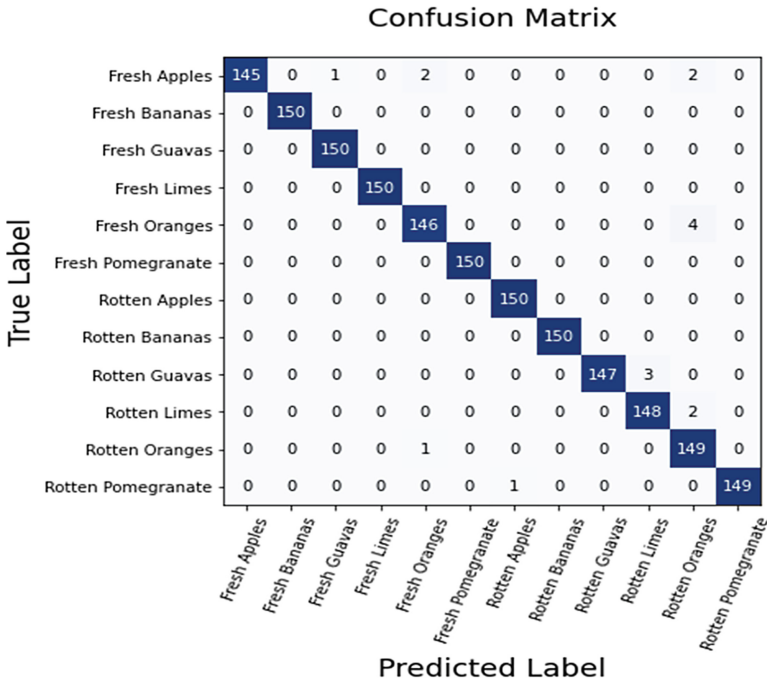


Fig. 6. Confusion matrix obtained for the DenseNet pre-trained network using Model-I for dataset-2

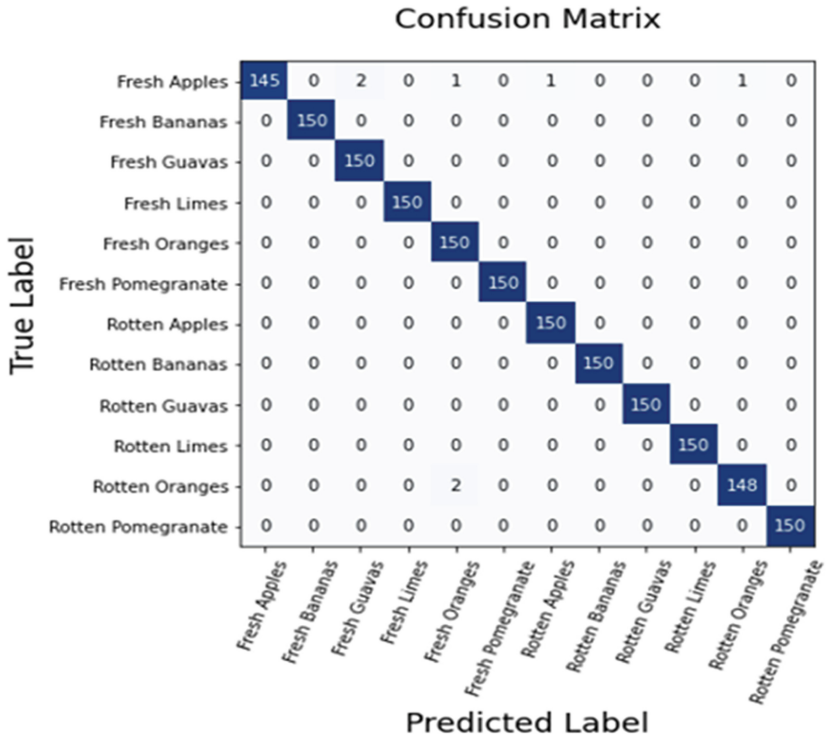


Fig. 7. Confusion matrix obtained for the DenseNet pre-trained network using Model-II for dataset-2

References

1. Agricultural & Processed Food Products Export Development Authority (Ministry of Commerce & Industry, Government of India). <http://agriexchange.apeda.gov.in/indexp/18headproduct.aspx>
2. Hameed, K., Chai, D., Rassau, A.: A comprehensive review of fruit and vegetable classification techniques. *Image Vis. Comput.* **80**, 24–44 (2018)
3. Bhargava, A., Bansal, A.: Fruits and vegetables quality evaluation using computer vision: A review. *J. King Saud Univ. – Comput. Inf. Sci.* **33**(3), 243–257 (2021)
4. Bhargava, A., Bansal, A.: Automatic detection and grading of multiple fruits by machine learning. *Food Anal. Methods* **13**(3), 751–761 (2019). <https://doi.org/10.1007/s12161-019-01690-6>
5. Behera, S.K., Rath, A.K., Sethy, P.K.: Maturity status classification of papaya fruits based on machine learning and transfer learning approach. *Inf. Process. Agric.* **8**(2), 244–250 (2021)
6. Geron, A.: *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 2nd edn. O’Reilly Media, Inc. (2019). ISBN 9781492032649
7. Rawat, W., Wang, Z.: Deep convolutional neural networks for image classification: a comprehensive review. *Neural Comput.* **29**(9), 2352–2449 (2017)
8. Weiss, K., Khoshgoftaar, T.M., Wang, D.: A survey of transfer learning. *J. Big Data* **3**, 9 (2016). <https://doi.org/10.1186/s40537-016-0043-6>

9. Meshram, V.A., Patil, K., Ramteke, S.D.: MNet: a framework to reduce fruit image misclassification. *Ingén. Syst. d'Inf.* **26**(2), 159–170 (2021)
10. Ismail, N., Malik, O.A.: Real-time visual inspection system for grading fruits using computer vision and deep learning techniques. *Inf. Process. Agric.* **9**, 24–37 (2021)
11. Singh, S., Singh, N.P.: Machine learning based classification of good and rotten apple. In: Khare, A., Tiwary, U., Sethi, I., Singh, N. (eds.) *Recent Trends in Communication, Computing, and Electronics. Lecture Notes in Electrical Engineering*, vol. 524, pp. 377–386. Springer, Singapore (2019). https://doi.org/10.1007/978-981-13-2685-1_36
12. Mahrishi, M., Morwal, S., Muzaffar, A.W., Bhatia, S., Dadheech, P., Rahmani, M.K.I.: Video index point detection and extraction framework using custom YoloV4 darknet object detection model. *IEEE Access* **9**, 143378–143391 (2021). <https://doi.org/10.1109/ACCESS.2021.3118048>
13. Long, N.T.M., Thinh, N.T.: Using machine learning to grade the mango's quality based on external features captured by vision system. *Appl. Sci.* **10**, 5775 (2020)
14. Hossain, M.S., Al-Hammadi, M., Muhammad, G.: Automatic fruit classification using deep learning for industrial applications. *IEEE Trans. Ind. Inf.* **15**(2), 1027–1034 (2018)
15. Katarzyna, R., Paweł, M.: A vision-based method utilizing deep convolutional neural networks for fruit variety classification in uncertainty conditions of retail sales. *Appl. Sci.* **9**(19), 3971 (2019)
16. 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.* **145**, 106922 (2020)
17. Wu, S.-L., Tung, H.-Y., Hsu, Y.-L.: Deep learning for automatic quality grading of mangoes: methods and insights. In: 19th IEEE International Conference on Machine Learning and Applications, pp. 446–453 (2020)
18. Rojas-Aranda, J.L., Nunez-Varela, J.I., Cuevas-Tello, J.C., Rangel-Ramirez, G.: Fruit classification for retail stores using deep learning. In: Figueroa Mora, K., Anzurez Marín, J., Cerda, J., Carrasco-Ochoa, J., Martínez-Trinidad, J., Olvera-López, J. (eds.) *Pattern Recognition MCPR 2020. Lecture Notes in Computer Science*, vol. 12088, pp. 3–13. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-49076-8_1
19. <https://www.kaggle.com/sriramr/fruits-fresh-and-rotten-for-classification>
20. <https://ieee-dataport.org/open-access/top-indian-fruits-quality>
21. Russakovsky, O., et al.: Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* **115**(3), 211–252 (2015)
22. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014)
23. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. *arXiv:1512.00567v3 [cs.CV]* (2015)
24. Chollet, F.: Xception: deep learning with depthwise separable convolutions. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1251–1258 (2017)
25. Huang, G., Liu, Z., Maaten, L.V.D., Weinberger, K.Q.: Densely connected convolutional networks. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. Honolulu, USA*, pp. 2261–2269 (2017)
26. Kaiming, H., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. Nevada, USA*, pp. 770–778 (2016)