Fruit-Net: Fruits Recognition System Using Convolutional Neural Network

Olivia Saha Mandal[1], Aniruddha Dey[2(✉)], Subhrapratim Nath[2], Rabindra Nath Shaw[3], and Ankush Ghosh[3]

[1] Department of Computer Science and Engineering, CIT, Kolkata, India
[2] Department of Computer Science and Engineering, MSIT, Kolkata, India
anidey007@gmail.com
[3] University Center for Research and Development (UCRD), Chandigarh University, Mohali, Punjab, India
{r.n.s,ankushghosh}@ieee.org

Abstract. For many industrial applications, classifying fruits is an essential process. A supermarket cashier can use a fruit classification system to distinguish between different types of fruit and their prices. Additionally, it can be used to determine whether a particular fruit species satisfies a person's nutritional needs. In this chapter, we propose a framework for fruit classification using deep learning techniques. More specifically, the framework is a comparison of two different deep learning architectures. The first is a 6-layer light model proposed for convolutional neural networks, and the second is a carefully tuned deep learning model for group-16 visual geometry. The proposed approach is tested using one publicly accessible color-image dataset. The images of fruit that were utilized for training came from our own photos, Google photos, and the data that ImageNet 2012 gave. This database contained 1.2 million images and 1,000 categories. The 1,200 fruit images that had been divided into six groups had been assessed and categorized. The average classification performance was 0.9688 out of a possible range of 0.8456 to 1.0 depending on the fruit, and each photo took about 0.25 s to classify. With only a few errors, the CNN algorithm was able to successfully classify the fruit photographs into the six categories. On the dataset, the CNN, VGG16, and Inception V3 models each achieved classification accuracy results of 96.88%, 72%, and 71.66% respectively.

Keywords: Fruit recognition · Convolutional neural network · Classification · VGG16 · Inception V3

1 Introduction

Fruits are an important part of a balanced diet and offer many health benefits. While some fruits are available throughout the year, some are exclusively during specific times of the year. India's economy continues to be significantly influenced by agriculture. In India, 70% of the land is used for agriculture. In terms of the top fruit growers worldwide,

R. N. Shaw et al. (Eds.): ICACIS 2022, CCIS 1749, pp. 120–133, 2023.
https://doi.org/10.1007/978-3-031-25088-0_10

India comes in third. Deep learning methods are therefore helpful for both marketers and consumers when used to categorize fruits. Currently, information technologies are playing a bigger role in the agriculture sector. We use deep learning-based techniques for fruit sorting to provide highest quality fruit to the customers.

Software for classifying and identifying fruits is crucial since it helps to raise the fruit's quality. It can be challenging to identify a fruit in a store. Manually classifying and valuing anything is difficult. The task of manually counting ripe fruits and assessing their quality is challenging. Rising labor costs, shortages of skilled workers, and rising storage costs are a few of the major problems with fruit production, marketing, storage and more**.** The soft computer vision system offers considerable information on the variety and quality of fruits by reducing costs, improving quality maintenance requirements, and delivering important information. Fruit classification and recognition is one of the most recent developments in computer vision. The set of features, the types of features, the features chosen from the extracted data, and the type of classifier used all have an impact on how accurate a fruit identification system is. Fruit images taken under poor conditions are of poor quality and hide recognizable features. In order to emphasize the nature and characteristics of fruit photographs, techniques for enhancing fruit images are needed.

In all facets of human living, including video surveillance, human-machine interfaces, and picture recovery, object detection [1–4] has received considerable attention. Face recognition in practical applications is extremely challenging due to the wide variations in illumination, posture, obstruction, and shoot point. A very significant and vibrant area of research is image classification. Face recognition, video analysis, image categorization, and other applications of image recognition are available. In the field of image recognition, deep learning (DL), a branch of machine learning (ML), has achieved great results [5]. Hierarchical structures are used to process image attributes by DL and greatly improve the effectiveness of image recognition [6]. In other words, the use of image recognition and DL in supply chain and logistics is starting to take hold as a concept. For example, picture recognition can improve logistics and shipping, as well as correct the faults that plague many fully automated transport vehicles as a result of widespread track identification issues [7].

Agro-related businesses such as food processing, marketing, packaging and fruit sorting have become an increasing focus of research in recent years. Because there are numerous types of the same fruit grown around the world (for example, over 7,100 different varieties of apple; see http://usapple.org), processing and sorting of special crop plants like banana, orange, cherry, apple, mango, and citrus require a lot of time and effort. Therefore, automation can reduce labor expenses and quickly boost production. In earlier studies, researchers proposed various approaches from CV to manually extract fruit traits and ML to classify the CV traits. Several DL approaches to quality assessment and robotic harvesting have been implemented for fruit detection and classification, but these algorithms have few classes and small datasets. In 2017, Liu et al. [8] presented literature analyses of novel fruit classification techniques. Fruit grading algorithms would need to quickly yield adequate accuracies given the development of deep learning [9, 10]. Modern computer vision techniques include real-time tracking of fruit and vegetable

objects [13], nitrogen estimation in fruits and vegetables [12], automated fruit and vegetable sorting [11], and others. Most of the scientific fruit sorting methods, including pattern sorting, are sorted. When rating fruit quality, attention is given to both the overall visual changes and freshness. Despite deep learning's recent surge in popularity, deep learning methods were not used in more than half of the studies [14].

The motivation thorough examination of the available classification methods, the following flaws are looked into:

1. Poor categorization results are caused by the heterogeneous character of images, which is another significant hurdle.
2. Similarities in fruit species include similarities in shape, color, texture, and intensity.
3. High diversity within the variety, depending on the ripeness and maturity of the fruit.

This chapter introduces various CNN, VGG16, Inception V3 deep learning frameworks for fruit image classification to overcome the above shortcomings.

Create a fruit classification model using deep learning applications. In the proposed study, convolution layers are used to extract features from CNN, VGG16, and Inception V3 is employed to categorize the fruits. The main contributions of the chapter are:

- CNN, VGG16, Inception V3 deep learning programmers were used to classify the fruit photos.
- CNN, VGG16, Inception V3 were integrated to create a fruit recognition system that is frequently used for both recognition and classification. This study examines all of these methods for doing fruit recognition and classification.
- The experiment carried out utilizing the suggested method produced pretty effective and encouraging fruit classification findings.

The rest of the chapter is organized as follows. Section 2 defines proposed architectures for the CNN, VGG16, and InceptionV3 models. The investigational results on the fruit database Sect. 3. Finally, Sect. 4 summarises the concluding remarks.

2 Proposed Methods and CNN Structure

Convolutional neural networks, or CNN for short, are one type of deep learning model. Potential elements of such networks include loss layers, ReLU layers, fully-linked layers, convolution layers and average pooling layers. A Rectified Unit (ReLU) layer is then added to each CNN model, which is then followed by a Pooling layer or multiple convolutional layers, and one or more fully connected layers. This is how a CNN is typically built. A CNN considers the architecture of the photographs when analyzing photos, as opposed to a typical neural network, which disregards the structure of the data being processed. Note that traditional neural networks transform their input into a one-dimensional array before training a classifier. The learnt classifier will become more responsive to changes in location as a result.

Some of the finest solutions to problems from the MNIST dataset have been demonstrated using inter deep neural networks. The study claims that they employ numerous maps within each layer as well as many layers of pseudo neurons. Although the complexity of such nets makes training them more challenging, graphics processors and programming created expressly for them may help to get around this problem. Winner-take-all neurons with maximum pooling are used to build the network, and these neurons select the winner.

According to the results of yet another study, convolutional networks have been demonstrated to attain higher levels of accuracy in the field of computer vision. An all-convolutional network that performs at exceedingly high levels is described in full in the publication. The research chapter advises substituting equal-function convolutional layers for pooling and convolution layers. The problem can be resolved by employing shorter convolutional inside the network, which also functions as a form of regularization, however this may increase the amount of variables and add inter-feature correlations The explanations from each of the strata that make up the CNN network are given below.

2.1 Convolutional Layers

The name of these layers was inspired by the convolutional method. Condensation is a mathematical process that, when applied to two functions, produces a third function that is a single transformed (convolved) version of the original function. As a result of the total that only a portion of the original purpose is translated, the resulting function offers an integral of a point-wise multiplication of the two roles. The amount of translation of one of the main focuses affects this integral.

In a convolutional layer, clusters of neurons are joined together to form kernels. The kernels reliably maintain the same depth as the input, despite their very small size. Neurons in the kernel are connected only to a small area of input called the receptive field. This is because for high-dimensional inputs like images, connecting every cell to every early stop is very inefficient. The receptive field is the name given to this area of the input. For illustrative purposes, a picture that is 100 $\times$ 100 has 10,000 pixels, but if the first 100 neurons were present, there would be 1,000,000 parameters. Instead of storing weights across the dimensions of the input, each neuron stores weights for the dimensions of its core input. The kernels traverse the input space both horizontally and vertically, extracting high-level properties and resulting in a two-dimensional activation map. A value that specifies the speed at which the kernel is floating is called a parameter. Convolution layers are created by stacking the resulting activation maps, and these layers are then used to choose the inputs for the following layer.

A convolutional layer is added to 32 $\times$ 32 image to produce 28 $\times$ 28 activation map. The picture size is decreased when the number of convolutional layers is increased, which causes data loss and the disappearing gradient problem. We use padding to make this appropriate. The padding of input data with constants can make it larger. Since this constant is often zero, the method is known as zero padding. This means that the generated feature map will be padded to the same extent as the feature map. If add an odd number of additional columns, a second column is added to keep left and right padding consistent. According to this criterion, “valid” padding is equivalent to “no padding”. This enables a kernel to ignore image pixels and not output them. The step affects

the behavior of the convolution process when using larger images and more complex kernels, where the kernel shifts the input and the strides argument is used to determine the number of positions to skip. When a kernel slides the input, the strides argument is used to determine how many positions to skip. Rectified Linear Units-based activation function max is used in this layer. The nonlinear properties of the network are improved, not diminished.

2.2 Pooling Layers

Convolution is used to reduce the spatial dimension of the representation and the computational load of the network. Overfitting can also be avoided by pooling layers. The most typical filter size with a stride of two is 2×2. The input is consequently decreased by a factor of four.

2.3 Fully Connected Layers

Normal neural network layers are regarded as fully coupled layers. Each output from the layer below is linked to every neuron in a layer that has full connectivity. The calculations that are carried out in the background of a convolution layer are the same as those carried out in a fully linked layer. So it is possible to switch back and forth between the two (Fig. 1).

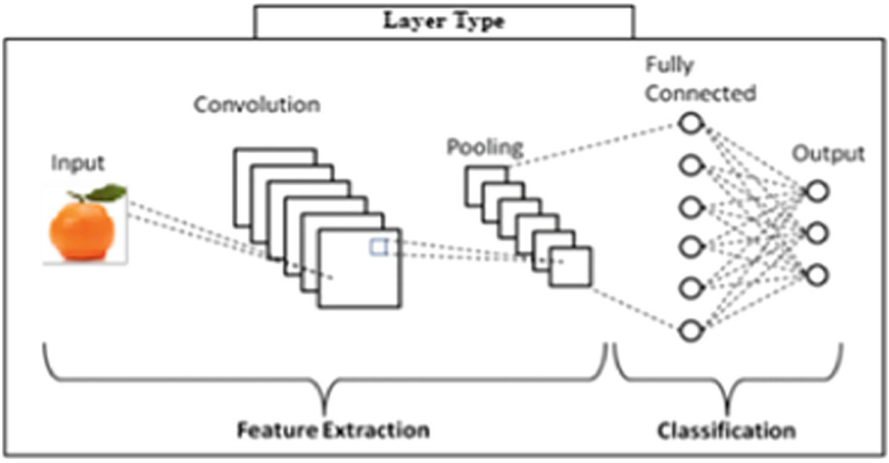


Fig. 1. Proposed CNN architecture

We employed a deep neural network to finish this challenge. The sorts of layers employed in this kind of network include convolution, pooling layers, rectified layers, convolution layer and loss layers, as was already established. Recurrent Unit (ReLU) layer, a Pooling layer, one or more convolutions, and finally one or even more fully connected layers are placed before each convolution. This is how a CNN is typically built.

Rectified Linear Unit (ReLU): A ReLU activation function executes a threshold operation to each input value, where any value negative is set as zero and for positive value output the input value [5]. ReLU function is defined as:

$$fr = ReLU(z) = max\{0, z\} \tag{1}$$

where fr is function RELU over input z and max() function takes values either 0 or input z.

Softmax: Softmax function that transforms a vector of numbers into a vector of probabilities [5]. Function is state as below:

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{2}$$

where σ is Softmax function, $\vec{z}$ input vector, e^{z_i} exponential function for input vector, K is number of classes, e^{z_j} is exponential function for output vector.

The CNN takes into account the shape of the images it is analyzing, unlike a typical neural network. One feature that sets the CNN apart from other neural networks is this one. A one-dimensional array is formed from an input before it is reassigned to a traditional neural network. The training classifier is less sensitive to changes in location as a result.

2.4 Architecture Based on VGG-16

A more complex CNN model is VGG-16. There are five convolutional operation blocks inside. A max-pooling layer connects adjacent blocks. Each block has a collection of 3×3 layers of convolutions. Within each block, the number of convolution kernels remains constant and increases from 64 in the first block to 512 in the last block [5, 7]. There are a total of 16 learnable layers.

2.5 Architecture Based on Inception V3

Convolutional neural networks are the foundation of the deep learning model known as Inception V3 that is used to classify images. The Inception V3 is a advanced version of the Inception V1, a foundational model that was first released as Google Net in 2014 [5, 7]. It was designed by a Google team, as the name suggests.

The data were over fit when numerous deep layers of convolutions were used in a model. The Inception V1 model employs the concept of having many filters of various sizes on the same level to prevent this from occurring. Thus, in the inception models, parallel layers are used in place of deep layers, making the model larger rather than deeper.

The first step is to identify the 200 kinds of objects within the image, also known as the local action of the item. The second is referred to as the separation of images and involves writing each image in one of the 1000 categories.

3 Empirical Results

Both the method used to create the data set and the information that is contained in it will be covered in this section. The apples were photographed while rotating it with a motor, and then selected frames from the video were used to produce the visuals. Fruits were seeded into a slow-speed motor (three revolutions per minute), rotor for a 20-s clip, which was then recorded. We placed a blank sheet of white chapter behind the fruit to serve as a backdrop. However, the backdrop was inconsistent due to the various ways the light was falling; therefore we had to provide an algorithm to distinguish the fruits from the background. Always start at the top of the image and mark all pixels there. The flood fill method is followed here. Then, if we find any pixels nearby that have a color range that is smaller than a certain value, we mark all of those pixels as well. Up until there are no more pixels to mark, we repeat the previous phase iteratively.

All of the defined pixels are taken into account as the background and are then filled with white. The pixel count after that is regarded as a segment of the item. A component of the algorithm used to create each movie is the maximum value that may be permitted for the distance between any two adjacent pixels. The fruit was reduced so that it would fit within a 300 $\times$ 300 image. Our aim to be able to handle considerably larger photos in the future, but this will require much prolonged training sessions. Table 1 describes number of fruit images for each fruit.

Table 1. Number of images for each fruit

Label	Training images	Test images
Freshapples	186	46
Freshbanana	190	43
Freshoranges	165	41
Rottenapples	263	65
Rottenbanana	245	61
Rottenoranges	178	44

Inception-v3 is introduced to get 72% accuracy whereas VGG16 used network model for image identification introduced to get 71.66% accuracy in the ImageNet database. The behavior of training and validation accuracy and loss versus epoch number during fine-tuning the VGG16 and Inception V3 is shown in Fig. 2 and Fig. 3, respectively (Fig. 4).

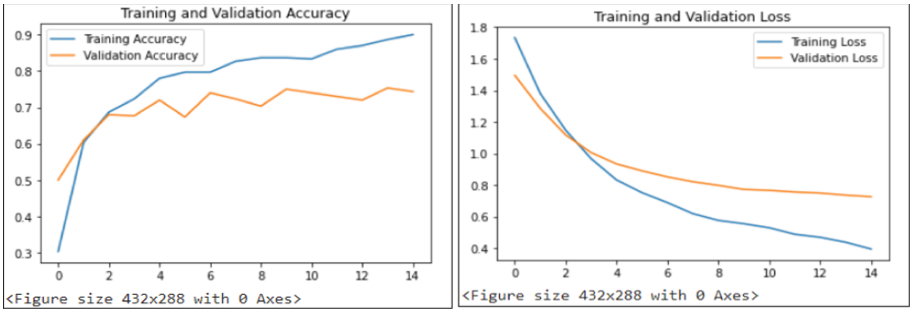


Fig. 2. Training and validation accuracy and loss of VGG16

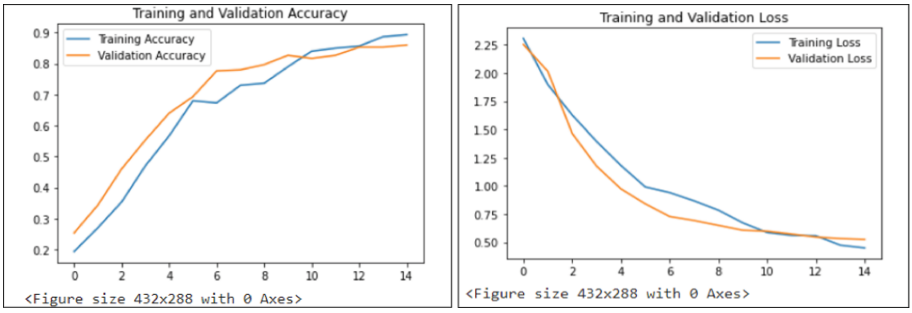


Fig. 3. Training and validation accuracy and loss of inception V3

Model Accuracy 0.72

	precision	recall	f1-score	support
apple	0.81	0.68	0.74	50
banana	0.74	0.90	0.81	50
cantaloupe	0.63	0.58	0.60	50
grapefruit	0.72	0.42	0.53	50
grapes	0.73	0.90	0.80	50
kiwi	0.70	0.84	0.76	50
accuracy			0.72	300
macro avg	0.72	0.72	0.71	300
weighted avg	0.72	0.72	0.71	300

Model Accuracy 0.7166666666666667

	precision	recall	f1-score	support
apple	0.58	0.72	0.64	50
banana	0.82	0.90	0.86	50
cantaloupe	0.71	0.48	0.57	50
grapefruit	0.59	0.40	0.48	50
grapes	0.72	0.94	0.82	50
kiwi	0.86	0.86	0.86	50
accuracy			0.72	300
macro avg	0.71	0.72	0.70	300
weighted avg	0.71	0.72	0.70	300

Fig. 4. Model accuracy of inception V3 and VGG16

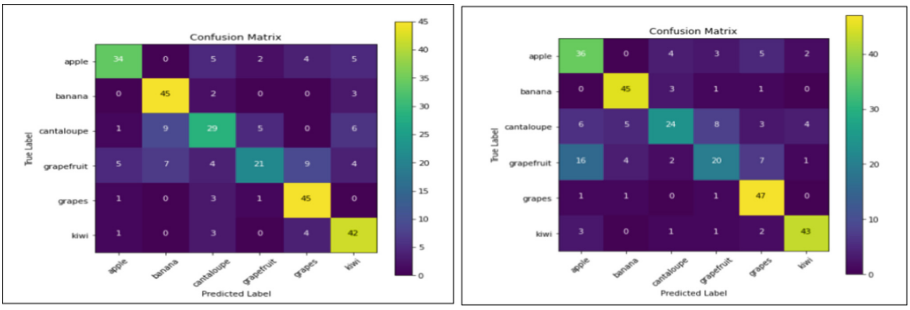


Fig. 5. Confusion matrix of inception V3 and VGG16

The confusion matrix of Inception V3 and VGG16 for the test dataset is illustrated in Fig. 5. The behavior of training and validation accuracy and loss versus epoch number during fine-tuning the model is shown in Fig. 6, Fig. 7 shows the summary of proposed CNN model for fruit recognition is given below:

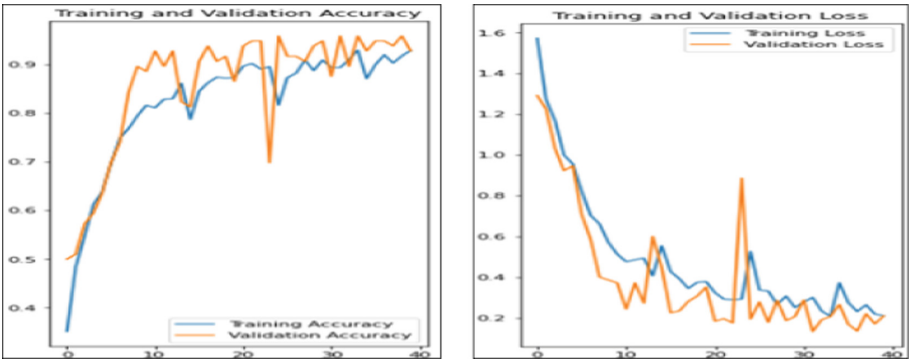
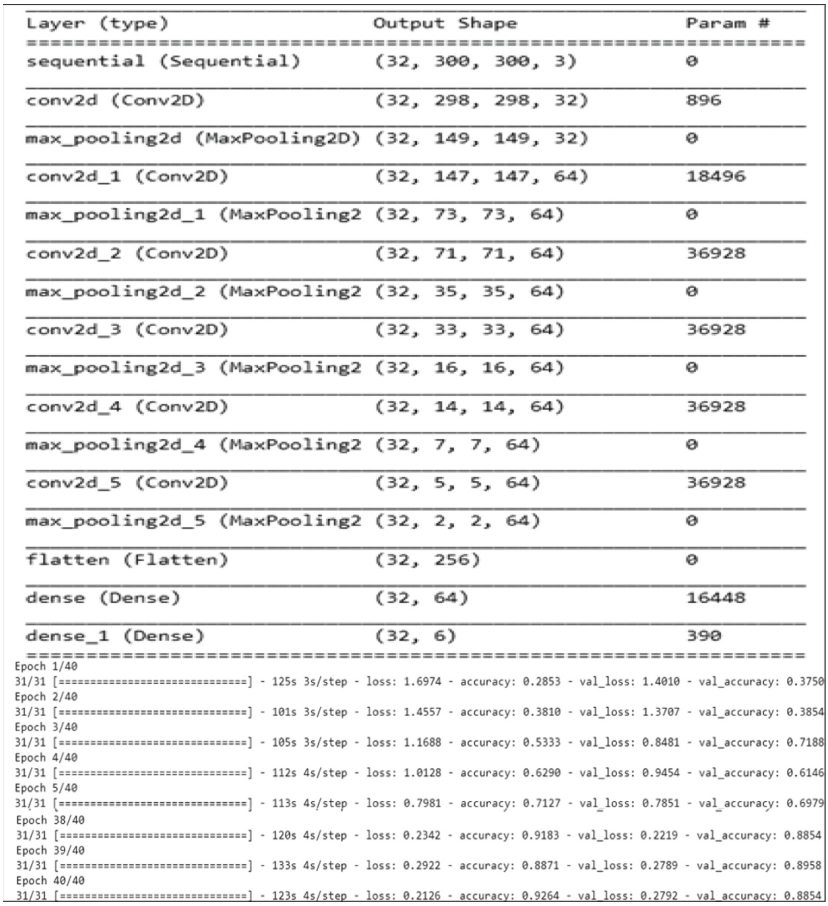


Fig. 6. Training and validation accuracy and loss of proposed CNN



Layer (type)	Output Shape	Param #
sequential (Sequential)	(32, 300, 300, 3)	0
conv2d (Conv2D)	(32, 298, 298, 32)	896
max_pooling2d (MaxPooling2D)	(32, 149, 149, 32)	0
conv2d_1 (Conv2D)	(32, 147, 147, 64)	18496
max_pooling2d_1 (MaxPooling2	(32, 73, 73, 64)	0
conv2d_2 (Conv2D)	(32, 71, 71, 64)	36928
max_pooling2d_2 (MaxPooling2	(32, 35, 35, 64)	0
conv2d_3 (Conv2D)	(32, 33, 33, 64)	36928
max_pooling2d_3 (MaxPooling2	(32, 16, 16, 64)	0
conv2d_4 (Conv2D)	(32, 14, 14, 64)	36928
max_pooling2d_4 (MaxPooling2	(32, 7, 7, 64)	0
conv2d_5 (Conv2D)	(32, 5, 5, 64)	36928
max_pooling2d_5 (MaxPooling2	(32, 2, 2, 64)	0
flatten (Flatten)	(32, 256)	0
dense (Dense)	(32, 64)	16448
dense_1 (Dense)	(32, 6)	390

```
Epoch 1/40
31/31 [==============================] - 125s 3s/step - loss: 1.6974 - accuracy: 0.2853 - val_loss: 1.4010 - val_accuracy: 0.3750
Epoch 2/40
31/31 [==============================] - 101s 3s/step - loss: 1.4557 - accuracy: 0.3810 - val_loss: 1.3707 - val_accuracy: 0.3854
Epoch 3/40
31/31 [==============================] - 105s 3s/step - loss: 1.1688 - accuracy: 0.5333 - val_loss: 0.8481 - val_accuracy: 0.7188
Epoch 4/40
31/31 [==============================] - 112s 4s/step - loss: 1.0128 - accuracy: 0.6290 - val_loss: 0.9454 - val_accuracy: 0.6146
Epoch 5/40
31/31 [==============================] - 113s 4s/step - loss: 0.7981 - accuracy: 0.7127 - val_loss: 0.7851 - val_accuracy: 0.6979
Epoch 38/40
31/31 [==============================] - 120s 4s/step - loss: 0.2342 - accuracy: 0.9183 - val_loss: 0.2219 - val_accuracy: 0.8854
Epoch 39/40
31/31 [==============================] - 133s 4s/step - loss: 0.2922 - accuracy: 0.8871 - val_loss: 0.2789 - val_accuracy: 0.8958
Epoch 40/40
31/31 [==============================] - 123s 4s/step - loss: 0.2126 - accuracy: 0.9264 - val_loss: 0.2792 - val_accuracy: 0.8854
```

Fig. 7. Summary CNN model

Some of the correctly and incorrectly classified fruit images are shown in Fig. 8 and Fig. 9.

Fig. 8. Some of the fruit images that classified correctly

Fig. 9. Some of the fruit images that classified incorrectly

4 Conclusion

In this chapter, a framework for classifying fruits based on deep learning was suggested. The suggested framework examined three CNN models: a small CNN model, a VGG-16 fine-tuned model, and Inception V3. The suggested framework was tested on two datasets of varying sizes and complexity. On both datasets, the VGG-16 fine-tuned model demonstrated excellent accuracy. On dataset, the CNN model also had good accuracy due to data augmentation. Two more methodologies from the literature have been used to compare the two models' performances. It was revealed that the proposed CNN models performed better than the two already-used approaches. Regarding further work, we'll assess the suggested framework across a wider range of classes (using extra fruit and vegetable species). We will also look into the impact of other parameters, including an activation function, a pooling function, and an optimization technique. A cloud-based framework can also be used with the suggested framework.

References

1. Bhattacharya, S., Ghosh, M., Dey, A.: Face detection in unconstrained environments using modified multitask cascade convolutional neural network. In: Proceeding of the ICI2C 2021, pp. 287–295 (2021)
2. Dey, A., Chakraborty, S., Kundu, D., Ghosh, M.: Elastic window for multiple face detection and tracking from video. In: Das, A., Nayak, J., Naik, B., Pati, S., Pelusi, D. (eds.) Computational Intelligence in Pattern Recognition, vol. 999, pp. 487–496. Springer, Cham (2019). https://doi.org/10.1007/978-981-13-9042-5_41
3. Dey, A.: A Contour based procedure for face detection and tracking from video. In: Proceeding of the RAIT 2016, pp. 252–256 (2016)
4. Chowdhury, S., Dey, A., Sing, J.K., Basu, D.K., Nasipuri, M.: A novel elastic window for face detection and recognition from video. In: Proceeding of the ICCICN 2014, pp. 252–256 (2014)
5. Pak, M., Kim, S.: 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 (2017)
6. Zhai, H.: 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
7. Biswas, S., Bianchini, M., Shaw, R.N., Ghosh, A.: Prediction of traffic movement for autonomous vehicles. In: Bianchini, M., Simic, M., Ghosh, A., Shaw, R.N. (eds.) Machine Learning for Robotics Applications. SCI, vol. 960, pp. 153–168. Springer, Singapore (2021). https://doi.org/10.1007/978-981-16-0598-7_12
8. Liu, F., Snetkov, L., Lima, D.: 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
9. Bhargava, A., Bansal, A.: Fruits and vegetables quality evaluation using computer vision: a review. J. King Saud Univ. Comput. Inf. Sci. **33**, 243–257 (2018)
10. Pandey, R., Naik, S., Marfatia, R.: Image processing and machine learning for automated fruit grading system: a technical review. Int. J. Comput. Appl. **81**(16), 29–39 (2013)

11. Cunha, J.B.: Application of image processing techniques in the characterization of plant leafs. In: IEEE International Symposium on Industrial Electronics, pp. 612–616 (2003)
12. Tewari, V.K., Arudra, A.K., Kumar, S.P., Pandey, V., Chandel, N.S.: Estimation of plant nitrogen content using digital image processing. Agric. Eng. Int. CIGR J. **15**(2), 78–86 (2013)
13. Mukhopadhyay, M., et al.: Facial emotion recognition based on textural pattern and convolutional neural network. In: 2021 IEEE 4th International Conference on Computing, Power and Communication Technologies (GUCON), pp. 1–6 (2021). https://doi.org/10.1109/GUCON50781.2021.9573860
14. Tripathi, M.K., Maktedar, D.D.: A role of computer vision in fruits and vegetables among various horticulture products of agriculture fields: a survey. Inf. Process. Agric. **7**, 183 (2019)