Developing a Tool to Classify Different Types of Fruits Using Deep Learning and VGG16

Chapter · October 2022

DOI: 10.1007/978-3-031-19958-5_61

7 authors, including:

Mahfuza Yousuf
East West University (Bangladesh)

Md. Saifur Rahman
Università degli Studi di Siena

Farhana Islam Proma
East West University (Bangladesh)

Ahmed Wasif Reza
East West University (Bangladesh)

Developing a Tool to Classify Different Types of Fruits Using Deep Learning and VGG16

Mobassir Ahsan[1], Mahfuza Yousuf[1], Md. Saifur Rahman[1], Farhana Islam Proma[1], Omar Tawhid Imam[2], Ahmed Wasif Reza[1], and Mohammad Shamsul Arefin[3,4](✉)

[1] Department of Computer Science and Engineering, East West University, Dhaka 1212, Bangladesh
wasif@ewubd.edu
[2] Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, Dhaka 1000, Bangladesh
[3] Department of Computer Science and Engineering, Daffodil International University, Dhaka 1341, Bangladesh
sarefin@cuet.ac.bd
[4] Department of Computer Science and Engineering, Chittagong University of Engineering and Technology, Chattogram 4349, Bangladesh

Abstract. In this paper, we present two methods for the classification of fruits of Bangladesh from image processing techniques. We have used deep learning convolutional neural network in our model and VGG16 in another model. From both models, we have found 99% accuracy. Initially, we used only five classes (apple, orange, jackfruit, watermelon, banana) for building these models. Evaluating our model gives us accuracy on the test dataset and by inputting one fruit image our model predicts the fruit what it is. We have checked and experimented with our model several times that it can detect fruit accurately from single fruit images. If our model goes through further improvement, it can be an application that will help shopkeepers or farmers on fixing price calculations on both online and offline platforms.

Keywords: Fruits classification · VGG16 · Convolutional neural networks · Deep learning · Image processing

1 Introduction

Fruit classification means the technique of classifying different types of fruits from images. Classifying several types of fruits or vegetables is one of the recent research topics in the agricultural industry. For this reason, several classification techniques for image processing are developed. Any agriculturist or common people can be benefited if automation machine vision tries to classify fruits or vegetables. It can be used in any shop or supermarket to detect and generate the price of a fruit automatically. There are many problems in the detection of fruit like variation in their size, shape, color, location, changes in illumination, etc. Color and shape are the strongest features in fruit

P. Vasant et al. (Eds.): ICICO 2022, LNNS 569, pp. 1–12, 2023.
https://doi.org/10.1007/978-3-031-19958-5_61

object identification. Color classification provides us with efficient reports on spectral characteristics and tries to detect items from a known dataset by matching. The texture is another significant component that attempts to separate unique examples of pictures by separating the subject of intensity among pixels and their neighboring pixels.

In our paper, we have developed a tool to classify different types of fruit in Bangladesh. We have done image processing techniques using deep learning and VGG16. Initially, we tried five fruits (apple, banana, orange, watermelon, and jackfruit) to build our classification model. At first, from images, our model checks out the color, shape, and texture of the object. Then, we process the similarity of extracted features with objects included in the database. Lastly, classify and identify fruits with the highest similarity.

The rest of the paper is organized as follows: Sect. 1 comprises the introduction of this research paper, Sect. 2 gives a brief review of existing or previous works related to fruit classification or detection systems, Sect. 3 provides dataset description, data preprocessing, algorithm and model description, Sect. 4 explains about experimental setup, result, implementation and performance evaluation of the proposed algorithm using deep learning and VGG16, which discusses the features of algorithm and finally Sect. 5 concludes with the limitations of our research and suggestions for future work of this research paper.

2 Related Work

In this section, we focus on the existing and related work of object classifications from images. In most of the earlier works, the authors have researched how to recognize and classify several types of fruits.

In [1], Md Shaddam Hossen, Mohammad Shamsul Arefin, and Rezaul Karim proposed a method using color, shape, and texture for detecting fruits from images, by doing image acquisition, color feature extraction, shape feature extraction, and creating grey level co-occurrence matrix and got high accuracy.

In [2], M. S. Hossain, M. Al-Hammadi, and G. Muhammad used to build the framework using two deep learning architectures. They used two color image datasets; however, the second dataset's precision is not good enough.

In [3], S. K. Behera, A. K. Rath, and P. K. Sethy presented both deep feature extraction and transfer learning, six top deep neural network architectures were used. They use deep features collected from fully connected layers of CNN (Convolution Neural Network) models to conduct SVM classification. Though their approach's high accuracy did not develop a smartphone application for the classification of Indian fruits with a vast and diverse image dataset.

In [4], A. Bhargava, and A. Bansal introduced a method that distinguishes between four sorts of fruits and ranks them according to their quality. They used four different classifiers (k-NN, SVM, SRC, ANN). The lack of this paper is It is unable to determine the number of distinct varieties of fruits from an image. Also, the accuracy of their system is not much satisfied.

In [5], Feng J, Zeng L, Liu G, and Si Y used the fuzzy inference system and the multilayer segmentation algorithm was created for the recognition of overlapping fruits.

A histogram-based head threshold detection technique was presented. However, its accuracy level is not so sufficient.

In [6], J. Feng, L. Zeng, L. He developed based on the pseudo-color and texture information from MSX photos for locking fruit locations. The system was effective for fruit detection. But they are not considered multi-view image fusion.

In [7], Hetal N. Patel, R.K. Jain, and M.V. Joshi proposed an efficient fruit recognition algorithm based on numerous features. Several characteristics are examined, including intensity, color, edge, and direction. They didn't consider some other features such as symmetry features.

In [8], Sreekanth G, Thangaraj P, and Kirubakaran S. applied the k-means algorithm to classify fruits. They considered automation uses the center of mass position to pluck the fruit. But they did not consider image shape or size and could not detect multiple fruit images.

In [9], J. Naranjo-Torres, M. Mora, R. Hernández-García, R. J. Barrientos, C. Fredes, and A. Valenzuela, the foundations, tools, and two examples of using CNNs for fruit sorting and quality control are presented in this article. They looked at several studies that looked at the application of CNN-based techniques for fruit image processing. Moreover, they did not give good accuracy according to their work.

In [10], D. Sahu, and C. Dewangan created an automated tool that can detect and classify mango fruits based on their shape, size, and color properties using digital image analysis. The presented method can be utilized to detect obvious faults, stems, size, and form of mangos, as well as classify them quickly and accurately. But they used only a single image, they did not work any multiple images.

In [11], PL. Chithra, M. Henila proposed a new approach. First, the RGB image was transformed to an HSI image. The region of interest was then segmented using Otsu's thresholding method, which only considered the HUE component picture of the HSI image. SVM classifier to classify the test images as apples and bananas and it gave good accuracy, but the KNN classification method did not give 100% accuracy.

In [12], Joseph J, Kumar V, and Mathew S developed a convolutional neural network model in deep learning with 131 different fruit and vegetable classes, which classified those images into different categories. They used TensorFlow backed to build the model and got 94.35% accuracy after 50 epochs of training.

In [13], Saranya N, Srinivasan K, Pravin Kumar S, Rukkumani V, and Ramya R tried both traditional machine learning methods and a deep learning approach and compared the results. In the machine learning model, KNN and SVM algorithms were applied. For that, basic features of fruit such as size color, height, and width were extracted. They found that the deep learning CNN model gave more effective results than the traditional machine learning model.

In [14], Sa I, Ge Z, Dayoub Fet al. proposed CNN (Faster R-CNN) deep neural network method for fruit detection and it is retrained to recognize seven different fruits. It is a vision-based fruit detection system. They achieved impressive performance with a good F1 score but only around half of the scaled-down item detection can be handled by the suggested detector. The entire approach took four hours per fruit to annotate and train the new model which is time-consuming.

In [15], Barbole D, Jadhav P, and Patil S. examined the issue of fruit harvesting as a traditional method that is costly and time-consuming. The researcher worked on a deep-learning (DL) strategy that was used to explore a mixture of all the most recent fruit identification and segmentation algorithms. A comparison of all fruit detection and segmentation approaches has been carried out. However, their conclusions have been drawn based on the review to direct future research challenges and scope. They didn't come up with a good result.

To construct this proposed methodology, the above-mentioned research publications were surveyed. In this paper, we proposed a system that classifies different types of fruit in Bangladesh based on color and shape. Furthermore, we will consider five types of fruit images.

3 System Architecture and Design

The proposed fruits classifying method's system architecture and design procedure are described in detail in this part. Color and shape are used to identify fruits when identifying fruits from images. The procedures of our fruit identification approach are described in the sections below.

3.1 Dataset Description

In our dataset, we have worked with five different types of fruits. The fruits used in our dataset are:

- Apple
- Orange
- Banana
- Watermelon
- Jackfruit

There are a total of 830 images in our dataset. Which includes 200 apples, 101 oranges, 193 bananas, 125 jackfruit, and 211 watermelon images. Almost all the images consist of a single image. There are some characteristic differences between fruits of the same types to reduce the bias of our model. E.g., the Apples and Jackfruits have variations of color between them, while the banana images might only have a single banana or multiple bananas.

3.2 Data Preprocessing

In our classification, we take a photo of single fruit as input and then preprocessed the image before checking the similarity of the input image. Firstly, we re-size the input image into 100*100 pixel where height is 100 pixels and width are 100 pixels. Re-sized the test image based on the input image and then upload the re-sized image again.

3.3 Proposed Algorithms

The steps need to implement our proposed model are depicted in the algorithms given below. First the procedures for implementing our entire model:

Procedure:
1: Load RGB image dataset by mounting google drive
2: Creating NumPy empty array for labels
3: Creating 4D empty NumPy array storing image
4: Read data from the dataset folder and add images (RGB values) to the array
5: Add an appropriate label for each image
6: Split the dataset into train-test sets
7: Apply one hot encoding on labels
8: Create and compile the CNN model
9: Train the model on the training dataset
10: Evaluate the model using the holdout test dataset

To train a model with our dataset, we need to be able to load the dataset into memory. For this purpose, we have used algorithm 1, which takes our dataset of preprocessed images, and loaded them and their appropriate class labels in NumPy arrays.

Algorithm 1: Load pre-processed images into a single array. To train a model first we need to load the train and test images into memory first. The following algorithm loads the images into memory as a NumPy array.

Input: Dataset path
Output: NumPy array containing all the images

1. **Begin**
2. Provide dataset path
3. Create an empty image and label the array
4. Get the list of classes
5. **For each class in the dataset do**
6. List all the files of that class
7. Load the RGB values of images on a 4D temporary array
8. Stack the temporary array on top of the image array
9. Append the labels to the labels array
10. **End for**
11. Split image and labels array into train-test sets with a ratio of 0.25
12. Convert train and lest labels using One hot encoding
13. **End**

And then, our proposed model is implemented in algorithm 2. Which is a custom sequential CNN model to classify fruits from our dataset.

Algorithm 2: Custom CNN model implementation. The following algorithm is used to build the layers of our model.

Input: A preprocessed image dataset
Output: Model for training

1. **Begin**
2. Initialize model as Sequential model
3. Set input size (100,100,3)
4. Add 1^{st} convolution layer with 3x3 kernel and 32 channels with ReLU activation
5. Add 1^{st} max pooling layer with pool size 2x2
6. Add 2^{nd} convolution layer with 64 channel and 5x5 kernel
7. Add 2^{nd} max pooling layer with pool size 2x2
8. Flatten the model
9. Add a dense layer with 1000 neurons and ReLU activation
10. Add a dropout layer with a dropout rate of 0.5
11. Add 2^{nd} dense layer with 500 neurons and ReLU activation
12. Add 2^{nd} dropout layer with a dropout rate of 0.5
13. Add 3^{rd} dense layer with 250 neurons and ReLU activation
14. Add output dense layer with 5 neurons and softmax activation
15. Compile model with 'categorical_crossentropy' loss, 'adam' optimizer, and accuracy as evaluation matrix
16. **End**

Algorithm 3: Training model. It is used to train the created model on our dataset.

Input: Image and labels dataset
Output: A trained model for classification

1. **Begin**
2. Load pre-trained weights if available
3. A fit model with image and labels array for 30 epochs with a validation split of 0.2
4. Run epochs till trained
5. Save newly trained weights
6. Plot graph of accuracy and loss changes during training
7. **End**

And finally, we test out our trained model with algorithm 4. Where we input any fruits image and check how successfully the model can classify the fruit.

Algorithm 4: Classification of fruits

Input: Trained model, the image of fruits
Output: Class of the fruit

1. **Begin**
2. Load image of fruit into memory
3. Resize image to (100x100) shape
4. Run model prediction on the resized image
5. Sort prediction array
6. Print the highest prediction value as a classified result of the fruit
7. **End**

3.4 Model Architecture and Design

We have developed multiple models for the fruit classification task. One is our own (I) custom model for high-accuracy fruits detection, and another one is a (II) modification of the VGG16 models to suit our fruits classification tasks. The architecture of both models has been visualized in Figs. 1 and 2, respectively.

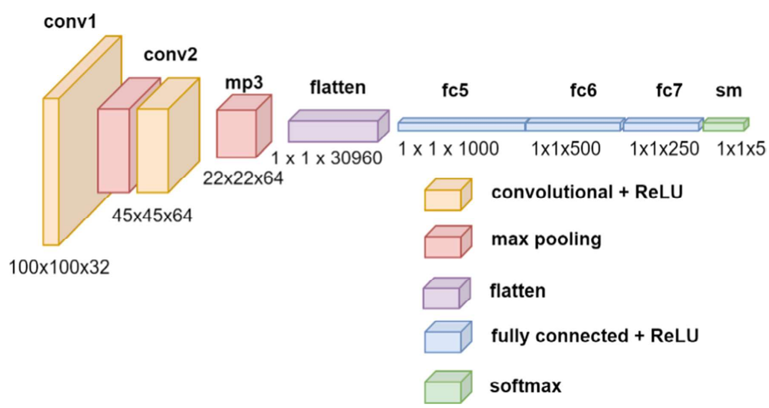

Fig. 1. Proposed custom fruits detection model for fruits classification

As our input image size is 100 × 100, the first layer of the model must be of the same size. Similarly, since there are only 5 classes as the output, the last layer of the neural network model must also have exactly only 5 neurons as the output layer.

To build our custom model, first, the model was declared as a sequential model. Then we added a 2d convolution layer with an input size of 100 × 100 with a kernel of (3, 3) and 32 channels with ReLU activation. Then we have added a MaxPooling layer of pool size (2, 2). Again, we added another 2d convolution layer with 64 channels using (5, 5) size kernel and another MaxPooling layer with pool size (2, 2). Then the 2d outputs were flattened and added to a dense layer with ReLU activation and 1000 neurons. On the front of which a dropout layer with a dropout rate of 0.5 was added to

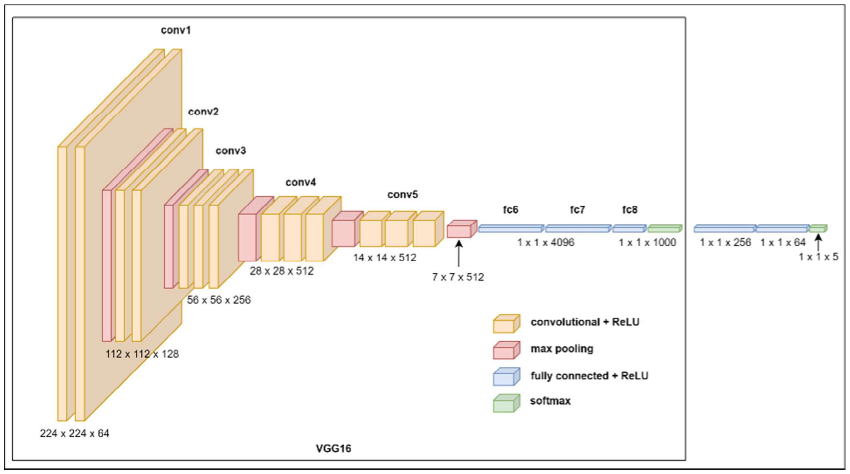

Fig. 2. Proposed custom VGG model for fruits classification

ensure satisfactory performance. And another dense layer with 500 neurons is added, and then another dropout layer with a dropout rate of 0.5, and then another dense layer of 250 neurons. And finally, the model is finished by adding a final dense layer of 5 neurons with softmax activation.

And with that our fully custom high accuracy fruits classification model was ready without needing any additional pre-trained model. The model architecture is complex in design and has 31,656,165 total parameters, but it is very fast and accurate in both training and classifying fruits.

Meanwhile, a VGG16-based model was also designed for more accurate classification. VGG16 with pre-trained weights were taken directly, and then the output of it was flattened and a dense layer with 256 neurons and ReLU activation was added. And then a dropout layer is added with a rate of 0.5. In front of that, another 64-neuron dense layer is added with ReLU activation. And finally, the 5-neuron output layer is added at the end to complete the model. The input shape is changed to 100*100 to be trainable with our dataset as well. This smaller image size should greatly reduce the computation cost of the model and make it usable on traditional hardware.

This model is a simple VGG16 model with 15,911,365 parameters total. Of which 1,196,677 are trainable and the rest are non-trainable. It can reach high accuracy and low loss in way fewer epochs. However, it takes a lot longer to train and classify compared to our custom model.

4 Implementation and Experimental Result

4.1 Experimental Setup

The proposed system has been completely implemented on the Google Colab notebook environment, with 13 GB of RAM and 108 GB of disk allocated for us. The backend used is Python3 Google Compute Engine with the TPU option enabled for faster training. The models are designed using TensorFlow and Keras machine learning frameworks and use the built-in layers and functions from these frameworks.

4.2 Experimental Result

To evaluate the performance of our model, we have tested our model by predicting the classes of various fruits from our holdout dataset (n = 208). In Table 1, the overall performance of the system while classifying each category of fruits from images has been given.

Table 1. Image detection accuracy of models for each fruit category

Fruits	No. of images checked	No. of images detected (custom)	Accuracy (custom model)	No. of images detected (VGG16)	Accuracy (VGG16)
Apple	47	47	100%	46	97.87%
Jackfruit	30	29	96.67%	30	100%
Orange	29	29	100%	27	93.10%
Watermelon	55	55	100%	55	100%
Banana	47	47	100%	47	100%

4.3 Implementation

For the classification task, we have built 2 distinct models using the Convolutional neural network architecture. First used a sequential CNN model to build a custom model for fruit classification. Furthermore, a pre-trained VGG16 was also taken and then modified to do the same fruits classification task as well. Our dataset differs vastly from the original dataset used for the original VGG16 model. It has a pixel resolution of 100 × 100 after preprocessing and thus requires a custom input size of 100 × 100 too. Meanwhile, we only had 5 classes, so the output layer of the models needed to have 5 layers too. We have used convolutional layers, maxpooling layers, flatten layers and dense layers to complete the architecture of both our models and have achieved satisfactory results. We have also tested with various parameters and different structures until we finally arrived at our final model which gave us the best accuracy.

The fruits of Bangladesh have unique varying characteristics. Some have different shapes, and some have different colors. All these distinct characteristics help our neural network model to identify and classify the individual type of fruits very easily. And that is why we have used CNN architecture which can extract these unique features of fruits very easily and make good predictions about them. Thus, both models were easy to build following the algorithms and the model architecture given in Figs. 1 and 2.

4.4 Performance Evaluation

From previous discussions, we can say that VGG16 performs better in our system. VGG16 gives us 99.2% of accuracy in the I validation part, whereas the custom CNN model gives us 98.4% accuracy (Table 2). Also in validation loss, VGG166 gives us less value. The custom CNN model also gives high accuracy, but not more than the VGG16 model.

Table 2. Model accuracy and loss of custom model and VGG16

Model	Training accuracy	Training loss	Validation accuracy	Validation loss
Custom model with CNN	100%	0.0014	98.4%	0.0813%
VGG16	100%	0.0032	99.2%	0.0394

In our model, we have evaluated it in two ways, one is by the accuracy of the test set, and another is by giving an input image of fruit whether our model can detect it or not. Both model custom and VGG16 are giving an accuracy of 99% on the test set.

Here are the graphs showing accuracy and loss against 25 epochs of our custom model using deep learning (Fig. 3).

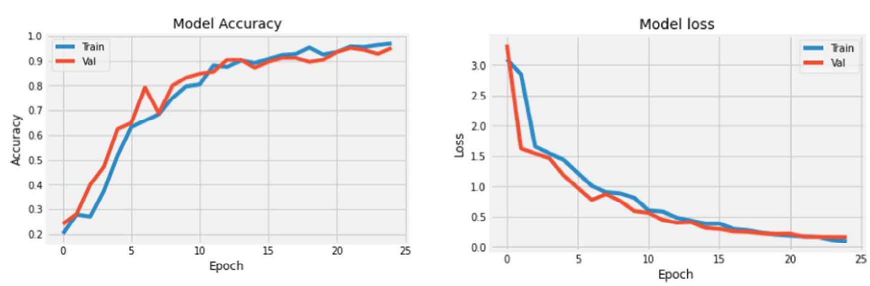

Fig. 3. Model accuracy and loss of custom model

Now, here are the graphs showing the accuracy and loss of the VGG16 model against 20 epochs (Fig. 4).

By comparing both models it can be seen that the VGG16 model takes fewer epochs to reach a high accuracy compared to the custom CNN model. However, the VGG16 model also takes ~61 s per epoch to train while the custom CNN model takes only ~14 s. So, the custom model can be trained much faster than the VGG16 model.

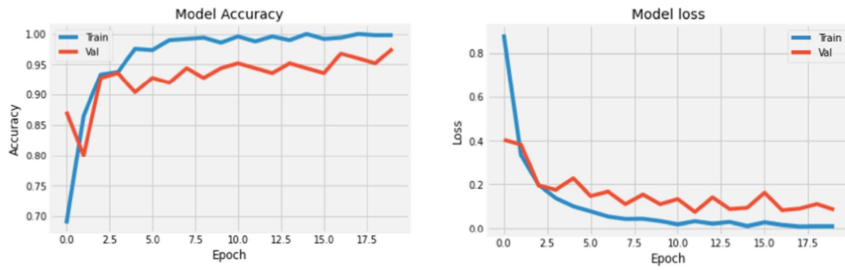

Fig. 4. Model accuracy and loss of VGG16

5 Conclusion

In this paper, we have applied deep learning to develop a custom VGG16 model to classify images of the five fruits of Bangladesh. The proposed method was successfully evaluated on images of these five different fruits: apple, orange, banana, watermelon, and jackfruit. And our models have successfully managed to achieve extremely high accuracy in classifying fruits (up to 99.5%). In most situations, the proposed approach was efficient in identifying the fruits. The constraint of this model is that it cannot classify multiple fruits in a single image as it was designed to only classify a single fruit. We intend to build image detection techniques for classifying multiple objects from a single image in the future. We can also improve our system's ability to classify by overlapping images of fruits and fine hyperparameter tuning. Furthermore, we plan to go beyond VGG16-based models and use transformer-based hybrid and state-of-the-art image classification techniques to detect a more diverse range of fruits in Bangladesh. Also, this research could be expanded to assist agriculturists in classifying diverse types of fruits in Bangladesh. If used properly our model could greatly benefit the agricultural community of Bangladesh.

References

1. Hossen, M.S., Arefin, M.S., Karim, R.: Developing a framework for fruits detection from images (2017). https://doi.org/10.1109/ECACE.2017.7913041
2. Shamim Hossain, M., Al-Hammadi, M., Muhammad, G.: Automatic fruit classification using deep learning for industrial applications. IEEE Trans. Ind. Inform. **15**(2) (2019). https://doi.org/10.1109/TII.2018.2875149
3. Behera, S.K., Rath, A.K., Sethy, P.K.: Fruit recognition using support vector machine based on deep features. Karbala Int. J. Mod. Sci. **6**(2) (2020). https://doi.org/10.33640/2405-609X.1675
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. Feng, J., Zeng, L., Liu, G., Si, Y.: Fruit recognition algorithm based on multi-source images fusion. Nongye Jixie Xuebao/Trans. Chin. Soc. Agric. Mach. **45**(2) (2014). https://doi.org/10.6041/j.issn.1000-1298.2014.02.013
6. Feng, J., Zeng, L., He, L.: Apple fruit recognition algorithm based on multi-spectral dynamic image analysis. Sensors (Switzerland) **19**(4) (2019). https://doi.org/10.3390/s19040949

7. Patel, H.N., Jain, D.R.K., Joshi, D.M.V.: Fruit detection using improved multiple features based algorithm. Int. J. Comput. Appl. **13**(2) (2011). https://doi.org/10.5120/1756-2395
8. Sreekanth, G.R., Thangaraj, P., Kirubakaran, S.: Fruit detection using improved K-means algorithm. J. Crit. Rev. **7**(12) (2020). https://doi.org/10.31838/jcr.07.12.02
9. Naranjo-Torres, J., Mora, M., Hernández-García, R., Barrientos, R.J., Fredes, C., Valenzuela, A.: A review of convolutional neural network applied to fruit image processing. Appl. Sci. (Switzerland) **10**(10) (2020). https://doi.org/10.3390/app10103443
10. Sahu, D., Dewangan, C.: Identification and classification of mango fruits using image processing. Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol. **2**(2), 203–210 (2017)
11. Chithra, P.L., Henila, M.: Fruits classification using image processing techniques. Int. J. Comput. Sci. Eng. **2**(2) (2019)
12. Joseph, J.L., Kumar, V.A., Mathew, S.P.: Fruit classification using deep learning. In: Mekhilef, S., Favorskaya, M., Pandey, R.K., Shaw, R.N. (eds.) Innovations in Electrical and Electronic Engineering, vol. 756, pp. 807–817. Springer, Singapore (2021). https://doi.org/10.1007/978-981-16-0749-3_62
13. Saranya, N., Srinivasan, K., Pravin Kumar, S.K., Rukkumani, V., Ramya, R.: Fruit classification using traditional machine learning and deep learning approach. In: Smys, S., Tavares, J.M.R.S., Balas, V.E., Iliyasu, A.M. (eds.) Computational Vision and Bio-Inspired Computing. AISC, vol. 1108, pp. 79–89. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-37218-7_10
14. Sa, I., Ge, Z., Dayoub, F., Upcroft, B., Perez, T., McCool, C.: Deepfruits: a fruit detection system using deep neural networks. Sensors (Switzerland) **16**(8) (2016). https://doi.org/10.3390/s16081222
15. Barbole, D.K., Jadhav, P.M., Patil, S.B.: A review on fruit detection and segmentation techniques in agricultural field. In: Chen, J.-Z., Tavares, J.M.R.S., Iliyasu, A.M., Du, K.-L. (eds.) Second International Conference on Image Processing and Capsule Networks, vol. 300, pp. 269–288. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-84760-9_24