

Automated Identification Algorithm Using CNN for Computer Vision in Smart Refrigerators

Pulkit Jain¹, Paras Chawla¹, Mehedi Masud^{2,*}, Shubham Mahajan³ and Amit Kant Pandit³

¹Chandigarh University, Mohali, 140413, India

²Department of Computer Science, College of Computers and Information Technology, Taif University, Taif, 21944, Saudi Arabia

³School of Electronics & Communication Engineering, Shri Mata Vaishno Devi University, Katra, 182320, India

*Corresponding Author: Mehedi Masud. Email: mmasud@tu.edu.sa

Received: 26 August 2021; Accepted: 18 October 2021

Abstract: Machine Learning has evolved with a variety of algorithms to enable state-of-the-art computer vision applications. In particular the need for automating the process of real-time food item identification, there is a huge surge of research so as to make smarter refrigerators. According to a survey by the Food and Agriculture Organization of the United Nations (FAO), it has been found that 1.3 billion tons of food is wasted by consumers around the world due to either food spoilage or expiry and a large amount of food is wasted from homes and restaurants itself. Smart refrigerators have been very successful in playing a pivotal role in mitigating this problem of food wastage. But a major issue is the high cost of available smart refrigerators and the lack of accurate design algorithms which can help achieve computer vision in any ordinary refrigerator. To address these issues, this work proposes an automated identification algorithm for computer vision in smart refrigerators using InceptionV3 and MobileNet Convolutional Neural Network (CNN) architectures. The designed module and algorithm have been elaborated in detail and are considerably evaluated for its accuracy using test images on standard fruits and vegetable datasets. A total of eight test cases are considered with accuracy and training time as the performance metric. In the end, real-time testing results are also presented which validates the system's performance.

Keywords: CNN; computer vision; Internet of Things (IoT); radio frequency identification (RFID); graphical user interface (GUI)

1 Introduction

'Smart home' is not a new concept now, as Internet of Things (IoT) is playing a great role in revolutionizing the way one ever thought of living in a home filled with sensors where every electronic appliance can talk to one another wirelessly. IoT has allowed the control and monitoring of electronic appliances in our homes with speech, text, and many other input methods from anywhere in the



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

world with the help of cloud platforms and smartphone applications. The kitchen is a standout amongst the most essential spots for smart home as it comprises numerous appliances that give better administration to the family. Thus incorporating IoT technology in kitchen appliances has brought significant changes leading to a more easy and modern lifestyle and serving as an aid for the household members [1].

As the kitchen appliances are to be used lifelong by the person and that's the reason people are ready to invest in these appliances without any second thought thereby creating a competition today among manufacturers to make these kitchen appliances smarter and smarter [2]. A refrigerator is one of those appliances used in every household to preserve perishable food items over a long period of time. Since the modern lifestyle is driving individuals to invest less energy in preparing healthy food at home, an enjoyable and sound way of living can be achieved with an appliance like 'Smart Refrigerator'. The refrigerator is one of the devices which have undergone several changes over the last two decades. It has evolved from being a cooling device to a smart device that has computer-like abilities incorporated in it [3]. To be able to think that a fridge could utilize Radio Frequency Identification (RFID) labels to identify items it contains and provide an expiry check on them seemed almost impossible a few decades back. But with technological enhancements, this scenario has been completely changed.

One of the most crucial tasks for any smart refrigerator is food item scanning and its correct identification [1]. It is evident from the literature that many smart refrigerators have been developed [2–13] for this core functionality using technologies like RFID scan, Quick Response (QR) code Scan, Image capture and processing, Fuzzy Logic, Artificial Intelligence (AI) and Computer Vision etc. The major objective is to avoid food wastage by early identification of food items that are near expiry via timely notification to the user via Graphical User Interface (GUI), Short Message Service (SMS) or email etc. Hsu et al. [2] developed a 3C smart system that makes the use of image processing, speech recognition, and speech broadcasting technique for food item identification and control respectively. The main features included a speech control system along with an auto dial system for ordering scarce items directly from the vendors. It also supported the wireless control of other interconnected home appliances. Zhang et al. [10] proposed a new approach for fruit recognition based on data fusion from multiple sources. Both weight information and data obtained from multiple CNN models were fused for improving accuracy in the recognition of fruits. With the advent of IoT technology, several authors have incorporated novel ways to send alerts to the user via emails and by using cloud servers to send data to dedicated mobile applications for remote access of databases as well. A similar smart refrigerator system was proposed by Nasir et al. [11] which focused on the expiry check using both weight information and odour detection using sensors like MQ3 and DHT11. It also incorporated the cloud platform Thingspeak for remote access of data along with the Pushbullet application for notification and alerts. A summary of literature review is presented in [Tab. 1](#) based on parameters like scanning mechanism and whether food expiry check and cloud platform is provided or not.

Table 1: Summary of literature review

| Ref. | Year | Scanning mechanism | Expiry check | Cloud platform | Notification to user |
|------|------|---------------------------------------|--------------|----------------|----------------------|
| [2] | 2010 | Image processing & speech recognition | No | No | GUI |
| [3] | 2012 | RFID & barcode | Yes | Yes | Mobile app |
| [4] | 2013 | RFID | No | Yes | Android app |
| [5] | 2013 | Fuzzy logic | No | No | GUI |
| [6] | 2015 | Weight sensing | No | No | Android app |
| [7] | 2017 | Image sensing: raspberryPi & webcam | No | Yes | Android app |
| [8] | 2017 | Weight sensing & RFID scan | Yes | No | SMS via GSM |
| [9] | 2017 | AI based | No | Yes | SMS and android app |
| [10] | 2018 | Deep learning | No | Yes | GUI |
| [11] | 2018 | Weight and odour (IoT) | Yes | Yes | SMS, email and app |
| [12] | 2020 | QR code scan | Yes | Yes | Android app |
| [13] | 2021 | IoT and image sensing | Yes | Yes | Android app |

The major issue associated with smart refrigerators available in market today is high cost [7] and the availability of only brand-specific applications for remote access of database of the items kept inside the fridge. The need of the hour is to design such algorithms for intelligent and cost effective systems which can add smartness to existing conventional refrigerators. The review carried out in [Tab. 1](#) talks about various scanning techniques used by researchers so far in the design of smart refrigerators. Apart from these techniques, many researchers have made the use of CNN for automated classification of fruits and vegetables. Kodors et al. [14] used CNN models like MobileNet version 1 and 2 on FRUITS360 dataset for recognition of apples and pears. Basri et al. [15] made the use of Tensor Flow platform for detection of mango and pitaya fruits using MobileNet CNN model by testing on self-created dataset. Huang et al. [16] carried out testing of InceptionV3 model on FRUITS360 dataset comprising of 81 classes of fruits and vegetables using adam optimizer and achieved an accuracy of 96.5%. In the similar way, Femling et al. [17] made the use of Raspberry Pi (RPi), load cell and camera module to perform training of Inception and MobileNet model on a self-created dataset comprising of 400 images each of 10 classes of selected fruit items. Ashraf et al. [18] carried out testing using InceptionV3 model and presented a detailed comparison in accuracy values obtained using different

loss and optimization functions. The maximum accuracy obtained is 87.08% using cross entropy loss function and adagrad optimization function.

After literature review, one can easily point out the fact that no paper can be found which talks about designing of an intelligent module which can turn any ordinary refrigerator into a smart refrigerator. Moreover, no research article mentions about placing the weight measurement system and cameras outside the fridge as it can help avoiding the mess of wiring inside the refrigerator. To avoid these challenges and to fill the research gap, this paper proposes an automated identification algorithm for Computer Vision in Smart Refrigerators using standard CNN architectures. This paper carries forward the work done previously in the area of CNN for fruits and vegetables classification using improved CNN models namely InceptionV3 and MobileNetV3 on standard datasets. The paper is organized as below: Section 2 talks about the design of an intelligent module for ordinary to smart refrigerator conversion for achieving the task of automatic recognition of fruits and vegetables. The proposed module is portable and cost effective comprising of fruits and vegetables image scanning and weight sensing mechanism outside the refrigerator. Section 3 and 4 talks about standard datasets selected for training of the system using InceptionV3 and MobileNetV3 CNN models. The experimental results have been depicted in Section 5 followed by conclusion and future scope of the work at the end.

2 Intelligent Module Design and Working

To address the challenge of mess of wiring inside the refrigerator, the module was designed to avoid any wiring or modification required inside any compartment of refrigerator. The block diagram of the entire system is depicted in Fig. 1. It comprises of three major blocks i.e., Intelligent Module section, refrigerator with attached display screen and cloud server. The role of intelligent module which is in form of a portable trolley system is camera scanning for food item identification and noting down the weight readings via load cell (label 'D') attached at the bottom of weight sensing area (label 'F'). The camera sensing sub module consists of RPi Camera (label 'A') mounted on an L shaped arm which upon power up comes into position shown in Fig. 1 controlled via two servo motors (label 'B and 'E'). The camera module clicks the images of the food item when it is placed on weight sensing area depicted in Fig. 1. The Central Processing Unit (CPU) and weight sensing sub module consists of RPi (label 'C') which acts as the CPU of the system and a load cell of 200 Kg for sensing weight of item placed. With the help of these two sub modules the name of food item recognized along with weight readings are obtained and further sent to cloud server as well as display screen attached to the refrigerator.

The trolley system has stopper wheels (label 'I') and moreover its height can be easily adjusted up and down as per user requirement using screw arrangement (label 'G'). An Ultra Violet (UVC) disinfection box (label 'H') can also be attached at the bottom of trolley system which can work in standalone mode to disinfect food items and other daily use items like keys, wallet, mobile phones etc. to provide safety against the spread of viruses and bacteria. The data containing the food item name and weight information can then be passed onto cloud server i.e., Google Firebase which can be further accessed remotely using android application developed namely 'Fridge Assistant' on any smartphone as depicted in Fig. 2. The same database can also be displayed on touch screen which can be easily attached on the front door of refrigerator and it requires only single connection with intelligent module via touch screen connector shown in Fig. 1. The 'Fridge Assistant' android application as depicted in Fig. 2b gets real time updates using IoT as the items are stored in the refrigerator. The weight reading along with date and time stamp is also noted which provides a way to keep a check on expiry of items and sending alerts accordingly to the user to consume the item before a fixed stipulated time. Notes

can also be added as depicted in Fig. 2c. Moreover a shopping list is automatically created of scarce items which get added in shopping list tab of the application.

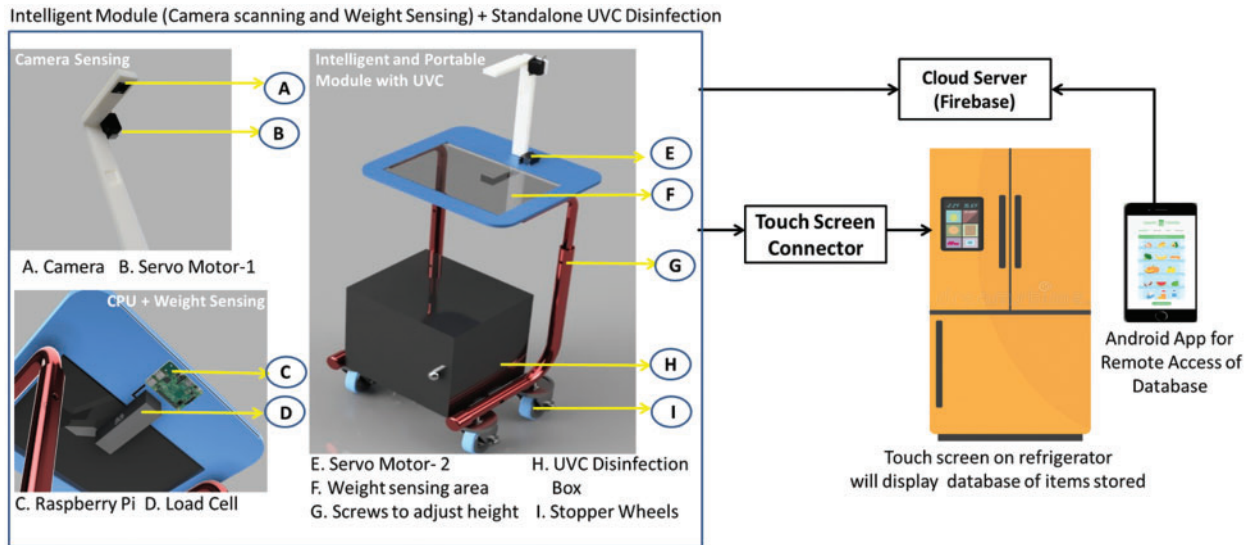


Figure 1: Detailed block diagram of complete system

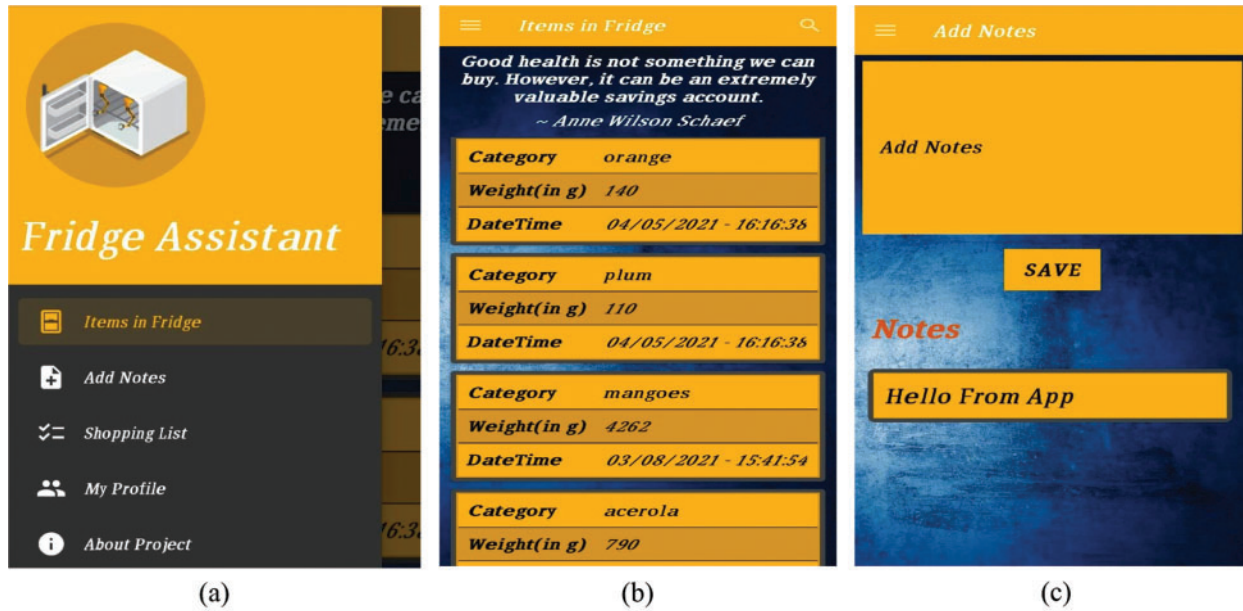


Figure 2: Fridge assistant application for remote access of database (a) home page (b) items stored with date and timestamp (c) add notes feature

3 Selected Dataset

There are many datasets available as open source to train the module. The standard datasets considered for this work are explained below.

3.1 FIDS30

FIDS30 dataset [19] is a small dataset comprising of a total of 30 different classes of fruits and 971 images in total. Each fruit class consists of 32 very diverse images in Joint Photographic Experts Group (JPEG) format including single fruit image, multiple fruits image of same kind and some images with noise such as leaves, plates, hands, trees and other noisy backgrounds. Certain classes of fruits included in this dataset are apples, bananas, cherries, coconuts, grapes, lemons, guava, oranges, kiwifruit, tomatoes, pomegranates, watermelons and strawberries etc. It is provided by Visual Cognitive Systems Lab and is publicly available for use and download.

3.2 FRUITS360

FRUITS360 [20] is one of the popular and a very huge dataset available as open source on Kaggle platform. It comprises of color images of size 100×100 pixels with a total of 67,692 training set images and 22,688 test set images. This dataset consists of 131 different fruits and vegetable classes with a total of 90,483 images.

4 CNN Models

CNN's are by far the most widely used models for training such problems of food item identification. They have been applied in providing solutions to numerous complex problems involving image classification in medical fields, design and optimization problems related to reconfigurable Radio Frequency circuits [21,22]. But they are now playing a major role in almost every object detection and related computer vision tasks. In order to understand CNN in detail one must have a general idea of a single layer CNN. A single layer CNN is explained as follows: If layer l is a convolutional layer, then one can calculate the output size of single convolutional layer see Eq. (3) from applied filter and input using the following Eqs. (1) and (2):

$$\text{Input} = \text{height} \times \text{width} \times \text{no of channels} = n_h^l \times n_w^l \times n_c^{l-1} \quad (1)$$

$$\text{Each Filter} = \text{height of filter} \times \text{width of filter} \times \text{no of channels} = f^l \times f^l \times n_c^{l-1} \quad (2)$$

$$\text{Output} = \text{height} \times \text{width} \times \text{no of filters} = n_h^l \times n_w^l \times n_c^l \quad (3)$$

Apart from input and output layer, a complete CNN consists of numerous hidden layers which further consist of convolution, softmax, pooling and fully connected layers. The most preferred CNN models for image recognition are Inception CNN and MobileNet CNN as they both are pre-trained networks. The detailed description about these two models is given in following subsections:

4.1 Inception CNN

GoogLeNet or InceptionV1 is a pre-trained and widely used deep convolutional neural network for image recognition applications [23]. The heart of inception network is the inception module block as depicted in Fig. 3. The entire InceptionV1 network comprised of nine repetitions of this inception module along with addition of fully connected layers and soft max layers at intermediate stages. The inception module comprises of previous activation layer which is first passed through bottleneck layer of 1×1 convolutions. The major computational cost savings are achieved at this layer before passing through expensive 3×3 and 5×5 convolutions. At the end all channels are stacked up using channel

concatenation. InceptionV2 network further provided cost savings in computation leading to improved accuracy using concept of factorized convolutions and by expanding the filter banks [24]. Further upgrades were carried out resulting in a better and accurate InceptionV3 network which has been used in this paper due to its better performance and low error rates.

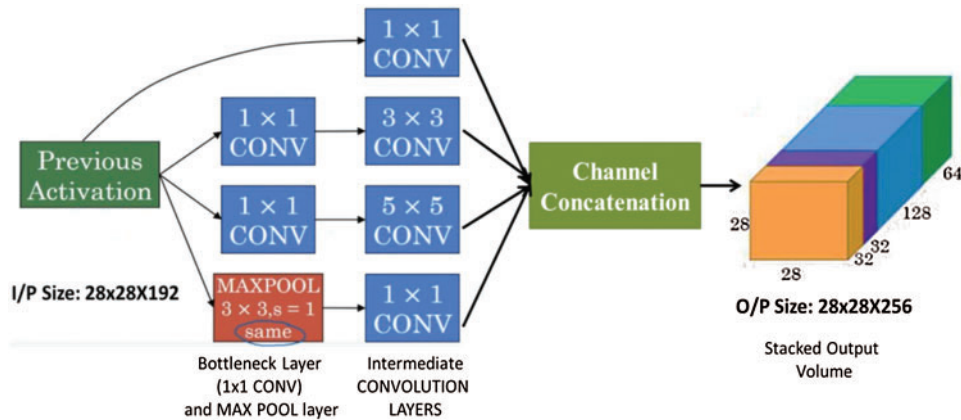


Figure 3: Basic inception module-heart of inceptionV3 Network

4.2 MobileNet CNN

A lot of classic neural networks including LeNet-5 [25], Alex-Net [26], VGG-16 and even powerful neural nets like Residual Neural Net (ResNet) [27], InceptionV3 are computationally very expensive. Moreover in order to run the neural network model on the system proposed in this paper having a less powerful CPU or Graphical Processing Unit (GPU), the best choice is MobileNet neural architecture. It is the network best preferred for mobile and computer vision related applications in embedded systems. With the development of MobileNetV1 in 2017 a new research area opened up in the use of deep learning in machine vision i.e., to design similar models which can run even in sophisticated embedded systems.

The computational cost is given by product of number of filter parameters, number of filter positions and number of filters. As per the formula, one can easily compute the cost summary for both convolution approaches i.e., normal convolution and depthwise separable convolution using the parameter values depicted in Fig. 4. For the parameters depicted, it can easily be obtained that the cost of normal convolution is 2,160 multiplications, whereas it is only 672 (432-depthwise and 240 pointwise) in case of convolution approach supported by MobileNet architecture. Depth wise separable convolution approach involves two main steps namely depth wise convolution and pointwise convolution. This approach can be designed to have similar inputs and output dimensions as normal convolution but it can be done at a much lower computational cost i.e., approximate 10 times more savings in computations. A more improved version MobileNetV2 [28], developed in 2018 further reduced the computational cost by adding a bottleneck block. This block comprises of a residual connection similar to ResNet and non-residual part comprised of additional expansion layer followed by depth wise separable convolution. The expansion layer increases the size of representation allowing the neural net to learn more features. Further at the end since it has to be deployed to a mobile device with memory constraints it is compressed down to smaller representation using projection or point wise convolution operation. The latest version MobileNetV3 [29] has been used in this work which

further improves the performance with addition of squeeze and excitation layers in basic MobileNetV2 version.

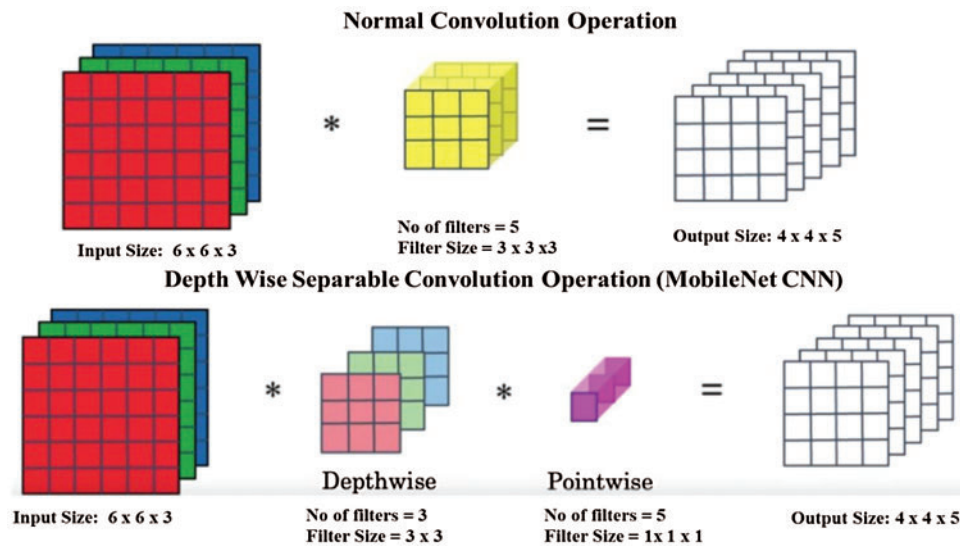


Figure 4: Comparison between normal and depthwise separable convolution operation

5 Experimental Results

The training accuracy results of previous related works using various CNN Models and datasets is depicted in [Tab. 2](#).

Table 2: Training accuracy results of previous related works

| Ref. | Dataset | Model(s) used | Classes | Images | Accuracy (%) |
|------|--------------|-------------------------------|---------|----------|--------------|
| [14] | FRUITS360 | MobileNetV1,V2 | 120 | 20622 | 99.8 |
| [15] | Self-created | MobileNet | 2 | 700 each | 70.6 |
| [16] | FRUITS360 | InceptionV3 | 81 | 55244 | 96.5 |
| [17] | Self-created | Inception, mobileNet | 10 | 400 each | 96 & 97 |
| [18] | Self-created | InceptionV3-transfer learning | 8 | 1868 | 87.08 |
| [30] | FIDS30 | ResNet-50 | 30 | 971 | 89.16 |
| [30] | FRUITS360 | ResNet-50 | 77 | 55244 | 99.5 |

A total of eight test cases were considered using InceptionV3 and MobileNetV3 CNN models as depicted in [Tab. 3](#). The test case ‘FIDS30-selected’ dataset comprises of only 9 most common and easily available fruit classes like apples, bananas, lemons, mangoes, oranges, pomegranates, strawberries, tomatoes and watermelons. Similarly, FRUITS360-selected dataset comprises of only

35 most common fruits and vegetables classes like apples, bananas, onions, cauliflower, ginger, lemon, mangoes, tomato, strawberry and watermelon. These test cases were selected so as to see the variations in accuracy and training time values using only selected items out of the entire dataset. Google Colab platform with time limited GPU support has been used to train the model running the Python script.

Table 3: Accuracy Results of CNN model testing used in the present work

| Test case | CNN model | Dataset | No of classes | Total images | Training time (in minutes) | Accuracy on test images (%) |
|-----------|-------------|--------------------|---------------|--------------|----------------------------|-----------------------------|
| Case 1 | InceptionV3 | FIDS30 | 30 | 971 | 168 | 89.3 |
| Case 2 | InceptionV3 | FIDS30-selected | 9 | 331 | 166 | 92.9 |
| Case 3 | InceptionV3 | FRUITS360 | 131 | 90,483 | 415 | 94.9 |
| Case 4 | InceptionV3 | FRUITS360-selected | 35 | 16,945 | 187 | 98.4 |
| Case 5 | MobileNetV3 | FIDS30 | 30 | 971 | 118 | 89.7 |
| Case 6 | MobileNetV3 | FIDS30-selected | 9 | 331 | 117 | 96.4 |
| Case 7 | MobileNetV3 | FRUITS360 | 131 | 90,483 | 361 | 99.9 |
| Case 8 | MobileNetV3 | FRUITS360-selected | 35 | 16,945 | 87 | 99.9 |

In all the test cases, the ratio of the training to validation set images is kept as 80% by 20%. The loss function, optimizer and activation function used for training of both models in the present work are cross entropy, gradient descent and Rectified Linear Units (ReLU) respectively. The accuracy values of each test case listed in Tab. 3 are obtained from graphs shown below. The accuracy v/s numbers of iterations graph in Fig. 5a shows the variations of training and validation accuracy of InceptionV3 model on FIDS30 dataset. The final validation accuracy depicted by blue line as obtained from graph is found to be 89.3%. Similarly, the loss or cross entropy depicted in Fig. 5b has a decreasing curve which reaches near zero value with increase in number of iterations.

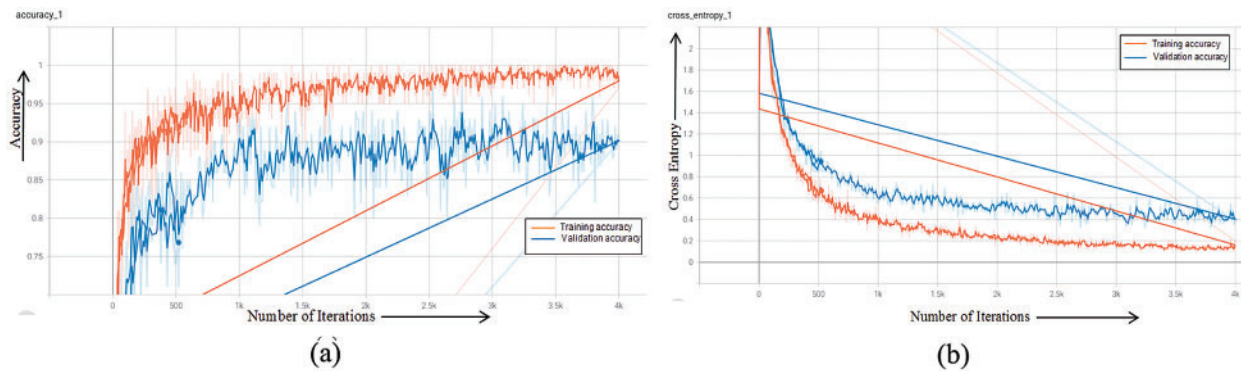


Figure 5: (a) Accuracy and (b) cross-entropy of inceptionV3 v/s number of iterations using FIDS30

In the 2nd test case using FIDS-30 with selected data items, very large variations are observed in the training and validation accuracy lines resulting in higher loss and lower accuracy value of 92.9% as shown in Fig. 6.

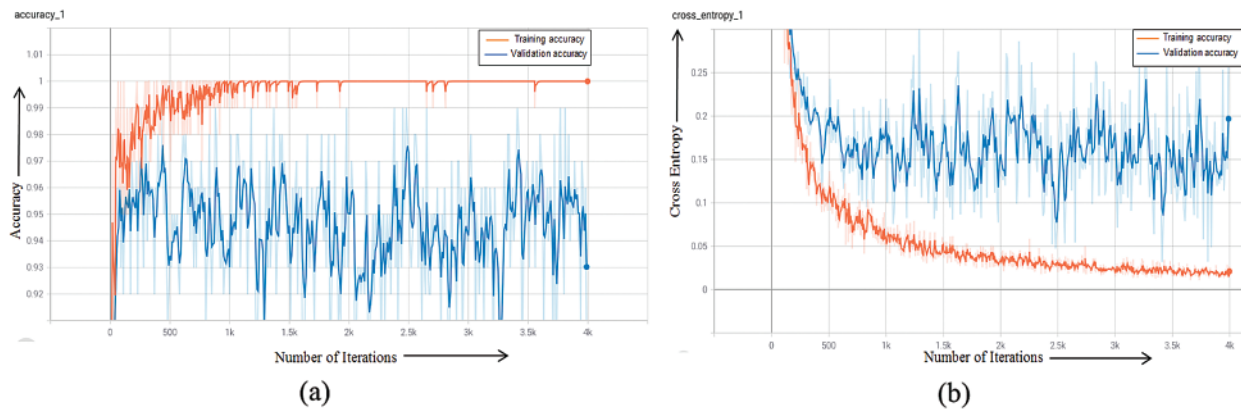


Figure 6: (a) Accuracy and (b) cross-entropy of the inceptionV3 v/s number of iterations using FIDS30-selected

The graphs in Fig. 7 show very smooth variations in both accuracy and loss value with increase in number of iterations.

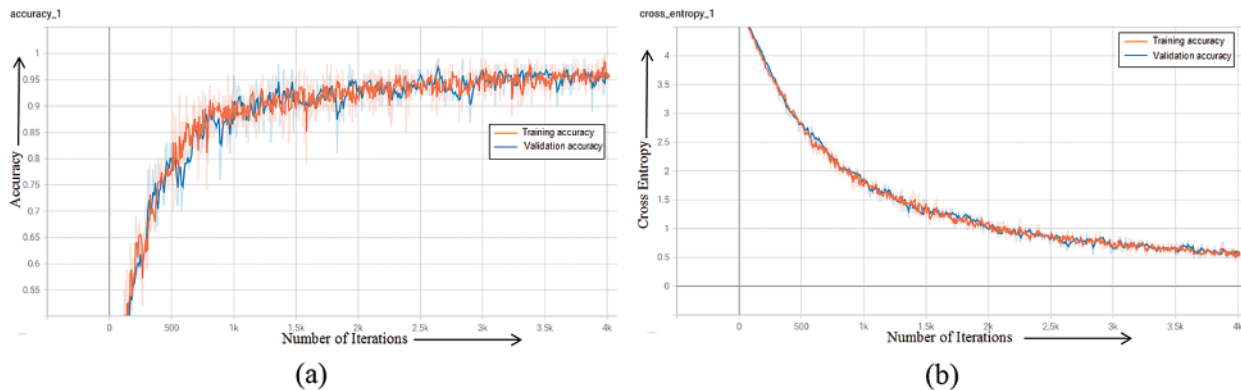


Figure 7: (a) Accuracy and (b) cross-entropy of inceptionV3 v/s number of iterations using FRUITS360

It can also be observed that both training and validation accuracy lines are very close to each other thus resulting in high accuracy value of 94.9% using FRUITS-360 dataset. A similar graph but with slight variations is obtained in Fig. 8 resulting in an accuracy value of 98.4% using FRUITS-360 selected dataset.

The last four test cases take into consideration the MobileNetV3 CNN model. The graphs shown in Fig. 9 depict very large variations between validation and training accuracy, with the final validation accuracy coming out to be 89.7%. In the test case-6 shown in Fig. 10 using FIDS-30 with selected data items, the accuracy achieved is 96.4% which is better than obtained using Inception-V3 model.

The graphs in Fig. 11 show very smooth variations in both accuracy and loss value with increase in number of iterations. It can also be observed that both training and validation accuracy lines are following each other thus resulting in highest accuracy value of 99.9% using FRUITS-360 daset. A

similar graph with similar accuracy value but with slight more variations in the initial iterations is obtained in Fig. 12 using FRUITS360-selected dataset.

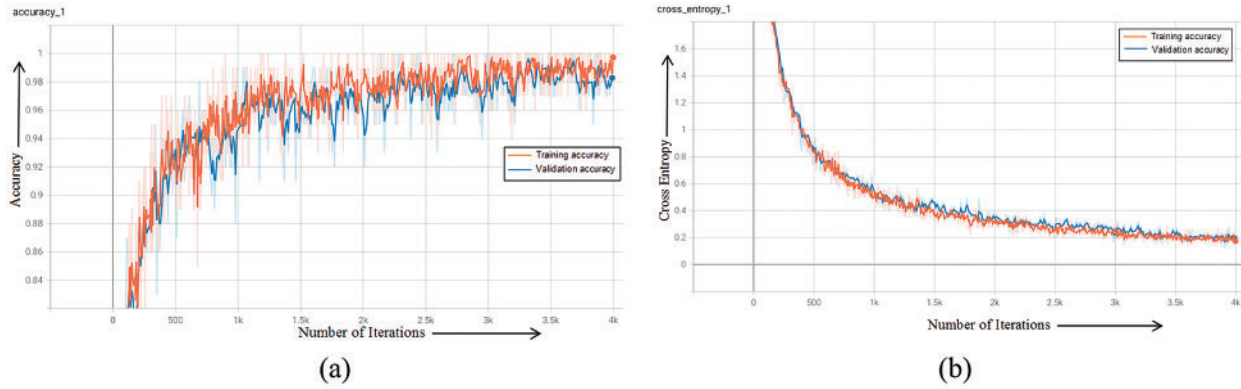


Figure 8: (a) Accuracy and (b) cross-entropy of inceptionV3 v/s number of iterations using FRUITS360-selected

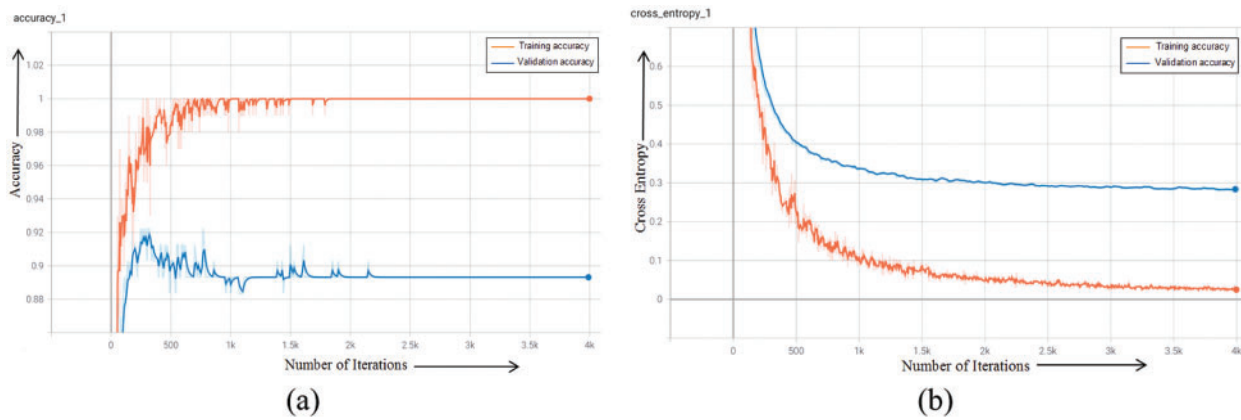


Figure 9: (a) Accuracy and (b) cross-entropy of mobileNetV3 v/s number of iterations using FIDS 30

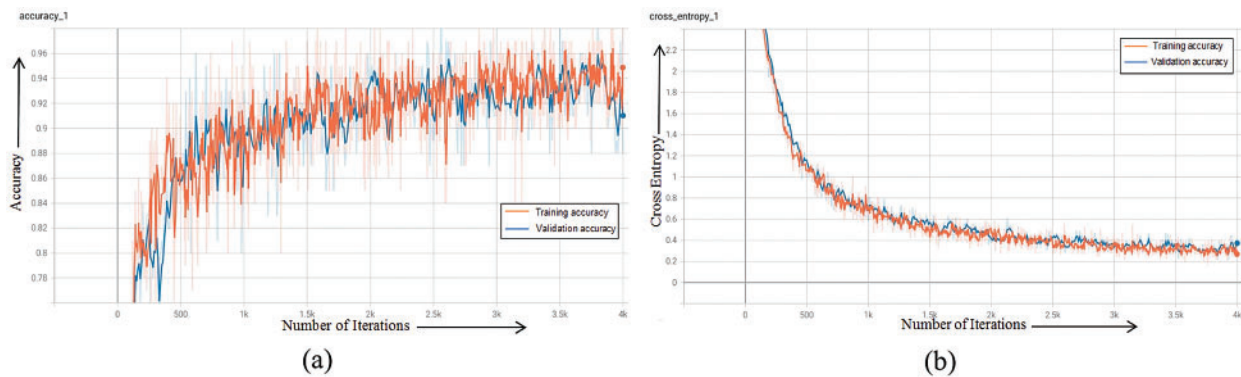


Figure 10: (a) Accuracy and (b) cross-entropy of mobileNetV3 v/s number of iterations using FIDS30-selected

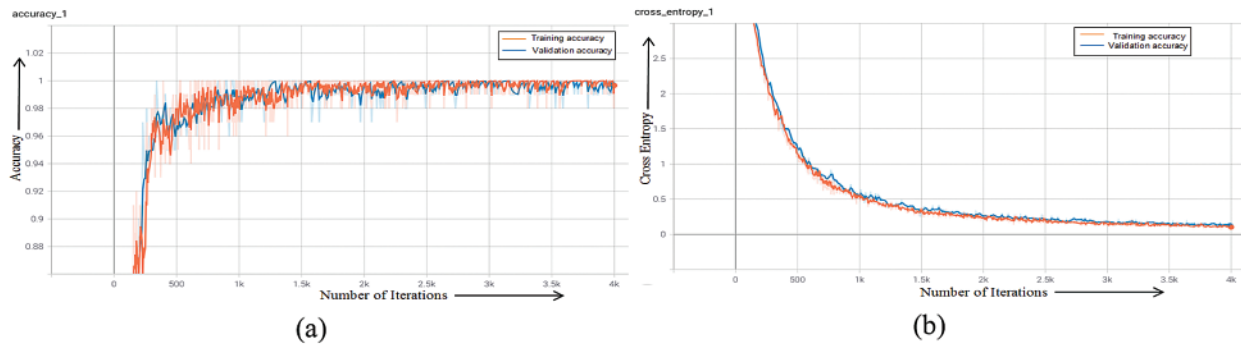


Figure 11: (a) Accuracy and (b) cross-entropy of mobileNetV3 v/s number of iterations on FRUITS360

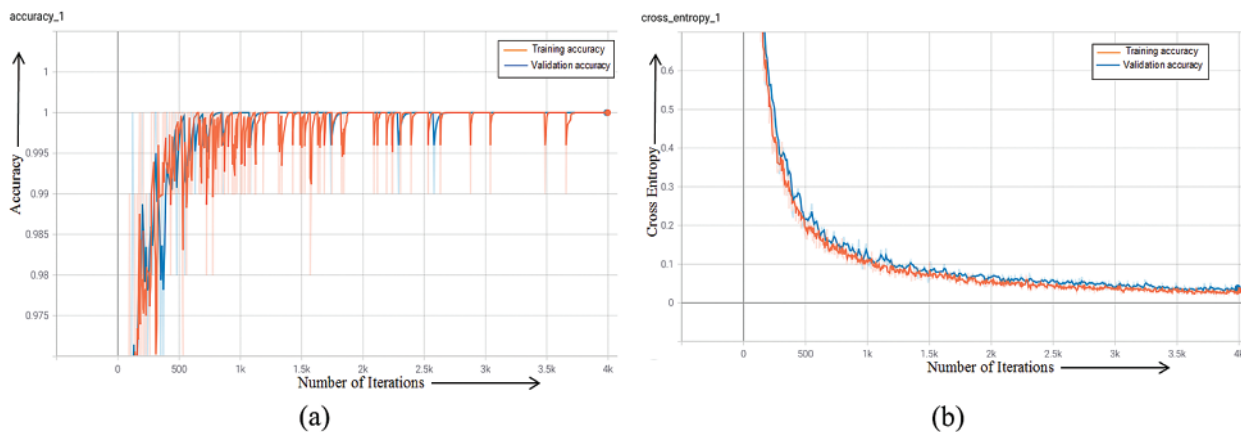


Figure 12: (a) Accuracy and (b) cross-entropy of mobileNetV3 v/s number of iterations using FRUITS360-selected

The graphical representation of the entire data tabulated in [Tab. 3](#) is presented in [Fig. 13](#). All the test cases considered are labeled on the x-axis and [Fig. 13a](#) represents the training time variation (in minutes) with respect to number of classes and total images in each test case. On other hand [Fig. 13b](#) represents the accuracy comparison (in %) among all test cases.

One can clearly observe from the graph that MobileNet model gives better results in all its four test cases in terms of shorter training times and better accuracy values. A comparison is also drawn in [Tab. 4](#) between the accuracy results obtained of present and previous related works. One can easily conclude that the accuracy obtained using the approach and models in the present work are clearly higher than that obtained in previous related works. Both InceptionV3 and MobileNetV3 clearly outperform the other CNN models used in previous works like ResNet, MobileNetV1, V2 etc. Even in case of self-created datasets the accuracy obtained as listed in [Tab. 2](#) of previous works is still lower than results obtained in the current work. Although the FRUITS360 dataset used in the current work is more diverse but still maximum accuracy of 99.9% is obtained using MobileNetV3, which clearly indicates that performance of the current work is far better than previous similar studies.

Several test images from the datasets were considered for the evaluation of the two trained CNN models. The test image considered, identification result containing top five results with accuracy values are depicted in [Tab. 5](#). It also depicts the item identified along with accuracy value in %.

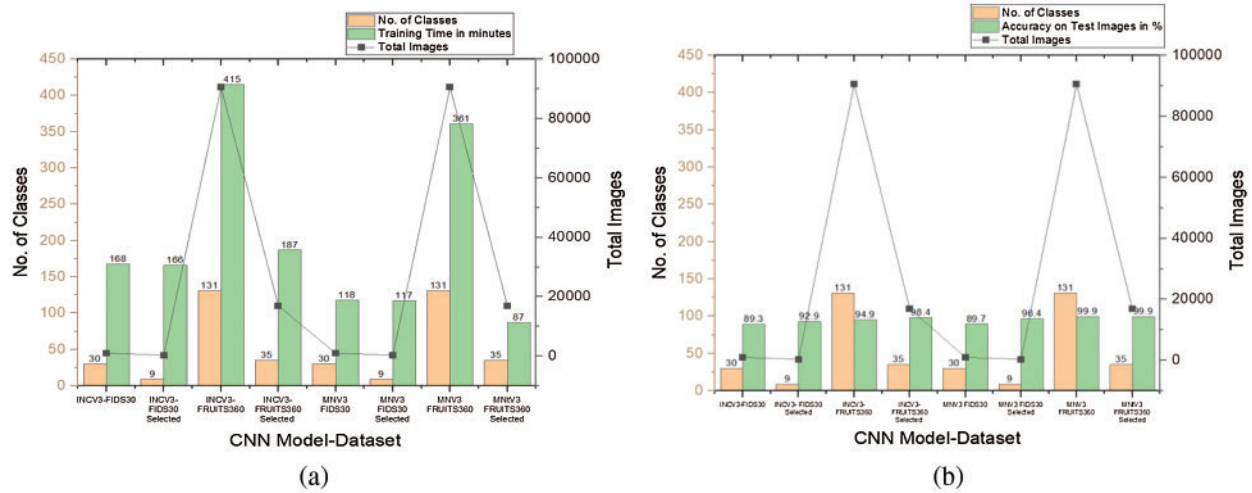


Figure 13: Comparison between (a) training Time (b) accuracy of eight test cases

Table 4: Comparison between accuracy results of previous and present work

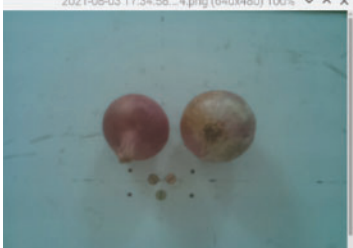
| Dataset | Previous work CNN model | Present work CNN model | Previous work accuracy (%) | Present work accuracy (%) |
|-----------------------|----------------------------|---------------------------|-------------------------------|------------------------------|
| FIDS30 | ResNet-50 | InceptionV3 | 89.16 | 89.30 |
| FRUIT360- selected | InceptionV3 | InceptionV3 | 96.5 | 98.4 |
| FRUIT360 | ResNet-50 | MobileNetV3 | 99.5 | 99.9 |
| FRUIT360 | MobileNetV1,V2 | MobileNetV3 | 99.8 | 99.9 |

The real time testing for fruits and vegetables classification from images is carried out on the proposed intelligent module containing RPi as the CPU. The results obtained are tabulated in Tab. 6 which depicts the real time image captured using 5 Megapixel resolution RPi camera module. It also shows the snapshots of identification result obtained on RPi Console followed by name of the identified food item. One can easily observe that all the items are correctly identified with the designed algorithm.

Table 5: Results of food item identification using sample test images

| Test image | Identification result | Item identified | Accuracy (in%) |
|---|---|-----------------|----------------|
|  | apple 0.9541986 pear 0.019847937 plum 0.0147903925 peach 0.0024664435 cherry 0.001903337 | Apple | 95.4 |
|  | apples 0.7312254 mangos 0.15027477 pomegranates 0.0805094 lemons 0.016639773 tomatoes 0.009054236 | Apple | 73.12 |
|  | bananas 0.9985536 mangos 0.0006435459 oranges 0.00018307295 watermelons 0.00014534965 pomegranates 0.00014009421 | Bananas | 99.85 |
|  | banana 0.9975829 mangoes 0.00052511576 pear 0.0002531129 apricot 0.00017490916 lemon 0.00015854815 | Bananas | 99.75 |
|  | lemons 0.9167958 oranges 0.082105055 mangos 0.0003572211 strawberries 0.00016873826 bananas 0.00016409573 | Lemons | 96.14 |
|  | lemon 0.99130213 limes 0.0059455046 orange 0.0009401499 avocado 0.0007495653 pear 0.00047933374 | Lemon | 99.13 |
|  | mangos 0.9537991 lemons 0.024002964 watermelons 0.012246129 apples 0.002726292 tomatoes 0.0025224928 | Mangoes | 95.3 |
|  | onion red 0.9270475 onion red peeled 0.06481003 apple red 1 0.0024105315 apple red yellow 1 0.0015849759 onion white 0.0010464309 | Onionred | 92.7 |
|  | tomatoes 0.9702776 strawberries 0.009189759 mangos 0.0071411985 pomegranates 0.003958951 apples 0.0034287476 | Tomatoes | 97.02 |
|  | tomatoes 0.99017966 apples 0.009055005 watermelons 0.0003363353 bananas 0.0001357901 mangos 0.00012824501 | Tomatoes | 99.01 |

Table 6: Results of food item identification using real images

| Real image captured using 5MP RPi camera | Identification result | Item identified |
|---|--|------------------|
|  | <pre>Processing Images. Please wait.. Arm down !! /home/pi/Smart_Fridge/captured_images/Add/2021-08-03 lemons Added Successfully in Firebase 622 Updated Successfully in Firebase /home/pi/Smart_Fridge/captured_images/Add/2021-08-03</pre> | Lemons |
|  | <pre>Updated Successfully in Firebase /home/pi/Smart_Fridge/captured_images/Add/2021-08-03 onion red peeled Added Successfully in Firebase 3326</pre> | Onion red peeled |
|  | <pre>Arm down !! /home/pi/Smart_Fridge/captured_images/Add/2021-08-03 mangoes 3067 Updated Successfully in Firebase</pre> | Mangoes |
|  | <pre>Added Successfully in Firebase /home/pi/Smart_Fridge/captured_images/Add/2021-08-03 tomatoes</pre> | Tomatoes |

6 Conclusion

The design of an intelligent module for automated identification of food items in particular fruits and vegetables for achieving the task of computer vision in smart refrigerators are proposed. The designed module and algorithm has been considerably evaluated for its accuracy by using pre-trained InceptionV3 and MobileNetV3 CNN models on standard fruits and vegetables dataset. Out of the two CNN models considered, it is evident from the results that MobileNetV3 CNN clearly outperformed the InceptionV3 model in terms of training time as well as the accuracy obtained with test images. A huge amount of training time approximately 45 min on an average is saved with the usage of a very light CNN network like MobileNetV3. Moreover a very high accuracy value of about 99.9% is achieved and that too on a bigger dataset like FRUITS360. Finally the results obtained from real time testing with fruits and vegetables clearly validate the performance of the system proposed. The proposed design algorithm mentions about the touch screen display for giving updates to the user about the data items stored. In future, it can be implemented with real system along with addition of more test cases to further validate the systems performance by enhancing the dataset.

Funding Statement: This work was supported by Taif University Researchers Supporting Project (TURSP) under number (TURSP-2020/10), Taif University, Taif, Saudi Arabia.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] R. Siddiqi, "Effectiveness of transfer learning and fine tuning in automated fruit image classification," in *Proc. 3rd Int. Conf. on Deep Learning Technologies*, Xiamen, China, pp. 91–100, 2019.
- [2] C. L. Hsu, S. Y. Yang and W. B. Wu, "3C intelligent home appliance control system—example with refrigerator," *Expert Systems with Applications*, vol. 37, no. 6, pp. 4337–4349, 2010.
- [3] J. Rouillard, "The pervasive fridge—a smart computer system against uneaten food loss," in *Seventh Int. Conf. on Systems*, Saint-Gilles, Réunion, pp. 135–140, 2012.
- [4] L. Xie, Y. Yin, X. Lu, B. Sheng and S. Lu, "Ifridge: an intelligent fridge for food management based on RFID technology," in *Proc. ACM Int. Joint Conf. on Pervasive and Ubiquitous Computing*, Zurich, Switzerland, pp. 291–294, 2013.
- [5] B. Bostanci, H. Hagrais and J. Dooley, "A neuro fuzzy embedded agent approach towards the development of an intelligent refrigerator," in *IEEE Int. Conf. on Fuzzy Systems*, Hyderabad, India, pp. 1–8, 2013.
- [6] H. Gürüler, "The design and implementation of a GSM based user-machine interacted refrigerator," in *Int. Symp. on Innovations in Intelligent Systems and Applications*, Madrid, Spain, pp. 1–5, 2015.
- [7] H. H. Wu and Y. T. Chuang, "Low-cost smart refrigerator," in *IEEE Int. Conf. on Edge Computing*, Honolulu, HI, USA, pp. 228–231, 2017.
- [8] S. Qiao, H. Zhu, L. Zheng and J. Ding, "Intelligent refrigerator based on internet of things," in *IEEE Int. Conf. on Computational Science and Engineering and IEEE Int. Conf. on Embedded and Ubiquitous Computing*, Guangzhou, China, pp. 406–409, 2017.
- [9] A. S. Shweta, "Intelligent refrigerator using artificial intelligence," in *11th Int. Conf. on Intelligent Systems and Control*, Coimbatore, India, pp. 464–468, 2017.
- [10] W. Zhang, Y. Zhang, J. Zhai, D. Zhao, L. Xu *et al.*, "Multi-source data fusion using deep learning for smart refrigerators," *Computers in Industry*, vol. 95, pp. 15–21, 2018.
- [11] H. Nasir, W. B. W. Aziz, F. Ali, K. Kadir and S. Khan, "The implementation of IoT based smart refrigerator system," in *2nd Int. Conf. on Smart Sensors and Application*, Kuching, Malaysia, pp. 48–52, 2018.
- [12] H. Almurashi, B. Sayed and R. Bouaziz, "Smart expiry of food tracking system," *In Advances on Smart and Soft Computing*, 1st ed., vol. 1188, Springer, Singapore, pp. 541–551, 2021.

- [13] A. Sharma, A. Sarkar, A. Ibrahim and R. K. Sharma, "Smart Refri: smart refrigerator for tracking human usage and prompting based on behavioral consumption," *In Emerging Technologies for Smart Cities*, 1st ed., vol. 765, Springer, Singapore, pp. 45–54, 2021.
- [14] S. Kodors, G. Lacis, V. Zhukov and T. Bartulsons, "Pear and apple recognition using deep learning and mobile," in *Proc. 19th Int. Scientific Conf. Engineering for Rural Development*, Jelgava, Latvia, pp. 1795–1800, 2020.
- [15] H. Basri, I. Syarif and S. Sukaridhoto, "Faster R-cNN implementation method for multi-fruit detection using tensorflow platform," in *Int. Electronics Symposium on Knowledge Creation and Intelligent Computing*, Bali, Indonesia, pp. 337–340, 2018.
- [16] Z. Huang, Y. Cao and T. Wang, "Transfer learning with efficient convolutional neural networks for fruit recognition," in *IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conf.*, Chengdu, China, pp. 358–362, 2019.
- [17] F. Femling, A. Olsson and F. A. Fernandez, "Fruit and vegetable identification using machine learning for retail applications," in *14th Int. Conf. on Signal-Image Technology & Internet-Based Systems*, Las Palmas de Gran Canaria, Spain, pp. 9–15, 2018.
- [18] S. Ashraf, I. Kadery, M. A. A. Chowdhury, T. Z. Mahbub and R. M. Rahman, "Fruit image classification using convolutional neural networks," *International Journal of Software Innovation*, vol. 7, no. 4, pp. 51–70, 2019.
- [19] Z. M. Khaing, Y. Naung and P. H. Htut, "Development of control system for fruit classification based on convolutional neural network," in *IEEE Conf. of Russian Young Researchers in Electrical and Electronic Engineering*, Moscow and St. Petersburg, Russia, pp. 1805–1807, 2018.
- [20] H. B. Ünal, E. Vural, B. K. Savaş and Y. Becerikli, "Fruit recognition and classification with deep learning support on embedded system (fruitnet)," in *Innovations in Intelligent Systems and Applications Conf.*, Istanbul, Turkey, pp. 1–5, 2020.
- [21] P. Chawla and R. Khanna, "A novel approach of design and analysis of fractal antenna using neuro-computational method for reconfigurable RF MEMS antenna," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 24, no. 3, pp. 1265–1278, 2016.
- [22] P. Chawla and R. Khanna, "Design and optimization of RF MEMS switch for reconfigurable antenna using feed-forward back-propagation ANN method," in *Nirma University Int. Conf. on Engineering*, Ahmedabad, India, pp. 1–6, 2012.
- [23] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed *et al.*, "Going deeper with convolutions," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Boston, MA, USA, pp. 1–9, 2015.
- [24] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, "Rethinking the inception architecture for computer vision," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, pp. 2818–2826, 2016.
- [25] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," in *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [26] R. A. Rahmat and S. B. Kutty, "Malaysian food recognition using alexnet CNN and transfer learning," in *11th IEEE Symp. on Computer Applications & Industrial Electronics*, Penang, Malaysia, pp. 59–64, 2021.
- [27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, pp. 770–778, 2016.
- [28] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L. C. Chen, "Mobilenetv2: inverted residuals and linear bottlenecks," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, pp. 4510–4520, 2018.
- [29] A. Howard, M. Sandler, G. Chu, L. C. Chen, B. Chen *et al.*, "Searching for mobilenetv3," in *IEEE/CVF Int. Conf. on Computer Vision*, Seoul, South Korea, pp. 1314–1324, 2019.
- [30] K. Munir, A. I. Umar and W. Yousaf, "Automatic fruits classification system based on deep neural network," *NUST Journal of Engineering Sciences*, vol. 13, no. 1, pp. 37–44, 2020.