

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/334881487>

Multi class fruit classification using efficient object detection and recognition techniques

Article in International Journal of Image, Graphics and Signal Processing · August 2019

DOI: 10.5815/ijigsp.2019.08.01

CITATIONS

6

READS

1,492

2 authors:



Rafflesia Khan
Khulna University

11 PUBLICATIONS 10 CITATIONS

[SEE PROFILE](#)



Rameswar Debnath
Khulna University

38 PUBLICATIONS 320 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Efficient Object Detection and Machine Learning Based Recognition from Image [View project](#)



Human Distraction Detection from Real Time Video Data using Artificial Intelligence and Deep Learning Techniques [View project](#)

Multi Class Fruit Classification Using Efficient Object Detection and Recognition Techniques

Rafflesia Khan*

Computer Science and Engineering discipline, Khulna University, Khulna, Bangladesh
Email: rafflesiakhan.nw@gmail.com

Rameswar Debnath

Computer Science and Engineering discipline, Khulna University, Khulna, Bangladesh
Email: rdebnath@cseku.ac.bd

Received: 21 May 2019; Accepted: 25 June 2019; Published: 08 August 2019

Abstract—In this paper, an efficient approach has been proposed to localize every clearly visible object or region of object from an image, using less memory and computing power. For object detection we have processed every input image to overcome several complexities, which are the main limitations to achieve better result, such as overlap between multiple objects, noise in the image background, poor resolution etc. We have also implemented an improved Convolutional Neural Network based classification or recognition algorithm which has proved to provide better performance than baseline works. Combining these two detection and recognition approaches, we have developed a competent multi-class Fruit Detection and Recognition (FDR) model that is very proficient regardless of different limitations such as high and poor image quality, complex background or lightening condition, different fruits of same shape and color, multiple overlapped fruits, existence of non-fruit object in the image and the variety in size, shape, angle and feature of fruit. This proposed FDR model is also capable of detecting every single fruit separately from a set of overlapping fruits. Another major contribution of our FDR model is that it is not a dataset oriented model which works better on only a particular dataset as it has been proved to provide better performance while applying on both real world images (e.g., our own dataset) and several states of art datasets. Nevertheless, taking a number of challenges into consideration, our proposed model is capable of detecting and recognizing fruits from image with a better accuracy and average precision rate of about 0.9875.

Index Terms—Image Processing, Edge Sharpening, Object Region Segmentation, Fruit Localization, Fruit Recognition, Convolutional Neural Networks.

I. INTRODUCTION

Computer vision is an interdisciplinary field which has been gaining a huge amount of attraction in recent years. The goal of computer vision is to teach a computer to extract information from image as close as possible to

humans who can naturally understand everything by seeing the same image. In order to achieve this goal, one of the integral parts of computer vision is object detection and recognition that deals with localizing a particular object region or contour from an image or video and then classifying it. Object detection and recognition have been applied to different problem domains over so many years, whether the objects were handwritten characters [1], house numbers [2], traffic signs [3, 4], objects from the VOC [5] dataset, objects from the 1000-category ImageNet dataset [6] or Caltech-101 dataset [7]. In recent days, object detection is being used for so many applications. There are some state-of-the-arts which work for different types of object detection such as flower detection [8], fruit detection [9, 10], food segmentation and detection [11] cats and dogs detection [12] etc. The main goal of all these detection algorithms is to obtain higher efficiency and cover different complex use cases by overcoming different limitations. Considering so many applications and subfields of object detection and recognition, this paper proposes a Fruit Detection and Recognition (FDR) model where efficient techniques for fruit detection improves the performance of deep learning based classification of fruit from scene image.

Fruits provide an essential role as a food in our everyday life. It provides nutrients vital for our health and maintenance of our body. Those who eat more fruits as a part of a healthy diet are likely to have reduced risk of some chronic diseases. However, not all fruits are treated equally and it is a matter of concern that not every person knows about every fruit well. With the help of Artificial Intelligence (AI) and Machine Learning (ML) we can develop an automatic fruit recognition system with an information dataset of each fruit. This system can help us to select fruit that is suitable for us and teach us about the characteristics of that particular fruit. These types of systems can help us to educate children and familiarize them with fruits. Furthermore, these systems can be used to teach a robot to find the correct fruit for its user and this becomes much important for those robots which are being used for fruit harvesting related works. Another major application of fruit detection and recognition is at

smart refrigerator. Now a days smart refrigerator can detect how fresh a fruit is, how many of which kind of fruits are left, which fruits are less in amount and need to be added in the shopping list. As people have more access to health information, it is often found that recommendation of healthy food is very essential. While shopping, an automatic fruit recognition system connected to information database can help the consumer to select healthier fruit along with nutrition details. Also, in recent time super shops use these kinds of systems to provide information about each type of fruit to customer, to keep track of the sold and in stock product and also to identify the most demanding fruit item. Even on-line shopping sites can use such automated system very easily. For all these functions a proper fruit detection and recognition system is a must.

Considering all these importance of fruit detection this topic is catching a huge research attention. A number of recent works [9, 10] are especially focused on fruit detection and recognition. But, still the existing works are facing some challenges. One of the main challenge of developing a perfect fruit detection and recognition system is that it needs extra effort in the detection part where in an image the difference between two different fruit is very limited. Fig. 1 shows an image of apple and tomato which are very similar in some color and shape. Differentiating red apple from red tomato needs an efficient detection algorithm that can identify objects of similar color and shape by exploiting the difference in their texture.



Fig.1. Fruits of similar color and shape.

Therefore, it requires huge computation and more complex cluster definition with more detailed features. But, most of the existing works only consider single type fruit detection e.g., apple detection [13], orange detection [14] and even models like [15] and [16] works for only red apple detection where the intensity difference between leaf and fruit is very high. Detecting only the fruit in an image is not enough for an efficient fruit detection algorithm, it also needs to distinguish one class of fruit such as apple in Fig. 1 from other classes, i.e. tomato. As a result, multi-class fruit detection makes fruit detection and recognition problem even more complex. At the same time, object classification from image is itself a very complex and time-consuming procedure.

Most of the well-known deep learning based object detection and recognition models such as FAST-RCNN [17], YOLO [18] and RCNN [19] have used a big data set and computation power over GPU in order to get an

accurate result within satisfactory time. To extent to the existing limitation of time and performance, object detection among same category objects with multiple classes is even more complex problem. For example if we need to detect a red apple, a green apple and an orange apple differently from the image shown in Fig. 2, we need a fine-grained algorithm that can efficiently detect the difference between the region of object (ROO) containing different apples regardless their similar shape.



Fig.2. Red, green and orange apple. (Same shape with different colour)

However, that is not the end of the complexities in case of fruit detection problem domain. Real life images as shown in Fig. 3(a), are taken with a very noisy background which can totally distract the machine learning model and can point out a completely wrong target object and the entire algorithm can be failed. But, models like [20, 21] and [22] do not consider fruit images with reach background. Most of the existing models work better on an image with single fruit in white background. In addition, objects in the test image can have different dimensions and can be taken from different angels as shown in Fig. 3(b). Another major challenge of fruit detection is that fruits are likely to be found in groups and inside of an image they might appear overlapping one another within a same region as shown in Fig. 3(b). But, most of the existing works like [9] and [23] avoid this challenge in case of fruit detection.



(a) Fruit with noisy background



(b) Fruits are in different angel in the image

Fig.3. Images of fruit with noisy background and different angel of fruits.

Also there may exist water drops and spots on the fruit surface that may create more misleading objects. The water drops in Fig. 4 is very small details, but still it can compromise the performance of the learning for deep learning algorithm. Therefore, these kinds of small yet influential details need to be removed before learning phase of the algorithm.



Fig.4. Object with misleading details

Therefore, a perfect machine learning algorithm for fruit detection and recognition needs to be capable of following characteristics,

1. Differentiating the target object, here in case fruit, from the background
2. Classifying the target fruit with any dimension or shape
3. Separating all connected and overlapping fruits as single one
4. Detecting proper features that can classify every fruit from a dataset
5. Recognizing each and every fruit from image with less computation and high accuracy

In addition, as each image is captured using different devices of different kinds and qualities, detecting and recognizing an object from any image is still a complex problem without any universal solution. Hence, this proposed work aims for solving some of the complexities for object detection and recognition from image. In a nutshell, the main contributions of this proposed system are as follows.

- An efficient novel fruit detection and recognition system has been developed with high accuracy and low computation and power consumption.
- For better performance, this proposed model works in two different phases, detection (e.g., finding the region of object) and recognition (e.g., classifying the fruit). Each phase uses different algorithm. The output of detection phase is used as the input for recognition phase.
- For object detection phase, a new and efficient detection algorithm has been introduced which is capable of detecting every region of object (ROO) from an image with less computation without using deep learning (i.e. CNN) or sliding window based algorithms.
- Then for recognition phase, a CNN based recognition model has been implemented with a moderate combination of CNN, pooling and dense layer. Therefore, the proposed recognition model is more meticulous and trustworthy in case of differentiating non fruit regions as *other objects* and classifying different class of fruits accurately.

- For model evaluation, we have evaluated our model's performance using various renowned dataset's images where each dataset usually have same type of images. We also have prepared our own dataset images where we have collected different types of images with different quality and aspects in order to validate the proposed model against different existing challenges mentioned earlier. As no similar universal dataset is available for multi-class fruit detection problem, our dataset can be used for different challenge solving in case of multi-class fruit detection.
- In both cases, either with existing dataset or with our new dataset, this proposed FDR model shows better performance which has proved that this proposed model is not a dataset oriented one.
- Finally we have compared our model's performance on different aspects with existing models and demonstrated our model's proficiency.

Fig. 5 shows a simple flow diagram of the working phases of proposed fruit detection and recognition (FDR) approach. Fig. 5 shows that in our model at first an input image is feed to the detection phase which outputs all the detected region of objects (ROO) of that image as separate images (k) each containing a single object and then every detected image (k) is feed to the recognition phase which distinguishes fruit ROO from non-fruit ROO and finally classify the fruit ROO to corresponding class of fruit. And non-fruit ROO as other object.

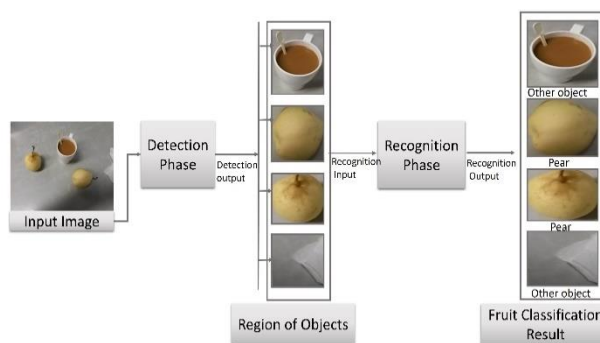


Fig.5. Framework for proposed fruit detection approach.

We have organized this paper in an order that some of the states of arts have been presented with their detail contribution in section II. The proposed frameworks along with significant contributions are discussed in section III. Section IV demonstrates significant improvement in the performance and efficiency of our model along with experimental set-up and evaluation of performance. A comparative analysis of proposed model with some of the existing state-of-the arts on different sectors has been presented in section IV. The final section V of this paper consists conclusion with some discussion and future plan of this work.

II. LITERATURE REVIEW

Fruit detection and recognition (FDR) is a topic of large practical real-life importance. Many recent works have addressed several challenges of FDR problem domain including feature detection, fruit segmentation, fruit recognition, template matching, fruit finding and some other. There is a large literature on object proposal methods, classification methods as well as methods that are capable of both object detection and recognition (ODR). There exists a number of works for FDR as well. Some of these research works use feature based techniques for recognizing or locating fruit from an image while some other require deep learning based algorithms. In this section, we mainly describe some of the well-known models that can be used for FDR along with some already developed fruit detection and recognition models.

A. Fruit Detection and Classification using Selected Feature Based Approach

Different applications such as fruit harvesting in garden, detecting and counting fruits in an orchard mainly uses feature based fruit classification techniques in order to locate or detect multiple classes of fruit [24-26] or single class of fruits [13, 14]. The approach described in [24] demonstrates a feature (main focus size and color) learning based approach for fruit segmentation from image. Both models [25] and [26] work with feature based algorithm for classifying multiple classes of fruit on the tree for fruit harvesting system. Model [25] presents a key point detector - Angular Invariant Maximal (AIM) which utilizes the distinct intensity and gradient orientation pattern formed on the surface of the fruit for detection. However, one of the main requirements of this model is that fruit needs to have a smooth and round shape. The algorithm described in [26] extracts the features like intensity, color, edge and orientation. These extracted features are integrated using weights according to their influence on the image and the integrated weigh map is segmented using global thresholding for generating the binary map from which the fruit regions are finally extracted. The single type fruit detection [13] only detects apples and relies on a combination of the apple's color, texture and 3D shape properties. On the other hand model [14] compares the results of both color and edge based segmentation for orange detection. Both [13] and [14] are only focus on the features of a single type of fruit. Although these feature based FDR systems are rich in accuracy at their specific considerations, developing an FDR system for every single type of fruit can never be the feasible solution. An efficient FDR system needs to consider all kinds of fruits and to be dynamically adoptable in order to classify any fruit.

Models like [24-26] can locate multi-category fruits but cannot classify them. In addition, there are a number of works which focus on a single feature of fruit and not consider more than one feature. For example, some

feature base fruit recognition models, like [20] focuses on color chromaticity, model [21] focuses on wavelet, model [22] focuses on shape feature and only consider recognizing fruit without detecting or locating it from the global image. In this paper, by global image we refer an image with multiple objects and rich background where a local image is an image with single object and simple background. Moreover, single feature focused methods cannot differentiate all category of fruits even in some cases shape, color and texture together are not enough for differentiating two fruits i.e. apple and peach. Fruits of different types can be differentiated using several features, e.g., shape, color, texture and size. Hence, it is almost impossible to capture all characteristics of fruits using a single feature. Therefore, FDR models needs to be dynamic and consider multiple features of objects.

B. Object Detection and Classification using Deep Neural Network

Deep neural network has gained huge attention in recent days in different problem domains. Similar to other problem domains, researchers are trying to implement efficient deep learning based model for object detection and classification. Single feature based algorithm is not a suitable solution for multi-class FDR problem. However, using multiple features for detection or classification of fruits is both time and computation consuming. So recently some of the deep neural network based ODR systems have gained huge popularity. Where using different deep neural network models such as convolutional neural network (CNN) as generic feature extractor makes it easy to select appropriate features for detecting and classifying objects.

Most of the recent popular CNN based ODR models such as FAST-RCNN [17], YOLO [18], RCNN [19], SSD [27] use the bounding box prediction technique to localize an object and then classifies it. All these works follow a generic work flow as following:

First, A CNN model is trained with regression (bounding box) and classification objective (loss function). Then features are extracted as a feature maps from the convolution layer. Then using feature map, a box is located on the image region. After that the proposed neural network model is learned to identify any object in that box. Then for filtering multiple boxes algorithms like non-maxima suppression is used. The Intersect Over Union (ioU) between the prediction and the ground truth is compared against each other. Finally, if ioU is bigger than some threshold that ioU is reshaped to a fixed size and send to a fully connected layer to continue the classification.

Each of these model is being used for fruit detection and recognition but each of these predicts more than thousands of region proposals and runs classifier on them. For this reason, these models work really efficient in performance but expensive in the terms of computation and time. In addition, these models use really big dataset for learning features. All these models have to train three different modules separately which are: (1) a CNN to

generate appropriate image features, (2) a classifier for predicting the class of object, and (3) a regression model to tighten the bounding boxes to predict object location. This characteristic makes the pipeline extremely hard to train and use.

C. Fruit Detection and Classification using Well Known CNN Based Object Detection and Recognition models

As previously mentioned, recently some well-known object detection and recognition methods that are based on deep learning [17–19, 27] are widely used for every kind of objects including fruits. Some of them are described as follows.

In the paper [10], Kuang, Hulin, et al. proposes a multi-class fruit detection (MCRD) based on image region selection and improved object proposals using effective image region selection and improved object proposals. They have created their own large dataset [28] with five kinds of fruits in it for detection and classification. They have combined five features (e.g., LBP, HOG, GaborLBP, Color Histogram and Global Shape) to improve the detection accuracy of fruit. To improve the detection speed and ensure region selection they have improved the traditional 'EdgeBox' [29] method which is usually used for object bounding box selection using edge feature comparison. For performance evaluation they have compared their system with several baseline methods: Sliding window, EdgeBoxes [29], DPM [30], CNN + SVM, Cascade and Faster RCNN [17] and successfully outperform them with average detection rate about 0.9623 and miss rate about 0.0377. Although their model has a better accuracy but their model uses images with similar characteristics for training and test. By similar characteristics we mean that all training and test images are captured using same camera within same color and lightening condition. However, in real world images can hardly be so close to each other in terms of characteristics.

H. Muresan and M. Oltean in model [9] introduce a dataset named Fruits-360 [31] that currently contains 49561 images of 74 kinds of fruit. They have created a software that can recognize fruit from images using deep learning. Their proposed model used CNN for recognition. Their main objective was to present the results of some numerical experiment for training a neural network to detect fruits. Their system does not need any detection operation as Fruits-360 contains only single object fruits for both training and testing. And this drawback makes this system a little inappropriate for real time use. Their proposed model works better with Fruits-360 only.

In the paper [32], a fruit recognition algorithm based on convolution neural network (CNN) is proposed by Hou and Lei et al. They have improved the recognition rate by proving that recognition rate of CNN combined with selective search algorithm is higher than using traditional CNN only. For evaluating their model they have used a small dataset with images of fruits in a white background and their model never checks the overlapping condition of fruits.

The main objective of Jidong and Lv et al. [33] is recognizing occluded apple under natural environment. For apples in overlapping condition they have carried out the processing through a method of adding a segmented binary image to an edged image and the overlapping and adhering parts between apple fruits were separated. But this kind of model cannot be generalized for multi class fruits.

Different research works have been proposed to solve multi-class fruit detection and recognition problem. However, these models still need improvement in terms of overcoming some limitations such as huge computation power consumption, too long learning phase, requirement of large dataset for training model, poor quality image, different focus length and lighting condition, overlapped fruits, noisy background and detection fruit from global image rather than local image etc. In this paper, an FDR model has been proposed to overcome these challenges without compromising model's efficiency and accuracy.

III. PROPOSED FRUIT DETECTION AND RECOGNITION APPROACH

Our proposed FDR model works mainly in two phases: (1) Fruit detection (detecting or localizing all fruit's region from input image) and (2) Fruit recognition (classifying the detected fruits). The detection process is all about identifying every single connected object from an image regardless of its location, quality of appearance, background of appearance and overlapping condition. Our proposed system can detect all connected objects from image. Here, a connected object refers to an object which contour (e.g., outline representing or bounding the shape of object) is a connected or closed entity. The recognition phase works for classifying every detected objects using CNN. Recognition phase mainly focuses on classifying each detected fruits from detection phase by their corresponding fruit class. In recognition phase non-fruit objects are marked as other object class. Therefore, they can be distinguished from fruits in an image. Fig. 5 shows a simple flow diagram that describes the overall working phases of our proposed model. The whole code of this proposed method is written in python language and soon we will make the code available in github.

A. Fruit Detection or Localization Process

The very first requirement of fruit detection phase of this proposed FDR model is to process the input image. Several pre-processing operations are performed on the input image (IMG) which is combinations of some efficient detection techniques that makes outline detection or connected contour detection really smooth and easy. Then all the connected contours from pre-processed output image are detected as several Region of Interest (ROI_i) where an object belongs. Each image becomes collection of ROIs. Therefore, each image can be described as equation (1). Each detected convex connected contour is then cropped out as a local single

object image from the global input image. These single object images can be represented as a collection of regions of objects (ROO) or regions of interests (ROI) or simply objects as in equation (1). Here ROI is an area that contains a connected contour or object.

$$IMG = ROI_1, ROI_2, \dots, ROI_n \quad (1)$$

The working flow diagram for proposed fruit detection approach is shown in Fig. 6. The main steps of object detection process can be described as following:

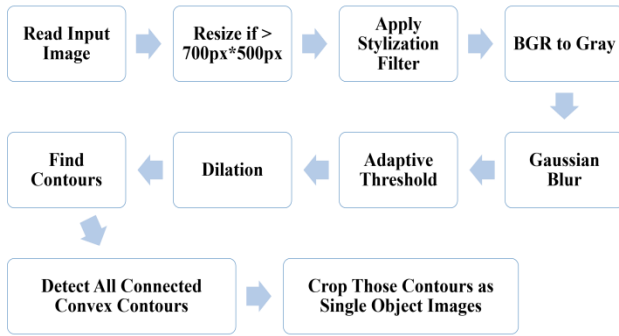


Fig.6. Work flow for proposed fruit detection approach.

Step 1: Read input image in BGR

At first the input image is read as a color image and the decoded image usually has the channels stored in BGR (Blue, Green, Red) order. The main version of the input image is stored as IMG and a copy of that image (CIMG) is used for further processing.

Step 2: Convert to unique size

Input image can be of several sizes. Processing really big sized image makes the process slow. It also requires unnecessary power and computation time. So, after running a set of test cases we have selected an optimal size for input image, which is 700×500 pixels. Each input image that exceeds this optimal size (700×500) is resized to the optimal size. Otherwise resizing is ignored.

Step 3: Stylize the image to sharp overlapping area

After resizing, we create a stylized version of the input image by applying stylization filter on it. Stylization filter sharpens each object's edges. In one of our previous work [34] we have introduced an Efficient Fine-grained Algorithm for Multi Category ODR where we have used 10 filters to detect every object's shape separately. But, here in this proposed model only stylization filter is enough as it is used along with improved detection approach. The stylization filter is one of the mostly used edge preserving filter [35]. Where there are so many edge preserving filters like Bilateral, Detail enhance, Pencil drawing etc. Among all these filters, stylization is faster than others [36] and also it uses Normalized Convolution (NC) filter which provides better accuracy. So, it performs accurately in case of smoothing similar image regions by also preserving and sharpening relevant edges.

Eventually, it produces an output that looks like the image is painted using water color with sharp edges. Even using single filter instead of multiple (i.e., 10) filters, the accuracy is not compromised. A number of different test images have been used and in each case stylization filter outperforms other filters in term of time, computation and accuracy. So, we have chosen stylization filter for better result. Stylization filter is applied on input image to make every object edge or outline or contour smooth and at the same time sharp. Applying this filter makes the image edges as well as object outline much visible and sharp. So, it helps the further pre-processing steps to detect and separate every object's outline or contour. Fig. 7 shows the comparison of Detail enhance, Pencil drawing edge preserving and Stylization filter where these filters are applied on the same image and a small portion of these three filters applied output image is high-lighten.

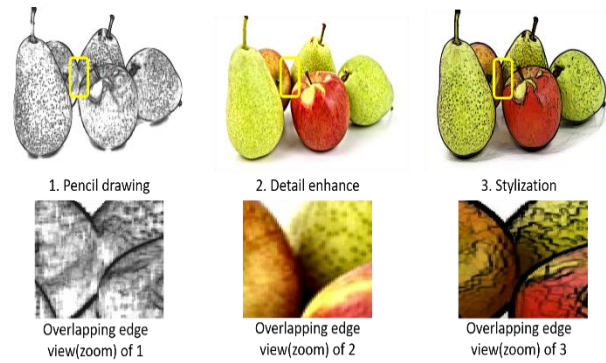


Fig.7. Comparison of Detail enhance, Pencil drawing edge preserving filters with Stylization.

In Fig. 7 it is shown that stylization filter perfectly smoothens and preserves every contour of every object of an image and draws a thick outline to every object edge.

The working process of stylization is as follows, At first image details are reduced by Gaussian Filtering.

$$I(x, y) = \frac{1}{2\sigma\pi} \times e^{-\frac{(x^2+y^2)}{2\sigma^2}} \quad (2)$$

Here (2) is used for performing Gaussian Filtering where $I(x, y)$ is image and x and y are representing the pixel coordinates and σ and π are constants.

After that region Smoothing is performed by Bilateral Filter using the following equations,

$$I_p^{bf} = \frac{1}{W_p} \sum_{x \in N(p)} G_{\sigma sd}(|p - x|) G_{\sigma rd}(|I_p - I_x|) I_x \quad (3)$$

$$W_p = \sum_{x \in N(p)} G_{\sigma sd}(|p - x|) G_{\sigma rd}(|I_p - I_x|) \quad (4)$$

Here, (3) $I(p)$ is image (I) pixel. bf represents Bilateral filter. G is Gaussian, W_p , calculated using (4), is the sum of weights, sd is spatial domain, rd is range domain.

Then coherence-enhancing diffusion (CED) performs nonlinear anisotropic diffusion filtering which can be formulated by (5).

$$\delta I / \delta t = \text{div} (D \nabla I) \quad (5)$$

Finally, the spatial hierarchical characteristics of different objects are high lighten by sharpening the edges in image using (6). I_{uu} in (6) is the second derivative of Image(I) along with direction u.

$$\delta I / \delta t = -\text{sign} (I_{uu}) |\nabla I| \quad (6)$$

These sharp edges make it easier to detect every object individually where Detail enhance and Pencil drawing fail to do so.

Step 4: Highlight the colour variation

After applying Stylization filter, the image channel is converted from BGR to Gray using (7).

$$G_{g(x,y)} = 0.114I_{B(x,y)} + 0.587I_{G(x,y)} + 0.299I_{R(x,y)} \quad (7)$$

When a BGR image is converted to gray scale only one-third of the data compared to the original color BGR image is needed to be processed which significantly reduces the memory and energy consumption as well as the amount of computation. This data reduction allows the algorithm to run in a reasonable amount of time. For many image processing applications especially video processing (e.g., real-time object tracking), huge computation time is one of the major limitations. Hence, to make our model light in weight and run the process faster the input image is converted from BGR to Gray. Also, in case of gray image pixel colour variation is much more visible and it makes the contour (e.g., outline of object) detection easier.

Step 5: Noise reduction

The main concern of the proposed model is to make every object boundary sharper and smoother so that the system can detect every connected contour from the image separately. Following this concern, at this stage any further noise from the gray image is reduced. After reducing the substantial amount of noise using linear Gaussian filtering (e.g., a non-uniform low pass filter), the input image becomes much smoother. Here Gaussian filter is used because it works faster and better than other existing noise reduction methods. The formula of a Gaussian function in two dimensions is shown in (8).

$$G_a(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (8)$$

To preserve image brightness and make the computation faster we have convolved the image with an integer valued 15 x 15 kernel of Gaussian values with a σ valued 0 in both x and y direction.

Step 6: Separate background and foreground

After removing noise from the image, we apply thresholding. The main objective of thresholding is to distinguish pixels that belong to true foreground regions with a single intensity from the pixels on background

regions with different intensities. Also, using a global threshold value may not be a good choice as an image may have different lighting conditions in different areas. So, here we have used Adaptive Thresholding [37]. It uses an algorithm that calculates the threshold for a small region of the image so that we can get different thresholds for different regions of the same image and it gives us better results for images with varying light conditions [38]. As lighting on image always manipulates every object detection model to misinterpret a brighter portion of an image area as an object location, thresholding helps our system to separate background and foreground and make the foreground objects much visible. Also this process avoids the lighting problem. This step finally outputs a binary image and also helps to reduce remaining noise after step 5.

Step 7: Dilation

We perform a Dilation [39] on the previous thresholded image. This procedure follows convolution with some kernel of a specific shape such as a square or a circle. This kernel has an anchor point, which denotes its center. The kernel is overlapped over the picture to compute maximum pixel value. Then the picture is replaced with an anchor at the center. With this procedure the areas of bright regions grow in size or size of foreground object increases. It also helps to join the broken parts of an object [40] which contributes in connecting a broken contour. In our work this dilation helps to separate two or more overlapped or closely connected objects which belong to same region overlapping one another. As the dilation of an image $I(x,y)$ by a structuring element or kernel produces a new binary image (denoted as D) with ones in all locations (x,y) if at least one pixel under the kernel is 1 i.e., for a binary image a pixel is set to 1 if any of the neighboring pixels have the value 1. In our work we have performed morphological dilation using an integer valued 3 x 3 kernel. The formula for this process is shown in (9).

$$D(x, y) = I(x, y) \oplus \text{kernel} (3 \times 3) \quad (9)$$

Fig. 8 shows the output results of an input image after adaptive thresholding in Fig. 8(a) along with after morphological dilation in Fig. 8(b). It is shown in Fig. 8(c) how dilation checks overlapping and broken edges. As dilation makes edges thicker eventually there forms one large connected contour containing all objects in same region and number of individual single object contour. And after this step our pre-processing operations ends.

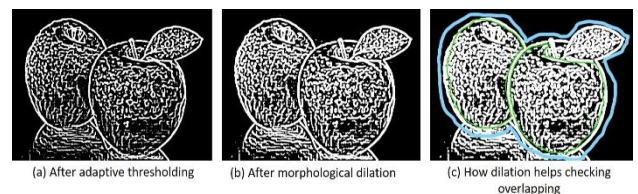


Fig.8. Dilation checking overlapping and creating separate edges.

Step 8: Contour detection

After all these pre-processing operations (from step: 1 to step 7), the input image is ready for detecting the contour of every object separately from it. So, at this stage all the visible contours are detected from the dilated image using `detect_contour()` method.

Step 9: Identify ROI only

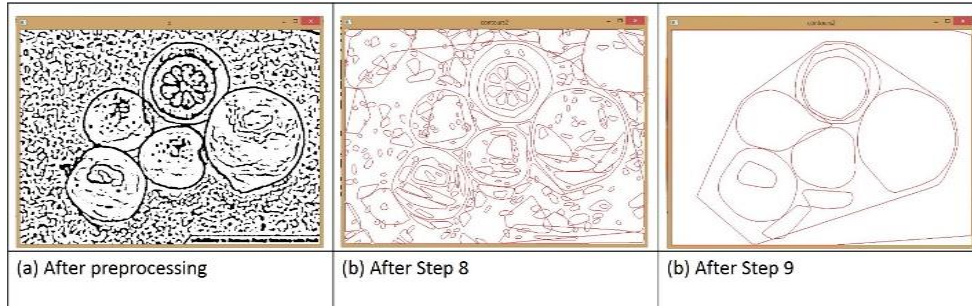


Fig.9. Condition of a sample input image after step 7, 8 and 9.

So, at this stage, from all contours we select top most large and connected contours and ignore all the small surrounding misleading contours.

Fig. 9 shows the output of step 7, 8 and 9 in a sample input image.

Step 10: Generate Individual image for each ROI

At the final step, the proposed algorithm crops all selected connected contour's region from IMG in a rectangular shape. These regions are the detected regions of object (ROI_i) of the input image. The detection phase of the proposed FDR model ends here. Hence, after the detection phase we get a set of single object or region of object containing images as output and this output set is used as the input of recognition phase. An example is shown in Fig. 10 which shows all the step's result for fruit detection process. Where the existing models [17–19, 40] predicts more than thousands of region proposals and runs classifier on them our proposed model detects exactly those regions where a connected object really exists. All the above described steps make our model capable of fruit localization as well as detection without checking thousands of region proposals and using highly complex algorithms.

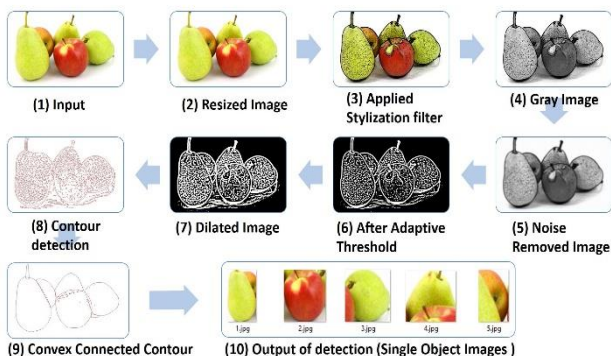


Fig.10. An example of fruit detection approach with every step's output

From the previous steps of operation we have detected all connected components as object contours. The set of contours can be comparatively large in number because some image might contain a lot of background as well as foreground objects in it.

B. Fruit Recognition or Classification Process

From the detection phase, all the foreground object regions have been successfully detected from image background and a set of single object images have been generated as shown in Fig. 10. These single object images are the test image for recognition model. Finally, each of those single object images is classified as either corresponding fruit class, or non-fruit objects as other class. Our main goal is to provide generalize solution regardless of dataset and camera angel. After detection phase every image is noise free and contains only one ROI. This single ROI is either a fruit or not. We are generating the average threshold image for each kind of fruit before training in neural network. Fig. 11 shows the average train image for each class apple, orange and persimmon.

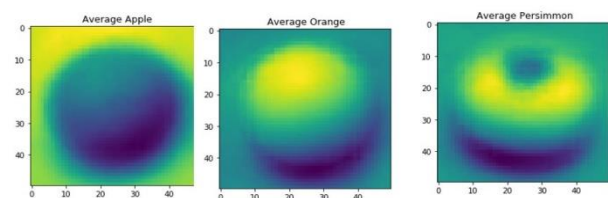


Fig.11. Average training sample for each fruit.

Each of the single image is of similar size, therefore, there is very small feature to be extracted. A simple neural network cannot deal with this amount of feature. Therefore, for recognition, we have used the Convolutional Neural Network (CNN), which is a feed-forward artificial neural network also called multi-layer perceptrons (MLPs) in Keras. One of the beneficial feature of CNN is that it can overcome model over fitting using dropout technique. In this process, an image is feed as input into the network, which goes through multiple convolutions, subsampling and fully connected layer and finally outputs the class name of the input fruit image

(e.g., class with the highest prediction value). Where convolution layer computes the output of neurons that are connected to local regions or receptive fields in the input, each computation leads to the extraction of a feature map from the input image. Fig. 12 shows a general structure of the proposed CNN based recognition model that includes CNN, subsampling, fully-connected and dropout layers.

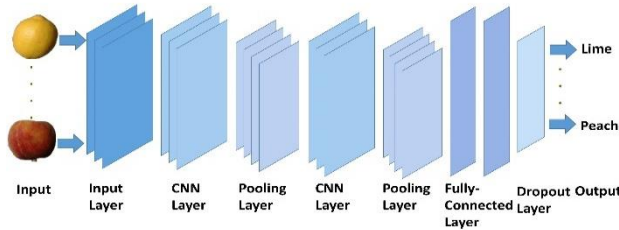


Fig.12. General CNN recognition model.

Table 1. Proposed CNN model (A) configuration

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 32, 80, 80)	896
conv2d_2 (Conv2D)	(None, 32, 80, 80)	9248
max_pooling2d_1(MaxPooling2D)	(None, 32, 40, 40)	0
conv2d_3 (Conv2D)	(None, 64, 40, 40)	18496
conv2d_4 (Conv2D)	(None, 64, 40, 40)	36928
max_pooling2d_2(MaxPooling2D)	(None, 64, 20, 20)	0
conv2d_5 (Conv2D)	(None,128,20,20)	73856
conv2d_6 (Conv2D)	(None,128,20,20)	147584
max_pooling2d_3(MaxPooling2D)	(None,256,10,10)	0
conv2d_7 (Conv2D)	(None,256,10,10)	295168
conv2d_8 (Conv2D)	(None,256,10,10)	590080
max_pooling2d_4(MaxPooling2D)	(None, 256, 5, 5)	0
flatten_1 (Flatten)	(None, 6400)	0
dense_1 (Dense)	(None, 256)	1638656
dropout_1 (Dropout)	(None, 256)	0
dense_2 (Dense)	(None, 256)	65792
dropout_2 (Dropout)	(None, 256)	0
dense_3 (Dense)	(None, 5)	1285
activation_1(Activation)	(None, 5)	0
Total params: 2,877,989		
Trainable params: 2,877,989		
Non-trainable params: 0		

According to [41], a consequence of replacing all layers with convolutional ones creates an increase in the number of parameters for the network. In addition, in case of deep learning various tweaks and changes to any layers as well as the introduction of new layers can provide completely different results. To provide some improvement over the networks that have fully connected layers in their proposed structure, another option is to replace all layers with convolutional layers. Hence, in our model, we have structured our CNN based recognition model in such a way that it combines a number of convolution, subsampling and fully connected layers in a moderate way which makes a more correct and trust worthy recognition model. We have found existing dataset like fruit360 or Imagenet very simple dataset. As

here most of the images are taken from same view point using same contrast and lightening. However, real life fruit images are often differ at shape and view point. Real life fruit images are complex and have noisy background. Therefore, we have designed two different neural network model based on the nature of dataset. The neural network model used for fruit recognition with complex training images is described in Table 1.

For each input image, a Convolutional layer performs set of mathematical operations and produces a single value output feature map. These layers then typically apply an activation function (sigmoid) on the output to introduce nonlinearities into the model [42]. Then Subsampling is used to get an input representation by reducing its dimensions, which helps to reduce overfitting. Max Pooling is used for subsampling layers which are also used to downsample image data extracted from convolutional layers to reduce the dimensionality of the feature map in order to decrease processing time. Finally, the fully connected layer is used to flatten the high-level features that are learned by convolutional layers and combining all the features. In a dense or fully connected

Table 2. Proposed CNN model (B) configuration for simple image such as F360 [31]

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 32, 45, 45)	896
conv2d_2 (Conv2D)	(None, 32, 45, 45)	9248
max_pooling2d_1(MaxPooling2D)	(None, 16, 22, 45)	0
conv2d_3 (Conv2D)	(None, 64, 22, 45)	9248
conv2d_4 (Conv2D)	(None, 64, 40, 40)	36928
max_pooling2d_2(MaxPooling2D)	(None, 32, 11, 45)	0
conv2d_5 (Conv2D)	(None,128,11,45)	36992
conv2d_6 (Conv2D)	(None,128,11,45)	147584
max_pooling2d_3(MaxPooling2D)	(None,64,5,45)	0
conv2d_7 (Conv2D)	(None,256,5,45)	147712
conv2d_8 (Conv2D)	(None,256,5,45)	590080
max_pooling2d_4(MaxPooling2D)	(None, 128, 2, 45)	0
flatten_1 (Flatten)	(None, 11520)	0
dense_1 (Dense)	(None, 256)	2939676
dropout_1 (Dropout)	(None, 256)	0
dense_2 (Dense)	(None, 256)	65792
dropout_2 (Dropout)	(None, 256)	0
dense_3 (Dense)	(None, 75)	19275
activation_1(Activation)	(None, 75)	0
Total params: 4,013,163		
Trainable params: 4,013,163		
Non-trainable params: 0		

layer, every node in the layer is connected to every node in the preceding layer which perform classification on the features extracted by the convolutional layers and downsampled by the pooling layers [42]. When a dense layer is followed by a dropout layer it reduces the complexity of the model with the goal to prevent overfitting. Eventually, fully connected layers passes the flattened output to the output layer where we have used an activation (e.g., softmax) to predict the input class label. Finally, the class label with the highest prediction

value is considered as the actual class of input image object.

Table 2. shows the design configuration for neural network model used for simple dataset. Even the class number increases from 5 to 75. In compare to model A, the number of parameters increases only up to double in model (B).

IV. PERFORMANCE AND COMPARATIVE ANALYSIS

Earlier, in this paper, we have explained that our FDR model works in two phases that are fruit detection and recognition. In this section, we are going to discuss about the performance and results we have found in both cases of detection and recognition phase along with some comparative analysis among different existing FDR models.

Before we go any further, here we provide a short description about our experimental setup and condition as we belief this might be helpful for evaluating our model's performance as well as for other research.

Experimental Setup and Condition

The experimental setup used in this model evaluation study are as follows: The processor used is Intel(R) Core(TM) i5-2550M, the computing speed of the CPU is 2.50 GHz, the installed memory (RAM) is 6 GB, and the operating system is Windows 8.1 64-bit. This configuration is really poor compared to the setup configuration of another existing research project like [10]. But, using this configuration we have achieved a comparatively better performance.

The first objective of this proposed model is to introduce a new efficient technique for object detection rather than using the traditional deep learning-based method which usually use sliding window or bounding box for object detection. This phase (detection phase) is designed in such a way that it is capable of detecting the accurate as well as the smallest number of regions. This detection process detects almost accurate number of object's region present in an image and thus minimizes the computation for second phase (recognition phase). In detection phase this model detects and creates a set of all the object's region that is visible or simply clear in an image. The second objective is to develop an improved CNN based model for recognizing those detected objects of the first phase. This phase is designed in such a way that it is capable of classifying every fruit from the set of the detected image region and classifying all other non-fruit region as other class. Here non-fruit regions refer to those regions where an object does belong but it's not a fruit.

For recognition we have trained the model using images of different class sample shown in Fig. 13. We have collected test image from real life and also from the Fruit-360 dataset as shown in Fig. 14(a). The test images for CNN model are generated from the input images as shown in Fig. 14(a) through detection phase. Some samples are shown in Fig. 14(b).



Fig.13. Training sample for each class of fruit from [28].



(a) Input sample for fruit detection and recognition.



(b) Output of fruit detection phase and test sample for fruit recognition.

Fig.14. Test Image (e.g., detection phase result) for fruit recognition

Up to now for fruit detection and recognition there is no renowned dataset but there exists some datasets provided by several researchers. For evaluating the performance of our proposed model we have executed our model on some of these benchmarked datasets and also on a huge number of real world images collected randomly from the internet or captured by ourselves. We will make our dataset available through git for further research work.

The following subsections are organized with each phase's performance followed by some comparisons with existing state-of-the arts.

A. Fruit Detection Phase

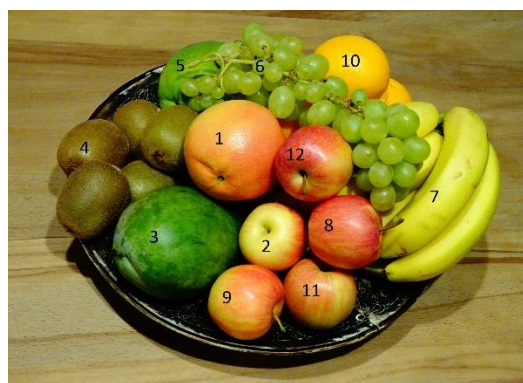
The performance evaluation and comparative analysis of our FDR model in case of object region detection is described as follows.

1) Performance evaluation in region detection

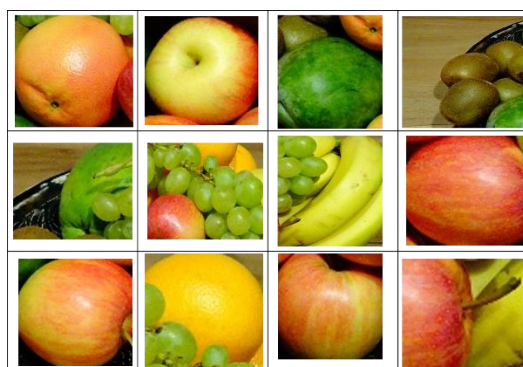
Our model is capable of detecting fruits from any kind of images, more specifically the region of an image where any fruit exists. Whether input image is blurry (blurred background) or non-blurry (rich non-blur background) or poor in quality (less resolution) or much bright or dark or any other particular type, the detection process described in Section III will ensure fruit region detection from that particular image. Some of the detection results are shown below. Here we are describing some different kind of images as a particular case and detecting fruit location from every kind of image. Where every case input image shows an input image and the case output image shows a set of output images that we get as detection result.

Case 1 - Input: A poor quality image

Most of the existing models do fruit detection but they always use preprocessed high-resolution images from where detecting any object's intensity difference is really easy. Our model works really well for both rich and poor quality images. By poor quality image we mean images that are of poor resolution and captured with less powerful camera. Fig. 15(a) shows a poor quality image and Fig. 15(b) shows the detection results found by our



(a) Input image.



(b) Output single fruit images detected.

Fig.15. Case 1 - Fruit detection from poor quality image.

detection phase from that input. Fig. 15(b) clearly shows that our model has perfectly separated and detected 12 fruit's region from the poor quality input image in Fig. 15(a).

Case 2 - Input: Real world image with rich background and lighting condition

In most of the cases of fruit detection from image, researchers' uses well-known dataset images such as Fruit-360 [9] or use high configuration 3-D sensor [43] to capture fruit image. In these cases images are captured by good photographers with cameras of high-resolution and lens aperture and usually these images are captured by blurring the background and focusing the foreground and also any kind of misleading lighting condition is handled by editing. But, what about a real life cameraman who has a simple mobile phone camera with no background eliminating or other facility? Giving priority to this phenomenon we have evaluated our model for Case 2 type images. Fig. 16(a) shows a real-world image captured by ourselves using a mobile phone camera which has both lighting condition and rich background. Fig. 16(a) shows the detection results found by our model using Fig. 16(b) as input.



(a) Input image.



(b) Output single fruit images detected.

Fig.16. Case 2 - Fruit detection from real world image with noisy background.

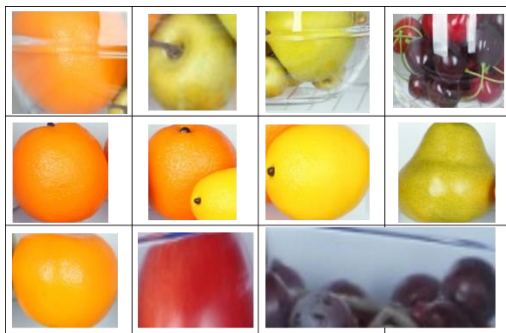
Case 3 - Input: Image from inside a refrigerator

Many of the recent research models are working for object detection from inside a refrigerator. The model proposed in [44] works for detecting fruits inside a refrigerator to make refrigerator smart. Smart refrigerator with fruit detection capability can help us detecting and recognizing fruits, number of every fruit and even freshness or lameness of fruit. Considering these reasons, to make our model even useful for smart refrigerators, we have evaluated our model's capability in case of fruit detection from an image inside a refrigerator and our

model has proved to find better result in this case. Fig. 17(a) shows input image and Fig. 17(b) shows the detection results found by our model from this input.



(a) Input image.



(b) Output single fruit images detected.

Fig.17. Case 3 - Fruit detection from real world image from inside a refrigerator.

Case 4 - Input: Image with fruits overlapping one another

In general, set of fruits are likely to be found in a place or in a bowl in a condition where those are closely connected to one another or simple in an overlapping condition. Most of the state of the arts like model [9, 32] ignores this phenomenon in case of building a FDR model. They used to evaluate their model using single or multiple non overlapping fruit's images. In this regard, one of the major contributions of our work is that this model is capable of detecting and recognizing every fruits separately even they exists in the same region and in an overlapping condition. In Section III we have described how our model handles overlapping. Fig. 18(a) shows input image and Fig. 18(b) shows the detection results found from the input.



(a) Input image.



(b) Output single fruit images detected.

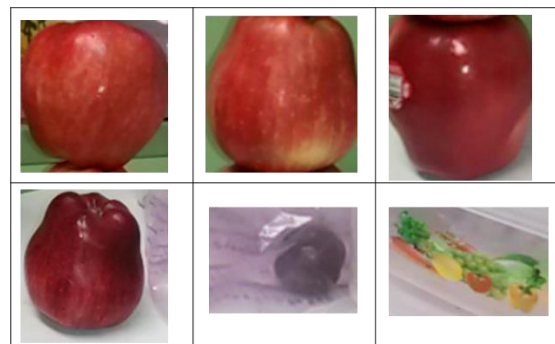
Fig.18. Case 4 - Fruit detection from image of overlapped fruit.

Case 5 - Input: Image from dataset [28]

In dataset [28], there are 1778 images for object detection set. The authors of [28] have taken every image on a different background. We have executed our fruit detection algorithm on almost 1000 images from [28] and were able to detect every fruit's region from those images. Fig. 19(a) shows input image (i.e., an image from dataset [28]) and Fig. 19(b) shows the detection results found by our model from the input. In case of this image our detection procedure is even capable of detecting the fruit from the background existing inside the poly bag as the 5th image of output. Also, it detects the fruit image on the box as the 6th image of output.



(a) Input image.



(b) Output single fruit images detected.

Fig.19. Case 5 - Fruit detection using input image from dataset [28].

2) Comparative analysis in case of region detection

Fruits in images taken from same angel differ in color, shape, size. Fruits generally have spherical shape which can be misclassified often with other non-fruit objects. Therefore, often deep learning based fruit detection approaches needs test images to have the some properties

which are, (1) Images are taken from same angel, (2) Each photo would have a single fruit, (3) Different neural network model is suitable for different data set. Often the same model can provide better result for some datasets not for all dataset. For evaluating fruit detection we have considered many cases. To show the new contribution of

our model we have considered many difficult cases of fruit detection from an image and proved that our model is capable of detecting fruit from any condition or kind of image. Table 3. shows how our model outperforms some of the existing models in case of fruit detection from image avoiding some different situations.

Table 3. Fruit detection phase's effectiveness comparison.

Comparison Topic > Model name v	Overlapping Fruit detection	Fruit detection from Image with rich background	Fruit detection from poor resolution image	Multiple Dataset fruit image detection	Fruit detection from image inside refrigerator	Fruit detection without sliding window, feature detection and CNN
Fruit recognition from images using deep learning [9]	X	X	X	X	X	X
Multi-class fruit detection based on image region selection and improved object proposals[10]	√	√	X	X	X	X
Fruit recognition based on convolution neural network [28]	√	X	X	X	X	X
A code based fruit recognition method via image conversion using multiple features [36]	X	X	X	X	X	X
Proposed Model	√	√	√	√	√	√

B. Fruit Recognition Phase

In this proposed model, as the fruit ROI are detected before testing in neural network, even with small amount of training model we can achieve higher precision. The performance evaluation and comparative analysis of our FDR model in case of fruit's class recognition is described as follows.

1) Recognition model evaluation

As mentioned before, for evaluating our proposed model we have used some of the benchmark datasets along with our own created dataset. For describing the recognition phase's performance and comparative analysis we have explained the different cases regarding different datasets.

a) Recognition model evaluation using dataset [31] and our collected images.

For recognition model evolution, at first we have trained our model using images from dataset [31] as this dataset have 74 types of fruits. In [31] each fruit kind has 490 train images and 164 test images. With this dataset we have tested our model using both their given test images and our detected test images which we have found as the output of fruit detection approach from various kinds of images. Our model performs well on dataset [31] images as here training image and test image have very little difference and most of the time the test images

differ from the train images by only view point. The test images of [31] are all same in size, camera view point and center at the image. This property eventually trains a model with good performance. So, for better evaluation of our model we also test our model on real word image. Table 2. shows the neural network model which have been used for F360 dataset [31].

We have compared our proposed model along with existing works. Fig. 20 shows some of the recognition results, here recognition model uses train images from [31] and test images from the result of our fruit detection process.

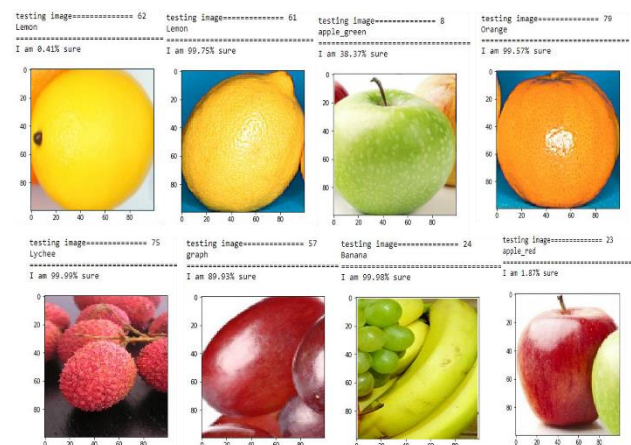


Fig.20. Some example recognition results.

In addition, in order to evaluate our model's recognition performance we have used test images which differ from the train images by viewpoint, color intensity, resolution, lighting, object position and so many other perspectives, we have prepared our own dataset which is combination of different benchmark dataset and images take by us. This is another significant contribution of our work, as we find there is very few dataset for fruit image containing image form different view point and with noisy background. In addition, our model proves to be capable of recognizing different fruit with a better accuracy. In our proposed work, we have achieved our targeted level of success by using a very small training dataset in comparison to VOC [5] and other image data set, where thousands of data have been used for only 2 categories such as cat and dog. This significantly is an efficient characteristic in terms of time and memory consumption for training the model.

We have also evaluated our recognition model by testing the given test images of [31]. The average precision rate of our detection model of detecting 74 types of fruits from [31] was 0.9875. In Fig. 21 the f1 score, recall and precision of classifying 10 types of fruits is shown.

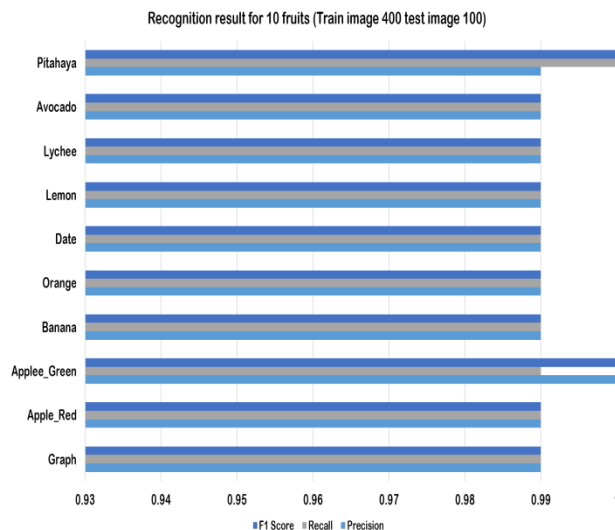


Fig.21. F1 score, recall and precision of classifying 10 types of fruits from [31].

b) Recognition model evaluation using dataset [28].

The dataset [28] is created by the authors of [10]. It contains 5 class of fruits. Table 3. shows the neural network model which have been used for dataset [28]. For our proposed recognition model evolution we have organized our recognition model using its training and test images. Executing our model through this dataset we were able to find an average recognition rate of 0.9783. For this test, the training image for each fruit was 1600 and test images were created from detection set of [28] using our fruit detection approach. Table 4. shows the precision, recall and f1 score of classifying each type of fruit from dataset [28].

Table 4. Precision, recall and f1 score of classifying each type of fruit from dataset [28].

	precision	recall	f1-Score
apple	1.00	0.98	0.99
kiwi	0.97	0.95	0.96
orange	0.97	0.97	0.97
pear	0.95	0.97	0.96
persimmon	0.97	1.00	0.99
avg/Total	0.98	0.97	0.97
loss	0.0237		
accuracy	0.9783		
epoch	10		
Batch size	16		

We define our recognition model is better as by applying it we were able to test 500 images of each fruit type with a 0.98 percent average precision rate even by training the model with only 500 images for each fruit class. And the amount of loss is only about 0.016. We have achieved this result just after 10 epochs and the batch size is only 16. Fig. 22 shows the training loss versus validation loss curve of the recognition training model using dataset [28].

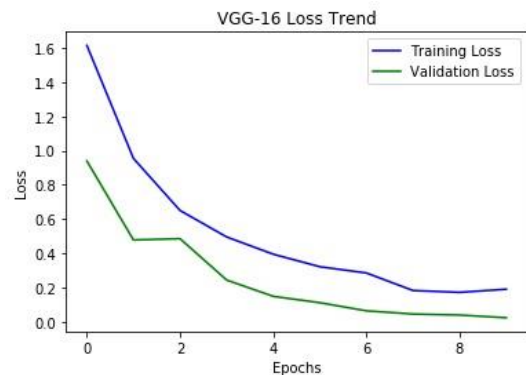


Fig.22. Training loss versus validation loss curve of the recognition training model using dataset [28].

This figure clearly shows that our model makes a good fit as here the validation error is lower than the training error. Running so many test cases, using dropout at a favorable label, checking early stopping at the end of an epoch give us a good model for recognition.

2) Comparative analysis in case of recognition model evaluation

For evaluating fruit recognition, we have compared our model with some existing models by using recognition accuracy, precision, recall and f1 score measure as evaluation measure. We have also compared our model's miss rate (e.g. rate of misclassification) with other model and in most of the cases we were able to prove that our model is performing comparatively better than others.

a) Performance of fruit detection and recognition on dataset [31]

We have appraised our model using dataset [31], the same way as [9] to assemble a comparison. In [31] there

are 49561 images of fruits spread across 74 classes. Here in their proposed model, every fruit class has more or less 490 training images and more or less 160 test images. So, eventually training set consists of 37101 images and testing set consists of 12460 images. Both of these training and test images are scaled down to 100×100 pixels and each image just contains the fruit region without any further background. The model (e.g., convolutional network in the form of a recurrent convolutional network) described in [9] gets a calculated accuracy of about 96.19 with their training and test set. Where by accuracy we mean the ratio of the number of samples recognized correctly with the number of samples tested. Using their dataset [31] we have made a training and testing set consists of exactly same number of images as model [9]. In case of classifying these 74 class of fruits our model gets an accuracy of about **98.75**.

b) Average precision and F1 score comparison for fruit recognition

Fig. 23 shows the comparative performance analysis of the proposed FDR model along with some existing model, in terms of Average Precision (AP) and F1 score. Here for this comparison we have used the dataset available in [28] which is created by MCRD [10]. According to [10], The AP value of MCRD [10] is 0.9747, which is higher than their four baseline models. The average precision of some of these baselines are as following: EdgeBoxes [29]: 0.9138, DPM [30]: 0.9575, and Faster RCNN [32]: 0.9662. Here, the F1 score measure is computed from Precision (P) and Recall (R) as follows

$$F1 = 2PR/(P + R) \quad (10)$$

The F1 measure of MCRD [10] is 0.9661, which is higher than their four baselines, Sliding window: 0.8829, Edge- Boxes: 0.8538, DPM: 0.9213, CNN + SVM: 0.9559, Cascade: 0.8933 and Faster RCNN: 0.9577. Using the same approach, dataset and combination of training and test images as [10] we have evaluated our model in case of Average Precision (AP) and F1 score. The AP value of our proposed model for the five class fruit dataset [28] is **0.9872** and the F1 score measure of our proposed model is **0.97**. We have compared this found Average Precision (AP) and F1 score result with MCRD [10], EdgeBoxes [29], DPM [30] and F- RCNN [17]. Fig. 23 shows the comparison result.

It is clearly shown in Fig. 23 that our model outperforms the existing [10, 17, 29, 30] models in case of both average precision and F1 score. Our FDR model also outperforms model [10] on the case of miss rate. Where we get a miss rate or loss rate of about 0.0237 which is comparatively better than 0.0377 of [10].

Our model is giving comparatively better result in case of both fruit detection and recognition. Because, at first we run a detection phase on every input image which outputs every fruit region (e.g. without much of the background) as a set of test image. And in case of recognition, the training is done with images that only contains the fruit region in a rectangular shape. Finally,

the detection result is passed through the recognition model to get the actual class. This overall process eventually makes the model capable of dealing with only the fruit region and comparing them to get a class which also helps the model avoid dealing with background region and get manipulated by it. On the grounds our model gives a better result.

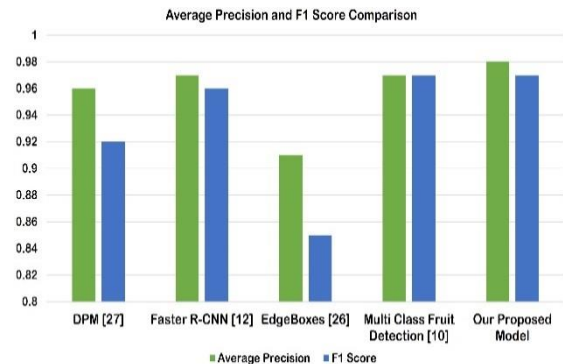


Fig.23. Average Precision and F1 Score Comparison.

We have also made some failure analysis where the model does not work properly. Fig. 24 shows some of the miss classified results. In most of the cases this misclassification is happening because the input image is a much blurry one (e.g., like Fig. 24(d) in Fig. 24). By blurry image we mean images with excessive amounts of blurring (e.g. low amount of high frequencies). In these kind of images the amount of Laplacian variance is very low and so recognizing object becomes hard.

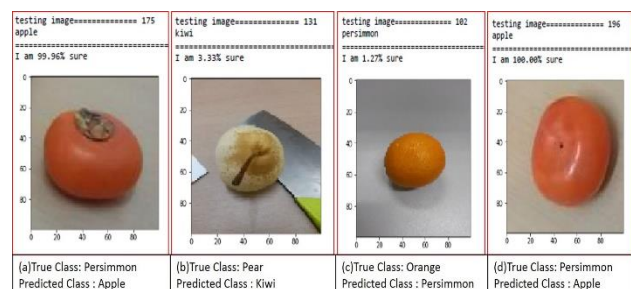


Fig.24. Some incorrectly recognized results.

Some other cases where misclassification is happening are with images that are the incorrect output of detection phase and contain much background with fruit location (e.g., like Fig. 24(b) and Fig. 24(c) in Fig. 24). And finally this misclassification is happening at some images where detecting the correct class is even tough for human. A fruit image as in Fig. 24(a) looks exactly as an image of an apple where it is an image of a persimmon can be considered as an example case.

V. CONCLUSION

One of the major contributions of this paper is that it proposes a novel model for both fruit detection and

recognition. This proposed model, as explained before, has reduced the computation time and complexities drastically. Instead of going with the traditional feature based or sliding window based detection procedure, this model uses a new set of approaches to separate every object from an image very well. Previously, in this paper, we have explained how these approaches make this model more capable of detecting fruit from any kind of image. We also have developed an efficient fruit recognition procedure using perfect combination of convolution, pooling and dense layer. As the recognition model does not need to execute the object finding part on every input image it can save that effort for other operations. These new detection techniques along with the well-designed recognition approach of our FDR model outperform some of the well-known existing models in performance. In addition, this FDR model is not a database oriented model as it provides high accuracy rate on several datasets in case of fruit classification. Furthermore, our proposed model provides superior performance in fruit region detection and as a better recognizer it has archived an average precision rate 0.9875 using images of dataset [31] and 0.98 using images of dataset [28].

Up to now our model has proved to perform well on many different kinds of images in case of fruit detection and recognition. In future, we look forward to improve apply this model on real time fruit detection from video. We also look forward to expanding our system in such a way that a single ODR framework can detect and recognize so many different kinds of objects with better accuracy and less power consumption.

ACKNOWLEDGEMENTS

The authors would like to express their token of thanks to Computer Science and Engineering Discipline, Khulna University, Khulna, Bangladesh for their efforts.

REFERENCES

- [1] Y. LeCUN, B. Boser, J. Denker, et al., "Hubbard. w., and jackel, ld:âAÿhand written digit recognition with a back-propagation network âAÿ, in âAÿ, Advances in neural information processing systems 2, 396404 (1989).
- [2] P. Sermanet, S. Chintala, and Y. LeCun, "Convolutional neural networks applied to house numbers digit classification," in Pattern Recognition (ICPR), 2012 21st International Conference on, 3288–3291, IEEE (2012).
- [3] D. Ciresan, U. Meier, and J. Schmidhuber, "Multicolumn deep neural networks for image classification," (2012). [doi:10.1109/cvpr.2012.6248110].
- [4] P. Sermanet and Y. LeCun, "Traffic sign recognition with multi-scale convolutional networks," in Neural Networks (IJCNN), The 2011 International Joint Conference on, 2809–2813, IEEE (2011) [doi:10.1109/IJCNN.2011.6033589].
- [5] M. Everingham, L. Van Gool, C. K. Williams, et al., "The pascal visual object classes (voc) challenge," International journal of computer vision 88(2), 303–338 (2010). [doi:10.1007/s11263-009-0275-4].
- [6] J. Deng, W. Dong, R. Socher, et al., "Imagenet: A large-scale hierarchical image database," in Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, 248–255, Ieee (2009). [doi:10.1109/CVPR.2009.5206848].
- [7] K. Jarrett, K. Kavukcuoglu, Y. LeCun, et al., "What is the best multi-stage architecture for object recognition?," in Computer Vision, 2009 IEEE 12th International Conference on, 2146–2153, IEEE (2009). [doi:10.1109/ICCV.2009.5459469].
- [8] B. V. Biradar and S. P. Shrikhande, "Flower detection and counting using morphological and segmentation technique," Int. J. Comput. Sci. Inform. Technol 6, 2498–2501 (2015).
- [9] H. Mure,san and M. Oltean, "Fruit recognition from images using deep learning," Acta Universitatis Sapientiae, Informatica 10(1), 26–42 (2018).
- [10] H. Kuang, C. Liu, L. L. H. Chan, et al., "Multi-class fruit detection based on image region selection and improved object proposals," Neurocomputing 283, 241–255 (2018). [doi:10.1016/j.neucom.2017.12.057].
- [11] Y. Lu, D. Allegra, M. Anthimopoulos, et al., "A multi-task learning approach for meal assessment," in Proceedings of the Joint Workshop on Multimedia for Cooking and Eating Activities and Multimedia Assisted Dietary Management, 46–52, ACM (2018).
- [12] Kaggle, "Dogs vs. cats,create an algorithm to distinguish dogs from cats." <https://www.kaggle.com/c/dogs-vs-cats> (2013).
- [13] J. Rakun, D. Stajanko, and D. Zazula, "Detecting fruits in natural scenes by using spatial-frequency based texture analysis and multiview geometry," Computers and Electronics in Agriculture 76(1), 80–88 (2011). [doi:10.1016/j.compag.2011.01.007].
- [14] R. Thendral, A. Suhasini, and N. Senthil, "A comparative analysis of edge and color based segmentation for orange fruit recognition," in Communications and Signal Processing (ICCSP), 2014 International Conference on, 463–466, IEEE (2014). [doi:10.1109/ICCSP.2014.6949884].
- [15] E. Parrish, A. Goksel, et al., "Pictorial pattern recognition applied to fruit harvesting," Transactions of the ASAE 20(5), 822–827 (1977).
- [16] G. R. A. Grand D Esnon and R.Pellenc, "A selfpropelled robot to pick apples," in ASAE paper, (87-1037) (1987).
- [17] S. Ren, K. He, R. Girshick, et al., "Faster r-cnn: Towards real-time object detection with region proposal networks," in Advances in Neural Information Processing Systems 28, C. Cortes, N. D. Lawrence, D. D. Lee, et al., Eds., 91–99, Curran Associates, Inc. (2015).
- [18] J. Redmon, S. Divvala, R. Girshick, et al., "You only look once: Unified, real-time object detection," in Proceedings of the IEEE conference on computer vision and pattern recognition, 779–788 (2016).
- [19] R. Girshick, J. Donahue, T. Darrell, et al., "Rich feature hierarchies for accurate object detection and semantic segmentation," in Proceedings of the IEEE conference on computer vision and pattern recognition, 580–587 (2014).
- [20] F. Garcia, J. Cervantes, A. Lopez, et al., "Fruit classification by extracting color chromaticity, shape and texture features: towards an application for supermarkets," IEEE Latin America Transactions 14(7), 3434–3443 (2016). [doi:10.1109/TLA.2016.7587652].
- [21] S. Riyadi, A. J. Ishak, M. M. Mustafa, et al., "Wavelet-based feature extraction technique for fruit shape classification," in Mechatronics and Its Applications, 2008. ISMA 2008. 5th International Symposium on, 1–5, IEEE (2008). "[doi:10.1109/ISMA.2008.4648858]".
- [22] S. Jana and R. Parekh, "Shape-based fruit recognition and classification," in International Conference on

- Computational Intelligence, Communications, and Business Analytics, 184–196, Springer (2017). [doi:10.1007/978-981-10-6430-2_15].
- [23] J.-y. Kim, M. Vogl, and S.-D. Kim, “A code based fruit recognition method via image conversion using multiple features,” in IT Convergence and Security (ICITCS), 2014 International Conference on, 1–4, IEEE (2014). [doi:10.1109/ICITCS.2014.7021706].
- [24] C. Hung, J. Underwood, J. Nieto, et al., “A feature learning based approach for automated fruit yield estimation,” in Field and Service Robotics, 485–498, Springer (2015). [doi:10.1007/978-3-319-07488-7_33].
- [25] Z. S. Pothan and S. Nuske, “Texture-based fruit detection via images using the smooth patterns on the fruit,” in Robotics and Automation (ICRA), 2016 IEEE International Conference on, 5171–5176, IEEE (2016). [doi:10.1109/ICRA.2016.7487722].
- [26] H. N. Patel, R. Jain, and M. V. Joshi, “Fruit detection using improved multiple features based algorithm,” International journal of computer applications 13(2), 1–5 (2011).
- [27] W. Liu, D. Anguelov, D. Erhan, et al., “Ssd: Single shot multibox detector,” in European conference on computer vision, 21–37, Springer (2016). [doi:10.1007/978-3-319-46448-0_2].
- [28] H. Kuang, “Hulin kuang. the university of calgary..” https://www.researchgate.net/profile/Hulin_Kuang.
- [29] C. L. Zitnick and P. Dollár, “Edge boxes: Locating object proposals from edges,” in European conference on computer vision, 391–405, Springer (2014). [doi:10.1007/978-3-319-10602-1_26].
- [30] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, et al., “Object detection with discriminatively trained part-based models,” IEEE transactions on pattern analysis and machine intelligence 32(9), 1627–1645 (2010). [doi:10.1109/TPAMI.2009.167].
- [31] Kaggle, “Fruits 360 dataset.” <https://www.kaggle.com/moltean/fruits>.
- [32] L. Hou, Q. Wu, Q. Sun, et al., “Fruit recognition based on convolution neural network,” in Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), 2016 12th International Conference on, 18–22, IEEE (2016). [doi:10.1109/FSKD.2016.7603144].
- [33] L. Jidong, Z. De-An, J. Wei, et al., “Recognition of apple fruit in natural environment,” Optik-International Journal for Light and Electron Optics 127(3), 1354–1362 (2016). [doi:10.1016/j.ijleo.2015.10.177].
- [34] R. Khan, T. Fariha Raisa, and R. Debnath, “An efficient contour based fine-grained algorithm for multi category object detection,” Journal of Image and Graphics 6, 127–136 (2018).
- [35] E. S. Gastal and M. M. Oliveira, “Domain transform for edge-aware image and video processing,” in ACM Transactions on Graphics (ToG), 30(4), 69, ACM (2011). [doi:10.1145/2010324.1964964].
- [36] E. Gastal, “Non photorealistic rendering using opencv(python, c++) | learn opencv.” <https://www.learnopencv.com/non-photorealisticrendering- using-opencv-python-c/>.
- [37] P. O. A. Thresholding, “Adaptive Thresholdings,” (2003). [Online; last accessed 06-April-2019].
- [38] S. F. BogoToBogo_K Hong Ph.D. Golden Gate Ave, “Image thresholding and segmentation..” https://www.bogotobogo.com/python/OpenCV_Python/python_opencv3_Image_Global_Thresholding_Adaptive_Thresholding_Otsus_Binarization_Segmentations.php (2013). [Online; accessed 19-July-2018].
- [39] Homepages.inf.ed.ac.uk, “Morphology - Dilation,” (2003). [Online; last accessed 06-April-2019].
- [40] O. S. C. V. OpenCV, “Morphological transformations.” https://docs.opencv.org/3.4/d9/d61/tutorial_py_morphological_ops.html.
- [41] J. T. Springenberg, A. Dosovitskiy, T. Brox, et al., “Striving for simplicity: The all convolutional net,” arXiv preprint arXiv:1412.6806 (2014).
- [42] T. Github, “Build a Convolutional Neural Network using Estimators.” <https://www.tensorflow.org/tutorials/estimators/cnn> (2008). [Online; accessed 19-July-2008].
- [43] A. R. Jiménez, A. K. Jain, R. Ceres, et al., “Automatic fruit recognition: a survey and new results using range/attenuation images,” Pattern recognition 32(10), 1719–1736 (1999).
- [44] W. Zhang, Y. Zhang, J. Zhai, et al., “Multi-source data fusion using deep learning for smart refrigerators,” Computers in Industry 95, 15–21 (2018).

Authors' Profiles



Rafflesia Khan is a student of M.Sc. at Computer Science and Engineering discipline, Khulna University, Khulna, Bangladesh. She has completed her Bachelor degree from Computer Science and Engineering discipline, Khulna University, Khulna, Bangladesh in 2017. Her research areas of interest are image processing, visual object detection & recognition, machine learning, pattern recognition, and internet of things security.



Rameswar Debnath is a Professor of Computer Science and Engineering discipline at Khulna University, Khulna, Bangladesh. He has completed his Bachelor degree from Computer Science and Engineering discipline, Khulna University, Khulna, Bangladesh in 1997. He has received his Masters in Engineering degree and PhD degree in Computer Science and Engineering from the University of Electro-Communications, Tokyo, Japan in 2002 and 2005 respectively. His research areas of interest are image processing, statistical machine learning and its applications to pattern recognition, visual object detection and natural language processing.

How to cite this paper: Rafflesia Khan, Rameswar Debnath, " Multi Class Fruit Classification Using Efficient Object Detection and Recognition Techniques", International Journal of Image, Graphics and Signal Processing(IJIGSP), Vol.11, No.8, pp. 1-18, 2019.DOI: 10.5815/ijigsp.2019.08.01