IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 22, Issue 2, Ser. I (Mar - Apr 2020), PP 01-13
www.iosrjournals.org

Convolutional Neural Networks (CNN) for Detecting Fruit Information Using Machine Learning Techniques

Fouzia Risdin[1], Pronab Kumar Mondal[1], Kazi Mahmudul Hassan[1]
[1]*(Jatiya Kabi Kazi Nazrul Islam University, Trishal, Mymensingh, Bangladesh)*
**Corresponding author:Fouzia Risdin*
Received; Accepted

Abstract:*This paper introduces a great approach to detection of fruits using deep convolutional neural networks. The aim is to build an accurate, fast and reliable fruit detection system using machine learning facts. The proposed system has applied convolutional neural network (CNN) to the tasks of detecting fruit images. Because of the wide diversity of types of fruit, image detection of fruit items is generally very difficult. However, deep learning has been shown recently to be a very powerful image detection technique, and CNN is a state-of-the-art approach to deep learning. A dataset is constructed that contains most frequent fruit items in a publicly available fruit-logging system, and used it to evaluate detection performance. CNN showed higher accuracy than did traditional support-vector-machine-based methods with handcrafted features. For fruit image detection, CNN also showed significantly higher accuracy than a conventional method did. Besides, this approach is also much quicker to deploy for new fruits. In this study, the model is retrained to perform the detection of four fruits based on a new dataset consist of 2403 data belonging 4 fruit classes. All the data in the data set were collected using Smart-phone camera and believed to be unique in every sense. The study has shown an accuracy of 99.89% which turned out to be promising. Furthermore, the study has carried out based on real-life scenario and the result was out of the mark. The ambition of this work is to extend system for other objects and real life applications.*

Keywords: *visual fruit detection; deep convolutional neural network; real-time performance; rapid training; detection accuracy and graph.*

Date of Submission: 17-03-2020 Date of Acceptance: 02-04-2020

I. Introduction

Fruits are common food consumed by human since prehistoric era. They make important nutritional contribution to human well-being because of their high nutritive value. It is need to ensure the quality of fruits that are consumed in any places. To do this, a fruit detection system can be established that can recognizes various types of fruits from images that are captured by any digital camera or smart phone from various places. This system will help us to check the quality of fruits and also help us to develop a robotic harvesting system from orchards. To develop the system, machine learning techniques have used in this system. Accurate and efficient fruit detection is of critical importance for a machine. A number of factors make fruit detection system a challenging task: Fruits occur in scenes of varying illumination, can be turned off by other objects and are sometimes hard to visually differentiate from the background. An ideal fruit detection system is accurate, can be trained on obtainable data sets, produces its predictions in real time, adapts to different types of various fruits and works using different modalities, such as infrared images and color images.

In recent years, deep learning methods have made important progress in addressing these requirements. Fruit detection can be considered and formulated as an image segmentation problem. The proposed system have used Convolutional Neural Networks (CNN) for detecting fruit information system form images. The proposed method is tried to overcome all the limitations of the related works of fruit detection system and obtain a high accuracy rather than other works. The system has provided performance with simplicity and efficiency. The contributions of this paper are therefore

- To develop a high-performance fruit detection system that can be quickly trained with a small number of images using a CNN
- To study the performance of the convolutional neural networks for detecting objects of images.
- To know how to build a fruit detection system using machine learning approaches based on convolutional neural networks.
- To improve the detection quality using deep convolutional neural networks compared to support vector machine.

The remainder of the paper consists of the following. Section 2 introduces literature review. Section 3 presents the descriptive comparisons between support vector machine (SVM) and the proposed Convolutional Neural Network (CNN) for fruit detection. Reasons for using CNN are also addressed in this section. Methodological descriptions of CNN for fruit detection are described in Section 4. The experimental results demonstrate in Section 5.Conclusions and future works are drawn in Section 6.

II. Literature Review

Machine learning approach for recognizing fruit is quite a familiar seen now a day. Many researchers applied this approach in their research but some of the study achieved a noticeable success. Although many researchers have noticed the problem of fruit detection, such as the works submitted in [2–7], the problem of building a fast and reliable fruit detection system persists, as found in the survey by [8]. It is due to high variation in the appearance of the fruits in field settings, including color, size, texture, shape and reflectance properties. Furthermore, in the majority of these settings, the fruits are partially abstracted and subject to continually-changing illumination and shadow conditions.

Various works introduced in the literature address the problem of fruit detection as an image segmentation problem i.e., fruit vs. background. Wang et al. [5] examined the issue of apple detection for yield prediction. They developed a system that detected apples based on their color and distinctive specular reflection pattern. Further information, for example the average size of apples, was used to either remove erroneous detections or to split regions that could contain various types of apples. Another heuristic appointed was to accept as detections only those regions that were mostly round. Bac et al. [6] proposed a segmentation approach for sweet peppers. They used a six band multi-spectral camera and used a range of features, including the raw multispectral data, normalized difference indices, as well as entropy-based texture features. Experiments in a highly strict glasshouse environment showed that this approach produced accurate segmentation results reasonably. However, the authors noted that it was not so much accurate to build a reliable obstacle map. Hung et al. [7] proposed the use of conditional random fields for almond segmentation. They proposed a five-class segmentation approach, which learned features using a Sparse Auto Encoder (SAE). These features then were used within a CRF framework and were shown to outperform the previous work. They gained impressive segmentation performance, but did not perform object detection. Furthermore, they noted that occlusion presented a major challenge. Intuitively, such an approach is only able to cope with low levels of occlusion.

Deep neural networks have already shown high promise when used for various multi-modal systems in regions outside agricultural automation, such as in [20], where audio/video has been used very successfully, and in [21, 22], where image/depth shows a better performance compared to the utilization of each modality alone. This work follows the same approach and exhibits the use of a multi-modal region-based fruit detection system. CNN is a very powerful algorithm closely related to deep neural networks which is widely used for image classification and object detection. The powerful feature extraction capabilities and the hierarchical structure from an image make CNN a very strong algorithm for various image and object recognition tasks.

III. Comparison of CNN With SVM

Support Vector Machine (SVM) can be used for both regression and classification tasks. But, it is widely used in classification objectives. However SVM algorithm has several key parameters that need to be set correctly to achieve the best classification results for any given problem. It is effective in that cases where number of dimensions is greater than the number of samples. But this algorithm is not suitable for large data sets. Because it needs much time for training. It differentiates the two classes appropriately. But it does not perform very well, when the data set has more noise. Besides the support vector classifier works by placing data points, above and below the classifying hyper plane there is no probabilistic explanation for the classification. So SVM is not worked very well for data prediction. In [1], SVM is used for mangoes classifications for defective and non-defective cases. Here FCM and K-Means algorithms are also used. But for FCM, SVM classification results are not so good. Besides SVM is used only for one fruit detection. It cannot be performed for various fruits and various combination of color features.

CNN (Convolutional Neural Network) algorithm is more powerful algorithm for classifications. CNN involves various classes for classifications compared to SVM where classification results are also so good. Besides CNN minimizes the hyper parameters used in the algorithm. As a result it needs not much time for training. The training and testing accuracy of CNN is very high compared to SVM method. That means it is relatively simple, quick to train, and easy to understand.

IV. Methodologies

Compared to Regular Neural Networks, Convolutional Neural Networks [27] have a different architecture. In regular Neural Networks, it transforms an input by putting it through a series of hidden layers. Here every layer is made up of a set of neurons, where each layer is fully connected to all neurons in the layer before. Finally, there is an architecture i.e. a last fully-connected layer — the output layer — that represent the predictions.

There are a bit different in Convolutional Neural Networks. At first, the layers are organized in 3 dimensions: width, height and depth. Here, the neurons in one layer do not connect to all the neurons in the next layer but only to a small region of it. Lastly, the final output of the system will be reduced to a single vector of probability scores, organized along the depth dimension.

CNN is considered as two major parts:

- **Feature Extraction**
  In feature extraction part, the network will perform a series of convolutions and pooling operations during which the features are detected. If we had a picture of a tiger, this is the part where the network would recognize its pelage, two ears, and four legs.
- **Classification**
  In the case of classification, the fully connected layers will serve as a classifier on top of these extracted features. Here, they will assign a probability for the object on the image being what the algorithm predicts it is.

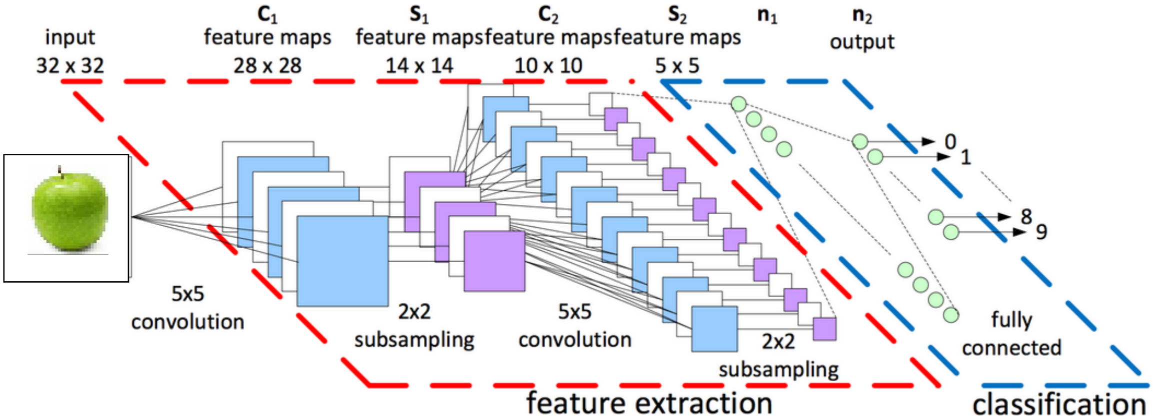


Figure 4.1: Convolutional Neural Networks architecture

Fully Connected Layer (FC Layer) of Classification: Adding a Fully-Connected layer [26] is a (usually) cheap way of learning non-linear combinations of the high-level features as represented by the output of the convolutional layer. In that space, the Fully-Connected layer is learning a possibly non-linear function. Example of CNN network is given below:

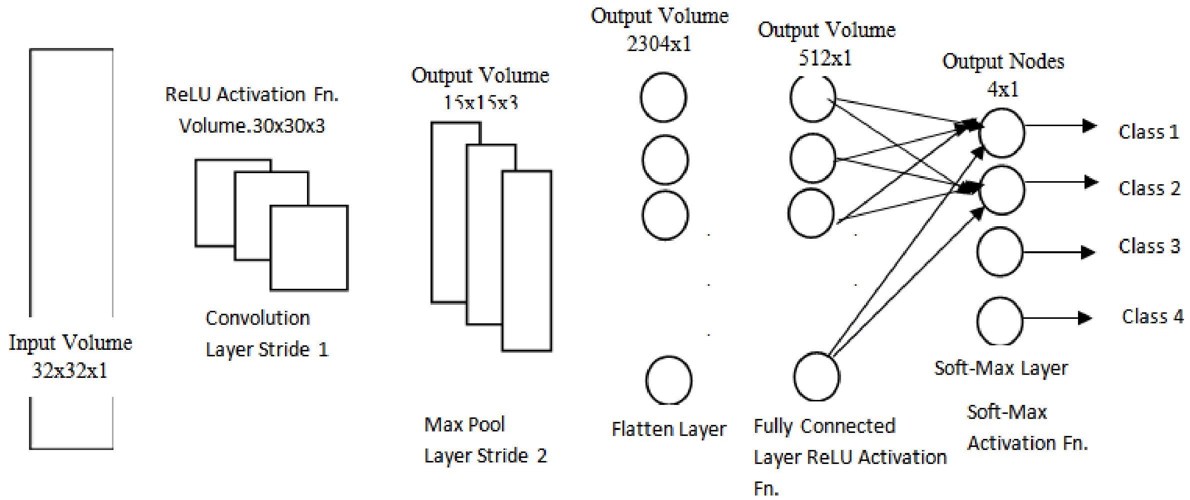


Figure 4.2: Fully Connected model

Here, the input images are converted into a suitable form and flatten the image into a column vector. Now the flattened output is fed to a feed-forward neural network and backpropagation applied to every iteration of training. After a series of epochs, the model is able to differentiate between dominating and certain low-level features in images and classify them using the Softmax Classification technique.

All the components that are needed to build a CNN: Convolution, ReLU and Pooling. Here the output of max pooling is fed into the classifier which is usually a multi-layer perceptron layer. In CNNs these layers are used more than once i.e. Convolution ->ReLU -> Max-Pool -> Convolution ->ReLU -> Max-Pool and so on. Now for the classification part, fully connected layer is used which involves ReLU->Dense->Soft-max and so on. Throughout the study Convolutional Neural Network is used to justify images of fruits containing 4 different classes and achieved accuracy of 99.89%.

A block diagram is a short road map for that graphically represents how the data moves through the existing system. The block diagram shown in figure 4.3 is used in design process. The block diagram provides facilitating communication between us and user. It shows the input and output information i.e. what kinds of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. However it does not show information about the timing of processes but shows the work procedure of the processes.

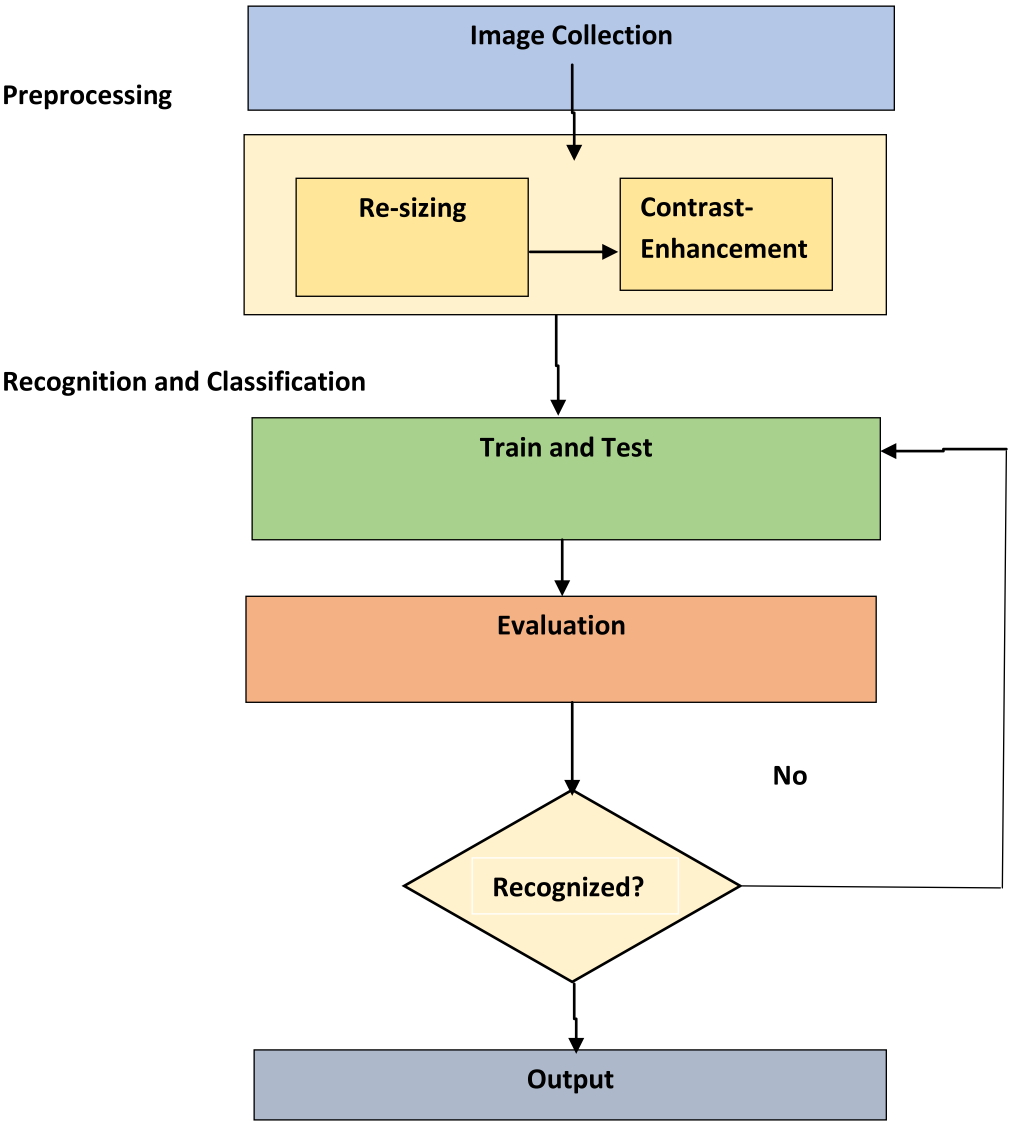


Figure 4.3: Block Diagram of Proposed System

4.1 Datasets

The dataset created by the images captured by smart phone. This dataset of fruits which is categorized into four classes. A challenging data set of 4 fruits categories, with 2403 real world images in total are introduced. The images were collected from different fruit shops with various angles. It incorporates different yet in addition outwardly and semantically comparative fruit classes where each class consists of 565 of image among which 100 are manually reviewed test images and 465 are training images.

Figure 4.4. Example of dataset

4.2 Preprocessing

As the collection process of the data by smart phone the images were in different shape and sizes and training a convolutional neural network on crude pictures will most likely lead to terrible classification exhibitions. So the images are resized into square shape (256 x 256 pixel) and reduced unnecessary object from the images.

4.3 Architect

This particular model is designed to recognize some fruits those are very familiar to us. Layer like Convolution is used followed by pooling layer. Also dense layer and a few regularization strategies such as batch normalization [21] along with dropout [22] are utilized to design this model. First and Second layer are convolution layer having a size of filter 32 and size of kernel 3. Layer 1 is considered to be the input layer which asks for the size 32x32 of RGB channel. It also used same padding and 1 as stride. Both layers use ReLU (1) activation and have the same property for padding and stride. Output of layer 2 is connected with max pooling layer. It has the pool size of 2 and stride 2. Consider this one as layer 3 which is connected to another convolution layer 4 of filter 64 and kernel size of 3. Layer 4 also holds the same property as layer 2 except the filter size. Another max pooling layer 5 having the pool size of 2 is attached to it. It also has the stride of 2.

$$\mathrm{ReLU}(X) = \mathrm{MAX}(0, X) \qquad (1)$$

Layer 6 a batch normalization layer is placed after layer 5. Batch normalization enables to uses the higher learning rate and makes the learning process quicker. Layer 7 is also a convolution with 128 filter size. Except for the filter size other thing remains the same as the other convolution layer. Layer 7 is attached to a max pooling layer 8 having the pool size of 2. After all of these 8 layers are placed, the outcome is smoothing into an array and undergo a dense layer which is considered as layer 9 with 256 concealed units and normalized with half (50%) dropout. And all the previous flow is associated with a dense layer 10 with 4 units which is fully connected with SoftMax (2) actuation. And this is how the model is built.

$$\sigma(z)_j = \frac{e^{z\ j}}{\sum_{k=1}^{k} e^{z\ k}} \text{ for j=1,...k} \qquad (2)$$

Algorithm for the proposed model

1. Set ADAM(Rate of learning = 0.0001)

2. For 30 iterations in all batch do:
{
i. Set first convolutional layer
Convolution 1(Filter =32, Kernel Size=3, Stride=1, Padding=SAME, Activation=relu)
ii. Rearrange first convolutional layer as second
Convolution 1(Filter =32, Kernel Size=3 Stride=1, Padding=SAME, Activation=relu)
iii. MaxPool 1(Pool Size=2, Stride=2)
a. do Dropout (Rate=25%)

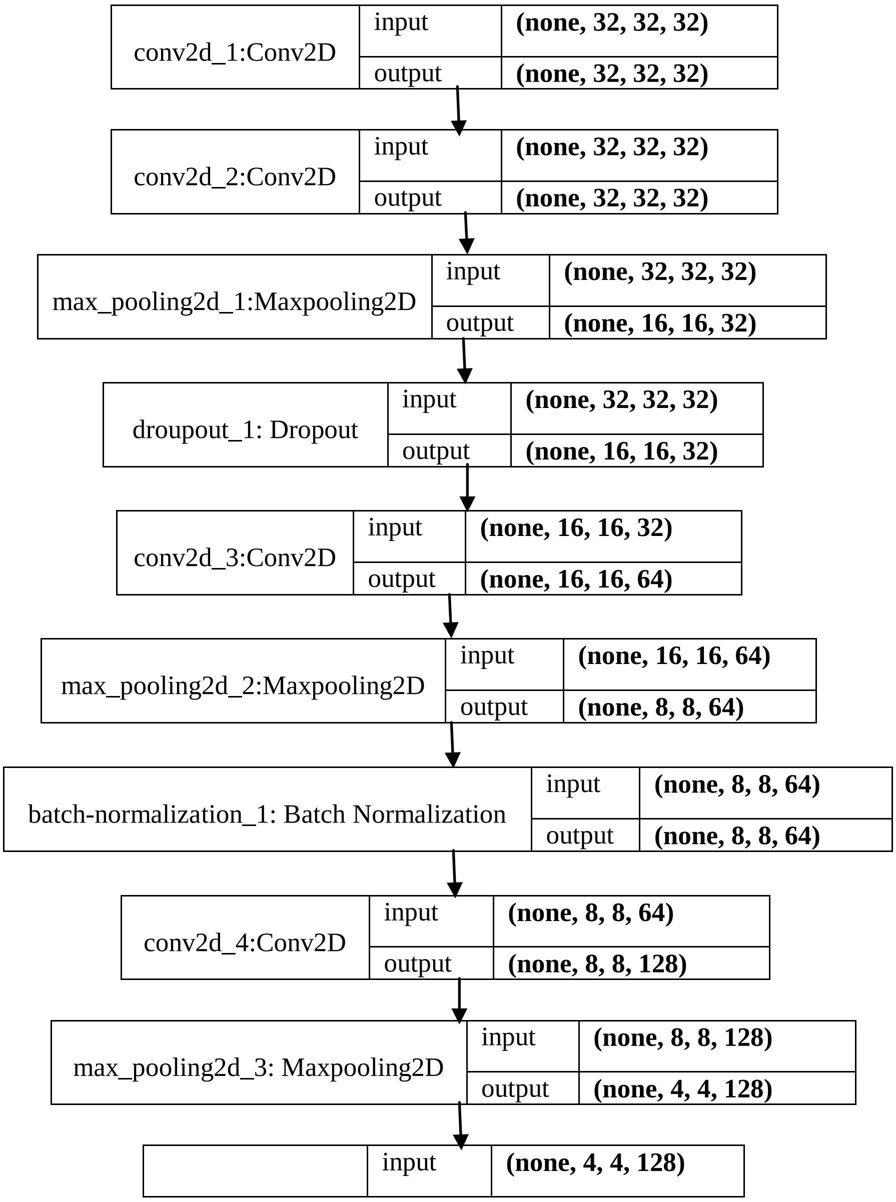
iv. Set third convolutional layer
Convolution 1(Filter =64, Kernel Size=3, Stride=1, Padding=SAME, Activation=relu)
v. MaxPool 2(Pool Size=2, Stride=2)
vi. do Batch Normalization()

vii. Set fourth convolutional layer
Convolution 1(Filter =128, Kernel Size=3, Stride=1, Padding=SAME, Activation=relu)
vii. MaxPool 3(Pool Size=2, Stride=2)
b. do Dropout (Rate=25%)

viii. for fully connected layer do
{
i. Flatten(Units=2304)
j. Dense(Units=256,Activation=relu)
c. do Dropout (Rate=50%)
k. Dense(Units=4,Activation=softmax)
}
}

3. end for

The figure 4.5 represents the Architecture for Proposed Model. Figure 4.5 shows each layers input and output and also indicates which layer feeds into next layer. This architecture also visualizes the models overall structure and layers that included into it step by step.



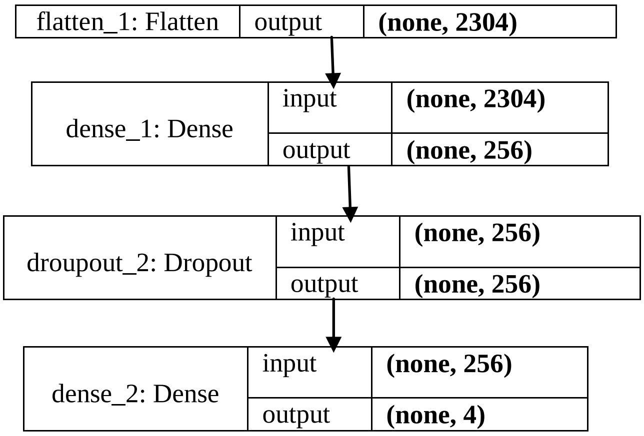


Figure 4.5: Architecture of Fruit Image Recognition

4.4 Optimizer and Learning rate

For minimizing the error of a model optimizer plays a bigger role. Adam optimizer [23] is used in this scenario. It replaced classical stochastic gradient descent method which updates network weights iteratively in training data. For its better execution, it is generally utilized by PC vision researchers. In this model Adam (3) optimizer is used with the learning rate of 0.0001.

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt[1]{\hat{v}_t + \varepsilon}}\hat{m}_t \quad (3)$$

Categorical cross entropy (4) function is used to calculate the error. Ongoing exploration demonstrates that cross entropy shows some quality rather than the other functions out there like classification mistake and also mean squared mistake etc. [24]. As out model is a multi-class classification it is the fittest choice for us.

$$L_i = -\sum_j t_{i,j} \log(P_{i,j}) \quad (4)$$

Call for training a convolutional neural network, Learning rate plays a huge role. The classification will be more perfect if the rate of learning is lower. However optimizer will set aside more exertion to accomplish the global optima reducing the loss. Apart from that higher learning rate may not be the best for the accuracy. As a result achieve the desired goal become harder. The automatic learning rate reduction method is used here to defeat this test [25]. The learning rate is set to 0.0001 toward the starting which is naturally dropped by checking the accuracy of validation.

4.5 Data augmentation

There is a popular theory goes around and that is the more data you have the better performance you get. As a result data augmentation was built to produce more data artificially by handling some operations. For this fruit recognition model augmentation can play a huge role which is beyond imagination. Here, the size of data can be certainly multiply by twice. For data augmentation, each image is rotated by degree of 40, shifted the width and height by 20% randomly, rescaled and zooms by 20%, flipped horizontally and shear with the range of 20%.

The figure 4.6 shows some augmented images when the images of training set are used to train the model.

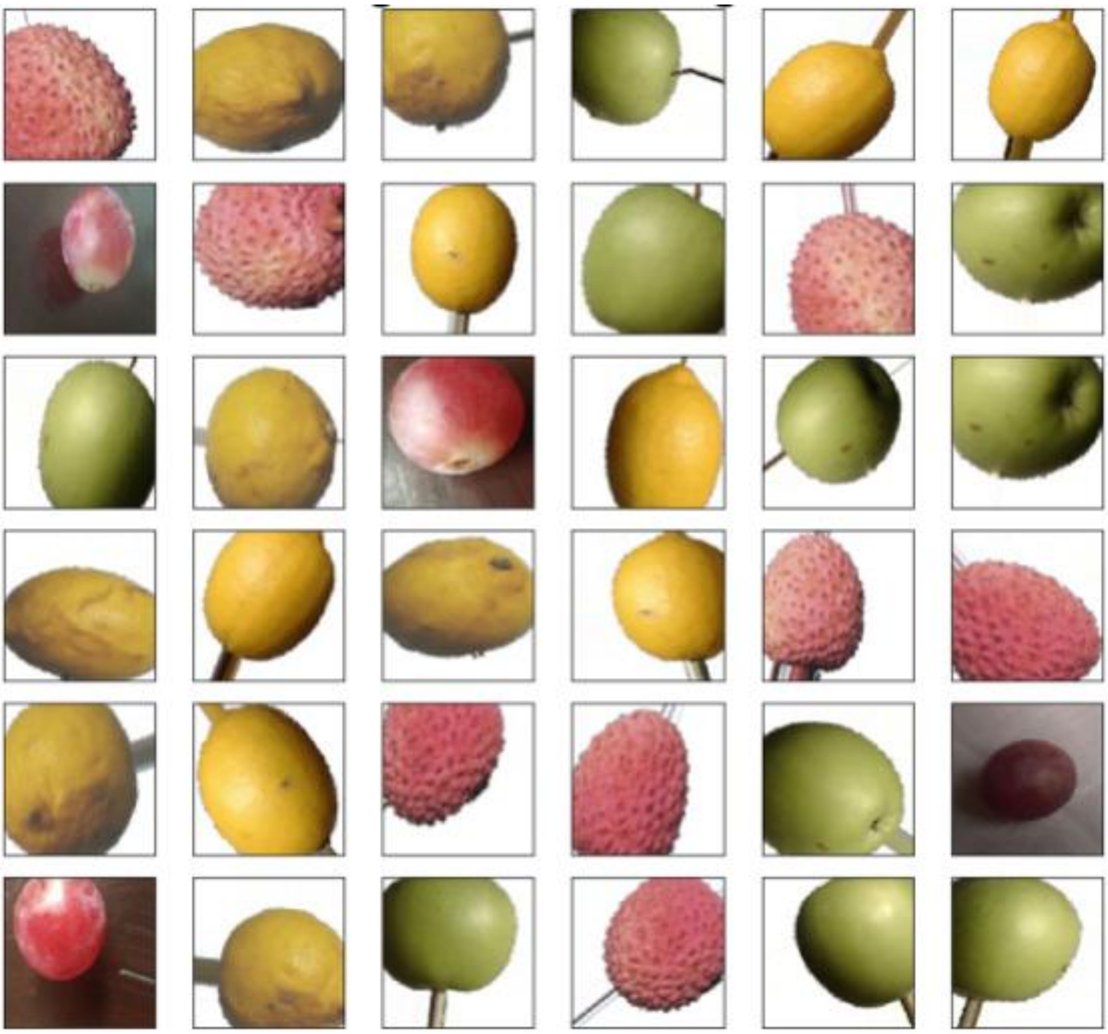


Figure 4.6: Augmented Images

4.6 Training the model

The proposed model is trained on unique dataset that was collected from several places. Batch size of 30 was used here. After 30 epochs the model achieves a satisfactory delicacy. The programmed learning rate decrease recipe helps the analyzer to meet quicker by keeping the learning rate decreased and after the training process is done the rate or learning diminished by 0.0001 to1×10^{-6}.

V. Experimental Results

5.1 Training, Testing and the Validation of the model:

To find out the performance of the model, separate training data, testing data and validation data is created. The training dataset is used to train the model. During training time for checking the model performance validation set is used which helped tuning the hyper-parameters of the model. The test data is used to finding out the performance of final model. The dataset has total 2403 food images. Around 25% of images (569 in total) used for validation and 75% of images (1834 in total) used to train the model. After the training is completed, random images of fruit 5 in total is used. The validation data consisted of various fruit pictures. Test images look like below-

Figure 5.1: Validation data

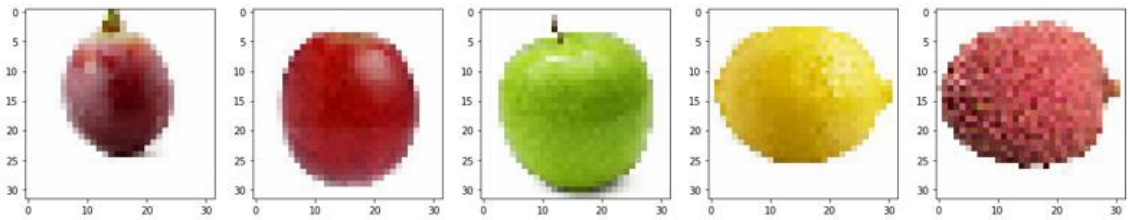


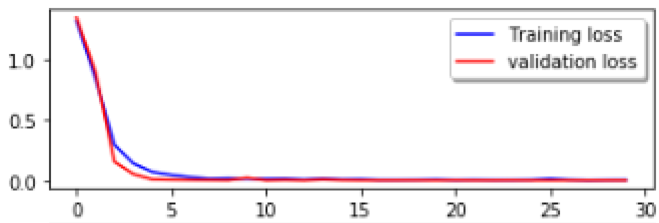
Figure 5.2: Computer Vision of the Validation data

To check the system validation, the images of validation data is used here. In this testing the model predicts the data which class it belongs to. During this type of testing, all of the images of validation data have to be cropped in 256×256 sizes.

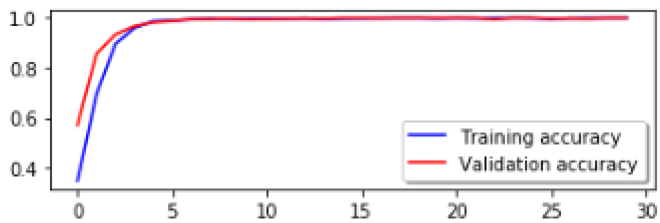
5.2 Model performance

Subsequent to running 30 epochs the proposed model gained the accuracy of 99.84% for the training dataset that was created and 99.89% on the validation dataset. Completing the training session the test on random images went pretty well. The model was able to accomplish a very successive rate. Breaking down the outcome and confusion Matrix it can be assumed that the performance of this model is acceptable for these kinds of fruits. The over-all performance of the model is illustrated in figure 5.3.

In figure 5.3(a), it indicates the training loss and validation loss of overall model performance. A very plain graph indicates the minimize loss for both training and testing of the model. Figure 5.3(b) shows the training accuracy and validation accuracy of the overall model performance. So the graph of figure 5.3 is pretty well for both loss and accuracy of training and testing of the model for datasets.



(a)



(b)

Figure 5.3: (a) Loss of Training and Validation (b) Accuracy of Training and Validation

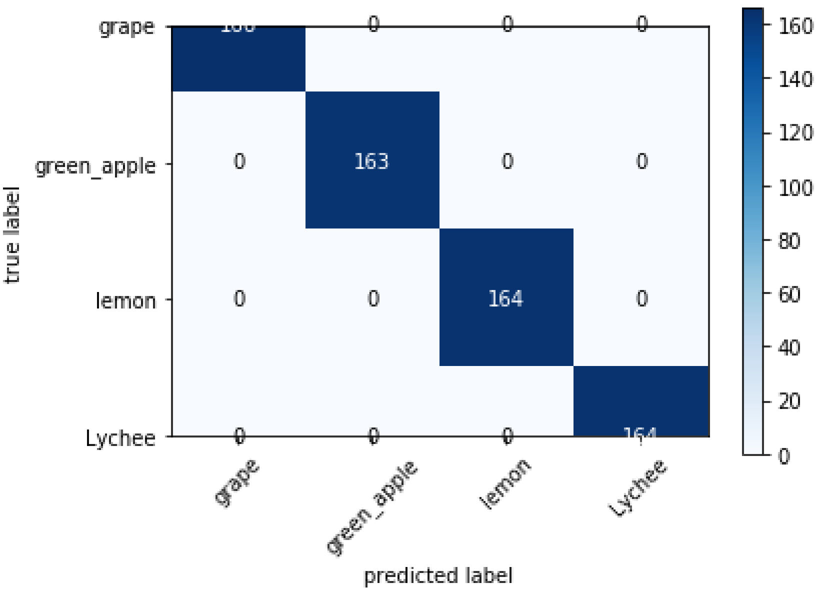
5.3 Confusion matrix

In terms of a classification model performance justification, confusion matrix (1) is widely used. It operates on a set of testing dataset where the true values are familiar. Figure 5.4 represents a drawing which is the confusion matrix for the proposed model. The entries in the matrix are True Positive (TP) rate, False Positive (FP) rate, True Negative (TN) rate, False Negative (FN) rate for each type of dataset. The accuracy is the division of the absolute number of predictions and the predictions that were correct. The accuracy of the confusion matrix is calculated by the following rules:

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{1}$$

```
Confusion Matrix
[[166   0   0   0]
 [  0 163   0   0]
 [  0   0 164   0]
 [  0   0   0 164]]
Classification Report
```

(a)



(b)

Figure 5.4. Confusion Matrix for four fruit Images

In the figure 5.4, we can see that the true label and the predicted label are same for four classes. If the values of confusion matrix of four classes indicate the large numbers then it seems to be good results for the testing data.

5.4 Model Summary

After training and testing the proposed model, the model can be summarized. The visualization of the model summary is given in figure 5.5. The figure also shows the architecture of the proposed model which includes a lot of layers to implement the model. The convolution and max-pooling layers are used in feature extraction part and the dense and soft-max layers are used as the fully connected layer.

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 32, 32, 32)        896
_________________________________________________________________
activation_1 (Activation)    (None, 32, 32, 32)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 30, 30, 32)        9248
_________________________________________________________________
activation_2 (Activation)    (None, 30, 30, 32)        0
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 15, 15, 32)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 15, 15, 32)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 15, 15, 64)        18496
_________________________________________________________________
activation_3 (Activation)    (None, 15, 15, 64)        0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 13, 13, 64)        36928
_________________________________________________________________
activation_4 (Activation)    (None, 13, 13, 64)        0
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 6, 6, 64)          0
_________________________________________________________________
dropout_2 (Dropout)          (None, 6, 6, 64)          0
_________________________________________________________________
flatten_1 (Flatten)          (None, 2304)              0
_________________________________________________________________
dense_1 (Dense)              (None, 512)               1180160
_________________________________________________________________
activation_5 (Activation)    (None, 512)               0
_________________________________________________________________
dropout_3 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_2 (Dense)              (None, 4)                 2052
_________________________________________________________________
activation_6 (Activation)    (None, 4)                 0
=================================================================
Total params: 1,247,780
Trainable params: 1,247,780
Non-trainable params: 0
```

Figure 5.5: Model Summary

5.5 Outcomes of the System

In the proposed system, a model is introduced to recognize fruits from images. During this type of work, a machine learning approach has developed to establish the model. In this study, a dataset of fruits of 4 classes is introduced for recognition. To perform the task of the model, Convolutional Neural Networks (CNNs) is used which was developed to perform on machine learning approaches. This model is able to get accuracy of 99.89% which proved that the performance of this model to recognize fruits from images is more advanced. The high accuracy of the model shows that CNN is very suitable for this kind of fruit recognition and also found a great algorithm for CNN which has implemented successfully for recognition of fruits. The optimized CNN's hyper parameters showed that CNN significantly improved the fruit recognition accuracy compared with a conventional method using a support vector machine (SVM) with hand-crafted features [1].

5.6 Result compression

Research for classifying this category of fruits can hardly be seen. But there are some paper those are very similar to this study. Table 1 is demonstrating some correlation between the proposed work and some past work on various fruits.

Table 1. Compression among some previous works.

Work	Technology used	Accuracy
Performance Analysis of Support Vector Machine in Defective and Non Defective Mangoes Classification [1]	SVM	87.0%
DeepFruits: A Fruit Detection System Using Deep Neural Networks [30]	Faster Region-based CNN	81.0%
Automatic fruit recognition and counting from multiple images [11]	Bag-of-Words, SVM	74.2%
A Fruit Recognition Technique using Multiple Features and Artificial Neural Network [23]	ANN	89.0%
Convolutional Neural Networks for Detecting Fruit Information Using Machine LearningTechniques	**CNN**	**99.89%**

VI. Conclusions And Future Works

This paper has presented a fruit detection system for image data captured by smart phone using the state-of-the-art detection framework, CNN. The system recognizes four types of fruits: grape, apple, lychee and lemon. The overall performance of the model to detect fruit images is really very well which has met a desired result of accuracy about 99.89%. The applied CNN method is a very powerful method for machine learning approaches that successfully recognizes the fruits of images for the proposed model and the applied algorithm of CNN is successfully performed for image classification and object detection. In future, the existing database will be enhanced with various types of fruit images. Also future work will involves the integration of the proposed system for a variety of fruits to achieve more accurate performance that will be able to build a robotic harvesting system for fruits from orchards.

References

[1] NeerajKumari, DAshutosh Kr. Bhatt, Rakesh Kr. Dwivedi, Rajendra Belwal" ,Performance Analysis of Support Vector Machine in Defective and Non Defective Mangoes Classification", International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-8 Issue-4, April 2019.

[2] Nuske, S.T.; Achar, S.; Bates, T.; Narasimhan, S.G.; Singh, S. Yield Estimation in Vineyards by Visual Grape Detection. In Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '11), San Francisco, CA, USA, 25–30 September 2011.

[3] Nuske, S.; Wilshusen, K.; Achar, S.; Yoder, L.; Narasimhan, S.; Singh, S. Automated visual yield estimation in vineyards. J. Field Robot. 2014, 31, 837–860.

[4] Yamamoto, K.; Guo,W.; Yoshioka, Y.; Ninomiya, S. On plant detection of intact tomato fruits using image analysis and machine learning methods. Sensors 2014, 14, 12191–12206.

[5] Wang, Q.; Nuske, S.T.; Bergerman, M.; Singh, S. Automated Crop Yield Estimation for Apple Orchards. In Proceedings of the 13th Internation Symposium on Experimental Robotics (ISER 2012), Québec City, QC, Canada, 17–22 June 2012.

[6] Bac, C.W.; Hemming, J.; van Henten, E.J. Robust pixel-based classification of obstacles for robotic harvesting of sweet-pepper. Comput. Electron. Agric. 2013, 96, 148–162.

[7] Hung, C.; Nieto, J.; Taylor, Z.; Underwood, J.; Sukkarieh, S. Orchard fruit segmentation using multi-spectral feature learning. In Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Tokyo, Japan, 3–7 November 2013; pp. 5314–5320.

[8] Kapach, K.; Barnea, E.; Mairon, R.; Edan, Y.; Ben-Shahar, O. Computer vision for fruit harvesting robots-state of the art and challenges ahead. Int. J. Comput. Vis. Robot. 2012, 3, 4–34.

[9] Bargoti, S. and Underwood, J., 2017, May. Deep fruit detection in orchards. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on* (pp. 3626–3633). IEEE.

[10] Md. Mijanur Rahman, Mahnuma Rahman Rinty, FouziaRisdin;" Extracting Text Information from Digital Images" International Journal of Scientific & Engineering Research Volume 10, Issue 6, June-2019.

[11] Song, Y.; Glasbey, C.; Horgan, G.; Polder, G.; Dieleman, J.; van der Heijden, G. Automatic fruit recognition and counting from multiple images. Biosyst. Eng. 2014, 118, 203–215.

[12] Simonyan, K.; Zisserman, A. Two-stream convolutional networks for action recognition in videos. In Proceedings of the Advances in Neural Information Processing Systems, Montréal, QC, Canada, 8–13 December 2014; pp. 568–576.

[13] Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Tahoe City, CA, USA, 3–8 December 2012; pp. 1097–1105.

[14] Everingham, M.; Eslami, S.M.A.; van Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. The pascal visual object classes challenge: A retrospective. Int. J. Comput. Vis. 2015, 111, 98–136.

[15] Uijlings, J.R.; van de Sande, K.E.; Gevers, T.; Smeulders, A.W. Selective search for object recognition. Int. J. Comput. Vis. 2013, 104, 154–171.

[16] Zitnick, C.L.; Dollár, P. Edge boxes: Locating object proposals from edges. In Computer Vision–ECCV 2014; Springer: Zurich, Switzerland, 2014; pp. 391–405.

[17] Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. In Proceedings of the Advances in Neural Information Processing Systems, Montréal, QC, Canada, 7–12 December 2015; pp. 91–99.

[18] He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Trans. Pattern Anal. Mach. Intell. 2015, 37, 1904–1916. Sensors 2016, 16, 1222 23 of 23

[19] Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 13–16 December 2015; pp. 1440–1448.

[20] Ngiam, J.; Khosla, A.; Kim, M.; Nam, J.; Lee, H.; Ng, A.Y. Multimodal deep learning. In Proceedings of the 28th international conference on machine learning (ICML-11), Bellevue,WA, USA, 28 June–2 July 2011; pp. 689–696.

[21] Eitel, A.; Springenberg, J.T.; Spinello, L.; Riedmiller, M.; Burgard,W. Multimodal deep learning for robust RGB-D object recognition. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 681–687.

[22] Lenz, I.; Lee, H.; Saxena, A. Deep learning for detecting robotic grasps. Int. J. Robot. Res. 2015, 34, 705–724.

[23] SaswatiNaskar, Tanmay Bhattacharya; "A Fruit Recognition Technique using Multiple Features and Artificial Neural Network", International Journal of Computer Applications (0975 –8887)Volume 116 –No. 20, April 2015.

[24] Kitamura, K., Yamasaki, T., Aizawa, K.: Fruit log by analyzing fruit images. In: Proceedings of the 16th ACM International Conference on Multimedia, pp. 999–1000. ACM,October, 2008.

[25] https://towardsdatascience.com/introduction-to-convolutional-neural-networks-cnn-with-tensorflow-57e2f4837e18.

[26] Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Tahoe City, CA, USA, 3–8 December 2012; pp. 1097–1105.

[27] https://medium.com/pylessons/convolutional-neural-networks-cnn-explained-step-by-step-69137a54e5e7.

[28] https://www.techradar.com/news/what-is-a-neural-network

[29] Everingham, M.; Eslami, S.M.A.; van Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. The pascal visual object classes challenge: A retrospective. Int. J. Comput. Vis. 2015, 111, 98–136.

[30] Inkyu Sa *, Zongyuan Ge, FerasDayoub, Ben Upcroft, Tristan Perez and Chris McCool, "DeepFruits: A Fruit Detection System Using Deep Neural Networks", Science and Engineering Faculty, Queensland University of Technology, Brisbane 4000, Australia;Vis. 2016, 104, 154–171.

Fouzia Risdin,etal. "Convolutional Neural Networks (CNN) for Detecting Fruit Information Using Machine Learning Techniques." *IOSR Journal of Computer Engineering (IOSR-JCE)*, 22.2 (2020), pp. 01-13.