# Mango (Mangifera Indica L.) classification using Convolutional Neural Network and Linear Classifiers

Sapan Naik[1*] and Purva Desai[1]

[1] Babu Madhav Institute of Information Technology, Uka Tarsadia University
Surat 394350, Gujrat, India
Sapan2307@gmail.com

**Abstract**: Identifying fruit quality of a mango is a vital aspect for farmers and consumers, additionally fruit classification is an imperative stage of fruit grading. Automation has been a boon in classification and grading of a mango (Mangifera Indica L.). In this paper, we picked up various categories of mangoes such as Aafush, Kesar, Jamadar, Rajapuri, Totapuri, langdo and Dahseri. This set of mangoes were used for classification process which includes dataset preparation and feature extraction using pre-trained Convolutional Neural Network (CNN) models. Four linear classifiers namely Support Vector Machine (SVM), Logistic Regression (LR), Naïve Bayes (NB) and Random Forest (RF) are used for classification and compared. The paper also addresses techniques and issues of classification of mangoes on the basis of non-destructive method in particular advancement in deep learning and CNN. However, we have also discussed five CNN models namely Inception v3, Xception, ResNet, Dense-Net and MobileNet. Several experiments were carried out through these models and the highest Rank-1 accuracy of 91.43% and lowest 22.86% accuracy was achieved. The model MobileNet is fastest while DenseNet was found to be the slowest. In CNN models, Xception and MobileNet while in linear classifiers SVM and LR performed well.

**Keywords:** Convolutional Neural Network, Feature extractions, Mango classification, Linear Classifier, Support Vector Machine

## 1 Introduction

Agriculture is the essential key sector which has played a vital role in the Indian economy [1,2]. Its production process is divided into three major phases namely cultivation, harvesting and post-harvesting. There is wide scope of automation introduced in the field of agriculture such as sensors, robots, computer and machine vision technology which benefits the post harvesting phase which have processes like cooling, cleaning, sorting, grading and packing. Apparently, an important concept that drives our attention called smart farming [1] or precision agriculture [2] allows automation to be held and carried out at a greater extent for the major agriculture phases.

Our scope for post harvesting phase is limited to automatic non-destructive fruit classification. Non-destructive fruit classification is identified through its parameters such as aroma, color, firmness(strength) and composition, size, shape, texture and defects and maturity [3]. The pre-stage of sorting and grading is fruit classification. Grading and classification is a necessity in the field of agriculture as manual classification and grading method are time-consuming, laborious, less efficient, monotonous as well as inconsistent. In contrast to that, automatic systems provide rapid, hygienic, consistent and objective assessment. In this paper, mango (Mangifera Indica L.) classification is performed on seven different varieties of mango fruit as it benefits us extraordinarily through its high-quality standards and ample of nutrients filled in it. Gujarat is a leading state with the largest area under mango cultivation stretching from Jamadar, Totapuri, Dahseri, Neelam, Langdo, Kesar, Payri, Rajapuri to Alphonso [4].

In computer vision, deep learning and Convolutional Neural Network (CNN) has gained more popularity for image classification task. Image classification is performed by extracting features and train the classifiers using them. Same process with deep learning, reduced errors in image recognition [5]. Team Hinton participated in image classification competition and won, from that period deep learning was enhanced and its influence was observed [6]. More and more work was performed on the initial model of CNN and as a result of that, we now have many modern CNN architecture models such as Inception, ResNet, Xception and MobileNet which results outstandingly [7]. In the below section, we have briefly reviewed the work that is been done for fruits and vegetables classification.

# 2 Related Work

Fruit recognition system in [9] uses CNN where fruit's region is extracted using a selective search algorithm and image entropy. Two methods color (RGB) and Near-Infrared (NIR) are combined using early and late fusion methods. They are used in Faster R-CNN model to detect seven different fruits [8]. The voting mechanism is used for classification with K-means. It was used with CNN for weed identification in [10] where 92.89% accuracy was achieved. Later it was stated that fine-tuning can give better results. For the online prediction of food materials, a fast auto-clean CNN model is proposed in [11]. Adapting learning were used by the model. Seven classes of mixed crops images (oil radish, barley, weed, stump, soil, equipment and unknown) are classified using deep CNN in [12] where VGG-16's modified version is used for implementation and 79% accuracy is achieved.

Multi-class kernel (Linear, Homogeneous Polynomial and Gaussian Radial Basis) support vector machine (kSVM) combined with color histogram, texture and shape features for fruits classification in [13]. Winner-Takes-All SVM, Max-Wins-Voting SVM, and Directed Acyclic Graph SVM are compared where Max-Wins-Voting SVM with Gaussian Radial Basis kernel performs best with 88.2% accuracy and Directed Acyclic Graph SVM is the fastest. Crop and weed plants are discriminated without segmentation in [14] where Random Forest classifier, Markov Random Field and Interpolation methods are used which gives 93.8% average accuracy. Co-occurrence and statistical features are computed from the sub-bands of the wavelet transform in [15]. 86% classification accuracy is achieved on 15 classes of 2635 images of fruits with minimum distance classifier.

### 2.1 Contributions

For classification of mangoes, we have used a dataset of mangoes which were diversely categories. The experiments were performed through different models on the dataset of 2333 images. This dataset was further bifurcated in the following manner: Aafush 137, Dahseri 250, Jamadar 146, Kesar 500, Langdo 300, Rajapuri 500 and Totapuri 500. The different models that were used with CNN for feature extraction are Inception v3, Xception, DenseNet, ResNet50, and MobileNet. Support Vector Machine (SVM), Logistic Regression (LR), Naïve Bayes (NB) and Random Forest (RF) classifier are utilized for training and classification purpose and all the outcomes of the experiment have been compared with available work.

## 3 Proposed Approach

Dataset preparation was carried out in order to classify mangoes of South Gujarat from 7 categories. The dataset was stacked with 2333 images which were captured in daylight from OnePlus 6T covering mangoes' top-view (putting mango beneath the camera) with white paper as background. As CNN needs fixed-size input images, captured images are resized from 2112*4608 pixels to 224*224 and 299*299 pixels according to the CNN model under use.

After completing dataset preparation, the CNN models need to get implemented with its data. CNN model can be implemented in four different ways. We have portrayed each way of implementation below.

1. The foremost way to implement CNN is to use pre-trained CNN model where the weights are derived from image dataset such as ImageNet. It has numerous advantages including trainings performs faster, classification results can be achieved without high performing infrastructure. The dataset is directly used and transfer learning technique is applied due to which only final layer of the model is trained again for new image dataset.
2. The other method to implement CNN model is to use it as feature extractor. This method imparts that the features that are extracted from the dataset using CNN are given to linear classifier for input and this way the classifier predicts the images.
3. The third approach includes some advantage and disadvantage of the method. The process of third approach is full CNN architecture model is trained from the beginning using a new dataset. The advantage of this is that great accuracy is benefited from the model when compared to transfer learning techniques. However, drawback of this method is it requires high performance computing resources and more time to perform training and testing.

4. The last method to implement CNN model is, everyone can create their own CNN architecture and use it for training and testing purpose. This way it will have more advantages like the model will be compacted and filtered with limited number of layers as per its requirement, processing time will be faster and it will serve higher accuracy.

In the following section, we have briefly described and discussed technical details of CNN.

### 3.1 Overview of Convolution Neural Network

CNN comprises of four layers namely convolution, non-linearity, pooling and fully connected [16]. These multiple levels have various training sets which may appear multiple times in the architecture. Input and output of each level are known as feature maps and number of each layers depend on the structure use.

*Convolution Layer*
The first layer of CNN comprises of Convolutional Layer to which the input is an input image. A filter (neuron/kernel) slides through full image by moving to the right by 1 pixel or unit, and repeat it. When the filter is found to be sliding (called convolving), it is multiplying the values in the filter with the original value set of the image. Herein, the multiplication is termed as computing element wise multiplication. All these moves produce one new number. Once sliding gets over, we will get a reduced two-dimensional array of numbers and this two-dimensional array, produced after convolving is called activation map or feature map [17].

The output of Convolutional layer is activation map where it illustrates the parts of the image; where there are most of the features available. More number of filters aims to give greater depth to the activation map; and this means more information about the input volume we can achieve [18]. More details about filters and their visualization can be found in [19].

*Non-linearity Layer*
Non-linearity layers have diverse activation functions applied to it such as Rectified Linear Units (Relu), tanh and sigmoid. The most preferable function of non-linearity is Relu as because of it the training process gets faster.

*Pooling Layer*
After Convolutional Layer, pooling layer is demonstrated in order to decrease the spatial size which are concluded in terms of weight and height excluding the depth. The main benefit is that as the number of parameters gets reduced eventually computation is also reduced and simultaneously overfitting is prevented due to a smaller number of parameters. Pooling layers has different forms and one of them is max pooling. It is been portrayed with the filter size of 2*2 and a stride of 2 and this normally reduces the input image by half size. Output of the pooling layer goes into a flattening step and this

step is the input to an artificial neural network. Through the flattening step the feature map is converted into a single vector.

*Fully Connected Layer*
Fully connected layer comprises of a fully connected neural network in which there is a working mechanism of input and output. The input to the layer is received through each neurons of the layer from the previous layer's neurons. Each neuron in the previous layer of fully connected neural network connects to a single neuron to give a prominent output. The output is formatted through a matrix multiplication followed by a bias offset.

This is stated as the basic working of CNN. CNN layer is tuned which is important aspect of the process. In the tuning process certain parameters are determined such as choosing architecture, deciding number of layers, weighing parameters, implementation platform and so on.

### 3.2 Tuning the CNN Model

CNN uses a model which needs to be tuned. Following are the three phases of tuning a CNN model:
1. Training the model
2. Validate the model
3. Test the model

In the former training phase, a network is to be prepared for the classification process. After the preparation for the process, the validation phase provides calibration for the network to correct its classification process. When the model is validated, it needs to be tested for the final outcome to achieve the best possible value of all parameters, hence the CNN model is sent to the testing phase.

### 3.3 CNN architecture models

Architectural models are evaluated based on their weight, number of layers and parameters and ImageNet errors. There is plethora of architecture models available for implementing CNN, some of them are VGGNet, Inception, Xception, RestNet50, DenseNet, MobileNet and more. The technical details of all these models are available in [20].

## 4 Results and Discussion

Experiments are performed on the MacBook Pro (13-inch, mid-2012) machine which has 2.5GHz Intel Core i5 processor, 10GB 1333 MHz DDR3 memory and Intel HD Graphics 4000 1536MB graphics card running on macOS High Sierra (version 10.13.6) and Keras and TensorFlow libraries are used for the implementation process. Implementation is performed with the help of following steps:

Step 1. Training dataset is prepared with mango images and with respective labels. Total 2333 images are collected which from 7 categories which bifurcate as Aafush 137, Dahseri 250, Jamadar 146, Kesar 500, Langdo 300, Rajapuri 500 and Totapuri 500.

Step 2. Number of images in all categories are not same which can lead to model overfitting. To resolve this issue, data augmentation technique is used (using ImageDataGenerator method of TensorFlow). We have used rotation, width shift, height shift, shear range, horizontal flip and zoom range for generating new images. Data augmentation method is only applied to Aafush, Dahseri, Jamadar and Langdo category of mangos. We have made these categories count to 500. So, after data augmentation we have 500 images of each 7 categories of mango which makes our dataset size to 3500 images.

Step 3. Images are resized to 299x299 for Inception v3 model. The two beneficial reasons for resizing are 1. CNN needs fixed-size input images 2. To reduce computational time. For initial parameter tuning Inception v3 model is chosen. Out of 3500 images, 80% (2800) images are taken for training while 20% (700) are taken for validation. We have received 92% of validation accuracy for this initial experiment. Misclassification occurs in 5 categories. Below table summarizes the misclassification during this experiment.

**Table 1.** Misclassification by Inception v3 model

| Jamadar | Dahseri | Kesar | Rajapuri | Totapuri | Total |
|---|---|---|---|---|---|
| 11 | 10 | 7 | 5 | 23 | 56 |

Step 4. We are going to use CNN as feature extractor. From the observations of initial experiment, we have taken 3500 training images for all CNN models. For testing purpose total 70 images are selected (10 of each category). Images are resized to 224x224 and 299x299 based on CNN model used. Parameters are set in configuration file. Inception v3, Xception, MobileNet, ResNet and DenseNet architecture models of CNN are selected. For experiment epoch value is set to 1000, learning rate to 0.01, training batch size to 100 and validation percentage to 10.

Step 5. Features are extracted from final fully connected layers of pre-trained CNN model and get stored as HDF5 format locally [5].

Step 6. Linear classifiers i.e SVM, LR, NB and RF are trained for extracted features and labels of step 3.

Step 7. Make the learning weight of linear classifiers intact, they are validated and tested.

As we have taken CNN models for extracting features training time varies based on model under use. Below table summarizes the training time (feature extraction time) for each model for 3500 images. Table 2 also summarizes time required by linear classifier for training. Table 3 summarizes overall result of all architecture models.

**Table 2.** Time required for feature extraction and classification

| CNN Model | Feature extraction time (CNN training) (Minutes) | Classifier training time (Minutes) | | | |
|---|---|---|---|---|---|
| | | SVM | LR | NB | RF |
| **Inception v3** | 84.89 | 3.51 | 1.58 | 0.35 | 0.23 |
| **Xception** | 90.68 | 0.40 | 0.46 | 0.17 | 0.17 |
| **DenseNet** | 134.72 | 2.31 | 1.32 | 0.21 | 0.19 |
| **ResNet** | 73.64 | 0.30 | 0.20 | 0.16 | 0.16 |
| **MobileNet** | 59.81 | 0.20 | 0.30 | 0.19 | 0.17 |

**Table 3.** Rank-1 accuracy

| Model Name | SVM | LR | NB | RF |
|---|---|---|---|---|
| **Inception v3** | 90 | 88.57 | 78.57 | 80 |
| **Xception** | 91.43 | 91.43 | 80 | 78.57 |
| **DenseNet** | 88.57 | 91.43 | 71.43 | 74.28 |
| **ResNet** | 37.14 | 22.86 | 22.86 | 28.57 |
| **MobileNet** | 91.43 | 88.57 | 78.57 | 81.43 |

The above facts and figures illustrate that through experiments performed by various models which includes pair of (Xception, SVM), (Xception, LR), (DenseNet, LR) and (MobileNet, SVM) has achieved highest accuracy of 91.43%. However, ResNet performed low accuracy. From the above experiment, we have concluded that maximum time is required by DenseNet and minimum time is required by MobileNet for training purpose. For feature extraction time and training, Xception and MobileNet CNN models performed well. In classifier, SVM and LR performed better and major misclassification occurred with Totapuri mango. While other misclassifications occurred in Dasheri, Kesar and Rajapuri categories of mango.

As the whole dataset is prepared by us, we did not compare the results of classification with others work and the main reason for not comparing was that we did not come across the same mango category. Based on feedforward neural network and fitness-scaled chaotic artificial bee colony algorithm, hybrid method is proposed in [21] where 1653 images of 18 fruit categories are considered for classification. Color histogram, Unser's texture and Eight Morphology based Shape Measures are used for feature extraction. principal component analysis is used for reducing number of features. We have compared our results with their work in Table 4.

**Table 4.** Comparisons of proposed method with different algorithms

| Algorithm | Classification accuracy (%) |
|---|---|
| GA–FNN | 84.8 |
| PSO–FNN | 87.9 |
| ABC–FNN | 85.4 |
| kSVM | 88.2 |
| FSCABC–FNN | 89.1 |
| Deep Learning - CNN | 91.43 |

## 5 Conclusion and Future Directions

Local features can be implemented and incorporated to increase classification rate. Proposed methods can be made generalize for fruits of south Gujarat, India. Fine tuning of parameters and combining machine learning methods with CNN can improve the results accuracy. Work can be done to decrease time required to classify a mango. Based on classification, grading and detection of skin disease can be identified. From this experiment we have concluded that, even CNN is used as feature extractor and our linear classifier are used for classification purpose. This method allows us to combine features of CNN with handcrafted features and applying different classifiers based on application and dataset in hand.

**Acknowledgments**

The authors acknowledge the help of Mr. Yash Rana in implementation.

**References**

1. "Smart farming means efficient agriculture[AGRI PRESS BENELUX]." [Online]. Available: http://www.agripressworld.com/start/artikel/458796/en. [Accessed: 28-Jan-2019].
2. McBratney, A., Whelan, B., Ancev, T. and Bouma, J., 2005. Future directions of precision agriculture. Precision agriculture, 6(1), pp.7-23.
3. Slaughter, D.C., 2009. Nondestructive maturity assessment methods for mango. University of California, Davis, pp.1-18.
4. Naik, S. and Patel, B., 2017, February. Thermal imaging with fuzzy classifier for maturity and size based non-destructive mango (Mangifera Indica L.) grading. In 2017 International Conference on Emerging Trends & Innovation in ICT (ICEI) (pp. 15-20). IEEE.
5. Ankit Sachan, Tensor flow Tutorial 2: Image Classifier Using Convolutional Neural Network, [Online]. Available. cv-tricks.com/tensorflow-tutorial/training-convolutional-neuralnetwork-for-image-classification/. (Accessed on 17/06/2018).
6. Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25, pp.1097-1105.

7. Gogul Ilango, Using Keras Pre-trained Deep Learning models for your own dataset. [Online]. Available: https://gogul09.github.io/software/flower-recognition-deep-learning. (Accessed on 15.07.2018).
8. Sa, I., Ge, Z., Dayoub, F., Upcroft, B., Perez, T. and McCool, C., 2016. Deepfruits: A fruit detection system using deep neural networks. sensors, 16(8), p.1222.
9. Hou, L., Wu, Q., Sun, Q., Yang, H. and Li, P., 2016, August. Fruit recognition based on convolution neural network. In 2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD) (pp. 18-22). IEEE.
10. Tang, J., Wang, D., Zhang, Z., He, L., Xin, J. and Xu, Y., 2017. Weed identification based on K-means feature learning combined with convolutional neural network. Computers and Electronics in Agriculture, 135, pp.63-70.
11. Chen, H., Xu, J., Xiao, G., Wu, Q. and Zhang, S., 2018. Fast auto-clean CNN model for online prediction of food materials. Journal of Parallel and Distributed Computing, 117, pp.218-227.
12. Mortensen, A.K., Dyrmann, M., Karstoft, H., Jørgensen, R.N. and Gislum, R., 2016. Semantic segmentation of mixed crops using deep convolutional neural network. In CIGR-AgEng Conference, 26-29 June 2016, Aarhus, Denmark. Abstracts and Full papers (pp. 1-6). Organising Committee, CIGR 2016.
13. Zhang, Y. and Wu, L., 2012. Classification of fruits using computer vision and a multiclass support vector machine. sensors, 12(9), pp.12489-12505.
14. Haug, S., Michaels, A., Biber, P. and Ostermann, J., 2014, March. Plant classification system for crop/weed discrimination without segmentation. In IEEE winter conference on applications of computer vision (pp. 1142-1149). IEEE.
15. Arivazhagan, S., Shebiah, R.N., Nidhyanandhan, S.S. and Ganesan, L., 2010. Fruit recognition using color and texture features. Journal of Emerging Trends in Computing and Information Sciences, 1(2), pp.90-94.
16. Bhandare, A., Bhide, M., Gokhale, P. and Chandavarkar, R., 2016. Applications of convolutional neural networks. International Journal of Computer Science and Information Technologies, 7(5), pp.2206-2215.
17. Nielsen, M.A., 2015. Neural networks and deep learning (Vol. 25). San Francisco, CA: Determination press.
18. "A Beginner's Guide To Understanding Convolutional Neural Networks – Adit Deshpande – CS Undergrad at UCLA ('19)." [Online]. Available: https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/. [Accessed: 28-Jan-2019].
19. Zeiler, M.D. and Fergus, R., 2014, September. Visualizing and understanding convolutional networks. In European conference on computer vision (pp. 818-833). Springer, Cham.
20. Naik, S. and Shah, H., 2021. Classification of Leaves Using Convolutional Neural Network and Logistic Regression. In ICT Systems and Sustainability (pp. 63-75). Springer, Singapore.
21. Zhang, Y., Wang, S., Ji, G. and Phillips, P., 2014. Fruit classification using computer vision and feedforward neural network. Journal of Food Engineering, 143, pp.167-177.