Mango Classification using Convolutional Neural Networks

Rohan Sriram[1], Amar Tejas M[2], Prof. J. Girija[3]

[1,2]Department of Computer Science and Engineering, Bangalore Institute of Technology, Bengaluru, India
[3]Associate Professor, Department of Computer Science and Engineering, Bangalore Institute of Technology, Bengaluru, India

---

Abstract - *This paper facilitates the development of an automated system for classification of mango images. Automatic identification and recognition of Mango species is necessary in the Indian market. In recent years, mango species recognition is carried out based on the shape, geometry and texture. While modern search engines provide methods to visually search for a query image that contains a mango class, it lacks in robustness because of the intra-class variation among millions of mango species around the world. Hence in this proposed research work, a Deep learning approach using Convolutional Neural Networks (CNN) is used to recognize mango species with high accuracy.*

Key Words: **ANN, CNN, transfer learning, mango classification**

1. INTRODUCTION

There are about numerous named species of mangoes in the world. Generally, experienced taxonomy experts can identify different species. However, it is difficult to distinguish these mangoes for most people. To know the names or characteristics of the mangoes, we usually consult with specialists, query with mango guidebooks or browse relevant web pages through keywords searching. An effective way to identify name can be done by classifying mango images, especially with the widely use of digital cameras, mobile phone, etc. While the mango classification is appealing in its usefulness and meaningfulness, several restrictions have limited its realization. Unlike other obvious objects classification in which we need to distinguish obvious categories such as car and desk from each other, mango classification is a more difficult task because of inter-class similarity and a large intra-class variation and some are deformed due to weather conditions. What is more, it is tough to distinguish the difference of some kinds that are very different based on aspect ratio and angle. Hence, by using the concept of Convolutional Neural Networks we classify mangoes with highest accuracy.

2. RELATED WORK

Vast amount of work has been done in the past related to mango classification. [1] C.S nandi used, Gaussian Mixture Model (GMM) is used to estimate the parameters of the individual classes for prediction of maturity. Size of the mango is calculated from the binary image of the fruit. Disadvantage being. When one has insufficiently many points per mixture, estimating the covariance matrices becomes difficult.[2] Preprocessing techniques used to obtain binary image. Later, morphological operations on digital images of different mango fruits using MATLAB.

[3] Uses the basic idea of classification of clothing items using convolution neural network. Area has applications in e-commerce websites, social media advertising. [4] Introduces a unified approach that can combine many features and classifiers that requires less training. All features are simply concatenated and fed independently to each classification algorithm. Besides that, the presented technique is amenable to continuous learning, both when refining a learned model and also when adding new classes to be discriminated. [5] we use an exisiting CNN which has been pretrained, retrained using tranfer learning. In order to train and test our system, we use a vast dataset of different classes of mango images captured in real-time.

3. METHDOLOGY

Once the CNN is trained to identify a feature, it is recognized irrespective of its position in the image. In this section, we describe in brief the working of CNN. In the initial step, the image is broken into a series of overlapping smaller image tiles. This is done by passing a sliding window over the image. Each image tile is then fed into a small neural network. Results from processing each tile are saved into a grid which is same as the arrangement of the original image. The resulting matrix contains information regarding the required features. This process is known as convolution as can be seen in Fig. 1. The output array may be large and hence it is down-sampled using the max pooling algorithm which selects the maximum value from each sub-region of a rectangular area and hence helps in reducing dimensionality. Finally, the reduced array is converted into a feature vector, fed into the fully connected neural network. Each feature is voted, the one with the maximum votes and minimum error rate is classified as the particular class of mango.

In practice, CNN is trained on a very large dataset and the trained CNN weights are used either as an initialization or a fixed feature extractor for the task of interest.

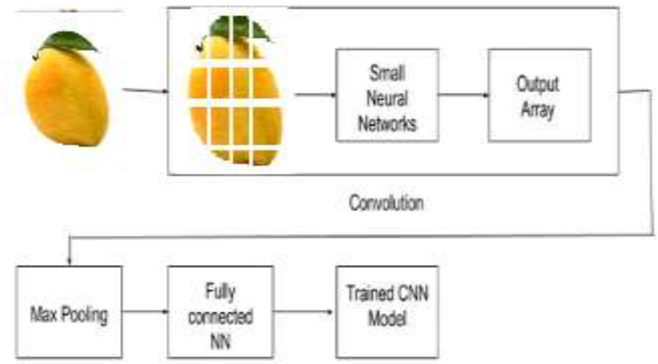

Fig. 1. Block diagram of building the trained model of the CNN

4. IMPLEMENTATION

In this section we discuss the implementation of our system which involves the following steps:

1. Data Collection
2. Training and Testing

Data collection

The data collection phase involved collecting data from the Indian market. . We intend to use this dataset solely for non-commercial, academic and research purposes. This dataset which we use to retrain the GoogLeNet consists of 5093 images ranging over 5 classes and is insufficient to train a network as complex as GoogLeNet from scratch. The GoogLeNet architecture [6] which we use here for mango classification is a 22 layer deep neural network and was initially trained on the ImageNet dataset[7] which consists of more than a one million images ranging over thousand classes. The network was designed with practicality and computational efficiency in mind so that inference could be run on individual devices including even those with limited resources for computational tasks.

Hence, we use the weights from the GoogLeNet which is trained on the ImageNet dataset.

We remove the final layer of the Inception v3 GoogLeNet model which is shown in Fig. and train a new final layer to classify the mango dataset.

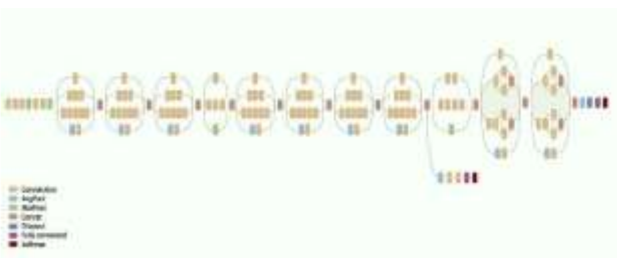

Fig 2 Model architecture of Inception V3 model [8]

Class ID	Class type	Number of training images	Number of testing Images
1	Totapuri	315	265
2	Alphanso	420	270
3	Malgoba	400	285
4	Raspuri	304	295
5	Badami	408	265

Training and testing

Transfer learning is a technique that shortcuts much of this by taking a piece of a model that has already been trained on a related task and reusing it in a new model. In this tutorial, we will reuse the feature extraction capabilities from powerful image classifiers trained on ImageNet and simply train a new classification layer on top.

The first phase analyzes all the images on disk and calculates and caches the bottleneck values for each of them. 'Bottleneck' is an informal term we often use for the layer just before the final output layer that actually does the classification. (TensorFlow Hub calls this an "image feature vector".) This penultimate layer has been trained to output a set of values that's good enough for the classifier to use to distinguish between all the classes it's been asked to recognize. That means it has to be a meaningful and compact summary of the images, since it has to contain enough information for the classifier to make a good choice in a very small set of values.

Once the bottlenecks are complete, the actual training of the top layer of the network begins. The training accuracy shows what percent of the images used in the current training batch were labeled with the correct class. The validation accuracy is the precision on a randomly-selected group of images from a different set.

A new softmax and fully-connected layer is added and trained to identify new classes. For training the model we run 4,000 training steps. Each step selects ten images at random from the training set, finds their bottleneck value from the cache, and feeds them to the final layer of the CNN to get predictions. These predictions are compared against the original labels to find the deviation. The final layer's weights are updated in accordance with the deviation observed through a back-propagation method. This method is used in the Gradient Descent Algorithm where the error rate is calculated at the output and distributed back to the previous layers. The algorithm adjusts the weights of the array. This phase goes under a loop until minimum error rate is fetched.

After all the training steps are complete, we run a test accuracy evaluation on a set of images that are kept separate from the images used for training. This test evaluation provides the best estimate of how the trained model will perform on the task of classification.

5. RESULTS DEPICTING PREDICTION

Fig 3 : Shows the user interface where the test mango images are fed into the model.

Fig 4: Output : Badami confirmed

From Fig. 4 we can see that the test image is a class Badami with an accuracy of 99% as compared to the predictions of the remaining classes.

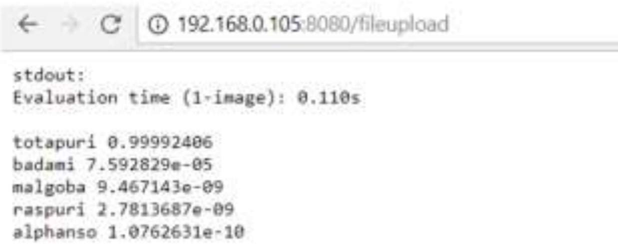

Fig 5: Totapuri confirmed

As seen from Fig.5 we can conclude that the test image is a class of Totapuri with 99% accuracy as compared to the predictions of the remaining classes.

6. CONCLUSION

Considering the benefits offered by the convolutional neural networks to recognize images we try to use them for mango classification. We have implemented a convolutional neural network by retraining final layer of ImageNet for the task of classifying mangoes. Classification of mangoes can be improved by replacing the traditional filters with image filters. We try to bring in metadata free databases, where the classifying algorithm determines the type of mango and its features. We have proposed a system for mango Classification using CNN and implemented the same with the highest accuracy.

7. REFERENCES

[1] C. S. Nandi, B. Tudu, and C. Koley, “A machine vision-based maturity prediction system for sorting of harvested mangoes,” IEEE Trans. Instrum. Meas., vol. 63, no. 7, pp. 1722–1730.

[2] A. Rocha, D. C. Hauagge, J. Wainer, and S. Goldenstein, “Automatic fruit and vegetable classification from images,” Comput. Electron. Agric., vol. 70, no. 1, pp. 96–104, 2010.

[3] A. Gongal, S. Amatya, M. Karkee, Q. Zhang, and K. Lewis, “Sensors and systems for fruit detection and localization: A review,” Comput. Electron. Agric., vol. 116, pp. 8–19, 2015.

[4] T. Meruliya, “Image Processing for Fruit Shape and Texture Feature Extraction - Review,” Int. J. Comput. Appl. (0975, vol. 129, no. 8, pp. 30–33, 2015. [10]. S. Poorani and P. G. Brindha, “Automatic detection of pomegranate fruits using K-mean clustering,” Int. J. Adv. Res. Sci. Eng., vol. 3, no. 8, pp. 198–202, 2014.

[5] “Image database: Mango „Kent,“” 2014. [Online]. Available:http://www.cofilab.com/portfolio/man goesdb/.

[6] Szegedy, Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. "Going deeper with convolutions." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1-9. 2015.

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In CVPR09, 2009.

[8] Szegedy, Christian, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. "Rethinking the inception architecture for computer vision." arXiv preprint arXiv:1512.00567 (2015).