The Pharma Innovation Journal 2023; 12(6): 1188-1192

www.ThePharmaJournal.com

The Pharma Innovation

ISSN (E): 2277-7695
ISSN (P): 2349-8242
NAAS Rating: 5.23
TPI 2023; 12(6): 1188-1192
© 2023 TPI
www.thepharmajournal.com
Received: 22-04-2023
Accepted: 23-05-2023

Sridevy Sridarane
Associate Professor, Department of PS & IT, Tamil Nadu, India

Kumaresan Palaniappan
Research Associate, Water Technology Centre, Tamil Nadu, India

M Nirmala Devi
Assistant Professor, Department of PS & IT, Tamil Nadu, India

M Djanaguiraman
Associate Professor, Department of Crop Physiology, TNAU, Coimbatore, Tamil Nadu, India

Corresponding Author:
Kumaresan Palaniappan
Research Associate, Water technology Centre, Tamil Nadu, India

Artificial intelligence approach for tomato detection and classification in greenhouse using Mobilenet V2

Sridevy Sridarane, Kumaresan Palaniappan, M Nirmala Devi and M Djanaguiraman

Abstract
Tomato is well known fruit since it has many essential and beneficial nutrients like antioxidant, vitamin C and A for human daily diet. Tomato picking by hand is both labor and time consuming. Therefore, to overcome these issues, tomato needs to be picked up automatically with the help of harvesting robot. Recently automation of fruit harvesting gains great popularity. To guide the harvesting robot to pick up the fruit correctly, it is important to correctly detect and find the location of the red mature fruit. A computer vision approach is proposed to detect the fruit by capturing the tomato images and classify. A deep learning classifier is utilized to make a robust decision that covers a wide variety of tomato apperance. Compact deep learning architecture, which is Mobile Net V2 has been fine-tuned to detect three types of tomato. The model is tested on 2000 images which is prepared by us. The results show that MobileNet V2 is able to detect & Classify up to more than 95% accuracy.

Keywords: Tomato fruit detection, harvesting robot, Artificial Intelligence, Mobilenet V2

Introduction
Tomato (*Solanum lycopersicum* L.) is a very important and popular fruit all over the world (Berova and Zlatev, 2000). It can be produced in both tropical and temperature regions. Tomato has many essential and beneficial nutrients like antioxidant, vitamin C and A for humans daily diet, which are derived from tomato fruit for many kinds of products. Due to its special nutritive values, it is one of the most important fruit to human. According to recent FAO statistics, almost 170 million metric tons are produced annually and contributing a billion-dollar industry. In China, tomato is considered as one of the most important fruit, which is grown both in open fields and greenhouse. According to FAO crop update as of year 2016, the total volume of tomatoes processed was around 5.6 million tonnes and it has an increment year by year (Malik *et al.*, 2018). For the purpose of producing tomato fruit for whole year, more and more tomatoes are grown in greenhouses. However, harvesting by labour is very time consuming, costly and does not have good efficiency. In expansion of greenhouse, the main hurdles are high labour cost and also shortage of labour availability, especially in China with the aged tendency of population. To solve these important issues, the guided robots can take place of human work. To get success in reducing the harvest cost and to increase the efficiency, the automatic harvest system by using automatic guided robots can be used in greenhouse (Lili *et al.*, 2017). Therefore in recent years in agriculture field, automation have become a major issue. Many different techniques have been presented for the picking of fruit from trees and plants by developing a harvesting robot. The main technically challenging issue for harvester robot is to recognize and localize fruit on plant and tree. Recognition and localization is mainly used to separate the intact objects to get the area of interest from the background mainly using image processing and machine vision techniques (Kootstra *et al.*, 2021).Tomato fruits usually are not ripening simultaneously. It changes from fully ripened tomato, half ripened tomato, Unripened green tomato on plant. Therefore, the harvesting robot should be able to detect red tomatoes and has ability to harvest or pick up the red mature tomato. In the meantime, because tomato fruit usually grow in bunch, occlusion and connection among fruits makes the single fruit localization difficult (Lawal, 2021). The objective of this project is to develop a new effective method for detecting red mature tomato fruit on plant in greenhouse. And finally we have deployed our model in Android Application.
Deep learning is a specific subfield of machine learning: a new take on learning representations from data that puts an emphasis on learning successive layers of increasingly meaningful representations.

The deep in deep learning isn't a reference to any kind of deeper understanding achieved by the approach; rather, it stands for this idea of successive layers of representations. How many layers contribute to a model of the data is called the depth of the model. Other appropriate names for the field could have been layered representations learning and hierarchical representations learning. Modern deep learning often involves tens or even hundreds of successive layers of representations and they're all learned automatically from exposure to training data. Meanwhile, other approaches to machine learning tend to focus on learning only one or two layers of representations of the data; hence, they're sometimes called shallow learning (Chollet, 2021).

In deep learning, these layered representations are (almost always) learned via models called neural networks, structured in literal layers stacked on top of each other. The term neural network is a reference to neurobiology, but although some of the central concepts in deep learning were developed in part by drawing inspiration from our understanding of the brain, deep-learning models are not models of the brain. There's no evidence that the brain implements anything like the learning mechanisms used in modern deep-learning models. You may come across pop-science articles proclaiming that deep learning works like the brain or was modeled after the brain, but that isn't the case. It would be confusing and counterproductive for newcomers to the field to think of deep learning as being in any way related to neurobiology; you don't need that shroud of "just like our minds" mystique and mystery, and you may as well forget anything you may have read about hypothetical links between deep learning and biology(Aish *et al.*, 2022). For our purposes, deep learning is a mathematical framework for learning representations from data.

Materials and Methods

Dataset acquisition and pre-processing

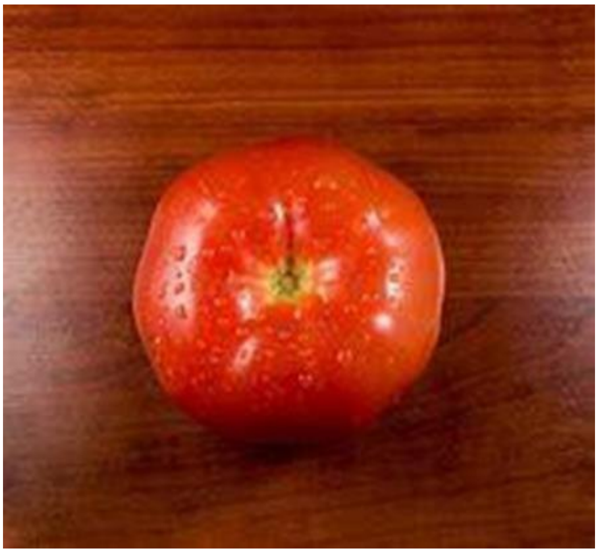

Fig 1: Fully ripened Tomato

The dataset is prepared by us which has three class of tomato i.e. each class divided as fully ripened, un ripened, half ripened. The total size of the dataset used in this work is 2000 images. The training images are of 1000, the validation set contains 500 images belongs to 3 classes, and the test set contains of 500 images which belong to 3 classes. The samples for each class in the dataset are shown .Now the entire dataset of images is reshaped to 224x224x3 and converted into numpy array for faster convolution in case of building the CNN model. Finally, the converted dataset of images is labeled according to each class they belong. Whereas, when training the dataset using transfer learning the image augmentation is applied, validation is done in parallel while training and tested upon the test set.

Fig 2: Half ripened tomato

Fig 3: Unripened tomato

Proposed Tomato Detection and classification Model

In our work, we proposed an accurate CNN model for classifying the tomato which is shown in above figure. It consists of three convolutional layers. The first convolution layer uses 16 convolution filters with a filter size of 3x3, kernel regularizer, and bias regularizer of 0.05. It also uses random_uniform, which is a kernel initializer. It is used to initialize the neural network with some weights and then update them to better values for every iteration. Random_uniform is an initializer that generates tensors with a uniform distribution. Its minimum value is -0.05 and the maximum value of 0.05.

Regularizer is used to add penalties on the layer while optimizing. These penalties are used in the loss function in which the network optimizes. No padding is used so the input and output tensors are of the same shape. The input image size is 224x224x3. Then before giving output tensor to max-pooling layer batch normalization is applied at each convolution layer which ensures that the mean activation is nearer to zero and the activation standard deviation nearer to 1. After normalizing RELU an activation function is used at every convolution. The rectified linear activation function (RELU) is a linear function. It will output the input when the output is positive, else it outputs zero. The output of each convolutional layer given as input to the max-pooling layer with the pool size of 2x2. This layer reduces number the parameters by down-sampling. Thus, it reduces the amount of memory and time required for computation. So, this layer aggregates only the required features for the classification. The finally a dropout of 0.5 is used for faster computation at each convolution.

The 2nd convolution layer uses 16 convolution filters with 5×5 kernel size and the third convolution layer use 16 convolution filters with 7x7 kernel size. Finally, we use a fully connected layer. Here dense layer is used. Before using dense we have to flatten the feature map of the third convolution. In our model, the loss function used is categorical cross-entropy and Adam optimizer with a learning rate of 0.0001. The architecture of the proposed CNN model is shown in figure.

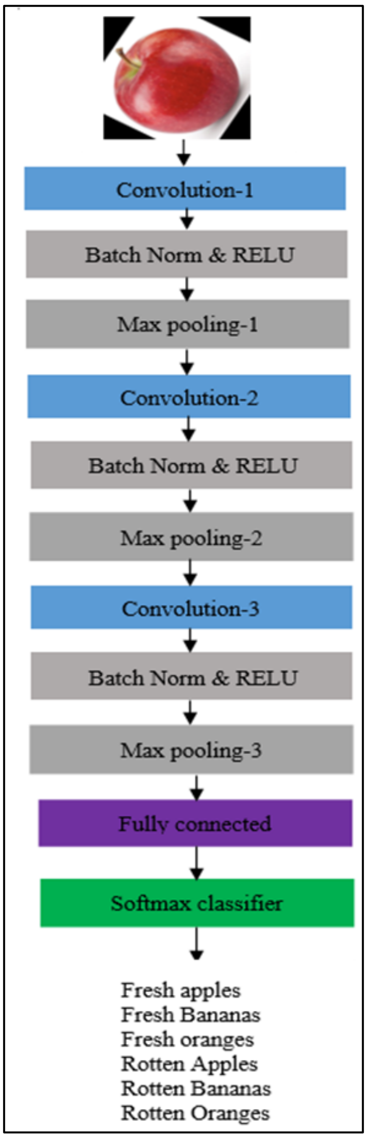

Fig 4: Architecture of Model

Discussion and Results

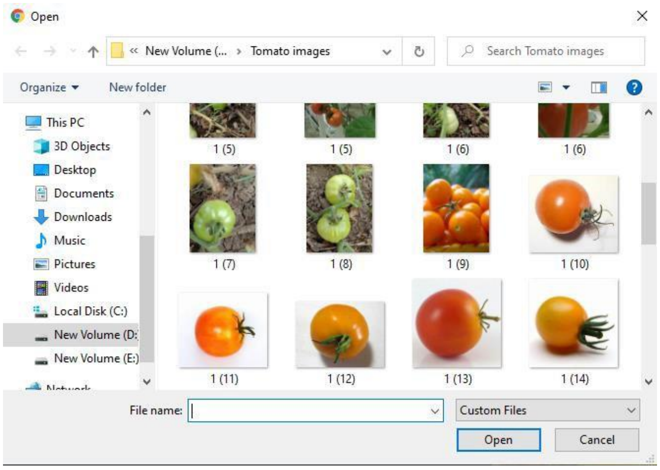

Fig 5: File Explorer will pop up for Uploading Input Image

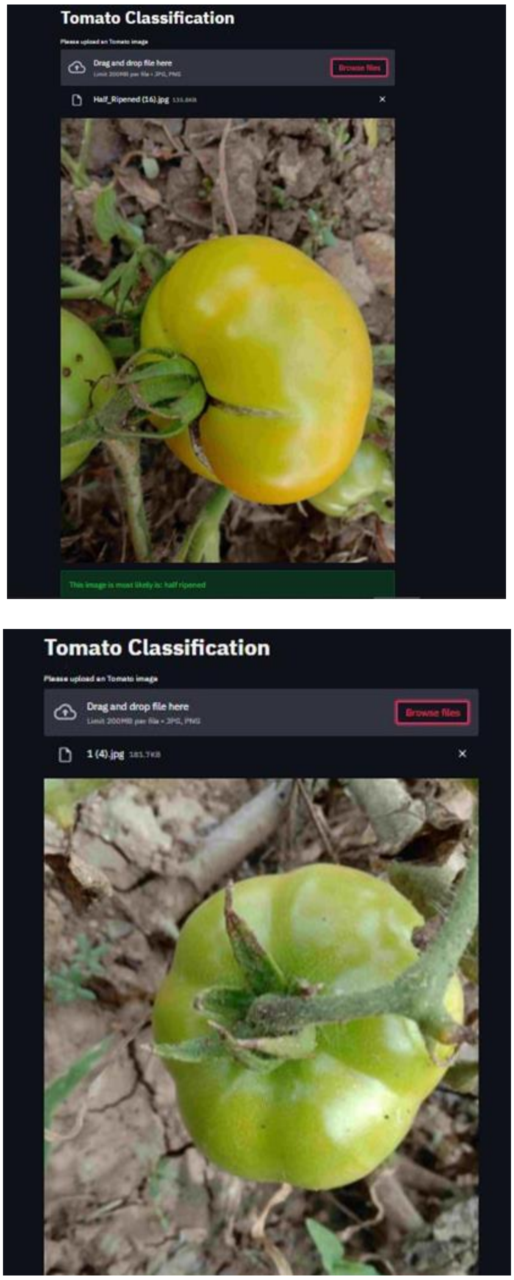

Fig 6: After Uploading, Image will be classified and displayed on the screen

Tensor Flow Lite

TensorFlow Lite is the lightweight version which is specifically designed for the mobile platform and embedded devices (Louis *et al.*, 2019). It provides machine learning solution to mobile with low latency and small binary size. TensorFlow supports a set of core operators which have been tuned for mobile platforms. It also supports custom operations in models. TensorFlow Lite tutorial defines a new file format based on FlatBuffers which is an open source platform serialization library. It consists of a new mobile interpreter which is used to keep apps small and faster. It uses a custom memory allocator for minimal load and execution latency (David *et al.*, 2021).

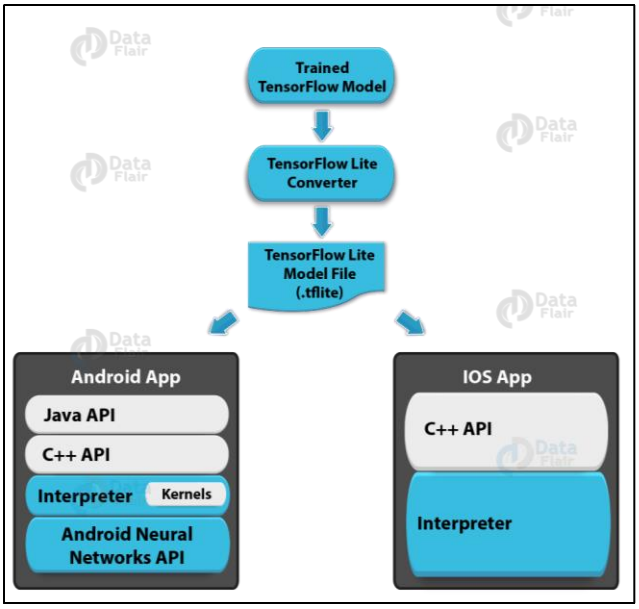

Fig 7: TensorFlow Lite architecture

Images are captured by mobileCamera:

Fig 8: Images are captured by mobile Camera

The above diagram you see is of TensorFlow Lite architecture. The trained TensorFlow model on the disk will convert into TensorFlow Lite file format (.tflite) using the TensorFlow Lite converter. Then we can use that converted file in the mobile application.

For deploying the Lite model file (Rimal *et al.*, 2023) [11]

- Java API: A wrapper around C++ API on Android.
- C++ API: It loads the Lite model and calls the interpreter.
- Interpreter: It executes the model. It uses selective kernel loading which is a unique feature of Lite in Tensorflow.
- You can also implement custom kernels using the C++ API. Some of the highlights of TensorFlow Lite are as follows (Abadi *et al.*, 2016) [1]

It supports a set of core operators which have been tuned for mobile platforms. TensorFlow also supports custom operations in models.

- A new file format based on FlatBuffers.
- On device interpreter which uses selective loading technique.
- When all supported operators are linked TensorFlow Lite is smaller than 300kb.
- Java and C++ API support.

Conclusion

In conclusion, MobileNet V2 has successfully implemented to classify tomatoes. The best classification performance is obtained when MobileNet V2 is trained using Adagrad with a batch size of 16. The experimental results also prove that a learning rate of 0.001 and data division of 4:1 ratio between training and testing deliver the most accurate classification performance.

- Deep learning architecture, which is MobileNet V2 has been fine-tuned to detect the types of tomato. .
- The results show that MobileNet V2 is able to detect & Classify up to more than 95% accuracy.
- And finally we have deployed our model in Android and Webapp.

References

1. Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J. Isard. Tensorflow: A system for large-scale machine learning. Osdi, c2016.
2. Aish MA, Abu-Naser SS, Abu-Jamie TN. Classification of pepper Using Deep Learning. 2022.
3. Berova M, Zlatev Z. Physiological response and yield of paclobutrazol treated tomato plants (*Lycopersicon esculentum* Mill.). Plant Growth Regulation. 2000;30:117-123.
4. Chollet F. Deep learning with Python: Simon and Schuster; c2021.
5. David R, Duke J, Jain A, Janapa Reddi V, Jeffries N, Li J, Kreeger N. Tensorflow lite micro: Embedded machine learning for tinyml systems. Proceedings of Machine Learning and Systems 2021;3:800-811.
6. Kootstra G, Wang X, Blok PM, Hemming J, Van Henten E. Selective harvesting robotics: current research, trends, and future directions. Current Robotics Reports. 2021;2:95-104.
7. Lawal MO. Tomato detection based on modified YOLOv3 framework. Scientific Reports 2021;11(1):1-11.
8. Lili W, Bo Z, Jinwei F, Xiaoan H, Shu W, Yashuo L. Chongfeng. Development of a tomato harvesting robot used in greenhouse. International Journal of Agricultural and Biological Engineering. 2017;10(4):140-149.
9. Louis MS, Azad Z, Delshadtehrani L, Gupta S, Warden P, Reddi VJ. Towards deep learning using tensorflow lite on risc-v." Third Workshop on Computer Architecture Research with RISC-V (CARRV). 2019.
10. Malik MH, Zhang T, Li H, Zhang M, Shabbir S, Saeed A. Mature tomato fruit detection algorithm based on improved HSV and watershed algorithm. IFAC-PapersOnLine. 2018;51(17):431-436.
11. Rimal K, Shah K, Jha A. Advanced multi-class deep learning convolution neural network approach for insect pest classification using TensorFlow." International Journal of Environmental Science and Technology. 2023;20(4):4003-4016.