e-YSIP 2017

Vegetable Identification Using Transfer Learning

Sanket Shanbhag
Supriya Suresh
Saurav, Naveen and Khalid
Duration of Internship: 22/05/2017 – 07/07/2017

2017, e-Yantra Publication

Vegetable Identification Using Transfer Learning

Abstract

This project aims to create a system for automatically logging farm produce from the green house. An image of the produce is captured and this image is fed to a neural network based on the Inception-v3 model provided by Google. This system then identifies the vegetable and classifies it into different classes(vegetables) it has been trained on. Also, a re-training system is implemented which automatically trains any new images captured and thereby further increasing the accuracy of the system over time.

Completion status

The system has been successfully trained for 18 classes (vegetables) achieving accuracy of 99.5% on a separate test set. New vegetables can also be added easily. Also a system has been implemented for auto-training any new images captured on the system every 4 days.

Hardware parts

This project focuses on Machine Learning to identify vegetables, and hardware design is not in the scope of this project. To view the hardware specifications and requirements please refer to this logging project and this project which describes the smart weighing machine.

Software used

- Python 3.5.2
- Tensorflow 1.2.0
- PrettyTensor 0.7.4
- Installation Steps:
  - Open a terminal in the project base folder and type:
    `pip3 install -r requirements.txt`
    (You might need to use sudo for this step.)

Software and Code

The complete code is available here. It is divided into various folders each with a self contained module of the project.

Downloading and formatting data

The folder `download_data` contains python scripts and shell scripts for downloading and formatting data from a list of URLS or from ImageNet. Information on using these scripts can be found in the wiki page. The data collected by us is present on the Eyantra ML Box in the `weighingmachine` user account.

Transfer Learning

The code for transfer learning can be found in the `transfer_on_inception_v3` folder. The `transferveg.py` file contains the code for running the model. We removed the final layer of the Inception-v3 model and add 3 fully connected layers of 4096, 2048, and 1024 nodes with a dropout layer after the 2048-node layer. This helped us achieve an accuracy of 99.5%

Module Integration

The `ghfarm.py` script contained in the `module_integration/raspberrypi` folder will run on the raspberry pi on the weighing machine. It interfaces with the server to predict the crop and sends data once the image is taken.

Auto-Training

The AutoTrain folder contains code for running a server to accept images and scripts for adding cron tasks to run the autotraining code every 4 days.

Steps for adding new crop to system

Refer to the logging project here and add a new crop to the database from the admin panel. Create a folder with the name as short_name as you defined in the database. Put the training images in this folder. Also, create a folder called test inside the folder with the short_name and add a few images for testing. Repeat this process for as many crops as you want to add to the system. All these folders should be placed in a top-level folder called veg and veg should be placed inside the transfer_on_inception_v3 folder. Running the transferveg.py script now will include your newly created crops in the checkpoints folder. Now overwrite the checkpoints folder and the veg.pkl file to the server in the django/ directory.This entire process is also documented here.

Use and Demo

Final Setup Image

Instructions for using the weighing machine can be found here and video demonstration can be found here. For Retraining and training from scratch, please refer to the github wiki.

Future Work

This project focuses mainly on limited number of vegetables mentioned in the wiki. In the future it can be extended to include other vegetables and fruits.

Challenges

1. Lack of proper training data.
2. Every vegetable has different varieties so our second challenge was to find enough training data to encompass all the varieties.

Figure 1.1: Setup

Results

- The system works well while identifying vegetables with a distinct texture and color such as tomato, bittergourd, ladyfinger etc. However, while identifying similar-looking vegetables like fenugreek, basil and coriander, we found that the system tends to make mistakes and/or have low confidence.

- Vegetables were tested on the following test cases:

  1. Variation of light - Dark, less light, normal light, bright day light(like window open).

  2. Variation in Vegetable storage device - zip-lock bag, tray, basket, direct on base-plate of machine.

  3. Variation in ambient environment captured by camera.

It was observed that the system predicted correct results with pretty good accuracy even when exposed to the above mentioned conditions.

- **Results for some Test cases were as follows:**

Figure 1.2: Brinjal under dark ligt condition
Obtained Confidence: 92.08%

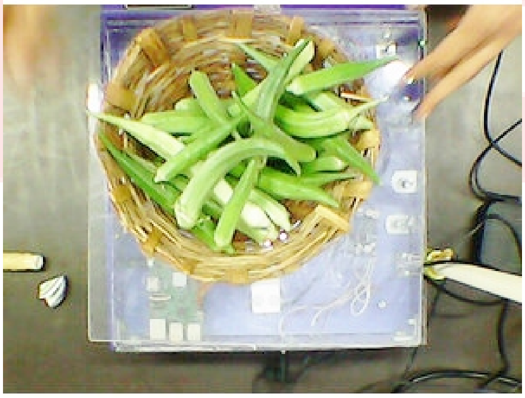

Figure 1.3: Lady finger under bright light condition
Obtained Confidence: 95.93%

Figure 1.4: Cucumber on tray
Obtained Confidence: 99.5%

Figure 1.5: Tomato inside zip-lock
Obtained Confidence: 92.03%

References

1. Inception v3 model by Google.
2. Architecture of Convolutional Neural Networks by Stanford.