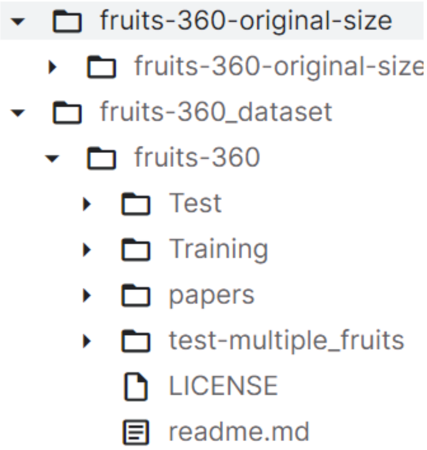
Detection on Fruit and Vegetables in 2D Image

Overview of the project

Abstract

In this final project we developed, it is able to identify different categories of fruits and vegetables and count the total number of each category in one single image, in total of 131 classes of fruit and vegetables from fruits-360 dataset on Kaggle**[1]**.

Warning: There are two dataset from the fruits-360 dataset(shown below), but we are only using the "fruits-360_dataset" folder in this project for training and testing.

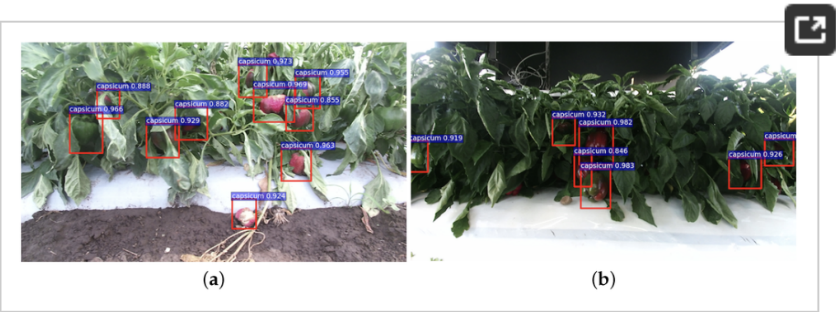


- fruits-360-original-size
  - fruits-360-original-size
- fruits-360_dataset
  - fruits-360
    - Test
    - Training
    - papers
    - test-multiple_fruits
    - LICENSE
    - readme.md

State of the art

DeepFruits: A Fruit Detection System Using Deep Neural Networks[2]

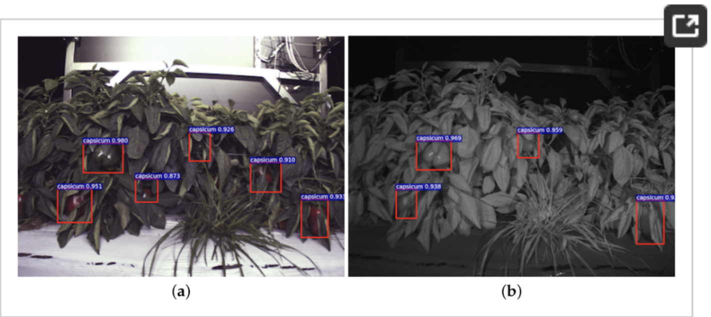
In this research paper we found, it is able to recognize different types of fruit in color images or Near-Infrared images. They also use pixel-wise for the image detection.



(a)

(b)

The image above shows the accuracy of the application identify capsicum in color image



(a)

(b)

The image above shows the accuracy of the application identify capsicum in Near-Infrared image

Input and output of the project

Our target is to recognize different fruits and vegetables in a raw image
Input: One single image that has different types of fruit and vegetables

Output: The predicted classes of fruits and vegetables and total number of each in an image

Summary of the team contributions

Name	Contribution	Percentage
Wenhao He	Proposal, coding part, report, and video recording	50%
Bochun Deng	Proposal, coding part, report, and video recording	50%

Approach

Details of the algorithm

We use a 3-layer CNN to train our model, and for clarification, we design functions that are able to do object detection based on Viola-Jones' concept. We first apply different sizes of kernels iterating through the image to extract features, then feed these features into our trained classifier. For each possible pattern we put in, our classifier will return a list that contains the probability of classes that the pattern could be, and since our train set is 100x100, every possible pattern different from this size will be balanced by a ratio depending on differences.

Aspects of the algorithm have code on own

In this final project, we coded our own on counting different classes of fruit and vegetables in one single image, since the dataset only contains one category of fruit or vegetables in an image only. We coded our own training process for the training model and the OpenCV part of the implementation for images to detect on fruit and vegetables.

Aspects of the algorithm have used from online resources

In order to keep high accuracy and low loss value of the algorithm, we reference the early stopping function from **[3]**, we also reference its convolution neural network tuning parameters for the training on the dataset.

Experimental Protocol

In this final project, we used the fruits-360 dataset from the Kaggle website for the used of the training and testing of the project dataset.

We evaluated the success by since the dataset it only contains one fruit per image, so we putted multiple fruit images into one single image (shown below)

Then we count the number of each category of the fruit in this one single image.
In this example image, we have input fruits and vegetables:

{Apple, Watermelon, Blueberry, Green Pepper, Pear, Potato} in 6 different fruits

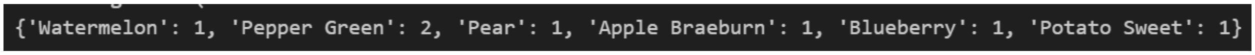
If the output gives us at least **4 out of 6 classes** of fruits correct and count their number of each, we consider our approach to the dataset a success.

For the compute resources we used:

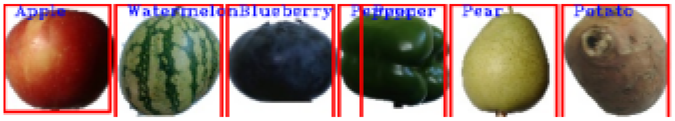
- Pytorch library
- OpenCV library
- Numpy library
- Matplotlib library

Results

We randomly selected images from the fruits-360 dataset we used to train, we are able to count different classes of fruits and vegetables and each of the classes with a very high prediction accuracy. Here is one of the example output we got:

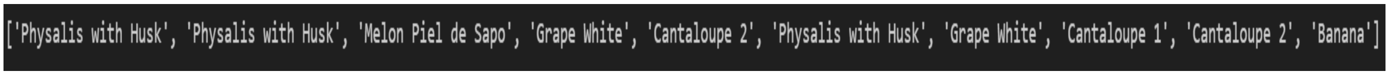


```
{'Watermelon': 1, 'Pepper Green': 2, 'Pear': 1, 'Apple Braeburn': 1, 'Blueberry': 1, 'Potato Sweet': 1}
```

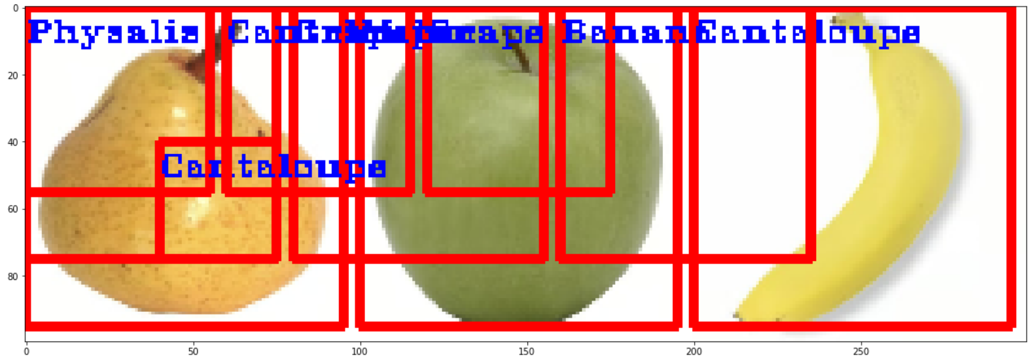


We also tested out images from online resource outside of the fruit 360 dataset**[4]**, here is one of the example output we got out of the top 10 prediction from the trained model:

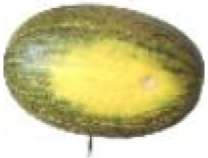
Image Output



```
['Physalis with Husk', 'Physalis with Husk', 'Melon Piel de Sapo', 'Grape White', 'Cantaloupe 2', 'Physalis with Husk', 'Grape White', 'Cantaloupe 1', 'Cantaloupe 2', 'Banana']
```



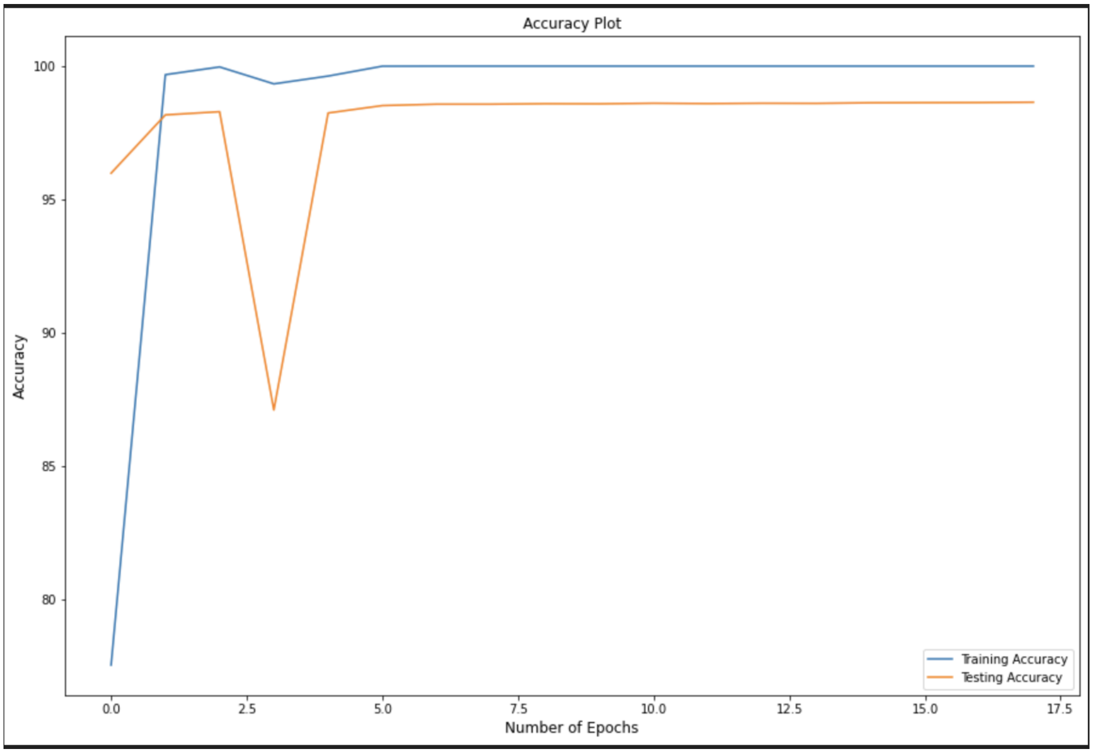
Here are the classes in the dataset for the result we got above:

‘Physalis with Husk’: , ‘Melón Piel de Sapo’:  , ‘Grape White’: , ‘Cantaloupe 2’: , ‘Cantaloupe 1’: , ‘Banana’:

As we can see, the predicted classes are really similar to the fruit and vegetables we downloaded from online resources, even though they are different classes of the fruit and vegetables. Even our humans could miss recognizing these fruits. Consider this dataset filmed those images from years ago, earliest to 02/25/2017 and the latest to 09/22/2019. We think that this dataset is not really suitable nowadays. Therefore, a clear, better resolution of image is a must for us to train the model to fit for the online resource and future application to use in our real world.

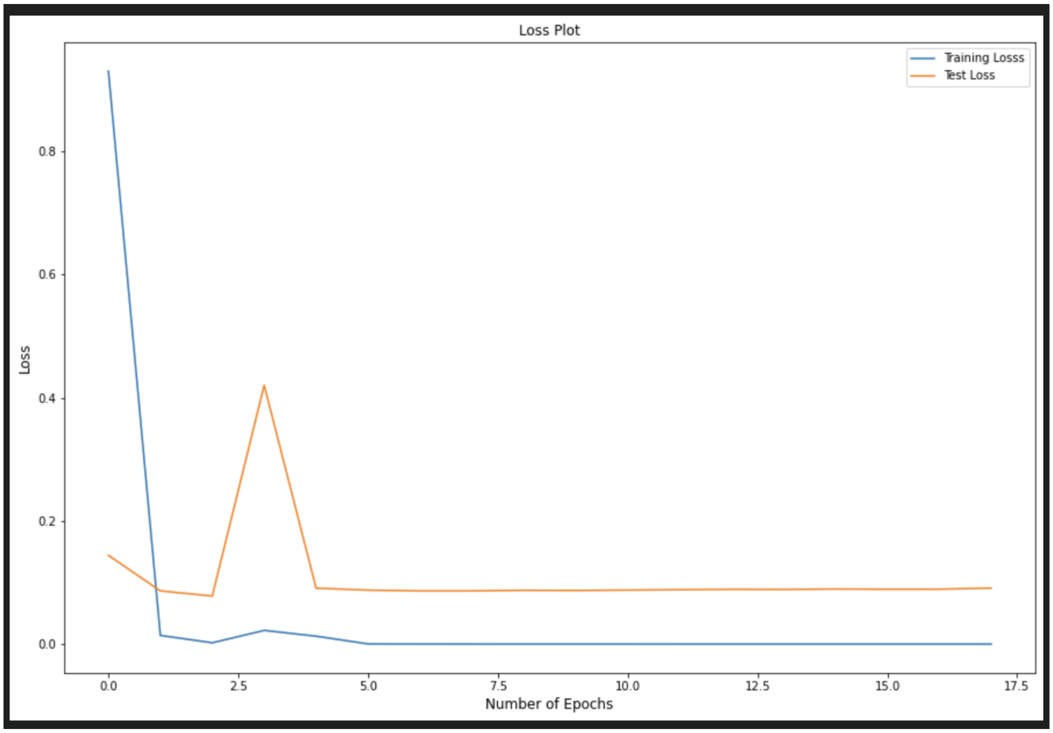
Analysis

Training & Testing Accuracy Plot



From the accuracy diagram above, we can see that the training and testing accuracy have very high scores for prediction, training accuracy always stays a very high value. Even though testing accuracy has dropped below 90% of the accuracy, it jumps back afterwards. Which we consider as a good prediction result for the dataset.

Training & Testing Loss Plot



From the loss diagram above, we can see that the training and testing loss have low value for prediction, training loss always stays at the bottom line, approximately closer to 0. Even though testing loss has a value that jumps up to 0.4 for the highest loss value, it jumps back afterwards. Which we consider as a good result for the dataset on loss.

Discussion and lessons learned

1) From this project, we learned how to use convolution neural networks to train datasets from the internet, learned how to use OpenCV knowledge from class to form images to read multiple classes of fruits and vegetables in the project.

2) From this final project we developed, it could be a big potential application for deeper development on the image processing part, it could be very beneficial for people who work in the orchard. With our application, they can take the picture from the whole orchard to help them recognize and count the total number of different categories of fruit and vegetables in torchard, instead of taking the picture one by one.

Bibliography

[1] Fruits-360 dataset on Kaggle: https://www.kaggle.com/datasets/moltean/fruits

[2] DeepFruits: A Fruit Detection System Using Deep Neural Networks: https://www.mdpi.com/1424-8220/16/8/1222

[3] Convolution Neural Network Structure: https://www.kaggle.com/code/darkwyvern/fruits360-pytorch-cnn/notebook

[4] Wegmans Fruit Images: https://shop.wegmans.com/shop/categories/75

[5] Pytorch Training a classifier: https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html

[6] Define a Neural Network in Pytorch: https://pytorch.org/tutorials/recipes/recipes/defining_a_neural_network.html

[7] Pytorch Sequential Container: https://pytorch.org/docs/stable/generated/torch.nn.Sequential.html