

CropCalories

CSE 473/573: Computer Vision and Image Processing, Spring 2023 Semester Project

1. Project Overview

CropCalories project aimed at predicting the calorie content of fruits and vegetables using image segmentation and machine learning algorithms. The project's main goal is to provide a tool that can be used for diet and nutrition tracking, meal planning, fitness training, and agriculture and food science research.

a) *State-of-the-art Methods:*

Hemalatha et.al. [3] implemented a multi-layer perceptron (MLP), convolutional neural network (CNN) for fruits and vegetables classification and compared its performance with support vector machine (SVM), K-nearest neighbors (KNN) and decision trees. NE Mimma et.al. [4] presented the development of automated fruit classification and detection systems using deep learning algorithms. The authors utilized two datasets with ResNet50 and VGG16 models and developed an Android application for real-time fruit classification and detection. Anish et.al. [5] developed an Android app called FruVegy which uses machine learning algorithms to identify and display nutritional values of 40 different fruits and vegetables. They created a dataset of 1600 images and achieved an accuracy of 98.1% using TensorFlow.

b) *Inputs and Outputs of the project:*

The inputs to the project are the images of fruits and vegetables, while the output is the calorie content of the predicted fruit or vegetable in the image.

c) *Summary of Contributions:*

The contributions of this project include the development of a pipeline that can take images of fruits and vegetables as inputs and predict their calorie content using machine learning algorithms. The project also demonstrates the potential use of image segmentation and machine learning in agriculture and food science research.

2. Approach

The project's approach involves the following steps:

1. *Data Collection:* The project starts with collecting the images of fruits and vegetables from a dataset [1].
2. *Feature Extraction using Image Segmentation:* The collected images are then processed using different image segmentation methods, including edge detection, thresholding, and morphological operations, to extract the relevant features and edges from the images.
3. *Machine Learning Algorithms:* The extracted featured images are converted into dataframes and used as inputs to the machine learning algorithms such as support vector classifier, KNN, decision trees, and random forests to train models and predict the calorie content of the fruits and vegetables in the images. Though these non-parametric ML algorithms are slow, they are good at classifying the images, with superior performance.
4. *Prediction using Trained ML model:* Use the trained models to predict the type of fruit or vegetable in the processed image.
5. *Calorie Calculation:* Finally, the calorie content of the predicted fruit or vegetable in the image is fetched from a dataset.

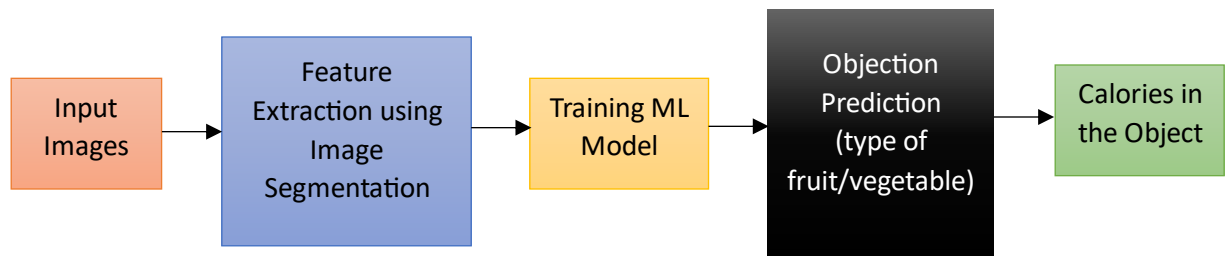


Figure 1: Project pipeline

Coding on my Own:

1. **Image Segmentation:** I have used opencv to implement Laplacian and Canny edge detection. I have written user-defined functions for Laplacian of Gaussina (LoG) and Sobel using opencv functions. Finally, Local Binary Pattern (LBP) feature extraction is implemented from scratch.
2. **ML algorithms:** Scikit-learn library is used to import the classical ML algorithm models.
 - I have created training and test dataframes from the train and test images.
 - Trained and tested the ML models.

Code used from Online Resources:

In addition to fetching the calories using a dataset, I have also found the calories using webscraping with Beautiful Soup. In order to carry out this task, I have referred to the work of Kushal [7]. In his study, the author also focused on classifying fruits and vegetables, but utilizing CNNs for the task.

3. Experimental Protocol

- This project uses two datasets. The first dataset consists of images of fruits and vegetables, which are divided into 36 classes. The dataset contains 3114 images for training and 359 images for testing, with each image scaled to 64x64. The distribution of images for each class is almost balanced in training and testing sets.
- The calorie content of predicted class is fetched using the dataset in [6]. The calories dataset [2] didn't have all the necessary classes of fruits and vegetables and hence, I have added the missing fruits and vegetables calorie content to the dataset in [2] and uploaded the new dataset in Kaggle, it is available at [6].
- To evaluate the success of our models, I have used accuracy as the evaluation metric, which measures the percentage of correctly classified images in the validation set. I also visually inspected the processed images to ensure that the image segmentation methods were accurately extracting the edges and features of the fruits and vegetables.
- The project was implemented on Google Colab and Kaggle kernel using Python programming language and its libraries like scikit-learn, Numpy, Pandas, OpenCV and few other necessary libraries.

4. Results

The qualitative visualization results of edge detection algorithms that were used as input to the machine learning models are presented in Figure 2.

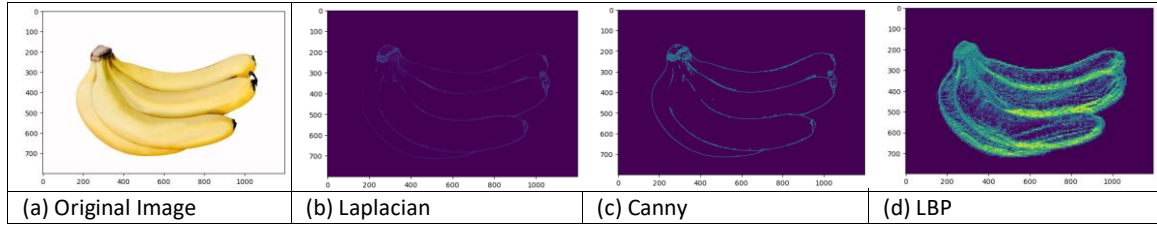


Figure 2: Output images of Edge detection methods

The accuracy of various edge detection algorithms with support vector classification (SVC) are presented in Table 1. Furthermore, a comparison of SVC with other machine learning algorithms, such as KNN, decision trees, and random forest, is provided in Table 2.

Testing of the Model:

Testing was done by selecting a test image from a randomly chosen class. This test image is given as input to the trained model, the predictions of the model are tabulated in Table 1.

Table 1: Summary of Edge detection methods with SVC($C=0.1$, kernel='linear', gamma='auto')

S.No.	Edge detection method	Accuracy in %	Actual class	Predicted class	Calories
1	Laplacian	96.66	Tomato	Tomato	20 cal per 1 tomato (111 g)
2	Canny	96.93	Spinach	Spinach	78 cal per 1 bunch (340 g)
3	Laplacian of Gaussian (LoG)	96.10	Orange	Orange	62 cal per 1 orange (131 g)
4	Sobel	96.38	Corn	Corn	562 cal per 1 cup (154 g)
5	Local Binary Pattern (LBP)	96.66	Carrot	Carrot	25 cal per 1 carrot (61 g)

Table 2: Comparison of different ML algorithms with Canny edge detected featured input

S.No.	Name of the ML algorithm	Accuracy in %
1	SVC	96.93
2	KNN	78.3
3	Decision Trees	95.82
4	Random Forest	96.38

From the given results, it can be concluded that all the tested edge detection methods have achieved high accuracy ranging from 96.10% to 96.93%. This indicates that the image processing techniques used in this project are effective in extracting relevant features from the input images. Furthermore, the predicted classes of the fruits and vegetables closely match their actual classes, indicating that the machine learning models have learned to accurately classify the images based on their features. Additionally, the provided calorie information for each predicted class is helpful in providing further insights and practical applications of the model.

Transfer Learning:

I have used convolutional neural networks and imported pre-trained EfficientNetB4 model for performing image classification but could achieve an accuracy of 96.1% only. Here, the raw images are fed into the

neural network. Further fine-tuning of hyperparameters is required to improve image classification accuracy.

5. Analysis

Based on the results obtained from the experiments, the edge detection algorithms such as Laplacian, Canny, LoG, Sobel, and LBP perform well in detecting edges of the given fruits and vegetables images. The results show that the accuracy of the classification model is high with all the tested edge detection methods.

However, there are limitations to the algorithms used in this project. For instance, Laplacian edge detection may detect a weak edge as a strong edge and vice versa, Canny edge detection may miss some weak edges, hence the accuracy could only reach up to 97% only. Moreover, the performance of edge detection algorithms may vary with different levels of difficulty of input images. Therefore, it is important to carefully choose the edge detection algorithm based on the nature of the input image to ensure optimal performance.

6. Discussion and Lessons Learned

In this project, I learned about different edge detection methods and their implementation in image processing. Additionally, I gained experience in using machine learning algorithms for image classification tasks. These skills will undoubtedly be beneficial in future projects involving image analysis and machine learning.

To extend this project in the future, the following steps could be taken:

- Use real-world images of rotten fruits and vegetables to improve the model's robustness.
- Utilize transfer learning to import advanced pre-trained convolutional neural network models to enhance accuracy.
- Build a mobile app that can be used to capture an image and receive an instant calorie count for the fruit or vegetable.
- Calculate calories more precisely by estimating the size or weight of the fruit or vegetable based on the input image.

Bibliography

1. Kritik Seth, [Fruits and Vegetables Image Recognition Dataset](#), Kaggle. (Last accessed on 05-07-2023)
2. Vaishnavi, [Food and their Calories](#), Kaggle. (Last accessed on 05-07-2023)
3. Hemalatha, N., P. Sukhetha, and Raji Sukumar. "[Classification of Fruits and Vegetables using Machine and Deep Learning Approach](#)." *2022 International Conference on Trends in Quantum Computing and Emerging Business Technologies (TQCEBT)*. IEEE, 2022.
4. Mimma, Nur-E., et al. "[Fruits Classification and Detection Application Using Deep Learning](#)." *Scientific Programming* 2022 (2022).
5. Appadoo, Anish, Yashna Gopaul, and Sameerchand Pudaruth. "[FruVegy: An Android App for the Automatic Identification of Fruits and Vegetables using Computer Vision and Machine Learning](#)." *International Journal of Computing and Digital Systems* 13.1 (2023): 169-178.
6. Madhusikha, [Food and Calories](#), Kaggle. (Last accessed on 05-07-2023)
7. Kushal Bhavsar, [Fruit and Vegetable Recognition](#), GitHub. (Last accessed on 05-07-2023)