FRUIT AND VEGETABLE RECOGNITION USING KNN CLASSIFIER

PATTERN RECOGNITION FINAL PROJECT

STUDENT NAME : EBUBEKİR ÇAĞRI ŞEN

INSTRUCTOR NAME : PROF. DR. CEM ÜNSALAN

İçindekiler

1. Overview

Recognition of fruits and vegetables is very popular challenge in pattern recognition. There are many ways to realize it. For this project, I have implemented specific methods to extract color and texture features from image. I have used a dataset that contains 2612 image for 15 different fruits and vegetables. Download link for the dataset is (http://www.ic.unicamp.br/~rocha/pub/downloads/tropical-fruits-DB-1024x768.tar.gz

2.) Features

For feature extraction, i used Color Coherence Vector and Local Binary Pattern.

2.1.) Color Coherence Vector

Color histograms are used to compare images in many applications. Their advantages are efficiency, and insensitivity to small changes in camera viewpoint. However, color histograms lack spatial information, so images with very different appearances can have similar histograms. For exampie, a picture of fall foliage might contain a large number of scattered red pixels; this could have a similar color histogram to a picture with a single large red object. We describe a histogram-based method for comparing images that incorporates spatial information. We classify each pixel in a given color bucket as either coherent or incoherent, based on whether or not it is part of a large similarly-colored region. A color coherence vector (CCV) stores the number of coherent versus incoherent pixels with each color. By separating coherent pixels from incoherent pixels, CCV'S provide finer distinctions than color histograms.

This method gives a 2 feature called alpha and beta. We concenate 2 feature vector as one. Histogram size for this method selected as 64 bin.

The code written for this feature is in ccv.py file.

2.2) Local Binary Pattern

Local binary patterns (LBP) is a type of visual descriptor used for classification in computer vision. LBP is the particular case of the Texture Spectrum model proposed in 1990.[1][2] LBP was first described in 1994.[3][4] It has since been found to be a powerful feature for texture classification; it has further been determined that when LBP is combined with the Histogram of oriented gradients (HOG) descriptor, it improves the detection performance considerably on some datasets.[5] A comparison of several improvements of the original LBP in the field of background subtraction was made in 2015 by Silva et al.[6] A full survey of the different versions of LBP can be found in Bouwmans et al.[7]

This method gives us a matrix of the size as the image file. We get its histogram to use it as feature. We also selected its bin size 64.

Therefore, we get a 128 dimesion feature vector to use in kNN classifier.

The code written for this feature is in lbp.py file.

3.) Classifier

In pattern recognition, the k-nearest neighbors algorithm (k-NN) is a non-parametric method used for classification and regression.[1] In both cases, the input consists of the k closest training examples in the feature space. The output depends on whether k-NN is used for classification or regression:

In k-NN classification, the output is a class membership. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of that single nearest neighbor. But in our project i have used k = 6, with this setup we gain more accuracy from classification.

In k-NN regression, the output is the property value for the object. This value is the average of the values of its k nearest neighbors.

k-NN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until classification. The k-NN algorithm is among the simplest of all machine learning algorithms.

4.) Performance

We labelled every class from 1 to 15. I split the dataset into 2 file for training and dataset to control the program. We got average % 75.9 accuracy. Here below we have the output for every classes and general result.

```
dataset/agata_potato          good:71 bad:29 ratio: % 71.00
dataset/asterix_potato        good:66 bad:24 ratio: % 73.33
dataset/cashew                good:99 bad:5 ratio: % 95.19
dataset/diamond_peach         good:67 bad:37 ratio: % 64.42
dataset/fuji_apple            good:74 bad:30 ratio: % 71.15
dataset/granny_smith_apple    good:65 bad:11 ratio: % 85.53
dataset/honneydew_melon       good:57 bad:15 ratio: % 79.17
dataset/kiwi                  good:60 bad:24 ratio: % 71.43
dataset/nectarine             good:84 bad:39 ratio: % 68.29
dataset/onion                 good:23 bad:13 ratio: % 63.89
dataset/orange                good:40 bad:11 ratio: % 78.43
dataset/plum                  good:104 bad:28 ratio: % 78.79
dataset/spanish_pear          good:44 bad:35 ratio: % 55.70
dataset/taiti_lime            good:50 bad:2 ratio: % 96.15
dataset/watermelon            good:85 bad:11 ratio: % 88.54
                              global good:989 global bad:314
                               ---------------------------------
                              global success: % 75.90
```

5.) References and repo link

The code and all files can be reached at https://github.com/sencagri/fruitrecognizer

1. DC. He and L. Wang (1990), "Texture Unit, Texture Spectrum, And Texture Analysis", Geoscience and Remote Sensing, IEEE Transactions on, vol. 28, pp. 509 - 512.
2. ^ L. Wang and DC. He (1990), "Texture Classification Using Texture Spectrum", Pattern Recognition, Vol. 23, No. 8, pp. 905 - 910.
3. ^ T. Ojala, M. Pietikäinen, and D. Harwood (1994), "Performance evaluation of texture measures with classification based on Kullback discrimination of distributions", Proceedings of the 12th IAPR International Conference on Pattern Recognition (ICPR 1994), vol. 1, pp. 582 - 585.
4. ^ T. Ojala, M. Pietikäinen, and D. Harwood (1996), "A Comparative Study of Texture Measures with Classification Based on Feature Distributions", Pattern Recognition, vol. 29, pp. 51-59.
5. ^ "An HOG-LBP Human Detector with Partial Occlusion Handling", Xiaoyu Wang, Tony X. Han, Shuicheng Yan, ICCV 2009
6. ^ C. Silva, T. Bouwmans, C. Frelicot, "An eXtended Center-Symmetric Local Binary Pattern for Background Modeling and Subtraction in Videos", VISAPP 2015, Berlin, Germany, March 2015.
7. ^ T. Bouwmans, C. Silva, C. Marghes, M. Zitouni, H. Bhaskar, C. Frelicot,, "On the Role and the Importance of Features for Background Modeling and Foreground Detection", https://arxiv.org/pdf/1611.09099v1.pdf