Journal of Food Engineering 143 (2014) 167–177

Contents lists available at ScienceDirect

Journal of Food Engineering

journal homepage: www.elsevier.com/locate/jfoodeng

Fruit classification using computer vision and feedforward neural network

Yudong Zhang [a,*], Shuihua Wang [a], Genlin Ji [a], Preetha Phillips [b]

[a] School of Computer Science and Technology, Nanjing Normal University, Nanjing, Jiangsu 210023, China
[b] School of Natural Sciences and Mathematics, Shepherd University, Shepherdstown, WV 25443, USA

ARTICLE INFO

Article history:
Received 18 October 2013
Received in revised form 30 June 2014
Accepted 1 July 2014
Available online 9 July 2014

Keywords:
Color histogram
Unser's texture analysis
Shape feature
Feedforward neural network
Fitness-scaled chaotic artificial bee colony

ABSTRACT

Fruit classification is a difficult challenge due to the numerous types of fruits. In order to recognize fruits more accurately, we proposed a hybrid classification method based on fitness-scaled chaotic artificial bee colony (FSCABC) algorithm and feedforward neural network (FNN). First, fruits images were acquired by a digital camera, and then the background of each image were removed by split-and-merge algorithm. We used a square window to capture the fruits, and download the square images to 256 × 256. Second, the color histogram, texture and shape features of each fruit image were extracted to compose a feature space. Third, principal component analysis was used to reduce the dimensions of the feature space. Finally, the reduced features were sent to the FNN, the weights/biases of which were trained by the FSCABC algorithm. We also used a stratified *K*-fold cross validation technique to enhance the generation ability of FNN. The experimental results of the 1653 color fruit images from the 18 categories demonstrated that the FSCABC–FNN achieved a classification accuracy of 89.1%. The classification accuracy was higher than Genetic Algorithm–FNN (GA–FNN) with 84.8%, Particle Swarm Optimization–FNN (PSO–FNN) with 87.9%, ABC–FNN with 85.4%, and kernel support vector machine with 88.2%. Therefore, the FSCABC–FNN was seen to be effective in classifying fruits.

1. Introduction

Fruit classification is a difficult and important task in supermarkets, since it is necessary for the cashier to know the categories of a particular fruit in order to determine its price. The use of barcodes has mostly resolved this problem for packaged products; however, most consumers want to pick their products by themselves. Some fruits cannot be packaged using barcodes, and thus must be weighted. One solution can be to issue codes for every fruit, but the tedious memorization of barcodes can lead to errors in pricing. Another solution can be to issue the cashier an inventory with pictures and codes; however, flipping over the inventory booklet can be overall time consuming (Rocha et al., 2010).

Scholars have proposed several effective solutions based on computer vision and machine learning to address similar problem during the last decade. Baltazar et al. (2008) first applied data fusion to nondestructive image of fresh intact tomatoes, followed by a three-class Bayesian classifier. Pennington and Fisher (2009) used clustering algorithm for fruits and vegetables classification. Pholpho et al. (2011) used visible spectroscopy for classification of non-bruised and bruised longan fruits, and combined the principal component analysis (PCA), Partial Least Square Discriminant Analysis and Soft Independent Modeling of Class Analogy to develop classification models. VeggieVision (Bolle et al., 1996) was a supermarket produce recognition system that consisted of an integrated scale and image system with a user-friendly interface. Hong et al. (2006) employed morphological examination to separate the walnut and hazelnut into three groups. Zhang and Wu (2012) chose the Max-Wins-Voting SVM with Gaussian RBF kernel to recognize the different categories of fruits. Cano Marchal et al. (2013) established an expert system based on computer vision to estimate the content of impurities in olive oil samples. Fan et al. (2013) used a two hidden layers of back-propagation artificial neural network to predict the texture characteristics from extrusion food surface images. Omid et al. (2013) developed an intelligent system based on combined fuzzy logic and machine vision techniques for grading of egg using parameters as defects and size of eggs.

The aforementioned techniques may have one or several of the following four shortcomings. First, they need extra sensors such as a gas sensor, an invisible light sensor, or a weight sensor. Second, the classifier is not suitable to all fruits, viz., it can only recognize

* Corresponding author.
E-mail addresses: zhangyudong@njnu.edu.cn (Y. Zhang), wangshuihua@njnu.edu.cn (S. Wang), glji@njnu.edu.cn (G. Ji), preetha.phillips@gmail.com (P. Phillips).

http://dx.doi.org/10.1016/j.jfoodeng.2014.07.001

the varieties of the same category. Third, the recognition systems are not robust because different fruit images may have similar or identical color and shape features. Fourth, customers complain that there is a misclassification among the fruits (Seng and Mirisaee, 2009).

The objective of this paper is to propose a novel fruit classification system based on computer vision, with aim of solving above four shortcomings to the utmost degree. First, we use merely a digital camera, getting rid of other complicated sensors. Second, the proposed classifier is expected to recognize as many types of fruits as possible. In this study, 18 types of fruits are included. Third, we capture not only conventional color and shape features, but also the influential texture features. Finally, the proposed classifier is expected to have good accuracy by the feedforward neural network (FNN), since it is a powerful tool among supervised classifiers and it can also classify nonlinear separable patterns and approximate an arbitrary continuous function (Coulibaly and Evora, 2007).

However, finding the optimal weights/biases of FNN is a difficult task because the traditional gradient-based optimization algorithms, such as back-propagation (BP) algorithms, are easily trapped in local extrema. Recently, there has been many global optimization algorithms available to train the FNN, such as Genetic Algorithm (GA), Simulating Annealing (SA) algorithm, and Particle Swarm Optimization (PSO). Unfortunately, the BP, GA, SA, and PSO algorithms all demand expensive computational costs, and can be easily trapped into the local best. This means that the result would probably end up without finding the optimal weights/biases of the FNN. In this study, we use the artificial bee colony (ABC) algorithm to find the optimal weights/biases of FNN. ABC algorithm was originally presented by Karaboga et al. (2004) under the inspiration of collective behavior on honey bees with better performance in function optimization problems compared with GA, differential evolution, and PSO (Karaboga and Basturk, 2008). As it is known, standard global optimization techniques conduct only one search operation in each iteration. For example, the PSO carries out a global search at the beginning stage and local search in the ending stage (Zhang and Wu, 2011); nevertheless, the ABC features the advantage in that it conducts both a global search and local search in each iteration. As a result, the probability of finding the optimal is significantly increased, which effectively avoids local optima to a large extent. In order to improve the performance of ABC, Zhang et al. embedded the fitness scaling strategy and chaotic theory into the ABC, naming their new method as fitness scaled chaotic artificial bee colony (FSCABC) (Zhang et al., 2011b).

The structure of the following paper is organized as: Section 2 contains the procedure of data processing, including the preprocessing, feature extraction and reduction, and stratified cross validation techniques. Section 3 describes the classifier FNN, introduces in the FSCABC algorithm to train the weights/biases of FNN, and summarizes the steps of our fruit recognition system. Section 4 presents the experimental results step by step. We also compared the results of our method FSCABC–FNN with GA–FNN, PSO–FNN, ABC–FNN, and kernel support vector machine (kSVM). Section 5 discussed the results. Final Section 6 is devoted to conclusions and future research.

2. Data processing

The acronyms used in the paper are listed in Table 1.

Table 1
Glossary.

Acronyms	Full expressions
ABC	Artificial bee colony
BP	Back-propagation
DE	Differential evolution
FNN	Feedforward neural network
FSCABC	Fitness-scaled chaotic artificial bee colony
GA	Genetic algorithm
kSVM	kernel support vector machine
LB	Lower bound
MBP	Momentum BP
MSE	Median square error
PCA	Principal component analysis
PSO	Particle swarm optimization
RBF	Radial basis function
SA	Simulating annealing
SVM	Support vector machine
UB	Upper bound

2.1. Preprocessing

First, we used image segmentation techniques to remove the background since our research only concentrate on the fruits. We chose the split-and-merge algorithm (Damiand and Resch, 2003; Xiao et al., 2001), which was based on a quadtree partition of an image. The method started at the root of the tree that represents the whole image. If the four sub-squares were found inhomogeneous, then it was split into four son-squares (the splitting process), and so on so forth. Conversely, if four sub-squares were homogeneous, they should be merged as several connected components (the merging process). The node in the tree was a segmented node. This process continued recursively until no further splits or merges were possible. Second, we used a square window to capture the fruit, making the fruit lying in the center of the window. Finally, we downsampled the square images to 256×256. Although the downsampling degraded the image quality, it made the algorithm performed faster. Fig. 1 shows all the steps of preprocessing.

2.2. Feature extraction and reduction

We proposed a composite feature space based on color, texture, and shape features of fruits. For a 256×256 image, we extracted 64 color features, 7 texture features, and 8 shape features. The feature extraction can be seen as the dimension reduction procedure. The original tri-color image vector space has a $256 \times 256 \times 3 = 196,608$ dimensions. By feature extraction operation, the size of the feature vector space is only $64 + 7 + 8 = 79$. Afterwards, PCA was employed to reduce the number of features further by the criterion that the reduced features should cover at least 95% variance of original features. Fig. 2 illustrates the flowchart of feature extraction and reduction.

2.2.1. Color discretization and histogram

So far, the color histogram is employed to represent the distribution of colors in an image (Siang Tan and Mat Isa, 2011). It represents the number of pixels that have colors in a fixed list of color range that span the image's color space (Maitra and Chatterjee, 2008). For monochromatic (grayscale) images, the set of possible color values is sufficiently small that each of those colors may be placed on a single range; then the histogram is merely the count of pixels that have each possible grayscale value. For color images using RGB space, the histogram is produced first by discretization of the colors in the image into a number of bins, and counting the number of image pixels in each bin.

Fig. 3 shows the illustration of color discretization. For each channel, the intensity value before discretization varies from 0 to 255 (Fig. 3a), hence, there are totally $256 \times 256 \times 256 = 2^{24} =$ 16,777,216 different colors. Then, we use 4 bins (Fig. 3b) to represent each color channel. Bin 0, 1, 2, 3 denotes intensities 0–63, 64–127, 128–191, and 192–255, respectively. This step is termed