

Date Fruits Classification using MLP and RBF Neural Networks

Khalid M. Alrajeh
Assistant Professor
Community College in Arriyadh
King Saud University

Tamer. A. A. Alzohairy
Assistant Professor
Community College in Arriyadh
King Saud University

ABSTRACT

This paper presents a new date fruits sorting system using artificial neural networks (ANN). The classification system are based on attributes extracted from dates fruits obtained from a computer vision system (CVS) used. Two different models of neural networks have been applied as classifiers: multi-layer perceptron (MLP) with backpropagation and radial basis function RBF networks. The aims of this study are to define a set of external quality features from the shape and color for different types of date fruits and to examine the effectiveness of the neural network models for image classification. In the experiments for performance evaluation the neural networks achieved a recognition rate equal to 87.5% and 91.1% respectively for MLP with backpropagation and RBF, which is consistent with the best results reported in the literature for the same data base and testing paradigms.

General Terms

Image Processing, Pattern Recognition.

Keywords

Backpropagation Algorithm, Classification, Color, Feature Extraction, LMS algorithm, Machine Vision, Multilayer Perceptrons (MLP) Neural Network, Neural Networks, K-means Clustering, Radial Basis Function (RBF) Neural Network.

1. INTRODUCTION

Neural networks have re-emerged as an important programming paradigm that attempts to mimic the functionality of the human brain. This area has been developed to solve demanding pattern processing problems, like speech and image processing, which were intractable or extremely cumbersome when implemented using traditional computing [2].

Over the last 20 years, there has been a higher and higher interest focused on the problems of recognition and identification of surfaces and image processing, via machine vision techniques and artificial intelligence.

Using these new technologies in developing classification systems of industrial products brought about various benefits such as: doing away with inconsistency and the dependence on human labor, raising accuracy and labor speed. Focusing on these premises, over the last years, several studies and researches have been carried out that hint to the adjustment of both these techniques and the image processing algorithms to the particularities of vegetal products classification.

The advantages and disadvantages of using human operators in the inspection and classification process of vegetal products

are presented in [7]. The disadvantage of using a semi-automatic sorting is given in [5]. The importance of using non-destructive techniques in the process of classifying vegetal products, based on video inspection has been stated in [1]. Automatic classification devices for vegetal products on the basis of color descriptor are given in [6]. Validates a series of identification algorithms of color and shape, by mixing up image preprocessing techniques with the algorithms belonging to the artificial intelligence neural networks is given in [10]. An algorithm of apples shape setting and identification of their stalk is developed in [11]. A sorting algorithm of fruit according to shape via neural networks is given in [9]. A quick method of orange classification based on image processing with correlation analysis in frequency domain was developed in [4]. Validates a series of classification algorithms of potatoes according to the shape, color, and analysis of faults via classification system is given in [8]. A series of identification using fuzzy and neural networks algorithms of the shape, color and size of the products by using processing techniques of images is given in [3].

From the above literatures, it was found that color, shape, size and surface defect of fruit are important features in classification.

The kingdom of Saudi Arabia is the world's largest producer of date fruit. It produces almost 400 date varieties in bulk. During the harvesting season the date grading and sorting pose problems for date growers. Since it is labor intensive and time consuming process, it delays the post harvesting operations which costs them dearly.

The date grading and sorting is repetitive process. In practice, it carried out by humans manually through visual inspection. The manual inspection poses further problems in maintaining consistency in grading and uniformity in sorting.

In this paper, we will design a neural prototypical computer vision system to speed up the process of date grading and sorting as well as maintain the consistency and uniformity. Taking into consideration the very large variation field of shape, color and size descriptors that are to be analyzed through the automatic classification process of date, all carried out while using both machine vision and neural networks techniques, going through some more phases is a must.

The paper is organized as follows: The two types of classifier neural networks are given in section 2. In section 3 we describe the basic components of a computer vision system (CVS) which is the core component of every computer mediated system. Section 4 provides the results and

discussion of the study. Finally, some conclusions are remarked in section 5.

2. ARTIFICIAL NEURAL NETWORKS AS A CLASSIFIER

Artificial Neural Networks (ANN) is an interconnection of simple processing nodes whose functionality is modeled after the node in the brain. The ANN consists of an input layer, an output layer and at least one layer of nonlinear processing elements, known as the hidden layer. The input values to the network are received from the input layer through the hidden layer to the output layer. The processing of input values is done within the individual nodes of the input layer and then the output values are forward to the nodes in the hidden layer. The values obtained as inputs by the hidden layer nodes are processed within them and forward to either the nodes of the next hidden layer or nodes of the output layer.

The processing capacity of the network is determined by the relative weights of the connections in the network. Supervised learning or training is the process of changing the weights of the links between the nodes of the network to map patterns presented at the input layer to target values of the output layer. This is done using training algorithms.

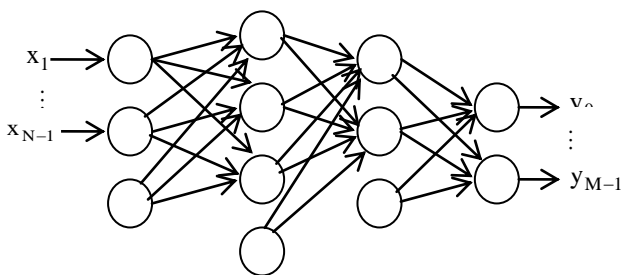
In the following subsections two different types of neural networks are presented for date fruits classification purposes.

2.1 MLP with BP Algorithm Network

In this subsection the structure of the MLP neural network and the backpropagation (BP) algorithm used to adjust its parameters for classification purpose are introduced.

2.1.1 Multilayer Perceptrons (MLP) Neural Network Architecture

The multilayer neural network architecture can have any number of layers. Fig. 1 illustrate a network architecture with 4 layers; first layer is called input layer and last layer is called output layer; in between first and last layers which are called hidden layers. In this paper two hidden layers are used to classify date fruit represented by features vector.



Input Layer Hidden Layer 1 Hidden Layer 2 Output Layer
Fig. 1: Multilayer perceptrons architecture.

2.1.2 Backpropagation Algorithm

Back propagation algorithm is the most popular in the supervised learning architecture because of the weight error correct rules. It is considered a generalization of the delta rule for nonlinear activation functions and multilayer networks.

Learning weights of the MLP by Backpropagation algorithm has two phases. First, a training input pattern is presented to the network input layer. The network propagates the input pattern from layer to layer until output pattern is generated by the output layer. If this pattern is different from the desired output, an error is calculated and then propagated backward through the network from the output layer to the input layer. The weights are modified as the error is propagated. The backpropagation training algorithm is an iterative gradient designed to minimize the mean square error between the actual output of multilayer feedforward perceptron and the desired output. It requires continuous differentiable non-linearity. The following assumes a sigmoid logistic nonlinearity.

Step 1: Initialize weights and offsets

Set all weights and node offsets to small random values.

Step 2: present input and desired outputs

Present a continuous valued input vector x_0, x_1, \dots, x_{N-1} and specify the desired output d_0, d_1, \dots, d_{M-1} . If the net is used as a classifier then all desired outputs are typically set to zero except for that corresponding to the class the input is from. That desired output is 1. The input could be new on each trial or samples from a training set could be presented cyclically until stabilize.

Step 3: Calculate actual output.

First calculate the formulas

$$I_j = f_j^{h1} \left[\sum_i w_{ji}^{h1} x_i + b_j^{h1} \right] \quad (1)$$

$$Z_j = f_r^{h2} \left[\sum_j w_{rj}^{h2} I_j + b_r^{h2} \right] \quad (2)$$

Where x_i is the i th network input, w_{ji}^{h1} is the connection weight from the i th input to the j th neuron in the 1st hidden layer, w_{rj}^{h2} is the connection weight from the j th neuron in the 1st hidden layer to the r th neuron in the 2nd hidden layer, b_j^{h1} is the weight from the bias to the j th neuron in the 1st hidden layer, b_r^{h2} is the weight from the bias to the r th neuron in the second hidden layer, $f_j^{h1}(\cdot)$ and $f_r^{h2}(\cdot)$ are nonlinear sigmoid activation functions defined as

$$f(\text{netinput}) = \frac{1}{1 + e^{-\text{netinput}}} \quad (3)$$

The network output is calculated by the following equation

$$\hat{y}(k+1) = f_k^O \left[\sum_k w_{kr}^O Z_r + b_k^O \right] \quad (4)$$

where w_{kr}^O is the weight connection of the k th neuron in the output layer to the r th neuron in the 2nd hidden layer, b_k^O is

the bias weight for the k^{th} output neuron, and f_k^O is the transformation function between 2nd hidden layer and output layer and it is a linear function.

Step 4: compute the error signal

$$e_j(k) = d_j(k) - \hat{y}_j(k) \quad (5)$$

Step 5: Adapt weights.

Use a recursive algorithm starting at the output nodes and working back to the first hidden layer. Adjust weights by

$$w_{ij}(k+1) = w_{ij}(k) + \eta \delta_j x_i \quad (6)$$

In this equation $w_{ji}(k)$ is the weight from hidden node i or from an input to node j at time k , x_i is either the output of node i or is an input, η is a gain term, and δ_j is an error term for node j . If node j is an output node then

$$\delta_j = \hat{y}_j(1 - \hat{y}_j)(d_j - \hat{y}_j) \quad (7)$$

where d_j is the desired output of node j and \hat{y}_j is the actual output.

If node j is an internal hidden node then

$$\delta_j = x_j(1 - x_j) \sum_k \delta_k w_{jk} \quad (8)$$

Where k is over all nodes in the layers above node j .

Internal node thresholds are adapted in a similar manner by assuming they are connection weights on links from auxiliary constant-valued inputs equal 1.

Step 6: Repeat by going to step 2.

2.2 RBF Neural Network

In this subsection the structure of the RBF neural network and the algorithms used to adjust its parameters for classification purpose are introduced.

2.2.1 Structure of RBF neural Network

Fig. 2 shows a typical RBF network, with q inputs (x_1, \dots, x_q) , and p outputs (y_1, \dots, y_p) . The hidden layer consists of h computing units connected to the output by h weight vectors $(\underline{\alpha}_1, \dots, \underline{\alpha}_h)$.

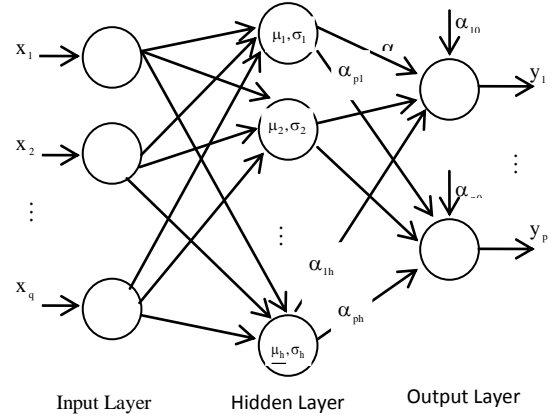


Fig. 2. Topology of a RBF neural network.

The response of one hidden unit to the network input at the i^{th} instant, \underline{x}_i , can be expressed by

$$\Phi_k(\underline{x}_i) = \exp\left[-\frac{1}{(\sigma_k^i)^2} \|\underline{x}_i - \underline{\mu}_k^i\|^2\right], \quad (k:1 \dots h) \quad (9)$$

where $\underline{\mu}_k^i$ is the center vector for k^{th} hidden unit at i^{th} instant, σ_k^i is the width of the Gaussian function at that time, and $\|\cdot\|$ denotes the Euclidean norm. The overall network response is given by

$$\hat{y}_i = f(\underline{x}_i) = \alpha_0^i + \sum_{k=1}^h \alpha_k^i \Phi_k(\underline{x}_i) \quad (10)$$

where $\underline{y}_i \in \mathbb{R}^p$, $\underline{x}_i \in \mathbb{R}^q$. The coefficient vector $\underline{\alpha}_k^i$ is the connecting weight vector of the k^{th} hidden unit to output layer, which is in the vector form of $\underline{\alpha}_k^i = [\alpha_{1k}^i, \dots, \alpha_{pk}^i]^T$. Thus the coefficient matrix of the network can be expressed as

$$A_{p \times h}^i = \begin{bmatrix} \alpha_{11}^i & \dots & \alpha_{1k}^i & \dots & \alpha_{1h}^i \\ \dots & \dots & \dots & \dots & \dots \\ \alpha_{p1}^i & \dots & \alpha_{pk}^i & \dots & \alpha_{ph}^i \end{bmatrix}$$

and the bias vector is $\alpha_0^i = [\alpha_{10}^i, \dots, \alpha_{i0}^i, \dots, \alpha_{p0}^i]^T$.

2.2.2 Training Radial Basis Function Network

Training RBF neural network consists of determining the location of centers and widths for the hidden layer and the weights of the output layer. It is trained using a two-phase approach: in the first phase, unsupervised learning occurs, whose main objective is to optimize the location of center and width. In the second phase, the output layer is trained in a supervised mode using the least mean-square (LMS) algorithm to adjust the weight so as to obtain the minimum mean square error at the output. The following are the three steps of the hybrid learning method for an RBF neural network, and they are discussed in more detail in the next three subsections:

- 1) Find the cluster centers of the radial basis function; use the k-means clustering algorithm.
- 2) Find the width of the radial basis function.
- 3) Find the weight; use LMS.

2.2.2.1 Calculation of Centers

To calculate the centers of the radial basis function we use the k-means clustering algorithm. The purpose of applying the k-means clustering algorithm is to find a set of clustered centers and a partition of training data into subclasses. The center of each cluster is initialized to a randomly chosen input datum. Then each training datum is assigned to the cluster that is nearest to itself. After training data have been assigned to a new cluster unit, the new center of a cluster represents the average of the training data associated with that cluster unit. After all the new centers have been calculated, the process is repeated until it converges. The recursive k-means algorithm is given as follows:

- 1) Choose a set of centers $\{\underline{\mu}_1, \underline{\mu}_2, \dots, \underline{\mu}_h\}$ arbitrarily and give the initial learning rate $\gamma(0) = 1$.

- 2) Compute the minimum Euclidean distance

$$L_i(k) = \|\underline{x}(k) - \underline{\mu}_i(k-1)\| \quad i: 1 \dots h \quad (11)$$

$$r = \arg \min L_i(k)$$

- 3) Adjust the position of this centers as follows:

$$\begin{aligned} \underline{\mu}_i(k) &= \underline{\mu}_i(k-1) + \gamma(k)(\underline{x}(k) - \underline{\mu}_i(k-1)) \quad (i = r) \\ &= \underline{\mu}_i(k-1) \quad (i \neq r) \end{aligned} \quad (12)$$

- 4) $k = k + 1$, $\gamma(k) = 0.998\gamma(k-1)$ and go to 2.

2.2.2.2 Width Calculation

After the RBF centers have been found, the width is calculated. The width represents a measure of the spread of data associated with each node. Calculation of the width is usually done using the P-nearest neighbor algorithm. A number P is chosen and for each center, the P nearest centers is found. The root-mean squared distance between the current cluster and its P nearest neighbors is calculated, and this is the value chosen for σ . So, if the current cluster center is $\underline{\mu}_j$, the value of width is given by:

$$\sigma_j = \sqrt{\frac{1}{P} \sum_{i=1}^P (\underline{\mu}_j - \underline{\mu}_i)^2} \quad (13)$$

A typical value of P is 2, in which case σ is set to be the average distance from the two nearest neighboring cluster centers.

2.2.2.3 Weight Estimation

Learning in the outer layer is performed after calculation of the centers and widths of the RBF in the hidden layer has been completed. The objective is to minimize the error between the observed output and desired one. It is commonly trained using the LMS algorithm [12] and is summarized as follows:

Training sample: Input signal vector = $\underline{\Phi}(k)$

Desired response = $\underline{d}(k)$

User-selected parameter: $0 < \eta < 1$

Initialization: Initialize the weights $\underline{\hat{w}}(0)$.

Computation: For $k = 1, 2, \dots$ Compute

$$\underline{e}(k) = \underline{d}(k) - \underline{\hat{w}}^T(k)\underline{\Phi}(k).$$

$$\underline{\hat{w}}(k+1) = \underline{\hat{w}}(k) + \eta \underline{\Phi}(k)\underline{e}(k).$$

3. COMPUTER VISION SYSTEM

In order to classify date fruits automatically we use a computer vision system which composed from two parts: 1) image acquisition hardware and 2) software recognition part. In the following subsections we will discuss these two parts.

3.1 Image Acquisition Hardware

The image acquisition hardware used in this paper is composed of two parts which are:

- 1) Sony Color camera.
- 2) Frame grabber (Matrox Meteor).

Which are connected to a compatible personal computer.

The system provides images of 768 per 576 pixels with a resolution of 3.5 mm pixel. The composite video signal is acquired by the frame grabber, and then digitized and decoded from the camera into three defined buffers in red, green and blue color coordinates (RGB). The lighting system used is composed of a ring-shaped florescent tube with hole in the top to place the camera. This setup for the hardware component will simplify the extraction of the object from background in the subsequence steps.

3.2 Software Recognition Part

The software part is based on the C++ programming language and Matlab routines. The software is basically used to process the acquired images in specific steps. All the processes to the acquired images, such as images enhancement, noise removal, and color separation of the required object from the background and implementation of the classification process are done in this section.

The software part involves three main tasks: 1) Separate foreground object from the background. 2) Extract features vector of the object. 3) Recognize the feature vectors obtained. In the following subsections we will discuss each task.

3.2.1 Separate Foreground Object from the Background

This stage is based on two different principles. Either on the similarity between the pixels in a region or on the discontinuities in the pixel values at the boundaries between regions [13], [14]. Segmentation approaches in the first category use thresholding and region growing, while those in the second category use on edge detection [15], [16].

Image thresholding is a simple and efficient technique to extract the regions in an image if the background of the object and surrounded light is controllable. In our case, the image thresholding approach works well, because there are regions of similar color levels contrasting with a different background level. In this case the histogram will show more than one mode, where each mode signifies one type of region. The thresholding values are adjusted to the valleys between the

modes in the histogram. It is an efficient technique in extracting the object from the background in our system.

3.2.2 Extract Features Vector of the Object

In this subsection, features such as perimeter, length, width, length to width ratio and color are obtained.

3.2.2.1 The Perimeter Feature

The perimeter can be computed by counting number of pixels in the boundary of the extracted object.

3.2.2.2 Length, Width and Length to Width Features

To get the length-to-width ratio of a shape, the vertical and horizontal projection for an extracted object is first calculated. Horizontal projection of the extracted object is used to count the total number of pixels of the extracted object in each row. The row with the maximum number of pixels is recorded. After that, beginning at this row, upper and lower rows are sought. The first upper row with a projected value that is smaller than a threshold is then marked as the top boundary (TB) of the interest object, and the first lower row with a projected value smaller than a threshold is marked as the bottom boundary (BB) of the object. The same technique is used on vertical projection to find the left boundary (LB) and right boundary (RB) of the interest object. The length and width of the interest object can be then calculated by:

$$\text{ObjectLength} = \text{TB} - \text{BB} \quad (14)$$

$$\text{ObjectWidth} = \text{RB} - \text{LB} \quad (15)$$

Finally, the length-to-width ratio (R) is derived as follows:

$$R = \frac{\text{Objectlength}}{\text{Objectwidth}} \quad (16)$$

3.2.2.3 Color Features

The color features provide powerful information for object recognition. A simple and effective recognition scheme is to represent and match images on the basis of color means. The three means of the basic colors (RGB) are computed and then they are put in the features vector.

3.2.3 Recognize the Feature Vectors

From above, the feature vector uses 7 features which are 1) the three colors, 2) perimeter 3) length, 4) width and 5) length-to-width ratio (R). This feature vector is used as input to the MLP with backpropagation neural network and RBF neural network given in section 2 for comparison.

Then the input layer, for both neural networks mentioned above, consists of 7 nodes. The values of the input vector must be scaled between the upper and lower bounds of the transfer functions, normally in the range [0, 1].

The number of output layer nodes depends on the types of date fruits needed to be recognized. In our study three types of date fruits in KSA are used as samples. The corresponding desired outputs are (1,0,0), (0,1,0) and (0,0,1). Those samples are Sufri, Sillaj and Sugii. The samples are brought from Kharj region. The number of samples under experiment is 220 date fruits. They are grouped into the following

categories: very good (VG), good (G), fair (F) and bad (B). In the beginning, classification is done manually based on human experience. Half of the samples have been used as training data for the neural networks. The second half has been used for testing each neural network.

4. RESULTS

In this section, the parameters given for each network type for classification purpose are given.

A RBF network with 20 hidden layer nodes and the learning rate used is $\eta = 0.1$. Training the RBF network is obtained in two phases for classification purpose:

First, the initial values for the location of centers and widths are obtained using unsupervised K-means clustering and P-nearest neighbor, given in subsection 2.2.2.1, and 2.2.2.2 respectively using Half of the samples.

Second, LMS method given in subsection 2.2.2.3 is used to find output layer weights initialized to very small values.

A MLP with 20 nodes in the 1st hidden layer and 10 nodes in the 2nd hidden layer are used. The learning rate is given by $\eta = 0.1$.

After training both networks it was found that. The RBF neural network performed best with average classification accuracy of 91.1% with respect to the number of correctly classified instances, it was followed by MLP neural network with backpropagation algorithm with a classification accuracy of 87.5%.

5. CONCLUSION

In this paper we have presented new date fruits sorting system. The system turns an image of the fruits into multiple features recognition problem where the extracted features of the same object are considered. Two neural networks are used for the recognition processes which are RBF neural network and MLP with backpropagation neural network. From the simulation results it was found that, RBF neural network performed better than MLP neural network with backpropagation algorithm when used to solve the classification problem.

6. ACKNOWLEDGEMENT

The author would like to thank King Saud University for its academic and financial support throughout the research work.

7. REFERENCES

- [1] Baoping, J. 1999. nondestructive technology for fruits grading. International conference on Agricultural Engineering Beijing.
- [2] Hecht-Nielson. 1989. Neurocomputing. Addison-Wesley, San Diego.
- [3] Buzera, M., Groza V., Prostean G. and Prostean O. 2008. techniques of analyzing the color or produces for automatic classification. 12th IEEE International Conference On Intelligent Engineering Systems, Miami, USA, pp. 209-214.
- [4] Cruvinel, P. E. et al. 2002. Image Processing in Automated Pattern Classification of Oranges. Transaction of the ASAE.

- [5] Dech Sidney H. 1994. Evaluations of semi-automated vegetable sorting concept. Transaction of the ASAE.
- [6] Laykin, S. et al. 2002. Image Processing Algorithm for Tomato Classification. Transaction of the ASAE, 45(3), 851-858.
- [7] Meyers, J. B. 1988. Improving dynamic visual inspection performance, Unpub. M.S. thesis, Univ. of Georgia, Athens.
- [8] Noordam, J. C. et al. 2000. High speed potato grading and quality inspection based on a color vision system, AGENG.
- [9] Simoes A. S. et al. 2002. Applying Neural Networks to Automated Visual Fruit Sorting. Transaction of the ASAE.
- [10] Sudhakara Rao P. Color Analysis of fruits using machine vision system for automatic sorting and grading. Journal Instrum. Soc. India. 34(4), 284-291.
- [11] Ying Y. et al. 2003. Detecting Stem and Shape of Pears using Fourier Transformation and an Artificial Neural Network. Transaction of the ASAE. 46(1).
- [12] Widrow and Hoff. 1960. Adaptive witching Circuits. IRE WESCON Convention Record, New York, pp. 96-104.
- [13] Thomas, A., Ferrari V., Leibe B., Tuytelaars T. and Gool L. V. 2009. Shape-from-recognition: Recognition enables meta-data transfer. Computer Vision and Image Understanding, 113(12), pp. 1222-1234.
- [14] Sun, T., Horng H., Liu C. and Tien F. 2009. Invariant 2D object recognition using KRA and GRA. Expert Systems with applications. 36(9), pp. 11517-11527.
- [15] Price, K. and Reddt R. 1979. Matching Segments of Image. IEEE Trans. On Pattern Analysis and Machine intelligence. Vol. Pami-1, no. 1, pp. 110-111.
- [16] Nevatia, R. and Ramesh K. 1980. linear feature Extraction and description. Computer graphics and Image Processing. Vol. 13, pp. 257-269.