Dates piece classification using Run-Length method with Back propagation neural network

Abdul rahman Finjan Shindi
Khulud Mohammed Gazi
M.Sc computer science

Abstract

There are many type of dates, in this research we will use the run-length method to extract the feature of these type of dates and then use one of classifying method (BPNN) to classify the type of dates.

Keywords:

Run-Length, Neural Networks, Dates classifying.

1.Introduction

There are many type of dates in nature so to distinguish between these types is not a simple task, its requires an expert able to cope this task. Such performances are difficult to achieve using an automatic system for diagnosis. Generally, Dates can be divided into two different types of dates: popular type and rare type, we will examine a different type (50 type) and extract the feature for each image (Date type) by using run-length method and then classify these types by using BPNN.

2. Background

When examining natural images, it is often necessary to interpret tissue appearance based on different characteristics such as smoothness, grain, regularity, and homogeneity. These attributes are related to the local intensity variations and can be captured by using various texture metrics [1]. Run-length statistics capture the coarseness of a texture in specified directions. A run is defined as a string of consecutive pixels which have the same gray level intensity along a specific linear orientation. Fine textures intensities tend to contain more short runs with similar gray level, while coarse textures have more long runs with significantly different gray level intensities [6]. Most dates image contains a gaps so its texture contains a runs.

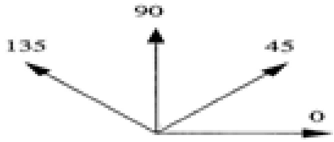
3. Definition of Run-Length Matrix [2]

The concept of run-length was proposed in the 1950s. A run-length matrix P is defined as follows: each element P*(i, j)* represents the number of runs with pixels of gray level intensity equal to *i* and length of run equal to *j* along a specific orientation. The size of the matrix P is *n* by *k*, where *n* is the maximum gray level in the texture and *k* is equal to the possible maximum run length in the corresponding image. An orientation is defined using a displacement vector *d(x, y)*, where *x* and *y* are the displacements for the x-axis and y-axis , respectively. The typical orientations are 0°, 45°, 90°, and 135°, and calculating the run-length matrix for each direction will produce four run-length matrices, see figure (1). Once the run-length matrices are calculated along each direction, several texture descriptors are calculated to capture the texture properties and differentiate among different textures. These descriptors can be used either with respect to each direction or by combining them if a global view of the texture information is required. Ten descriptors are typically extracted from the run-length matrices: short run emphasis (SRE), long run emphasis (LRE), high gray-level run emphasis (HGRE), low gray-level run emphasis (LGRE), pair-wise combinations of the length and gray level emphasis (SRLGE, SRHGE, LRLGE, LRHGE), run-length nonuniformity (RLNU), and grey-level non-uniformity (GLNU).

Original image

1	1	1	2	2
3	4	2	2	3
4	4	4	4	4
5	5	3	3	3
1	1	3	4	5

Four directions



Run length

0°	1	2	3	4	5
1	0	1	1	0	0
2	0	2	0	0	0
3	3	0	1	0	0
4	2	0	0	0	1
5	1	1	0	0	0

Run length

45°	1	2	3	4	5
1	5	0	0	0	0
2	0	2	0	0	0
3	4	0	1	0	0
4	5	1	0	0	0
5	3	0	0	0	0

90°	1	2	3	4	5
1	5	0	0	0	0
2	2	1	0	0	0
3	4	1	0	0	0
4	5	1	0	0	0
5	3	0	0	0	0

135°	1	2	3	4	5
1	5	0	0	0	0
2	4	0	0	0	0
3	6	0	0	0	0
4	5	1	0	0	0
5	3	3	0	0	0

Figure 1

Example of run-length matrices with image of 5x5 pixels

4. Run-Length feature [3,4]

In this paragraph we will explain the ten feature of the run-length method and its formula:

a) *Short Run Emphasis (SRE)*

$$SRE = \frac{1}{n_r}\sum_{i=1}^{M}\sum_{j=1}^{N}\frac{p(i,j)}{j^2}$$

This feature measure the distribution of short runs for the image . The Short Run Emphasis is highly dependent on the occurrence of short runs and is expected large for fine textures. So in our work (dates classification) this feature work strongly where many dates type contains fine texture.

b) *<u>Long Run Emphasis (LRE)</u>*

$$LRE = \frac{1}{n_r}\sum_{i=1}^{M}\sum_{j=1}^{N}P(i,j)*j^2$$

This feature measure the distribution of long runs. The Long Run Emphasis is highly dependent on the occurrence of long runs and is expected large for coarse structural textures. Also in dates classification we will find coarser area for each date piece so this feature work strongly.

c) *<u>High Gray-Level Run Emphasis (HGRE)</u>*

$$HGRE = \frac{1}{n_r}\sum_{i=1}^{M}\sum_{j=1}^{N}P(i,j)*i^2$$

This feature measures the distribution of high gray level values. The High Gray Run Emphasis is expected large for the image with high gray level values. This feature consider with the gray level so this will examine in our work.

d) *<u>Low Gray-Level Run Emphasis (LGRE)</u>*

$$LGRE = \frac{1}{n_r}\sum_{i=1}^{M}\sum_{j=1}^{N}\frac{p(i,j)}{i^2}$$

This feature measures the distribution of low gray level values. The Large Gray Run Emphasis is expected large for the image with low gray level values. This feature consider with the gray level so this will examine in our work.

e) *<u>Short Run Low Gray-Level Emphasis (SRLGE)</u>*

$$SRLGE = \frac{1}{n_r}\sum_{i=1}^{M}\sum_{j=1}^{N}\frac{p(i,j)}{i^2*j^2}$$

This feature measures the joint distribution of short runs and low gray level values. The Short Run Low Gray Emphasis is expected large for the image with many short runs and lower gray level values. In dates image it's rare to find short runs so we expect it's will not work strongly here.

f) *<u>Short Run High Gray-Level Emphasis (SRHGE)</u>*

$$SRHGE = \frac{1}{n_r}\sum_{i=1}^{M}\sum_{j=1}^{N}\frac{p(i,j)*i^2}{j^2}$$

This feature measures the joint distribution of short runs and high gray level values. The SRHGE is expected large for the image with many short runs and high gray level values.

g) *Long Run Low Gray-Level Emphasis (LRLGE)*

$$LRLGE = \frac{1}{n_r}\sum_{i=1}^{M}\sum_{j=1}^{N}\frac{p(i,j)*j^2}{i^2}$$

This feature measures the joint distribution of long runs and low gray level values. The Long Run Low Gray Emphasis is expected large for the image with many long runs and low gray level values.

h) *Long Run High Gray-Level Emphasis (LRHGE)*

$$LRHGE = \frac{1}{n_r}\sum_{i=1}^{M}\sum_{j=1}^{N}P(i,j)*i^2*j^2$$

This feature Measures the joint distribution of long runs and high gray level values. The Long Run High Gray-Level Emphasis is expected large for images with many long runs and high gray level values.

i) *Gray-Level Non-uniformity (GLNU)*

$$GLNU = 1/n_r \sum_{j=1}^{N}\left(\sum_{i=1}^{N} P(i,j)\right)^2$$

This feature measures the similarity of gray level values throughout the image. The Gray Level Non-uniformity is expected small if the gray level values are alike throughout the image.

j) *Run Length Non-uniformity (RLNU)*

$$RLNU = 1/n_r \sum_{i=1}^{N}\left(\sum_{j=1}^{N} P(i,j)\right)^2$$

This feature measures the similarity of the length of runs throughout the image. The Run Length Non-uniformity is expected small if the run lengths are alike throughout the image. Figure 2 shows the run-length histogram for some dates images.

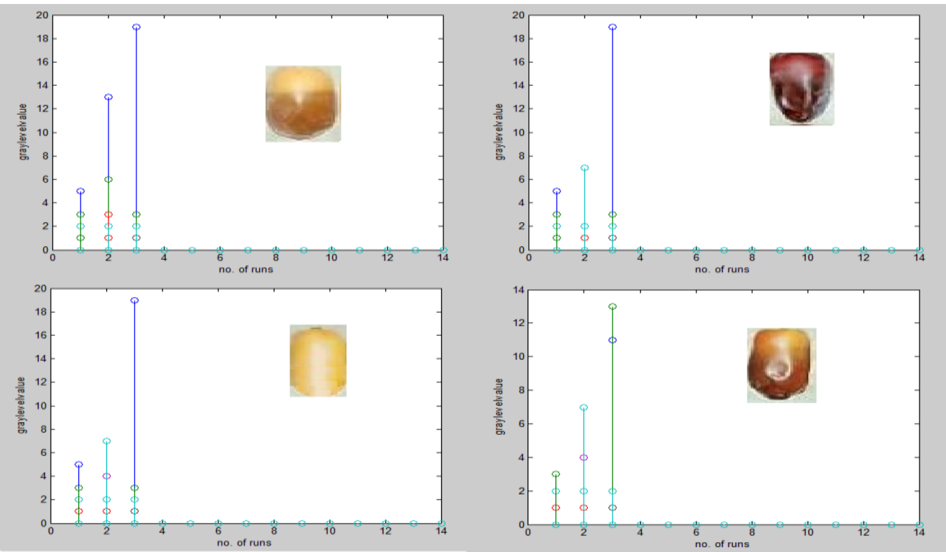


Figure 2

Run-length histogram for some dates images

5. Classification algorithm

Since the main focus of this work is the feature extraction algorithm, we use a neural network as a classifier for our work. For several decades, Artificial Neural Network (ANNs) have been extensively studied with particular applications to the field of machine learning. Their widespread appeal is in their adaptively, parallelizability and ability to model many different kinds of systems, including highly nonlinear ones. One of the most popular and perhaps simplest structures for implementing ANNs is the feed forward neural network, where information passes between multiple layers of neurons in only one direction. Recurrent neural networks take this concept one step further and allow for information to pass in the other direction, creating feedback that can improve learning in dynamic setting. With both types of neural networks, the most popular method of learning is through back propagation, where error between a target output and output from the network is propagated back through the network, as means of updating its parameters (or weights).[6,7]

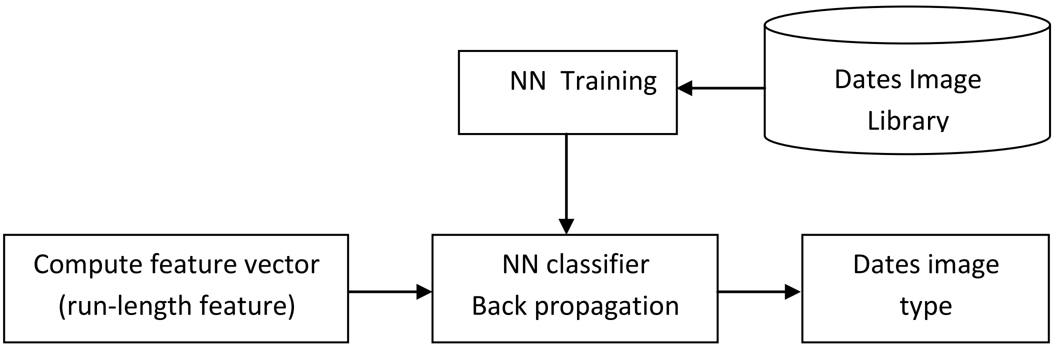


Figure 3: Dates texture recognition algorithm tasks.

6. Neural Network Structure

We used a feed forward backpropagation neural network with adaptable learning rate. The NN have 3 layer; an input layer (10 neuron), a hidden layer (30 neuron) , and output layer (1 neuron). The activation function used is the tan sigmoid function, for both the hidden and the output layer. The input to the neural network is the feature vector containing 10 components these are the texture feature (run-length feature), the NN has only one output as shown in figure 4.

7. Dates Texture Library [5]

The library of *Palm Texture* is comprised of 100 images of date textures of size 32X32 pixel. The library consisted of variety of date types of different. Samples of that library is shown in figure 3.

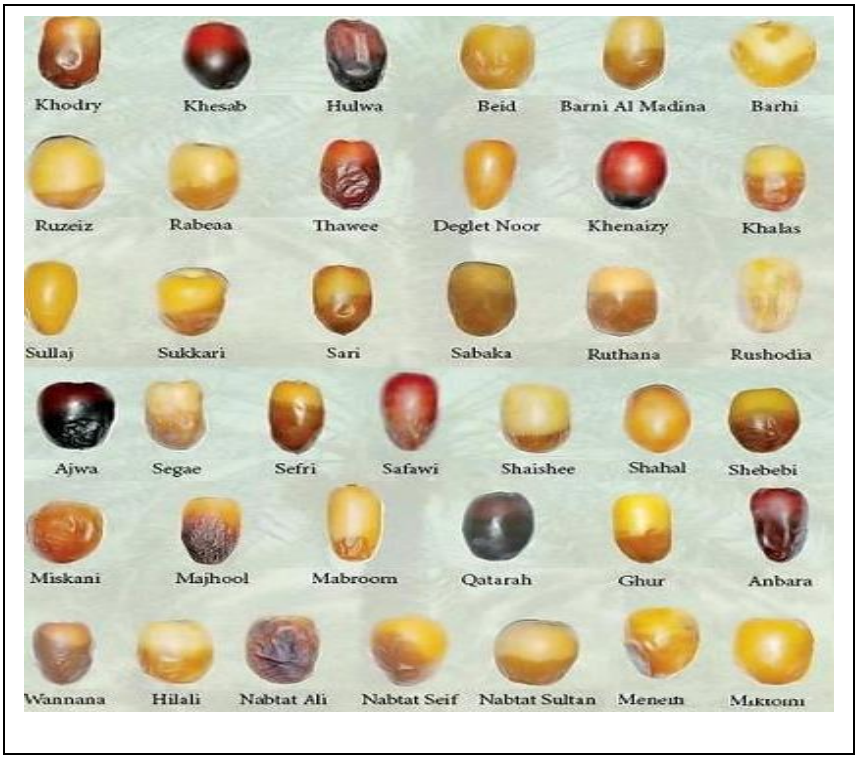


Figure 4: The skin library samples.

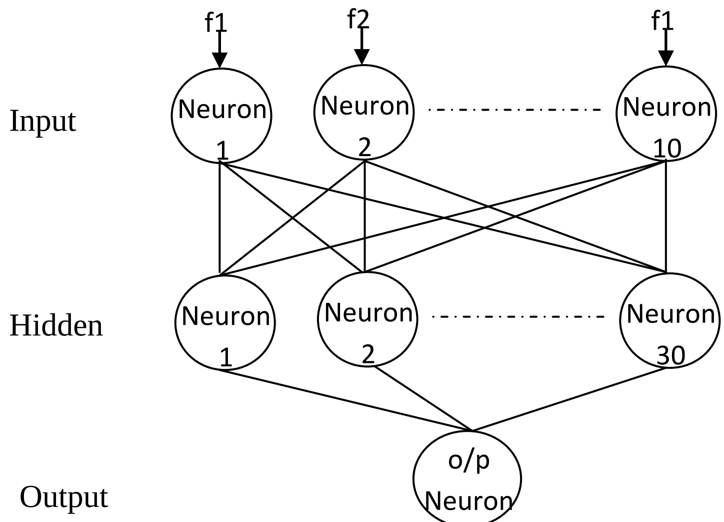


Figure 5: The neural network structure.

References

1-Xiaoou Tang," Dominant run-length method for image classification", 1997, Woods Hole Oceanographic institute.

2-Jong ding & Dianeu Tan, "Blind Image steganalysis based on run-length histogram analysis", 2008, National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences.

3- Jong ding , Wei Wang, Tieniu Tan and Yun Q. Shi."Run-length and edge statistics based approach for image splicing detection", 2008, National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences.

4- Dong-Hui Xu, Arati S. Kurani, Jacob D. Furst, Daniela S. Raicu," RUN-LENGTH ENCODING FOR VOLUMETRIC TEXTURE", 2004,Intelligent Multimedia Processing Laboratory.

5-www.AL-Q.com.

6- Nidhal K. Al abbadi, Nizar Saadi Dahir& Zaid Abd Alkareem, " SKIN TEXTURE RECOGNITION USING NEURAL NETWORKS",2008, IT Research and Development Center, Kufa University, Najaf – Iraq

7- Vicente & Itamar, "A Backpropagation Neural Network Design Using Adder-Only Arithmatic",2007, Department of Electrical Engineering & Computer Science.