International Journal on
"Technical and Physical Problems of Engineering"
(IJTPE)
Published by International Organization of IOTPE

ISSN 2077-3528
IJTPE Journal
www.iotpe.com
ijtpe@iotpe.com

March 2015 | Issue 22 | Volume 7 | Number 1 | Pages 78-85

SKIN DISEASE IMAGES RECOGNITION BASED ON CLASSIFICATION METHODS

S. Salimi M. Sabbagh Nobarian S. Rajebi

Electrical Engineering Department, Seraj Higher Education Institute, Tabriz, Iran
soheil.salimi.n@gmail.com, mehran.ms20@yahoo.com, s.rajebi@seraj.ac.ir

Abstract- Pattern Recognition methods developed in recent five decade are a scientific way to classification of objects into a number of classes. Bayesian Algorithm, *K* Nearest Neighbor (KNN), Parzen Window and Multi-layer Perceptron (MLP) are number of some methods that each has performance in specific fields. This paper presents the performance of this methods in classifying the numerous skin diseases images into twelve classes that each class represent a type of skin disease. It will focus on improving the Accuracy Percentage VS to classifying time.

Keywords: Classification, *K* Nearest Neighbor (KNN), Bayesian, Parzen Window, Multilayer Perceptron (MLP), Divergence.

I. INTRODUCTION

In the statistical approach, each pattern is representing in terms of d features or measurements and is view as a point in a d-dimensional space. The goal is to choose those features that allow pattern vectors belonging to different categories to occupy compact and disjoint regions in a *d*-dimensional feature space. The effectiveness of the representation space (feature set) is determined by how well patterns from different classes can be separated. Given a set of training patterns from each class, the objective is to establish decision boundaries in the feature space, which separate patterns belonging to different classes. In the statistical decision theoretic approach, the decision boundaries are determined by the probability distributions of the patterns belonging to each class, which either must be specified or learned [1, 4, 5].

Machine vision is an important area in pattern recognition. A machine vision system captures images via a camera and analyzes them to extract feature from them. A typical application of a machine vision system is in the medicine cases. Illness detection from an image, pathology photo, radiology photo, heart signal even an image taken directly from skin, can be an application of pattern recognition. For example, in detection of skin Disease, we can use a camera to takes the photos for detecting the type of disease. Thus, images have to be analyze on time, and the pattern recognition system has to classify the photos into disease classes [2]. There are many algorithm and methods to achieve the best classifier. Classification Methods that used in this paper involves the Bayesian, KNN, Parzen window and MLP.

II. BAYESIAN METHOD

One of most commonly density functions in practice is the Gaussian or normal density function. The symbol $N(\mu,\sum)$ is used to denote a Gaussian PDF with mean value μ and covariance $\sum$. For a set of data, that have Gaussian PDF can be used Bayesian rule in order to classification the unknown data samples in known classes. In this method for each new test data, probability of each class is expressed by the following equation:

$$P_i(x)=\frac{1}{(2\pi)^{d/2}|\Sigma_i|^{1/2}}e^{\left(-\frac{1}{2}(x-\mu_i)^t\Sigma_i^{-1}(x-\mu_i)\right)} \quad (1)$$

where, P_i is the probability of classifying the test sample (x) in class(i), $\sum_i$ is the covariance matrix of the training data of class(i), μ_i is the mean value of training data of class(i), d is the number of features and $|\sum_i|$ is the determinant of the $\sum_i$ [2]. General form of a Gaussian density function with its characteristics is shown in Figure 1.

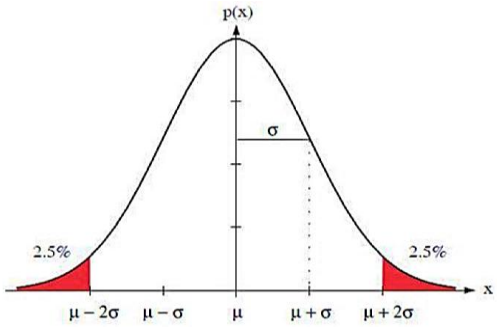

Figure 1. General form of Gaussian pdf [1]

In practice if number of feature were high, the inverse of covariance matrix maybe not defined because of zero determinant of covariance matrix. In order to solve this problem in practice one of solution is that trace of the covariance matrix multiply to a constant value. In experimental result its value is between 1.3-1.5 [2].

III. K-NEAREST NEIGHBOR

In this method similar to Bayesian method, for each test samples the probability of sitting in each class is calculate but the difference between those is in their algorithms.

The *K*-Nearest Neighbor method that called KNN method can be used for both normal & abnormal density functions. *K* is the number of Neighbors from special class that lies in nearest distance of test data. The farthest sample from *K* sample determine the radius of range in calculations. In this method, value of *K* in first step of algorithm has defined. Follow equation is used to calculate the probabilities [2].

$$P_i(x) = K_i / NV \tag{2}$$

where P_i is the occurrence probability of class(i) for input test data (x), K_i is the number of neighbors from class(i), N is total samples from all classes that lies in increases radius and V is the volume determined by increases radius that often in practice is equal to radius instead of suppose sphere volume.

$$\text{Sphere Volume}: \ V = 4\pi r^3 / 3 \tag{3}$$

Error of this method in between Bayesian error and double of Bayesian error [1].

$$P_B < P_{knn} < 2P_B \tag{4}$$

In this method sometimes same probability for classes are calculated and finally the algorithm cannot classify the test sample. In such case, there are some approximation to classify test data. One of this approximation can be choose the first response of the KNN method, but surely will have not good result. One another approximation can be using the Minimum distance classification method that will explained in next section. Recent approximation will have a better response than prior

IV. MINIMUM DISTANCE

If we have classes with equal probability, this method can be useful. Philosophy of this method is based on distance of test sample from another samples, therefor first step in this method is the calculation this distances. There are two way to approach the point-to-point distance of two sample [2].

A. Euclidean Distance

If the classes have equal covariance, then the Equation (5) is used [2].

$$d_E = |x - u_i| \tag{5}$$

B. Mahalanobis Distance

If the classes have different covariance, then the Equation (6) is used [2].

$$d_m = \left(|x - u_i|^T \Sigma^{-1} |x - u_i| \right)^{1/2} \tag{6}$$

V. PARZEN WINDOW

This method is very similar to KNN method. The difference is that radius of volume is constant instead of *K*. For calculate the probabilities of classes we will use from Equation (7) [2]:

$$P_i(x) = \frac{K}{N} \tag{7}$$

where K is the number of samples from class(i) lies inside the defined radius and N is the total samples lies inside defined radius around the test data.

In this method main effort is around the finding the best radius to calculate the probabilities. In order to approach this we can divide the training samples into two part that called Parzen test and Parzen training data. Then we must train the system by Parzen training data for various radius and then test the system by Parzen test data. Finally, by a simple comparison between the derived results we can select the best Parzen window radius [2].

Sometimes this method have more than one response and sometime have not response. To solve these problems we can use the approximations that introduced in KNN method section [2].

VI. Multilayer Perceptron

Neural networks can be viewed as massively parallel computing systems consisting of an extremely large number of simple processors with many interconnections. Neural network models attempt to use some organizational principles (such as learning, generalization, adaptively, fault tolerance and distributed representation, and computation) in a network of weighted directed graphs in which the nodes are artificial neurons and directed edges (with weights) are connections between neuron outputs and neuron inputs. The main characteristics of neural networks are that they have the ability to learn complex nonlinear input-output relationships, use sequential training procedures, and adapt themselves to the data [3].

The most commonly used family of neural networks for pattern classification tasks is the feed-forward network, which include multilayer perceptron network. This network are organized into layers and have unidirectional connections between the layers. The learning process involves updating network architecture and connection weights so that a network can efficiently perform a specific classification task. The increasing popularity of neural network models to solve pattern recognition problems has been primarily due to their seemingly low dependence on domain-specific knowledge (relative to model-based and rule-based approaches) and due to the availability of efficient learning algorithms for practitioners to use [1, 3].

A general mapping of multilayer perceptron network is shown in Figure 2. As shown in Figure 2, each neuron from each unit is connected to all neurons of next unit with weight of W_{ij} or W_{jk}. Activation function of input unit neurons considered equal to 1. However, for hidden unit and output unit neurons assumed a sigmoid function called φ [3].

$$\varphi = 1/(1 + e^{-\alpha x}) \tag{8}$$

where α is appositive constant coefficient that depends on data samples.

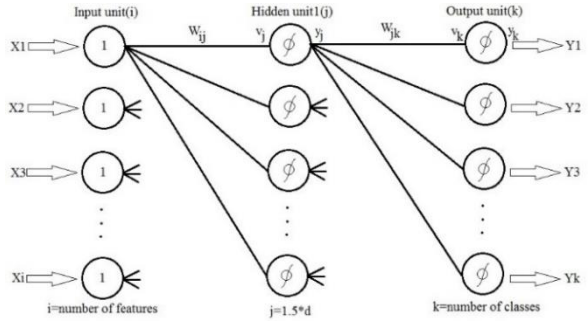

Figure 2. Two layer perceptron network mapping [3]

In this method before starting the training of system. In first step random values between -1, +1 are assigned to layer weights and then start training network. Network training process is carry out in follow two main phase:

A. Feed Forward

In this phase, one of test samples is applied to input unit and then outputs of network are calculated in follow way:

$$y(k)=\varphi\left(v(k)\right) \tag{9}$$

$$v(k)=W_{kj}*y(j) \tag{10}$$

$$y(j)=\varphi\left(v(j)\right) \tag{11}$$

$$v(j)=W_{ij}*x(i) \tag{12}$$

Number of neurons in output unit is equal to number of classes, number of neurons in input unit is equal to number of features, and number of neurons in hidden unit is equal to 1.5-time neuron number in input unit [3].

B. Back Propagation

After derivation the outputs in feed forward phase, layer weights must be modified. In order to modify this weights its need to calculate the error of outputs that earned by difference of resulting outputs and ideal outputs. It is derive in following way:

$$e(k)=d(k)-y(k) \tag{13}$$

$$\varepsilon=\frac{1}{2}\sum_{k=1}^{class}e^2(k) \tag{14}$$

$$\Delta W_{jk}=\frac{\partial\varepsilon}{\partial W_{jk}}=\frac{\partial\varepsilon}{\partial e_k}*\frac{\partial e_k}{\partial y_k}*\frac{\partial y_k}{\partial v_k}*\frac{\partial v_k}{\partial W_{jk}}$$

$$W_{jk}=\frac{\partial\varepsilon}{\partial W_{jk}}=-e_k\,\varphi'(v_k)\,y_j \tag{15}$$

$$W_{jk}(new)=W_{jk}+\eta\Delta W_{jk} \tag{16}$$

where η is a constant coefficient between 0.2-0.5. In order to modify the W_{ij} similar process in acted as Equation (17).

$$\frac{\partial\varepsilon}{\partial W_{ij}}=\frac{\partial\varepsilon}{\partial e_k}*\frac{\partial e_k}{\partial y_k}*\frac{\partial y_k}{\partial v_k}*\frac{\partial v_k}{\partial y_j}*\frac{\partial y_j}{\partial v_i}*\frac{\partial v_i}{\partial W_{ij}}$$

$$W_{ij}=\frac{\partial\varepsilon}{\partial W_{ij}}=\left(\sum_{k=1}^{class}-e_k\,\varphi'(v_k)W_{jk}\right)*\varphi'\left(v_j\right)x_i \tag{17}$$

$$W_{ij}(new)=W_{ij}+\eta\Delta W_{ij}$$

After modifying the weights in back propagation phase for first training data, repeat same two phase for all training data. In each level of repeat this process, weights are modified gradually.

This operation is known as one Epoch. At end of each Epoch, mean of output errors is calculated by following way that indicate the Epoch error [3].

$$\varepsilon_{av}=\frac{1}{12}\sum_{k=1}^{class}e_k \tag{18}$$

To have a desirable result in multilayer perceptron network we must training the system in several Epochs. 20 Epoch is a normal iterate to approach a desirable classification of samples. Mean error at end of Epochs must be approximately constant after 20th Epoch [3].

VII. DATA WHITENING

Both training and test data sample that used in neural network must be white data. Data that have features with mean near zero and variance equal to 1 is called as white data. However, in practice it is probable to have a distributed data with various mean and variance of features.

There is an approximation in order to convert data to white data that main property of feature not changed. Following equation, perform such conversion on data to be white data [3].

$$\text{new feature} = \frac{\text{feature - mean of feature}}{\text{square mean variance of feature}}$$

VIII. DIVERGENCE OF CLASSES

When an error is occurred in some classes, it is very important that what the separability measure of classes is? There are some methods to achieve a measure specify the separability of classes. One best of those is the Divergence meaning.

For a two class problem case, the classification error probability depends on the difference between $P(w_1|x)$, $P(w_2|x)$. Hence, the ratio $\frac{P(w_1\mid x)}{P(w_2\mid x)}$ can convey useful information concerning the discriminatory capabilities associated with an adopted feature vector x. the same information desire from $\ln\frac{P(x\mid w_1)}{P(x\mid w_2)}$. Since x takes different values, it is natural to consider the mean value over classes [2].

$$D_{12}=\int_{-\infty}^{+\infty}P(x\mid w_1).\ln\frac{P(x\mid w_1)}{P(x\mid w_2)}dx \tag{19}$$

$$D_{21}=\int_{-\infty}^{+\infty}P(x\mid w_2).\ln\frac{P(x\mid w_2)}{P(x\mid w_1)}dx \tag{20}$$

In addition, total divergence between two classes is sum of Equations (19) and (20):

$$d_{12}=D_{12}+D_{21} \tag{21}$$

Assuming now that the density functions are Gaussian, the computation of divergence is simplified and it is not difficult to show that [2].

$$d_{ij} = \frac{1}{2}\text{trace}\left(\Sigma_i^{-1}\Sigma_j + \Sigma_j^{-1}\Sigma_i - 2I\right) + \\ + \frac{1}{2}\left(\mu_i - \mu_j\right)^T\left(\Sigma_i^{-1} + \Sigma_j^{-1}\right)\left(\mu_i - \mu_j\right) \quad (22)$$

IX. FEATURE SELECTION

In problems that our data has enormous features, the simulation running time will be very long due to the large size of matrixes in order to overcome to this problem we can only select the features that they have majority effect on discrimination function.

First step in this process is finding the features that have majority effects. In order to there are many methods to find majority effective features and one best those is called Fisher's Discriminant Ratio (FDR).

This approach is specify a matrix called separability matrix. In this matrix elements of trace is present the strength of separability of each feature proportional to its index. Elements that has a large value has a great effect in discriminant function and elements with less value has a negligible effect in result of classification. This matrix is calculated by Equation (23) [2].

$$FDR = \frac{\left(\mu_1 - \mu_2\right)^2}{\sigma_1^2 + \sigma_2^2} \quad (23)$$

For problems that there are L number of classes must be use from Equations (24):

$$\mu_k = \frac{1}{N_k}\sum_{x^q \epsilon C_k} x^q \quad (24)$$

$$S_k = \sum_{x^q \epsilon C_k}\left(x^q - \mu_k\right)\left(x^q - \mu_k\right)^T \quad , \quad k = 1,\ldots,L \quad (25)$$

$$\text{Within Scatter Matrix:} \quad S_W = \sum_{k=1}^{L} S_K \quad (26)$$

$$S_B = \sum_{k=1}^{L} N_K\left(\mu_k - \mu\right)\left(\mu_k - \mu\right)^T \quad (27)$$

$$FDR = \text{Separability Matrix} = S_W^{-1}S_B \quad (28)$$

X. SIMULATION RESULTS

In this paper we used the explained algorithms to classifying the 504 number of skin disease images into 12 class that each class represent a typical diseases of skin. This image is preprocessed and generated features from images are in our hands. Total number of samples is equal to 768 that 264 number of those are training samples, 504 number of those are test samples, and each sample has 48 number of features.

A. Divergence of Classes

Divergences of classes for these samples are calculated and the results are shown in Table 1 as a matrix form. Total divergence of system is defined as follow that has a value about 2512 for this data:

$$d = \sum_{i=1}^{M}\sum_{j=1}^{M} P\left(w_i\right)P\left(w_j\right)d_{ij} \quad (29)$$

Minimum divergence of classes is between the class 1&3 and is equal to 108.8 that is 4% of total divergence. Minimum divergence between two classes express that discriminant of those classes is difficult and error probability is higher.

B. Bayesian Results

By multiplying the trace of covariance matrix to constant value of 1.5 the classification results is improved and finally the Correct Classification Rate (CCR) matrix that represent the assignment of data in classes is derived that shown in Table 2.

By attention to Table 2, error is occurred between class 10&11 and by attention to Table 1 Divergence between this two classes is equal to 156 that is 6% of total divergence and is 20th minimum divergence

This method has a very good result for our data percentage of accuracy is derived 99.6%. In addition, time of simulation is little than other methods. It is running is 0.6 seconds in our simulator system.

C. KNN Results

After simulation of this method for our data samples for k=1 to k=20 percentage of accuracy is derived as shown in Table 3 and Figure 3.

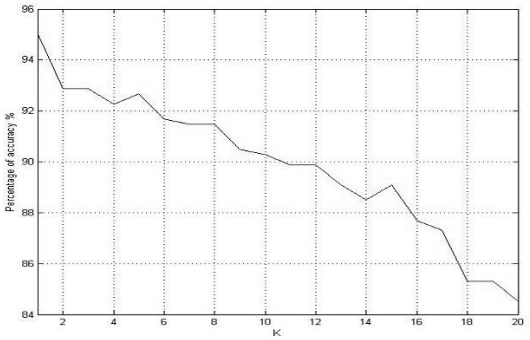

Figure 3. Percentage of accuracy of KNN method for k = 1–20

The CCR matrix resulted from this method is shown in Table 4. As shown in Table 4 majority of errors is occurred between classes 1&3 that there in minimum divergence equal to 108.8. For k=1 the classification accuracy of method is equal to 95.04% that running in 3 seconds in our simulator system.

D. Parzen Results

Dividing of training samples into Parzen test & Parzen training samples must be a random selection of training samples; therefor this method must be applied for several state of random selected samples to find the best Parzen window radius. We has run simulation program in three state of random selection to approach a desirable Parzen window. Each state results a value as best Parzen window that best of those selected as best radius.

In each state by increasing the Parzen window from two to 300 by step of 0.5, classifier are designed and test sequential and finally the best radius is selected according to its accuracy. In order to have a representation an accuracy versus Parzen radius, those are presented in Figure 4.

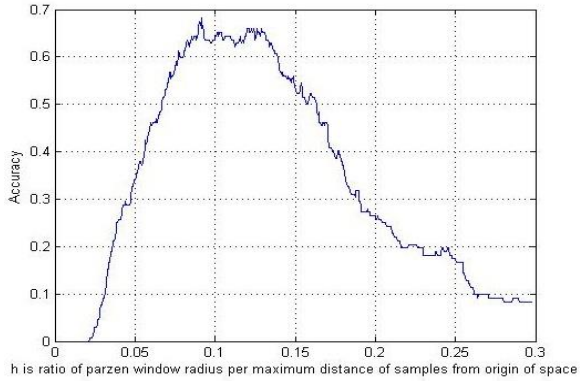

Figure 4. Accuracy of system versus Parzen window size, *h* is ratio of Parzen window radius per max. distance of samples from origin of space

Sometime this method unable to classify the input data in any classes, in such cases we use Minimum distance algorithm to specify it to a class. Also sometimes, there is a stat that Parzen method have found several response for an input data to classifying. In such cases, we select the class that its samples in nearby the input sample.

CCR matrix of this method is shown in Table 5. According to Table 5 majority of classification error is occurred between classes 1&3 that those has minimum of divergence between of classes. This method have a not bad result for classification of our data that it has a percentage of accuracy about 94.25%, but is running in much long time. It has taken about 7 minute to simulation of method for classification our samples.

E. Multilayer Perceptron Results

In this method as the weights in first step of algorithm have specified by random values therefore it is probable that this method Results a different percentage of Accuracy Responses in every running of simulation program due to the random values in first step. Parameters that effect on this problem are:

1) "α" in sigmoid function of neurons;
2) "η" in equation of modifying the weights
3) Number of epochs

By assuming 25 epoch for this method, and linear decreasing the "α", "η" from 2.5 to 0.3, this method has results a good classification. By 10 time repeating the algorithm to find the minimum and maximum of percentage of accuracy a good classification result is derived. Minimum percentage of accuracy is equal to 97.4% and Maximum percentage of accuracy is derived 98.01%.

Average error of outputs in multilayer perceptron at end of each Epoch is calculated and shown in Figure 5.

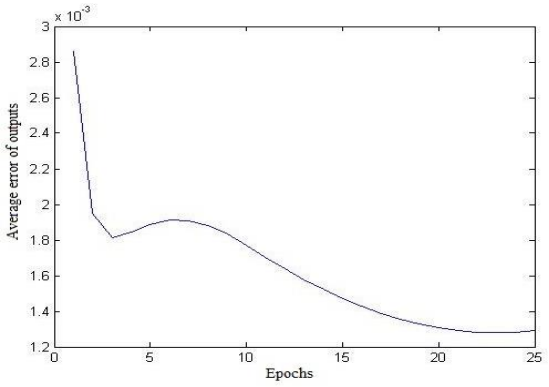

Figure 5. Average error of output at end of Epochs

F. Feature Selection Results

By using the Fisher Discriminant Method, separability matrix is derived that is a 48×48 matrix. Trace of FDR matrix determine the cost of each feature proportional to its index. Trace of FDR matrix is shown as a vector in Table 6.

Respectively lowest features is shown in Table 7 that first element is the low cost feature and last element is show the high cost feature in our data. By respectively elimination of low cost features accuracy of methods are studied and finally the results is shown in Tables 8-11. Proportional to Tables 8-11 same results are shown in Figures 6-9 in order to have a better present of Feature elimination results in various methods.

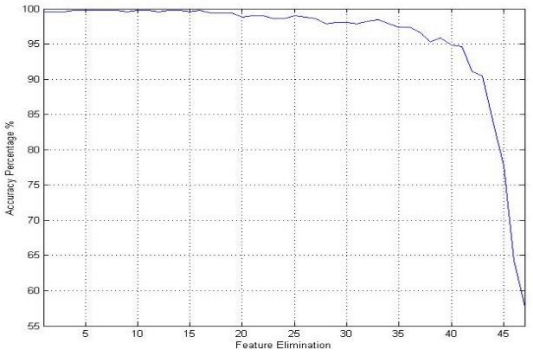

Figure 6. Accuracy of Bayesian method VS to elimination of features

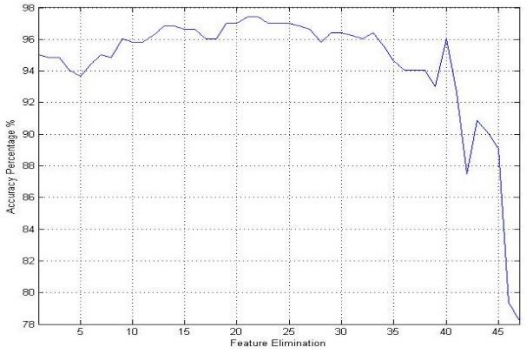

Figure 7. Accuracy of KNN method VS to elimination of features

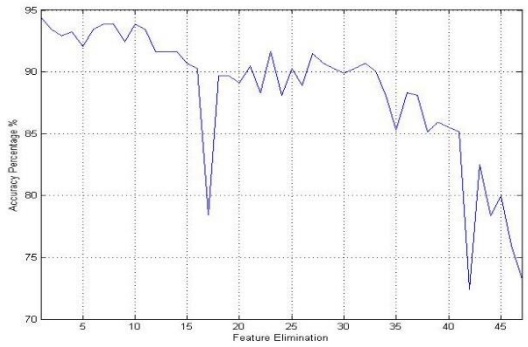

Figure 8. Accuracy of Parzen method VS to elimination of Features

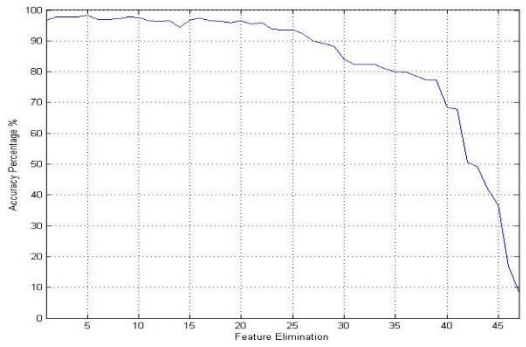

Figure 9. Accuracy of Perceptron method VS to elimination of features

Table 1. Divergence of classes for our samples

DIV	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8	Class 9	Class 10	Class 11	Class 12
Class 1	0	822	109	1031	1630	428	1877	4263	156	2449	2062	631
Class 2	822	0	765	2591	2197	2336	6800	4141	976	770	399	361
Class 3	109	765	0	1578	767	794	2324	1425	312	1616	1869	587
Class 4	1031	2591	1578	0	10218	790	3542	4910	465	2992	2664	2492
Class 5	1630	2197	767	10218	0	7026	9332	2902	3365	1953	2978	1599
Class 6	428	2336	794	790	7026	0	2767	6709	334	4877	3998	1711
Class 7	1877	6800	2324	3542	9332	2767	0	4871	1781	10788	10816	5885
Class 8	4263	4141	1425	4910	2902	6709	4871	0	4456	2867	3734	3702
Class 9	156	976	312	465	3365	334	1781	4456	0	2140	1861	869
Class 10	2449	770	1616	2992	1953	4877	10788	2867	2140	0	156	765
Class 11	2062	399	1869	2664	2978	3998	10816	3734	1861	156	0	503
Class 12	631	361	587	2492	1599	1711	5885	3702	869	765	503	0

Table 2. CCR Matrix of Bayesian Method

CCR	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8	Class 9	Class 10	Class 11	Class 12
Class 1	42	0	0	0	0	0	0	0	0	0	0	0
Class 2	0	42	0	0	0	0	0	0	0	0	0	0
Class 3	0	0	42	0	0	0	0	0	0	0	0	0
Class 4	0	0	0	42	0	0	0	0	0	0	0	0
Class 5	0	0	0	0	42	0	0	0	0	0	0	0
Class 6	0	0	0	0	0	42	0	0	0	0	0	0
Class 7	0	0	0	0	0	0	42	0	0	0	0	0
Class 8	0	0	0	0	0	0	0	42	0	0	0	0
Class 9	0	0	0	0	0	0	0	0	42	0	0	0
Class 10	0	0	0	0	0	0	0	0	0	41	1	0
Class 11	0	0	0	0	0	0	0	0	0	1	41	0
Class 12	0	0	0	0	0	0	0	0	0	0	0	42

Table 3. Percentage of accuracy of KNN method for k = 1–20

K	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
ACC %	95.0	92.8	92.8	92.2	92.6	91.6	91.4	91.4	90.4	90.2	89.8	89.8	89.0	88.4	89.0	87.7	87.3	85.3

Table 4. CCR Matrix of KNN Method

CCR	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8	Class 9	Class 10	Class 11	Class 12
Class 1	36	0	4	0	0	0	0	0	2	0	0	0
Class 2	0	42	0	0	0	0	0	0	0	0	0	0
Class 3	7	0	35	0	0	0	0	0	0	0	0	0
Class 4	0	0	0	42	0	0	0	0	0	0	0	0
Class 5	0	0	0	0	42	0	0	0	0	0	0	0
Class 6	0	0	0	0	0	40	0	0	2	0	0	0
Class 7	0	0	0	0	0	0	42	0	0	0	0	0
Class 8	0	0	0	0	0	0	0	42	0	0	0	0
Class 9	0	0	0	0	0	2	0	0	40	0	0	0
Class 10	0	0	0	0	0	0	0	0	0	42	0	0
Class 11	0	0	0	0	0	0	0	0	0	0	38	4
Class 12	0	0	0	0	0	0	0	0	0	0	4	38

Table 5. CCR matrix of Parzen method

CCR	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8	Class 9	Class 10	Class 11	Class 12
Class 1	37	0	2	0	0	0	0	0	3	0	0	0
Class 2	0	42	0	0	0	0	0	0	0	0	0	0
Class 3	10	0	32	0	0	0	0	0	0	0	0	0
Class 4	0	0	0	42	0	0	0	0	0	0	0	0
Class 5	0	0	0	0	42	0	0	0	0	0	0	0
Class 6	0	0	0	0	0	40	0	0	2	0	0	0
Class 7	0	0	0	0	0	0	42	0	0	0	0	0
Class 8	0	0	0	0	0	0	0	42	0	0	0	0
Class 9	0	0	0	0	0	4	0	0	38	0	0	0
Class 10	0	0	0	0	0	0	0	0	0	42	0	0
Class 11	0	0	0	0	0	0	0	0	0	0	38	4
Class 12	0	0	0	0	0	0	0	0	0	0	4	38

Table 6. Trace of FDR matrix

	1	2	3	4	5	6	7	8	9	10	11	12
FDR_T	0.1030	0.4456	0.0560	0.2758	1.0000	0.4921	0.2749	0.3031	0.0927	0.1919	0.1063	0.0778
	13	14	15	16	17	18	19	20	21	22	23	24
FDR_T	0.0559	0.0336	0.0888	0.0042	0.0143	0.0191	0.0494	0.0350	0.1646	0.0952	0.0758	0.0294
	25	26	27	28	29	30	31	32	33	34	35	36
FDR_T	0.2462	0.2972	0.3784	0.0542	0.2837	0.2538	0.2029	0.2289	0.3145	0.2805	0.1870	0.0104
	37	38	39	40	41	42	43	44	45	46	47	48
FDR_T	0.0741	0.1464	0.0067	0.1775	0.1837	0.3054	0.0774	0.0540	0.1072	0.0257	0.0369	0.0518

Table 7. Respectively lowest features costs

	1	2	3	4	5	6	7	8	9	10	11	12
F_COST	16	39	36	17	18	46	24	14	20	47	19	48
	13	14	15	16	17	18	19	20	21	22	23	24
F_COST	44	28	13	3	37	23	43	12	15	9	22	1
	25	26	27	28	29	30	31	32	33	34	35	36
F_COST	11	45	38	21	40	41	35	10	31	32	25	30
	37	38	39	40	41	42	43	44	45	46	47	48
F_COST	7	4	34	29	26	8	42	33	27	2	6	5

Table 8. Accuracy of Bayesian method vs to elimination of features

Feature	1	2	3	4	5	6	7	8	9	10	11	12
ACC %	99.6	99.6	99.6	99.8	99.8	99.8	99.8	99.8	99.6	99.8	99.8	99.6
Feature	13	14	15	16	17	18	19	20	21	22	23	24
ACC %	99.8	99.8	99.6	99.8	99.4	99.4	99.4	98.8	99.01	99.01	98.61	98.61
Feature	25	26	27	28	29	30	31	32	33	34	35	36
ACC %	99.01	98.81	98.61	97.82	98.02	98.02	97.82	98.21	98.41	97.82	97.42	97.42
Feature	37	38	39	40	41	42	43	44	45	46	47	
ACC %	96.62	95.25	95.83	94.84	94.64	91.07	90.48	83.73	77.78	64.29	57.74	

Table 9. Accuracy of KNN method vs to elimination of features

Feature	1	2	3	4	5	6	7	8	9	10	11	12
ACC%	95.04	94.84	94.84	94.05	93.65	94.44	95.04	94.84	96.03	95.83	95.83	96.23
Feature	13	14	15	16	17	18	19	20	21	22	23	24
ACC%	96.83	96.83	96.63	96.63	96.03	96.03	97.02	97.02	97.42	97.42	97.02	97.02
Feature	25	26	27	28	29	30	31	32	33	34	35	36
ACC%	97.02	96.83	96.63	95.83	96.43	96.43	96.23	96.03	96.43	95.63	94.64	94.64
Feature	37	38	39	40	41	42	43	44	45	46	47	
ACC%	94.05	94.05	93.03	96.06	92.66	87.5	90.87	90.08	89.09	79.37	78.17	

Table 10. Accuracy of Parzen method vs to elimination of Features

Feature	1	2	3	4	5	6	7	8	9	10	11	12
ACC %	94.41	93.45	92.89	93.25	92.06	93.45	93.85	93.85	92.46	93.85	93.45	91.67
Feature	13	14	15	16	17	18	19	20	21	22	23	24
ACC %	91.67	91.67	90.67	90.28	87.45	89.68	89.68	89.09	90.48	88.29	91.67	88.1
Feature	25	26	27	28	29	30	31	32	33	34	35	36
ACC %	90.28	88.89	91.47	90.67	90.28	89.88	90.28	90.67	90.08	88.1	85.32	88.29
Feature	37	38	39	40	41	42	43	44	45	46	47	
ACC %	88.1	85.12	85.91	85.52	85.12	72.42	82.52	78.37	79.96	75.99	73.21	

Table 11. Accuracy of Perceptron method vs to elimination of features

Feature	1	2	3	4	5	6	7	8	9	10	11	12
ACC %	96.62	97.81	97.81	97.81	98.21	97.02	97.02	97.22	97.81	97.61	96.42	96.23
Feature	13	14	15	16	17	18	19	20	21	22	23	24
ACC %	96.42	94.44	96.62	97.42	96.42	96.23	95.83	96.42	95.43	95.83	93.84	93.45
Feature	25	26	27	28	29	30	31	32	33	34	35	36
ACC %	93.45	92.26	89.88	89.08	88.29	84.12	82.34	82.34	82.34	80.95	79.96	79.76
Feature	37	38	39	40	41	42	43	44	45	46	47	
ACC %	78.57	77.18	77.18	68.45	67.65	50.59	49.2	41.46	36.7	16.86	8.33	

XI. CONCLUSIONS

According to aforementioned discussions, pattern recognition methods to classification unknown samples have a different result depends on data. In this case that studied in this paper, Bayesian algorithm has a very good result to classification our unknown samples in order to detecting the kind of diseases. Bayesian method contain a two important preference, one is the very good accuracy of percentage that in best feature selection result 99.8%, and one other is the time of simulation running in simulator system that about equal to 0.3 second in our simulator system. Also, Perceptron Network have result a good accuracy about 98.2% in best feature selection.

As show in last section it is very important to select features that has majority effect in classification, also, some features have a destructive effect of classification accuracy, hence we can specify o select features that have majority effect of classification results. By constructing the FDR matrix and selection the best features, we found that 15-majority feature from 48 total features give us about 90% of maximum Accuracy, that can be said by selecting the less third of features we can achieve to 90% of accuracy. This approach can be makes to decreasing the simulation time due to the small size of matrixes.

REFERENCES

[1] A.K. Jain, R.P.W. Duin, J. Mao, "Statistical Pattern Recognition: A Review", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, No. 1, January 2000.

[2] S. Theodoridis, K. Koutroumbas, "Pattern Recognition", Academic Press, San Diego, USA, 1999.

[3] C.M. Bishop, "Neural Network for Pattern Recognition", Clarendon Press, Oxford, 1995.

[4] L. Devroye, L. Gyorfi, G. Lugosi, "A Probabilistic Theory of Pattern Recognition", Springer-Verlag, Berlin, 1996.

[5] R.O. Duda, P.E. Hart, "Pattern Classification and Scene Analysis", John Wiley & Sons, New York, 1973.

BIOGRAPHIES

Soheil Salimi was born in Ardabil, Iran, in 1989. He received the B.Sc. degree in Electrical Engineering from University of Tabriz, Bonab Faculty, Bonab, Iran in 2012.

Mehran Sabbagh Nobarian was born in Tabriz, Iran in 1992. He received the B.Sc. degree in Electronics Engineering from Seraj Higher Education Institute, Tabriz, Iran in 2013. He is a member of Yong Researcher Club in Tabriz Branch, Islamic Azad University, Tabriz, Iran.

Saman Rajebi was born in Tabriz, Iran in 1981. He received his B.Sc. degree in Electronics Engineering from University of Tabriz, Tabriz, Iran in 2003. He received the M.Sc. degree in Communication Engineering from Urmia University, Urmia, Iran. Currently, he is a Ph.D. student at Urmia University. While pursuing his studies for M.Sc., he worked for Urmia ICT Center and Iranian Telecommunication Research Center. He is a full ime Lecturer at Seraj Higher Education Institute, Tabriz, Iran and a member of executive committee of ICTPE Conferences. His areas of interests are PLC, microwave applications and modeling and simulation of communication systems and antenna designing.