**ÇUKUROVA UNIVERSITY**

**INSTITUTE OF NATURAL AND APPLIED SCIENCES**

**MASTER THESIS**

**Emine GEZMEZ**

# CLUSTERING MRI IMAGES WITH PRINCIPAL COMPONENT ANALYSIS METHODS

**DEPARMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING**

**ADANA-2007**

**ÇUKUROVA ÜNİVERSİTESİ**

**FEN BİLİMLERİ ENSTİTÜSÜ**

**CLUSTERING MRI IMAGES WITH PRINCIPAL COMPONENT ANALYSIS METHODS**

Emine GEZMEZ

**YÜKSEK LİSANS TEZİ**

ELEKTRİK-ELEKTRONİK ANABİLİM DALI

Bu tez 24 / 12 / 2007 tarihinde Aşağıdaki Jüri Üyeleri Tarafından Oybirliği ile Kabul Edilmiştir.

| **Yrd. Doç. Dr. Turgay İBRIKÇİ** | **Yrd. Doç. Dr. Sami ARICA** | **Yrd. Doç. Dr. Bülent MİTİŞ** |
|---|---|---|
| **DANIŞMAN** | **ÜYE** | **ÜYE** |

Bu tez Enstitümüz Elektrik-Elektronik Anabilim Dalında hazırlanmıştır.

**Kod No:**

**Prof. Dr. AzizERTUNÇ**
Enstitü Müdürü

# ÖZ

## YÜKSEK LİSANS TEZİ

**TEMEL BİLEŞENLER ANALİZİ METODLARI İLE MRI GÖRÜNTÜLERİNİN KÜMELENMESİ**

Emine GEZMEZ

ÇUKUROVA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK-ELEKTRONİK ANA BİLİM DALI

Danışman : Yrd. Doç. Dr. Turgay İBRİKÇİ
Yıl : 2007 Sayfa : 44
Juri : Yrd. Doç. Dr. Turgay İBRİKÇİ
Yrd. Doç. Dr. Sami ARICA
Yrd. Doç. Dr. Bülent MITIŞ

Görüntülerin sınıflandırılmasındaki en önemli problemlerden biri de görüntü boyutlarının çok fazla olmasıdır. Bu yüzden, görüntüleri sınıflandırmada iyi sonuç alabilmek için görüntü boyutlarının indirgenmesi gerekir.

Bu çalışmada, Temel Bileşenler Analizi ile beynin MRI görüntülerinin boyutları indirgendikten sonra elde edilen görüntülere sınıflandırma metodları uygulanmıştır. General Hebbian Algorithm, Diamantras and Kung's APEX Rule, Expectation-Maximization Principle Component Analysis, Probabilistic Principal Components Analysis ve True-PCA olmak üzere 5 farklı Temel Bileşenler Analizi metodu kullanılmıştır. Elde edilen yeni görüntülere ise K-Means ve Fuzzy C-Means sınıflandırma metodları uygulanmıştır. Bu metodlar sonucunda elde edilen görüntüler karşılaştırılarak en iyi sonucu veren metodlar tespit edilmiştir

**Anahtar Kelimeler:** Temel Bileşenler Analizi, Kümeleme, MRI

**ABSTRACT**

**MSc THESIS**

**CLUSTERING MRI IMAGES WITH PRINCIPAL COMPONENT ANALYSIS METHODS**

Emine GEZMEZ

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING
INSTITUTE OF NATURAL AND APPLIED SCIENCES
UNIVERSITY OF ÇUKUROVA

Supervisor : Asst. Prof. Dr. Turgay İBRİKÇİ
Year : 2007 Pages : 44
Jury : Asst. Prof. Dr. Turgay İBRİKÇİ
Asst. Prof. Dr. Sami ARICA
Asst. Prof. Dr. Bülent MITIŞ

It's problem that the images have a complicated high dimensional structure in image clustering. Because of this, the dimensions of MRI images must be reduced.

The aim in the thesis is to implement PCA and image clustering methods and compare the methods. Different PCA and image clustering methods were implemented in Matlab. MRI images were used in the thesis. In the beginning, PCA methods were implemented on MRI images. The dimensions of MRI images were reduced by using PCA methods without much loss of information. After PCA methods were implemented, image clustering methods were implemented on MRI images. In the thesis five PCA methods (General Hebbian Algorithm, Adaptive Principal Component Analysis, Expectation-Maximization Principle Component Analysis, Probabilistic Principal Components Analysis and True-PCA) and two image clustering methods (K-Means and Fuzzy C-Means) were implemented.

**Keywords:** Principal Component Analysis, Clustering, MRI

**ACKNOWLEDGEMENTS**

I would like to express my respects and gratitude to my supervisor Asst. Prof. Dr. Turgay İBRİKÇİ. Many thanks for all of his supports, guidance, patience, cooperation, suggestions on initiating, improving and completing this study.

I would like to thank to my family, my father Ali, my mother Fatma, my sister Bilge and my brother Cihan. They always encouraged and supported me with their loves and inspirations.

I would like to thank to my committe members Asst. Prof. Dr. Sami ARICA and Asst. Prof. Dr. Bülent MITIŞ for their supports and very valuable discussions.

Finally, I thank all my friends, especially my friend Seyhan Yılmaz and great people that I couldn't mention the name of here, for their good wishes and encouragements.

**CONTENTS** **PAGE**

**NOTATIONS**

**W** : The synaptic weights

**y** : Output matrix

**x** : Input matrix

**i** : Input node

**j** : Output node

**C** : A lower triangular matrix

**N** : The number of data

$\mathbf{n_c}$ : The number of center

$\mathbf{c_j}$ : Any center

$\mathbf{v_i}$ : The data sample belonging to center $c_j$

**t** : The centres and the data are written in terms of time *t*

**v(t)** : The data

**z** : The nearest center to the data *v(t)*

$\mathbf{c_z(t-1)}$ : The center site at the previous clustering step

**η(t)** : The adaptation rate

**Y** : *pxn* matrix of all the observed data

**X** : *kxn* matrix of unknown states

**k** : The number of leading eigenvectors

**E** : Square Error Cost Function

**U** : Membership Matrix

$\mathbf{d_{ij}}$ : The Euclidean dinstance

**c** : The number of clusters

$\mathbf{c_i}$ : Cluster center of fuzzy cluster *i*

**μ** : The data mean

**ε** : A noise model

**A** : The data matrix where each column is a data point

**V** : The space of the first *k* principal component

$\mathbf{n_z(t)}$ : The number of data samples that have been assigned to the center up to the time *t*

**a** : Parameter constant

**b** : Parameter constant

**m** **:** Weighting exponent

$\Re$ **:** Real numbers

**p** **:** The number of eigenvectors

**LIST OF FIGURES** **PAGE**

# 1. INTRODUCTION

Principal Components Analysis (PCA) is a way of identifying patterns in data, and expressing the data for emphasizing their similarities and differences. Since patterns in data can be hard to find in data of high dimension, where the luxury of graphical representation is not available. PCA is a powerful tool for analysing data. The other main advantage of PCA is that once these patterns in the data can be found, and the data can be compressed by reducing the number of dimensions, without much loss of information. Principle Component Analysis is a dimension reduction technique. It is a technique for creating new variables which are linear combinations of the original variables. The new variables are referred to as the principal components and are selected such that they are uncorrelated with each other.

Furthermore, the first principal component accounts for the maximum variance in the data, the second principal component accounts for the maximum of the variance not yet explained by the first component, and so on. The maximum number of new variables that can be formed is equal to the number of original variables. They are probably not all needed and a selected number of principal components are used.

Image clustering is a means for high-level description of image content. The goal is to find a mapping of the image into classes (clusters) provide essentially the same prediction, or information. The generated classes provide a concise summarization and visualization of the image content.

In most clustering methods (e.g. K-Means), a distance measure between two data points or between a data point and a class center is given a priori as part of the problem setup. The clustering task is to find a small number of classes with low intra-class variability. However in many clustering problems, e.g. image clustering, the objects we want to classify have a complicated high dimensional structure and choosing the right distance measure is not a straight-forward task. A choice of a specific distance measure can influence the clustering results.

In clustering methods, different new approachs were improved. Fuzzy C- Means clustering and genetic algorithm (GA) for an automatic segmentation of white matter (WM), gray matter (GM), cerebro spinal fluid (CSF), the extra cranial regions and the presence of tumor regions were used and implemented (Selvathi, Arulmurgan, Thamarai

Seivi, Alagappan, 2005). Also, nonlinear PCA (KPCA) and regularized least squares classification (RLSC) algorithm were used to differentiate malignant (cancerous) from benign (noncancerous) soft tissues tumors in MRI images (Juntu, Sijbers and Dyck, 2007). In Juntu's study, it was mentioned that the effect of bias fields on the PCA analysis and propose to carry out PCA in the Fourier domain so that the principal components are extracted form certain frequency coefficients of the MRI image.

In the other study, a fully automatic technique was proposed to obtain image clusters. A modifed Fuzzy C-Means (FCM) clustering algorithm is used to provide a fuzzy partition. This method is less sensitive to noise as it filters the image while clustering it and the filter parameters are enhanced in each iteration by the clustering process. This method was applied t on a noisy CT scan and on a single channel MRI scan ( Mohamed, Ahmed, Farag, 1999).

The Ejection Fraction is one of improved researches. The Ejection Fraction is also an important measurement for the early prognosis and treatment monitoring of Cardio Vascular Diseases. In this study, it was defined as the volume of blood pumped from the heart between the diastolic (muscle relaxed) and systolic (muscle contracted) phases. Multi-slice synchronised MR images were used at both the systolic and diastolic phases. The images were first smoothed using an Adaptive Smoothing Algorithm. The images were then segmented using a K-Means Unsupervised Clustering technique (Lynch, 2000).

The aim in the thesis is to implement PCAs and image clustering methods, and compare these methods. Different PCAs and image clustering methods were implemented in Matlab (Matlab, 2001). MRI image was used in the thesis. In the beginning, PCA methods were implemented on MRI images. The dimensions of MRI image were reduced by using PCA methods without much loss of information. Because it's problem that the images have a complicated high dimensional structure in image clustering. Because of this, after PCA methods were implemented, image clustering methods were implemented on MRI images.

In the thesis, five PCA methods and two image clustering methods were implemented.

**PCA Methods**

- The Generalized Hebbian Algoritm (GHA)
- Adaptive Principal Component Analysis (APEX)
- Probabilistic Principal Components Analysis (PPCA)
- Expectation-Maximization Principle Component Analysis (EM-PCA)
- True-PCA

**Image Clustering Methods**

- K-Means Algorithm
- Fuzzy C-Means Algorithm

In the final analysis, PCA and image clustering methods were compared and determined the best method .

The thesis is organized as follows:

Principle Component Analysis (PCA) is reviewed in Chapter 2.

In Chapter 3, PCA methods (GHA, APEX, EM-PCA, PPCA and True-PCA) discussed, and they are compared with each others.

Image clustering methods (K-Means and Fuzzy C-Means) are reviewed in Chapter 4.

Experimental results are discussed in Chapter 5.

In Chapter 6, the conclusions are discussed.

# 2. PRINCIPLE COMPONENT ANALYSIS (PCA)

Principal Components Analysis (PCA) is a method that reduces data dimensionality by performing a covariance matrix of data set. It is suitable for data sets in multiple dimensions. PCA is recommended as an exploratory tool. PCA will explore correlations between samples. The goal of PCA is to summarize the data, it is not considered a clustering tool. In other words, PCA is a powerful tool for analysing data (Ricardo, 1998). The other main advantage of PCA is that once these patterns in the data can be found, and the data can be compressed, ie. by reducing the number of dimensions, without much loss of information (Smith, 2002).

## 2.1. To Perform Principle Component Analysis

There are five steps to perform a Principle Component Analysis on a set of data.

- Subtract The Mean
- Calculate The Covariance Matrix
- Calculate The Eigenvectors And Eigenvalues Of The Covariance Matrix
- Choosing Components And Forming A Feature Vector
- Deriving The New Data Set

### 2.1.1. Subtract The Mean

For PCA to work properly, the mean is subtracted from each of the data dimensions. The mean subtracted is the average across each dimension. So, all the $x$ values have the mean of the $x$ values of all the data points subtracted, and all the $y$ values have the mean of the $y$ values of all the data points subtracted. This produces a data set whose mean is zero (Smith, 2002).

| Data = | x | y |
|---|---|---|
| | 1 | 2 |
| | 2.5 | 3 |
| | 0.5 | 1.5 |
| | 1.5 | 1 |
| | 2 | 0.5 |

| DataAdjust = | x | y |
|---|---|---|
| | -1.5 | 0.4 |
| | 0 | 1.4 |
| | -2 | -0.1 |
| | -1 | -0.6 |
| | -0.5 | -1.1 |

**Table 1.** PCA example data

### 2.1.2. Calculate The Covariance Matrix

The covariance matrix of new data set is calculated in this step. Since the data is 2 dimensional, the covariance matrix (Equation 2) will be 2×2. The covariance matrix are computed using the cov function in Matlab. Cov (DataAdjust) returns the covariance matrix of new data set. The formula for covariance is:

$$\operatorname{cov}(x,y) = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{n-1} \tag{1}$$

where $n$ is a number of inputs. The covariance matrix is given Equation 2.

$$\operatorname{cov} = \begin{pmatrix} \operatorname{cov}(x,x) & \operatorname{cov}(x,y) \\ \operatorname{cov}(y,x) & \operatorname{cov}(y,y) \end{pmatrix} \tag{2}$$

$$\operatorname{cov} = \begin{pmatrix} .6250 & .1875 \\ .1875 & .9250 \end{pmatrix}$$

### 2.1.3. Calculate The Eigenvectors And Eigenvalues Of The Covariance Matrix

After the covariance matrix is calculated, the eigenvectors and eigenvalues are calculated for this matrix. These are rather important, because this information is useful for us about the data. Many algorithms are improved for calculating the eigenvectors and eigenvalues. The eigenvalues and eigenvectors are computed using the eig function in Matlab. [V,D] = eig(cov) produces matrices of eigenvalues (D) and eigenvectors (V) of matrix cov.

$$D(eigenvalues) = \begin{pmatrix} 0.5349 \\ 1.0151 \end{pmatrix} \qquad V(eigenvectors) = \begin{pmatrix} -.9013 & .4332 \\ .4322 & .9013 \end{pmatrix}$$

### 2.1.4. Choosing Components And Forming A Feature Vector

After eigenvectors are found from the covariance matrix, the next step is to order them by eigenvalue, highest to lowest. This gives us the components in order of significance. The components of lesser significance are ignored. Therefore some information can be lost. But if the eigenvalues are small, information can not be lost much. If some components are left out, the final data set will have less dimensions than the original. To be precise, if you originally have *n* dimensions in your data, and so you calculate *n* eigenvectors and eigenvalues, and then you choose only the first *p* eigenvectors, then the final data set has only *p* dimensions. What needs to be done now is you need to form a feature vector, which is just a fancy name for a matrix of vectors. This is constructed by taking the eigenvectors.

$$FeatureVector = (eig_1, eig_2, eig_3, ..., eig_n) \tag{3}$$

Given the example set of data, and the fact that there are two eigenvectors, there are two choices. It can be either form a feature vector with both of the eigenvectors:

$$(eig_1, eig_2) = \begin{pmatrix} .4332 & -.9013 \\ .9013 & .4332 \end{pmatrix}$$

or, it can be chosen to leave out the smaller, less significant component and only have a single column:

$$eig_1 = \begin{pmatrix} .4332 \\ .9013 \end{pmatrix}$$

### 2.1.5. Deriving The New Data Set

The final step in PCA is to derive the new data set, and is also the easiest. Once the components (eigenvectors) are chosen and formed a feature vector, the transpose of the vector is taken and multiplied it on the left of the original data set, transposed.

FinalData = RowFeatureVector × RowDataAdjust (4)

where *RowFeatureVector* is the matrix with the eigenvectors in the columns transposed so that the eigenvectors are now in the rows, with the most significant eigenvector at the top, and *RowDataAdjust* is the mean-adjusted data transposed.

*FinalData* is the final data set, with data items in columns, and dimensions along rows. The data has been changed from being in terms of the axes *x* and *y*, and now they are in terms of two eigenvectors (Smith, 2002).

| *x* | *y* |
|---|---|
| -.2893 | 1.5252 |
| 1.2618 | .6065 |
| -.9565 | 1.7593 |
| -.9740 | .6414 |
| -1.2080 | -.0259 |

**Table 2.** Row Data

If the dimensionality is reduced, obviously, when reconstructing the data those dimensions chosen to discard are lost. In this example, it is assumed that the *x* dimension is only considered (Smith, 2002).

## 3. PCA WITH NEURAL NETWORKS

For principal component networks, the neural interactions are modeled as a simplified linear computational unit as shown in Figure 1. The output value $y \in \Re$ is linearly related to the synaptic weights and the input as:

$$y = W^T x \tag{5}$$

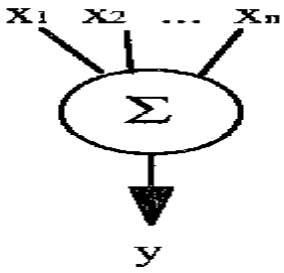


**Figure 1.** Simplified Linear Neuron

Most PCA Neural Networks use some form of Hebbian learning. For example, General Hebbian Algorithm and Adaptive Principle Component Analysis Extrator are used Hebbian Learning.

### 3.1. Hebbian Learning (Oja's Rule)

If there are two units (neurons) *A* and *B*, and there is a connection (synapse) between the units, adjust the strength of the connection (weight *W*) in proportion to the product of their activations (Ziyad, Gilmore and Chouikha, 1998). If *x* denotes the input excitation and *y* denotes the output of the neuron, then the synaptic weights *W* are updated.

A simple Hebbian rule, as known as Oja's rule for simple principal component extraction, in its simplest form is:

$$W(n+1) = W(n) + h\,y(n)\,X(n) \tag{6}$$

where $\eta$ is the learning rate.

The simplest way to stabilize Hebb's rule is to normalize the weights after every iteration.

$$W(n+1) = W(n) + h\,(x(n) - y(n)W(n))\,y(n) \tag{7}$$

So, Oja's rule will give the first eigenvector for a small enough step size. This is the first learning algorithm for PCA (Gleich, 2002).

How do other eigenvectors are computed?

Deflation procedure is adopted to compute the other eigenvectors.

**Step 1:** Find the first principal component using Oja's rule

**Step 2:** Compute the projection of the first eigenvector on the input

$$y = W_1^T x \tag{8}$$

**Step 3:** Generate the modified input as

$$\hat{x} = x - W_1 y = x - W_1 W_1^T x \tag{9}$$

**Step 4:** Repeat Oja's rule on the modified data

The steps can be repeated the generate all the eigenvectors.

### 3.2. General Hebbian Algorithm (GHA)

Sanger proposed using Oja's rule as the basis for extending the Hebbian learning approach for the extraction of multiple principal components, which is known as the General Hebbian Algorithm (GHA) (Diamantaras and Kung, 1996). This model has $M$ inputs, $L$ outputs, and uses feedforward weights between the input and output, where $y \in \Re^L$, $x \in \Re^M$, $W \in \Re^{LxM}$. Each output $y_i$ corresponds to the output of the *ith* principal component neuron and is a linear function of the input,

$$y_i = W_i^T x \tag{10}$$

**Properties of GHA Model**

- Oja's rule + Deflation = Sanger's rule (GHA).
- Sanger extended Oja's rule to extract multiple eigenvectors using deflation procedure.
- The rule is simple to implement.
- The algorithm is on-line.

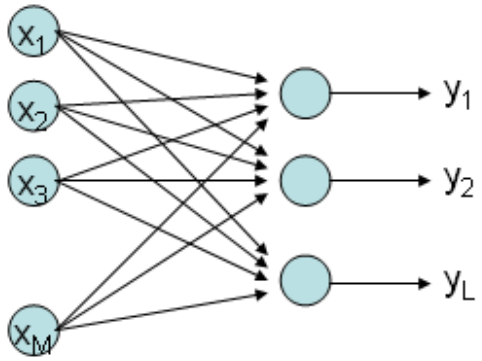


**Figure 2.** The schematic projection of GHA model

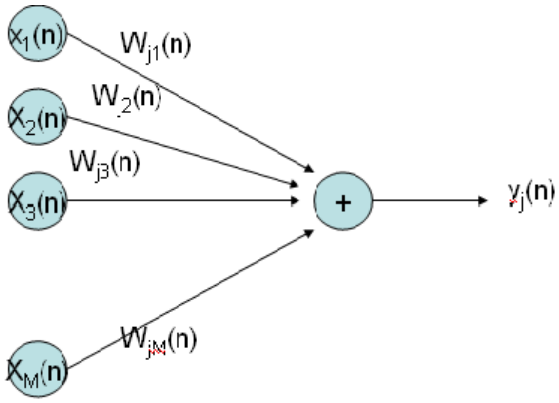


**Figure 3.** Another schematic projection of GHA model

$W_{ji}$ represents the scalar weight between input node $i$ and output node $j$. Network has $M$ inputs and $L$ outputs.

$$y_j(n) = \sum_{i=1}^{m} W_{ji}(n) X_i(n) \quad \text{i=1,2,3,4,….M} \qquad \text{j=1,2,3,4,….L} \tag{11}$$

$$\Delta W_{ji}(n) = h\left[ y_j(n) x_i(n) - y_j(n) \sum_{k=1}^{j} W_{ki}(n) y_k(n) \right] \tag{12}$$

i = 1,2,3,4,….M j = 1,2,3,4,….L

- Oja's rule is the the basic learning rule for PCA and extracts the first principal component.
- Deflation procedure can be used to estimate the minor eigencomponent.
- Sanger's rule does an on-line deflation and uses Oja's rule to estimate the eigencomponents.

**Disadvantanges of Sanger's Rule**

- Sanger's rule is non-local.
- Sanger's rule converges slowly than APEX Algorithm.

### 3.3. Adaptive Principal Component Extrator (APEX)

The APEX network, proposed by Diamantaras (Diamantaras and Kung, 1996), is a technique for extracting multiple principal components which uses a lateral connection network topology trained via Oja's simple Hebbian rule. The lateral connections work towards the orthogonalization of the synaptic weights of the $m_{th}$ neuron versus the extracted principal components stored in the weights of the previous *m-1* neurons. This type of network topology allows the network model to increase or decrease in size without the need of retraining the past neurons.

**Properties of APEX Model**

- Input vector *x* and output vector *y* is given by

$$y = Wx + Cy \tag{13}$$

$$y_1 = W_1^T x \tag{14}$$

$$y_2 = W_2^T x + C y_1 \tag{15}$$

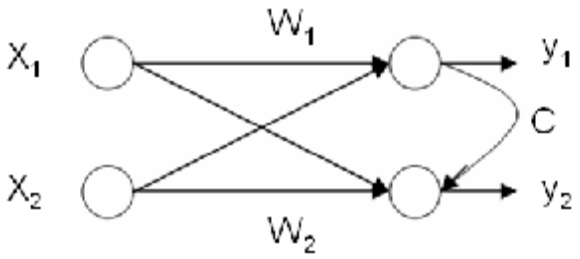


**Figure 4.** The schematic projection of APEX model

- $C$ is a lower triangular matrix and this is usually called as the lateral weight matrix or the lateral inhibitor matrix.
- Feedforward weights $W$ are trained using Oja's rule.
- Lateral weights are trained using

$$\Delta C_{ij}(n+1) = -h y_i(n) y_j(n) \tag{16}$$

where $\eta$ is the learning rate.

### 3.4. Expectation-Maximization Principle Component Analysis

The EM algorithm is used for finding the Maximum Likelihood estimate of the parameters of a dataset when there are missing values in the data.

There are two main applications of the EM algorithm

- When the data indeed has incomplete, missing or corrupted values as a result of a faulty observation process.
- When assuming the existence of missing or hidden parameters can simplify the likelihood function.

EM algorithm consists of two steps: Expectation step (E-step) and Maximization (M-step). The E-step is based on the expected value of the data. The M-step is

maximized log-likelihood function to give revised parameter estimation based on sufficient statistics calculated in E-step.

Principal component analysis (PCA) is a widely used dimensionality reduction technique in data analysis. But PCA models have several shortcomings (Roweis, 1998). One is that some methods for finding the principal components have trouble with high dimensional data. The sample covariance matrix of $n$ vectors in a space of $p$ dimensions when $n$ and $p$ are several hundred or several thousand is very costly. This is requiring $O(np^2)$ operations. The expectation-maximization (EM) algorithm (Dempster, Laird, and Rubin, 1977) for learning the principal components of a dataset does not require computing the sample covariance and has a complexity limited by $O(knp)$ operations. $k$ is the number of leading eigenvectors.

Another shortcoming of PCA is that it is not obvious how to deal with missing data. The EM algorithm for PCA uses EM algorithms for estimating the maximum likelihood values for missing information directly at each iteration (Ghahramani and Jordan, 1994).

The key observation of this note is that even though the principal components can be computed explicitly, there is still an EM algorithm for learning them. It can be easily derived as the zero noise limit of the standard algorithms by replacing the usual E-step with the projection above. The algorithm is:

$$E-step: X = \left(V^T V\right)^{-1} V^T Y \tag{17}$$

$$M-step: V^{new} = YX^T (XX^T)^{-1} \tag{18}$$

where $Y$ is a $pxn$ matrix of all the observed data and $X$ is a $kxn$ matrix of unknown states. The columns of $V$ is the space of the first $k$ principle component.

The algorithm can be performed online using only a single datapoint at a time and so its storage requirements are only $O(kp)+O(k^2)$.

The EM learning algorithm for PCA amounts to an iterative procedure for finding the subspace spanned by the $k$ leading eigenvectors without explicit computation of the sample covariance. It is attractive for small $k$ because its complexity is limited by $O(knp)$

per iteration and so depends only linearly on both the dimensionality of the data and the number of points. The methods that explicitly compute the sample covariance matrix have complexities limited by $O(np^2)$ while methods like the snap-shot method that form linear combinations of the data must compute and diagonalize a matrix of all possible inner products between points and thus are limited by $O(n^2 p)$ complexity. As expected, the EM algorithm scales more favourably in cases where k is small and both p and n are large. If $k \approx p \approx n$ then all methods are $O(p^3)$.

The method has some advantages. It allows simple and efficient computation of a few eigenvectors and eigenvalues when working with high dimensional data. It permits this computation even in the presence of missing data.

### 3.5. Probabilistic Principal Components Analysis (PPCA)

PCA is merely a rotation of an *n*-dimensional data space and selection of m dimensions in the rotated space as the new *m*-dimensional linear subspace. If the data in the original space is Gaussian then the data in the rotated subspace is also Gaussian. Therefore, PPCA is a Gaussian modeller that defines the relation between the Gaussians in the original space and the subspace ( Zhao, Chai and Cong, 2006). The generative model:

$$t = Wx + m + e \tag{19}$$

specifies the relation between these two Gaussians where *t* (*n*-dimensional) is the data vector, *x* (*m*-dimensional) is the subspace vector, *W* are the *m* dominant eigenvectors (principal components, or PCs), μ is the data mean, and $e$ is a noise model which is assumed isotropic Gaussian (i.e. $e$ ~ N (0, I)) approximating the average of the minor eigenvalues .

### 3.6. True-PCA

This method is traditional PCA. Usual way is used to do PCA in this method. By finding the eigenvalues and eigenvectors of the covariance matrix, the eigenvectors are found that with the largest eigenvalues correspond to the dimensions that have the strongest correlation in the dataset. The original measurements are finally projected onto the reduced vector. The eigenvectors are calculated using function *eig* in Matlab. Also, the covariance matrix are calculated using function *cov* in Matlab.

# 4. IMAGE CLUSTERING

Image clustering is used for high-level description of image content. The goal is to find a mapping of the archive images into classes provide essentially the same information about the image. The generated classes provide a concise summarization and visualization of the image content that can be used for different tasks related to image database management. Image clustering can be a useful tool when dealing with gray scale images (Parekh and Herling, 2003). Image clustering enables the implementation of efficient retrieval algorithms (Goldberger, Greenspan and Gordon, 2002).

Clustering analysis is based on partitioning a collection of data points into a number of clusters, where the objects inside a cluster show a certain degree of closeness or similarity. It has been playing an important role in solving many problems in pattern recognition and image processing. Clustering methods can be considered as either hard or fuzzy depending on whether a pattern data belongs exclusively to a single cluster or to several clusters with different degrees. In hard clustering, a membership value of zero or one is assigned to each pattern data (feature vector), whereas in fuzzy clustering, a value between zero and one is assigned to each pattern by a membership function. In general, fuzzy clustering methods can be considered to be superior to that of its hard counterparts since they can represent the relationship between the input pattern data and clusters more naturally. Clustering algorithms such as K-Means, as known Hard C-Means, and Fuzzy C-Means (FCM) are based on the sum of intracluster distances criterion (Kaya, 2005).

## 4.1. K-Means Algorithm

K-Means clustering is the most widely used clustering algorithm. The purpose of K-Means clustering is to classify the data (Teknomo, 2006). K-Means clustering is one of the simplest unsupervised clustering algorithms (Arefi, Hahn, Samadzadegan, Lindenberger, 2005). There are two versions of K-Means clustering, a non-adaptive version and an adaptive version. The non-adaptive version was introduced by Lloyd (Lloyd, 1957). But the adaptive version was introduced by MacQueen (MacQueen,1967). The most commonly used K-Means clustering is the adaptive K-Means clustering based on the Euclidean distance (Darken and Moody,1990). K-Means

clustering algorithm can be sensitive to the initial centers and the search for the optimum center locations. For K-Means clustering algorithm, it is assumed that the initial centers are provided. The search for the final clusters or centers starts from these initial centers (Mashor, 1998).

The centers should be selected to minimize the total distance between the data and the centers for the centers can perform the data. A simple and widely used square error cost function is used to measure the distance, which is defined as:

$$E = \sum_{j=1}^{n_c} \sum_{i=1}^{N} \left( \left\| x_i - c_j \right\| \right)^2 \tag{20}$$

where $N$, and $n_c$ are the number of data and the number of centers respectively; $x_i$ is the data sample belonging to center $c_j$.

During the clustering process, the centers are adjusted according to the total distance in the Equation 20 is minimized. K-Means clustering tries to minimize the cost function by searching for the center $c_j$ on-line as the data are presented. As the data sample is presented, the Euclidean distances between the data sample and all the centers are calculated and the nearest center is updated according to:

$$\Delta c_z(t) = h(t)\left[v(t) - c_z(t-1)\right] \tag{21}$$

where $z$ indicates the nearest center to the data $v(t)$. Notice that, the center and the data are written in terms of time $t$ where $c_z(t-1)$ represents the center location at the previous clustering step.

The adaptation rate, $\eta(t)$, can be selected in a number of ways. The problem of assigning the adaptation rate to adaptive K-Means clustering is very similar to the problem of assigning the learning rate to the back propagation algorithm. Therefore, all the methods that are used to choose the learning rate for the back propagation algorithm may also be applied for the adaptation rate in K-Means clustering. The usual approach is to update $\eta(t)$ according to the variation of the cost function during the clustering process (Hertz, Krogh and Palmer, 1991).

$$\Delta\eta(t) = \begin{cases} +a & \text{if } \Delta E < 0 \text{ consistently} \\ -b\eta(t-1) & \text{if } \Delta E > 0 \\ 0 & \text{otherwise} \end{cases} \quad (22)$$

where $\Delta E$ is the change in the cost function and, $a$ and $b$ are parameter constants.

**Advantages of K-Means Method**

- With a large number of variables, K-Means may be computationally faster than hierarchical clustering (if K is small).
- K-Means may produce tighter clusters than hierarchical clustering, especially if the clusters are globular.

**Disadvantages of K-Means Method**

- Difficulty in comparing quality of the clusters produced (e.g. for different initial partitions or values of K affect outcome).
- Fixed number of clusters can make it difficult to predict what K should be.
- Does not work well with non-globular clusters.
- Different initial partitions can result in different final clusters. It is helpful to return the program using the same as well as different K values, to compare the results achieved.

### 4.2. Fuzzy C-Means Algorithm

Fuzzy C-Means (FCM) is developed by Dunn (Dunn, 1974) and improved by Bezdek (Bezdek,1981). This method is usually used in pattern recognition (Albayrak and Amasyalı, 2003).

Fuzzy C-Means Clustering is an clustering technique which is separated from K-Means. Because the K-Means clustering uses hard partitioning. But the Fuzzy C-Means uses fuzzy partitioning. That is to say a data point can belong to all groups with different degrees (Berks, etc. 2000).

The aim of FCM is to find cluster centers that minimize the a dissimilarity function (Albayrak, Amasyalı, 2003). First, for the introduction of fuzzy partitioning, the

membership matrix(U) is randomly initialized according to Equation 23. The dissimilarity function which is used in FCM is given Equation 24.

$$\sum_{i=1}^{c} u_{ij} = 1 \quad , \quad \forall j = 1,...,n \tag{23}$$

$$J(U, c_1, c_2, ..., c_c) = \sum_{i=1}^{c} J_i = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^{m} d_{ij}^{2} \tag{24}$$

where $u_{ij}$ is between 0 and 1, $d_{ij} = \|c_i - x_j\|$ is the Euclidean dinstance between *ith* cluster center anf *jth* data point, *c* is the number of clusters, $c_i$ is cluster center of fuzzy cluster *i*, *n* is the number of data point and $m \in [1, \infty)$ is weighting exponent.

There are two conditions for reaching a minimum of dissimilarity function These are the following Equation 25 and Equation 26.

$$c_i = \frac{\sum_{j=1}^{n} u_{ij}^{m} x_j}{\sum_{j=1}^{n} u_{ij}^{m}} \tag{25}$$

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left( \frac{d_{ij}}{d_{kj}} \right)^{2/(m-1)}} \tag{26}$$

Fuzzy C-means algorithm determines the following steps (Jang, Sun and Mizutani,1997).

**Step 1.** Randomly initialize the membership matrix (*U*) that has constraints in Equation 23.

**Step 2.** Calculate centers ($c_i$) by using Equation 25.

**Step 3.** Compute dissimilarity between centers and data points using Equation 24. Stop if its improvement over previous iteration is below a threshold.

**Step 4.** Compute a new *U* using Equation 26. Go to Step 2.

By updating the cluster centers and the membership grades for each data point, FCM iteratively moves the cluster centers to the right location within a data set.

FCM doesn't always converge to an good solution. Because of cluster centers are initialize using *U*. Solution depends on initial centers. For a strong approach, there are two ways below.

- Using an algorithm to determine all of the centers. (for example: arithmetic means of all data points)
- Run FCM several times each starting with different initial centers.

The important feature of Fuzzy C-Means algorithm is membership function and an object can belong to several classes at the same time but with different degrees of belongingness.

## 5. EXPERIMENTAL RESULTS

MRI image that was taken from Çukurova University Medical Hospital was used in the thesis. The MRI image is taken from the patient who has brain tumor. The original image has 512 lines and 512 columns. First, the dimensions of original image was reduced. Then PCA methods were applied to the resized image.

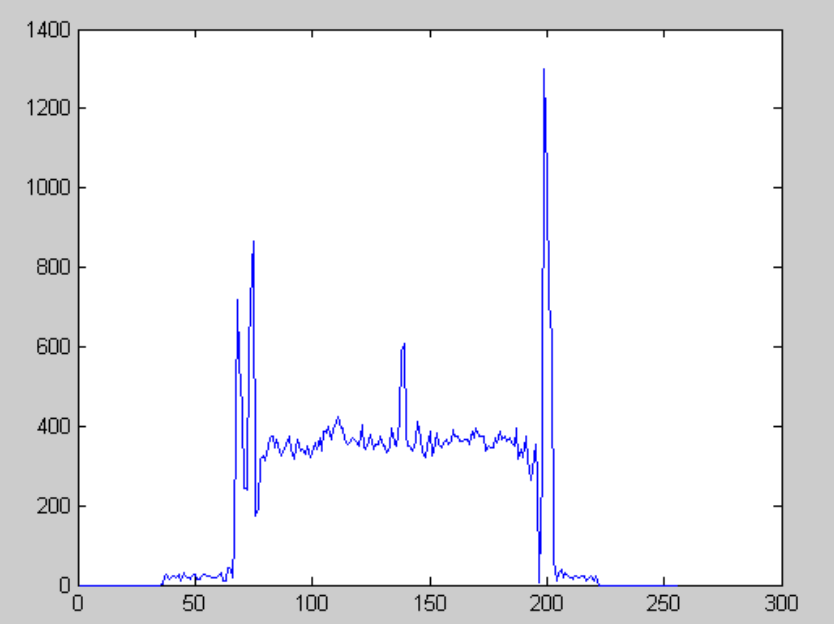


**Figure 5.** The histogram of orijinal image

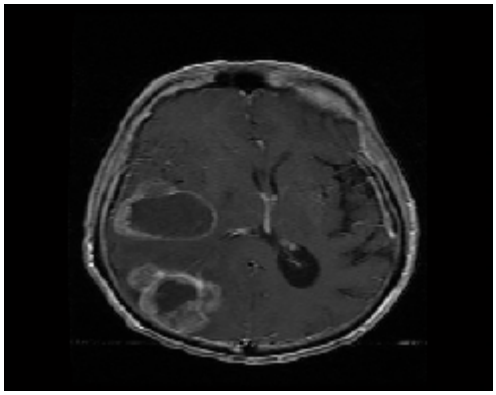

**Figure 6.a.**Original MRI Image with 512*512

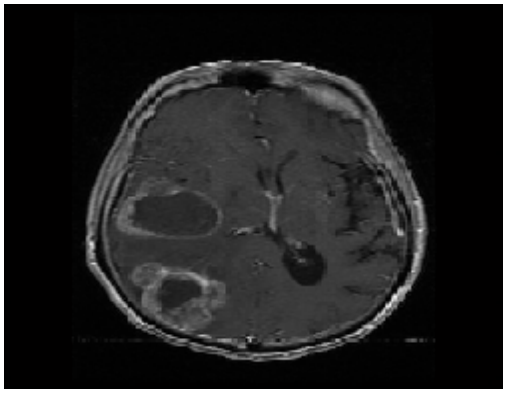

**Figure 6.b.** Resized MRI Image has 256 lines and 256 columns

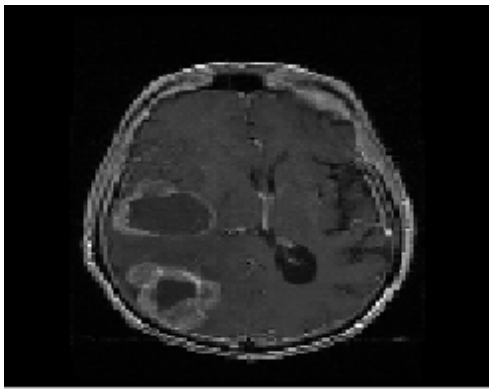

**Figure 6.c.** Resized MRI Image has 128 lines and 128 columns.

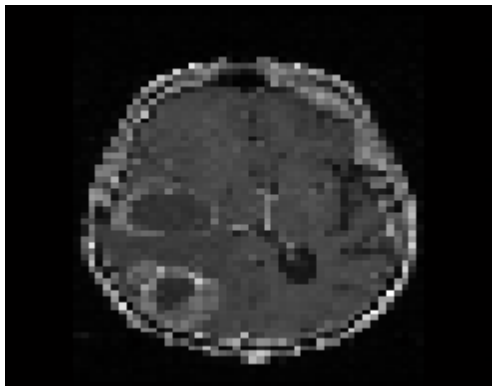

**Figure 6.d.** Resized MRI Image has 64 lines and 64 columns.

Below, the resized MRI images are used with 256*256, 128*128 and 64*64 and the original image with 512*512. In the beginning, PCA methods were implemented on the MRI image. When PCA methods were implemented, different window-sizes were used. Window-size is determined as 8, when the images at the below were obtained. First, the dimensions of the MRI image was reduced by using PCA method without much loss of information.

**EM-PCA Test Results**

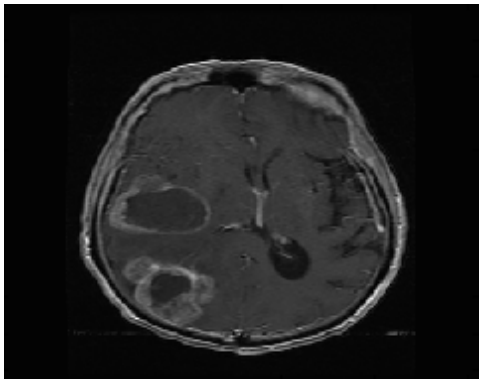

**Figure 7.a.** The result of EM-PCA on the MRI image (512*512)

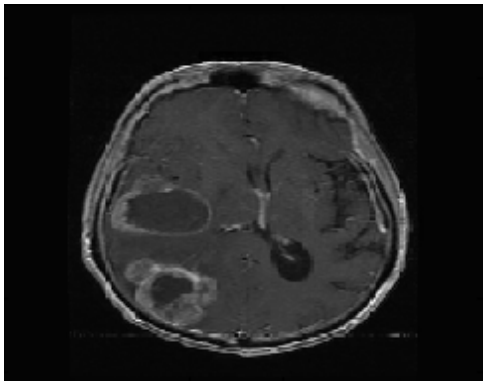

**Figure 7.b.** The result of EM-PCA on the MRI image (256*256)

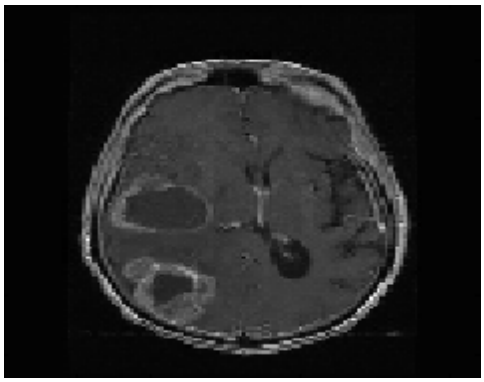

**Figure 7.c.** The result of EM-PCA on the MRI image (128*128)

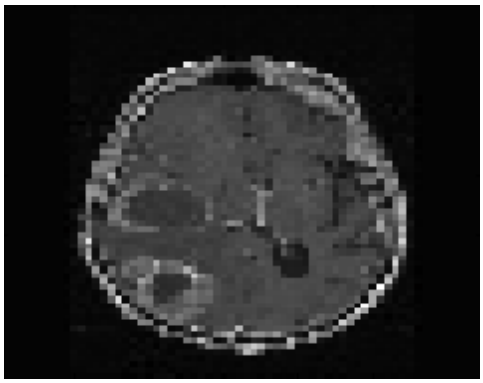

**Figure 7.d.** The result of EM-PCA on the MRI image (64*64)

**Probabilistic PCA Test Results**

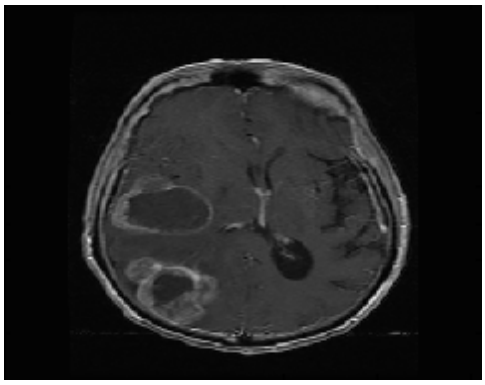

**Figure 8.a.** The result of PPCA on the MRI image (512*512)

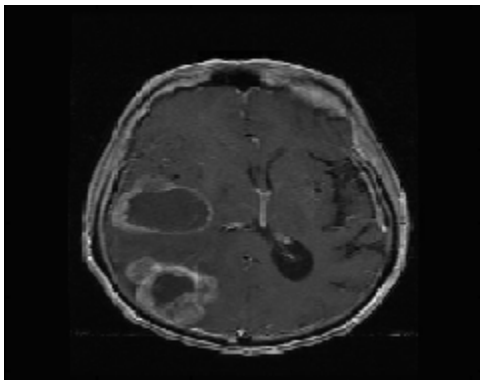

**Figure 8.b.** The result of PPCA on the MRI image (256*256)

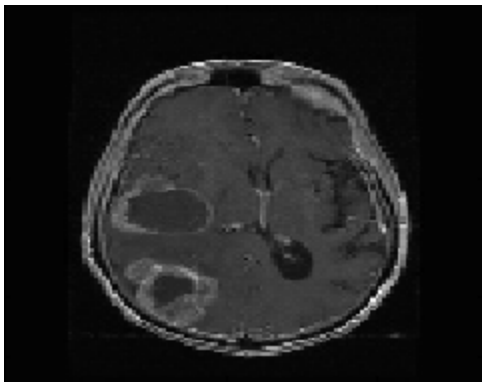

**Figure 8.c.** The result of PPCA on the MRI image (128*128)

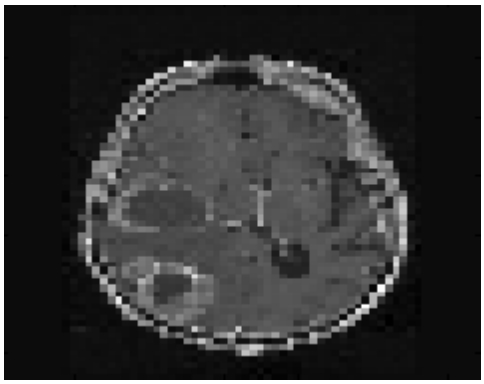

**Figure 8.d.** The result of PPCA on the MRI image (64*64)

**APEX Results**

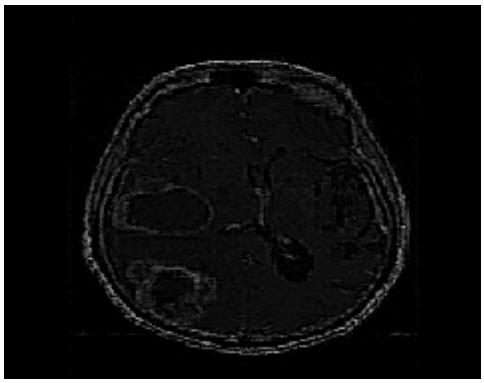

**Figure 9.a.** The result of APEX on the MRI image (512*512)

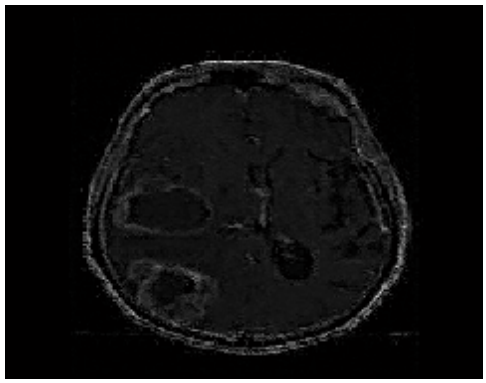

**Figure 9.b.** The result of APEX on the MRI image (256*256)

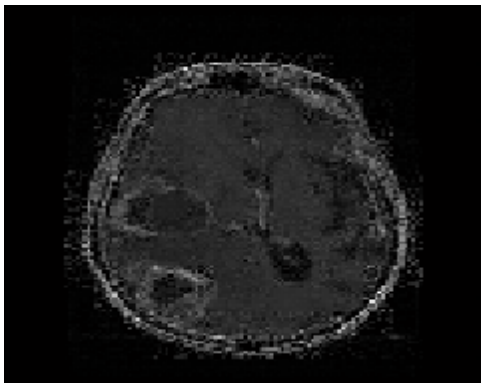

**Figure 9.c.** The result of APEX on the MRI image (128*128)

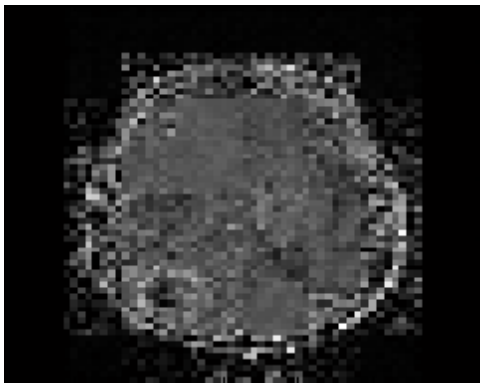

**Figure 9.d.** The result of APEX on the MRI image (64*64)

**GHA Results**

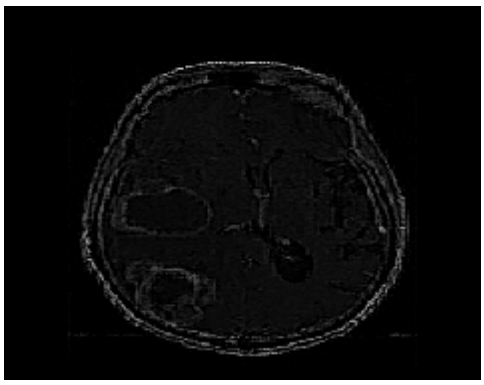

**Figure 10.a.** This image is the result of GHA on the MRI image (512*512)

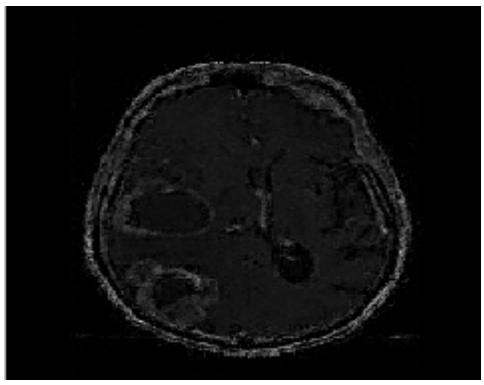

**Figure 10.b.** This image is the result of GHA on the MRI image (256*256)

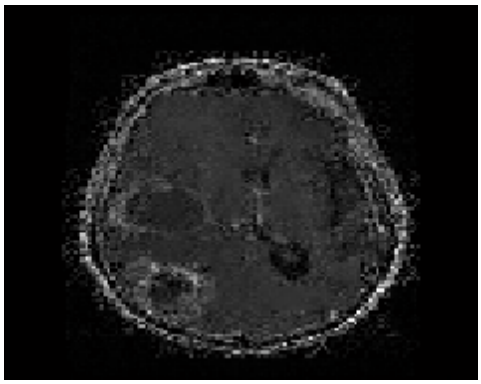

**Figure 10.c.** This image is the result of GHA on the MRI image (128*128)

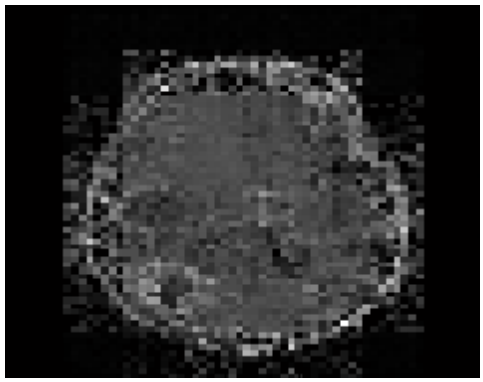

**Figure 10.d.** This image is the result of GHA on the MRI image (64*64)

**True PCA Results**

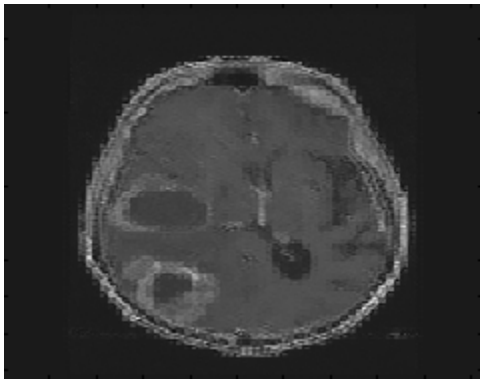

**Figure 11.a.** This image is the result of True PCA on the MRI image (512*512)

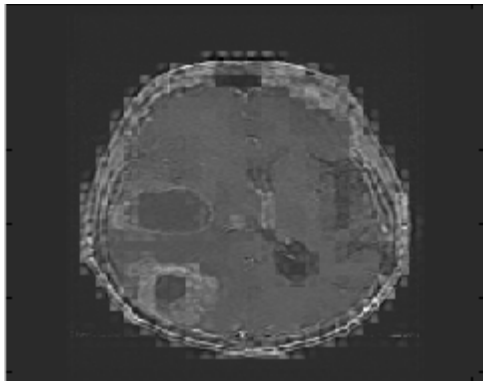

**Figure 11.b.** This image is the result of True PCA on the MRI image (256*256)

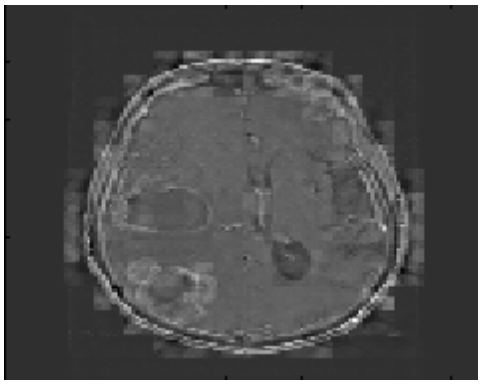

**Figure 11.c.** This image is the result of True PCA on the MRI image (128*128)

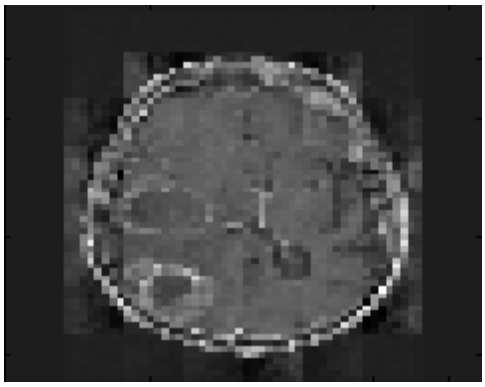

**Figure 11.d.** This image is the result of True PCA on the MRI image (64*64)

According to the observed images and error rates, it was agreed that EM-PCA and PPCA methods were the best methods as regards the others. Because the EM-PCA and PPCA obtain the eigenvectors without explicit computation of the sample covariance and allow simple and efficient computation of a few eigenvectors and eigenvalues when working with high dimensional data.

Also, different window-sizes were used on resized MRI image with 256*256 at the below. According to observed images, it was agreed that better results were obtained when window-size was small. Espacially, True-PCA method obtained very good results when the window-size was small.

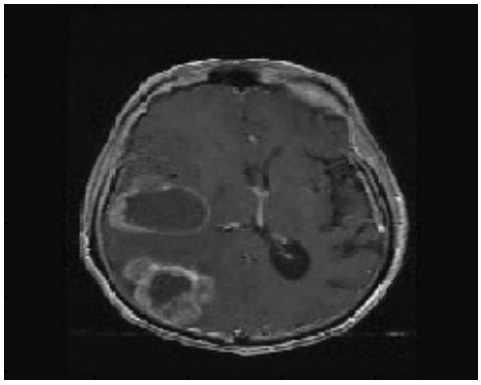

**Figure 12.a.** The result of EMPCA with 256*256 (Windows size=16)

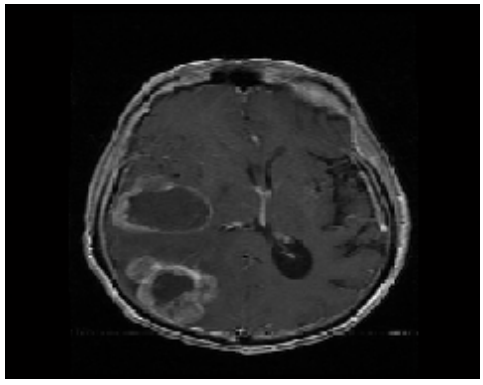

**Figure 12.b.** The result of EMPCA with 256*256 (Windows size=4)

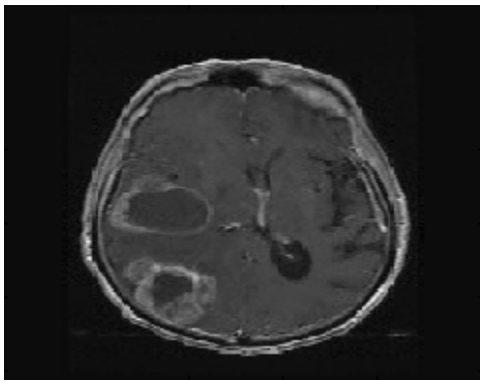

**Figure 13.a.** The result of PPCA with 256*256 (Windows size=16)

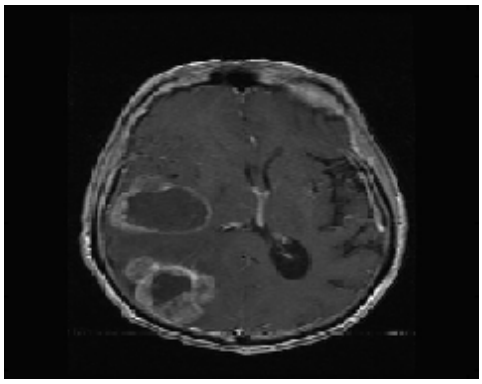

**Figure 13.b.** The result of PPCA with 256*256 (Windows size=4)

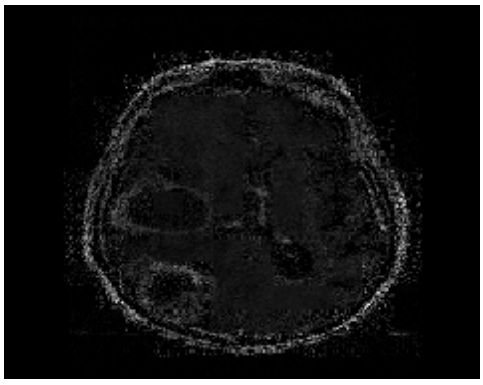

**Figure 14.a.** The result of APEX with 256*256 (Windows size=16)

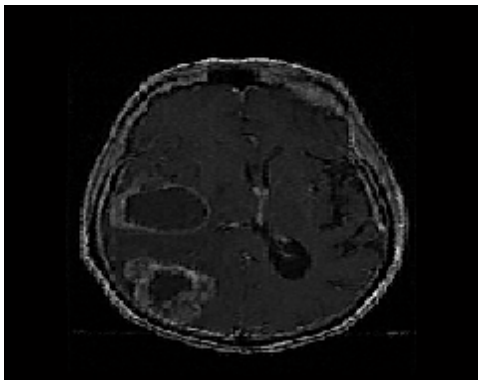

**Figure 14.b.** The result of APEX with 256*256 (Windows size=4)

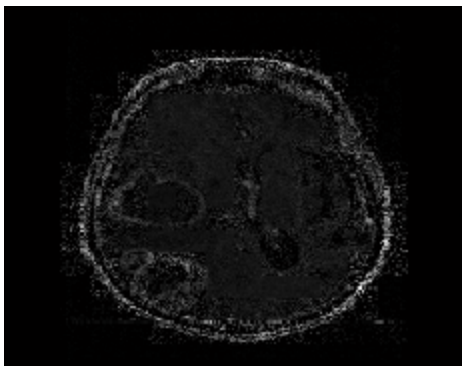

**Figure 15.a.** The result of GHA with 256*256 (Windows size=16)

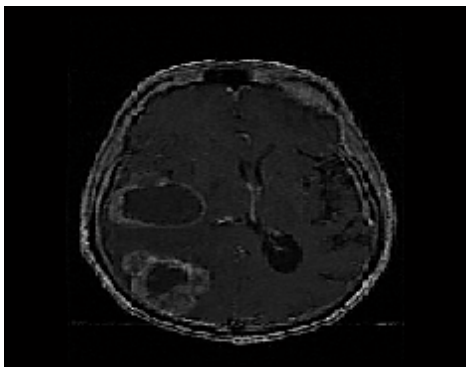

**Figure 15.b.** The result of GHA with 256*256 (Windows size=4)

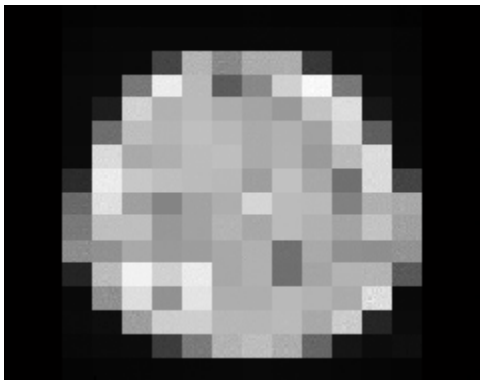

**Figure 16.a.** The result of True-PCA with 256*256 (Windows size=16)

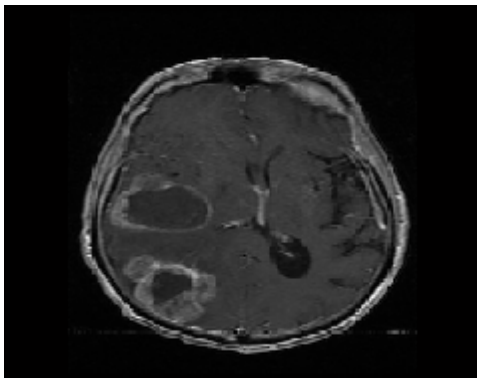

**Figure 16.b.** The result of True-PCA with 256*256 (Windows size=4)

After PCA methods were implemented, K-Means and Fuzzy C-Means image clustering methods were implemented on resized MRI images obtained with PCA methods. According to the histogram of the original image (Figure 5), the number of clusters was determined as 5 in the image clustering methods. Window-size was determined as 8 in the clustering methods. According to results, among image clustering methods Fuzzy C-Means algorithm gives beter results than K-Means algorithm. Because the performance of K-Means algorithm depends on the initial positions of centers. So the algorithm gives no guarantee for an optimum solution. But FCM is an iterative algorithm. The aim of FCM is to find cluster centers that minimize a dissimilarity function Equation 24. Fuzzy C-Means algorithm is separated from K-Means that employs hard partitioning. It employs fuzzy partitioning such that a data point can belong to all groups with the degree of belongingness specified by membership grades between 0 and 1. Row data (512*512) is better than others. Because the data was not lost data. When the data dimensional size is reduced, the results are getting the worst.

**K-Means Results**

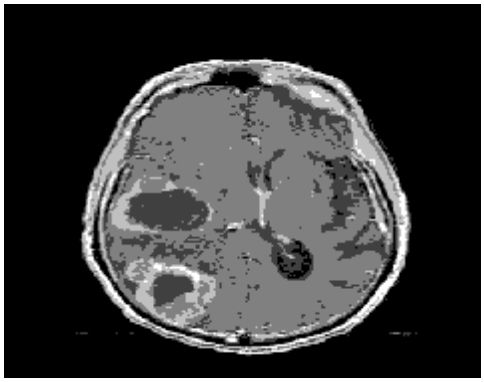

**Figure 17.a.** The result of K-Means on EM-PCA with 512*512

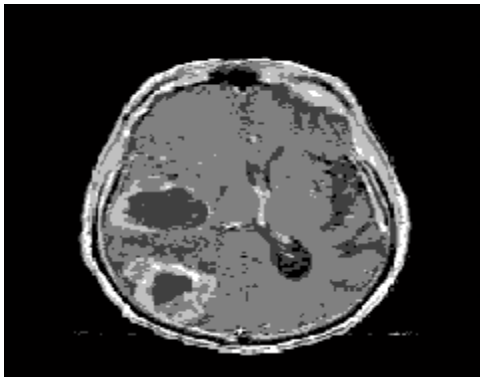

**Figure 17.b.** The result of K-Means on EM-PCA with 256*256

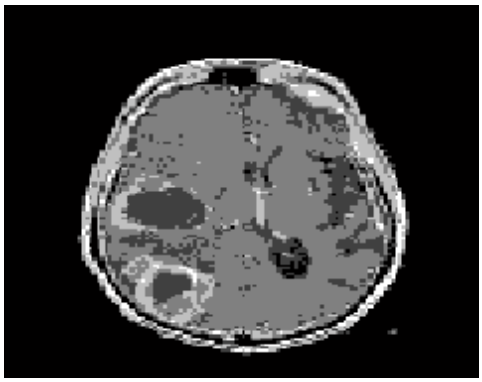

**Figure 17.c.** The result of K-Means on EM-PCA with 128*128

**Figure 17.d.** The result of K-Means on EM-PCA with 64*64

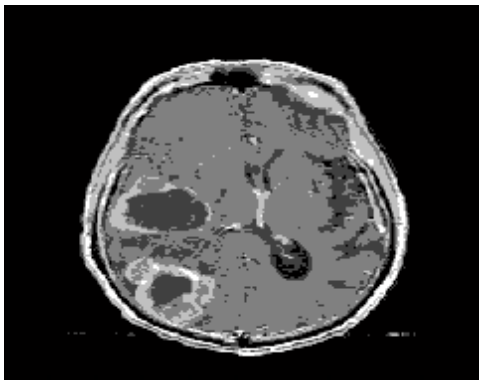

**Figure 18.a.** The result of K-Means on PPCA with 512*512

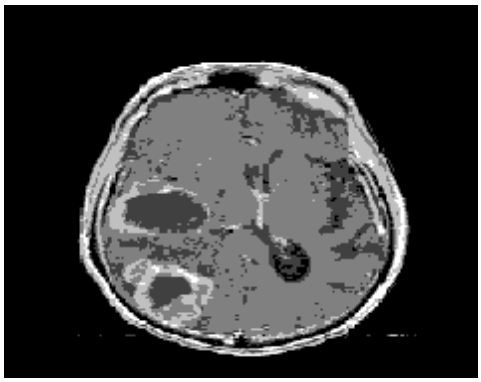

**Figure 18.b.** The result of K-Means on PPCA with 256*256

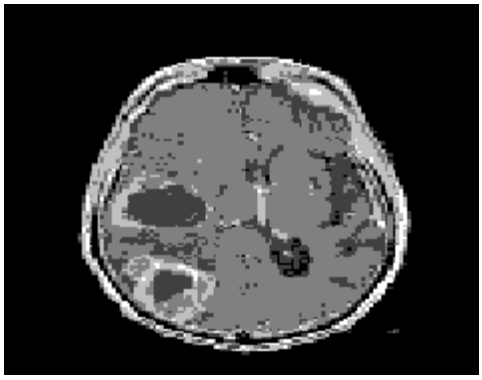

**Figure 18.c.** The result of K-Means on PPCA with 128*128

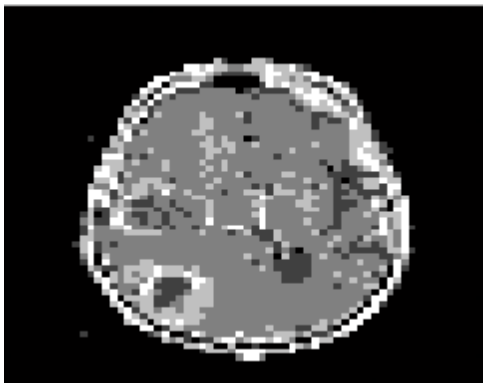

**Figure 18.d.** The result of K-Means on PPCA with 64*64

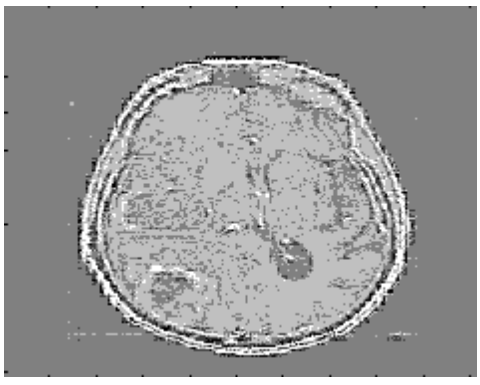

**Figure 19.a.** The result of K-Means on APEX with 512*512

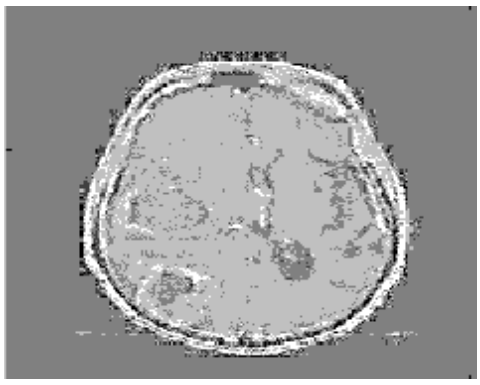

**Figure 19.b.** The result of K-Means on APEX with 256*256

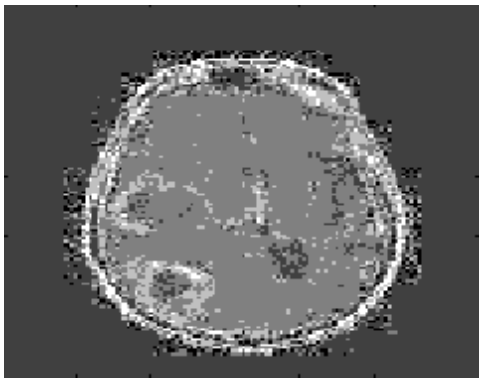

**Figure 19.c.** The result of K-Means on APEX with 128*128

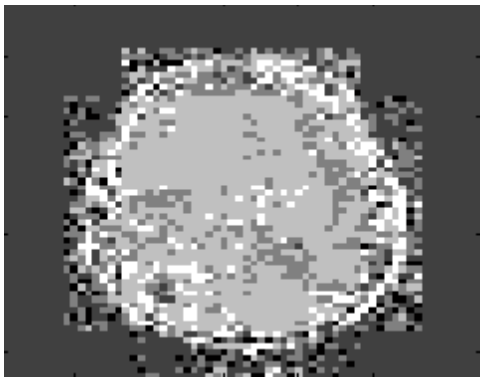

**Figure 19.d.** The result of K-Means on APEX with 64*64

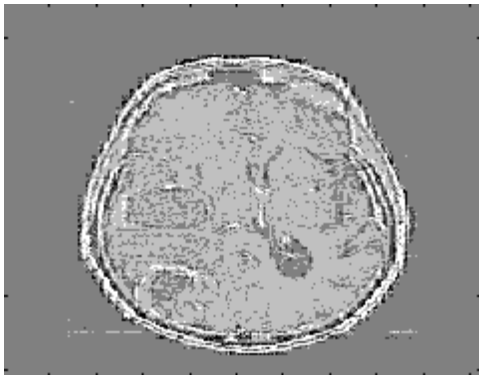

**Figure 20.a.** The result of K-Means on GHA with 512*512

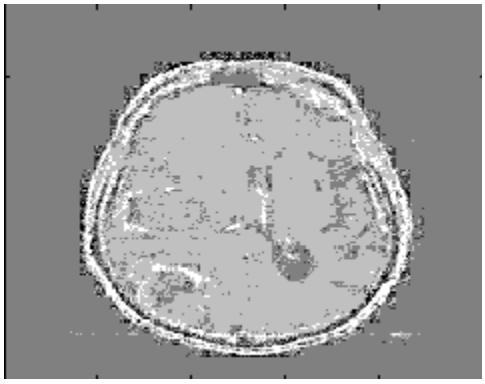

**Figure 20.b.** The result of K-Means on GHA with 256*256

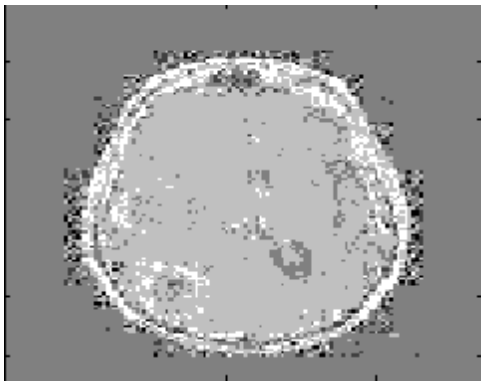

**Figure 20.c.** The result of K-Means on GHA with 128*128

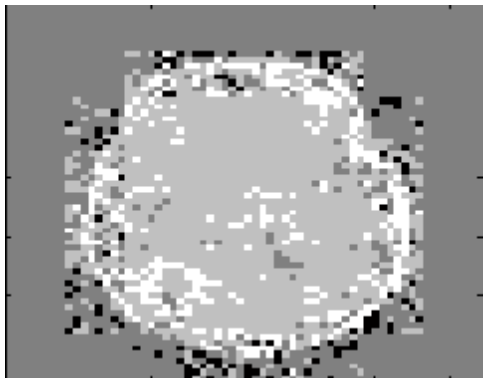

**Figure 20.d.** The result of K-Means on GHA with 64*64

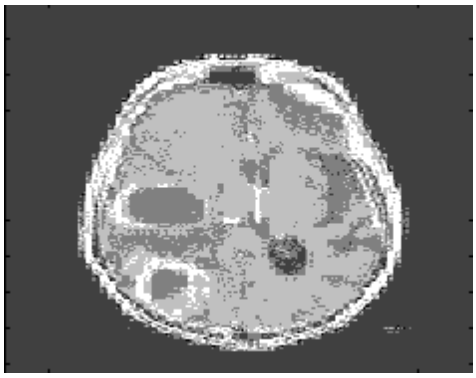

**Figure 21.a.** The result of K-Means on True-PCA with 512*512

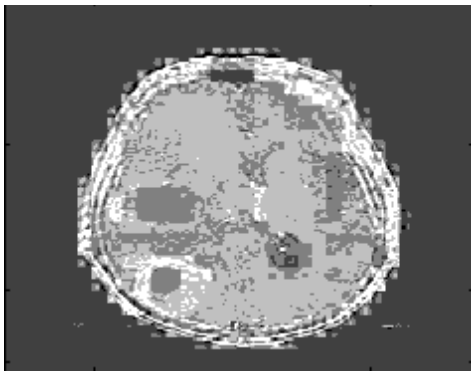

**Figure 21.b.** The result of K-Means on True-PCA with 256*256

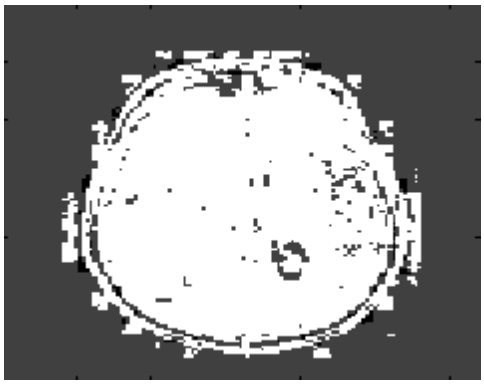

**Figure 21.c.** The result of K-Means on True-PCA with 128*128

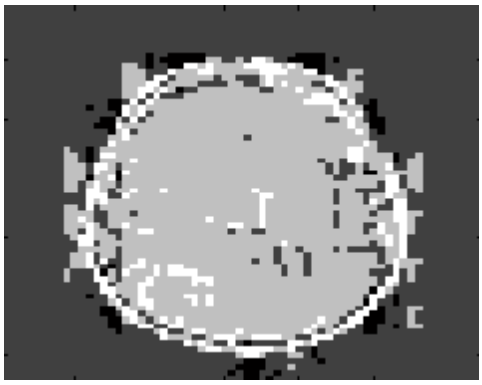

**Figure 21.d.** The result of K-Means on True-PCA with 64*64

**Fuzzy C-Means Results**

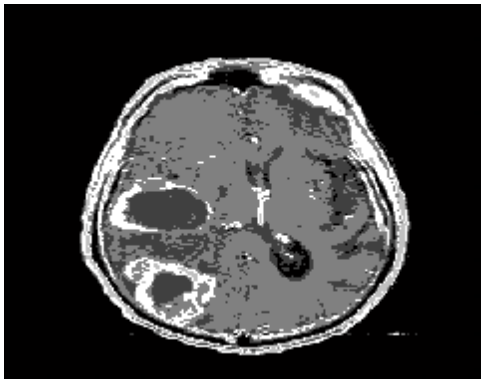

**Figure 22.a.** The result of Fuzzy C-Means on EMPCA with 512*512

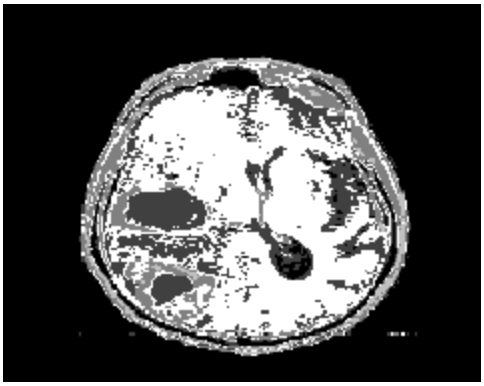

**Figure 22.b.** The result of Fuzzy C-Means on EMPCA with 256*256

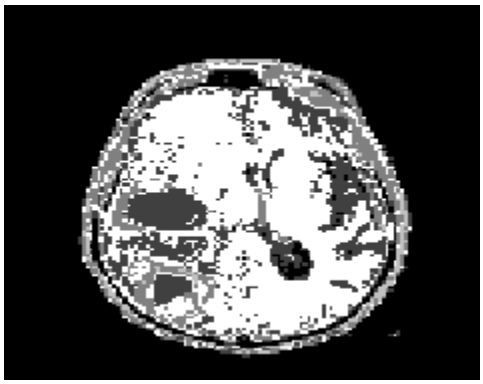

**Figure 22.c.** The result of Fuzzy C-Means on EMPCA with 128*128

**Figure 22.d.** The result of Fuzzy C-Means on EMPCA with 64*64

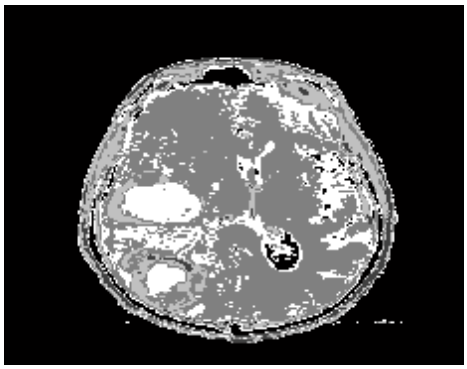

**Figure 23.a.** The result of Fuzzy C-Means on PPCA with 512*512

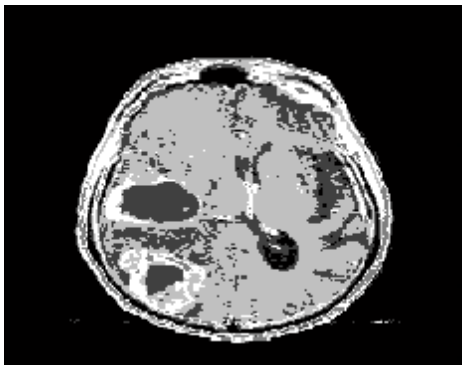

**Figure 23.b.** The result of Fuzzy C-Means on PPCA with 256*256

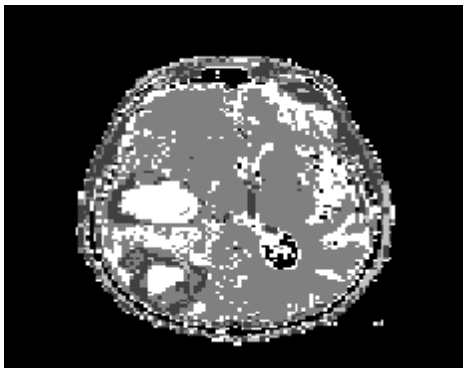

**Figure 23.c.** The result of Fuzzy C-Means on PPCA with 128*128

**Figure 23.d.** The result of Fuzzy C-Means on PPCA with 64*64

**Figure 24.a.** The result of Fuzzy C-Means on APEX with 512*512

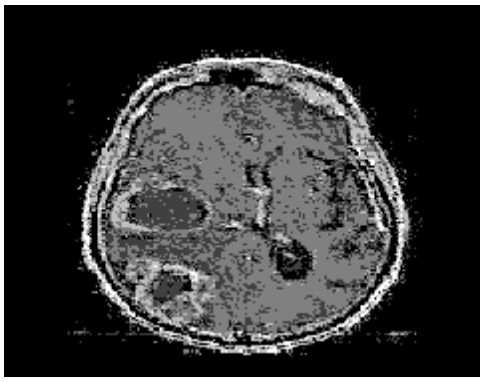

**Figure 24.b.** The result of Fuzzy C-Means on APEX with 256*256

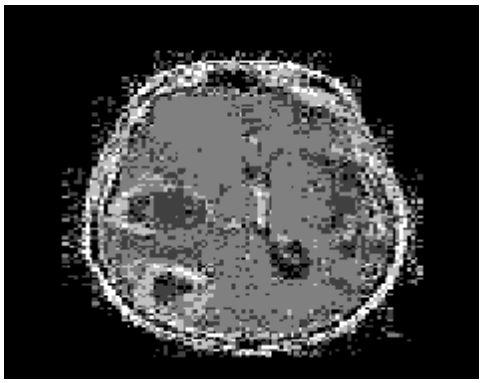

**Figure 24.c.** The result of Fuzzy C-Means on APEX with 128*128

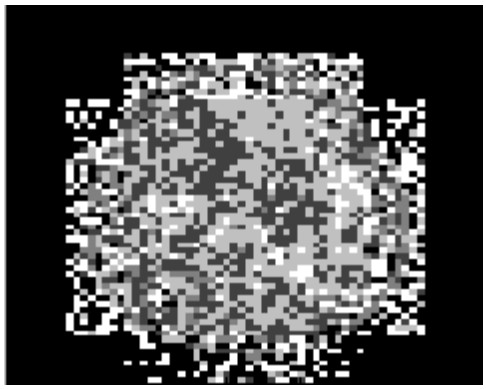

**Figure 24.d.** The result of Fuzzy C-Means on APEX with 64*64

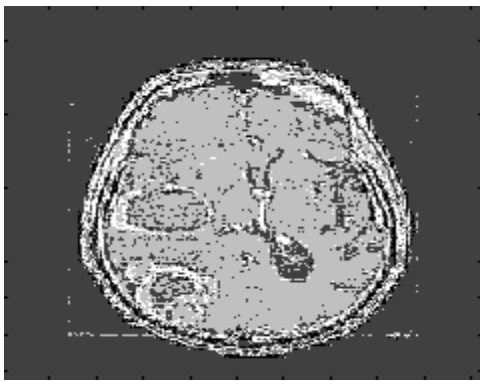

**Figure 25.a.** The result of Fuzzy C-Means on GHA with 512*512

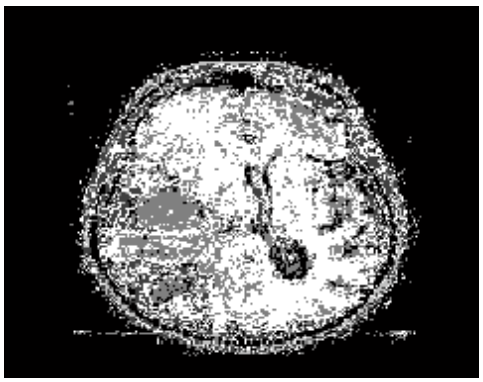

**Figure 25.b.** The result of Fuzzy C-Means on GHA with 256*256

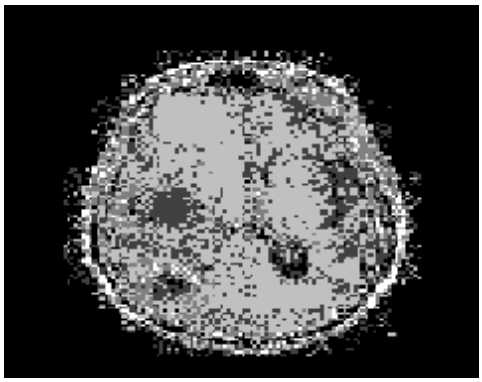

**Figure 25.c.** The result of Fuzzy C-Means on GHA with 128*128

**Figure 25.d.** The result of Fuzzy C-Means on GHA with 64*64

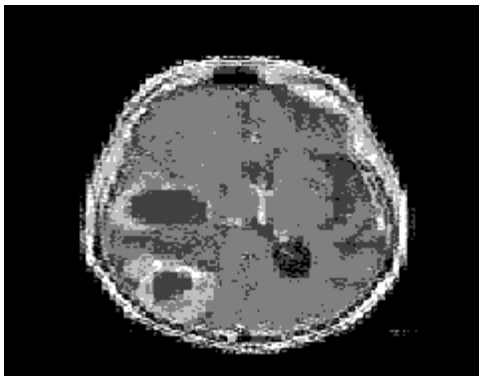

**Figure 26.a.** The result of Fuzzy C-Means on True-PCA with 512*512

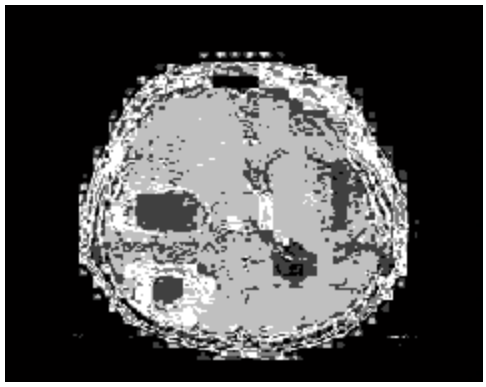

**Figure 26.b.** The result of Fuzzy C-Means on True-PCA with 256*256

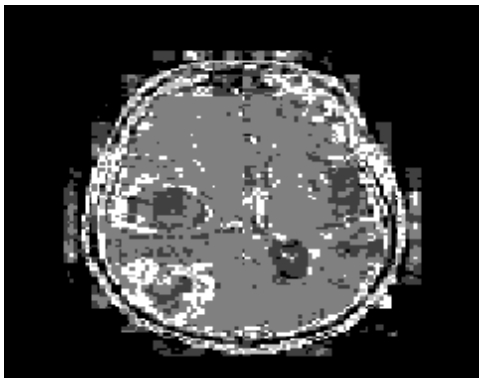

**Figure 26.c.** The result of Fuzzy C-Means on True-PCA with 128*128

**Figure 26.d.** The result of Fuzzy C-Means on True-PCA with 64*64

Besides, after PCA methods were implemented on resized MRI image, the average reconstruction errors of MRI images obtained from PCA methods were calculated. The average reconstruction error was calculated using by Equation 27.

$$Error = mean(sum((A - W * (W' * A)).\hat{}2)) \quad (27)$$

where $A$ is the data matrix where each column is a data point, $W$ is the eigenvector matrix. When reconstruction error was calculated, the number of iterations was determined according to the dimensions of MRI image.

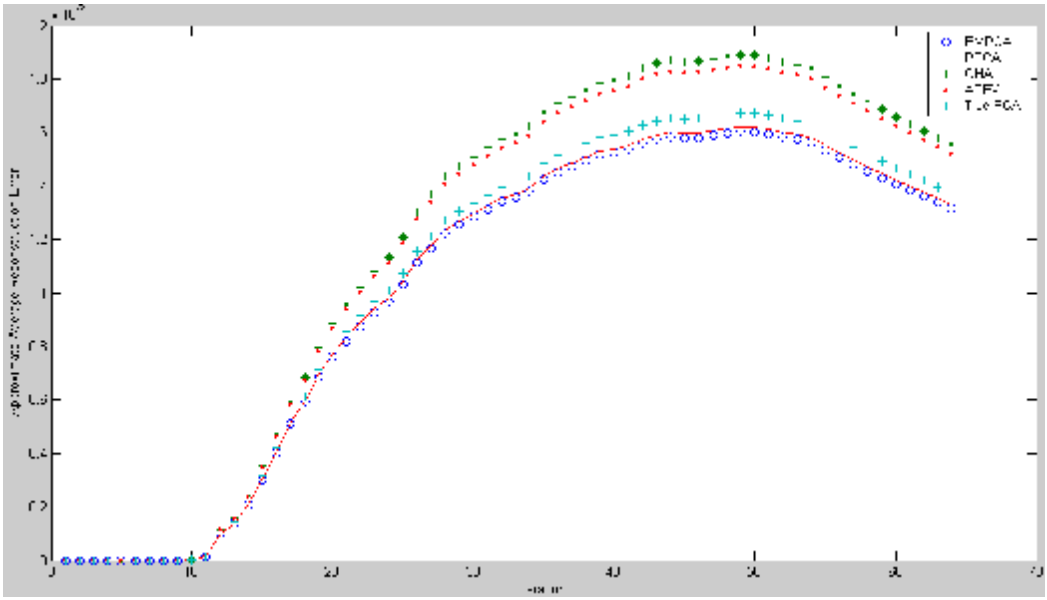

**Figure 27.** The approximate error graphic (64*64)

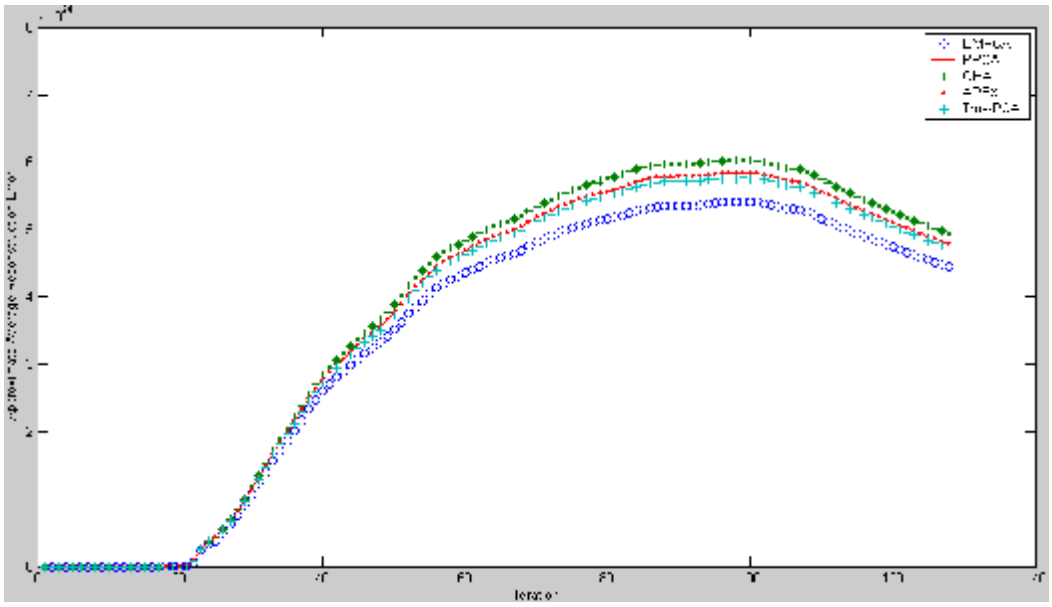

**Figure 28.** The approximate error graphic (128*128)

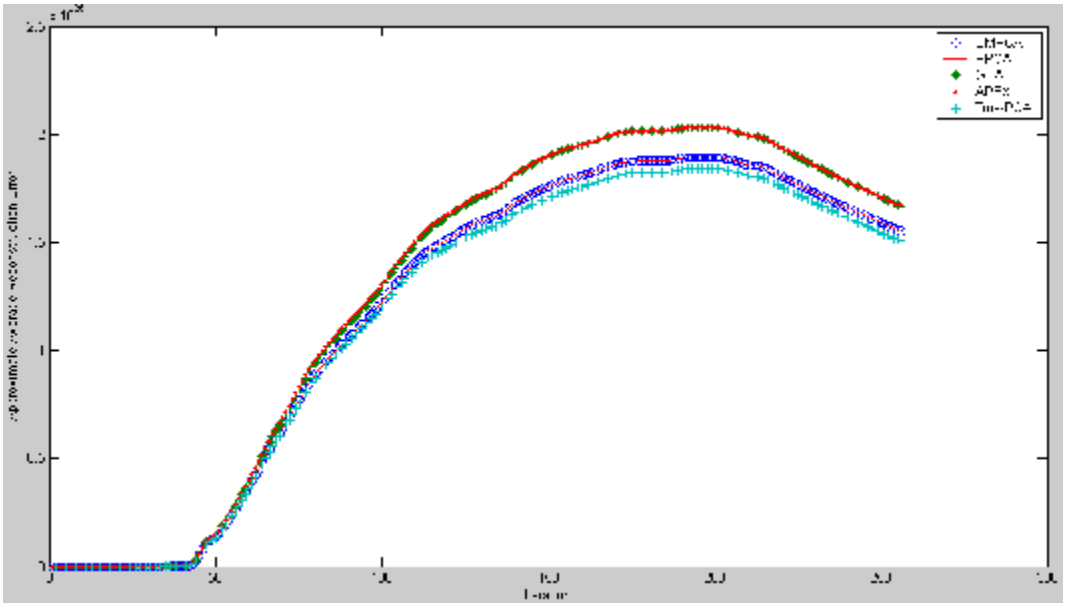

**Figure 29.** The approximate error graphic (256*256)

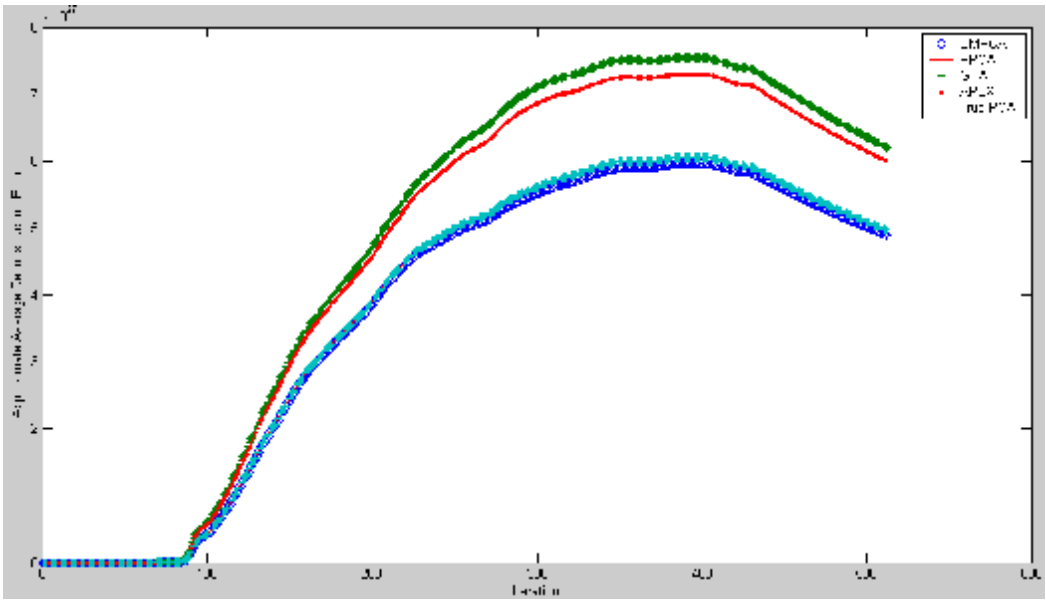

**Figure 30.** The approximate error graphic (512*512)

As to obtained error rates, when the dimensions of images are high, error rates are high too. Also according to error rates in PCA methods, the best results were obtained with EM-PCA and PPCA methods.

| | **(512*512)** | **(256*256)** | **(128*128)** | **(64*64)** |
|---|---|---|---|---|
| **EMPCA** | 3.743e+027 | 1.1916e+026 | 3.4214e+024 | 1.0261e+023 |
| **PPCA** | 3.7991e+027 | 1.1918e+026 | 3.6829e+024 | 1.0151e+023 |
| **GHA** | 4.7339e+027 | 1.2767e+026 | 3.8947e+024 | 1.1686e+023 |
| **APEX** | 4.5778e+027 | 1.2878e+026 | 3.6832e+024 | 1.1943e+023 |
| **True-PCA** | 3.7993e+027 | 1.1597e+026 | 3.7795e+024 | 1.0583e+023 |

**Table 3.** The Approximate Error for different sizes and different PCAs

# 6. CONCLUSIONS

Image clustering is used for high-level description of image content. It has been playing an important role in solving many problems in pattern recognition and image processing. It's problem that the images wanted to classify have a complicated high dimensional structure in image clustering. Because of this, after PCA methods were implemented, image clustering methods were implemented on MRI images in the thesis.

MRI image has 512 lines and 512 columns were used in the thesis. First the dimensions of original image were reduced. PCA methods were applied to the resized image. First, the dimensions of the MRI image was reduced by using PCA method without much loss of information. Principle Components were obtained by using PCA methods. According to the observed images and error rates, it was agreed that PPCA and EM-PCA methods were the best methods as regards the others. Because the EM-PCA and PPCA obtain the eigenvectors without explicit computation of the sample covariance and allow simple and efficient computation of a few eigenvectors and eigenvalues when working with high dimensional data.

After PCA methods were implemented, image clustering methods were implemented on resized MRI image obtained with PCA methods. In the image clustering methods, the number of clusters was determined as 5, according to the histogram of the original image. In PCA methods, the best results were obtained with Probabilistic PCA and EM-PCA methods implemented on resized MRI images for all size of the image. Among image clustering methods, Fuzzy C-Means algorithm gives better results than K-Means algorithm. Because the performance of K-Means algorithm depends on the initial positions of centers. But FCM is an iterative algorithm that FCM finds cluster centers that minimize a dissimilarity function Equation 24. Also, in the clustering methods, the best results were obtained with Fuzzy C-Means algorithm implemented on resized MRI image obtained with EM-PCA and PPCA.

**REFERENCES**


ALBAYRAK, S., AMASYALI, F., 2003. Fuzzy C-Means Clustering On Medical Diagnostic Systems. International XII. Turkish Symposium on Artificial Intelligence and Neural Networks-TAINN 2003, İstanbul.

AREFI, H., HAHN, M., SAMADZADEGAN, F., LINDENBERGER, J., 2004. Comparison Of Clustering Techniques Applied To Laser Data. ISPRS 2004 International Society for Photogrammetry and Remote Sensing, İstanbul.

BERKS, G., KEYSERLINGK, D.G. , JANTZEN, J. ,DOTOLI, M., AXER, H., 2000. Fuzzy Clustering- A Versatile Mean to Explore Medical Database. ESIT2000, Aachen, Germany.

BEZDEK, J.C., 1973. Pattern Recognition with Fuzzy Objective Function Algorithms. PhD Thesis, Applied Math. Center, Cornell University, Ithaca.

CHEN, S., BILLINGS, S.A., GRANT, P.M., 1992. Recursive hybrid algorithm for non-linear system identification using radial basis function Networks. International Journal of Control, pp. 1051-1070.

DARKEN, C., MOODY, J., 1990. Fast adaptive k-means clustering: Some empirical results. IJCNN International Joint Conference on Neural Networks, pp. 233-238.

DEMPSTER, A. P., LAIRD, N. M., RUBIN, D. B., 1977. Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society Series B, pp. 1-38.

DIAMANTARAS, I., KUNG, S. Y., 1996. A Neural Network Learning Algorithm For Adaptive Principle Component Extraction. John Wiley & Sons, New York, 255s.

DUNN, J.C., 1974. Well Separated Clusters and Optimal Fuzzy Partitions. J. Cybern, Vol. 4, pp. 95-104.

GHAHRAMANI, Z., JORDAN, M.I., 1994. Supervised learning from incomplete data via an EM approach. Advances in Neural Information Processing Systems, pp. 120–127.

GLEICH, D., 2002. Principal Component Analysis and Independent Component Analysis with Neural Networks. Available: http://www.stanford.edu/~dgleich/publications/pca_neural_nets_website

GOLDBERGER, J., GREENSPAN, H., GORDON, S., 2002. Unsupervised Image Clustering Using the Information Bottleneck Method. Springer Berlin, Heidelberg.

HERTZ, J., KROGH, A., PALMER R.G., 1991. Introduction to the theory of neural computation. Addison-Wesley Longman Publishing Co., USA, 327s.

JANG, J.S.R., SUN, C.T., MIZUTANI, E., 1997. Neuro-Fuzzy and Soft Computing. Pearson Education, pp. 426-427.

JUNTU, J., SIJBERS, J., DYCK, D. V., 2007. Classification of Soft Tissue Tumors in MRI Images using Kernel PCA and Regularized Least Square Classifier. Signal Processing, Pattern Recognition and Applications SPPRA- 2007 Innsbruck, Austria.

KAYA, M., 2005. An Algorithm for Image Clustering and Compression. Turk J Elec Engin, Tübitak, Vol.13, pp.81-83.

LLOYD, S.P., 1957. Least squares quantization in PCM. Bell Laboratories Internal Technical Report, IEEE Trans. on Information Theory.

LYNCH, M., 2000. Analysis of Cardiac Images in MRI. Vision Systems Group, Dublin City University.

MACQUEEN, J., 1967. Some methods for classification and analysis of multi-variate observations. Proc. of the Fifth Berkeley Symp. on Math., Statistics and Probability, pp. 281.

MASHOR, M.Y., 1998. Improving the Performance of K-Means Clustering Algorithm to Position the Centres of RBF Network. International Journal of the Computer, The Internet and Management, vol.6.

MATLAB, 2001. The MathWorks (Version 6.5), USA.

MOHAMED, N.A., AHMED M.N., FARAG A., 1998. Modifed Fuzzy C-Means in Medical Image Segmentation. IEEE Int. Conf. on Engineering Medicine and Biological Sciences, pp. 1377-1380.

MUSA, M.E.M., DUIN, R.P.W., RIDDER, D., 2000. Modelling Handwitten Digit Data using Probabilistic Principal Component Analysis. 6th Annual Conf. of the Advanced School for Computing and Imaging, Belgium, pp. 145-152.

PAREKH, K., HERLING, N., 2003. Image Analysis. Available: http://math.la.asu.edu/~cbs/pdfs/projects/Fall_2003/ presentationgroupBIV.pdf

RICARDO ,G., 1998. Principle Component Analysis. Texas A&M University, pp. 8-11.

ROWEIS, S. , 1997. EM Algorithm for PCA and SPCA. Neural Information Processing Systems (NIPS)'97, California, pp. 1-7.

SELVATHI, D., ARULMURGAN, A., THAMARAI SEIVI, S., ALAGAPPAN, S., 2005. MRI image segmentation using unsupervised clustering techniques. Computational Intelligence and Multimedia Applications, Sixth International Conference, pp. 105-110.

SMITH, L.I., 2002. A tutorial on Principal Components Analysis. Web Published, pp. 12-20.

TEKNOMO K., 2006, K-Means Clustering, Available: http://people.revoledu.com/kardi/tutorial/kMean/WhatIs.htm

ZHAO, L., CHAI, T., CONG, Q., June 2006. Operating Condition Recognition of Pre-denitrification Bioprocess Using Robust EMPCA and FCM. Intelligent Control and Automation, WCICA 2006, The Sixth World Congress, pp. 9386- 9390.

ZIYAD, N.A., GILMORE, E.T., and CHOUIKHA, M.F., 1998. Improvements For Image Compression Using Adaptive Principal Component Extraction (Apex). Conference Record of the Thirty-Second Asilomar Conference, pp. 969-972.

## BIOGRAPHY

Emine GEZMEZ was born on April 13th. 1981 in Adana, TURKEY. She graduated from Mersin Mezitli Science High School in 1998. She received her B.Sc. degree from Computer Engineering Department of Mersin University in 2003.

After she graduated from University, she started as a master student at the Electrical and Electronics Engineering Department of Çukurova University.