

A Novel Approach to Retrieving an Image using Content Based Kernel

¹Gumpina V V Satya Prasad, ²Satya P Kumar Somayajula

¹IT Dept, Sir. C. R. Reddy College of Engg, Eluru, A.P., India.

²CSE Dept, Avanthi Institute of Engg & Tech, Tamaram, Visakhapatnam, A.P., India.

Abstract

In this paper we are implementing medical domain dedicated medical database management server to manage collection of medical data and alphanumerical values. By using database management techniques the classical operations for databases with the including of different series of algorithms used to extract visual data from image (images including all colors and texture data also). In Hue Saturation Value space contains 166 colors with histogram represents the information of color images. By applying Gabor Filters we can obtain information of texture in the form of vector with 12 values. Finally, the retrieved data is successfully stored in a database by using DBMS applications in the form of IMAGE. It is also used for visual queries. Here we are using some clustering algorithms to increase image retrieval speed.

Keywords

medical DBMS server; Image and texture based retrieval; clustering..

I. Introduction

Medical information is not limited only to strings. In order to facilitate different multimedia processing, advanced DBM systems can integrate various data types (such as: Non-numeric information, images, video and text.) in a single database. When a database is created, the aim is to be used for the query process. There are two types of queries that can be executed. one type of query process is the classical one that use simple text based query. It can be used in the following cases:

- a) The doctor knows the name of the patient and he wants to find all information about him.
- b) The doctor knows a certain diagnosis and wants to find all similar cases with the same diagnosis.

Another modern type of query is the content based retrieval. That means that the search is made using similar characteristics of an image. The images have to be processed, extracted the characteristics and then compared these characteristics in order to find the most appropriate. As an immediate effect, traditional information management and processing systems cannot be used on this large collection of various data types. Most times, information in raw format is no use. The real benefit comes when different decisions are taken based on it or it can be explored and obtained provenience information. There are a few areas where large information volumes are stored in centralized or distributed database. Among the most important areas it can be enumerated: imagistic medicine, digital libraries, image archives, bioinformatics, health, finances and investments, production, marketing strategies, telecommunication networks, scientific areas, WWW and biometry. Usually in these databases the multimedia information as images, video, audio, etc, are stored either in separate files, or inside the database as a BLOB type. This making harder to execute visual content based queries directly. All this are leading to the necessity for developing data a medical server that include intelligent data storage and analysis methods used for content based retrieval and automate data classification. Regarding content based retrieval it has been used a clustering

method for grouping similar images. Clustering is the process of grouping a set of physical or abstract objects into classes of similar objects. As a product of clustering process, associations between different actions on the platform can easily be inferred from the logged data. In general, the activities that are present in the same profile tend to be found together in the same session[1]. To manage medical domain databases we can use a dedicated database management server (DBMS) based on the relational model and used for managing medium sized databases. For clustering process it is used Weka package. In medical domain we can easily manage medium sized image collections and alphanumerical information. The DBMS is platform independent. In the Section 2 it is presented similar implementations from well known servers, along their particularities. Section 3 presents the architecture of the server, in Section 4 it is presented the database management and querying, and in the Section 5 it is presented a solution for clustering the images.

II. Corresponding Work

Most of the applications that use multimedia information are based on database management systems as Microsoft Structure Query Language Server, My Structure Query Language or Inter base. Each of them offers a partial support for multimedia content. MYSQL offers only the BLOB data type that can be used to store images. A BLOB is a binary large object that can hold a variable amount of data. BLOB columns have no character set, and sorting and comparison are based on the numeric values of the bytes in column values[2]. MICROSOFT STRUCTURE QUERY LANGUAGE Server offers a data Type called "image", but with no other support. It is considered a variable-length binary data from 0 through 231-1 (2,147,483,647) bytes. There are no pre-defined functions that can be used for extracting characteristics or building content-based queries. More than that, in the Microsoft Structure query language server 2008 it is specified that "n-text", "text" and "image" data types will be removed in a future version of Microsoft SQL Server. The recommendation is to avoid using these data types in new development work, and plan to modify applications that currently use them. It should be used nvarchar(max), varchar(max), varbinary(max) instead [3,4]. The complete solution is provided by Oracle – the Oracle 10g database server and Intermediate tool that can manage all kind of multimedia data, including DICOM files [5,6]. In addition to the image support offered via the ORD Image object type, in Oracle 10g version intermediate provides support for the first edition. This standard defines object relational types for images and image characteristics. Each object type includes attributes, methods, and associated SQL functions and procedures. Use of the SQL standard interface may make some applications more portable across various vendor databases.

For the clustering algorithms, there are many methods detailed in the literature: partitioning methods, hierarchical methods, density-based methods such as, grid-based methods or model-based methods [7]. Hierarchical clustering algorithms like the Single-Link method or OPTICS compute a representation of the possible hierarchical clustering structure of the database in the

form of a dendrogram or a reachability plot from which clusters at various resolutions can be extracted.

Because we are dealing with numeric attributes, iterative- based clustering is taken into consideration from partitioning methods. The classic k-means algorithm is a very simple method of creating clusters. Firstly, it is specified how many clusters are being thought: this is the parameter k. Then k points are chosen at random as cluster centers. Instances are assigned to their closest cluster center according to the ordinary Euclidean function. Next the centroid, or the mean, of all instances in each cluster is calculated – this is the “means” part. These centroids are taken to be the new center values for their respective clusters. Finally, the whole process is repeated with the new cluster centers. Iteration continues until the same points are assigned to each cluster in consecutive rounds, at each point the cluster centers have stabilized and will remain the same thereafter.

From a different perspective for a cluster there may be computed the following parameters: means, standard deviation and probability (μ , σ and p). The expectation- maximization (EM) algorithm that is employed is a k-means clustering algorithm type. It takes into consideration that we now neither parameters. It starts with initial guess for the parameters, use them to calculate the cluster probabilities for each instance, use these probabilities to estimate the parameters, and repeat. This is called the EM algorithm for “expectation-maximization”. The first step, the calculation of cluster probabilities (which are the “expected” class values) is “expectation”; the second, calculation of the distribution parameters is “maximization” of the likelihood of the distributions given the data .

The quality of clustering process is done by computing the likelihood of a set of test data given the obtained model. The goodness-of-fit is measured by computing the logarithm of likelihood, or log-likelihood: and the larger this quantity, the better the model fits the data. Instead of using a single test set, it is also possible to compute a cross validation estimate of the log-likelihood.

III. Medical Database Kernel – Overview

The kernel it is a tool that can be used for creation, maintenance, simple text based query and content based visual query using color and texture characteristics of the multimedia digital collections from medical domain. This dedicated MDBMS permits database creation, table and constraints adding (primary key, foreign keys), inserting images and alphanumeric information, simple text based query and content based query using color and texture characteristics. The software tool is easy to be used because it respects the SQL standard. It does not need advanced informatics knowledge and has the advantage of low cost. It is a good alternative for a classical database management system (Microsoft Access, Microsoft structured query language server, Oracle10g Server as Intermediate), which would need higher costs for database server and for designing applications for content-based retrieval [11].

The next fig. presents the general architecture of the medical system. In the first step the application that uses the server should connect to the database. This way it will be created a communication channel between them. All commands and responses will use this channel to send queries requests and receive answers. The server has two main modules: kernel engine and database files manager. The kernel engine includes all functions implemented in the DBMS. It is composed from several sub-modules each of them with specific tasks:

- The main module. It is the module which manages all

communications with the client. It is the one that receives all queries requests, check what is the type of query requested, extracts the parameters of the query and calls the specific module to execute it.

- Queries response module. After the query is executed, the results will be sent to the Queries Response Module. It will compact the result using a standard format and then return it to the client. The client will receive it on the same communication channel used to send the request.
- Select/Select/Image Processing module. If the main module concludes that is a SELECT SQL command, it will call the SelectProcessing module. This module extracts the parameters from the query and then search in the database. General architecture of the systemfiles for specific information. If it use for comparison the similitude of characteristics instead equality of parameters.

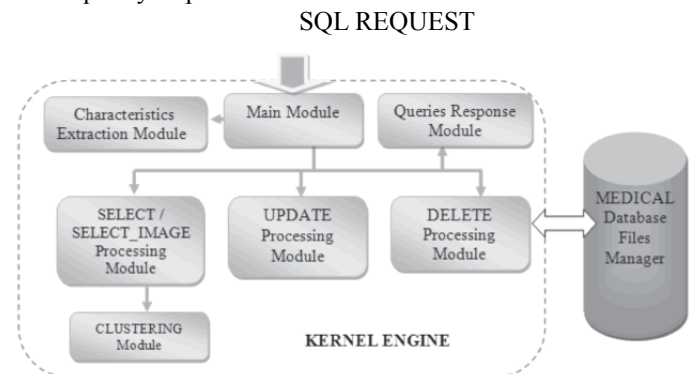


Fig. 1: General Architecture of the System

- Characteristics Extraction Module. When the main module receives a SELECT IMAGE or a UPDATE query which uses an image that is not already in the database it is needed first to process it. This module is called to extract the color and texture characteristics of the image. The data of the results will be used to initialize a variable of IMAGE data type.
- Clustering Module. The role of this module is to assign each image to a cluster containing similar images.
- Update Processing Module. When the query received from the user is an UPDATE command, it will be called to execute it.
- Delete Processing Module. It is called when the user executes a DELETE command. The kernel executes only logic deletes. It never executes physical deletes. The physical deletes are executed only when a “Compact Database” command is sent by the user.

The second main module is the Medical Database Files Manager. It is the only module that has access for reads and writes to the files in the database. It is his job to search for information in the files, to read and write into files and to manage locks over databases. When a client module request a read form a file it is enabled a read lock for the specific file (that represents a table in the database). All other read requests will be permitted but no writes will be allowed. If the client module request a write to file, it will be enabled a write lock. No other requests will be allowed until the lock is canceled. The results will always be returned to the module which made the request. The data read or wrote to files is not structured in any way. This module does not modify the structure in any way. All the results will be raw data, as it is read from the files or received from client modules.

IV.Expectation-Maximization Algorithm & Gabor Filters

- From a different perspective for a cluster there may be computed the following parameters: means, standard deviation and probability (μ , σ and p). The expectation-maximization (EM) algorithm that is employed is a k-means clustering algorithm type.
- In this we know neither parameter.
- It starts with initial guess for the parameters; use them to calculate the cluster probabilities for each instance.
- Use these probabilities to estimate the parameters, and repeat. This is called the EM algorithm for “expectation-maximization”.
- The first step, the calculation of cluster probabilities (which are the “expected” class values) is “expectation”.
- The second, calculation of the distribution parameters is “maximization” of the likelihood of the distributions given the data.

Sample Algorithm

```
function [label, model, llh] = emgm(X, init)
% EM algorithm for Gaussian mixture model
%% initialization
fprintf('EM for Gaussian mixture: running ... ');
R = initialization(X,init);

tol = 1e-6;
maxiter = 500;
llh = -inf(1,maxiter);
converged = false;
t = 1;
while ~converged && t < maxiter
    t = t+1;
    model = maximization(X,R);
    [R, llh(t)] = expectation(X,model);
    converged = llh(t)-llh(t-1) < tol*abs(llh(t));
end
[~,label(1,:)] = max(R,[],2);
llh = llh(2:t);
if converged
    fprintf('converged in %d steps.\n',t);
else
    fprintf('not converged in %d steps.\n',maxiter);
end
function R = initialization(X, init)
[d,n] = size(X);
if isstruct(init) % initialize with model
    R = expectation(X,init);
elseif length(init) == 1 % random initialization
    k = init;
    idx = randsample(n,k);
    m = X(:,idx);
    [~,label] = max(bsxfun(@minus,m'*X,sum(m.^2,1)'/2));
    while k ~= unique(label)
        idx = randsample(n,k);
        m = X(:,idx);
        [~,label] = max(bsxfun(@minus,m'*X,sum(m.^2,1)'/2));
    end
    R = full(sparse(1:n,label,1,n,k,n));
elseif size(init,1) == 1 && size(init,2) == n % initialize with labels
    label = init;
```

```
    k = max(label);
    R = full(sparse(1:n,label,1,n,k,n));
elseif size(init,1) == d && size(init,2) > 1 %initialize with only centers
    k = size(init,2);
    m = init;
    [~,label] = max(bsxfun(@minus,m'*X,sum(m.^2,1)'/2));
    R = full(sparse(1:n,label,1,n,k,n));
else
    error('ERROR: init is not valid.');
```

```
end

function [R, llh] = expectation(X, model)
mu = model.mu;
Sigma = model.Sigma;
w = model.weight;

n = size(X,2);
k = size(mu,2);
R = zeros(n,k);
for i = 1:k
    R(:,i) = loggausspdf(X,mu(:,i),Sigma(:,i));
end
R = bsxfun(@plus,R,log(w));
T = logsumexp(R,2);
llh = sum(T)/n; % loglikelihood
R = bsxfun(@minus,R,T);
R = exp(R);

function model = maximization(X, R)
[d,n] = size(X);
k = size(R,2);
sigma0 = eye(d)*(1e-6); % regularization factor for covariance

s = sum(R,1);
w = s/n;
mu = bsxfun(@rdivide, X*R, s);
Sigma = zeros(d,d,k);
for i = 1:k
    Xo = bsxfun(@minus,X,mu(:,i));
    Xo = bsxfun(@times,Xo,sqrt(R(:,i)'));
    Sigma(:,i) = (Xo*Xo'+sigma0)/s(i);
end

model.mu = mu;
model.Sigma = Sigma;
model.weight = w;
```

V. Gabor Filters

A Gabor function in the spatial domain is a sinusoidal modulated Gaussian. For a 2-D Gaussian curve with a spread of x s and y s in the x and y directions, respectively, and a modulating frequency of 0 u, the real impulse response of the filter is given by Where are the standard deviations of the Gaussian envelopes along the x and y direction. Then a set of Gabor filters can be obtained by appropriate dilations and rotations of $g(x, y)$:

$$g_{mn}(x, y) = a^{-m} g(x', y')$$

$$x' = a^{-m} (x \cos \theta + y \sin \theta)$$

$$y' = a^{-m} (-x \sin \theta + y \cos \theta)$$

1. Where $a > 1$, $\theta = n\pi/K$, $n = 0, 1, \dots, K-1$, and $m = 0, 1, \dots, S-1$. K and S are the number of orientations and scales. The scale factor a^{-m} is to ensure that energy is independent of m .
2. The impulse response function is shown graphically in Fig.

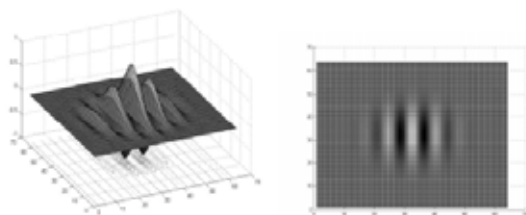


Fig. Gabor function in the spatial domain

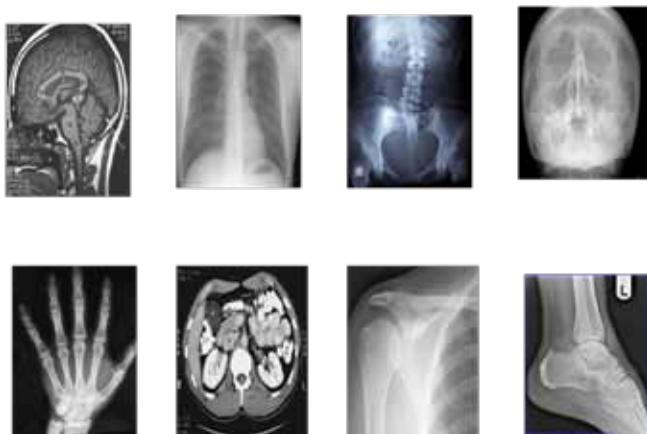
VI. Implementation

A. Insert Image

In this we are giving the medical image as an input. After giving the image we are extracting the characteristics for the image greater than or equal to 12. If the values extracted are less than 12 then that image will not consider. Now the values are called in Java language and doing clustering by comparing each and every values in all the clusters and also it stores the values in database. Finally if the values are matched similar, then image will be uploaded successfully in the database in that particular cluster.

B. Retrieve Image

In this we are giving the medical image as an input. After giving the image we are extracting the characteristics for the image (color and texture characteristics) from Mat lab and these values are stored in a .csv format. The values that are extracted must be greater than or equal to 12. If the values extracted are less than 12 then that image will not consider. Now the values are called in Java language and doing clustering by comparing each and every values in all the clusters. Finally if the values are matched similar, then image will be displayed successfully from the database.



The following are the reference data sets and images used in this paper:

C. Data Sets

```
@relation Medical images
@attribute image type {appendicitis, brain}
@attribute texture-v1 numeric
@attribute texture-v2 numeric
@attribute texture-v3 numeric
@attribute texture-v4 numeric
```

```
@attribute texture-v5 numeric
@attribute texture-v6 numeric
@attribute texture-v7 numeric
@attribute texture-v9 numeric
@attribute texture-v10 numeric
@attribute texture-v11 numeric
@attribute texture-v12 numeric
@attribute color-hue numeric
@attribute color-saturation numeric
@attribute color-value numeric
@data
appendicitis,88.189,157.44,138.75,76.465,43.19,
56.558,56.393,37.387,49.792,89.371,107.78,
93.866, 0.61111,0.03,100
appendicitis,99.81,118.58,76.218,73.896,107.55,
104.38,44.165,0.99921,34.671,93.909,113.57,
96.152, 0.2004,0.72414,116
bran,96.673,118.09,85.931,68.715,59.81,48.395,57.484,99,134.
46,122.71,67.148,21.16,0.15537,0.56731,104
bran,92.541,146.19,122.48,79.987,66.641,78.555,80.912,70.10
7,66.685,79.741,1
05.55,121.9,0.095238,0.067308,104
```

VII. Medical Database Management and Query

This kernel it is a database engine that can be used by all kind of applications working with databases. The applications can be implemented in C++, Java or any other language that supports TCP/IP communication.

An element of originality is that among the classic operations that can be executed (like creation, maintenance, simple text based query) it is included a new module for content based visual query using color and texture characteristics of the multimedia digital collections. This module will search in the database for images with similar texture and color and return all the associated information (the similar image along with the rest of the columns).

In order to execute a query the client application has to connect first to the database server. The connection is made using TCP standard and should include a username and a password. Each user has defined a series of rights that he can benefits from: create databases, create tables in a specific database, modify tables' structures, insert data into tables and execute queries. Each of these rights is defined at the database level. When a database is created, only users who have defined rights can work with it.

A. Inserting Data into Database

The communication between application and DBMS is based on messages exchange. All messages represent commands written in SQL language. There are two possible commands:

1. Inserting only text:

Insert into PATIENT values ("disease", polyps")

In this case the main module receives the command, checks the user's rights and if the user has rights to insert new values, calls the Update Processing Module to add data to database.

2. Inserting text and images

Insert into PATIENT values ("disease", "polyps", "analysis.bmp")

In this case, after the Update Processing Module is called, it checks the metadata of the table and finds out that one of the attributes has the type image. It notifies the MAIN Module that it should receive from the client a file image called "analysis.bmp". After the image

is received, it will call the CHARACTERISTICS EXTRACTION Module to process the image and create the image type value. Only after this type is created, it is inserted into database.

In order to increase the retrieval speed for future queries, it is also called the CLUSTERING Module that will assign the new image to the cluster that contains the images with similar characteristics.

A. Simple Text Based Queries

In this case the main module receives the SELECT command, checks the user's rights and if the user has rights to execute select commands, calls the SELECT Module to search data into the database.

C. Visual Content Based Queries

The MDBMS includes a special select command that can be used for content based retrieval operations. If the user sends a SELECT_IMAGE command to the system, the DBMS will try to find all the records in the table that contain similar images.

The syntax of the query is:

```
Select_Image from [table_name] where
[image_column_name] like [image_type_value]
```

Where:

“tbl_name” is the name of the table

“image_column_name” is the name of the column that contains the image itself

“image_type_value” is the image type value resulted after processing the image.

Eg: Select_Image from PATIENTS where image like image1

This type of query can be visually built, in two different ways. In the first case the user sends a SELECT ALL query that will retrieve all the records in the table. From these records it selects the image he wishes to use it as a query and then sends a SELECT_IMAGE query. The DBMS returns all records containing similar images. In the second way, the user has his own image and first calls the Processing Module that creates the image type value. Then it is sent to the system a SELECT_IMAGE command. In order to increase the retrieval speed it is used the Clustering module. In this case, first it is searched for the cluster that contains images most similar with the query image. After this cluster is located, the system looks for similar images only inside it. In this way the quantity of data needed to be compared is reduced substantially.

VIII. Clustering The Images

In order to increase the speed of the retrieval process, after an image is inserted in the database, it should be assigned to a cluster containing similar images. For this operation we chose to use the algorithms provided by Weka package. Weka is a collection of machine learning algorithms for data mining tasks. It contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. The Expectation-Maximization (EM) algorithm from Weka clustering package EM is a statistical model that makes use of the finite Gaussian mixtures model. The algorithm is similar to the K-means procedure in that a set of parameters are re-computed until a desired convergence value is achieved. It needs the input data to be in a custom format called arff. Under these circumstances we have developed an offline Java application that queries the database and creates the input data file called activity.arff. This process is automated and is driven by a property file in which there is specified what data will lay in activity.arff file. The most important step in this

procedure is the attribute selection and the granularity of their nominal values. The number of attributes and their meaning has a crucial importance for the whole process, since irrelevant attributes may degrade classification performance in sense of relevance. On the other hand, the more attributes we have the more time the algorithm will take to produce a result. Domain knowledge and of course common sense, are crucial assets for obtaining relevant results. For an image we may have a very large number of attributes. Still, in our procedure we used only two sets:

color characteristics and texture characteristics. Here is how the arff file looks like:

```
@relation images
@texture
{v1,v2,v3,v4,v5, .. v12}
@color
{v1,v2,v3,v4,v5,...,v166}
@data
Texture1 v1,v2,...
Color1 v1,v2, ..
Texture2 v1,v2,...
Color2 v1,v2, ..
```

As it can be seen from the definition of the attributes each of them has a set of five nominal values from which only one may be assigned. The values of the attributes are computed for each of the XXXX images and are set in the @data section of the file. For example, the first line says that the image 1 has value v1 for parameter named color1 and has value v1 for parameter named texture2. The granularity for the nominal values of the attributes can be also increased. In our study we considered only five possible values but we can consider to test the algorithm with more possible values. This should have great impact on the number of the clusters obtained. The time taken by the algorithm to produce results should also increase. Running the EM algorithm created XX (e.g. three) clusters. The procedure clustered XX (e.g. 91) instances (34%) in cluster 0, 42 instances (16%) in cluster 1 and 135 instances (50%) in cluster 3. The final step is to check how well the model fits the data by computing the likelihood of a set of test data given the model. Weka measures goodness-of-fit by the logarithm of the likelihood, or log-likelihood: and the larger this quantity, the better the model fits the data. Instead of using a single test set, it is also possible to compute across validation estimate of the log-likelihood. For instances the value of the log-likelihood is -2.61092 which represent a promising result in the sense that instances (in our case images) may be classified in three disjoint clusters based on their characteristics.

This function of the database is activated after processing the image, when the user sends an INSERT/UPDATE command. The system calls the Clustering Module in order to include the newly inserted image in a cluster. The main problem in this case for the Clustering Module is to determine which the cluster containing similar images are. After this cluster is located, only images contained here are computed in order to find images with the highest similarity to the query image.

IX. Conclusion and Future Work

We presented a very comprehensive and practical Database Kernel for Image Retrieval system for medical domain. The proposed system is created for managing and querying medium sized personal digital collections that contain both alphanumeric information and digital images. The software tool allows inserting and retrieving images in database. The user can use several types of data as integer, char, double and image. There are also implemented the two constraints used in relational

model: primary key and referential integrity. The software tool will execute content based visual query at characteristics. These characteristics are automatically extracted when the images are inserted in the database. And also studying and implementing indexing algorithms for data inserted in the tables. The speed of the image retrieval module is increased by using a clustering module that will group the images in clusters. This software can be extended in the following directions: Adding new types of traditional and medical data types such as video type or DICOM type.

References

- [1] R. Agrawal R. Srikant, "Fast algorithms for mining association rules," Proc. of the 20th VLDB Conference, pp. 487-499, Santiago, Chile, 1994.
- [2] MySQL 5.0 Reference Manual, (2009). [Online] Available : <http://dev.mysql.com/doc/refman/5.0/en/blob.html>
- [3] SQL Server 2008, Books Online, (2009). [Online] Available : <http://msdn.microsoft.com/en-us/library/ms187993.aspx>
- [4] A. Chigrik, SQL Server 2000 vs Oracle 9i, (2007). [Online] Available : www.mssqlcity.com/Articles/Compare/sql_server_vs_oracle.htm
- [5] Oracle® Database SQL Reference 10g Release 2 (10.2), [Online] Available : http://download.oracle.com/docs/cd/B19306_01/server.102/b14200/sql_elements001.html
- [6] New inter Media Features in Oracle 10g, [Online] Available : http://www.oracle.com/technology/products/intermedia/htdocs/imedia_new_features_in_10g.html
- [7] M. Ester, H.P.Kriegel, J. Sander, X.Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise", Proc. KDD'96, Portland, OR, pp.226-231,1996.
- [8] R. Sibson, "SLINK: An Optimally Efficient Algorithm for the Single-link Cluster Method", The Computer Journal, 16(1): 30-34,1973.
- [9] [Online] Available : www.cs.waikato.ac.nz/ml/weka
- [10] L. Stanescu, D.D. Burdescu, M.Brezovan, C. Stoica Spahiu, A.Ion, "A New Software Tool For Managing and Querying the Personal Medical Digital Imagery", Proceedings of the International Conference on Health Informatics, Porto-Portugal, pp. 199-204,2009
- [11] L. Stanescu, M.C. Mihaescu, D.D.Burdescu, E. Georgescu, L.Florea, "An Improsved Platform for Medical E-Learning", Lecture Notes in Computer Science 4823, Springer, pp.392-403, 2008.
- [12] [Online] Available : http://grb.mnsu.edu/grbts/doc/manual/Expectation_Maximization_EM.html
- [13] James Foulds. Learning instance weights in multi-instance learning. Master's thesis, Department of Computer Science, University of Waikato, 2008 [Online] Available : http://adt.waikato.ac.nz/uploads/approved/adt-uow_20080308.150730/public/02whole.pdf



Image Processing, Data Mining, Networks security, Web security and Software Engineering.

Mr. Gumpina V V Satya Prasad is working as an Asst. Professor, in IT Department, Sir C. R. Reddy College of Engg, Eluru, A.P., India. He has received his B.Tech CS&IT from Avanathi Institue of Engg & Tech Visakhapatnam and M.Tech (SE) from Gandhi Institute of Technology And Management University (GITAM University), Visakhapatnam, A.P., INDIA. His research interests include



International journals & 5 National journals. His research interests include Image Processing, Networks security, Web security, Information security, Data Mining and Software Engineering.

Mr. Satya P Kumar Somayajula is working as an Asst.Professor, in CSE Department, Avanathi Institue of Engg & Tech, Tamaram, Visakhapatnam, A.P., India. He has received his M.Sc (Physics) from Andhra University, Visakhapatnam and M.Tech (CST) from Gandhi Institute of Technology And Management University (GITAM University), Visakhapatnam, A.P., INDIA. He published 7 papers in reputed