ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2016

A Study on Content Based Image Retrieval System using K-NN Algorithm and Mobile Agents

Yogita B. Verulkar[1], Bharat K. Chaudhari[2]

M.E. Student, Dept. of Computer Engineering, Padm. Dr. V. B. Kolte College of Engineering, Maharashtra, India[1]

Head of Department, Dept. of Computer Engineering , Padm. Dr. V. B. Kolte College of Engineering, Maharashtra.India[2]

ABSTRACT: As we know that various images are stored over different locations on network. Content Based Image Retrieval (CBIR) System retrieves the images according to the contents or features such as color, shape and texture of images. So for retrieving the images over network we review the mobile agent technology. Mobile agents are having the capability of migrating freely over the network to collect the required information and returns back to the user with desired results. K-NN algorithm is an efficient algorithm in order to find the similarity between different images So in this paper we are presenting these technology i.e. CBIR and mobile agents along with K-NN algorithm for retrieving images over network.

KEYWORDS: Content based image retrieval, distributed CBIR, mobile agents.

I. INTRODUCTION

As multimedia devices are becoming very usual, huge collections of digital images are available nowadays. Finding images belonging to a particular category in these ever growing collections is a problematic task since searching within by hand has become impossible. Content Based Image Retrieval has been successfully proposed to answer this problem. In a CBIR system, for image descriptionslow-level visual image features such as color, texture, and shape are automatically extracted[6]. In order to search for desirable images, a user presents an image as an example of similarity, and the system gives a set of similar images based on the extracted features of that image. The problem of such observes is the well-known semantic gap between the numerical values committed to images and the semantical concepts to which they belongs. In order to decrease the gap, machine learning techniques have been effectively adapted to train a similarity function in interaction with the user (usinglabeling of the consequences) leading to the so called “relevance feedback”. The main aim is to figureout the representation of the image which are based on its content, and then to find a relation between this semantic and therepresentation we associate to the image. Machine learning techniques such as active learning have been effectively improved order to deal with image retrieval distributed over a network[10],[11]. The best enhancement was done with this is the introduction of relevance feedback into the process[3]. The major part of CBIR computation being devoted to the processing of the image descriptors, the fact that images are spread over various sources should be more an advantage than a drawback since it means a possible paralleling. In these system, the links between peers of the network are optimizedso as to propagate the query to relevant hosts. Here the smart cooperation is taken among the interactive CBIR and a localization learning based on mobile agents.Along this we are also usin one more concept of K-NN algorithm. K Nearest Neighbour (i.e, K-NN) is an algorithms which is simple to understand but works incredibly well in practic

II. LITERATURE SURVEY

The internet consists of huge volume of information. The search engines have been developed to tackle the search into these collections in find the best localizations of data’s matching a query. J. Cho, S. Roy et al. explains the complete overview of the search engines in the paper [1]. If we consider the data mining in multimedia documents such as image,

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2016

audio, video etc, web search engines usually give poor results. The web search engines use the contextual web page, or the meta information attached to the multimedia objects. The text used for the indexing process is often far from the semantic meaning that the user usually attaches to the content of the document. Hence, the results of web search engines are far from expected regarding the semantics of the documents.
So to overcome all the above deficiencies related to the multimedia data search such as for image search, Content based image retrieval (CBIR) is proposed. R. Veltkamp et al. explains the logic behind the content based image retrieval [2], in that the main idea is to build the representation of an image based on its content and then find the relation between this representation and the semantic we associate to the image.. If the query fails then user has to improve the query with only available features in the descriptor and the user's knowledge cannot be exploited by the system.
With the introduction of relevance feedback [3] the major improvements can be done in CBIR framework. M.E.J. Wood, N.W. Campbell and B. T. Thomas et al. presents how a query system can exploit the user's knowledge to a higher extent by employing relevance feedback to iteratively refine queries at run-time.Various relevance feedback techniques have been applied in content based image retrieval. Thomas S. Huang and Xiang Sean Zhou et al. [4] gives a brief review on existing techniques.There are many CBIR systems are developed. Out of which Blobworld [5] is one of the system for image retrieval based on finding coherent image regions which roughly correspond to objects. Each image is automatically segmented into regions (\blobs") with associated color and texture descriptors. Querying is based on the attributes of one or two regions of interest, rather than a description of the entire image.
David G. Lowe et al. describes [6] image features that have many properties that make them suitable for matching differing images of an object or scene.The paper also presents a method for extracting distinctive invariant features from images that can be used to perform reliable matching between different views of an object or scene. V. Roth, U. Pinsdorf, and J. Peters et al. presents [7] the resulting scenario of CBIR using mobile agents in which it launch several mobile agents with a copy of the query. They crawl the network in search of image collections. When an agent gets to a site, a dialog with a local agent is established. The local agent indexes the images and performs the processing.The ant-agents move from one peer to another and mark the visited hosts. The host is marked by changing a numerical value locally stored on these hosts, called marker. These markers can be viewed as a collective memory of paths leading to the relevant sites. This behavior-based mapping of the network is well adapted to inconsistent networks such as peer-to-peer networks, since the marked paths evolve with the global trend of the agent movements [8]

III. CONTENT BASED IMAGE RETRIEVAL SYSTEM

The Internet or peer-to-peer networks provides huge volumes of Information and to search these information search engines have been developed in order to find the best localizations of data matching a query[1]. Content-based Image Retrieval find similarity among images using Computer-based methods, where there is no human interaction to evaluates images[10]. These methods try to estimated the similarity of visually perceivable media objects exclusively from their visual content. They can be applied on media objects if they are represented in digitized, uniform, computable media chunks. The main work of CBIR is to find similar images without taking into account the similarity between their contained objects. Moreover, the similarity is measured by visual descriptors without considering any semantic information like text, title, tags, location, date or comments. For many applications these meta-information are very useful to increase the precision of finding similar images [EGoNaN99]. Figure 1. shows an example of similar images retrieved by a CBIR system. The question is why are these images similar? For a human user the images of a sunset, a candle and a firework are not similar at all, but for an Image Retrieval system they are. From an abstract point of view, there is a bright yellow round object located in the middle of a dark background. Therefore, the images would be highly similar. This difference in similarity assessment comes from different levels of perception. CBIR systems would judge these images as highly similar: Their content is similar in colors and textures of the foreground objects[2].

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2016

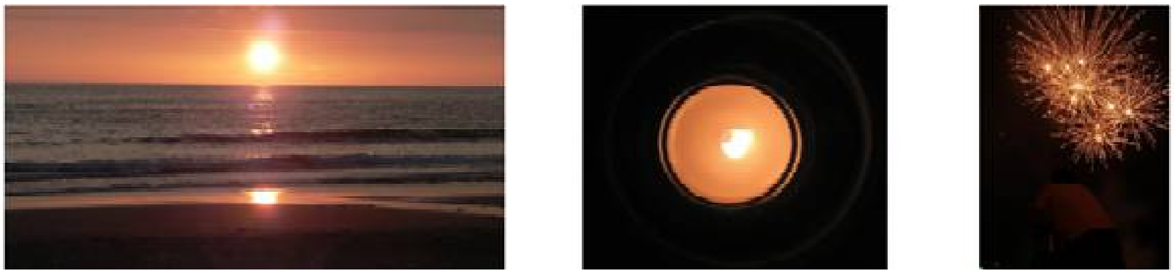

Figure 1. Similar Images according to CBIR

In typical content-based image retrieval systems as shown in figure 2.the visual contents of the images in the database are pull out and described by multi-dimensional feature vectors. The feature vectors of the images in the database form a feature database. To retrieve images, users provide the retrieval system with an example images or any sketched image.The system then changes these examples into its internal representation of feature vectors. The similarities or the distances between the feature vectors of the query example or sketch and those of the images in the database are then premeditated and retrieval is performed with the aid of an indexing scheme. The indexing scheme offers an efficient way to explore for the image database. Recent retrieval systems have combined users' relevance feedback to modify the retrieval process in order to generate perceptually and semantically more meaningful retrieval results[4].

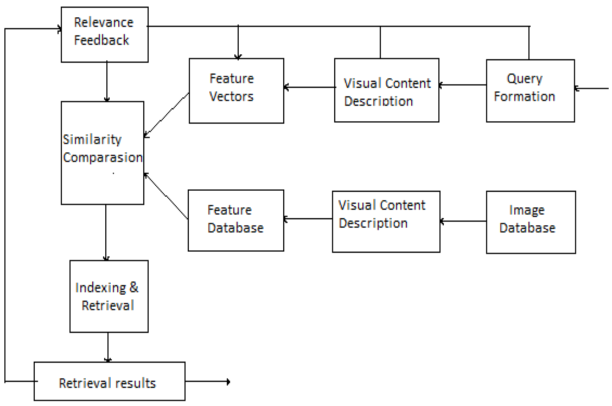

Figure 2. Diagram for content-based image retrieval system

An example image serves as a query to the system and all similar images are retrieved as outcome. CBIR system is not only used for image retrieval purpose but also to organize the content of common intent image databases of a digital library. By comparing pairs of images and assigning a similarity measure to every pair a CBIR system selects images from a dataset. The comparison is made through automatically extracted features from each image. The feature extraction may be carried out for a complete image or for image sub parts (sub-images). By using standard grid, like 3x3, 4x4 or more pixels, sub-images can be created, to split the image into square shaped sub-sections. Superior outcome can be produced using more section (6x6, 10x10) but the execution speed becomes very time-consuming[12].

IV. K-NN ALGORITHM

K Nearest Neighbour (i.e, K-NN) is an algorithms which is simple to understand but works incredibly well in practice. K-NN is a non parametric lazy learning algorithm It isalso versatile and its applications range from vision to proteins to computational geometry, graphs and so on. K-NN is one of the top 10 mining algorithm**.** That is a pretty concise statement. When you say a technique is non parametric, which means that it does not make any assumptions on the underlying information distribution. As in the real world, most of the practical information does not obey the typical

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2016

theoretical assumptions made for example the Gaussian mixtures, linearly separable, this algorithm is very useful. Non parametric algorithms like K-NN come to the rescue here.

It is a lazy algorithm which means it does not use the training data points to any *generalization* we can also say that, there is *no explicit training phase* or it is very minimal. This means training phase is pretty fast. Lack of generalization means that K-NN keeps all the training data. More exactly, all the training data is needed during the testing phase. (Well this is an exaggeration, but not far from truth). Other techniques like SVM where you can discard all non-supporting vectors without any problem are contrast to this. Most of the lazy algorithms – especially K-NN – make decision based on the entire training data set (in the best case a subset of them).The dichotomy is pretty obvious here – There is a minimal training phase but a costly testing phase. The cost is in terms of both memory and time. More time might be needed as in the worst case; all data points might take point in decision. More memory is needed as we need to store all training data. In pattern recognition, the *k*-nearest neighbour algorithm (*k*-NN) is a non-parametric method for classifying objects based on closest training examples in the feature space. *k*-NN is a type of lazy leazy learningor instance-based learning, where the function is only approximated locally and all computation is deferred until classification.

The *k*-nearest neighbour algorithm is amongst the easiest and simplest of all machine learning algorithms: an object is classified by a majority vote of its neighbours, with the object being assigned to the class most common amongst its *k* nearest neighbours (*k* is typicallya small positive integer). If *k* = 1, then the object is simply assigned to the class of its nearest neighbour.An equivalent method can be used for regression, by simply assigning the property value for the object to be the average of the values of its *k* nearest neighbours. It can be very useful to weight the contributions of the neighbours, so that the closer neighbours contribute more to the average than the more distant ones.

The value of property is known from these sets of objects the neighbours are taken for the correct classification or, in the case of regression, the value of. This can be thought of as the training set for the algorithm, though no explicit training step is required. The *k*-nn algorithm is very sensitive to the local structure of the data.The decision boundary is implicitly calculated by Nearest neighbour rules. It is also probable to compute the decision boundary explicitly, and to do so efficiently, so that the computational complexity is a function of the boundary complexity.Nearest neighbour problem has been widely studied in the field of Computational geometry under the name closest pair of point's problem.

Example:-

Consider the following example in figure 3.of *k*-NN classification. The test sample (yellow circle) should be classified either to the first class of green squares or to the second class of red triangles. If *k = 5* (dashed line circle) it is assigned to the first class because there are three squares and only two triangles inside the outer circle. If *k = 3* (dark line circle) it is assigned to the second class (two triangles and only one square inside the inner circle).

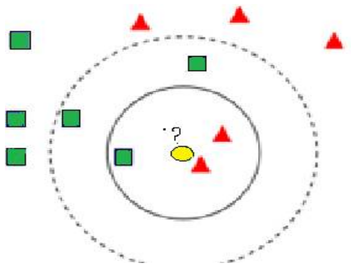

Figure 3. k-NN by example

The training examples are vectors in a multidimensional feature space, each having a class label. The training phase of the algorithm consists only class labels of the training of storing the feature vectors.In the classification phase, *k* is a user-defined constant, which is generally positive and a small integer and an unlabelled vector (a query or test point) is classified by passing on the label which is most frequent among the *k* training samples nearest to that query point.Rote classifier is one of the simplest and rather trivial classifiers is the, which memorizes the entire training data and performs classification only if the attributes of the test object match one of the training examples exactly. A more sophisticated approach, *k*-nearest neighbour (*k*-NN) classification, finds a group of *k* objects in the training set that are

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2016

closest to the test object, and bases the assignment of a label on the predominance of a particular class in this neighbourhood. In general, there are three key elements of this approach: the first is a set of labelled objects, i.e, a set of stored records, the second is a distance or similarity metric to compute distance between objects, and the third is value of *k*, the number of nearest neighbours. To order an unlabelled object, the distance of this object to the labelled objects is considered, its *k*-nearest neighbours are identified, and the class labels of these nearest neighbours are then used to regulate the class label of the object.

Following line provides a high-level summary of the nearest-neighbour classification method. Given a training set *T*and a test object *x* = *(***x**'*, y')*, the algorithm calculates the distance (or Similarity) between zand all the training objects *(***x***, y)* ∈*T*to determine its nearest-neighbour list, *Tz* (**x** is the data of a training object, while *y* is its class. Likewise, **x**' is the information of the test object and *y*' is its class).Once the nearest-neighbour list is obtained, the test object is classified based on the majority class of its nearest neighbours:

Majority Voting: $y' = argmax \sum_{(xi,yi) \in Tz} I(c = yi)$,

where c is a class label, c_i is the class label for the i^{th} nearest neighbours, and I (·) is an indicator function that returns the value 1 if its argument is true and 0 otherwise.

Input:*T*, the set of k training objects, and test object *z= (x'.y')*

Process: Compute *d (x',x)*, the distance between *z* and every object, *(x,y)*∈*T.*

Select *Tz*⊆*T*, the set of *k* closest training objecttoz.

Output: $y' = argmax \sum_{(xi,yi) \in Tz} I(c = yi)$

V. MOBILE AGENTS

The term software agents refer to software programs that perform certain tasks as given by the user or the client host. Software agents are mainly classified as static agents and mobile agents. Static agents achieve the goal by executing on a single machine. Mobile agents migrate from one host to another and executes on several machines. Mobility increases the functionality of the mobile agent. Mobile agents have the ability to move from host to host, executing at each place and then keeping the results before moving to the next server.To perform the required parallelization of feature vector processing, we have chosen to use mobile agents. The working of mobile agents is classified like this. A mobile agent consists of the program execution state and the program code. Initially mobile agents are processes dispatched from a source host to accomplish a specified task. Mobility rises the functionality of the mobile agent. A mobile agent consists of the program code along with the program execution state. Initially a mobile agent resides on a computer called the home machine[7]. The agent is then forwarded to execute on a remote computer called a mobile agent host. When a mobile agent is dispatched the complete code of the mobile agent and the execution state of the mobile agent is transferred to the host. The host gives a suitable execution environment for the mobile agent to execute. Another feature of mobile agent is that it can be cloned itselfto execute on several hosts. Upon completion, the mobile agent gives the results to the sending client or to another server.

Aglet Technology is a framework for programming mobile network agents in Java which is developed by the IBM™ Japan research group. The IBM's mobile agent is known as 'Aglet',which is a lightweight Java object. An aglet can be dispatched to any remote host that supports the Java Virtual Machine. This needs from the remote host to pre-install Tahiti, a small aglet server program implemented in Java and provided by the Aglet Framework. The IBM Aglet team provided the so-called "FijiApplet" in order toallow aglets to be fired from within applets,, an abstract applet class that is part of a Java package called "Fiji Kit". FijiApplet maintains some sort of an aglet context. From within this system, aglets can be created[14], dispatched from and retracted back to the FijiApplet.

The reason for using mobility is to improve performance which can be achieved by moving the agent closer to the new host, where it can use the services locally. Agent needs data from several host situated on different platforms. It uses remote procedure call (RPC) where it can request the required data and obtain the results by invoking the remote methods. This RPC follows the client-server paradigm. But if the volume of data is large it can create bandwidth and network traffic problem. In such cases the mobile agent can migrate to those remote hosts and perform the functions locally and come back with the desired results. It would be a more efficient way to process the data. The ability of an agent to migrate from one environment to another is not a requirement for agent hood. Still mobility is an important property for many agent-based systems and necessary for a certain class of application. The basic architecture of the mobile agent can be thought of as a client sends out an agent who crwal the network visiting servers in order to perform some required action . The Mobile agent which provides a new design model for applications as compared to the

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2016

traditional client server model, the mobile agent blows apart the very notion of server and the client. With mobile agents, the flow of control actually moves across the network, in place of the request/response architecture of client-server. In effect,each and every node is a server in the agent network, the agent moves to the location where it may discover the services it needs to run at each point in its execution .

A. ***GENERIC MOBILE AGENT SYSTEM ARCHITECTURE:***

In order to make a mobile agent system work, it is not enough to build the agents themselves. A program at each site is also needed to handle the incoming agents and send out agents. This program is often called an agency. The agency be built differently depending on which type of agent system is needed, [13]but a general architecture can be seen in figure 4.

The generic mobile agent system can have a range of varying components. It needs a communication module that handles incoming and outgoing agents, as well as the messaging between non-local agents. It has a repository that performs authentication, sets priorities and queues up agents for later execution. The executing module has an interpreter and can sometimes run agents written in different languages[15]. The state engine contains the current state of the agency and can have some kind of rule or inference engine that decides what to do with the agents. It also handles local inter-agent communication. There is also some kind of database or directory where data are stored or retrieved by the agents. The security module acts as a kind of sandbox that keeps track of what the agents are allowed to do. It also monitors the agency. There can, of course, be security functionality in the other modules too, such as encryption in the communication module

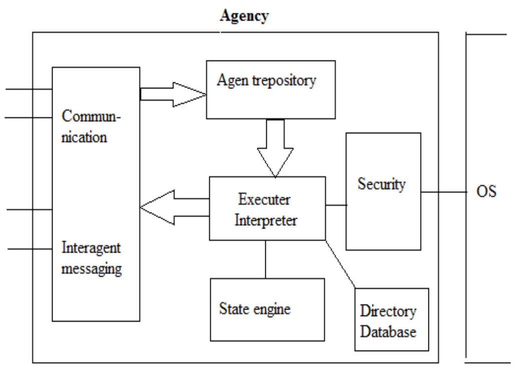

Figure 4: Generic mobile agent system architecture

B. ANT-LIKE AGENTS:

In the case of distributed retrieval, ant-agents travel the network to find the relevant documents. These ant-agents travel over the network by moving from one host to another host and mark the visited host by changing a numerical value which are locally stored on these hosts[9]. These values are known as marker. Software agents travel over the network starting from the client host to next host in order to search the desired information. Now how the “ant” agent could deposit “pheromone-like” markers? How to specify the searched information? These markers can be viewed as a collective memory of paths leading to the relevant sites. Since the marked paths evolve with the global trend of the agent movements, this behavior-based mapping of the network is well adapted to inconsistent networks such as peer-to-peer networks[8],[10]. We have to do several travels between the user’s computer and the information sources in our distributed CBIR context. Ant-algorithm seems to be a good solution for learning the relevant paths through the dynamics of active learning[11].

VI. Architecture Of Our System

The figure 5. shows the architecture of the existing system.In this architecture user gives the query, which means the user can provide the input for the system. After accepting the input from the user, the system can divide the query according to color, texture and shape. Then from that input the system builds the similarity function by using K-NN algorithm. Then this input function is transferred to the mobile agent. Mobile agent accepts the input function from the

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2016

system. Then mobile agent creates its multiple copies of itself with the same query. Then after creating multiple copies of mobile agent, the mobile agents' launches along with the network and by applying the k-nn they find out the resultant image over the network. After getting the resultant image over network, the mobile agent comes back and returns the result to the user.

Although the basic k-NN algorithm and some of its variations, such as weighted k-NN and assigning weights to objects, are relatively well known, some of the more advanced techniques for k-NN are much less known. For example, it is typically possible to eliminate many of the stored data objects, but still retain the classification accuracy of the k-NN classifier. This is known as 'condensing' and can greatly speed up the classification of new objects. In addition, data objects can be removed to improve classification accuracy, a process known as "editing".There has also been a considerable amount of work on the application of proximity graphs (nearest neighbor graphs, minimum spanning trees, relative neighborhood graphs, Delaunay triangulations, and Gabriel graphs) to the k-NN problem. Recent papers by Toussaint, which emphasize a proximity graph viewpoint, provide an overview of work addressing these three areas and indicate some remaining open problems.Other important resources include the collection of papers on it.

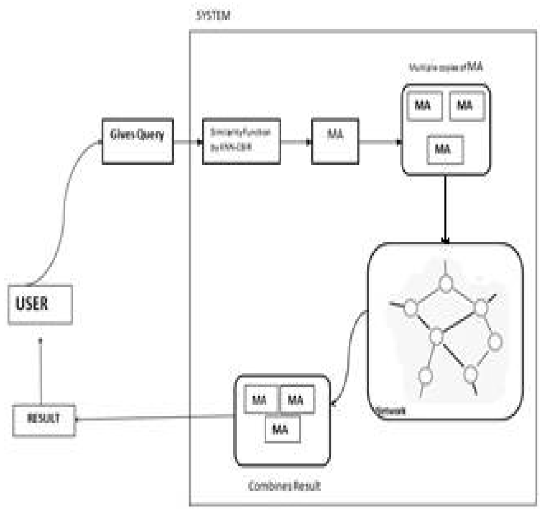

Figure 5. Proposed system architecture

VII. SIMULATION RESULTS

As shown in the following figures shows the implementation of our system:

In the figure 6,here we are searching for a mobile agent.the mobile has been searched and it has been initiated for the intended task given by the user.In figure 7 we are applying a K-NN algorithm in order to extract the features of the images. After applying K-NN algorithm system wiil ask to select a image to be searched . In figure 8 it shows a window for the selection of image. After the image selection the mobile agent travel over the network and search for the similar images in the database. When the search has been completed it will shows all the similar images as an output which are similar to the inout image, as shown in figure 9.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2016

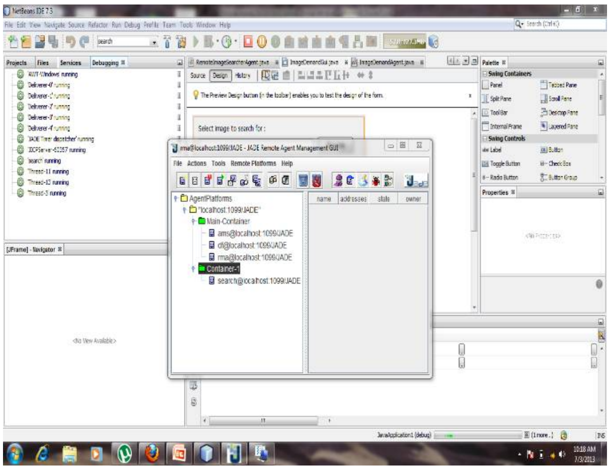

Figure 6. Search Mobile agent

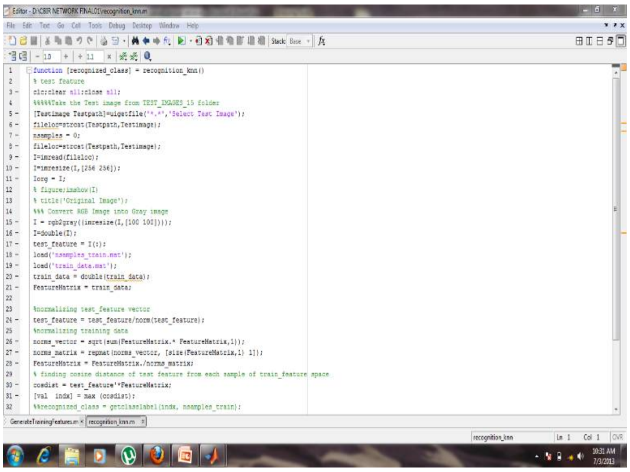

Figure 7. Applying K-NN

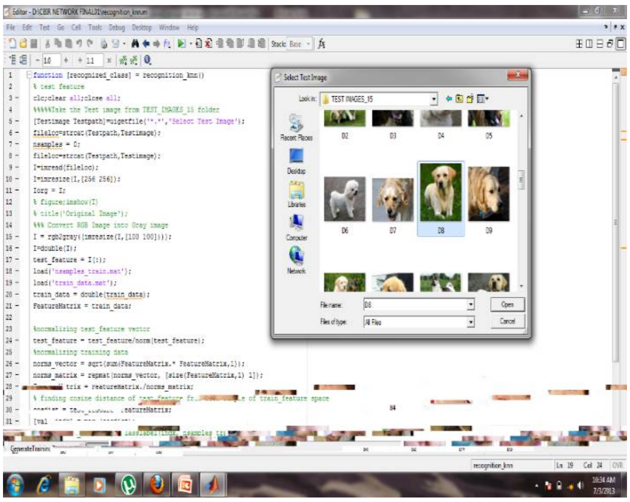

Figure 8. Giving System Input

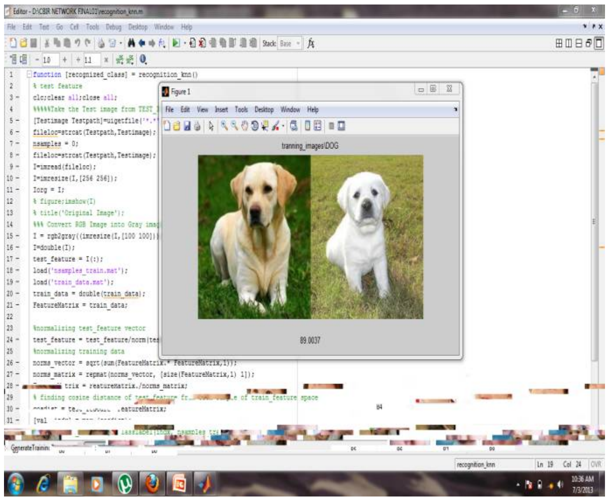

Figure 9. Displaying Result

VIII. CONCLUSION

In this paper a new approach for image retrieval is proposed It provides an overview of the Distributed Content based Image Retrieval System by using mobile agent technology along with a K-NN algorithm, which gives us an efficient and faster image retrieval of images over a huge metadata network. The use of the K-NN algorithm shows that the performance of the system has improved in order to retrieve the similar images from the huge database.

REFERENCES

1. J. Cho and S. Roy, “Impact of search engines on page popularity,” in WWW’04:Proc. 13th Int. Conf. WorldWideWeb, New York, 2004, pp.20–29.
2. R. Veltkamp, “Content-Based Image Retrieval System: A Survey” Univ. Utrecht, The Netherlands, 2002, Tech. Rep.
3. M. Wood, N. Campbell, and B. Thomas, “Iterative refinement by relevance feedback in content-based digital image retrieval,” in ACM Multimedia98, Bristol,UK, Sep. 1998, pp. 13–20.

4. T. Huang and X. Zhou, “Image retrieval with relevance feedback: From heuristic weight adjustment to optimal learning methods,” in Int. Conf.Image Processing (ICIP’01), Thessaloniki, Greece, Oct. 2001, vol. 3, pp. 2–5.
5. C. Carson, M. Thomas, S. Belongie, J. Hellerstein, and J. Malik, “Blobworld: A system for region-based image indexing and retrieval,” in Third Int. Conf. on Visual Information Systems, June 1999.
6. D. Lowe, “Distinctive image features from scale-invariant keypoints,” Int. J.Comput. Vis., vol. 20, pp. 91–110, 2003.
7. V. Roth, U. Pinsdorf, and J. Peters, “A distributed content-based search engine based on mobile code,” in SAC ’05: Proc. 2005 ACM Symposiumon Applied Computing, New York, 2005, pp. 66–73.
8. A. Revel, “Web-agents inspired by ethology:A population of “ant”-like agents to help finding user-oriented information,” in IEEE WIC’2003:Int. Conf. WebIntelligence, Halifax, NS, Canada, Oct. 2003, pp. 482–485
9. J. Callan, “Distributed Information Retrieval”. Norwell, MA: Kluwer, 2000, pp. 127– 150, ir 5.
10. King, C. H. Ng, and K. C. Sia, “Distributed content-based visual information retrieval system on peer-to-peer networks,” ACM Trans. Inform. Syst., vol. 22, no. 3, pp. 477–501, 2004.
11. Picard, M. Cord, and A. Revel,”Image Retrieval Over Networks: Active Learning Using Ant Algorithm” IEEE TRANSACTIONS ON MULTIMEDIA, VOL. 10, NO. 7, NOVEMBER 2008.
12. Sabu .M Thampi1, Dr. K. Chandra Sekaran2,” Mobile Agents For Content-Based Www Distributed Image Retrieval”, International Conference on Human Machine Interfaces 20-23, Dec. 2004
13. Sabu .M Thampi ,Dr. K. Chandra Sekaran “Content Based Image Retrieval Using Mobile Agents and Steganography”
14. Danny B.Lange and Mitsuru Oshima, “Programming and Developing Java Mobile Agents with Aglets”
15. Rakesh Kumar, Dr.Mayank Dave,” Mobile Agent as an Approach to Improve QoS in Vehicular Ad Hoc Network”, IJCA Special Issue on “Mobile Ad-hoc Networks” MANETs, 2010