IMAGE RETRIEVAL USING CONTOUR FEATURES WITH A ROUGH SET METHOD

PHEERAWIT WASINPHONGWANIT

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR
THE DEGREE OF MASTER OF SCIENCE
(TECHNOLOGY OF INFORMATION SYSTEM MANAGEMENT)
FACULTY OF GRADUATE STUDIES
MAHIDOL UNIVERSITY
2010

Thesis
entitled

IMAGE RETRIEVAL USING CONTOUR FEATURES WITH A ROUGH SET METHOD

……………….………….…..………
Mr. Pheerawit Wasinphongwanit
Candidate

……………………….…….…..………
Asst. Prof. Pisit Phokharatkul,
D.Eng. (Electrical Engineering)
Major-advisor

……………………….…….…..………
Lect. Songpol Ongwattanakul,
Ph.D. (Computer Engineering)
Co-advisor

……………………….…….…..………
Asst. Prof. Rangsipan Marukatat,
Ph.D. (Computer Science)
Co-advisor

……………………….…….…..………
Prof. Banchong Mahaisavariya,
M.D., Dip Thai Board of Orthopedics
Dean
Faculty of Graduate Studies
Mahidol University

……………………….…….…..………
Asst. Prof. Rawin Raviwongse,
Ph.D. (Engineering Management)
Program Director
Master of Science Program in
Technology of Information System
Management
Faculty of Engineering
Mahidol University

Thesis
entitled

IMAGE RETRIEVAL USING CONTOUR FEATURES WITH A ROUGH SET METHOD

was submitted to the Faculty of Graduate Studies, Mahidol University
for the degree of Master of Science
(Technology of Information System Management)
on

Aug 10, 2010

……………….…………….…..………
Mr. Pheerawit Wasinphongwanit
Candidate

…………………….…..………………
Lect. Noppadol Wanichworanant,
Ph.D. (Electrical Engineering)
Chair

…………………….…..……………...…
Asst. Prof. Pisit Phokharatkul,
D.Eng. (Electrical Engineering)
Member

…………………….…..……….……
Lect. Songpol Ongwattanakul,
Ph.D. (Computer Engineering)
Member

…………………….…..……………...…
Assoc. Prof. Chom Kimpan,
D.Eng. (Electrical Engineering)
Member

…………………….…..………………..
Asst. Prof. Rangsipan Marukatat,
Ph.D.(Computer Science)
Member

…………………….…..……………...…
Prof. Banchong Mahaisavariya,
M.D., Dip Thai Board of Orthopedics
Dean
Faculty of Graduate Studies
Mahidol University

…………………….…..……………..…
Asst. Prof. Rawin Raviwongse,
Ph.D. (Engineering Management)
Dean
Faculty of Engineering
Mahidol University

ACKNOWLEDGEMENTS

The success of this thesis would not have been succeeded without the support, hard work and efforts of a number of people. First and foremost, I would like to express my sincere gratitude to my major advisor, Asst. Prof. Dr. Pisit Phokharatkul for his supervision and valuable guidance to this research.

I also would like to express my gratitude to Asst. Prof. Dr. Rangsipan Marukatat and Dr. Songpol Ongwattanakul, my co-advisor for much help with the useful suggestions and methodology. I sincerely thank to Assoc. Prof. Dr. Chom Kimpan, the external examiner of the thesis defense and Dr. Noppadol Wanichworanant, the chair of thesis defense for their kindness in valuable suggestion and many helpful comments.

Besides, thank to my friends and staffs in Master of Science Program in Technology of Information System Management, Mahidol University.

Finally, this thesis would not have been completed without understanding and support of my family, especially my parents for full of support, entirely care, love and encouragement through my life. I hope that this thesis will usefully for the processes of data mining for classification. The usefulness of this thesis. I dedicate to my father, my mother and all the teachers of me.

Pheerawit Wasinphongwanit

IMAGE RETRIEVAL USING CONTOUR FEATURES WITH A ROUGH SET METHOD

PHEERAWIT WASINPHONGWANIT 4837226 EGTI / M

M.Sc. (TECHNOLOGY OF INFORMATION SYSTEM MANAGEMENT)

THESIS ADVISORY COMMITTEE : PISIT PHOKHARATKUL, D.Eng. (ELECTRICAL ENGINEERING), SONGPOL ONGWATTANAKUL, Ph.D. (COMPUTER ENGINEERING), RANGSIPAN MARUKATAT, Ph.D. (COMPUTER SCIENCE)

ABSTRACT

Content based image retrieval (CBIR) is well-known in the field of image retrieval. It uses the analysis of the content of an image from image processing to retrieve images that users look for from an image search. In this paper, we used binary images for demonstration. Fourier descriptors were used to analyze testing and training images in the preprocessing step. Fourier coefficients were quantized into multiple attributes, and rough set theory was used to generate a rule-based system. Rough set theory was used as a data mining technique. When comparing similarity measurements and rough set data mining in the image retrieval results, usage times, and usage memories are all advantageous in rough set data mining. Precision and accuracy are almost the same in both methods. Similarity measurement has an advantage in recall.

KEY WORDS: ROUGH SET / IMAGE RETRIEVAL / FOURIER DESCRIPTOR

68 Pages

การค้นคืนภาพโดยใช้รูปร่างของขอบภาพด้วยวิธีรัฟเซต

IMAGE RETRIEVAL USING CONTOUR FEATURES WITH A ROUGH SET METHOD

พีรวิช วศินพงศ์วณิช 4837226 EGTI / M

วท.ม. (เทคโนโลยีการจัดการระบบสารสนเทศ)

คณะกรรมการที่ปรึกษาวิทยานิพนธ์ : พิศิษฏ์ โภคารัตน์กุล, D.Eng. (ELECTRICAL ENGINEERING), ทรงพล องค์วัฒนกุล, Ph.D. (COMPUTER ENGINEERING), รังสิพรรณ มฤคทัต, Ph.D. (COMPUTER SCIENCE)

บทคัดย่อ

การค้นคืนภาพโดยใช้คอนเทนท์ในภาพเป็นที่ทราบกันอย่างแพร่หลายในสาขาการค้นคืนภาพ การค้นคืนภาพจะใช้คอนเทนท์ในภาพจากกระบวนการการประมวลผลภาพและการวิเคราะห์ภาพในการสืบค้นหาภาพที่ผู้ใช้ต้องการค้นหา ในงานวิจัยนี้ ได้ใช้ภาพที่เป็นไบนารี่ในการทดลอง โดยผู้ทดลองได้นำตัวอธิบายฟูเรียร์มาใช้ในการวิเคราะห์ภาพที่ใช้ในการทดสอบและเรียนรู้ในขั้นตอนการทำประมวลผลล่วงหน้า จากนั้นค่าสัมประสิทธิ์ของตัวอธิบายฟูเรียร์จะถูกควอนไทเซชั่นให้เป็นค่าแบบหยาบแล้วนำทฤษฎีรัฟเซตมาใช้ในการสร้างระบบกฎเกณฑ์ ทฤษฎีรัฟเซตถูกนำมาใช้เสมือนเทคนิคการทำเหมืองข้อมูล ส่วนการเปรียบเทียบระหว่างผลที่ได้จากการทดลองในการวัดความเหมือนและเทคนิคการทำเหมืองข้อมูลโดยใช้รัฟเซตในการสืบค้นคืนภาพนั้น จำนวนเวลาและหน่วยความจำที่ใช้นั้นทฤษฎีรัฟเซตมีข้อได้เปรียบกว่าการวัดความเหมือน ค่าความแม่นยำและค่าความถูกต้องนั้นมีความแตกต่างกันเพียงเล็กน้อย ค่าการค้นคืนการวัดความเหมือนมีค่าสูงกว่ารัฟเซต

68 หน้า

CONTENTS

CONTENTS (cont.)

CONTENTS (cont.)

LIST OF TABLES

LIST OF FIGURES

CHAPTER I
INTRODUCTION

1.1 Background and Problem Statement

We use a database system for storage information. The type of the information such as images, texts, music and movies that a difference kind of encoding and file types. The data images can use some methods of image retrieval technologies for query the database of digital images. The most traditional and common methods of image retrieval utilize some methods of adding metadata such as captioning, keywords or descriptions to the images. The content based image retrieval is a method that uses the content of digital image for querying image. The contents of images (such as color, shape, texture etc.) are used to process in numerical computing of digital image retrieval. For example, we use a color for classify the groups or species of flowers. We have many features of colors of each image and we can not use all color features to process in all digital images. We might be used all color features to process it but we use over time for processing. However, we can use the data mining for solve this problem.

Data mining is the process of automatically discovering useful information in large data repositories. Data mining techniques are deployed to scour large databases in order to find novel and useful patterns that might otherwise remain unknown. They also provide capabilities to predict the outcome of a future observation, such as predicting whether a newly arrived customer will spend more than $100 at a department store [1].

In the field of classification we can use the rough set theory for data mining technique apply to image retrieval. This method use the features (such as color, texture, shape and layout) of images data, which are done preprocessing before, for mining images data and classify images data that receive from resources. The rough set theory makes the rule sets from data for classifying. We not necessary to use all data to processing but we can use some data that unique from other for classifying and processing.

1.2 Objectives

1.2.1 To study the theory and method for a classification of rough set theory.

1.2.2 To create and develop image retrieval system using rough set theory.

1.3 Scope of Work

1.3.1 To retrieve an image by rough set theory.

1.3.2 To use general PNG files as the only binary image.

1.3.3 To test a method on MATLAB of 3000 binary images.

1.4 Step of Work

1.4.1 Study details of a theory and method for the rough set theory.

1.4.2 Collect images for testing.

1.4.3 Analysis and design of the rough set theory apply to image retrieval.

1.4.4 Develop the rough set for data mining.

1.4.5 Test and debug the rough set data mining.

1.4.6 Conclusion and recommendation.

1.5 Expected Results

1.5.1 The theory and method for classification of rough set theory apply to image retrieval are clearly understood.

1.5.2 Image retrieval based on rough set data mining is developed in MATLAB.

CHAPTER II
LITERATURE REVIEW

2.1 An overview of content based image retrieval

Content-based image retrieval (CBIR), also known as query by image content (QBIC). The content-based visual information retrieval (CBVIR) is the application of computer vision to the image retrieval problem, which is a problem of searching for digital images in large databases. The content of image might refer colors, shapes, textures or any other information that can be derived from image itself.

The term CBIR seems to have originated in 1992, when it was used by T. Kato to describe experiments into automatic retrieval of images from a database, based on the colors and shapes present. Since then, the term has been used to describe the process of retrieving desired images from a large collection on the basis of syntactical image features [2].

2.2 Feature extraction technique for CBIR [3]

The commonest features used are the mathematical measurement of color, texture or shape.

2.2.1 Color retrieval

Each image added to the collection is analyzed to compute a color histogram which shows the proportion of pixels of each color within the image. The color histogram for each image is then stored in the database. At search time, the user can either specify the desired proportion of each color (75% olive green and 25% red, for example), or submit an example image from which a color histogram is calculated.

2.2.2 Texture retrieval

The ability to retrieve images on the basis of texture similarity may not seem very useful. But the ability to match on texture similarity can often be useful in distinguishing between areas of images with similar color (such as sky and sea, or leaves and grass). A variety of techniques has been used for measuring texture similarity; the best established rely on comparing values of what are known as second-order statistics calculated from query and stored images. Essentially, these calculate the relative brightness of selected pairs of pixels from each image.

2.2.3 Shape retrieval

Shape is a fairly well defined concept and there is considerable evidence that natural objects are primarily recognized by their shape. A number of features characteristic of object shape (but independent of size or orientation) are computed for every object identified within each stored image. Queries are then answered by computing the same set of features for the query image, and retrieving those stored images whose features most closely match those of the query. Two main types of shape feature are commonly used global features such as aspect ratio, circularity and moment invariants and local features such as sets of consecutive boundary segments.

From these techniques of feature extraction, we can see that the features have various data and large size. In searching technique, if data are very large that the problem about the time. We can use the rough set data mining technique to solve this problem and optimize some data for searching and classification data.

2.3 Fourier descriptor

Fourier Descriptor is the most techniques for extracting the contents of image contour. The coordinate boundaries will be transformed to complex value to calculate the Fourier coefficients. From the boundaries of images, we can make sure that the image invariant for translation and rotating by moving the centroid to origin (0,0) as following:

$$x_i' = x_i - x_c \text{ and } y_i' = y_i - y_c \text{ with } i = 1, \ldots, L \qquad (1)$$

Where L denotes the number of boundary points.

x_c is coordinate x of centroid.

y_c is coordinate y of centroid.

The discrete Fourier Transform of image contour $C_{(x,y)}$ is given by:

$$a_k = \frac{1}{L}\sum_{m=0}^{L-1} x_m e^{-j2\pi km/L} \qquad (2)$$

$$b_k = \frac{1}{L}\sum_{m=0}^{L-1} y_m e^{-j2\pi km/L} \qquad (3)$$

Where k = 0, 1, …, L-1

a_k and b_k have to be changed to polar form and normalize.

$$r_k = \sqrt{|a_k|^2 + |b_k|^2} \qquad (4)$$

From this polar form, cut DC component (r_0) and normalize.

$$s_k = \frac{r_k}{r_1} \quad \text{with } k = 1, \ldots, L-1 \qquad (5)$$

Fourier Descriptor (s_k) is now invariant to translation, rotation and starting point.

2.4 Rough set theory [6]

The rough set methodology was introduced by Pawlak in the early 1980s as a mathematical tool to deal with uncertainty. It is a formal approximation of a crisp set in terms of a pair of sets which give the lower and the upper approximation of the original set. In the standard version of rough set theory (Pawlak 1991), the lower and upper approximation sets are crisp sets, but in other variations, the approximating sets may be fuzzy sets.

2.4.1 Information systems

An information system contains a data set represented in a table. Each row in the table represents a core, an event, a patient, or simply an object. Every column represents an attribute (a variable, an observation, a property, etc.) that can be measured for each object, the attribute may be also supplied by a human expert or user.

An information system, S, is defined as:

$$S = (U, A, \{V_a : \in A\}) \quad (6)$$

Where U is a non-empty finite set of objects,
A is a finite, nonempty set of mappings,
$a: U \rightarrow V_a$, and
V_a is the value set of a.

Example 1: To illustrate the concept of information systems, we present an example as a Table 2.1. There are five animals or objects, and four attributes (Size, Type, Color, and Flying).

Table 2.1 An example information system

	Size	Type	Color	Flying
X1	Small	Bird	White	Yes
X2	Medium	Bird	Black	Yes
X3	Medium	Cat	Brown	No
X4	Small	Bird	White	Yes
X5	Big	Horse	White	No

We interpret U as a set of objects (or situations) and A as a set of attribute mappings each of which assigns to an object a value which a may take under the respective attribute. This information is expressed by one distinguished attribute called decision attribute, the process is known as supervised learning. Information system of this kind is called decision systems. The information system in table 2.1 is also a decision system $\mathrm{S} = (\mathrm{U}, \{\mathrm{Size}, \mathrm{Type}, \mathrm{Color}\} \cup \{\mathrm{Flying}\})$ with decision attribute Flying.

2.4.2 Indiscernibility

One of the most important concepts of rough set theory is indiscernibility, which is used to define equivalence classes for the objects. Each subset of attributes $B \subseteq A$ defines an equivalence relation, denote IND(B) and will be called an indiscernibility relation. This indiscernibility relation is defined as:

$$\mathrm{IND}(\mathrm{B}) = \{(\mathrm{x}, \mathrm{y}) \in \mathrm{U} \times \mathrm{U} : \mathcal{F}(\mathrm{x}, \mathrm{a}) = \mathcal{F}(\mathrm{y}, \mathrm{a}) \text{ for every a} \in \mathrm{B}\} \qquad (7)$$

In equation 7, the subset of attributes B will define a partitioning of the universe into sets such that each object in a set cannot be distinguished from other objects in the set using only the attribute in B. The sets of the condition as mentioned on above are called equivalence classes. The family of all equivalence classes of the equivalence relation IND(B) is denoted U/IND(B).

Example 2: In the information system shown in Table 2.1, we see that objects X_1 and X_4 are indiscernible. The indiscernibility relations from the table 2.1 are as follows:

$\mathrm{IND}(\mathrm{Size}) = \{\{\mathrm{x}_1, \mathrm{x}_4\}, \{\mathrm{x}_2, \mathrm{x}_3\}, \{\mathrm{x}_5\}\}$

$\mathrm{IND}(\mathrm{Type}) = \{\{\mathrm{x}_1, \mathrm{x}_2, \mathrm{x}_4\}, \{\mathrm{x}_3\}, \{\mathrm{x}_5\}\}$

$\mathrm{IND}(\mathrm{Color}) = \{\{\mathrm{x}_1, \mathrm{x}_4, \mathrm{x}_5\}, \{\mathrm{x}_2\}, \{\mathrm{x}_3\}\}$

$\mathrm{IND}(\mathrm{Flying}) = \{\{\mathrm{x}_1, \mathrm{x}_2, \mathrm{x}_4\}, \{\mathrm{x}_3, \mathrm{x}_5\}\}$

$\mathrm{IND}(\mathrm{Size}, \mathrm{Type}, \mathrm{Color}, \mathrm{Flying}) = \{\{\mathrm{x}_1, \mathrm{x}_4\}, \{\mathrm{x}_2\}, \{\mathrm{x}_3\}, \{\mathrm{x}_5\}\}$

2.4.3 Decision systems

If we add the decision attributes into the information into the information systems, this system is called a decision system. The decision systems is defined (Bjanger, 1999)

$$S = (U, A \cup \{d\}) \quad (8)$$

Where $d \notin A$ is a distinguish attribute called the decision attribute,

The elements of A are called condition attributes, Since a decision system is a special kind of information system, the decision is not necessarily constant on the equivalence classes. That is, although the two objects belonging to the same equivalence class, the values of the decision attribute may be different. In this case, the decision system is inconsistent. The decision system is consistent if there exists any class $E \in A \cup \{d\}$ for which a unique classification can be made.

Example 3: The simple information system $S = (U, A)$ shown in Table 2.1 is a result of measuring the three characteristics Size, Color, and Flying of different animals.

Now, let us extend the information system as shown in Table 2.1. The additional object of animal has been stored to the table, which is shown in Table 2.2.

Table 2.2 An example of the extended information system

	Size	Type	Color	Flying
X1	Small	Bird	White	Yes
X2	Medium	Bird	Black	Yes
X3	Medium	Cat	Brown	No
X4	Small	Bird	White	Yes
X5	Big	Horse	White	No
X6	Medium	Bird	Black	No

From this example we can see some of the problems that rough set theory addresses. For instance, objects x_2 and x_6 belong to the same equivalence classes, but they have different values for the decision Flying attribute. This means that the information system in this example is inconsistent.

The objects are classified according to condition attributes $C = \{Size, Type, Color\}$, and the classification is represented as the decision attribute set

$D = \{Flying\}$. By removing the inconsistent attributes and their values. We can search a set of equivalence relations from the decision table, and the objects are represented in terms of object classes, each of which is supported by a number of examples, given in the rightmost column of the table. Thus, there are three object classes such as x_1 and x_4, x_3 and x_5 classes respectively.

2.4.4 Set approximation

Approximation spaces are the core mathematical concept of rough set data analysis, and their usage reflects the idea the granulation of information can be described by classes of and indiscernibility relation. Given an information system, $S = (U, A)$, and subset of attributes, $B \subseteq A$, we would like to approximate a set of objects, X, using only the information contained in B.

For $X \subseteq U$, and $B \subseteq A$ we associate two sets defined as follows (Lken, 1999):

$$B_*X = \{x|[x]_B \subseteq X\} \quad (9)$$

Is the lower approximation or positive region of X, and

$$B^*X = \{x|[x]_B \cap X \neq \emptyset\} \quad (10)$$

Is the upper approximation or possible region of X.

The lower approximation is the set containing all objects for which the equivalence class corresponding to the object, which is a subset of the set we would like to approximate. This set contains all objects, which with certainty belong to the set X.

The upper approximation is the set containing the objects for which the intersection of the object's equivalence class, and this set we would like to approximate is not the empty set. This set contains all objects, which possibly belong to the set X.

A set $BN_B(X) = B^*X - B_*X$ will be called B-boundary of X.

If $X \subseteq U$ is given by a predicate P and $x \in U$, then

- $x \in B_*X$ means that x certainly has property P,
- $x \in B^*X$ means that x possible has property P, and
- $x \in U\backslash B^*X$ means that x definitely does not have property P.

The area of uncertainly extends over $B^*X \setminus B_*X$, and the area of certainly is $B_*X \cup B^*X$, respectively. The concepts of lower approximation, upper approximation and B-boundary region are illustrated in Figure 2.1.

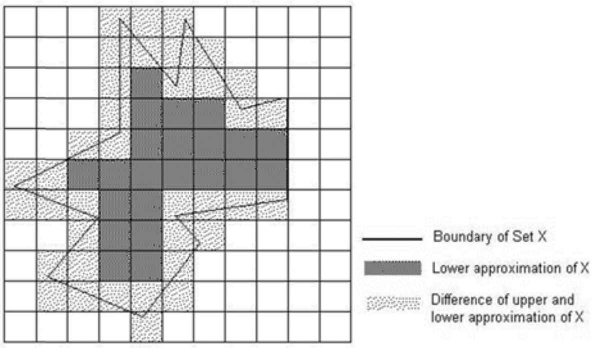

Figure 2.1 Rough set approximation

Example 4: Approximating the set of animal sizes in Table 2.1 that can fly is shown in Figure 2.2.

Figure 2.2 Approximation of the yes set

2.4.5 Core and reduction of attributes

The set of attributes in the information system describes objects in the system. We use the set of attributes to classify the objects into classes that are determined by anywhere from one to several attributes. Sometimes it happens that the set of attributes provides more information about objects then is needed to distinguish them. In such a situation some of the attributes may be reduced without losing the ability to group the objects. In the rough set approach, there are two fundamental concepts used in the reduction of knowledge, a reduct and the core. Intuitively, a reduct is one essential part of an information system, which can discern all objects that are discernible by the original information system. The core comprises the attributes that are indispensable to the discrimination of objects, the attributes that are contained in all the reducts. These concepts are formally defined in Definition 1 and 2 (Lken, 1999), but first it is necessary to introduce some auxiliary notions.

Definition 1: (dispensability) Given $S = (U, A)$, an attribute a is said to be dispensable in $B \subseteq A$ if $IND(B) = IND(B - \{a\})$, otherwise the attribute is indispensable in B. If all attribute $a \in B$ s are indispensable in B, then B is called orthogonal or independent.

Definition 2: (Reducts) Given an information system $S = (U, A)$, let $B \subseteq A$. A reduct of B is a set of attributes $B' \subseteq B$ such that all attributes $A \in B - B'$ are dispensable (B' is orthogonal), and $IND(B') = IND(B)$. The set of reducts of B is denoted $RED(B)$.

$$CORE(B) = \cap RED_{RED \in RED(B)} \quad (11)$$

From these definition, we can say that attributes $a \in B$ is superfluous in B, if $IND(B - \{a\}) = IND(B)$, otherwise it is indispensable.

If all attributes $a \in B$ are indispensable in B. So the set of all indispensable attributes in B will be called the core of B, and will denote $CORE(B)$. The core is a set of attributes such that the removal of a single attribute impacts on the original ability to distinguish the objects. The core of attributes is the interaction of all reductions.

Example 5: To discern between the different equivalence classes in the example in Table 2.1, only the attributes Size, Type, and Color are necessary. So {Size, Type, Color} is an example of a reduct:

$$IND_A(\{Size, Type, Color\}) = IND_A(A) \quad (12)$$

Reducts can be computed on the basis of discernibility matrices and discernibility functions.

2.4.6 Discernibility matrices

A discernibility matrix of S is a symmetric $n \times n$ matrix with entries (Bjanger, 1999)

$$C_{ij} = a \in A|a(x_i) \neq a(x_j) \text{ for } i,j = 1,2, \ldots, n \qquad (13)$$

The entries for each object are thus the attributes that are needed in order to discern object i from object j.

Example 6: The discernibility matrix for our example in table 2.1 is shown in table 2.3. For readability, the three attributes are abbreviated s, t, c respectively.

Table 2.3 Discernibility matrix

	x_1	x_2	x_3	x_4	x_5
x_1	$\emptyset$	s, c	s, t, c	$\emptyset$	s, t
x_2		$\emptyset$	t, c	s, c	s, t, c
x_3			$\emptyset$	s, t, c	s, t, c
x_4				$\emptyset$	s, t
x_5					$\emptyset$

In a discernibility matrix the diagonal elements are naturally $\emptyset$ and $C_{ij} = C_{ji}$. Therefore, the lower triangular part in Table 2.3 is omitted. In the matrix, $C_{14} = \emptyset$ because the values of decision attributes for object s, t, and c are the same. $C_{12} = \{s, c\}$, because only the values of condition attributes s and c are different for the case of examples x_1 and x_4.

2.4.7 Discernibility functions

A discernibility function (Bjanger, 1999) is a Boolean product-of-sum function that expresses how an object or a set of objects can be discerned from a certain subset of the full universe of objects.

From the discernibility matrix, we can build a discernibility function. A discernibility function $\mathcal{F}_A$ for an information system S is a Boolean function of m Boolean variables $a_1^*, \dots, a_m^*$ (corresponding to the attributes $a_1, \dots, a_m$) defined as below.

$$\mathcal{F}_A(a_1^*, \dots, a_m^*) = \wedge \{\vee C_{ij}^* | 1 \le j \le i \le n, C_{ij} \ne \emptyset\} \quad (14)$$

where $C_{ij}^* = \{a^* | a \in C_{ij}\}$

The discernibility function is a conjunction of all the entries in the discernibility matrix that are not the empty set. The conjunction may, if possible, be simplified. The results of simplification are the possible reducts for the information system.

It is also possible to generate a discernibility function from the discernibility matrix for one of the objects in the information system. This is done by looking at only one row (or column) in the discernibility matrix, and forming a conjunction of all the entries in this row (or column). When we simplify this conjunction, we get possible reducts for the particular object in question.

2.4.8 Decision rules

For decision systems, $S = (U, A \cup \{d\})$, we would like to find an approximation of the decision, d. This can be done by constructing the decision-relative discernibility matrix of S. This matrix tell us how to discern an object from objects elonging to another decision class. The process of computing this matrix is called computing the discernibility matrix modulo the decision attribute.

We can obtain the decision rules from an information system or a reduction of condition attributes. Some condition values may be unnecessary in the decision rules, so we can remove them from the rules or change them into "don't care" values. The process of conditional value removed is called values reduction.

Let $S = (U, A, C, D)$ be an information system with n objects, where U is a finite set of objects. A is a set of attributes further classified into disjointed condition attributes C and decision attributes $D, A = C \cup D, V = \cup V_a, V_a$ is the domain of attribute a ∈ A. Hence, decision rules are often presented as implications and are often called "If … then …". If a, b, c, d are the condition attributes of set A, D is decision attribute. We can express the rules as

Rule 1

IF a = 1 THEN D = 1 (for the class 1)

Rule 2

IF b = S and c = M THEN D = 2 (for the class 2)

……………………………………………

Rule n

IF c = L THEN D = n (for the class n)

2.4.9 Rough membership function

In classical set theory, either an element belongs to a set or it does not. The corresponding membership function is the characteristic function for the set, i.e. the function takes values 0 and 1, respectively, In the case of rough sets, the notion of membership is different. The rough membership function quantifies. The degree of relation overlap between the set X and the equivalence $[X_B]$ class to which x belongs. It is defined as follows (Komorowski et.al.):

$$\mu_x^S : U \rightarrow [0,1] \text{and } \mu_x^S(x) = \frac{[x]_B \cap X}{|[x]_B|} \qquad (15)$$

where S = (U, B) is information system and $\emptyset \neq X \subseteq U$.

2.5 Related Research

2.5.1 Image retrieval

Xiang Sean Zhou, Ira Cohen, Qi Tian and Thomas S. Huang proposed a revised edge based structural feature extraction approach by following guidelines obtained from summarizing the existing feature extraction approaches. The experiments showed that the proposed algorithm yield comparable results to original set and better results than random sets.

Livari Kunttu, Leena Lepisto, Juhani Rauhamaa and Ari Visa presented a new approach to the histogram based retrieval of the images. Methods based on the binary histograms offer a simple and fast way to make image indexing for retrieval purposes without segmentation.

Qasim Iqbal and J.K. Aggarwal presented a study of the comparison of the performance of content based image retrieval systems based on structure with those based on histogram and texture analysis methods, where retrieval was concerned with locating images containing manmade objects. The experiments showed the percent of precision, recall and efficiency of 3 classes (building, non building and intermediate).

2.5.2 Fourier descriptor

Dengsheng Zhang and Guojun Lu build a java retrieval framework to compare shape retrieval using Fourier descriptors derived from different signatures. Common issues and techniques for shape representation and normalization are also analyzed in the paper. Data is given to show the retrieval result.

Andre Folkers and Hanan Samet implement the abstraction of the contour of the shape which is invariant against translation, scale, rotation, and starting point that is based on the use of Fourier descriptors in the retrieval process. These abstractions are used in a system to locate logos in an image database. The utility of this approach is illustrated using some sample queries.

2.5.3 Rough set theory

Pisit Phokharatkul presented a various techniques for recognizing invariant Thai printed characters to investigate ring projections and Fourier descriptors in an invariant Thai printed characters recognition system. His thesis development consists of two phases.

The first phase involves a recognizer using a ring projection and rough set method. The ring projection method is invariant to rotations, translations and scales, which are based on the total number of foreground pixels as distributed along circular rings. Then, the object's attributes are set up by these values. The rough set system is used to reduce the object's attribute data and generate the decision-making rules for coarse and fine classification.

The second phase involves a recognizer using Fourier descriptors and neural networks.

The test character set consists of 2,752 characters. The recognition rated 87% accuracy for ring projection and rough set method and 91% for Fourier descriptor and neural network method respectively. Some miss-classified to be groups are caused in the first phase, which causes the total rate of recognition to be lower than the method in the second phase.

Zia Akbar described a simple method of data classification using rough set theory as implemented by the tool provided in Alexander Ohrn's software application ROSETTA. Details are given of one such tool, Jonson's algorithm. The algorithm is applied to a typical set of marketing data from satisfaction surveys.

Shusaku Tsumoto overviews the following two important issues on the correspondence between Pawlak's rough set model and medical reasoning. The first main idea of rough sets is that a given concept can be approximated by partition-based knowledge as upper and lower approximation. Interestingly, these approximations correspond to the focusing mechanism of differential medical diagnosis; upper approximation as selection of candidates and lower approximation as concluding a final diagnosis. The second idea of rough sets is that a concept, observations can be represented as partitions in a given data set, where rough sets provides a rule induction method from a given data. Thus, this model can be used to extract rule-based knowledge from medical databases. Especially, rule induction based on the focusing mechanism is obtained in a natural way.

CHAPTER III
METHODOLOGY

3.1 Step of research methodology

3.1.1 To study the rough set theory for image retrieval.
3.1.2 To collect the images.
3.1.3 To analyze and design the prototype.
3.1.4 To create and develop the prototype.
3.1.5 To implement and test the prototype.
3.1.6 To evaluate the efficiency of prototype.
3.1.7 To conclude the result.

3.1.1 To study the rough set theory for image retrieval.

- Rough set theory for pattern classification and retrieval.
- Design and architecture of the rough rule-based system.

3.1.2 To collect the images.

We use the total number of images training data is 71928 binary images for all our experiments and use the total number of images testing data is 15984 binary images for testing the system. The images training data consist of 999 image classes and 72 rotations (initial angle is 1 degree and 1 rotation per 5 degree increasingly). The images testing data for testing process that is 999 image classes and 16 rotations (initial angle is 3 degrees and 1 rotation per 23 degrees).

3.1.3 To analyze and design the prototype.

- Define the input and output of system.
- Define the algorithm and operation of system.

3.1.4 To create and develop the prototype.

- Create and develop all algorithm and operation by MATLAB

3.1.5 To implement and test the prototype.

- Testing sub-program.
- Testing whole program after testing sub-program.

3.1.6 To evaluate the efficiency of prototype.

Efficiency of prototype is performed by percent of accurate retrieved images.

3.1.7 To conclude the result.

Conclude the researched result if correctly in line with the objectives, present the idea in order to better the system and issue the research document.

3.2 Research tools

3.2.1 Software

- Operating System: Window XP
- Application program development: MATLAB
- Graphics: Adobe Photoshop
- Structure Design: Microsoft Visio
- Document: Microsoft word

3.2.2 Hardware

- Personal Computer **:** CPU Pentium Core Duo 1.66 GHz up**,** Memory 1 GB up**,** Hard disk 80 GB

CHAPTER IV
EXPERIMENTAL AND RESULTS

4.1 Rough set method for image retrieval

4.1.1 System overview

We attempt to apply the rough rule based system to image retrieval. In our experiments, we use the Fourier descriptor and Quantization in the preprocessing step. The rough rule based system uses the Quantized Fourier descriptor to generate the rule based.

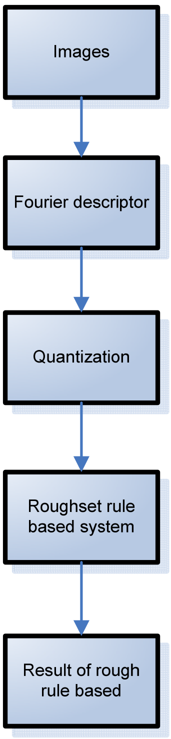

Figure 4.1 Process of rough set method for image retrieval.

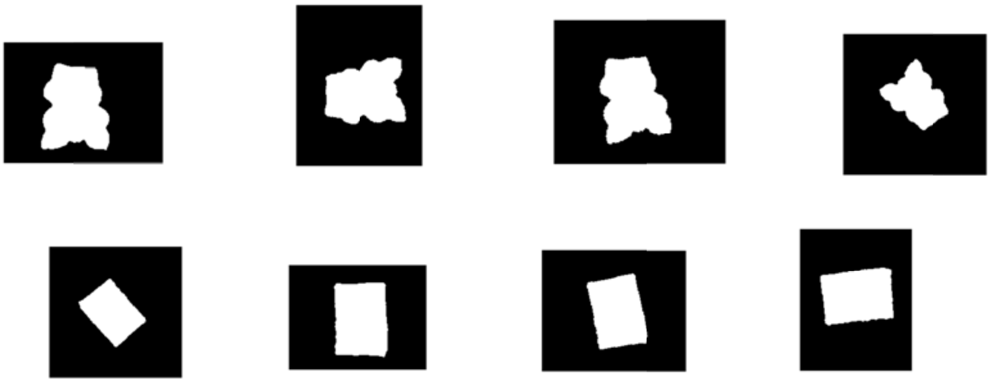

Figure 4.2 Example of training data.

4.1.2 Feature extraction and Quantization

The Fourier descriptors of training images are float value. They have to be changed for classification and rough rule based generation by Quantization. Quantization is the procedure of constraining something from a continuous set of values (such as the real numbers) to a discrete set (such as the integers).

Table 4.1 Example of training images Fourier descriptors.

Fourier coefficients attributes						Decision Attribute
a1	a2	a3	...	a14	a15	Image Class
0.10462	0.096699	0.055561	...	0.005693	0.005332	1
0.3701	0.20715	0.127	...	0.005442	0.002445	2
0.08215	0.080718	0.01913	...	0.001284	0.002858	3
0.05052	0.034928	0.024766	...	0.003413	0.002551	4
0.054908	0.047457	0.026874	...	0.001632	0.002636	5
0.10071	0.048087	0.052778	...	0.002494	0.003303	6
0.029128	0.10884	0.013497	...	0.004474	0.002265	7
0.035251	0.10177	0.019936	...	0.00258	0.003515	8
0.1274	0.050737	0.091551	...	0.000537	0.005549	9
0.2101	0.090454	0.079208	...	0.005774	0.003158	10
0.041131	0.10114	0.014615	...	0.003071	0.003723	11
0.010128	0.11767	0.011232	...	0.003745	0.003134	12
0.018454	0.11126	0.008772	...	0.004318	0.001964	13
0.07529	0.11206	0.030993	...	0.001762	0.004511	14
0.033152	0.024563	0.033566	...	0.005592	0.002585	15
0.00962	0.11024	0.007374	...	0.004132	0.001023	16

Table 4.1 Example of training images Fourier descriptors (continue).

Fourier coefficients attributes						Decision Attribute
a1	a2	a3	...	a14	a15	Image Class
0.37004	0.12003	0.12776	...	0.006674	0.006501	17
0.009883	0.10146	0.00413	...	0.004021	0.00192	18
0.007551	0.11077	0.004222	...	0.005099	0.002027	19
0.22829	0.075504	0.064461	...	0.003426	0.003199	20
0.009212	0.10755	0.003338	...	0.004867	0.001223	21
0.15389	0.044848	0.046393	...	0.004842	0.003655	22
0.28234	0.1056	0.085218	...	0.007637	0.002976	23
0.021768	0.095823	0.012351	...	0.003062	0.00239	24
0.060867	0.068595	0.017841	...	0.002665	0.000726	25
0.004061	0.10503	0.00247	...	0.004032	0.001807	26
0.068566	0.098481	0.033601	...	0.004628	0.003981	27
0.27086	0.19462	0.073008	...	0.006979	0.006221	28
...	...	...	...	...	...	...
0.011968	0.11702	0.011481	...	0.002076	0.002459	999
0.021108	0.10063	0.002689	...	0.002141	0.004016	1000

Table 4.2 Quantization of Fourier descriptors.

Quantized Fourier descriptors attributes						Decision Attribute
a1	a2	a3	...	a14	a15	Decision
5	6	10	...	15	15	1
3	4	5	...	15	15	2
7	7	14	...	15	15	3
10	12	13	...	15	15	4
10	11	13	...	15	15	5
5	11	10	...	15	15	6
13	5	14	...	15	15	7
12	5	14	...	15	15	8
5	10	6	...	15	15	9
4	6	8	...	15	15	10
11	5	14	...	15	15	11
14	5	14	...	15	15	12
14	5	15	...	15	15	13
8	5	12	...	15	15	14
12	13	12	...	15	15	15
15	5	15	...	15	15	16
3	5	5	...	15	15	17
15	5	15	...	15	15	18

Table 4.2 Quantization of Fourier descriptor (continue).

Quantized Fourier descriptors attributes						Decision Attribute
a1	a2	a3	...	a14	a15	Decision
15	5	15	...	15	15	19
4	8	9	...	15	15	20
15	5	15	...	15	15	21
5	11	11	...	15	15	22
4	5	7	...	15	15	23
13	6	14	...	15	15	24
9	9	14	...	15	15	25
15	5	15	...	15	15	26
9	6	12	...	15	15	27
4	5	8	...	15	15	28
...	...	...	...	...	...	...
14	5	14	...	15	15	999
13	5	15	...	15	15	1000

Our Quantization uses a simple discrete value as following pseudo code:

```
if((temp_image_fdes_sum(k) <= 1) & (temp_image_fdes_sum(k) > 0.5))
        attribute(index) = 1;
elseif((temp_image_fdes_sum(k) <= 0.5) & (temp_image_fdes_sum(k) > 0.4))
        attribute(index) = 2;
elseif((temp_image_fdes_sum(k) <= 0.4) & (temp_image_fdes_sum(k) > 0.3))
        attribute(index) = 3;
elseif((temp_image_fdes_sum(k) <= 0.3) & (temp_image_fdes_sum(k) > 0.2))
        attribute(index) = 4;
elseif((temp_image_fdes_sum(k) <= 0.2) & (temp_image_fdes_sum(k) > 0.1))
        attribute(index) = 5;
elseif((temp_image_fdes_sum(k) <= 0.1) & (temp_image_fdes_sum(k) > 0.09))
        attribute(index) = 6;
elseif((temp_image_fdes_sum(k) <= 0.09) & (temp_image_fdes_sum(k) > 0.08))
        attribute(index) = 7;
elseif((temp_image_fdes_sum(k) <= 0.08) & (temp_image_fdes_sum(k) > 0.07))
        attribute(index) = 8;
elseif((temp_image_fdes_sum(k) <= 0.07) & (temp_image_fdes_sum(k) > 0.06))
        attribute(index) = 9;
elseif((temp_image_fdes_sum(k) <= 0.06) & (temp_image_fdes_sum(k) > 0.05))
        attribute(index) = 10;
elseif((temp_image_fdes_sum(k) <= 0.05) & (temp_image_fdes_sum(k) > 0.04))
        attribute(index) = 11;
elseif((temp_image_fdes_sum(k) <= 0.04) & (temp_image_fdes_sum(k) > 0.03))
        attribute(index) = 12;
elseif((temp_image_fdes_sum(k) <= 0.03) & (temp_image_fdes_sum(k) > 0.02))
        attribute(index) = 13;
```

```
elseif((temp_image_fdes_sum(k) <= 0.02) & (temp_image_fdes_sum(k) > 0.01))
        attribute(index) = 14;
else
        attribute(index) = 15;
end;
```

Which “temp_image_fdes_sum” is the Fourier descriptor attribute and “attribute” is the Quantization attribute. Where k = 1, 2, …, 15 and index = 1, 2, …, 15.

4.1.3 Training and testing data

The images training data consist of 999 image classes and 72 rotations (initial angle is 1 degree and 1 rotation per 5 degree increasingly). Thus, the total number of images training data is 71928 images training data. We use the images testing data for testing process that is 999 image classes and 16 rotations (initial angle is 3 degrees and 1 rotation per 23 degrees increasingly). The total number of images testing data is 15984 images testing data.

From the preprocessing step, we use the Fourier descriptor and Quantization to create the attribute for generating rule based system to use in testing process. The training process will generate rule based from images training data.

Table 4.3 Image training data.

Quantized Fourier descriptors attributes						Decision Attribute
No.	a1	a2	...	a14	a15	Decision
1	5	6	...	15	15	1
2	5	6	...	15	15	1
3	5	6	...	15	15	1
...	...	...	...	...	...	...
70	5	6	...	15	15	1
...	...	...	...	...	...	...
35929	7	4	...	15	15	500
35930	7	4	...	15	15	500
35931	8	4	...	15	15	500
...	...	...	...	...	...	...
35997	7	4	...	15	15	500
35998	7	4	...	15	15	500
35999	7	4	...	15	15	500

Table 4.3 Image training data (continue).

Quantized Fourier descriptors attributes						Decision Attribute
No.	a1	a2	...	a14	a15	Decision
...	...	...	...	...	...	...
71857	13	5	...	15	15	1000
71858	13	6	...	15	15	1000
71859	13	6	...	15	15	1000
...	...	...	...	...	...	...
71926	13	6	...	15	15	1000
71927	13	5	...	15	15	1000
71928	14	6	...	15	15	1000

Table 4.4 Image testing data.

Quantized Fourier descriptors attributes						Decision Attribute
No.	a1	a2	...	a14	a15	Decision
1	5	6	...	15	15	1
2	6	6	...	15	15	1
3	5	6	...	15	15	1
...	...	...	...	...	...	...
7969	7	4	...	15	15	500
7970	9	4	...	14	15	500
7971	9	4	...	15	15	500
7972	8	4	...	15	15	500
...	...	...	...	...	...	...
15982	13	7	...	15	15	1000
15983	13	8	...	15	15	1000
15984	14	6	...	15	15	1000

4.1.4 Equivalence class

The equivalence class is generated in each training data (we assume the all images training data is the objects) that all rows has same value attribute. From the 71928 objects, we found 24962 equivalence classes.

Table 4.5 Example of objects.

Object attributes															Decisions
a1	a2	a3	a4	a5	a6	a7	a8	a9	a10	a11	a12	a13	a14	a15	D
5	6	10	11	14	14	14	14	13	14	15	15	15	15	15	1
5	6	10	11	14	14	14	14	13	14	15	15	15	15	15	1

Table 4.5 Example of objects (continue).

Object attributes															Decisions
a1	a2	a3	a4	a5	a6	a7	a8	a9	a10	a11	a12	a13	a14	a15	D
5	6	9	11	14	14	14	14	13	14	15	15	15	15	15	1
6	6	10	10	14	14	14	14	13	15	15	15	15	15	15	1
6	6	9	10	14	14	13	14	13	15	15	15	15	15	15	1
6	6	9	10	14	14	14	14	13	15	15	15	15	15	15	1
5	6	9	10	14	14	14	13	13	15	15	15	15	15	15	1
5	6	8	10	14	14	14	13	13	15	15	15	15	15	15	1
5	6	8	10	14	14	14	13	13	15	15	15	15	15	15	1
5	6	8	10	14	14	14	13	13	15	15	15	15	15	15	1
5	6	8	10	14	14	14	13	13	14	15	15	15	15	15	1
5	6	8	10	14	14	14	13	13	14	15	15	15	15	15	1
5	6	8	10	14	14	14	13	13	14	15	15	15	15	15	1
6	6	8	10	14	14	13	13	13	14	15	15	15	15	15	1
5	6	9	10	14	14	14	14	13	14	15	15	15	15	15	1
3	4	5	7	14	13	12	13	14	14	15	14	15	15	15	2
3	4	5	7	13	13	12	13	14	14	14	14	15	15	15	2
3	4	5	7	13	13	12	13	14	14	15	15	15	15	15	2
4	5	5	11	13	13	14	14	14	14	14	14	15	15	15	2
4	5	6	11	13	13	14	14	14	14	14	15	15	15	15	2
4	5	6	11	13	13	14	14	14	14	14	14	15	15	15	2
4	5	6	12	13	13	14	15	14	14	14	14	15	15	15	2
4	5	6	12	13	13	14	15	14	14	15	15	15	14	15	2

Table 4.6 Equivalence class from example table 4.5.

Object attributes															Decisions
a1	a2	a3	a4	a5	a6	a7	a8	a9	a10	a11	a12	a13	a14	a15	D
6	6	10	10	14	14	14	14	13	15	15	15	15	15	15	1
6	6	9	10	14	14	14	14	13	15	15	15	15	15	15	1
5	6	8	10	14	14	14	13	13	14	15	15	15	15	15	1
6	6	8	10	14	14	13	13	13	14	15	15	15	15	15	1
5	6	9	10	14	14	14	14	13	14	15	15	15	15	15	1
3	4	5	7	14	13	12	13	14	14	15	14	15	15	15	2
3	4	5	7	13	13	12	13	14	14	14	14	15	15	15	2
3	4	5	7	14	13	12	13	14	14	15	14	15	15	15	2
3	4	5	7	13	13	12	13	14	14	14	14	15	15	15	2
3	4	5	7	13	13	12	13	14	14	15	15	15	15	15	2
4	5	5	11	13	13	14	14	14	14	14	14	15	15	15	2
4	5	6	11	13	13	14	14	14	14	14	15	15	15	15	2
4	5	6	11	13	13	14	14	14	14	14	14	15	15	15	2

Table 4.6 Equivalence class from example table 4.5 (continue).

Object attributes															Decisions
a1	a2	a3	a4	a5	a6	a7	a8	a9	a10	a11	a12	a13	a14	a15	D
4	5	6	12	13	13	14	15	14	14	14	14	15	15	15	2
4	5	6	12	13	13	14	15	14	14	15	15	15	14	15	2

4.1.5 Rough rule based system

The rough rule based system is generated from the training data that from preprocessing step. In this stage, the rough set theory for classification is used to discover knowledge from data and generate rule based from equivalence class. The discernibility matrix is created for finding dispensable classes and reduction of training data. The core attribute is found from dispensable classes. We use Johnson reduction method [7] to reduce the rules that generated from equivalence class. The attribute which is reduced will be defined to zero.

Table 4.7 Example of rough rule based system.

Attributes															Decisions
a1	a2	a3	a4	a5	a6	a7	a8	a9	a10	a11	a12	a13	a14	a15	D
0	6	10	11	14	0	0	0	13	0	0	15	0	0	0	1
0	6	9	11	14	0	0	14	13	0	0	15	0	0	0	1
6	6	10	0	0	0	0	0	13	15	0	0	0	0	0	1
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
12	5	13	12	14	14	14	15	0	0	0	0	0	0	0	24
12	5	13	12	14	14	14	15	0	0	0	0	0	0	0	64
12	5	13	12	14	14	14	15	0	0	0	0	0	0	0	327
12	5	13	12	14	14	14	15	0	0	0	0	0	0	0	600
12	5	13	12	14	15	14	0	0	0	15	0	0	0	0	24
10	9	14	12	14	0	14	0	0	0	0	0	0	0	0	25
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
12	7	12	13	13	14	13	0	0	0	0	0	0	0	0	996
12	7	12	13	13	14	13	0	0	0	0	0	0	0	0	997
12	7	0	0	14	14	13	14	0	0	0	0	0	0	0	996
12	7	0	0	14	14	13	14	0	0	0	0	0	0	0	997
11	6	0	14	0	14	13	0	14	0	0	0	0	0	0	996
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
13	6	15	14	15	0	0	14	0	0	0	0	0	0	0	1000
13	6	14	13	15	14	14	14	0	0	0	0	0	0	0	1000
14	6	15	13	15	0	14	0	0	0	0	0	0	0	0	1000

The rough rules based are used in testing process to classify and retrieve the images that similar to the image testing data.

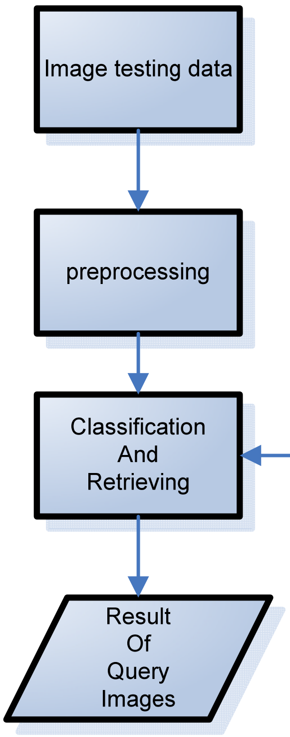

Figure 4.3 Using rough rule based for classification and retrieving.

4.2 Precision, recall and accuracy

The results of the rough data mining for image retrieval will be shown in three values of data retrieval evaluation. They are precision, recall and accuracy. We use the similarity measurement of image signature to compare to rough rule base system. For each retrieving, we only retrieved 20 first unique decisions.

In the context of classification tasks, the terms true positives, true negatives, false positives and false negatives are used to compare the given classification of an item (the class label assigned to the item by a classifier) with the desired correct classification (the class the item actually belongs to). These are illustrated by the table below:

Table 4.8 The terms of classification table tasks.

		correct result / classification	
		E1	E2
obtained result / classification	E1	tp (true positive)	fp (false positive)
	E2	fn (false negative)	tn (true negative)

$$\text{Precision} = \frac{\text{tp}}{\text{tp+fp}} \quad (16)$$

$$\text{Recall} = \frac{\text{tp}}{\text{tp+fn}} \quad (17)$$

$$\text{Accuracy} = \frac{\text{tp+tn}}{\text{tp+tn+fp+fn}} \quad (18)$$

4.3 Result of rough rule based image retrieval

Table 4.9 Example results of rough rule based.

Testing object No.	Precision	Recall	Accuracy	Relevant	Irrelevant	Relevant total	Irrelevant total	Retrieved total	Corrected class
1	0.15556	0.25	0.99725	7	38	28	21451	45	1
2	0.15686	0.28571	0.99707	8	43	28	21451	51	1
3	0.17391	0.28571	0.9973	8	38	28	21451	46	1
4	0.2	0.32143	0.99744	9	36	28	21451	45	1
5	0.19149	0.32143	0.99735	9	38	28	21451	47	1
6	0.19643	0.39286	0.99711	11	45	28	21451	56	1
7	0.17391	0.28571	0.9973	8	38	28	21451	46	1
8	0.16667	0.32143	0.99702	9	45	28	21451	54	1
9	0.15556	0.25	0.99725	7	38	28	21451	45	1
10	0.17391	0.28571	0.9973	8	38	28	21451	46	1
11	0.17391	0.28571	0.9973	8	38	28	21451	46	1
12	0.25	0.35714	0.99777	10	30	28	21451	40	1
13	0.19149	0.32143	0.99735	9	38	28	21451	47	1
14	0.23684	0.32143	0.99777	9	29	28	21451	38	1

Table 4.9 Example results of rough rule based (continue).

Testing object No.	Precision	Recall	Accuracy	Relevant	Irrelevant	Relevant total	Irrelevant total	Retrieved total	Corrected class
15	0.15556	0.25	0.99725	7	38	28	21451	45	1
16	0.25714	0.32143	0.9979	9	26	28	21451	35	1
17	0.09375	0.083333	0.99711	3	29	36	21443	32	2
18	0.20588	0.19444	0.99739	7	27	36	21443	34	2
19	0.15789	0.16667	0.99711	6	32	36	21443	38	2
20	0.038462	0.027778	0.99721	1	25	36	21443	26	2
21	0.09375	0.083333	0.99711	3	29	36	21443	32	2
22	0.19444	0.19444	0.9973	7	29	36	21443	36	2
23	0.2	0.25	0.99707	9	36	36	21443	45	2
24	0.14706	0.13889	0.99721	5	29	36	21443	34	2
25	0.16667	0.13889	0.99739	5	25	36	21443	30	2
26	0.2093	0.25	0.99716	9	34	36	21443	43	2
27	0.10638	0.13889	0.9966	5	42	36	21443	47	2
28	0.12821	0.13889	0.99697	5	34	36	21443	39	2
29	0.125	0.11111	0.99721	4	28	36	21443	32	2
30	0.10714	0.083333	0.9973	3	25	36	21443	28	2
31	0.076923	0.083333	0.99679	3	36	36	21443	39	2
32	0.16667	0.13889	0.99739	5	25	36	21443	30	2
33	0.14286	0.11628	0.99683	5	30	43	21436	35	3
34	0.18421	0.16279	0.99688	7	31	43	21436	38	3
35	0.052632	0.046512	0.99642	2	36	43	21436	38	3
36	0	0	0.99623	0	38	43	21436	38	3
37	0.14286	0.11628	0.99683	5	30	43	21436	35	3
38	0.15556	0.16279	0.99655	7	38	43	21436	45	3
39	0.15556	0.16279	0.99655	7	38	43	21436	45	3
40	0.11364	0.11628	0.99642	5	39	43	21436	44	3
41	0.076923	0.046512	0.99697	2	24	43	21436	26	3
42	0.17143	0.13953	0.99693	6	29	43	21436	35	3
43	0.11905	0.11628	0.99651	5	37	43	21436	42	3
44	0.10526	0.093023	0.9966	4	34	43	21436	38	3
45	0.14286	0.11628	0.99683	5	30	43	21436	35	3
46	0.12121	0.093023	0.99683	4	29	43	21436	33	3
47	0.051282	0.046512	0.99637	2	37	43	21436	39	3
48	0.097561	0.093023	0.99646	4	37	43	21436	41	3
49	0.27778	0.22222	0.99716	10	26	45	21434	36	4
50	0.078947	0.066667	0.99642	3	35	45	21434	38	4
51	0.175	0.15556	0.99669	7	33	45	21434	40	4
52	0.0625	0.066667	0.99595	3	45	45	21434	48	4
53	0.1	0.088889	0.99642	4	36	45	21434	40	4
54	0.13636	0.13333	0.99642	6	38	45	21434	44	4

Table 4.9 Example results of rough rule based (continue).

Testing object No.	Precision	Recall	Accuracy	Relevant	Irrelevant	Relevant total	Irrelevant total	Retrieved total	Corrected class
55	0.19444	0.15556	0.99688	7	29	45	21434	36	4
56	0.21622	0.17778	0.99693	8	29	45	21434	37	4
57	0.29412	0.22222	0.99725	10	24	45	21434	34	4
58	0.1875	0.13333	0.99697	6	26	45	21434	32	4
59	0.24324	0.2	0.99702	9	28	45	21434	37	4
60	0.12195	0.11111	0.99646	5	36	45	21434	41	4
61	0.097561	0.088889	0.99637	4	37	45	21434	41	4
62	0.069767	0.066667	0.99618	3	40	45	21434	43	4
63	0.097561	0.088889	0.99637	4	37	45	21434	41	4
64	0.16667	0.15556	0.9966	7	35	45	21434	42	4
65	0.10638	0.15625	0.99679	5	42	32	21447	47	5
66	0.18182	0.25	0.99721	8	36	32	21447	44	5
67	0.17949	0.21875	0.99735	7	32	32	21447	39	5
68	0.18	0.28125	0.99702	9	41	32	21447	50	5
69	0.04918	0.09375	0.99595	3	58	32	21447	61	5
70	0.025641	0.03125	0.99679	1	38	32	21447	39	5
71	0.17949	0.21875	0.99735	7	32	32	21447	39	5
72	0.11905	0.15625	0.99702	5	37	32	21447	42	5
73	0.13333	0.1875	0.99697	6	39	32	21447	45	5
74	0.18182	0.25	0.99721	8	36	32	21447	44	5
75	0.19512	0.25	0.99735	8	33	32	21447	41	5
76	0.13514	0.15625	0.99725	5	32	32	21447	37	5
77	0.16667	0.25	0.99702	8	40	32	21447	48	5
78	0.15909	0.21875	0.99711	7	37	32	21447	44	5
79	0.14634	0.1875	0.99716	6	35	32	21447	41	5
80	0.13514	0.15625	0.99725	5	32	32	21447	37	5
81	0.085106	0.33333	0.99763	4	43	12	21467	47	6
82	0.086957	0.33333	0.99767	4	42	12	21467	46	6
83	0.086957	0.33333	0.99767	4	42	12	21467	46	6
84	0.11628	0.41667	0.9979	5	38	12	21467	43	6
85	0.085106	0.33333	0.99763	4	43	12	21467	47	6
86	0.11628	0.41667	0.9979	5	38	12	21467	43	6
87	0.13889	0.41667	0.99823	5	31	12	21467	36	6
88	0.085106	0.33333	0.99763	4	43	12	21467	47	6
89	0.085106	0.33333	0.99763	4	43	12	21467	47	6
90	0.11429	0.33333	0.99818	4	31	12	21467	35	6
91	0.10204	0.41667	0.99763	5	44	12	21467	49	6
92	0.085106	0.33333	0.99763	4	43	12	21467	47	6
93	0.085106	0.33333	0.99763	4	43	12	21467	47	6
94	0.13889	0.41667	0.99823	5	31	12	21467	36	6

Table 4.9 Example results of rough rule based (continue).

Testing object No.	Precision	Recall	Accuracy	Relevant	Irrelevant	Relevant total	Irrelevant total	Retrieved total	Corrected class
95	0.078947	0.25	0.99795	3	35	12	21467	38	6
96	0.085106	0.33333	0.99763	4	43	12	21467	47	6
97	0.10204	0.15152	0.99665	5	44	33	21446	49	7
98	0.15217	0.21212	0.99697	7	39	33	21446	46	7
99	0.13636	0.18182	0.99697	6	38	33	21446	44	7
100	0.046512	0.060606	0.99665	2	41	33	21446	43	7
…	…	…	…	…	…	…	…	…	…
2100	0.036145	1	0.99628	3	80	3	21476	83	133
2101	0.032787	0.66667	0.99721	2	59	3	21476	61	133
2102	0.036145	1	0.99628	3	80	3	21476	83	133
2103	0.033898	0.66667	0.9973	2	57	3	21476	59	133
2104	0.033898	0.66667	0.9973	2	57	3	21476	59	133
2105	0.036145	1	0.99628	3	80	3	21476	83	133
2106	0.036145	1	0.99628	3	80	3	21476	83	133
2107	0.036145	1	0.99628	3	80	3	21476	83	133
2108	0.036145	1	0.99628	3	80	3	21476	83	133
2109	0.036145	1	0.99628	3	80	3	21476	83	133
2110	0.036145	1	0.99628	3	80	3	21476	83	133
2111	0.036145	1	0.99628	3	80	3	21476	83	133
2112	0.036145	1	0.99628	3	80	3	21476	83	133
2113	0.12195	0.22727	0.99753	5	36	22	21457	41	134
2114	0.13043	0.27273	0.99739	6	40	22	21457	46	134
2115	0.090909	0.18182	0.9973	4	40	22	21457	44	134
2116	0.15789	0.27273	0.99777	6	32	22	21457	38	134
2117	0.13953	0.27273	0.99753	6	37	22	21457	43	134
2118	0.052632	0.090909	0.99739	2	36	22	21457	38	134
2119	0.10638	0.22727	0.99725	5	42	22	21457	47	134
2120	0.13889	0.22727	0.99777	5	31	22	21457	36	134
2121	0.15789	0.27273	0.99777	6	32	22	21457	38	134
2122	0.15909	0.31818	0.99758	7	37	22	21457	44	134
2123	0.13043	0.27273	0.99739	6	40	22	21457	46	134
2124	0.13158	0.22727	0.99767	5	33	22	21457	38	134
2125	0.13953	0.27273	0.99753	6	37	22	21457	43	134
2126	0.13043	0.27273	0.99739	6	40	22	21457	46	134
2127	0.10256	0.18182	0.99753	4	35	22	21457	39	134
2128	0.095238	0.18182	0.99739	4	38	22	21457	42	134
2129	0.12903	0.26667	0.99823	4	27	15	21464	31	135
2130	0.18182	0.4	0.99832	6	27	15	21464	33	135
2131	0.14634	0.4	0.99795	6	35	15	21464	41	135
2132	0.15152	0.33333	0.99823	5	28	15	21464	33	135

Table 4.9 Example results of rough rule based (continue).

Testing object No.	Precision	Recall	Accuracy	Relevant	Irrelevant	Relevant total	Irrelevant total	Retrieved total	Corrected class
2133	0.15152	0.33333	0.99823	5	28	15	21464	33	135
2134	0.14634	0.4	0.99795	6	35	15	21464	41	135
2135	0.125	0.33333	0.9979	5	35	15	21464	40	135
2136	0.14894	0.46667	0.99777	7	40	15	21464	47	135
2137	0.15152	0.33333	0.99823	5	28	15	21464	33	135
2138	0.078947	0.2	0.99781	3	35	15	21464	38	135
2139	0.10417	0.33333	0.99753	5	43	15	21464	48	135
2140	0.16667	0.46667	0.998	7	35	15	21464	42	135
2141	0.18182	0.4	0.99832	6	27	15	21464	33	135
2142	0.16667	0.46667	0.998	7	35	15	21464	42	135
2143	0.088889	0.26667	0.99758	4	41	15	21464	45	135
2144	0.13158	0.33333	0.998	5	33	15	21464	38	135
2145	0.19512	0.25	0.99735	8	33	32	21447	41	136
2146	0.18519	0.3125	0.99693	10	44	32	21447	54	136
2147	0.225	0.28125	0.99749	9	31	32	21447	40	136
2148	0.18182	0.1875	0.99753	6	27	32	21447	33	136
2149	0.29268	0.375	0.99772	12	29	32	21447	41	136
2150	0.2449	0.375	0.99735	12	37	32	21447	49	136
2151	0.18919	0.21875	0.99744	7	30	32	21447	37	136
2152	0.17391	0.25	0.99711	8	38	32	21447	46	136
2153	0.24444	0.34375	0.99744	11	34	32	21447	45	136
2154	0.2381	0.3125	0.99749	10	32	32	21447	42	136
2155	0.27907	0.375	0.99763	12	31	32	21447	43	136
2156	0.16279	0.21875	0.99716	7	36	32	21447	43	136
2157	0.22222	0.3125	0.99735	10	35	32	21447	45	136
2158	0.22222	0.3125	0.99735	10	35	32	21447	45	136
2159	0.2	0.21875	0.99753	7	28	32	21447	35	136
2160	0.24444	0.34375	0.99744	11	34	32	21447	45	136
2161	0.14286	0.27778	0.998	5	30	18	21461	35	137
2162	0.17949	0.38889	0.998	7	32	18	21461	39	137
2163	0.18919	0.38889	0.99809	7	30	18	21461	37	137
2164	0.24242	0.44444	0.99837	8	25	18	21461	33	137
2165	0.2	0.44444	0.99804	8	32	18	21461	40	137
2166	0.17949	0.38889	0.998	7	32	18	21461	39	137
2167	0.10811	0.22222	0.99781	4	33	18	21461	37	137
2168	0.18919	0.38889	0.99809	7	30	18	21461	37	137
2169	0.2	0.44444	0.99804	8	32	18	21461	40	137
2170	0.18919	0.38889	0.99809	7	30	18	21461	37	137
2171	0.10811	0.22222	0.99781	4	33	18	21461	37	137
2172	0.2	0.44444	0.99804	8	32	18	21461	40	137

Table 4.9 Example results of rough rule based (continue).

Testing object No.	Precision	Recall	Accuracy	Relevant	Irrelevant	Relevant total	Irrelevant total	Retrieved total	Corrected class
2173	0.24242	0.44444	0.99837	8	25	18	21461	33	137
2174	0.17949	0.38889	0.998	7	32	18	21461	39	137
2175	0.10811	0.22222	0.99781	4	33	18	21461	37	137
2176	0.2	0.44444	0.99804	8	32	18	21461	40	137
2177	0.052632	0.11765	0.99763	2	36	17	21462	38	138
2178	0.13889	0.29412	0.998	5	31	17	21462	36	138
2179	0.11765	0.23529	0.998	4	30	17	21462	34	138
2180	0.11429	0.23529	0.99795	4	31	17	21462	35	138
2181	0.058824	0.11765	0.99781	2	32	17	21462	34	138
2182	0.13889	0.29412	0.998	5	31	17	21462	36	138
2183	0.11765	0.23529	0.998	4	30	17	21462	34	138
2184	0.11429	0.23529	0.99795	4	31	17	21462	35	138
2185	0.054054	0.11765	0.99767	2	35	17	21462	37	138
2186	0.18421	0.41176	0.99809	7	31	17	21462	38	138
2187	0.1	0.23529	0.99772	4	36	17	21462	40	138
2188	0.11429	0.23529	0.99795	4	31	17	21462	35	138
2189	0.026316	0.058824	0.99753	1	37	17	21462	38	138
2190	0.090909	0.17647	0.99795	3	30	17	21462	33	138
2191	0.10811	0.23529	0.99786	4	33	17	21462	37	138
2192	0.11111	0.23529	0.9979	4	32	17	21462	36	138
2193	0.16327	0.21053	0.99669	8	41	38	21441	49	139
2194	0.18	0.23684	0.99674	9	41	38	21441	50	139
2195	0.1	0.10526	0.99674	4	36	38	21441	40	139
2196	0.1	0.13158	0.99637	5	45	38	21441	50	139
2197	0.2	0.23684	0.99697	9	36	38	21441	45	139
2198	0.1875	0.23684	0.99683	9	39	38	21441	48	139
2199	0.1	0.10526	0.99674	4	36	38	21441	40	139
…	…	…	…	…	…	…	…	…	…
7854	0.063492	0.66667	0.99716	4	59	6	21473	63	492
7855	0.057692	0.5	0.99758	3	49	6	21473	52	492
7856	0.056604	0.5	0.99753	3	50	6	21473	53	492
7857	0	0	0.99465	0	100	15	21464	100	493
7858	0	0	0.99544	0	83	15	21464	83	493
7859	0.071429	0.26667	0.99707	4	52	15	21464	56	493
7860	0	0	0.99544	0	83	15	21464	83	493
7861	0	0	0.99465	0	100	15	21464	100	493
7862	0	0	0.99581	0	75	15	21464	75	493
7863	0.071429	0.26667	0.99707	4	52	15	21464	56	493
7864	0	0	0.99544	0	83	15	21464	83	493
7865	0	0	0.99465	0	100	15	21464	100	493

Table 4.9 Example results of rough rule based (continue).

Testing object No.	Precision	Recall	Accuracy	Relevant	Irrelevant	Relevant total	Irrelevant total	Retrieved total	Corrected class
7866	0.047619	0.26667	0.99576	4	80	15	21464	84	493
7867	0.046875	0.2	0.9966	3	61	15	21464	64	493
7868	0.038462	0.2	0.99595	3	75	15	21464	78	493
7869	0	0	0.99465	0	100	15	21464	100	493
7870	0	0	0.99544	0	83	15	21464	83	493
7871	0.046875	0.2	0.9966	3	61	15	21464	64	493
7872	0	0	0.99483	0	96	15	21464	96	493
7873	0.041096	0.33333	0.99646	3	70	9	21470	73	494
7874	0.058824	0.44444	0.99679	4	64	9	21470	68	494
7875	0.071429	0.44444	0.99735	4	52	9	21470	56	494
7876	0.072727	0.44444	0.99739	4	51	9	21470	55	494
7877	0.041096	0.33333	0.99646	3	70	9	21470	73	494
7878	0.058824	0.44444	0.99679	4	64	9	21470	68	494
7879	0.054054	0.22222	0.99804	2	35	9	21470	37	494
7880	0.056338	0.44444	0.99665	4	67	9	21470	71	494
7881	0.041096	0.33333	0.99646	3	70	9	21470	73	494
7882	0.058824	0.44444	0.99679	4	64	9	21470	68	494
7883	0.071429	0.44444	0.99735	4	52	9	21470	56	494
7884	0.068966	0.44444	0.99725	4	54	9	21470	58	494
7885	0.056338	0.44444	0.99665	4	67	9	21470	71	494
7886	0.071429	0.44444	0.99735	4	52	9	21470	56	494
7887	0.054054	0.22222	0.99804	2	35	9	21470	37	494
7888	0.056338	0.44444	0.99665	4	67	9	21470	71	494
7889	0.10256	0.14815	0.9973	4	35	27	21452	39	495
7890	0.085714	0.11111	0.99739	3	32	27	21452	35	495
7891	0.11429	0.14815	0.99749	4	31	27	21452	35	495
7892	0.18182	0.22222	0.99777	6	27	27	21452	33	495
7893	0.093023	0.14815	0.99711	4	39	27	21452	43	495
7894	0.076923	0.11111	0.99721	3	36	27	21452	39	495
7895	0.11429	0.14815	0.99749	4	31	27	21452	35	495
7896	0	0	0.99702	0	37	27	21452	37	495
7897	0.12903	0.14815	0.99767	4	27	27	21452	31	495
7898	0.15789	0.22222	0.99753	6	32	27	21452	38	495
7899	0.17073	0.25926	0.99749	7	34	27	21452	41	495
7900	0.18182	0.22222	0.99777	6	27	27	21452	33	495
7901	0.093023	0.14815	0.99711	4	39	27	21452	43	495
7902	0.17073	0.25926	0.99749	7	34	27	21452	41	495
7903	0.17073	0.25926	0.99749	7	34	27	21452	41	495
7904	0.097561	0.14815	0.99721	4	37	27	21452	41	495
7905	0.22642	0.18182	0.99558	12	41	66	21413	53	496

Table 4.9 Example results of rough rule based (continue).

Testing object No.	Precision	Recall	Accuracy	Relevant	Irrelevant	Relevant total	Irrelevant total	Retrieved total	Corrected class
7906	0.11628	0.075758	0.99539	5	38	66	21413	43	496
7907	0	0	0.99483	0	45	66	21413	45	496
7908	0	0	0.99432	0	56	66	21413	56	496
7909	0.12727	0.10606	0.99502	7	48	66	21413	55	496
7910	0.2	0.10606	0.99595	7	28	66	21413	35	496
7911	0.15909	0.10606	0.99553	7	37	66	21413	44	496
7912	0.037736	0.030303	0.99465	2	51	66	21413	53	496
7913	0.051724	0.045455	0.99451	3	55	66	21413	58	496
7914	0.098039	0.075758	0.99502	5	46	66	21413	51	496
7915	0.12245	0.090909	0.9952	6	43	66	21413	49	496
7916	0.073529	0.075758	0.99423	5	63	66	21413	68	496
7917	0.17647	0.13636	0.99539	9	42	66	21413	51	496
7918	0	0	0.99469	0	48	66	21413	48	496
7919	0	0	0.99474	0	47	66	21413	47	496
7920	0	0	0.99451	0	52	66	21413	52	496
7921	0.097561	0.33333	0.9979	4	37	12	21467	41	497
7922	0.081081	0.25	0.998	3	34	12	21467	37	497
7923	0.068182	0.25	0.99767	3	41	12	21467	44	497
7924	0.10638	0.41667	0.99772	5	42	12	21467	47	497
7925	0.043478	0.16667	0.99749	2	44	12	21467	46	497
7926	0.081081	0.25	0.998	3	34	12	21467	37	497
7927	0.12821	0.41667	0.99809	5	34	12	21467	39	497
7928	0.10638	0.41667	0.99772	5	42	12	21467	47	497
7929	0.097561	0.33333	0.9979	4	37	12	21467	41	497
…	…	…	…	…	…	…	…	…	…
11074	0.20513	0.25	0.99744	8	31	32	21447	39	694
11075	0.15152	0.15625	0.99744	5	28	32	21447	33	694
11076	0.10638	0.15625	0.99679	5	42	32	21447	47	694
11077	0.13462	0.21875	0.99674	7	45	32	21447	52	694
11078	0.12245	0.1875	0.99679	6	43	32	21447	49	694
11079	0.18182	0.25	0.99721	8	36	32	21447	44	694
11080	0.12	0.1875	0.99674	6	44	32	21447	50	694
11081	0.13462	0.21875	0.99674	7	45	32	21447	52	694
11082	0.10345	0.09375	0.99744	3	26	32	21447	29	694
11083	0.15789	0.1875	0.9973	6	32	32	21447	38	694
11084	0.13043	0.1875	0.99693	6	40	32	21447	46	694
11085	0.13462	0.21875	0.99674	7	45	32	21447	52	694
11086	0.13333	0.1875	0.99697	6	39	32	21447	45	694
11087	0.20513	0.25	0.99744	8	31	32	21447	39	694
11088	0.20513	0.25	0.99744	8	31	32	21447	39	694

Table 4.9 Example results of rough rule based (continue).

Testing object No.	Precision	Recall	Accuracy	Relevant	Irrelevant	Relevant total	Irrelevant total	Retrieved total	Corrected class
11089	0.21622	0.17778	0.99693	8	29	45	21434	37	695
11090	0.16667	0.15556	0.9966	7	35	45	21434	42	695
11091	0.13514	0.11111	0.99665	5	32	45	21434	37	695
11092	0.22222	0.22222	0.99674	10	35	45	21434	45	695
11093	0.18182	0.22222	0.99628	10	45	45	21434	55	695
11094	0.15217	0.15556	0.99642	7	39	45	21434	46	695
11095	0.13158	0.11111	0.9966	5	33	45	21434	38	695
11096	0.18605	0.17778	0.99665	8	35	45	21434	43	695
11097	0.17143	0.13333	0.99683	6	29	45	21434	35	695
11098	0.14634	0.13333	0.99655	6	35	45	21434	41	695
11099	0.17308	0.2	0.99632	9	43	45	21434	52	695
11100	0.2	0.17778	0.99679	8	32	45	21434	40	695
11101	0.13462	0.15556	0.99614	7	45	45	21434	52	695
11102	0.098039	0.11111	0.996	5	46	45	21434	51	695
11103	0.25641	0.22222	0.99702	10	29	45	21434	39	695
11104	0.18	0.2	0.99642	9	41	45	21434	50	695
11105	0.2973	0.32353	0.99772	11	26	34	21445	37	696
11106	0.26471	0.26471	0.99767	9	25	34	21445	34	696
11107	0.17647	0.17647	0.99739	6	28	34	21445	34	696
11108	0.24242	0.23529	0.99763	8	25	34	21445	33	696
11109	0.1875	0.17647	0.99749	6	26	34	21445	32	696
11110	0.23077	0.17647	0.99777	6	20	34	21445	26	696
11111	0.28125	0.26471	0.99777	9	23	34	21445	32	696
11112	0.11765	0.11765	0.99721	4	30	34	21445	34	696
11113	0.17241	0.14706	0.99753	5	24	34	21445	29	696
11114	0.20588	0.20588	0.99749	7	27	34	21445	34	696
11115	0.24324	0.26471	0.99753	9	28	34	21445	37	696
11116	0.3	0.26471	0.99786	9	21	34	21445	30	696
11117	0.30435	0.20588	0.998	7	16	34	21445	23	696
11118	0.21212	0.20588	0.99753	7	26	34	21445	33	696
11119	0.28125	0.26471	0.99777	9	23	34	21445	32	696
11120	0.2973	0.32353	0.99772	11	26	34	21445	37	696
11121	0.21951	0.15	0.99614	9	32	60	21419	41	697
11122	0.22222	0.16667	0.99604	10	35	60	21419	45	697
11123	0.16667	0.11667	0.9959	7	35	60	21419	42	697
11124	0.21622	0.13333	0.99623	8	29	60	21419	37	697
11125	0.16667	0.083333	0.99628	5	25	60	21419	30	697
11126	0.20588	0.11667	0.99628	7	27	60	21419	34	697
11127	0.20588	0.11667	0.99628	7	27	60	21419	34	697
11128	0.27027	0.16667	0.99642	10	27	60	21419	37	697

Table 4.9 Example results of rough rule based (continue).

Testing object No.	Precision	Recall	Accuracy	Relevant	Irrelevant	Relevant total	Irrelevant total	Retrieved total	Corrected class
11129	0.2	0.11667	0.99623	7	28	60	21419	35	697
11130	0.27778	0.16667	0.99646	10	26	60	21419	36	697
11131	0.18182	0.1	0.99623	6	27	60	21419	33	697
11132	0.375	0.25	0.99674	15	25	60	21419	40	697
11133	0.22581	0.11667	0.99642	7	24	60	21419	31	697
11134	0.12903	0.066667	0.99614	4	27	60	21419	31	697
11135	0.275	0.18333	0.99637	11	29	60	21419	40	697
11136	0.19444	0.11667	0.99618	7	29	60	21419	36	697
11137	0.22727	0.34483	0.99753	10	34	29	21450	44	698
11138	0.2381	0.34483	0.99763	10	32	29	21450	42	698
11139	0.13514	0.17241	0.99739	5	32	29	21450	37	698
11140	0.16279	0.24138	0.9973	7	36	29	21450	43	698
11141	0.19149	0.31034	0.9973	9	38	29	21450	47	698
11142	0.13514	0.17241	0.99739	5	32	29	21450	37	698
11143	0.20455	0.31034	0.99744	9	35	29	21450	44	698
11144	0.23684	0.31034	0.99772	9	29	29	21450	38	698
11145	0.19149	0.31034	0.9973	9	38	29	21450	47	698
11146	0.2381	0.34483	0.99763	10	32	29	21450	42	698
11147	0.23684	0.31034	0.99772	9	29	29	21450	38	698
11148	0.1	0.10345	0.99753	3	27	29	21450	30	698
11149	0.19149	0.31034	0.9973	9	38	29	21450	47	698
11150	0.27027	0.34483	0.99786	10	27	29	21450	37	698
…	…	…	…	…	…	…	…	…	…
15964	0.11475	0.30435	0.99674	7	54	23	21456	61	999
15965	0.075472	0.17391	0.99683	4	49	23	21456	53	999
15966	0.071429	0.26087	0.99558	6	78	23	21456	84	999
15967	0.057692	0.13043	0.99679	3	49	23	21456	52	999
15968	0.075472	0.17391	0.99683	4	49	23	21456	53	999
15969	0.21154	0.32353	0.99702	11	41	34	21445	52	1000
15970	0.057692	0.088235	0.99628	3	49	34	21445	52	1000
15971	0.078947	0.088235	0.99693	3	35	34	21445	38	1000
15972	0.098039	0.14706	0.99651	5	46	34	21445	51	1000
15973	0.13158	0.14706	0.99711	5	33	34	21445	38	1000
15974	0.13158	0.14706	0.99711	5	33	34	21445	38	1000
15975	0.10256	0.11765	0.99697	4	35	34	21445	39	1000
15976	0.067797	0.11765	0.99604	4	55	34	21445	59	1000
15977	0.13158	0.14706	0.99711	5	33	34	21445	38	1000
15978	0.039216	0.058824	0.99623	2	49	34	21445	51	1000
15979	0.078947	0.088235	0.99693	3	35	34	21445	38	1000

Table 4.9 Example results of rough rule based (continue).

Testing object No.	Precision	Recall	Accuracy	Relevant	Irrelevant	Relevant total	Irrelevant total	Retrieved total	Corrected class
15980	0.086207	0.14706	0.99618	5	53	34	21445	58	1000
15981	0.085714	0.088235	0.99707	3	32	34	21445	35	1000
15982	0.052632	0.088235	0.99604	3	54	34	21445	57	1000
15983	0	0	0.9966	0	39	34	21445	39	1000
15984	0.1	0.17647	0.99618	6	54	34	21445	60	1000

Table 4.10 Overall results of rough rules based image retrieval.

total testing data	15,984 objects
total rough rules based number	21,479 rules
total usage times	13,286 seconds
total usage memory for rules based	2,886,948 bytes
average precision	0.1297
average recall	0.261
average accuracy	0.9971

4.4 Result of similarity measurement

The similarity measurement is the one of the retrieved techniques for shape based image retrieval with Fourier descriptor. A model shape indexed by Fourier Descriptor feature $f_m = [f_m^1, f_m^2, \dots, f_m^N]$ and a testing data shape indexed by Fourier Descriptor feature $f_d = [f_d^1, f_d^2, \dots, f_d^N]$, since both features are normalized as to translation, rotation and scale, the Euclidean distance between the two feature vectors can be used as the similarity measurement.

$$d = \left(\sum_{i=0}^{N} |f_m^i - f_d^i|^2\right)^{\frac{1}{2}} \qquad (19)$$

Where N is the truncated number of harmonics needed to index the shape.

Table 4.11 Example results of similarity measurement.

Testing object No.	Precision	Recall	Accuracy	Relevant	Irrelevant	Relevant total	Irrelevant total	Retrieved total	Corrected class
1	0.05	1	0.98098	72	1368	72	71856	1440	1
2	0.050035	1	0.98099	72	1367	72	71856	1439	1

Table 4.11 Example results of similarity measurement (continue).

Testing object No.	Precision	Recall	Accuracy	Relevant	Irrelevant	Relevant total	Irrelevant total	Retrieved total	Corrected class
3	0.051502	1	0.98156	72	1326	72	71856	1398	1
4	0.050104	1	0.98102	72	1365	72	71856	1437	1
5	0.05	1	0.98098	72	1368	72	71856	1440	1
6	0.050491	1	0.98118	72	1354	72	71856	1426	1
7	0.052825	1	0.98205	72	1291	72	71856	1363	1
8	0.050139	1	0.98104	72	1364	72	71856	1436	1
9	0.05	1	0.98098	72	1368	72	71856	1440	1
10	0.050104	1	0.98102	72	1365	72	71856	1437	1
11	0.05225	1	0.98184	72	1306	72	71856	1378	1
12	0.050174	1	0.98105	72	1363	72	71856	1435	1
13	0.05	1	0.98098	72	1368	72	71856	1440	1
14	0.050633	1	0.98123	72	1350	72	71856	1422	1
15	0.052516	1	0.98194	72	1299	72	71856	1371	1
16	0.05	1	0.98098	72	1368	72	71856	1440	1
17	0.071642	0.66667	0.99102	48	622	72	71856	670	2
18	0.053156	0.66667	0.98778	48	855	72	71856	903	2
19	0.050209	0.66667	0.98704	48	908	72	71856	956	2
20	0.058111	0.66667	0.98885	48	778	72	71856	826	2
21	0.06867	0.66667	0.99062	48	651	72	71856	699	2
22	0.047198	0.66667	0.98619	48	969	72	71856	1017	2
23	0.042105	0.66667	0.98448	48	1092	72	71856	1140	2
24	0.055236	0.66667	0.98825	48	821	72	71856	869	2
25	0.61538	0.33333	0.99912	24	15	72	71856	39	2
26	0.053452	0.66667	0.98785	48	850	72	71856	898	2
27	0.045627	0.66667	0.98571	48	1004	72	71856	1052	2
28	0.07465	0.66667	0.99139	48	595	72	71856	643	2
29	0.51064	0.33333	0.99901	24	23	72	71856	47	2
30	0.35821	0.33333	0.99873	24	43	72	71856	67	2
31	0.04908	0.66667	0.98674	48	930	72	71856	978	2
32	0.63158	0.33333	0.99914	24	14	72	71856	38	2
33	0.05	1	0.98098	72	1368	72	71856	1440	3
34	0.05	1	0.98098	72	1368	72	71856	1440	3
35	0.05	1	0.98098	72	1368	72	71856	1440	3
36	0.05	1	0.98098	72	1368	72	71856	1440	3
37	0.05	1	0.98098	72	1368	72	71856	1440	3
38	0.05	1	0.98098	72	1368	72	71856	1440	3
39	0.049306	0.98611	0.98095	71	1369	72	71856	1440	3
40	0.05	1	0.98098	72	1368	72	71856	1440	3
41	0.05	1	0.98098	72	1368	72	71856	1440	3
42	0.05	1	0.98098	72	1368	72	71856	1440	3

Table 4.11 Example results of similarity measurement (continue).

Testing object No.	Precision	Recall	Accuracy	Relevant	Irrelevant	Relevant total	Irrelevant total	Retrieved total	Corrected class
43	0.05	1	0.98098	72	1368	72	71856	1440	3
44	0.05	1	0.98098	72	1368	72	71856	1440	3
45	0.049306	0.98611	0.98095	71	1369	72	71856	1440	3
46	0.05	1	0.98098	72	1368	72	71856	1440	3
47	0.049306	0.98611	0.98095	71	1369	72	71856	1440	3
48	0.05	1	0.98098	72	1368	72	71856	1440	3
49	0.05	1	0.98098	72	1368	72	71856	1440	4
50	0.05	1	0.98098	72	1368	72	71856	1440	4
51	0.05	1	0.98098	72	1368	72	71856	1440	4
52	0.05	1	0.98098	72	1368	72	71856	1440	4
53	0.05	1	0.98098	72	1368	72	71856	1440	4
54	0.05	1	0.98098	72	1368	72	71856	1440	4
55	0.05	1	0.98098	72	1368	72	71856	1440	4
56	0.05	1	0.98098	72	1368	72	71856	1440	4
57	0.05	1	0.98098	72	1368	72	71856	1440	4
58	0.05	1	0.98098	72	1368	72	71856	1440	4
59	0.05	1	0.98098	72	1368	72	71856	1440	4
60	0.05	1	0.98098	72	1368	72	71856	1440	4
61	0.05	1	0.98098	72	1368	72	71856	1440	4
62	0.05	1	0.98098	72	1368	72	71856	1440	4
63	0.05	1	0.98098	72	1368	72	71856	1440	4
64	0.05	1	0.98098	72	1368	72	71856	1440	4
65	0.05	1	0.98098	72	1368	72	71856	1440	5
66	0.05	1	0.98098	72	1368	72	71856	1440	5
67	0.05	1	0.98098	72	1368	72	71856	1440	5
68	0.05	1	0.98098	72	1368	72	71856	1440	5
69	0.05	1	0.98098	72	1368	72	71856	1440	5
70	0.05	1	0.98098	72	1368	72	71856	1440	5
71	0.05	1	0.98098	72	1368	72	71856	1440	5
72	0.05	1	0.98098	72	1368	72	71856	1440	5
73	0.05	1	0.98098	72	1368	72	71856	1440	5
74	0.05	1	0.98098	72	1368	72	71856	1440	5
75	0.05	1	0.98098	72	1368	72	71856	1440	5
76	0.05	1	0.98098	72	1368	72	71856	1440	5
77	0.05	1	0.98098	72	1368	72	71856	1440	5
78	0.05	1	0.98098	72	1368	72	71856	1440	5
79	0.05	1	0.98098	72	1368	72	71856	1440	5
80	0.05	1	0.98098	72	1368	72	71856	1440	5
81	0.05	1	0.98098	72	1368	72	71856	1440	6
82	0.05	1	0.98098	72	1368	72	71856	1440	6

Table 4.11 Example results of similarity measurement (continue).

Testing object No.	Precision	Recall	Accuracy	Relevant	Irrelevant	Relevant total	Irrelevant total	Retrieved total	Corrected class
83	0.05	1	0.98098	72	1368	72	71856	1440	6
84	0.05	1	0.98098	72	1368	72	71856	1440	6
85	0.05	1	0.98098	72	1368	72	71856	1440	6
86	0.05	1	0.98098	72	1368	72	71856	1440	6
87	0.05	1	0.98098	72	1368	72	71856	1440	6
88	0.05	1	0.98098	72	1368	72	71856	1440	6
89	0.05	1	0.98098	72	1368	72	71856	1440	6
90	0.05	1	0.98098	72	1368	72	71856	1440	6
91	0.05	1	0.98098	72	1368	72	71856	1440	6
92	0.05	1	0.98098	72	1368	72	71856	1440	6
93	0.05	1	0.98098	72	1368	72	71856	1440	6
94	0.05	1	0.98098	72	1368	72	71856	1440	6
95	0.05	1	0.98098	72	1368	72	71856	1440	6
96	0.05	1	0.98098	72	1368	72	71856	1440	6
97	0.05	1	0.98098	72	1368	72	71856	1440	7
98	0.05	1	0.98098	72	1368	72	71856	1440	7
99	0.05	1	0.98098	72	1368	72	71856	1440	7
…	…	…	…	…	…	…	…	…	…
2421	0.05	1	0.98098	72	1368	72	71856	1440	153
2422	0.05	1	0.98098	72	1368	72	71856	1440	153
2423	0.05	1	0.98098	72	1368	72	71856	1440	153
2424	0.05	1	0.98098	72	1368	72	71856	1440	153
2425	0.05	1	0.98098	72	1368	72	71856	1440	153
2426	0.05	1	0.98098	72	1368	72	71856	1440	153
2427	0.05	1	0.98098	72	1368	72	71856	1440	153
2428	0.05	1	0.98098	72	1368	72	71856	1440	153
2429	0.05	1	0.98098	72	1368	72	71856	1440	153
2430	0.05	1	0.98098	72	1368	72	71856	1440	153
2431	0.05	1	0.98098	72	1368	72	71856	1440	153
2432	0.05	1	0.98098	72	1368	72	71856	1440	153
2433	0.05	1	0.98098	72	1368	72	71856	1440	154
2434	0.05	1	0.98098	72	1368	72	71856	1440	154
2435	0.05	1	0.98098	72	1368	72	71856	1440	154
2436	0.05	1	0.98098	72	1368	72	71856	1440	154
2437	0.05	1	0.98098	72	1368	72	71856	1440	154
2438	0.05	1	0.98098	72	1368	72	71856	1440	154
2439	0.05	1	0.98098	72	1368	72	71856	1440	154
2440	0.05	1	0.98098	72	1368	72	71856	1440	154
2441	0.05	1	0.98098	72	1368	72	71856	1440	154
2442	0.05	1	0.98098	72	1368	72	71856	1440	154

Table 4.11 Example results of similarity measurement (continue).

Testing object No.	Precision	Recall	Accuracy	Relevant	Irrelevant	Relevant total	Irrelevant total	Retrieved total	Corrected class
2443	0.05	1	0.98098	72	1368	72	71856	1440	154
2444	0.05	1	0.98098	72	1368	72	71856	1440	154
2445	0.05	1	0.98098	72	1368	72	71856	1440	154
2446	0.05	1	0.98098	72	1368	72	71856	1440	154
2447	0.05	1	0.98098	72	1368	72	71856	1440	154
2448	0.05	1	0.98098	72	1368	72	71856	1440	154
2449	0.052709	1	0.98201	72	1294	72	71856	1366	155
2450	0.051687	1	0.98163	72	1321	72	71856	1393	155
2451	0.052863	1	0.98207	72	1290	72	71856	1362	155
2452	0.051028	1	0.98138	72	1339	72	71856	1411	155
2453	0.052902	1	0.98208	72	1289	72	71856	1361	155
2454	0.052632	1	0.98198	72	1296	72	71856	1368	155
2455	0.052174	1	0.98182	72	1308	72	71856	1380	155
2456	0.051986	1	0.98175	72	1313	72	71856	1385	155
2457	0.051502	1	0.98156	72	1326	72	71856	1398	155
2458	0.051687	1	0.98163	72	1321	72	71856	1393	155
2459	0.051502	1	0.98156	72	1326	72	71856	1398	155
2460	0.050919	1	0.98134	72	1342	72	71856	1414	155
2461	0.05267	1	0.982	72	1295	72	71856	1367	155
2462	0.051911	1	0.98172	72	1315	72	71856	1387	155
2463	0.051209	1	0.98145	72	1334	72	71856	1406	155
2464	0.050955	1	0.98136	72	1341	72	71856	1413	155
2465	0.05	1	0.98098	72	1368	72	71856	1440	156
2466	0.05	1	0.98098	72	1368	72	71856	1440	156
2467	0.05	1	0.98098	72	1368	72	71856	1440	156
2468	0.05	1	0.98098	72	1368	72	71856	1440	156
2469	0.05	1	0.98098	72	1368	72	71856	1440	156
2470	0.05	1	0.98098	72	1368	72	71856	1440	156
2471	0.05	1	0.98098	72	1368	72	71856	1440	156
2472	0.05	1	0.98098	72	1368	72	71856	1440	156
2473	0.05	1	0.98098	72	1368	72	71856	1440	156
2474	0.05	1	0.98098	72	1368	72	71856	1440	156
2475	0.05	1	0.98098	72	1368	72	71856	1440	156
2476	0.05	1	0.98098	72	1368	72	71856	1440	156
2477	0.05	1	0.98098	72	1368	72	71856	1440	156
2478	0.05	1	0.98098	72	1368	72	71856	1440	156
2479	0.05	1	0.98098	72	1368	72	71856	1440	156
2480	0.05	1	0.98098	72	1368	72	71856	1440	156
2481	0.05	1	0.98098	72	1368	72	71856	1440	157
2482	0.05	1	0.98098	72	1368	72	71856	1440	157

Table 4.11 Example results of similarity measurement (continue).

Testing object No.	Precision	Recall	Accuracy	Relevant	Irrelevant	Relevant total	Irrelevant total	Retrieved total	Corrected class
2483	0.05	1	0.98098	72	1368	72	71856	1440	157
2484	0.05	1	0.98098	72	1368	72	71856	1440	157
2485	0.05	1	0.98098	72	1368	72	71856	1440	157
2486	0.05	1	0.98098	72	1368	72	71856	1440	157
2487	0.05	1	0.98098	72	1368	72	71856	1440	157
2488	0.05	1	0.98098	72	1368	72	71856	1440	157
2489	0.05	1	0.98098	72	1368	72	71856	1440	157
2490	0.05	1	0.98098	72	1368	72	71856	1440	157
2491	0.05	1	0.98098	72	1368	72	71856	1440	157
2492	0.05	1	0.98098	72	1368	72	71856	1440	157
2493	0.05	1	0.98098	72	1368	72	71856	1440	157
2494	0.05	1	0.98098	72	1368	72	71856	1440	157
2495	0.05	1	0.98098	72	1368	72	71856	1440	157
2496	0.05	1	0.98098	72	1368	72	71856	1440	157
2497	0.12224	1	0.99281	72	517	72	71856	589	158
2498	0.1073	1	0.99167	72	599	72	71856	671	158
2499	0.10405	1	0.99138	72	620	72	71856	692	158
2500	0.1791	1	0.99541	72	330	72	71856	402	158
2501	0.14257	1	0.99398	72	433	72	71856	505	158
2502	0.15551	1	0.99456	72	391	72	71856	463	158
2503	0.11784	1	0.99251	72	539	72	71856	611	158
2504	0.13714	1	0.9937	72	453	72	71856	525	158
2505	0.13309	1	0.99348	72	469	72	71856	541	158
2506	0.12162	1	0.99277	72	520	72	71856	592	158
2507	0.13953	1	0.99383	72	444	72	71856	516	158
2508	0.16438	1	0.99491	72	366	72	71856	438	158
2509	0.13235	1	0.99344	72	472	72	71856	544	158
2510	0.10926	1	0.99184	72	587	72	71856	659	158
2511	0.12162	1	0.99277	72	520	72	71856	592	158
2512	0.12903	1	0.99324	72	486	72	71856	558	158
2513	1	1	1	72	0	72	71856	72	159
2514	0.85714	1	0.99983	72	12	72	71856	84	159
2515	0.9	1	0.99989	72	8	72	71856	80	159
2516	1	1	1	72	0	72	71856	72	159
2517	1	1	1	72	0	72	71856	72	159
2518	0.91139	1	0.9999	72	7	72	71856	79	159
2519	0.86747	1	0.99985	72	11	72	71856	83	159
2520	0.9863	1	0.99999	72	1	72	71856	73	159
2521	0.97297	1	0.99997	72	2	72	71856	74	159
2522	0.84706	1	0.99982	72	13	72	71856	85	159

Table 4.11 Example results of similarity measurement (continue).

Testing object No.	Precision	Recall	Accuracy	Relevant	Irrelevant	Relevant total	Irrelevant total	Retrieved total	Corrected class
2523	0.88889	1	0.99987	72	9	72	71856	81	159
2524	1	1	1	72	0	72	71856	72	159
2525	0.85714	1	0.99983	72	12	72	71856	84	159
2526	0.86747	1	0.99985	72	11	72	71856	83	159
2527	0.9	1	0.99989	72	8	72	71856	80	159
2528	1	1	1	72	0	72	71856	72	159
2529	0.05	1	0.98098	72	1368	72	71856	1440	160
2530	0.05	1	0.98098	72	1368	72	71856	1440	160
2531	0.050035	1	0.98099	72	1367	72	71856	1439	160
2532	0.050035	1	0.98099	72	1367	72	71856	1439	160
2533	0.05	1	0.98098	72	1368	72	71856	1440	160
2534	0.05	1	0.98098	72	1368	72	71856	1440	160
…	…	…	…	…	…	…	…	…	…
7959	1	1	1	72	0	72	71856	72	499
7960	1	1	1	72	0	72	71856	72	499
7961	1	1	1	72	0	72	71856	72	499
7962	1	1	1	72	0	72	71856	72	499
7963	1	1	1	72	0	72	71856	72	499
7964	1	1	1	72	0	72	71856	72	499
7965	1	0.94444	0.99994	68	0	72	71856	68	499
7966	1	1	1	72	0	72	71856	72	499
7967	1	1	1	72	0	72	71856	72	499
7968	1	1	1	72	0	72	71856	72	499
7969	1	1	1	72	0	72	71856	72	500
7970	0.97297	1	0.99997	72	2	72	71856	74	500
7971	0.97297	1	0.99997	72	2	72	71856	74	500
7972	1	1	1	72	0	72	71856	72	500
7973	1	1	1	72	0	72	71856	72	500
7974	0.9863	1	0.99999	72	1	72	71856	73	500
7975	0.9863	1	0.99999	72	1	72	71856	73	500
7976	1	1	1	72	0	72	71856	72	500
7977	1	1	1	72	0	72	71856	72	500
7978	1	1	1	72	0	72	71856	72	500
7979	1	1	1	72	0	72	71856	72	500
7980	1	1	1	72	0	72	71856	72	500
7981	1	1	1	72	0	72	71856	72	500
7982	1	1	1	72	0	72	71856	72	500
7983	1	1	1	72	0	72	71856	72	500
7984	1	1	1	72	0	72	71856	72	500
7985	0.05	1	0.98098	72	1368	72	71856	1440	501

Table 4.11 Example results of similarity measurement (continue).

Testing object No.	Precision	Recall	Accuracy	Relevant	Irrelevant	Relevant total	Irrelevant total	Retrieved total	Corrected class
7986	0.05	1	0.98098	72	1368	72	71856	1440	501
7987	0.05	1	0.98098	72	1368	72	71856	1440	501
7988	0.05	1	0.98098	72	1368	72	71856	1440	501
7989	0.05	1	0.98098	72	1368	72	71856	1440	501
7990	0.05	1	0.98098	72	1368	72	71856	1440	501
7991	0.05	1	0.98098	72	1368	72	71856	1440	501
7992	0.05	1	0.98098	72	1368	72	71856	1440	501
7993	0.05	1	0.98098	72	1368	72	71856	1440	501
7994	0.05	1	0.98098	72	1368	72	71856	1440	501
7995	0.05	1	0.98098	72	1368	72	71856	1440	501
7996	0.05	1	0.98098	72	1368	72	71856	1440	501
7997	0.05	1	0.98098	72	1368	72	71856	1440	501
7998	0.05	1	0.98098	72	1368	72	71856	1440	501
7999	0.05	1	0.98098	72	1368	72	71856	1440	501
8000	0.05	1	0.98098	72	1368	72	71856	1440	501
8001	0.1141	1	0.99223	72	559	72	71856	631	502
8002	0.050104	1	0.98102	72	1365	72	71856	1437	502
8003	0.05	1	0.98098	72	1368	72	71856	1440	502
8004	0.052023	1	0.98176	72	1312	72	71856	1384	502
8005	0.089776	1	0.98985	72	730	72	71856	802	502
8006	0.057007	1	0.98344	72	1191	72	71856	1263	502
8007	0.052288	1	0.98186	72	1305	72	71856	1377	502
8008	0.11765	1	0.99249	72	540	72	71856	612	502
8009	0.12654	1	0.99309	72	497	72	71856	569	502
8010	0.05868	1	0.98394	72	1155	72	71856	1227	502
8011	0.064458	1	0.98547	72	1045	72	71856	1117	502
8012	0.073171	1	0.98732	72	912	72	71856	984	502
8013	0.083045	1	0.98895	72	795	72	71856	867	502
8014	0.050279	1	0.98109	72	1360	72	71856	1432	502
8015	0.06175	1	0.98479	72	1094	72	71856	1166	502
8016	0.056827	1	0.98339	72	1195	72	71856	1267	502
8017	0.2915	1	0.99757	72	175	72	71856	247	503
8018	0.21302	1	0.9963	72	266	72	71856	338	503
8019	0.11726	1	0.99246	72	542	72	71856	614	503
8020	0.15789	1	0.99466	72	384	72	71856	456	503
8021	0.21752	1	0.9964	72	259	72	71856	331	503
8022	0.1618	1	0.99481	72	373	72	71856	445	503
8023	0.11632	1	0.9924	72	547	72	71856	619	503
8024	0.17476	1	0.99527	72	340	72	71856	412	503
8025	0.23762	1	0.99679	72	231	72	71856	303	503

Table 4.11 Example results of similarity measurement (continue).

Testing object No.	Precision	Recall	Accuracy	Relevant	Irrelevant	Relevant total	Irrelevant total	Retrieved total	Corrected class
8026	0.1295	1	0.99327	72	484	72	71856	556	503
8027	0.11356	1	0.99219	72	562	72	71856	634	503
8028	0.14118	1	0.99391	72	438	72	71856	510	503
8029	0.27481	1	0.99736	72	190	72	71856	262	503
8030	0.11502	1	0.9923	72	554	72	71856	626	503
8031	0.13187	1	0.99341	72	474	72	71856	546	503
8032	0.13115	1	0.99337	72	477	72	71856	549	503
8033	0.1033	1	0.99131	72	625	72	71856	697	504
8034	0.10573	1	0.99153	72	609	72	71856	681	504
8035	0.15352	1	0.99448	72	397	72	71856	469	504
8036	0.11374	1	0.9922	72	561	72	71856	633	504
8037	0.15418	1	0.99451	72	395	72	71856	467	504
8038	0.1194	1	0.99262	72	531	72	71856	603	504
8039	0.18321	1	0.99554	72	321	72	71856	393	504
8040	0.14845	1	0.99426	72	413	72	71856	485	504
8041	0.11707	1	0.99245	72	543	72	71856	615	504
8042	0.13067	1	0.99334	72	479	72	71856	551	504
8043	0.15031	1	0.99434	72	407	72	71856	479	504
8044	0.13163	1	0.9934	72	475	72	71856	547	504
8045	0.1152	1	0.99231	72	553	72	71856	625	504
8046	0.14118	1	0.99391	72	438	72	71856	510	504
8047	0.14008	1	0.99385	72	442	72	71856	514	504
8048	0.12435	1	0.99295	72	507	72	71856	579	504
8049	0.050209	1	0.98106	72	1362	72	71856	1434	505
8050	0.050104	1	0.98102	72	1365	72	71856	1437	505
8051	0.051173	1	0.98144	72	1335	72	71856	1407	505
8052	0.051576	1	0.98159	72	1324	72	71856	1396	505
…	…	…	…	…	…	…	…	…	…
12046	0.059406	1	0.98415	72	1140	72	71856	1212	754
12047	0.057371	1	0.98355	72	1183	72	71856	1255	754
12048	0.07611	1	0.98785	72	874	72	71856	946	754
12049	0.05	1	0.98098	72	1368	72	71856	1440	755
12050	0.05	1	0.98098	72	1368	72	71856	1440	755
12051	0.05	1	0.98098	72	1368	72	71856	1440	755
12052	0.05	1	0.98098	72	1368	72	71856	1440	755
12053	0.05	1	0.98098	72	1368	72	71856	1440	755
12054	0.05	1	0.98098	72	1368	72	71856	1440	755
12055	0.05	1	0.98098	72	1368	72	71856	1440	755
12056	0.05	1	0.98098	72	1368	72	71856	1440	755
12057	0.05	1	0.98098	72	1368	72	71856	1440	755

Table 4.11 Example results of similarity measurement (continue).

Testing object No.	Precision	Recall	Accuracy	Relevant	Irrelevant	Relevant total	Irrelevant total	Retrieved total	Corrected class
12058	0.05	1	0.98098	72	1368	72	71856	1440	755
12059	0.05	1	0.98098	72	1368	72	71856	1440	755
12060	0.05	1	0.98098	72	1368	72	71856	1440	755
12061	0.05	1	0.98098	72	1368	72	71856	1440	755
12062	0.05	1	0.98098	72	1368	72	71856	1440	755
12063	0.05	1	0.98098	72	1368	72	71856	1440	755
12064	0.05	1	0.98098	72	1368	72	71856	1440	755
12065	0.05	1	0.98098	72	1368	72	71856	1440	756
12066	0.05	1	0.98098	72	1368	72	71856	1440	756
12067	0.05	1	0.98098	72	1368	72	71856	1440	756
12068	0.05	1	0.98098	72	1368	72	71856	1440	756
12069	0.05	1	0.98098	72	1368	72	71856	1440	756
12070	0.05	1	0.98098	72	1368	72	71856	1440	756
12071	0.05	1	0.98098	72	1368	72	71856	1440	756
12072	0.05	1	0.98098	72	1368	72	71856	1440	756
12073	0.05	1	0.98098	72	1368	72	71856	1440	756
12074	0.05	1	0.98098	72	1368	72	71856	1440	756
12075	0.05	1	0.98098	72	1368	72	71856	1440	756
12076	0.05	1	0.98098	72	1368	72	71856	1440	756
12077	0.05	1	0.98098	72	1368	72	71856	1440	756
12078	0.05	1	0.98098	72	1368	72	71856	1440	756
12079	0.05	1	0.98098	72	1368	72	71856	1440	756
12080	0.05	1	0.98098	72	1368	72	71856	1440	756
12081	0.05	1	0.98098	72	1368	72	71856	1440	757
12082	0.05	1	0.98098	72	1368	72	71856	1440	757
12083	0.05	1	0.98098	72	1368	72	71856	1440	757
12084	0.05	1	0.98098	72	1368	72	71856	1440	757
12085	0.05	1	0.98098	72	1368	72	71856	1440	757
12086	0.05	1	0.98098	72	1368	72	71856	1440	757
12087	0.05	1	0.98098	72	1368	72	71856	1440	757
12088	0.05	1	0.98098	72	1368	72	71856	1440	757
12089	0.05	1	0.98098	72	1368	72	71856	1440	757
12090	0.05	1	0.98098	72	1368	72	71856	1440	757
12091	0.05	1	0.98098	72	1368	72	71856	1440	757
12092	0.05	1	0.98098	72	1368	72	71856	1440	757
12093	0.05	1	0.98098	72	1368	72	71856	1440	757
12094	0.05	1	0.98098	72	1368	72	71856	1440	757
12095	0.05	1	0.98098	72	1368	72	71856	1440	757
12096	0.05	1	0.98098	72	1368	72	71856	1440	757
12097	0.05	1	0.98098	72	1368	72	71856	1440	758

Table 4.11 Example results of similarity measurement (continue).

Testing object No.	Precision	Recall	Accuracy	Relevant	Irrelevant	Relevant total	Irrelevant total	Retrieved total	Corrected class
12098	0.05	1	0.98098	72	1368	72	71856	1440	758
12099	0.05	1	0.98098	72	1368	72	71856	1440	758
12100	0.05	1	0.98098	72	1368	72	71856	1440	758
12101	0.05	1	0.98098	72	1368	72	71856	1440	758
12102	0.05	1	0.98098	72	1368	72	71856	1440	758
12103	0.05	1	0.98098	72	1368	72	71856	1440	758
12104	0.05	1	0.98098	72	1368	72	71856	1440	758
12105	0.05	1	0.98098	72	1368	72	71856	1440	758
12106	0.05	1	0.98098	72	1368	72	71856	1440	758
12107	0.05	1	0.98098	72	1368	72	71856	1440	758
12108	0.05	1	0.98098	72	1368	72	71856	1440	758
12109	0.05	1	0.98098	72	1368	72	71856	1440	758
12110	0.05	1	0.98098	72	1368	72	71856	1440	758
12111	0.05	1	0.98098	72	1368	72	71856	1440	758
12112	0.05	1	0.98098	72	1368	72	71856	1440	758
12113	0.05	1	0.98098	72	1368	72	71856	1440	759
12114	0.05	1	0.98098	72	1368	72	71856	1440	759
12115	0.05	1	0.98098	72	1368	72	71856	1440	759
12116	0.05	1	0.98098	72	1368	72	71856	1440	759
12117	0.05	1	0.98098	72	1368	72	71856	1440	759
12118	0.05	1	0.98098	72	1368	72	71856	1440	759
12119	0.05	1	0.98098	72	1368	72	71856	1440	759
12120	0.05	1	0.98098	72	1368	72	71856	1440	759
12121	0.05	1	0.98098	72	1368	72	71856	1440	759
12122	0.05	1	0.98098	72	1368	72	71856	1440	759
12123	0.05	1	0.98098	72	1368	72	71856	1440	759
12124	0.05	1	0.98098	72	1368	72	71856	1440	759
12125	0.05	1	0.98098	72	1368	72	71856	1440	759
12126	0.05	1	0.98098	72	1368	72	71856	1440	759
12127	0.05	1	0.98098	72	1368	72	71856	1440	759
12128	0.05	1	0.98098	72	1368	72	71856	1440	759
12129	0.05	1	0.98098	72	1368	72	71856	1440	760
12130	0.05	1	0.98098	72	1368	72	71856	1440	760
12131	0.05	1	0.98098	72	1368	72	71856	1440	760
12132	0.05	1	0.98098	72	1368	72	71856	1440	760
12133	0.05	1	0.98098	72	1368	72	71856	1440	760
12134	0.05	1	0.98098	72	1368	72	71856	1440	760
12135	0.05	1	0.98098	72	1368	72	71856	1440	760
12136	0.05	1	0.98098	72	1368	72	71856	1440	760
12137	0.05	1	0.98098	72	1368	72	71856	1440	760

Table 4.11 Example results of similarity measurement (continue).

Testing object No.	Precision	Recall	Accuracy	Relevant	Irrelevant	Relevant total	Irrelevant total	Retrieved total	Corrected class
12138	0.05	1	0.98098	72	1368	72	71856	1440	760
12139	0.05	1	0.98098	72	1368	72	71856	1440	760
12140	0.05	1	0.98098	72	1368	72	71856	1440	760
12141	0.05	1	0.98098	72	1368	72	71856	1440	760
12142	0.05	1	0.98098	72	1368	72	71856	1440	760
12143	0.05	1	0.98098	72	1368	72	71856	1440	760
12144	0.05	1	0.98098	72	1368	72	71856	1440	760
12145	0.05	1	0.98098	72	1368	72	71856	1440	761
12146	0.05	1	0.98098	72	1368	72	71856	1440	761
12147	0.05	1	0.98098	72	1368	72	71856	1440	761
12148	0.05	1	0.98098	72	1368	72	71856	1440	761
12149	0.05	1	0.98098	72	1368	72	71856	1440	761
12150	0.05	1	0.98098	72	1368	72	71856	1440	761
12151	0.05	1	0.98098	72	1368	72	71856	1440	761
12152	0.05	1	0.98098	72	1368	72	71856	1440	761
12153	0.05	1	0.98098	72	1368	72	71856	1440	761
12154	0.05	1	0.98098	72	1368	72	71856	1440	761
12155	0.05	1	0.98098	72	1368	72	71856	1440	761
12156	0.05	1	0.98098	72	1368	72	71856	1440	761
12157	0.05	1	0.98098	72	1368	72	71856	1440	761
12158	0.05	1	0.98098	72	1368	72	71856	1440	761
12159	0.05	1	0.98098	72	1368	72	71856	1440	761
12160	0.05	1	0.98098	72	1368	72	71856	1440	761
12161	0.011111	0.22222	0.97942	16	1424	72	71856	1440	762
12162	0.038889	0.77778	0.98054	56	1384	72	71856	1440	762
12163	0.038889	0.77778	0.98054	56	1384	72	71856	1440	762
12164	0.038889	0.77778	0.98054	56	1384	72	71856	1440	762
12165	0.011111	0.22222	0.97942	16	1424	72	71856	1440	762
12166	0.038889	0.77778	0.98054	56	1384	72	71856	1440	762
12167	0.038889	0.77778	0.98054	56	1384	72	71856	1440	762
12168	0.038889	0.77778	0.98054	56	1384	72	71856	1440	762
12169	0.011111	0.22222	0.97942	16	1424	72	71856	1440	762
12170	0.038889	0.77778	0.98054	56	1384	72	71856	1440	762
12171	0.038889	0.77778	0.98054	56	1384	72	71856	1440	762
12172	0.038889	0.77778	0.98054	56	1384	72	71856	1440	762
12173	0.011111	0.22222	0.97942	16	1424	72	71856	1440	762
12174	0.038889	0.77778	0.98054	56	1384	72	71856	1440	762
12175	0.038889	0.77778	0.98054	56	1384	72	71856	1440	762
12176	0.011111	0.22222	0.97942	16	1424	72	71856	1440	762
12177	0.05	1	0.98098	72	1368	72	71856	1440	763

Table 4.11 Example results of similarity measurement (continue).

Testing object No.	Precision	Recall	Accuracy	Relevant	Irrelevant	Relevant total	Irrelevant total	Retrieved total	Corrected class
12178	0.05	1	0.98098	72	1368	72	71856	1440	763
12179	0.05	1	0.98098	72	1368	72	71856	1440	763
12180	0.05	1	0.98098	72	1368	72	71856	1440	763
12181	0.05	1	0.98098	72	1368	72	71856	1440	763
12182	0.050314	1	0.98111	72	1359	72	71856	1431	763
12183	0.05	1	0.98098	72	1368	72	71856	1440	763
12184	0.05	1	0.98098	72	1368	72	71856	1440	763
12185	0.05	1	0.98098	72	1368	72	71856	1440	763
12186	0.0511	1	0.98141	72	1337	72	71856	1409	763
12187	0.05	1	0.98098	72	1368	72	71856	1440	763
12188	0.05	1	0.98098	72	1368	72	71856	1440	763
12189	0.05	1	0.98098	72	1368	72	71856	1440	763
12190	0.05	1	0.98098	72	1368	72	71856	1440	763
12191	0.05	1	0.98098	72	1368	72	71856	1440	763
12192	0.05	1	0.98098	72	1368	72	71856	1440	763
12193	0.05	1	0.98098	72	1368	72	71856	1440	764
12194	0.05	1	0.98098	72	1368	72	71856	1440	764
12195	0.05	1	0.98098	72	1368	72	71856	1440	764
12196	0.05	1	0.98098	72	1368	72	71856	1440	764
12197	0.05	1	0.98098	72	1368	72	71856	1440	764
12198	0.05	1	0.98098	72	1368	72	71856	1440	764
12199	0.05	1	0.98098	72	1368	72	71856	1440	764
12200	0.05	1	0.98098	72	1368	72	71856	1440	764
12201	0.05	1	0.98098	72	1368	72	71856	1440	764
12202	0.05	1	0.98098	72	1368	72	71856	1440	764
12203	0.05	1	0.98098	72	1368	72	71856	1440	764
12204	0.05	1	0.98098	72	1368	72	71856	1440	764
12205	0.05	1	0.98098	72	1368	72	71856	1440	764
12206	0.05	1	0.98098	72	1368	72	71856	1440	764
12207	0.05	1	0.98098	72	1368	72	71856	1440	764
12208	0.05	1	0.98098	72	1368	72	71856	1440	764
12209	0.050244	1	0.98108	72	1361	72	71856	1433	765
12210	0.050035	1	0.98099	72	1367	72	71856	1439	765
12211	0.050491	1	0.98118	72	1354	72	71856	1426	765
12212	0.05	1	0.98098	72	1368	72	71856	1440	765
12213	0.05007	1	0.98101	72	1366	72	71856	1438	765
12214	0.050035	1	0.98099	72	1367	72	71856	1439	765
12215	0.05007	1	0.98101	72	1366	72	71856	1438	765
12216	0.05042	1	0.98115	72	1356	72	71856	1428	765
12217	0.05007	1	0.98101	72	1366	72	71856	1438	765

Table 4.11 Example results of similarity measurement (continue).

Testing object No.	Precision	Recall	Accuracy	Relevant	Irrelevant	Relevant total	Irrelevant total	Retrieved total	Corrected class
12218	0.050035	1	0.98099	72	1367	72	71856	1439	765
12219	0.050174	1	0.98105	72	1363	72	71856	1435	765
12220	0.0511	1	0.98141	72	1337	72	71856	1409	765
12221	0.05007	1	0.98101	72	1366	72	71856	1438	765
…	…	…	…	…	…	…	…	…	…
15932	0.05	1	0.98098	72	1368	72	71856	1440	997
15933	0.05	1	0.98098	72	1368	72	71856	1440	997
15934	0.05	1	0.98098	72	1368	72	71856	1440	997
15935	0.05	1	0.98098	72	1368	72	71856	1440	997
15936	0.05	1	0.98098	72	1368	72	71856	1440	997
15937	0.05007	1	0.98101	72	1366	72	71856	1438	998
15938	0.050174	1	0.98105	72	1363	72	71856	1435	998
15939	0.05074	1	0.98127	72	1347	72	71856	1419	998
15940	0.05007	1	0.98101	72	1366	72	71856	1438	998
15941	0.050314	1	0.98111	72	1359	72	71856	1431	998
15942	0.050955	1	0.98136	72	1341	72	71856	1413	998
15943	0.051429	1	0.98154	72	1328	72	71856	1400	998
15944	0.050035	1	0.98099	72	1367	72	71856	1439	998
15945	0.050244	1	0.98108	72	1361	72	71856	1433	998
15947	0.050526	1	0.98119	72	1353	72	71856	1425	998
15948	0.05007	1	0.98101	72	1366	72	71856	1438	998
15949	0.04934	0.98611	0.98097	71	1368	72	71856	1439	998
15950	0.05007	1	0.98101	72	1366	72	71856	1438	998
15951	0.050633	1	0.98123	72	1350	72	71856	1422	998
15952	0.050491	1	0.98118	72	1354	72	71856	1426	998
15953	0.05	1	0.98098	72	1368	72	71856	1440	999
15954	0.05	1	0.98098	72	1368	72	71856	1440	999
15955	0.05	1	0.98098	72	1368	72	71856	1440	999
15956	0.05	1	0.98098	72	1368	72	71856	1440	999
15957	0.05	1	0.98098	72	1368	72	71856	1440	999
15958	0.05	1	0.98098	72	1368	72	71856	1440	999
15959	0.05	1	0.98098	72	1368	72	71856	1440	999
15960	0.05	1	0.98098	72	1368	72	71856	1440	999
15961	0.05	1	0.98098	72	1368	72	71856	1440	999
15962	0.05	1	0.98098	72	1368	72	71856	1440	999
15963	0.05	1	0.98098	72	1368	72	71856	1440	999
15964	0.05	1	0.98098	72	1368	72	71856	1440	999
15965	0.05	1	0.98098	72	1368	72	71856	1440	999
15966	0.05	1	0.98098	72	1368	72	71856	1440	999
15967	0.05	1	0.98098	72	1368	72	71856	1440	999

Table 4.11 Example results of similarity measurement (continue).

Testing object No.	Precision	Recall	Accuracy	Relevant	Irrelevant	Relevant total	Irrelevant total	Retrieved total	Corrected class
15968	0.05	1	0.98098	72	1368	72	71856	1440	999
15969	0.05	1	0.98098	72	1368	72	71856	1440	1000
15970	0.05	1	0.98098	72	1368	72	71856	1440	1000
15971	0.05	1	0.98098	72	1368	72	71856	1440	1000
15972	0.05	1	0.98098	72	1368	72	71856	1440	1000
15973	0.05	1	0.98098	72	1368	72	71856	1440	1000
15974	0.05	1	0.98098	72	1368	72	71856	1440	1000
15975	0.05	1	0.98098	72	1368	72	71856	1440	1000
15976	0.05	1	0.98098	72	1368	72	71856	1440	1000
15977	0.05	1	0.98098	72	1368	72	71856	1440	1000
15978	0.05	1	0.98098	72	1368	72	71856	1440	1000
15979	0.05	1	0.98098	72	1368	72	71856	1440	1000
15980	0.05	1	0.98098	72	1368	72	71856	1440	1000
15981	0.05	1	0.98098	72	1368	72	71856	1440	1000
15982	0.05	1	0.98098	72	1368	72	71856	1440	1000
15983	0.05	1	0.98098	72	1368	72	71856	1440	1000
15984	0.05	1	0.98098	72	1368	72	71856	1440	1000

Table 4.12 overall results of similarity measurement.

total testing data	15,984 objects
total signature number	71,928 signatures
total usage times	19,365 seconds
total usage memory for signature	8,631,360 bytes
average precision	0.1619
average recall	0.9651
average accuracy	0.9851

CHAPTER V
DISCUSSION

This research proposed the rough set data mining of shape based image retrieval using the Fourier descriptor feature extraction.

From chapter IV, the results between a rough set method and a similarity measurement were found that the data miner techniques can be applied to the field of image retrieval. A rough set method can reduce the usage time for retrieving, yet the precision and the accuracy of a rough set method is not different much with a similarity measurement. The usage memory of a rough set method can be reduced depend on reduction rules; beside, one of a similarity measurement is depend on a number of image data in database. The recall is very interesting; the overall recall values between a rough set method and a similarity measurement are very different. This can be explained that because the rough set method is focused in a rule based system while, the similarity measurement is interest in a pattern matching, so it retrieved all of image data that match or similar to an input image, but the rough set method retrieves image data that be accepted by rules. These are a cause why recalls between them very different. This recall issue was resolved by improving quantization process. After that, the recalls were better than previous, but they were not good enough. On the other hand, when recall values were improved, the precision values became lower significantly.

5.1 Tools in the research

The Matlab is a numerical computer environment. Developed by the MathWorks, Matlab allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces and interfacing with programs written in other languages. It is easy to use Matlab for researching and calculating.

5.2 Problem during researching

In the preprocessing stage, we found that the binary images for researching in the same class are some objects in the images were split more than 1 part of objects cause of rotating. The objects that split from main objects in images have a small area, so it will be cut off.

5.3 Advantages

The rough set data mining is used to apply to image retrieval. It discovered the knowledge from the raw image feature extraction data and generated the rules based system. This method will reduce the usage memory for classifying data and usage times in the retrieving stage. Moreover, the rough set data mining can be applied to find some pattern of other image feature extraction.

5.4 Disadvantages

The training stage of a rough set data mining uses the much time but it depends on amount of data. In this experiment, use the many class images (999 classes). Then, the pattern of various classes has a same involuntarily pattern. Some pattern is classified into other class that is the cause of miss retrieving data in the testing stage.

CHAPTER VI
CONCLUSION AND RECOMMEDATION

6.1 Conclusion

This research uses the rough data mining for image retrieval. The results were compared to the similarity measurement. From the chapter IV, the usage times and the usage memory are less than the similarity measurement method. The precision and accuracy are little different. The recalls of two methods are absolutely difference. In other words, the average recall of rough set method is 0.261 but the average recall of similarity measurement method is 0.9651. The similarity measurement retrieved data that is all relevant data in each retrieving for the various amount of irrelevant data in each retrieving. From the table 4.2 and 4.4, we will see the total retrieved data in each row of two tables are very different. The rough data mining method has the total retrieved data less than or equal 100 data in each retrieving but the similarity measurement has the total retrieved data more than 1000 data.

We found that the rough set data mining method uses the resource less than the similarity measurement but the overall is a little different. The usage time, usage memory and accuracy are advantage to the rough set data mining method.

6.2 Recommendation

The rough set data mining method can be used with other image pattern feature extraction. The thing that we have to consider is the image feature extraction we want to discover the knowledge from raw data. The accuracy depends on the image feature that we choose to extract from image.

The training time is the much usage time for training process. For the large image data, we can separate the classification stage into various steps such as first step, we use some features for rough classifying and second step, we use the all feature for fine classifying to decrease training time and testing time in learning and evaluating stage.

REFERENCES

1 P.N. Tan, Michael Steinbach, Vipin Kumar. Introduction to Data Mining, Pearson International Edition 2006.

2 G. Brunner, Structure Features for Content-Based Image Retrieval and Classification Problems, Doctoral Thesis, 2006.

3 Image Retrieval [Online]. Available from: http://en.wikipedia.org/wiki/Content-based_image_retrieval [Accessed 2007 July 15].

4 D. Zhang, G. Lu. A Comparative Study on Shape Retrieval Using Fourier Descriptors with Different Shape Signatures, In Proc. of International Conference on Intelligent Multimedia and Distance Education (ICIMADE01), pp.1-9, Fargo, ND, USA, June 1-3, 2001.

5 A. Folkers, and H. Samet, Content-based Image Retrieval Using Fourier Descriptors on a Logo Database, ICPR (3), pp. 521-524, 2002.

6 P. Phokharatkul, Invariant Thai Printed Characters Recognition, Doctoral Thesis, 2002.

7 Z. Akbar, Marketing data classification using Johnson's algorithm, Fourth International Conference *Knowledge Discovery and Discrete Mathematics*, pp. 257-266, Sep. 3-6, 2003.

8 S. Tsumoto, Pawlak Rough Set Model, Medical Reasoning and Rule Mining, Lecture Notes in Computer Science, Springer, pp. 53-70, November 03, 2006.

APPENDICES

APPENDIX A
Some MATLAB Code

Preprocess.m

```
clear;
disp('preprocessing raw data......');
for i=1:1000
   atr_index = 1;
   eval(sprintf('load ./trainData/loadTraingData%d;',i));
   if sum(sum(trainingData{1,1})) == 0
      %do nothing
   else
      eval(sprintf('load "./rawdata/%d"',i));
      [row col] = size(image_fdes_sum_for_shape);

      for l = 1 : col
         [R C] = size(image_fdes_sum_for_shape{l});
         test_result_circular(l) =
image_fdes_sum_for_shape{l}(2)/sum(image_fdes_sum_for_shape{l}(2:R));
         test_result_symmetry(l) =
sum(image_fdes_sum_for_shape{l}(2:2:R))/sum(image_fdes_sum_for_shape{l}(2:R));
         temp = checkMod(image_fdes_sum_for_shape{l}(3:R),3);
         test_result_kFold(l) = sum(temp)/sum(image_fdes_sum_for_shape{l}(2:R));
         biRtemp = real(image_fdes_sum_A{l} + image_fdes_sum_B{l});
         biItemp = imag(image_fdes_sum_A{l} + image_fdes_sum_B{l});
         biSumtemp = sum(abs(biRtemp)+abs(biItemp));
         test_result_bilateral(l) = sum(abs(biRtemp))/biSumtemp;
      end;

      [row col] = size(image_fdes_sum);


      for j = 1:row
         index = 1;
         temp_image_fdes_sum = image_fdes_sum(j,2:16);
         for k = 1:15
             if(temp_image_fdes_sum(k) > 1)
                attribute(atr_index,index) = roundn(temp_image_fdes_sum(k),0);
             elseif((temp_image_fdes_sum(k) <= 1)&(temp_image_fdes_sum(k) > 0.5))
                attribute(atr_index,index) = 1;
            elseif((temp_image_fdes_sum(k) <= 0.5) & (temp_image_fdes_sum(k) > 0.4))
                attribute(atr_index,index) = 2;
            elseif((temp_image_fdes_sum(k) <= 0.4) & (temp_image_fdes_sum(k) > 0.3))
                attribute(atr_index,index) = 3;
            elseif((temp_image_fdes_sum(k) <= 0.3) & (temp_image_fdes_sum(k) > 0.2))
               attribute(atr_index,index) = 4;
            elseif((temp_image_fdes_sum(k) <= 0.2) & (temp_image_fdes_sum(k) > 0.1))
                attribute(atr_index,index) = 5;
            elseif((temp_image_fdes_sum(k) <= 0.1) & (temp_image_fdes_sum(k) > 0.09))
               attribute(atr_index,index) = 6;
            elseif((temp_image_fdes_sum(k) <= 0.09) & (temp_image_fdes_sum(k) > 0.08))
               attribute(atr_index,index) = 7;
            elseif((temp_image_fdes_sum(k) <= 0.08) & (temp_image_fdes_sum(k) > 0.07))
```

```
                attribute(atr_index,index) = 8;
             elseif((temp_image_fdes_sum(k) <= 0.07) & (temp_image_fdes_sum(k) > 0.06))
                attribute(atr_index,index) = 9;
             elseif((temp_image_fdes_sum(k) <= 0.06) & (temp_image_fdes_sum(k) > 0.05))
                attribute(atr_index, index) = 10;
             elseif((temp_image_fdes_sum(k) <= 0.05) & (temp_image_fdes_sum(k) > 0.04))
                attribute(atr_index, index) = 11;
             elseif((temp_image_fdes_sum(k) <= 0.04) & (temp_image_fdes_sum(k) > 0.03))
                attribute(atr_index, index) = 12;
             elseif((temp_image_fdes_sum(k) <= 0.03) & (temp_image_fdes_sum(k) > 0.02))
                attribute(atr_index, index) = 13;
             elseif((temp_image_fdes_sum(k) <= 0.02) & (temp_image_fdes_sum(k) > 0.01))
                attribute(atr_index, index) = 14;
             else
                attribute(atr_index, index) = 15;
             end;

           index = index + 1;
        end;

         atr_index = atr_index + 1;
      end;
      output_attribute = i;
      eval(sprintf('save ("./initial_data/%d","attribute","output_attribute")',i));
   end;
end;
```

ProcessDiscernibility.m

```
clear;
clc;

load 'dataUniqueClass';

DiscernibilityMatrix = cell(1,length(uniqueClass));
for i = 1:length(uniqueClass)

   for j = 1:length(uniqueClass)
      if(~(i == j))
         DiscernibilityMatrix{j} = findDiffField(uniqueClass(i,:),uniqueClass(j,:));

      else
         DiscernibilityMatrix{j} = [];
      end;
   end;
   eval(sprintf('save("./discernibilities/discernibilities%d","DiscernibilityMatrix");',i));
   eval(sprintf('disp("process discernibility......%d");',i));
end;

clear;
```

ProcessDiscFunc.m

```
clear;
clc;
load './classes/classes';
%find unique class and unsort rows for real class not sorted rows
```

```
[bEquiv ,mEquiv ,nEquiv] = unique(equivClass(:,1:15),'rows');
uniqueClass = equivClass(sort(mEquiv),1:15);
lenClass = length(uniqueClass);
%////////////////////////////////////////////////////////////////
clear bEquiv mEquiv nEquiv equivClass;
for im = 1:lenClass
   eval(sprintf('load "./discernibilities/discernibilities%d";',im));
   index = 1;
   sortCellObj = sortCell(DiscernibilityMatrix);
   for i = 1:length(sortCellObj)
      if(i ~= 1)%coz discernibilities matrix of class 1
         if(index == 1)
            discFunction{index} = DiscernibilityMatrix{sortCellObj(i)};
            index = index + 1;
         else
            temp = intersect(discFunction{1},DiscernibilityMatrix{sortCellObj(i)});
            if(isempty(temp))
               dontCare = 0;
               for j = 1:length(discFunction)
                  temp = intersect(discFunction{j},DiscernibilityMatrix{sortCellObj(i)});
                  if(~isempty(temp)) & (isequal(temp,discFunction{j}))
                     dontCare = 1;
                  end;
               end;
               if(dontCare == 0)
                  discFunction{index} = DiscernibilityMatrix{sortCellObj(i)};
                  index = index + 1;
               end;
            end;
         end;
      end;
   end;

eval(sprintf('save("./discFunctions/discFunctions%d","discFunction");',im));
clear discFunction sortCellObj;
eval(sprintf('disp("process discFunction.....%d")',im));
end;
```

ProcessEquivClass.m

```
clear;
clc;
load './objects/objects';
index = 1;
equivClass(index,:) = object(1,:);
index = index + 1;
for i = 2:length(object)
   if((checkObjects(object(i,:),equivClass)) == 0)
      equivClass(index,:) = object(i,:);
      index = index + 1;
   end;
   eval(sprintf('disp("processEquivClass.....%d");',index));
end;
save('./classes/classes','equivClass');
```

JohnsonAlgo.m

```
clear;
clc;
load './classes/classes';
load 'dataUniqueClass';

lenClass = length(uniqueClass);
for im = 1:lenClass
   eval(sprintf('load "./discernibilities/discernibilities%d";',im));
   index = 1;
   disTempIndex = 1;
   atrCount = [];
   disTemp{disTempIndex} = [];
   for j = 1:15
      atrCount(j) = 0;
   end;

   for k = 1:length(DiscernibilityMatrix)
         if ~isempty(DiscernibilityMatrix{k})
            for l = 1:length(DiscernibilityMatrix{k})
               atrCount(DiscernibilityMatrix{k}(l)) = 1 + atrCount(DiscernibilityMatrix{k}(l));
            end;
            disTemp{disTempIndex} = DiscernibilityMatrix{k};
            disTempIndex = disTempIndex + 1;

         end;
   end;



   while checkEmpty(disTemp)
      [value,i] = max(atrCount);

      discFunction{im}(index) = i;

      setEmptyValue = i;

      index = index + 1;

      for disIndex = 1:length(disTemp)
         if ~isempty(disTemp{disIndex})
            if intersect(setEmptyValue,disTemp{disIndex}) == setEmptyValue
               disTemp{disIndex} = [];
            end;
         end;
      end;
      atrCount(:) = 0;


      for k = 1:length(disTemp)
         if ~isempty(disTemp{k})
            for l = 1:length(disTemp{k})
               atrCount(disTemp{k}(l)) = 1 + atrCount(disTemp{k}(l));
            end;
         end;
      end;
```

```
    end;
    clear disTemp

eval(sprintf('disp("process discFunction.....%d")',im));
end;

save('dataDiscFunction');
```

APPENDIX B
CMCE2010 Submitted Paper

IMAGE RETRIEVAL USING CONTOUR FEATURE WITH ROUGH SET METHOD

Pheerawit Wasinphongwanit
Technology of Information System Management
Faculty of Graduated Studies, Mahidol University
Nakhon Pathom, 73170 Thailand
E-mail: pheerawit@hotmail.com

Pisit Phokharatkul
Department of Computer Engineering
Faculty of Engineering, Mahidol University
Nakhon Pathom, 73170 Thailand
E-mail: egpph@mahidol.ac.th

***Abstract*—Content based image retrieval (CBIR) is well-known in the field of image retrieval. It uses contents of an image from image processing and analysis to retrieve images that users were looking for from an image search. Shape based image retrieval was focused in this paper. A data mining was considered to find knowledge in the image database. Fourier descriptor is a most technique to extract contour feature of images. It was used to analyze the testing and training images in the preprocessing step. Fourier coefficients were quantized into multiple attributes and rough set theory was used to generate a rule-based system. Rough set theory is used as a data mining technique. It was compared to similarity measurement. We use 15,984 testing image data with 71,928 training image data in this experiment. A total usage time of rough set method is 13,286 seconds. A total usage time of similarity measurement is 19,365 seconds. A total usage memory of rough set method and similarity measurement are 2.8 Mbytes and 8.6 Mbytes respectively. An average precision, an average recall and an average accuracy of rough set method are 0.1297, 0.261 and 0.9971. An average precision, an average recall and an average accuracy of similarity measurement are 0.1619, 0.9651 and 0.9852. The rough set method is advantage to the usage time and the usage memory.**

Keywords- Rough set; Image retrieval; Fourier descriptors

I. INTRODUCTION

We use a database system for storage information. The type of the information such as images, texts, music and movies that a difference kind of encoding and file types. The image data can use some methods of image retrieval technologies for query database of digital images. The most traditional and common methods of image retrieval utilize some methods of adding metadata such as captioning, keywords or descriptions to the images. The content based image retrieval is a method that uses the content of digital image for querying image. The contents of images are used to process in numerical computing of digital image retrieval. For example, we use a color for classify the groups or species of flowers. We have many features of colors of each image and we cannot use all color features to process in all digital images. We might be used all color features to process it but we use over time for processing. However, we can use the data mining for solve this problem.

Data mining involves fitting models to or determining patterns from ob served data. The fitted models play the role of inferred knowledge. Data mining techniques are deployed to scour large databases in order to find novel and useful patterns that might otherwise remain unknown [1]. A particular data mining algorithm is usually an instantiation of the model, preference, and search components.

In the field of classification we can use the rough set theory for data mining technique apply to image retrieval. This method use the features [2] (such as color, texture, shape and layout) of images data, which are done preprocessing before, for mining images [3] data and classify images data that receive from resources. The rough set theory makes the rule sets from data for classifying. We not necessary to use all data to processing but we can use some data that unique from other for classifying and processing.

II. PRE-PROCESSING

The preprocessing is to extract coordinates of the contour binary images. The block diagram for preprocessing is shown in Figure 1.

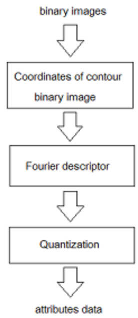

Figure 1. Preprocessing of contour feature.

The first step is to find coordinates of the contour binary image by using 8-connectivity contour tracing technique. Fourier descriptor [4, 5] is the most techniques for extracting the contents of image contours. Coordinates of the contour binary image will be transformed to complex value to calculate the fourier coefficients. We can make sure that the image invariant for translation and rotating by moving the centroid to origin (0,0) as following:

$$x_i' = x_i - x_c \text{, and } y_i' = y_i - y_c \text{, with } i = 1, \ldots, L \quad (1)$$

where L denotes the number of boundary points,
x_c is coordinate x of centroid, and
y_c is coordinate y of centroid.

The discrete Fourier Transform of image contour $C_{(x,y)}$ is given by:

$$a_k = \frac{1}{L}\sum_{m=0}^{L-1} x_m e^{-j2\pi km/L} \quad (2)$$

$$b_k = \frac{1}{L}\sum_{m=0}^{L-1} y_m e^{-j2\pi km/L} \quad (3)$$

where k = 0, 1, …, L-1, and
x_m, y_m are the co-ordinates of boundary contour.

a_k and b_k have to be changed to polar form and normalize.

$$r_k = \sqrt{|a_k|^2 + |b_k|^2} \quad (4)$$

From this polar form, cut DC component (r_0) and normalize.

$$s_k = \frac{r_k}{r_1} \text{, with } k = 1, \ldots, L\text{-}1 \quad (5)$$

Fourier Descriptor (s_k) is now invariant to translation, rotation and starting point. The Fourier coefficients have to be changed for classification and rough rule based generation by Quantization. Quantization is the procedure of constraining something from a continuous set of values (such as the real numbers) to a discrete set (such as the integers).

III. ROUGH SET THEORY

Rough set Theory [6, 7, 8] is proposed by Pawlak in the early 1980s as a mathematical tool to deal with uncertainty. It is an extension of conventional set theory that supports approximations in terms of a pair of sets which give the lower and the upper approximation of the original set. In the standard version of rough set theory (Pawlak 1991), the lower and upper approximation sets are crisp sets, but in other variations, the approximating sets may be fuzzy sets. An information system contains a data set represented in a table. Each table row represents an object. Every column represents an attribute that can be measured for each object.

An information system, S, is defined as:

$$S = (U, A, \{V_a : \in A\}) \quad (6)$$

Where U is a non-empty finite set of objects,
A is a finite, nonempty set of mappings,
$a: U \to V_a$, and
V_a is the value set of a.

One of the most important concepts of rough set theory is indiscernibility, which is used to define equivalence classes for the objects. Each subset of attributes $B \subseteq A$ defines an equivalence relation, denote IND(B) and will be called an indiscernibility relation. This indiscernibility relation is defined as:

$$IND(B) = \{(x, y) \in U \times U : \mathcal{F}(x, a) = \mathcal{F}(y, a) \text{ for every } a \in B\} \quad (7)$$

If we add the decision attributes into the information into the information systems, this system is called a decision system. The decision system is defined (Bjanger, 1999)

$$S = (U, A \cup \{d\}) \quad (8)$$

where $d \notin A$ is a distinguish attribute called the decision attribute.

Suppose an information system, $S = (U, A)$, $X \subseteq U$, and $B \subseteq A$. Let us define two operations assigning to every $X \subseteq U$ two sets $B_*(X)$ and $B^*(X)$, called the B-lower and the B-upper approximation of X, respectively. The boundary approximation defined as follows:

$$B_*X = \{x \mid [x]_B \subseteq X\} \quad (9)$$

Is the lower approximation or positive region of X, and

$$B^*X = \{x \mid [x]_B \cap X \neq \emptyset\} \quad (10)$$

Is the upper approximation or possible region of X.

The lower approximation is the set containing all objects for which the equivalence class corresponding to the object, which is a subset of the set we would like to approximate. This set contains all objects, which with certainty belong to the set X.

The upper approximation is the set containing the objects for which the intersection of the object's equivalence class, and this set we would like to approximate is not the empty set. This set contains all objects, which possibly belong to the set X.

Indiscernibility relation reduces the data by identifying equivalence classes, i.e., objects that are indiscernible, using the available attributes. Only one element of the equivalence class is needed to represent the entire class. Reduction can also be done by keeping only those attributes that preserve the indiscernibilitty relation and, consequently, set approximation. Sometimes it happens that the set of attributes provides more information about objects then is needed to distinguish them. In such a situation some of the attributes may be reduced without losing the ability to group the objects. In the rough set approach, there are two fundamental concepts used in the reduction of knowledge, a reduct and the core. Intuitively, a reduct is one essential part of an information system, which can discern all objects that are discernible by the original information system. The core comprises the attributes that are indispensable to the discrimination of objects, the attributes that are contained in all the reducts.

IV. ROUGH SET METHOD FOR IMAGE RETRIEVAL

The Fourier descriptors of training images are floating coefficient values. They have to be changed for classification and rough rule based generation by quantization. The example of Fourier coefficient attributes and quantized Fourier coefficient attributes as shown in Table I and II, respectively.

Table I Example of Fourier coefficient attributes.

Fourier coefficients attributes					Decision Attribute
a1	a2	…	a14	a15	Image Class
0.37004	0.12003	…	0.006674	0.006501	17
0.009883	0.10146	…	0.004021	0.00192	18
0.007551	0.11077	…	0.005099	0.002027	19
0.22829	0.075504	…	0.003426	0.003199	20
0.009212	0.10755	…	0.004867	0.001223	21
0.15389	0.044848	…	0.004842	0.003655	22
0.28234	0.1056	…	0.007637	0.002976	23
0.021768	0.095823	…	0.003062	0.00239	24
0.060867	0.068595	…	0.002665	0.000726	25
0.004061	0.10503	…	0.004032	0.001807	26
0.068566	0.098481	…	0.004628	0.003981	27
0.27086	0.19462	…	0.006979	0.006221	28
…	…	…	…	…	…

Table II Example of quantized Fourier coefficient attributes.

Quantized Fourier descriptors attributes					Decision Attribute
a1	a2	…	a14	a15	Decision
15	5	…	15	15	19
4	8	…	15	15	20
15	5	…	15	15	21
5	11	…	15	15	22
4	5	…	15	15	23
13	6	…	15	15	24
9	9	…	15	15	25
15	5	…	15	15	26
9	6	…	15	15	27
4	5	…	15	15	28
…	…	…	…	…	…

The next step, we use the quantized Fourier attributes to create equivalent classes for generating the rule based system as shown in Table III.

Table III Example of equivalent classes.

Object attributes							Decisions
No.	A1	A2	A3	…	A14	A15	D
1	5	6	9	…	15	15	1
2	5	6	8	…	15	15	1
3	3	4	5	…	15	15	{1, 2}
4	3	4	5	…	15	15	2
…	…	…	…	…	…	…	…

Finally, the rough set theory is used to discover knowledge from data attributes and generate rule based from equivalence classes. The discernibility matrix is created for finding dispensable classes and reduction of training data. The core attribute is found from dispensable classes. We use Johnson reduction method [7] to reduce the attributes and generated rule based from equivalence classes. The attribute which is reduced will be defined to zero as shown in Table IV.

Table IV Example of reduced attributes for rough rule based.

Attributes							Decisions
a1	a2	a3	…	a13	a14	a15	D
0	6	10	…	0	0	0	1
0	6	9	…	0	0	0	1
6	6	10	…	0	0	0	1
…	…	…	…	…	…	…	…

We can obtain the decision rules from an information system or a reduction of condition attributes. Some condition values may be unnecessary in the decision rules, so we can remove them from the rules or change them into "don't care" values. The process of conditional value removed is called values reduction.

Let $S = (U, A, C, D)$ be an information system with n objects, where U is a finite set of objects. A is a set of attributes further classified into disjointed condition attributes C and decision attributes $D, A = C \cup D, V = \cup V_a, V_a$ is the domain of attribute $a \in A$. Hence, decision rules are often presented as implications and are often called "If … then …". If a, b, c, d are the condition attributes of set A, D is decision attribute. We can express the rules as

Rule 1
IF a = 1 THEN D = 1 (for the class 1)

Rule 2
IF b = S and c = M THEN D= 2 (for the class 2)

……………………………………………

Rule n
IF c = L THEN D = n (for the class n)

IV. SIMILARITY MEASUREMNT

The similarity measurement is the one of the retrieved techniques for shape based image retrieval with Fourier descriptor. A model shape indexed by Fourier Descriptor feature $f_m = [f_m^1, f_m^2, \dots, f_m^N]$ and a testing data shape indexed by Fourier Descriptor feature $f_d = [f_d^1, f_d^2, \dots, f_d^N]$, since both features are normalized as to translation, rotation and scale, the Euclidean distance between the two feature vectors can be used as the similarity measurement.

$$d = (\sum_{i=0}^{N} |f_m^i - f_d^i|^2)^{\frac{1}{2}} \quad (16)$$

where N is the truncated number of harmonics needed to index the shape.

V. Results

The results of the rough data mining for image retrieval will be shown in three values of data retrieval evaluation. They are precision, recall and accuracy. We use the similarity measurement of image signature to compare to rough rule base system. For each retrieving, we only retrieved 20 first unique decisions.

In the context of classification tasks, the terms true positives, true negatives, false positives and false negatives are used to compare the given classification of an item (the class label assigned to the item by a classifier) with the desired correct classification (the class the item actually belongs to). These are illustrated by the table below:

Table V The terms of classification table tasks.

		correct result / classification	
		E1	E2
obtained result / classification	E1	tp (true positive)	fp (false positive)
	E2	fn (false negative)	tn (true negative)

$$\text{Precision} = \frac{tp}{tp+fp} \quad (16)$$

$$\text{Recall} = \frac{tp}{tp+fn} \quad (17)$$

$$\text{Accuracy} = \frac{tp+tn}{tp+tn+fp+fn} \quad (18)$$

Table VI Overall results of rough rules based image retrieval.

total testing data	15,984 objects
total signature number	71,928 signatures
total usage times	19,365 seconds
total usage memory for signature	8,631,360 bytes
average precision	0.1619
average recall	0.9651
average accuracy	0.9851

Table VII overall results of similarity measurement.

total testing data	15,984 objects
total rough rules based number	21,479 rules
total usage times	13,286 seconds
total usage memory for rules based	2,886,948 bytes
average precision	0.1297
average recall	0.261
average accuracy	0.9971

VI. Conclusion

This research uses the rough data mining for image retrieval. The results were compared to the similarity measurement. From the chapter IV, the usage times and the usage memory are less than the similarity measurement method. The precision and accuracy are little different. The recalls of two methods are absolutely difference. In other words, the average recall of rough set method is 0.261 but the average recall of similarity measurement method is 0.9651. The similarity measurement retrieved data that is all relevant data in each retrieving for the various amount of irrelevant data in each retrieving. We have seen the total retrieved data in each row of two tables are very different. The rough data mining method has the total retrieved data less than or equal 100 data in each retrieving but the similarity measurement has the total retrieved data more than 1000 data.

We found that the rough set data mining method uses the resource less than the similarity measurement but the overall is a little different. The usage time, usage memory and accuracy are advantage to the rough set data mining method.

Acknowledgment

First of all, I thank my family for listening and supporting in the last couple of years.

I deeply thank my advisor, Asst. Prof. Dr. Pisit Phokharatkul, whose help, advice and supervision was invaluable.

Finally, I would like to thank the Technology and Information System Management for supporting during the period this research took place is greatly acknowledged.

References

[1] P.N. Tan, Michael Steinbach, Vipin Kumar. Introduction to Data Mining, Pearson International Edition 2006.

[2] G. Brunner, Structure Features for Content-Based Image Retrieval and Classification Problems, Doctoral Thesis, 2006.

[3] Image Retrieval [Online]. Available from: http://en.wikipedia.org /wiki/Content-based_image_retrieval [Accessed 2007 July 15].

[4] D. Zhang and G. Lu. "A Comparative Study on Shape Retrieval Using Fourier Descriptors with Different Shape Signatures," In Proc. of International Conference on Intelligent Multimedia and Distance Education (ICIMADE01), pp.1-9, Fargo, ND, USA, June 1-3, 2001.

[5] A. Folkers, and H. Samet, "Content-Based Image Retrieval Using Fourier Descriptors on a Logo Database," ICPR (3), pp. 521-524, 2002.

[6] P. Phokharatkul, Invariant Thai Printed Characters Recognition, Doctoral Thesis, 2002.

[7] Z. Akbar, "Marketing data classification using Johnson's algorithm," Fourth International Conference *Knowledge Discovery and Discrete Mathematics*, pp. 257-266, Sep. 3-6, 2003.

[8] S. Tsumoto, Pawlak Rough Set Model, Medical Reasoning and Rule Mining, Lecture Notes in Computer Science, Springer, pp. 53-70, November 03, 2006.

BIOGRAPHY

NAME	MR. Pheerawit Wasinphongwanit
DATE OF BIRTH	May 8, 1982
PLACE OF BIRTH	Nakornsithammarat, Thailand
INSTITUTIONS ATTENDED	Pathumwan Institute of Technology, 2000 – 2004: Bachelor of Engineer (Mechatronics)
HOME ADDRESS	Soi Saima 99/69, Rattanathibet Road, Sai - Ma, Muang, Nonthaburi, 11000, Thailand. Tel. 08–7053–3387 E–mail: pheerawit@hotmail.com