VISUAL RERANKING USING MULTIPLE SEARCH ENGINES

By

Dennis Lim Thye Loon

A REPORT SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements for the degree of

BACHELOR OF COMPUTER SCIENCE (HONS)

Faculty of Information and Communication Technology

(Perak Campus)

MAY 2013

DECLARATION OF ORIGINALITY

I declare that this proposal entitled "**VISUAL RERANKING USING MULTIPLE SEARCH ENGINES**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature: _____

Name: _____

Date: _____

ACKNOWLEDGEMENT

First, I would like to give a very grateful thank to my supervisor, Dr. Tan Hung Khoon who given me this opportunity to take part in develop this expert system project. It is the first time for me to involve in a software development project. Besides, he has taught me in many areas no matter in report writing or technical work. Again, a million thanks to him.

Last, I would like to thank the authority of University Tunku Abdul Rahman (UTAR) for providing me with a good environment and facilities to complete this project. I learn a lot from this project and I believe what I learn here is useful for my bright future.

ABSTRACTS

Re-ranking method is a technique to improve initial search result of a single search engine. However, most of them are relied on a single search engine. When such single search engine is produce incorrect image search results based on complex queries given, some ranking method may base on such irrelevant images to produce incorrect re-ranking result.

The main objective in this project is to develop an offline prototype system that used to design a proposed method which can improve the images search result of a single search engine by observe the information obtained from other search results of multiple search engines.

The basic idea of the proposed method is to divide the ranking process into six stages, where in the first stage is to extract the features of top hundred lists of images from multiple search results; the second stage is to cluster each of the extracted features to form a bag-of-words; the third stage is represent each images based on the occur rate of each of the clustered visual words; the fourth stage is perform mining to get the salient patterns and construct a graph to get the concurrent patterns; the fifth stage is perform random walk based on the graph constructed; and the last stage is to perform re-ranking. This system will used several tools which are C++, QT Creator, OpenCV and etc.

For System Development Life Cycle methodology, I have chosen Incremental Model as my methodology guideline in development of the system. All the detail of the re-ranking method will be discussing throughout this proposal.

Table of Contents

Declaration of Originality ..... **i**

Acknowledge ..... **ii**

Abstract ..... **iii**

Table of Contents ..... **iv**

List of Figures ..... **vi**

List of Tables ..... **viii**

Chapter 1: Introduction **1-7**

1.1 Project Background 1-4

1.2 Visual Re-ranking 5

1.3 Problem Statement 5-6

1.4 Objective 6

1.5 Project Scope 7

Chapter 2: Literature Review **8-14**

2.1 Type of Re-ranking Method 8-10

2.1.1 With User Intervention 11

2.1.2 Without User Intervention 12

2.1.2.1 Self re-ranking 12-13

2.1.2.2 Crowd re-ranking 13-14

Chapter 3: Methodology **15-27**

3.1 Software Development Life Cycle 15

3.1.1 System Analysis 15

3.1.2 System Design 15-17

3.2 Timeline 18

3.3 The Proposed Methods 19

3.3.1 Feature Extraction Part 20

3.3.2 Learning Part 20

3.3.3 Image Representation Part 20

3.3.4 Knowledge Mining Part 21

3.3.4.1 Generate Salient Pattern 22

3.3.4.2 Generate Concurrent Pattern 22

3.3.5 Fusion Part 23

3.3.6 Ranking Part 24

3.4 Story Board 25-27

Chapter 4 - Experiment Result and Tools 28-37

4.1 Development Tools 28

4.2 Comparisons Result 28-33

4.3 Time Taken 34-42

Chapter 5: Conclusion 43

5.1 Limitation and Future Work 43

Reference 44-45

List of Figures

Figure No.	Description	Pages
Figure 1	By given a simple query (“phone”), the initial search result display well from Google search engine.	2
Figure 2	By given a complex query (“breaking a phone”), the initial search result performed by such search engine not so well.	2
Figure 3	By given a simple query (“phone”), the initial search result display well from Bing search engine.	3
Figure 4	By given a complex query (“breaking a phone”), the initial search result display not so well from Bing search engine.	3
Figure 5	Search Engine 0 adopts 2 different methods to perform initial search result based on the same complex query given.	4
Figure 6	The hierarchy diagram briefly shows several type of re-ranking method.	9
Figure 7	The diagram briefly shows how Pseudo-Relevance Feedback performs re-ranking method.	11
Figure 8	The diagram briefly shows how self re-ranking performs re-ranking method.	13
Figure 9	The diagram briefly shows how crowd re-ranking performs re-ranking method.	14
Figure 10	System Diagram – Component used in the proposed methods.	16
Figure 11	Data Flow Diagram – the progress involved in the proposed methods.	17
Figure 11(a)	Flow Chart – process flow in component 2 – Learning part.	17
Figure 12(a)	Gantt chart – FYP1 progress.	18
Figure 12(b)	Gantt chart – FYP2 progress.	18
Figure 13	Main Interface for such prototype system.	25

Figure 14(a)	Bar chart – The average precision of the original vs. re-ranked results from top 10 images in each search engine, given a query called "japanese nuclear disaster".	33
Figure 14(b)	Bar chart – The average precision of the original vs. re-ranked results from top 10 images in each search engine, given a query called "indonesia haze".	33
Figure 15(a)	Bar Chart (with table provided) – The average processing time in creating library files, given a query called "japanese nuclear disaster".	35
Figure 15(b)	Bar Chart (with table provided) – The average processing time in loading library files, given a query called "japanese nuclear disaster".	36
Figure 15(c)	Bar Chart (with table provided) – The average processing time in learning phase (for both cases, i.e. library folder have/no exists), given a query called "japanese nuclear disaster".	37
Figure 16(a)	Bar Chart (with table provided) –The average processing time in creating library files, given a query called "indonesia haze".	38
Figure 16(b)	Bar Chart (with table provided) – The average processing time in loading library files, given a query called "indonesia haze".	39
Figure 16(c)	Bar Chart (with table provided) – The average processing time in learning phase (for both cases, i.e. library folder have/no exists), given a query called "indonesia haze".	40
Figure 17(a)	Bar Chart (with table provided) – Total average processing time, given a query called "japanese nuclear disaster".	41
Figure 17(b)	Bar Chart (with table provided) – Total average processing time, given a query called "indonesia haze".	42

List of Tables

Table No.	Title	Page
Table 1	Type of re-ranking method.	10
Table 2	The summarization of the entire progress of the re-ranking method.	19
Table 3	Function in each component of main interface of prototype system.	26-27
Table 4(a)	The background in dark color represented as positive relevant image for both original ranked result and re-ranked result, from Ask search engine, given a query called “japanese nuclear disaster”.	29
Table 4(b)	The background in dark color represented as positive relevant image for both original ranked result and re-ranked result, from Ask search engine, given a query called “indonesia haze”.	29
Table 5(a)	All images for both original ranked result and re-ranked result are represented as positive relevant images, from Bing search engine, given a query called “japanese nuclear disaster”.	30
Table 5(b)	All images for both original ranked result and re-ranked result are represented as positive relevant images, from Bing search engine, given a query called “indonesia haze”.	30
Table 6(a)	All images for both original ranked result and re-ranked result are represented as positive relevant images, from Google search engine, given a query called “japanese nuclear disaster”.	31
Table 6(b)	The background in dark color represented as positive relevant image for both original ranked result and re-ranked result, from Google search engine, given a query called “indonesia haze”.	31
Table 7(a)	The background in dark color represented as positive relevant image for both original ranked result and re-ranked result, from Yahoo search engine, given a query called “japanese nuclear disaster”.	32
Table 7(b)	All images for both original ranked result and re-ranked result are represented as positive relevant images, from Yahoo search engine, given a query called “indonesia haze”.	32

Chapter 1 - Introduction

1.1 Project Background

Most popular visual search engines (e.g. Google, Yahoo!, Live, Bing, etc.) mostly focus on text search techniques by giving a query to search for some relevant visual contents. For example, pick a random visual search engine, within this search engine, key in any object, words, event or any kind of query to search for some images. And the initial search result will display a list of images which will base on the given query. For example, given a query called "phone" to search a list of images from 2 different image search engines (Google and Bing) as shown in Figure 1 and Figure 3. Since it is a simple query, therefore both search engines will provide a list of relevant images. However, for some complicated query such as given a query called "breaking a phone" to search a list of images from both search engines as shown in Figure 2 and Figure 4. We can see that when the query becoming more and more complicated, due to the noisy and ambiguous information provided, the search result for both search engines might not perform well. In short, most image search engines will display good result for simple queries, but not so much on the complex queries which is more similar to the "natural language" queries.

As shown in Figure 2 and Figure 4, we can see that the initial search result in Figure 4 is better than the result in Figure 2. Although both search engines using the same query to perform an initial search, however, both search results are different with each other. Thus, we can conclude that different search engines will adopt different methods to perform their initial search result. Nowadays, most methods where used to perform an initial search result are based on the information that provided by a single search engine instead of observing the result from other search engines. While handle a complex query, due to lack of information in determine which images are relevant or irrelevant, on which the images are more closely related to the given query, the search result might not perform well. The images with a "star" marked on its top-left side as show in Figure 2, Figure 4, and Figure 5 are consider as the relevant images on which are assume to be arrange to the top list of the initial search result. Figure 5 show the comparison output between the information that provided by a single search engine and the information that observe from other search engines where such information are used to perform an initial search result. Through such sample, we can conclude that based on the information that observes from other search engines, such search

engine will be able to perform nice search result. Thus, we are going to investigate such method as the area of study in this project.

Figure 1: By given a simple query ("phone"), the initial search result display well from Google search engine.

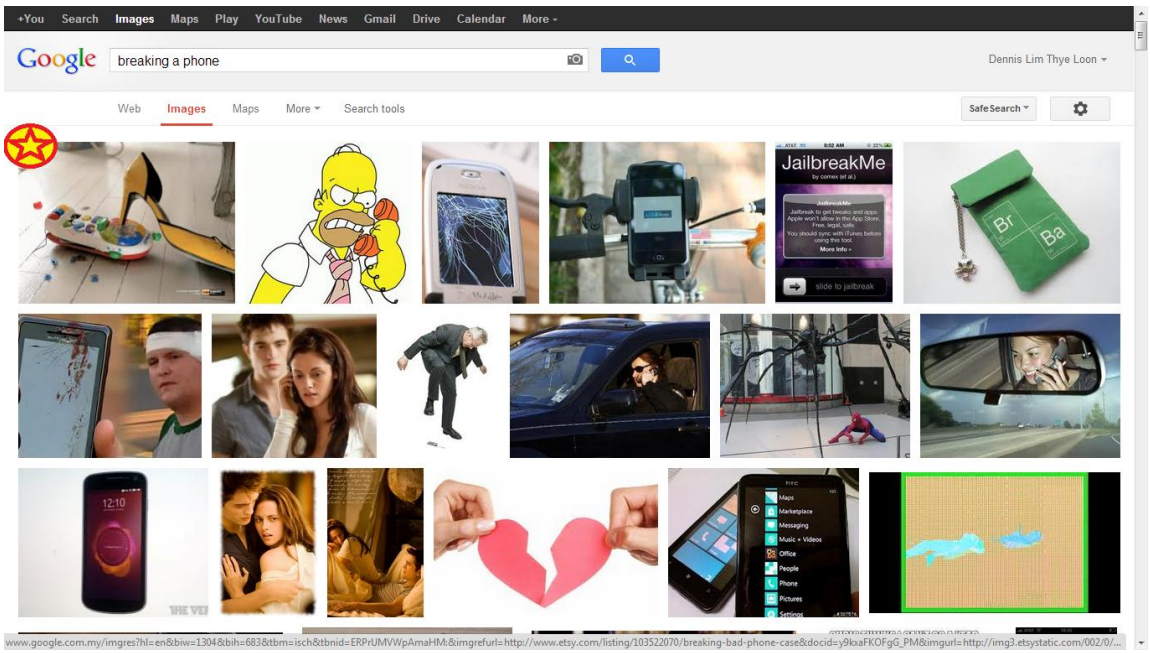


Figure 2: By given a complex query ("breaking a phone"), the initial search result display not so well from Google search engine.

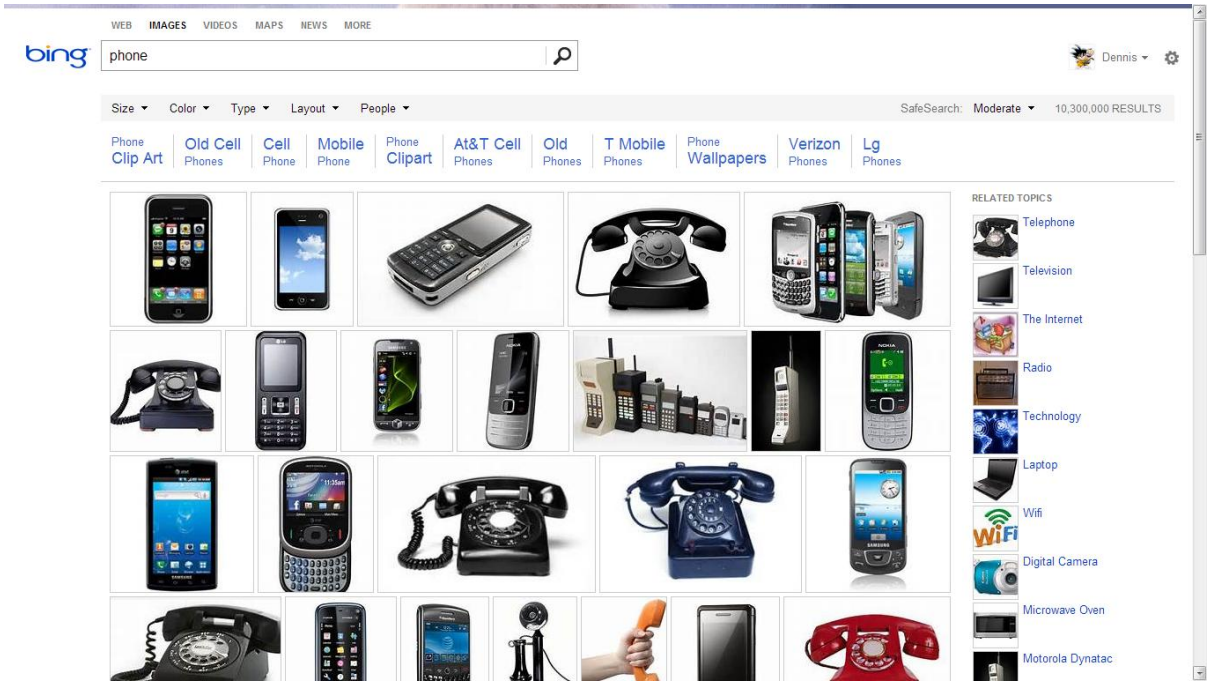


Figure 3: By given a simple query ("phone"), the initial search result display well from Bing search engine.

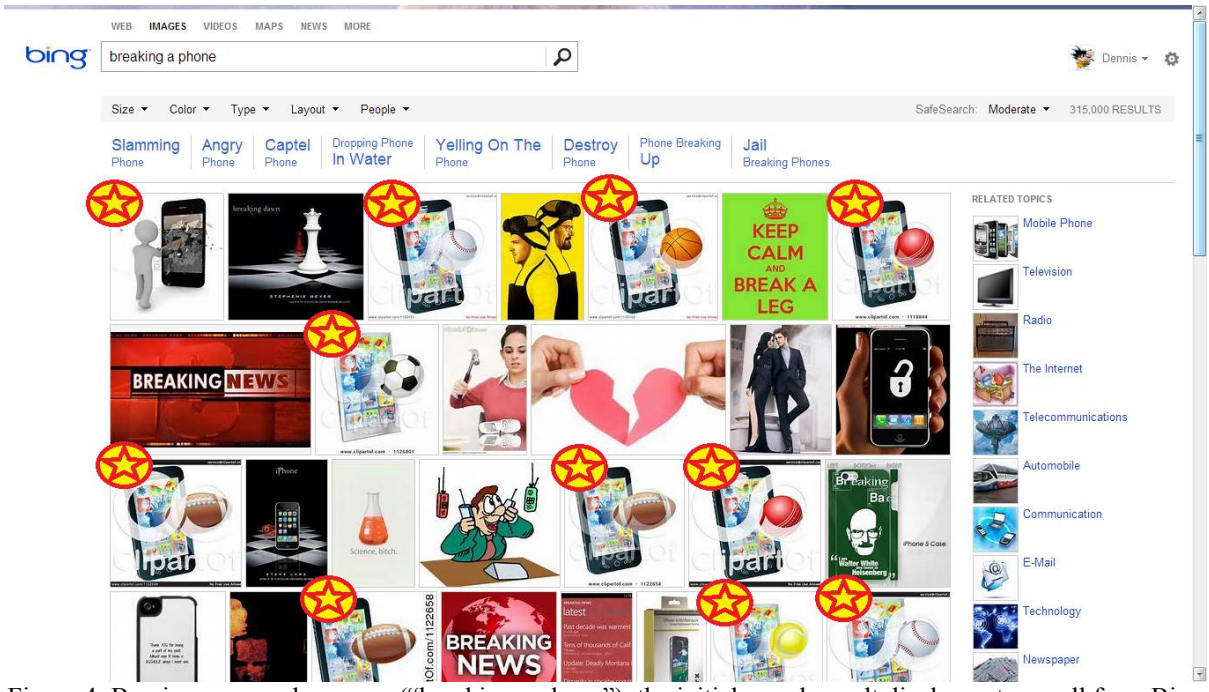
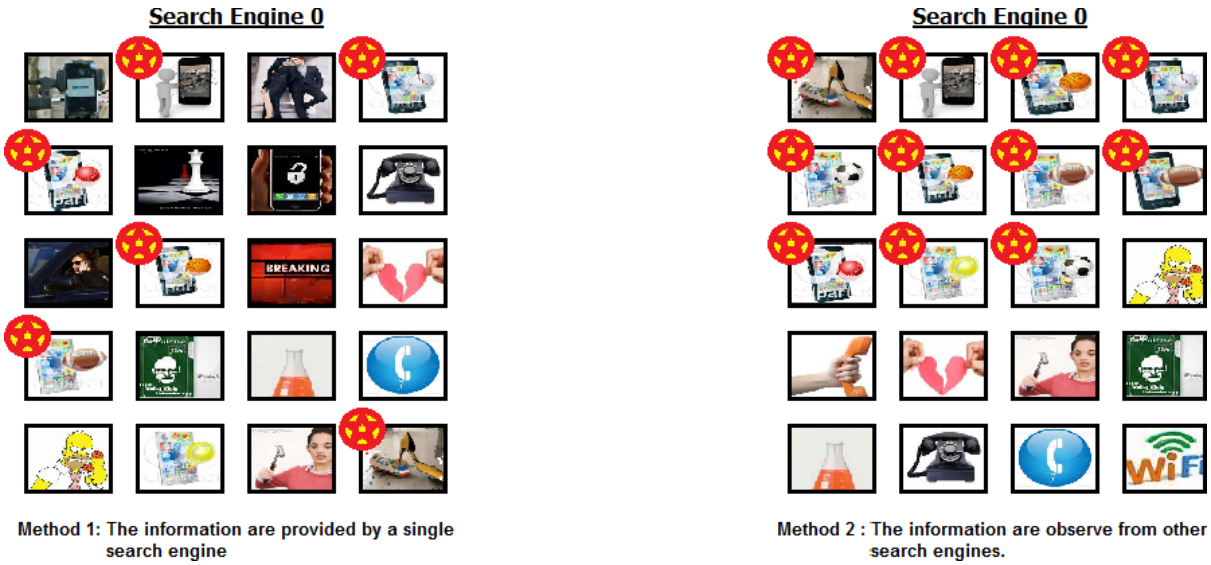


Figure 4: By given a complex query ("breaking a phone"), the initial search result display not so well from Bing search engine.



Method 1: The information are provided by a single search engine

Method 2 : The information are observe from other search engines.

Figure 5: Search Engine 0 adopts 2 different methods to perform initial search result based on the same complex query given.

1.2 Visual Re-ranking

Visual **re-ranking** (Popescuy et al. n.d.) is a technique that used to improve the initial search result by detecting the visual patterns (Owl.english.purdue.edu 2010) within the images, comparing and utilizing each of them to rearrange and reorder the search result, in order to pull more positive images to the first top list of the search result. Nowadays, most users will provide a text-based query while searching some relevant images. However, users may provide different queries, on which will cause such queries becoming more and more complicated and will difficult for the search engines to perform a better search result. In order to resolve such issues, re-ranking method is introduced. In short, improving the initial search result using re-ranking method is becoming popular nowadays. Most multimedia retrieval algorithms adopt such re-ranking method to perform visual search result by detecting the recurrent visual patterns within the top images in the initial search result. There exists several type of re-ranking method which being proposed to improve the initial search result. Such method can be categorize into two major groups which are re-ranking methods with user intervention and the re-ranking methods without user intervention.

1.3 Problem Statement

Most image search engines will display good result for simple queries, however, when such queries becoming more and more complicated, due to noisy and ambiguous textual info provided, the initial search result might not display well. Nowadays, most re-ranking method has been proposed to solve such issues. However, most of them are relied on a single search engine, due to lack of information in detecting relevant visual patterns, the initial search result might not look so well. Thus, in order to solve such problem, Crowd re-ranking algorithm (Ting Yao et al. n.d.) is introduced. Since different search engines may perform different search results, by given a query, those search results will provide some relevant images, normally every relevant images are always look similar with each other while the less might not look same. Thus, Crowd re-ranking algorithm will base on such cues to perform re-ranking. However, since such algorithm still not much been done in such area and due to the code for such algorithm did not shared in any places. Therefore, it is difficult to perform benchmarking. Since we need such code to perform benchmarking, therefore, in this project, we are going to develop a prototype system which will able to perform Crowd re-

ranking algorithm as describe in the research paper. So that, other people can use such prototype to perform benchmarking in future by defined new algorithm to further improve such algorithm. And at the same time, such prototype system also can serve as platform for evaluation, to view the strength and limitation of such algorithm method.

1.4 Objective

The main objective in doing this project is to develop a Crowd re-ranking prototype system which will able to do the following task:

1. **Able perform re-ranking automatically via multiple search engines.**
   Distinctive search engines might have distinct methods for re-ranking their search results. By observing and comparing each and other relevant images collected from other search results of multiple search engines, Crowd re-ranking algorithm will have enough cues to perform re-ranking automatically without any user intervention.

2. **Able to perform visual mining via multiple search engines.**
   Based on the visual keywords (Ijcsi.org 2013) provided, we can use this features to re-ranked the prototype system. Besides this, it is able also to performing visual learning (Inspiration.com n.d.) from multiple search engines. By utilise the information provided by multiple search engines, such system will also able to do the multiple patterns.

3. **The developed system will be able to serve as foundation for future research in Crowd re-ranking.**
   In the end, this prototype system can serve as foundation for Crowd re-ranking. Besides this, such prototype can also serve as platform to identify the strength and weakness of Crowd re-ranking algorithm. In future, people able to implement such algorithm as benchmarking and define new algorithm for further improvement.

1.5 Project Scope

In order to develop a Crowd re-ranking prototype system, we need to investigate 2 types of visual patterns which are Salient pattern and Concurrent pattern. Salient pattern represent how much time such visual word (En.wikipedia.org 2012) occur within an image while Concurrent pattern represent the relation between each visual words.

In this project, we are going to develop a prototype system that will able to implement Crowd re-ranking algorithm as describe in the research paper. And also, we are going to develop such algorithm in offline environment since the purpose of doing this project is to verify the proof of concept of such algorithm.

We will create an image dataset by storing a list of images that collected from multiple image search engines. Through such dataset, we will cluster each of the visual patterns into several groups of visual words. By adopting PageRank technique (Baluja & Jing n.d.), we will construct a graph where each visual word represent as a node and the edges between each nodes represent as co-occurring. Through such progress, we will able to develop a prototype system which will able to implement Crowd re-ranking algorithm.

Chapter 2 - Literature Review

2.1 Type of Re-ranking Method

There exist several types of re-ranking method that have been proposed previously. Such method can be categorize into two major groups which are re-ranking methods with user intervention and the re-ranking methods without user intervention. The hierarchy diagram as shown in Figure 6 briefly shows several type of re-ranking method, and Table 1 summarizes several type of re-ranking method as mention in Figure 6.

The re-ranking methods with user intervention are rely on the feedback provided by the user. Such re-ranking methods can be categorized to several types of re-ranking method. Such as Pseudo-Relevance Feedback (PRF) (Nlp.stanford.edu 2001) by asking the user to select the relevant images from the initial search results. Based on the feedback provided by the user to further improve the search result; and etc.

While, the re-ranking methods without user intervention do not rely on the feedback provided by the user, thus, it will perform re-ranking automatically. Under such type of method, there exist several types of re-ranking methods. Such as Self re-ranking (Ting Yao et al. n.d.) which only focus on solely detect the recurrent patterns from several images in the initial search results; and etc. Under such ranking algorithm, can be categorize to several type of re-ranking algorithm. Such as Query-based re-ranking (Ting Yao et al. n.d.) which arrange relevant and irrelevant images to top and bottom search list respectively based on self assumption.; Information Bottleneck (IB) principle (Winston et al. 2006) which find the optimal clusters with the highest mutual information (Scholarpedia.org 1949); Context re-ranking (Winston et al. n.d.) which utilize the contextual patterns and perform random walk along the context graph; and etc.

However, most re-ranking algorithm are relied on a single search engine, thus, the initial search result may not perform well due to lack of information provided. In order to solve such problem, there exists another type of re-ranking method which have been describe in the research paper, called Crowd re-ranking algorithm. Crowd re-ranking is one of the re-ranking method by observing and utilising each relevant visual patterns collected from other search results of multiple search engines and based on such patterns to reorder the images within the initial search result. Since different search engines may perform different search results, each search result will display several relevant images based on the given query.

Normally, most relevant images will appear recurrently, while other images might not look same. Thus, Crowd re-ranking will base on such knowledge to perform ranking by mining relevant visual patterns from each of the collected images.

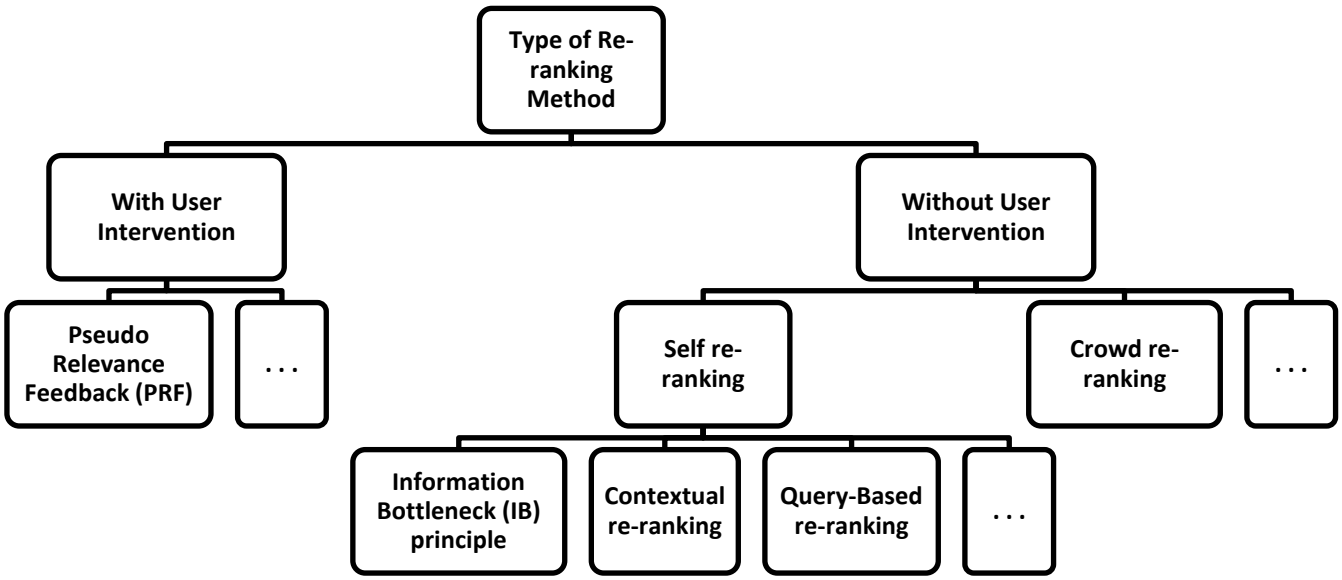


Figure 6: The hierarchy diagram briefly shows several type of re-ranking method.

<table>
<tr><th colspan="2">Type of Re-ranking Algorithm</th><th>Description</th><th>Strengths</th><th>Limitations</th></tr>
<tr><td>With User Intervention</td><td>Pseudo-Relevance Feedback (PRF) (Nlp.stanford.edu 2001)</td><td>Asking the user to select the relevant images from the initial search results. Based on the feedback provided by the user to further improve the search result</td><td>More relevant images will be show based on user feedback.</td><td>User does not have patient to provide feedback.</td></tr>
<tr><td rowspan="3">Without User Intervention</td><td rowspan="2">Self re-ranking (Ting Yao et al. n.d.)</td><td>Focus on mining relevant visual patterns solely within initial search result.</td><td>More relevant images will be pulled to the top list of search result.</td><td>Only base on the initial search result perform by a single search engines.</td></tr>
<tr><td colspan="3">Type of Self re-ranking
<table><tr><th colspan="3">Description</th></tr><tr><th>Query-Based re-ranking (Ting Yao et al. n.d.)</th><th>Information Bottleneck (IB) principle (Winston et al. 2006)</th><th>Context re- ranking (Winston et al. n.d.)</th></tr><tr><td>Arrange relevant and irrelevant images to top and bottom search list respectively based on self assumption.</td><td>Find the optimal clusters with the highest mutual information.</td><td>Utilize the contextual patterns and perform random walk along the context graph</td></tr></table></td></tr>
<tr><td>Crowd re-ranking (Ting Yao et al. n.d.)</td><td>Mining all relevant visual patterns collected from multiple image search engines via Internet.</td><td>Easy to detect relevant images.</td><td>Performance may be slow than other re-ranking method.</td></tr>
</table>

Table 1: Type of re-ranking method.

2.1.1 With User Intervention

Re-ranking with user intervention is one of the types of re-ranking method. Such method performing an initial search result based on the feedback given by the user. On which such re-ranking method will involve the user in retrieval process while perform re-ranking. Such re-ranking methods can be categorized into several types of re-ranking algorithm. Pseudo-Relevance Feedback (PRF) (Nlp.stanford.edu 2001), is one of the re-ranking algorithm that under such category.

PRF algorithm is a method that perform ranking by asking the user to select the relevant images from the initial search results. Based on the feedback provided by the user, such algorithm will perform the re-ranking method by detecting the visual pattern within the visual information that provided by user, utilising each of them, and pull the most relevant images to the first top list of the initial search result based on the users' feedback. The purpose of such re-ranking is based on the feedback given by user to further improve the visual search result.

Since such algorithm could perform good results based on users' feedback, however, when users are reluctant to provide any relevant images as a feedback to such algorithm, then such algorithm will difficult to perform a nice image search result. Figure 7 briefly shows how PRF algorithm performs re-ranking by asking the user to provide relevant feedback. The images with a "star" marked on its top-left side are considered as the relevant feedback provided by the user.

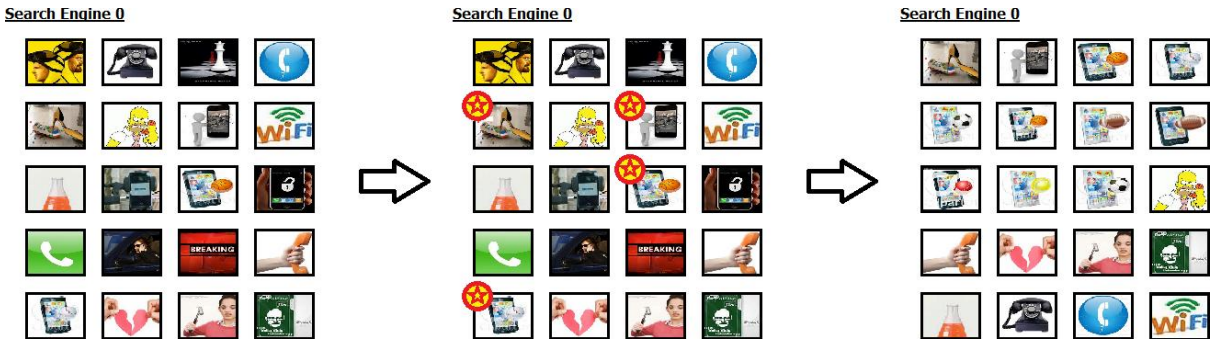


Figure 7:The diagram briefly shows how Pseudo-Relevance Feedback performs re-ranking method.

2.1.2 Without User Intervention

Re-ranking without user intervention is one of the types of re-ranking method. The different between such type of re-ranking method and another type of method is such method will not relied on the user feedback while performing an initial search result based on the external knowledge provided. On which such re-ranking method will based on its knowledge provided to perform re-ranking automatically without any feedback from the user. Such re-ranking methods can be categorized into several types of re-ranking algorithm. Such as Self re-ranking (Ting Yao et al. n.d.) where perform ranking within a single search engine, Crowd re-ranking (Ting Yao et al. n.d.) where perform ranking via multiple search engines and other type of re-ranking algorithm.

2.1.2.1 Self re-ranking

Self re-ranking is one of the re-ranking algorithm that under re-ranking method without user intervention, on which perform re-ranking by focusing on mining recurrent visual patterns solely within an initial search result without any user intervention. Based on the recurrent visual pattern mined, such algorithm will have enough cues on perform re-ranking by pull more recurrent images to the top list of the initial search result as shown in Figure 8.

Since such algorithm perform ranking method is only by mining each and other recurrent patterns within an initial search result, thus, the result of such method might not perform well. Besides this, since such algorithm is relied on a single search engine, while handle a complex query, due to noisy and ambiguous textual information provided, and based on its ranking method (which perform ranking based on recurrent patterns instead of relevant patterns), the result of such method might not look so well. In short, using Self re-ranking to perform an initial search result might not a good idea. Thus, under such algorithm, there exists several type of re-ranking method in order to overcome such issues, such as Query-based re-ranking (Ting Yao et al. n.d.) which arrange relevant and irrelevant images to top and bottom list respectively based on self assumption, Information Bottleneck (IB) principle (Winston et al. 2006) which find the optimal clusters with the highest mutual information (Scholarpedia.org 1949), Context re-ranking (Winston et al. n.d.) which utilize the contextual patterns and perform random walk along the context graph, and etc.

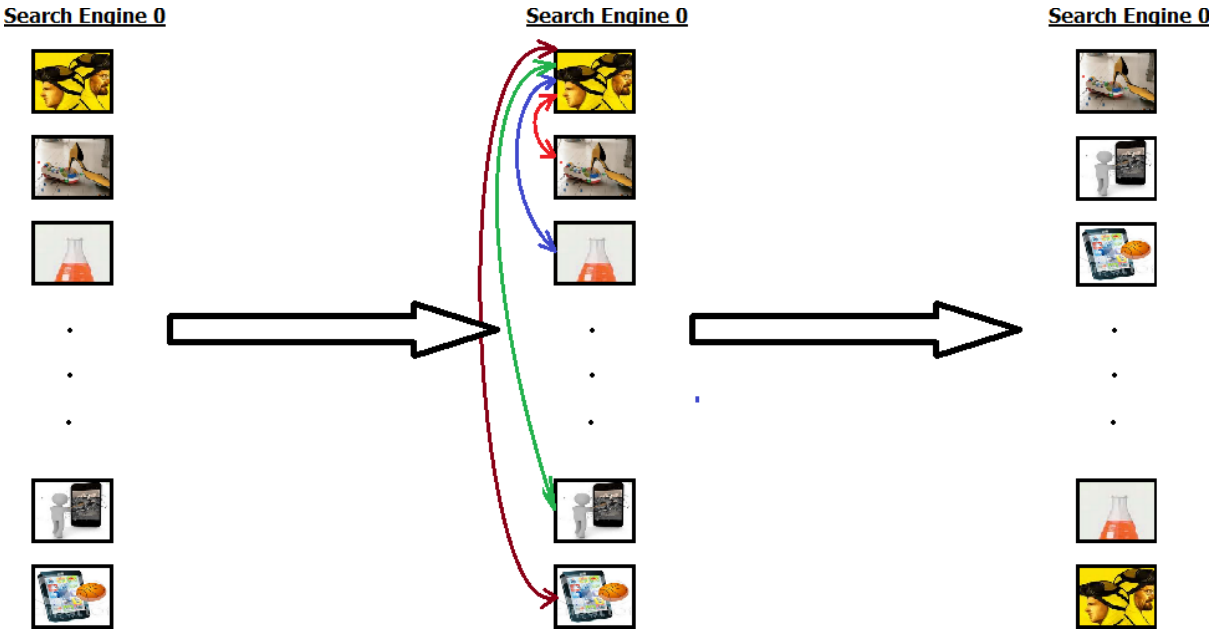


Figure 8: The diagram briefly shows how self re-ranking performs re-ranking method.

2.1.2.2 Crowd re-ranking

Crowd re-ranking algorithm is one of the re-ranking methods that performed without user intervention as same with Self re-ranking algorithm. However, the only different between Self re-ranking algorithm is, such algorithm perform re-ranking via multiple search engines, instead of a single search engine compared with Self re-ranking algorithm.

Given a query, most search results will perform several relevant images within each search engines. Normally, most relevant images will appear recurrently, while other images might not look similar with each other. By mining all relevant visual patterns that collected from multiple search engines, due to the information have widely given from other search results of multiple search engines. Crowd re-ranking will have enough knowledge to perform ranking by mining each of the relevant visual patterns and based on such patterns to reorder the images within the initial search result as shown in Figure 9. However, due to mining each of the images collected from other search results, the performance of such algorithm maybe very slow compared to other type of re-ranking method.

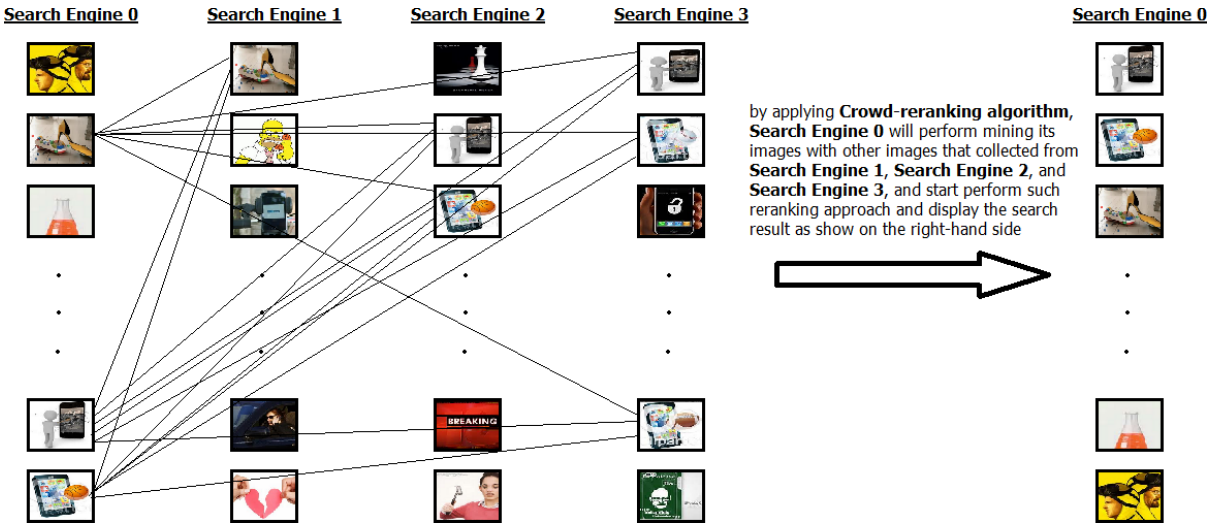


Figure 9: The diagram briefly shows how crowd re-ranking performs re-ranking method.

Chapter 3 - Methodology

3.1 Software Development Life Cycle

In this project development process, Incremental model is implemented. The Incremental model consists of 4 important phases, which are analysis phase, design phase, implementation phase, and maintenance phase. It is actually the combination of the elements of Waterfall model with iterative feature of Prototyping model. The reason of this Incremental model is choose in this project development process is because in this project, its able to develop prototype system quickly and also able to detect and manage risk earlier during the development life cycle.

3.1.1 System Analysis

In the analysis phase, we studied and evaluated various image re-ranking methods that were proposed in the previous research papers. The result of those reviews have been given in chapter 2, Literature Review part, in which describe each type of re-ranking algorithms with their strength and weakness. After that, we analyzed the user requirement by defining the project background, problem statement, objective and also the project scope in which have mentioned in chapter 1, Introduction part, to develop such offline prototyping system.

3.1.2 System Design

The diagram as shown in Figure 10 shows a high level overview of the entire progress that involved in the proposed methods. Based on the diagram, there consists six parts of components in which used to implement the proposed method. The process flow for each part of components are as shown in Figure 11, where Figure 11(a) briefly shows the process flow of the 2nd part of the component (Learning part), in which shows the progress in perform training. Those total six parts of components will be detailed describe in later chapter 3.3, which is the Proposed Methods part.

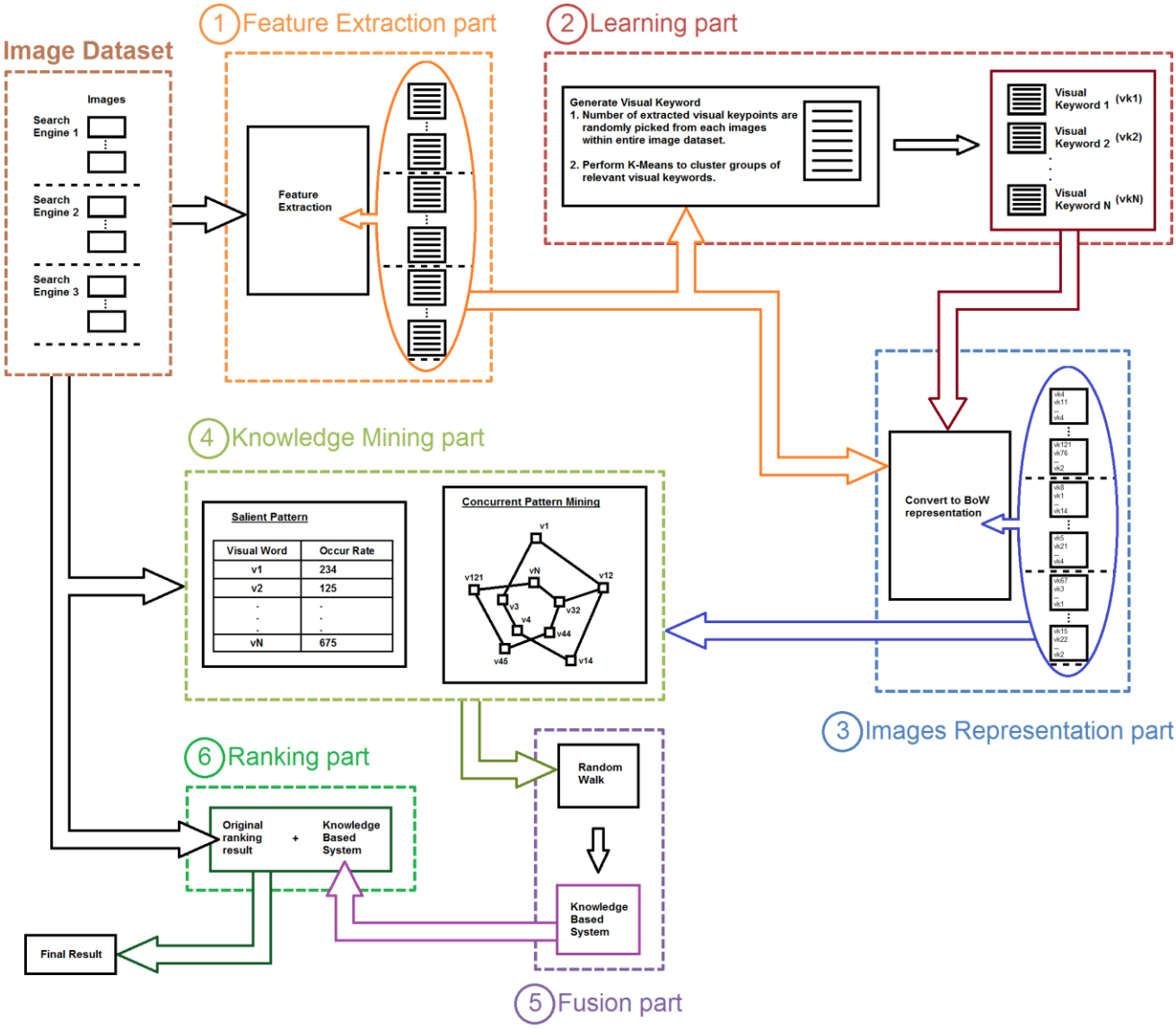


Figure 10: System Diagram – Component used in the proposed methods.

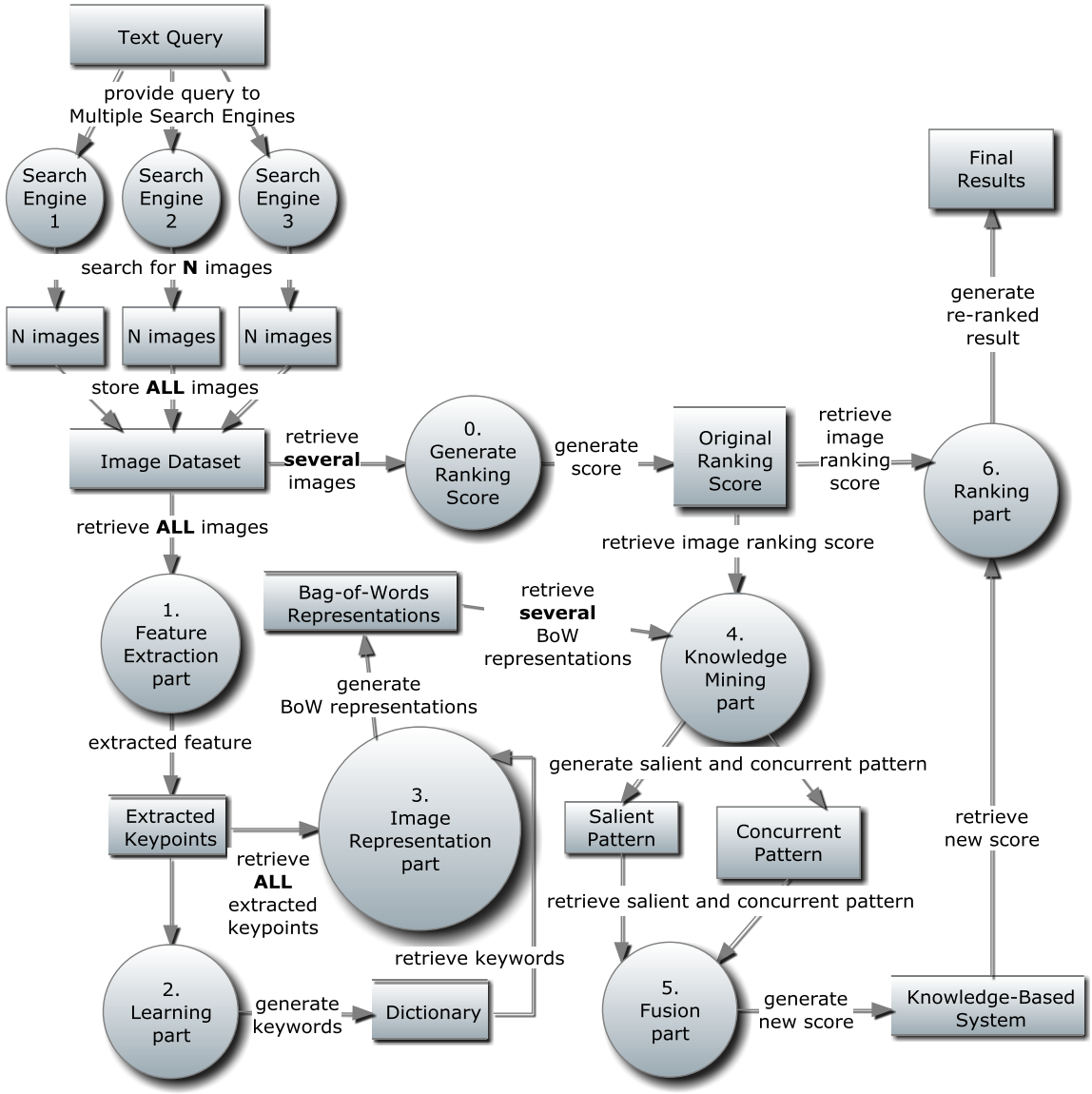


Figure 11: Data Flow Diagram – the progress involved in the proposed methods.

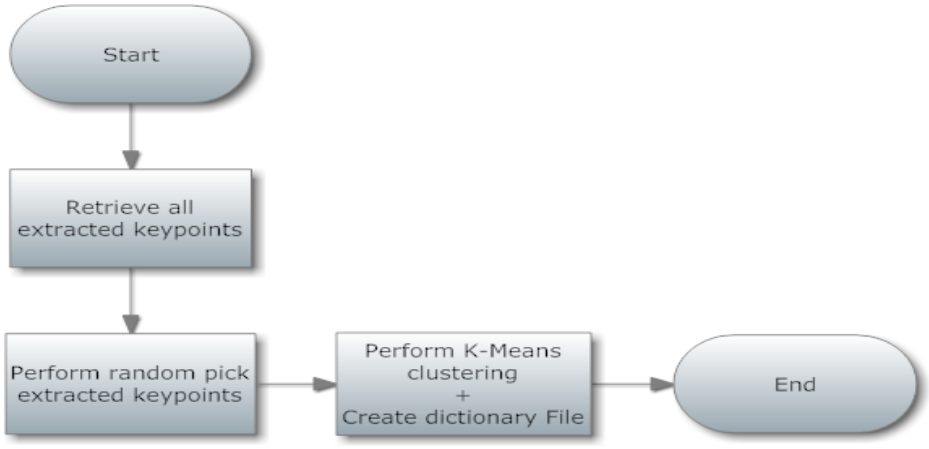


Figure 11(a): Flow Chart – process flow in component 2 – Learning part.

3.2 Timeline

This timeline is developed to summarize the schedule of the entire progress for the whole Final Year Project (FYP). Figure 12(a) shows the time schedule in FYP1, and Figure 12(b) shows the time schedule for FYP2. There have several tasks for this entire FYP2, and each progress of each task are summarize as shown in Figure 12(a) and 12(b).

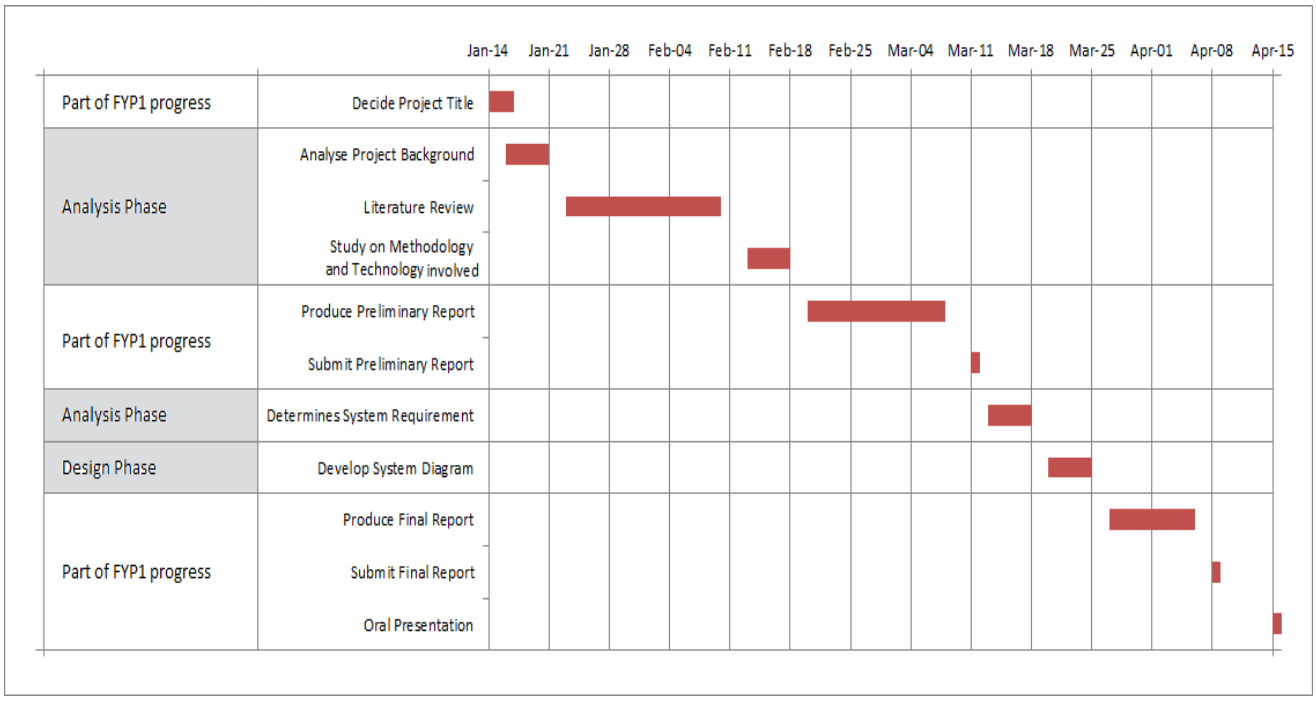


Figure 12(a): Gantt chart – FYP1 progress.

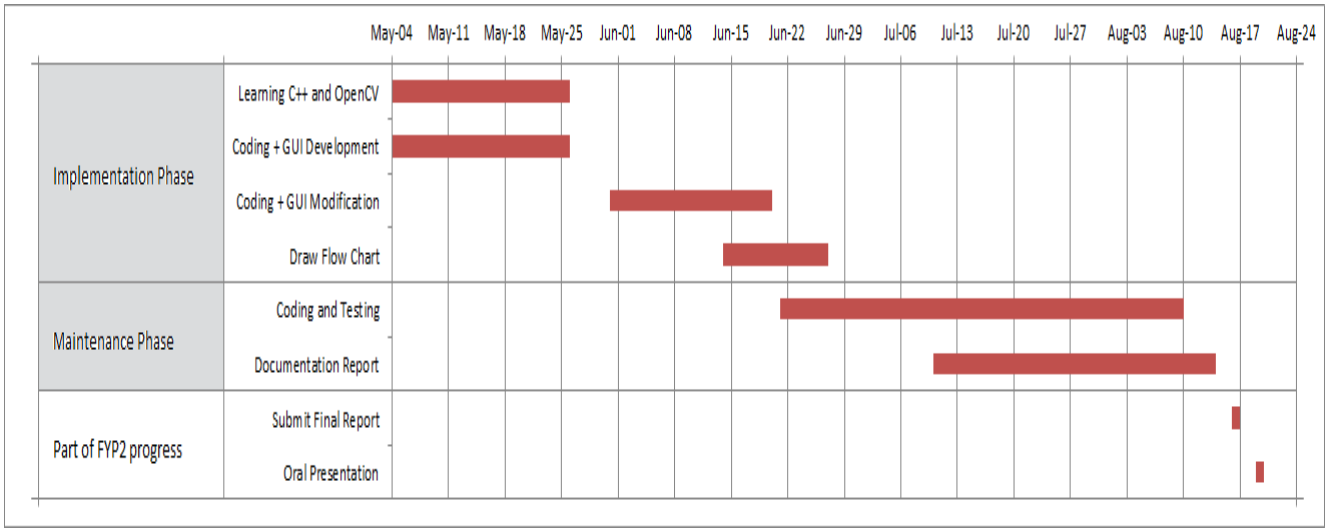


Figure 12(b): Gantt chart – FYP2 progress.

3.3 The Proposed Methods

In this subchapter, the proposed Crowd re-ranking technique is described. The basic idea of the proposed method is to divide the re-ranking process into six parts, in which as shown in Figure 10. Thus, this subchapter will describe the process flow of these six parts of components in detailed. Table 2 briefly summarize the process flow of such propose methods.

Part		Process Description
ID	Name	
1	Feature Extraction	Detect and extract each and every features or keypoints in all images collected from image dataset or multiple search engines that based on the same query given.
2	Learning	Based on the extracted keypoints, randomly select several keypoints from top sample images in each search engines to define a set of relevant visual keywords or Bag-of-Words (BoW) (Mubarak Shah 2012) by perform K-Means cluster.
3	Image Representation	Based on the occur rate of each keywords (perform in Learning part) and the keypoints (extracted from each images in Feature Extraction part), each images will able to represent as a histogram or BoW representation.
4	Knowledge Mining	Based on each BoW representation, compute the total score for each keywords to get the salient patterns. And construct a graph using PageRank-like technique to compute the occur rate of each two keywords to get the concurrent patterns.
5	Fusion	Based on the constructed graph, perform random walk by finding each and every keyword with the optimal relation scores and generate a knowledge-based system to store those optimal score for each keyword.
6	Ranking	Based on the original ranking score (score from each images) and the knowledge-based system (score from each keywords, computed in Fusion part), such prototype system will able to perform re-ranking by based on the highest score, in order to pull more relevant images to the top list of the ranking result.

Table 2: The summarization of the entire progress of the re-ranking method.

3.3.1 Feature Extraction Part

Before proceed to this part, since we do not have a server to store all the image results collected from multiple search engines based on the query provided, thus, we will create an image dataset to represent such server by providing a text query for each search engine and storing around N images collected from each search engines as shown in part of Figure 10 (in the form of system diagram) or Figure 11 (in the form of data flow diagram). After that, in order to define the information contained in those images, we need to define the keypoints (features) in each image by using SURF Feature Detector to detect each and every keypoints from each image. Based on those detected keypoints, we will able to represent those keypoints we found to compare them with other keypoints in each image by extracting them using SURF Descriptor Extractor.

3.3.2 Learning Part

In this part, we will first randomly select several amounts of extracted keypoints from top sample images in each search engine respectively and store those selected keypoints into local features of image patches. Through such patches, we will able to cluster those keypoints, by using K-means cluster to come out with the words to have a visual vocabulary, since visual keywords didn't have vocabulary to represent an image. After finish clustering, we will have around several representative visual keywords. The flow chart as shown in Figure 11(a) shows the entire progress in which perform such function by random select extracted keypoints, and perform K-Means cluster to generate the dictionary file that contain each clustered keywords (visual keywords).

3.3.3 Image Representation Part

In this part, we are going to convert all images (collected from Image Dataset, as shown in part of Figure 10 or Figure 11) into representative document by converting each image from each search engines into histogram or Bag-of-Words (BoW) representation where each keywords (as store in dictionary file, in which generated in Learning part) are represented as node and the frequency represent as the occur rate in which indicate current keywords have occur in the given image.

3.3.4 Knowledge Mining Part

In order to find the relation between each keyword (as generated in Learning part) with each and every images, and also the relation between each both keywords that occur in each images, we need to generate score for each keywords in order to get the salient pattern and also construct an affinity matrix using PageRank-like technique to generate the concurrent pattern. Thus, in this knowledge mining part, there consists of 2 main stages which known as generating salient pattern stage and generating concurrent pattern stage. Since most search results does not contain ranking score for each ranked images, thus, in this part, in order to generate each score for each keywords, we will firstly generate ranking score ($\overline{r}$) for each images by using sigmoid function, and each image (from each search engines) will share the same ranking score base on their current ranking position.

The sigmoid function will work like this: $\overline{r} = \frac{1}{1+ e^{-a\,(x-c)}}$ where:

- $\overline{r}$ = ranking score for x
- a = the slope at c
- c = the inflection point
- x = current ranking position of an image, starting from 1 to M
- M = number of images needed to use for perform re-ranking.

After that, in order to make the ranking result more accurately, we will perform ranking based on top M images (in each search engines), since most images in the first few list of the search result mostly are consider as positive relevant images, thus, we will take top M number of Bag-of-Words (BoW) representation (generated from Image Representation part) from each search engines to generate salient pattern (q) and concurrent pattern (P). Figure 11 shows part of the progress in which perform such function by generating salient pattern and concurrent pattern.

3.3.4.1 Generate Salient Pattern

In this stage, we will find the relation between each keywords (from dictionary file) with each images (top sample images from each search engines) by generate score for each keywords based on the appearance of those keywords that occur in each images. We will generate salient pattern (q) by providing score for those keywords based on the ranking score ($\overline{r}$) by accumulating the score for each keywords if current keyword occurred in current image and based on the ranking position of such image.

3.3.4.2 Generate Concurrent Pattern

In this stage, we will find the relation between each keyword ***i*** and ***j*** that occur in each and every sample images (top images from each search engines). We will firstly construct an affinity matrix by using PageRank-like technique to generate the concurrent pattern by calculating the appearance of those two keywords in which have occurred together in the given image from each search engines respectively. Based on the occurrence of such two keywords that occur together in each image, the concurrent pattern (P) through this knowledge mining part is then generated.

3.3.5 Fusion Part

Before proceed to this part, in order to define the keywords that have pointed to / by each other keywords, we will first normalize the generated concurrent pattern (P) by each column to construct a correlation graph (Pc), in order to find out the relationship between each 2 keywords, where each columns represented as current keywords and each rows (in each columns) represented as the neighbor keyword(s) to be pointed by such current keyword. Based on the constructed graph (Pc), we will able to perform random walk by performing some equation in order to adjust those score (from each keywords) and find each of those keywords with their optimal scores.

Such equation will work like this: $q_{new} = Part1 + Part2$ where:

i. q_{new} = knowledge-based system, in which store the optimal score for each keywords

ii. $Part1 = \varepsilon q$

= adjustment for q (score from current keyword)

iii. $Part2 = (1 - \varepsilon)(Pc \times q_{tmp})$

= adjustment for $(Pc \times q)$, score from each keyword (neighbor with current keyword)

Through this equation, we will able to find the optimal score for each keyword and store those scores into knowledge-based system (q_{new}).

3.3.6 Ranking Part

In order to pull more relevant images to the first top list in each search engines, we need to generate a new score for those images and based on the highest score to rank those images. Thus, based on the original ranking score ($\overline{r}$) for each images (top M number of images from each search engines) generated in Knowledge-Mining part and knowledge-based system (q_{new}) computed in Fusion part, we will able to generate a new ranking score ($\overline{r}_{new}$) for those images based on their current ranking position.

Such equation for generate a new ranking score for each image will look like this:

$$\overline{r}_{new} = \frac{1}{2}\,(2\overline{r} + \lambda S)$$

Where:

- $\overline{r}_{new}$ = new ranking score for each top M images (in each search engines)
- $\overline{r}$ = original ranking score for each M images
- λ = a value to control the result compute from S
- S = *Part1* + *Part2* where:
  i. *Part1* = total score (q_{new}) from each keywords that occurred in current image
  ii. *Part2* = $\frac{1}{2}\left(Pc_{ij} + Pc_{ji}\right)$ for each keywords ***i*** and ***j*** that occurred in current image

Through this equation, we will able to perform re-ranking based on the highest ranking score stored in ($\overline{r}_{new}$) for each M images in each search engines.

3.4 Story Board

Figure 13 shown the main interface layout of such prototype system and Table 3 describe each of its components with their respective function. When user choose ONE of the text query from the Drop-Down-List (ID: D1) and click the Search-Button (ID: B1). The system will display the initial image results from each FOUR search engines in the Image-List (ID: L1, L2, L3, L4) respectively. And at the same time, such system will display its progress in the Message-Box (ID: M1) for user to read. After done display list of initial search results, such system will allow user to click the Learning-Button (ID: B2) to perform training. After done perform training, such system provide several feature, such as: allow user to double-click any ONE image from the Image-List (ID: L1, L2, L3, L4) to view the selected image and with its keypoints in the pop-out-windows; Clear-Button (ID: B4) is enabled to clear those pop-out-windows once user have randomly select an image to view its keypoints; and, allow user to click the Loading-Button (ID: B3) to perform ranking and display the ranked image for each FOUR search engines in the Image-List (ID: L5, L6, L7, L8). Page-Buttons (ID: P1 – P5) from each of the image lists (ID: L1 – L8) represent as page buttons for user to click and view for the next and previous page of images. Based on such Figure also, we can see that the function perform by Learning-Button (ID: B2) and Loading-Button (ID: B3) are exactly based on the proposed methods in which have mentioned in chapter 3.3, the Proposed Methods part.

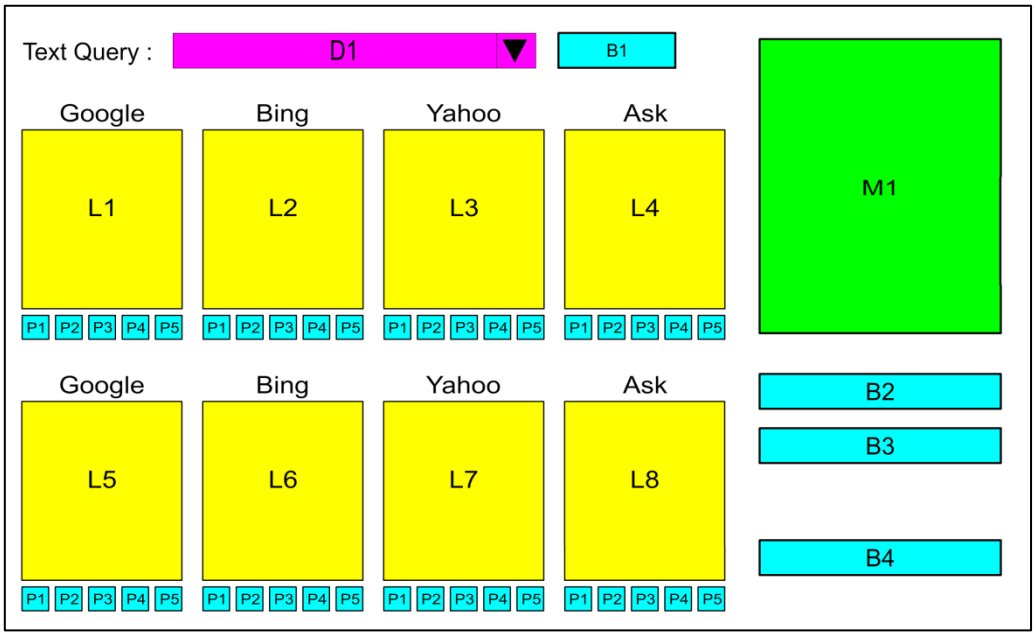


Figure 13: Main Interface for such prototype system.

ID	Name	Object	Function Description
D1	-	Drop down list	- Contain TWO text query to choose : i. japanese nuclear disaster i. indonesia haze
B1	Search	Button	- Used to search for images and display them in the Image-List (ID: L1, L2, L3, L4). When user has chosen one of the text queries from the Drop-Down-List (ID: D1).
B2	Learning Model	Button	- Enabled when user select one of the text query from the Drop-Down-List (ID: D1) and clicked the Search-Button (ID: B1). - Used to perform training purpose and will disabled once done perform training, to prevent user re-clicks it and re-perform training again. - Allow user to double-click one of the image in the Image-List (ID: L1, L2, L3, L4) and will pop out two windows to display the selected image and with its keypoints.
B3	Loading Model	Button	- Enabled when Learning-Button (ID: B2) is clicked. - Used to perform ranking purpose and will disabled once done perform ranking. - Display ranked images for each FOUR search engines in the Image-List (ID: L5, L6, L7, L8).
B4	Clear	Button	- Enabled when pop-out-window showed. - Disable again once activate its function. - Used for clear all the pop-out-window.
M1	Message-Box	Plain Text	- Display all the progress of such prototype system for user to read while waiting.
L1, L2, L3, L4	-	List widget	- Display initial search result from Google, Bing, Yahoo, Ask search engines. Once Search-Button (ID: B1) has clicked. - Allow user to randomly pick an image to view its keypoints once Learning-Button (ID: B2) has clicked.
L5, L6, L7,	-	List widget	- Display ranked result from Google, Bing, Yahoo, Ask search engines respectively. Once Loading-Button (ID: B3) has clicked.

L8			
P1	Previous	Button	- Used to go back to the previous page number. - Disable when current page is at the beginning page.
P2, P3, P4	Page #	Button	- Used to continue display other images based on the current page number. - Disable when user is in the current page. - Enabled again once user is in other page.
P5	Next	Button	- Used to go forward to the next page number - Disable when current page is at the last page.

Table 3: Function in each component of main interface of prototype system.

Chapter 4 - Experiment Result and Discussion

4.1 Development Tools

In this project, QT Creator is chosen as the development tool for develop an offline prototype system that able to implement Crowd re-ranking algorithm. The programming language that is going to used is C++ programming and OpenCV is serve as library of image processing function. The reason of chosing QT Creator as development tool is because of such tool provides drag-and-drop features for creating user interface and it's easy to use. The reason C++ programming is chosen is because it is portable in which can run in any platform, able to perform multi-threaded, and easy to write, compile and detect bug than other programming languages. And OpenCV is chosen is because it is easy to use for image processing purpose.

4.2 Comparisons Result

In this sub-chapter, we will select top 100 images from each 4 search engines (Google, Bing, Yahoo and Ask) to perform images re-ranking. By providing two queries, where the 1st query called "japanese nuclear disaster" and the 2nd query called "indonesia haze", and we will then select top 10 images from the original ranked result and the re-ranked result via each search engine to perform the image ranking result comparisons.

The 4 tables as shown in Table 4(a), Table 5(a), Table 6(a) and Table 7(a) shows the comparison between the original ranked results and the re-ranked results based on the 1st query given, and the other 4 tables as shown in Table 4(b), Table 5(b), Table 6(b) and Table 7(b) shows the same comparisons based on the 2nd query given, where each table represented as each search engine. The dark color background (for each image) represented as relevant image, in which based on the query given, and the dark color background (for each ranking number) represented as both original ranked result and re-ranked result are relevant images.

The 2 bar-charts as shown in Figure 14(a) and Figure 14(b) summarizes the result of the comparisons, where each figure represented as the provided query, each search engines in each figure represented as nodes, and the 2 legends from each node represented as the total number of the relevant image from each original ranked result and from each re-ranked result respectively.

Ask (japanese nuclear disaster)					
Rank	**Original Result**	**Ranked Result**	**Rank**	**Original Result**	**Ranked Result**
1			**6**		
2			**7**		
3			**8**		
4			**9**		
5			**10**		

Table 4(a): The background in dark color represented as positive relevant image for both original ranked result and re-ranked result, from Ask search engine, given a query called “japanese nuclear disaster”.

Ask (indonesia haze)					
Rank	**Original Result**	**Ranked Result**	**Rank**	**Original Result**	**Ranked Result**
1			**6**		
2			**7**		
3			**8**		
4			**9**		
5			**10**		

Table 4(b): The background in dark color represented as positive relevant image for both original ranked result and re-ranked result, from Ask search engine, given a query called “indonesia haze”.

Bing (japanese nuclear disaster)					
Rank	**Original Result**	**Ranked Result**	**Rank**	**Original Result**	**Ranked Result**
1			**6**		
2			**7**		
3			**8**		
4			**9**		
5			**10**		

Table 5(a): All images for both original ranked result and re-ranked result are represented as positive relevant images, from Bing search engine, given a query called "japanese nuclear disaster".

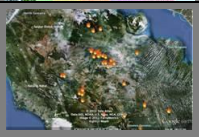
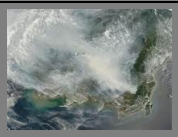
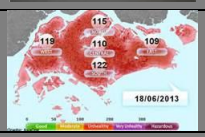
Bing (indonesia haze)					
Rank	**Original Result**	**Ranked Result**	**Rank**	**Original Result**	**Ranked Result**
1			**6**		
2			**7**		
3			**8**		
4			**9**		
5			**10**		

Table 5(b): All images for both original ranked result and re-ranked result are represented as positive relevant images, from Bing search engine, given a query called "indonesia haze".

Google (japanese nuclear disaster)					
Rank	**Original Result**	**Ranked Result**	**Rank**	**Original Result**	**Ranked Result**
1			**6**		
2			**7**		
3			**8**		
4			**9**		
5			**10**		

Table 6(a): All images for both original ranked result and re-ranked result are represented as positive relevant images, from Google search engine, given a query called "japanese nuclear disaster".

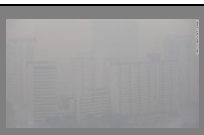
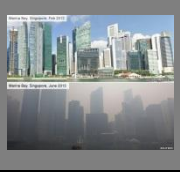
Google (indonesia haze)					
Rank	**Original Result**	**Ranked Result**	**Rank**	**Original Result**	**Ranked Result**
1			**6**		
2			**7**		
3			**8**		
4			**9**		
5			**10**		

Table 6(b): The background in dark color represented as positive relevant image for both original ranked result and re-ranked result, from Google search engine, given a query called "indonesia haze".

Yahoo (japanese nuclear disaster)					
Rank	**Original Result**	**Ranked Result**	**Rank**	**Original Result**	**Ranked Result**
1			**6**		
2			**7**		
3			**8**		
4			**9**		
5			**10**		

Table 7(a): The background in dark color represented as positive relevant image for both original ranked result and re-ranked result, from Yahoo search engine, given a query called “japanese nuclear disaster”.

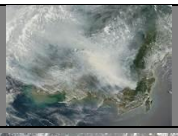
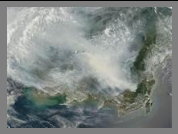
Yahoo (indonesia haze)					
Rank	**Original Result**	**Ranked Result**	**Rank**	**Original Result**	**Ranked Result**
1			**6**		
2			**7**		
3			**8**		
4			**9**		
5			**10**		

Table 7(b): All images for both original ranked result and re-ranked result are represented as positive relevant images, from Yahoo search engine, given a query called “indonesia haze”.

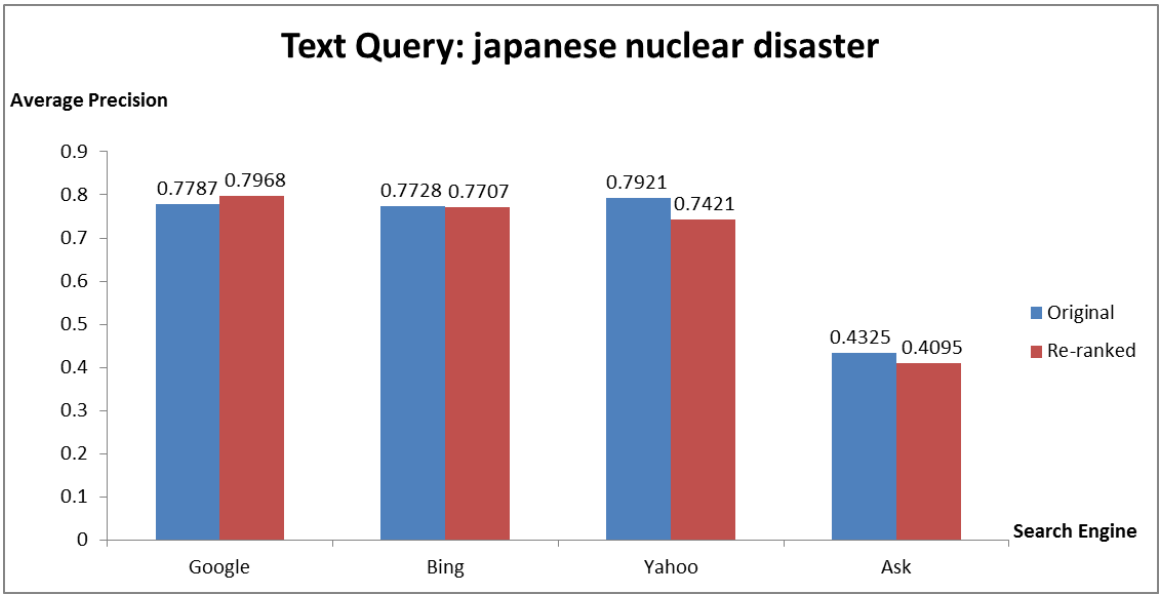


Figure 14(a): Bar chart – The average precision of the original vs. re-ranked results from top 10 images in each search engine, given a query called "japanese nuclear disaster".

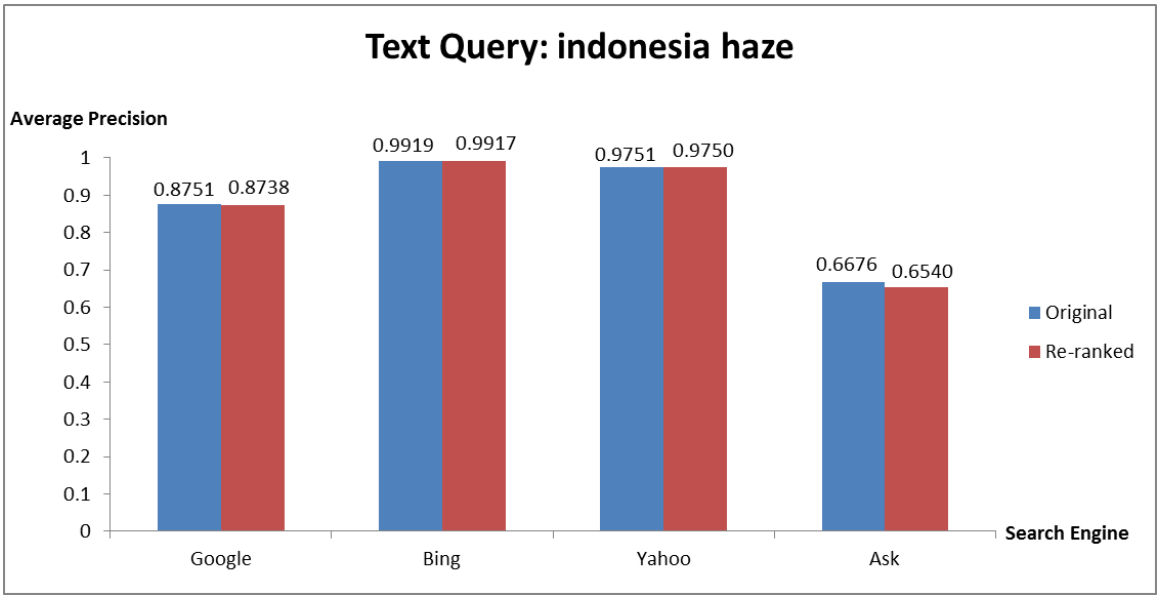


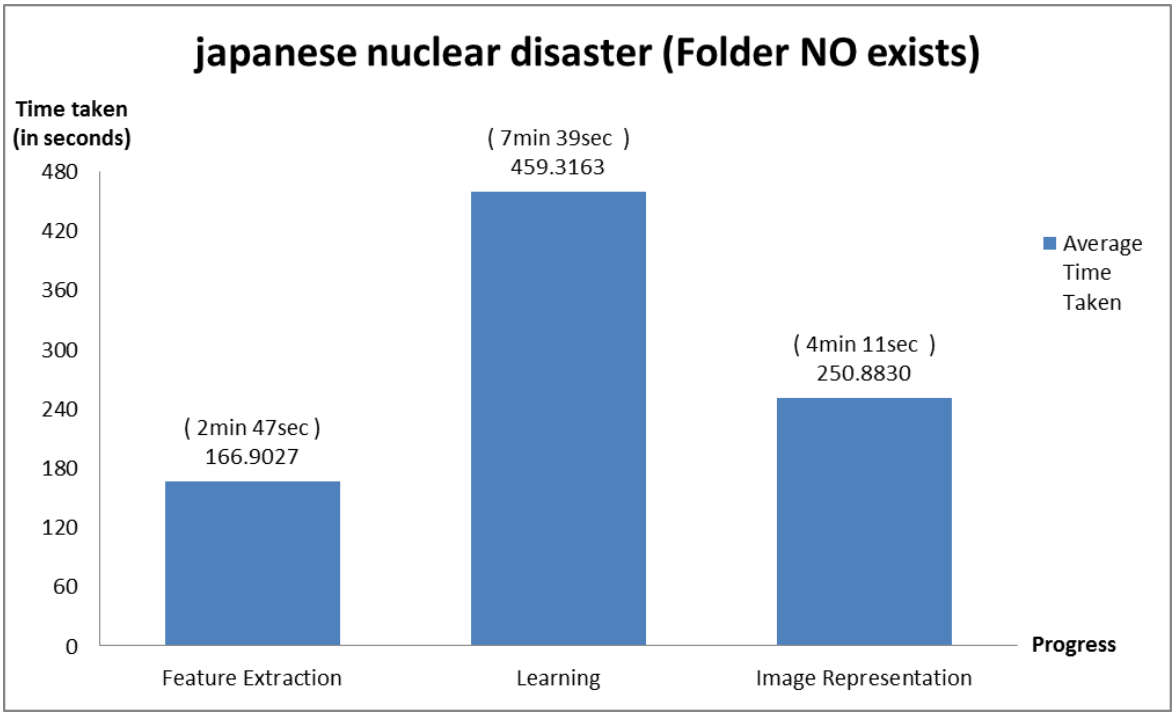
Figure 14(b): Bar chart – The average precision of the original vs. re-ranked results from top 10 images in each search engine, given a query called "indonesia haze".

4.3 Time Taken

In this sub-chapter, we will test and measure the processing time of such prototype system by providing two queries (where the 1st query called "japanese nuclear disaster" and the 2nd query called "indonesia haze", and each query consists of 1000 images collected from each 4 search engines, thus, each query contains total 4000 images). We will measure the processing time by determined the existing library folder, where such folder serves as the main resource in which used to perform the entire progress of such prototype system. Thus, in this sub-chapter, we will measure the processing time of such system based on two cases in which used to determine the existing of such library folder, and we will perform three times testing and running to measure the processing time of such system by providing such two queries in order to make the report more accurately.

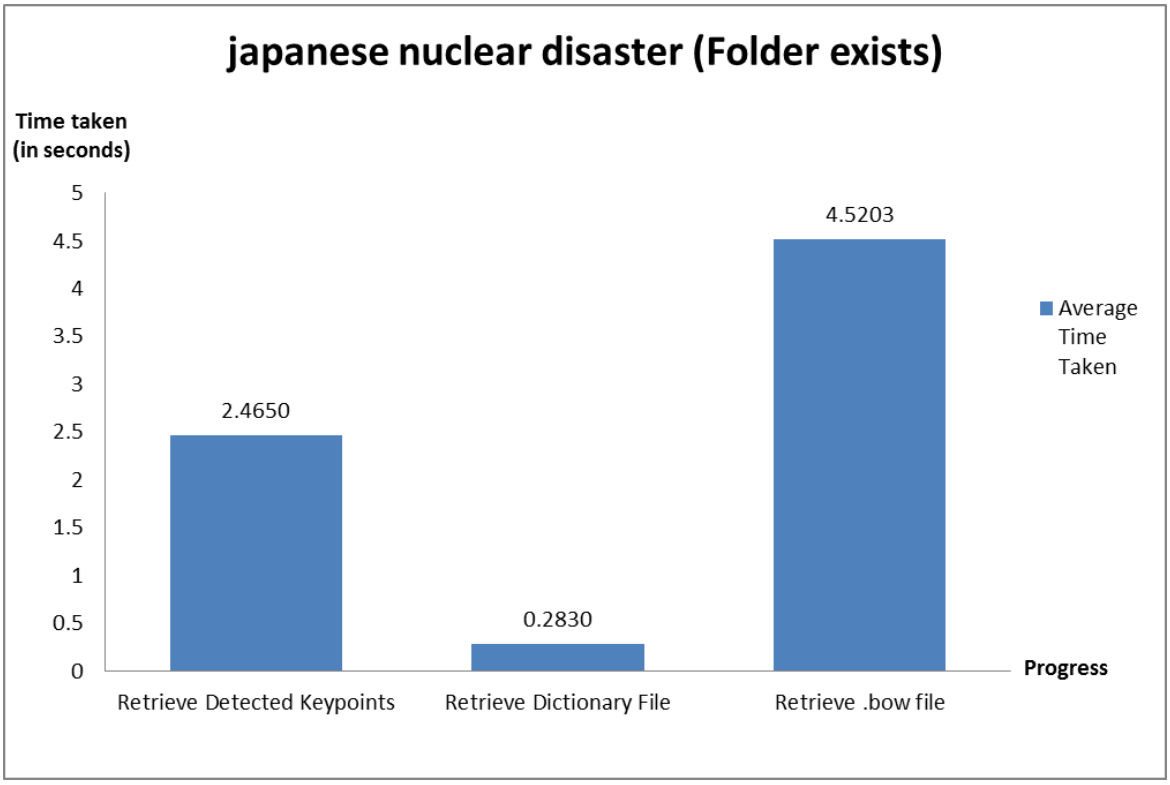
The first case is considered as such folder does not exist in the library. In this case, we will measure the processing time in creating such library files based on the process flow of the 1st three parts of the proposed methods, known as Feature-Extraction part, Learning part and Image-Representation part, in which have mentioned in chapter 3.3, the Proposed Methods. The 2 bar-charts as shown in Figure 15(a) and 16(a) shows the average processing time of such prototype system based on such case, where Figure 15(a) represented as the 1st provided query and Figure 16(a) represented as the 2nd provided query; For the 2nd case, we will measure the time taken of such system in retrieving those library files as shown in Figure 15(b) and 16(b), where each figure represented as each provided query.

After that, we will continue measure the processing time based on the process flow of the last three parts of the proposed methods, known as Knowledge-Mining part, Fusion part and Ranking part, in which have also mentioned in chapter 3.3, the Proposed Methods. The 2 bar-charts as shown in Figure 15(c) and 16(c) shows the average processing time in which perform such last three parts of proposed methods, where each figure represented as each provided query. The 2 bar-charts as shown in Figure 17(a) and 17(b) summarizes the total average time taken of the entire progress of such prototype system, where each figure represented as each provided query, and the two cases in each figure represented as nodes in which used to shows the average time per seconds in creating / retrieving library files and perform last three parts of the proposed methods to perform image re-ranking based on each query provided.



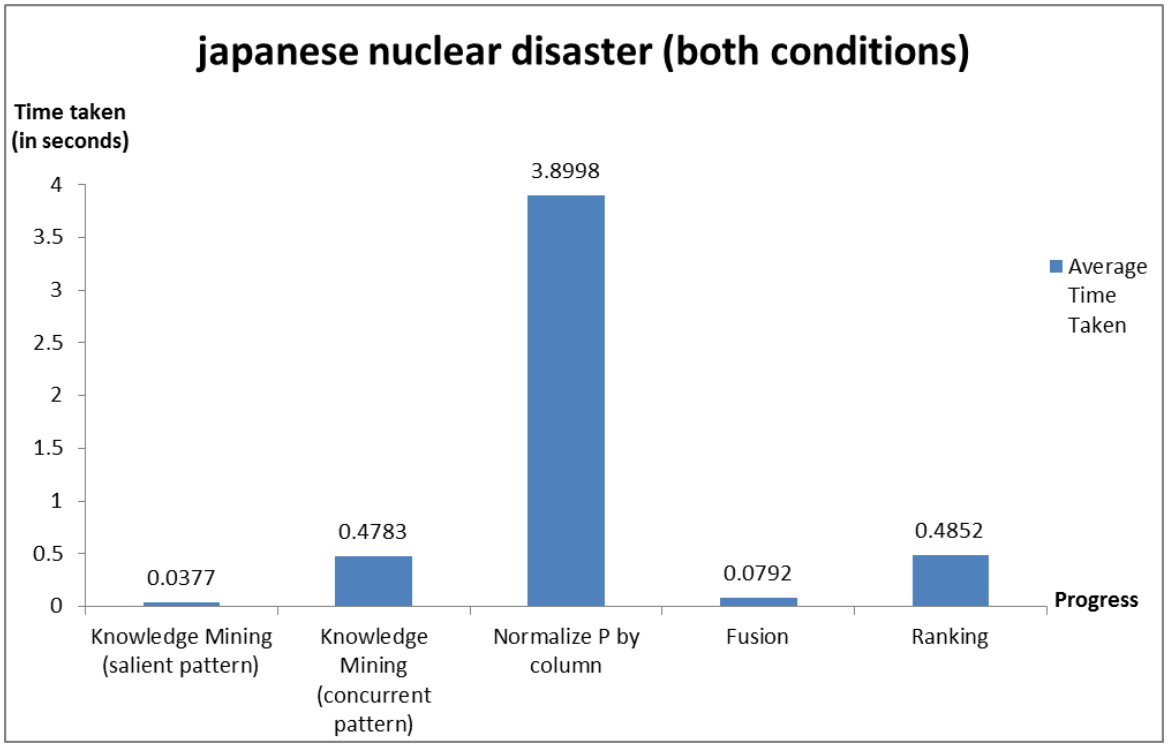
Create Library File (if Folder NO exists)			
Progress		**Average Time Taken**	
ID	**Name**	**in seconds**	**in minute / seconds**
1	Feature Extraction	166.9027	2min 47sec
2	Learning	459.3163	7min 39sec
3	Image Representation	250.8830	4min 11sec
	Total Average Time	**877.1020**	**14min 37sec**

Figure 15(a): Bar Chart (with table provided) – The average processing time in creating library files, given a query called “japanese nuclear disaster”.



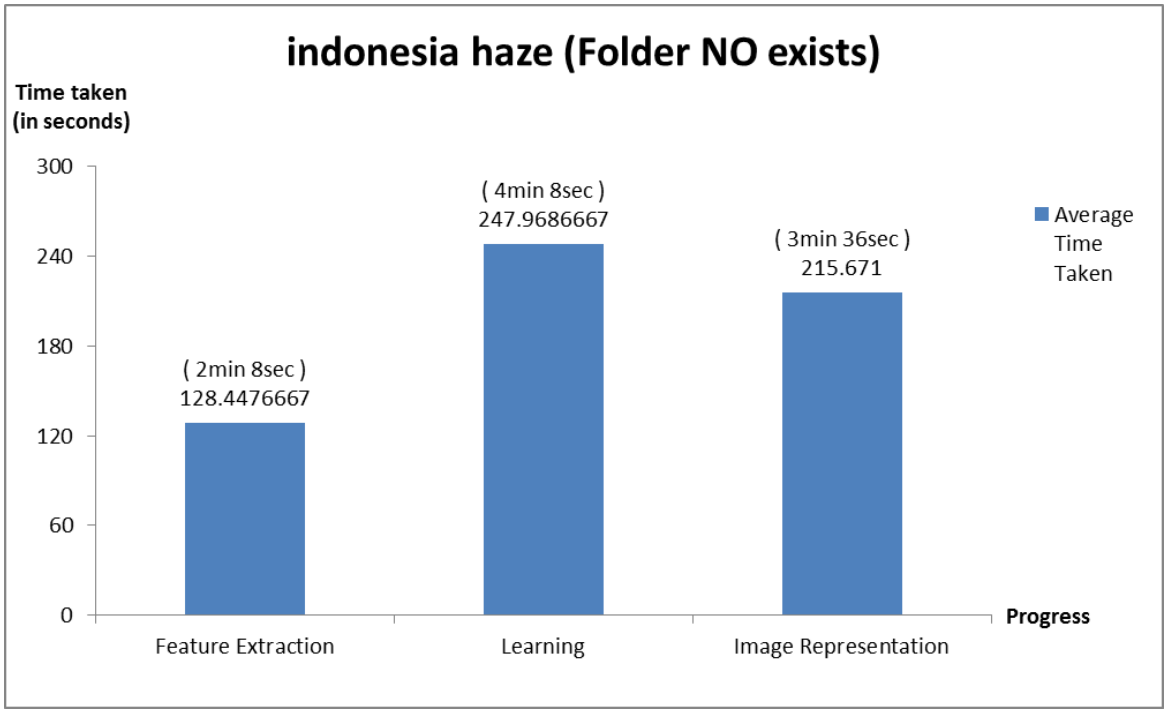
Load Library File (if Folder exists)		
Progress		Average Time Taken (in seconds)
ID	Name	
1	Retrieve Detected Keypoints	2.4650
2	Retrieve Dictionary File	0.2830
3	Retrieve .bow file	4.5203
	Total Average Time	**7.2683**

Figure 15(b): Bar Chart (with table provided) – The average processing time in loading library files, given a query called “japanese nuclear disaster”.



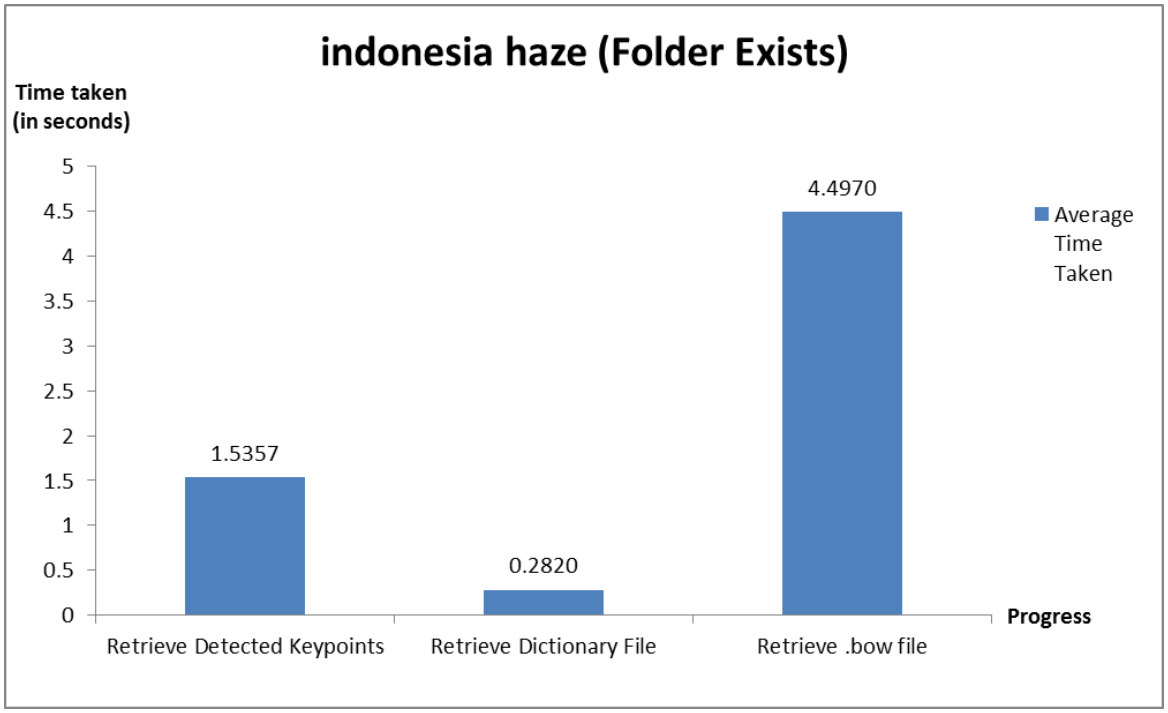
Learning Phase		
Progress		Average Time Taken (in seconds)
ID	Name	
1	Knowledge Mining (salient pattern)	0.0377
2	Knowledge Mining (concurrent pattern)	0.4783
3	Normalize P by column	3.8998
4	Fusion	0.0792
5	Ranking	0.4852
	Total Average Time	**4.9801**

Figure 15(c): Bar Chart (with table provided) – The average processing time in learning phase (for both cases, i.e. library folder have/no exists), given a query called "japanese nuclear disaster".



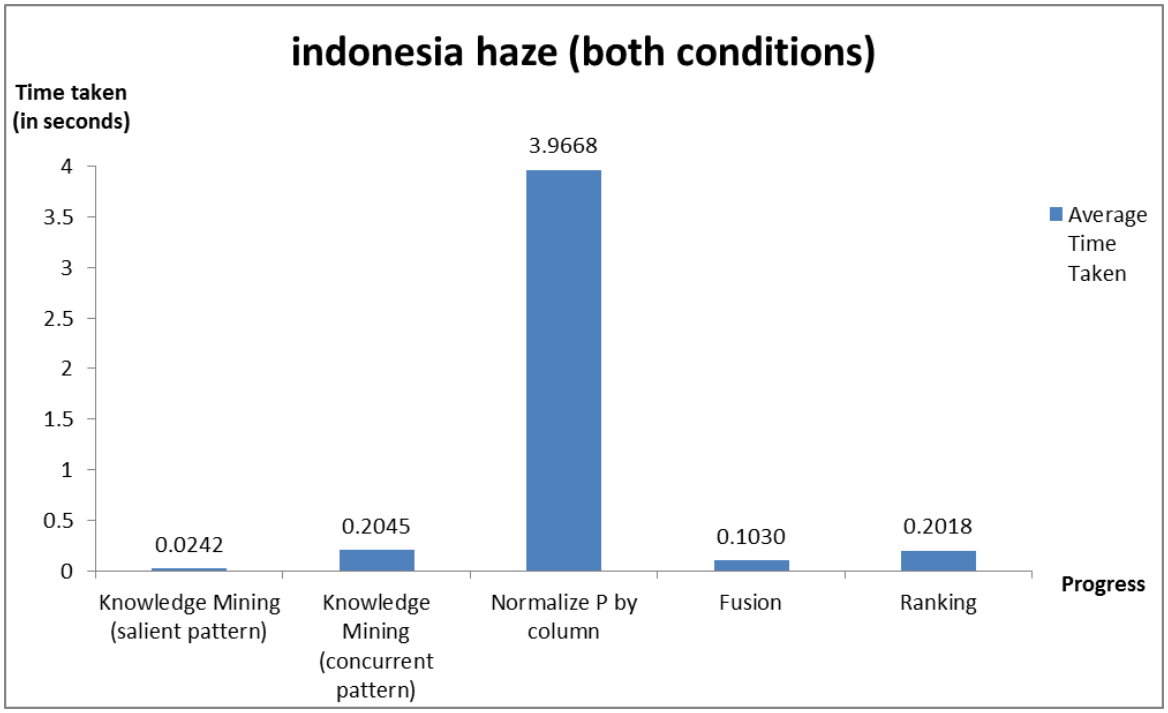
Create Library File (if Folder NO exists)			
Progress		**Average Time Taken**	
ID	**Name**	**in seconds**	**in minute / seconds**
1	Feature Extraction	128.4477	2min 8sec
2	Learning	247.9687	4min 8sec
3	Image Representation	215.6710	3min 36sec
	Total Average Time	**592.0873**	**9min 52sec**

Figure 16(a): Bar Chart (with table provided) –The average processing time in creating library files, given a query called “indonesia haze”.



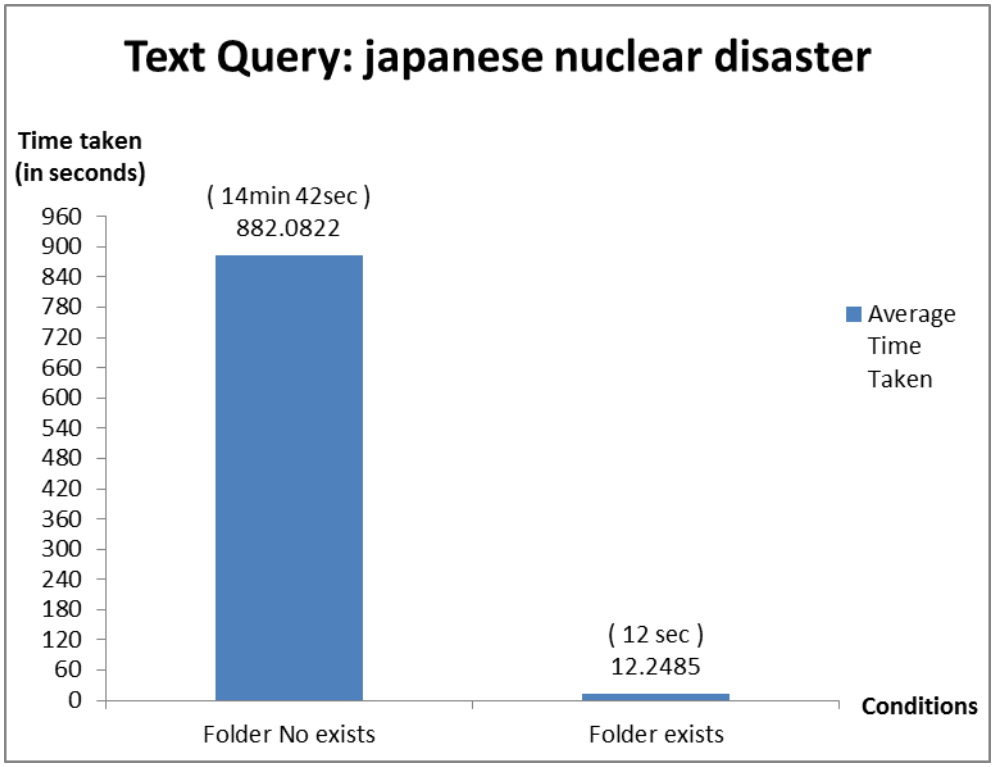
Load Library File (if folder exists)		
Progress		**Average Time Taken (in seconds)**
ID	**Name**	
1	Retrieve Detected Keypoints	1.5357
2	Retrieve Dictionary File	0.2820
3	Retrieve .bow file	4.4970
	Total Average Time	**6.3147**

Figure 16(b): Bar Chart (with table provided) – The average processing time in loading library files, given a query called "indonesia haze".



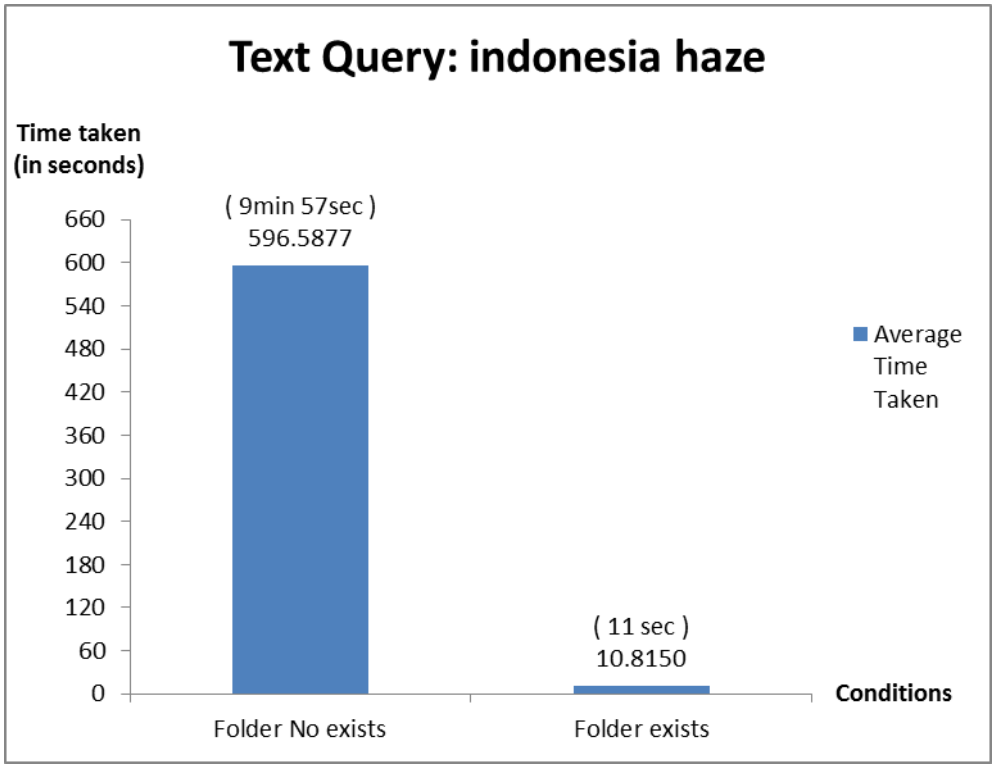
Learning Phase		
Progress		**Average Time Taken (in seconds)**
ID	**Name**	
1	Knowledge Mining (salient pattern)	0.0242
2	Knowledge Mining (concurrent pattern)	0.2045
3	Normalize P by column	3.9668
4	Fusion	0.1030
5	Ranking	0.2018
	Total Average Time	**4.5003**

Figure 16(c): Bar Chart (with table provided) – The average processing time in learning phase (for both cases, i.e. library folder have/no exists), given a query called "indonesia haze".



Conditions		Average Time Taken	
ID	**Name**	**in seconds**	**in minute / seconds**
1	Folder No exists	882.0821	14min 42sec
2	Folder exists	12.2485	12 sec

Figure 17(a): Bar Chart (with table provided) – Total average processing time, given a query called "japanese nuclear disaster".



Conditions		Average Time Taken	
ID	**Name**	**in seconds**	**in minute / seconds**
1	Folder No exists	596.5877	9min 57sec
2	Folder exists	10.8150	11 sec

Figure 17(b): Bar Chart (with table provided) – Total average processing time, given a query called "indonesia haze".

Chapter 5 - Conclusion

Crowd re-ranking is one of the re-ranking algorithm that under the category of re-ranking methods without user intervention, in which will perform image re-ranking and via multiple search engines automatically. Thus, in this project, in order to verify the proof of concept of such re-ranking algorithm, we are going to develop an offline prototype system that will able to implement such algorithm based on the proposed methods. By developing such prototype system, such system will able to serve as a foundation for future research purpose.

5.1 Limitation and Future Work

The proposed technique is still facing some minor problems. First of all, based on the progressing time, since we does not have a server to store and retrieve the library files, thus, for the first time in creating such library files (if does not exists), such system will used about 4 – 5 minutes to perform image feature extraction, 15 – 20 minutes to generate dictionary file by performing K-Means cluster, and 1 – 2 minutes to convert each image into Bag-of-Words representation from each search engine. Thus, such system will spend overall 20 – 30 minute to create such library files. However, it is depends on the number of keypoints detected in each image and also based on the CPU processing speed.

For some feature detectors, if current image is a positive relevant image but contains a mist with no proper structure, such detector will difficult to perform keypoints detection, in which the result may base on the following 2 conditions, where the 1st condition consider as no keypoints detected in such image; and the 2nd condition consider as such detectors perform incorrect detection result, in which the information contained in each keypoints does not related to the estimated information of such image.

For the 1st condition, in order to detect more keypoints, we will set the threshold value for such feature detector to become smaller. However, the smaller the threshold value, the more time taken (in create and retrieve) and storage capacity needed to consume. This is because such threshold value will apply the same detection methods in other images and different image contains different kind of structures. Lastly, for the 2nd condition, since such technique do not consider the present of the noises and ambiguous information contained in each image. Hence, such incorrect information will decrease the accuracy of the estimated information to perform keypoints detection. Thus, based on these 2 conditions, the re-ranking result may ranked incorrectly or does not ranked very well compared to the original ranked result (initial search result).

Reference

Mubarak Shah (2012) *Lecture 05 - Scale-invariant Feature Transform (SIFT)*. [video online] Available at: http://www.youtube.com/watch?v=NPcMS49V5hg [Accessed: March 2, 2013].

Baluja, S. and Jing, Y. (n.d.) *PageRank for Product Image Search*. [online] Available at: http://www.tamaraberg.com/teaching/Fall_08/presentations/PageRank.pdf [Accessed: March 10, 2013].

China-papers.com (2010) *Research on Video Search Reranking | China Papers*. [online] Available at: http://www.china-papers.com/?p=129492 [Accessed: Feb 28, 2013].

Inspiration.com (n.d.) *Visual learning and outlining in the classroom | inspiration.com*. [online] Available at: http://www.inspiration.com/visual-learning [Accessed: Feb 11, 2013].

Csie.ntu.edu.tw (2003) *Reranking Methods for Visual Search*. [online] Available at: http://www.csie.ntu.edu.tw/~winston/papers/hsu07reranking.html [Accessed: Feb 12, 2013].

Ebizmba.com (2013) *Top 15 Most Popular Search Engines*. [online] Available at: http://www.ebizmba.com/articles/search-engines [Accessed: March 7, 2013].

En.wikipedia.org (2012) *Visual Word - Wikipedia, the free encyclopedia*. [online] Available at: http://en.wikipedia.org/wiki/Visual_Word [Accessed: March 10, 2013].

Monash.com (2004) *A Helpful Guide to Web Search Engines -- Search Engines Ranking Methods andAlgorithms*. [online] Available at: http://www.monash.com/search_engine_ranking.html [Accessed: March 7, 2013].

Ijcsi.org (2013) *International Journal of Computer Science Issues - IJCSI*. [online] Available at: http://www.ijcsi.org/articles/Visual-keyword-based-image-retrieval.php [Accessed: March 10, 2013].

Mubarak Shah (2012) *Lecture 17: Bag-of-Features (Bag-of-Words)*. [video online] Available at: http://www.youtube.com/watch?v=iGZpJZhqEME [Accessed: Jan 20, 2013].

Nlp.stanford.edu (2001) *Relevance feedback and pseudo relevance feedback*. [online] Available at: http://nlp.stanford.edu/IR-book/html/htmledition/relevance-feedback-and-pseudo-relevance-feedback-1.html [Accessed: Feb 28, 2013].

Reference

Owl.english.purdue.edu (2010) *Purdue OWL: Pattern and Variation: Visual*. [online] Available at: http://owl.english.purdue.edu/owl/resource/569/01/ [Accessed: March 10, 2013].

Popescuy, A. et al. (n.d.) *Visual Reranking for Image Retrieval over the Wikipedia Corpus*. [online] Available at: http://clef.isti.cnr.it/2009/working_notes/myoupo-paperCLEF2009.pdf [Accessed: March 9, 2013].

Rong , Y. et al. (n.d.) *Multimedia Search with Pseudo-Relevance Feedback*. [online] Available at: http://www.aaai.org/Papers/Symposia/Spring/2003/SS-03-08/SS03-08-018.pdf [Accessed: Feb 28, 2013].

Scholarpedia.org (1949) *Mutual information - Scholarpedia*. [online] Available at: http://www.scholarpedia.org/article/Mutual_information [Accessed: March 5, 2013].

Ting Yao et al. (n.d.) *Co-reranking by Mutual Reinforcement for Image Search*. [online] Available at: http://vireo.cs.cityu.edu.hk/papers/Co-reranking%20by%20Mutual%20Reinforcement%20for%20Image%20Search_civr10.pdf [Accessed: Feb 23, 2013].

Winston, . et al. (2006) *Video Search Reranking via Information Bottleneck Principle*. [online] Available at: http://www.ee.columbia.edu/~lyndon/pubs/acmmm2006-rerank.pdf [Accessed: Feb 26, 2013].

Winston, H. et al. (n.d.) *Video Search Reranking through Random Walk over Document-Level Context Graph*. [online] Available at: http://www.ee.columbia.edu/ln/dvmm/publications/07/hsu07video.pdf [Accessed: Feb 12, 2013].