

**Web Content Mining**  
**Web Mining and Data Analysis**



**Autumn 2005**

**HØGSKOLEN I AGDER**  
**Agder University College**  
**Faculty of Engineering and**  
**Science**

<b>Authors:</b> Wen Hu Li Zhang Xianghan Zheng	<b>Supervisors:</b> Ole-Christoffer Granmo Mikael Snaprud
<b>Version: 1.0</b> <b>Status: FINAL</b>	<b>Pages: 25</b> <b>Modified date: 2005-11-20</b>
<b>Keywords:</b> web content mining, image classification, entropy, line-scan, standard deviation	
<b>Abstract:</b> <p>The goal of this project is to create a crawler/classifier that downloads the images in a web page and tries to classify the content of each image into different categories, e.g., mathematical formula, logo, buttons, and so on. The focus should be on automatic detection of image usage that reduces the accessibility of a web page.</p> <p>Finally, we accomplish to classify the images into real image, graphs, and text (include formula) by using three algorithms. The three algorithms are Entropy, Line-Scan, and Standard Deviation. We text 10 images each of categories with these three algorithms in both using filter and without using filter.</p> <p>By analysis the results, we can get a conclusion that Entropy turn out to be the best algorithm for it takes the least time to get the test results and shows the most notable effect. And Standard deviation is the worst one. It costs long time to run but not show very powerful function. And its worth to mention that, using filter makes the test result very different from without using it. It can make the differences between each category more visible.</p> <p>Our web page is : <a href="http://home.hia.no/~xiangz05/">http://home.hia.no/~xiangz05/</a></p>	
<b>License:</b> This paper is copyrighted and protected by the laws. The paper is not for sale, but can be used by Agder University College for teaching purposes.	

# Table of contents

Executive summary.....	3
Chapter 1 Introduction.....	4
1.1 Motivation.....	4
1.2 The objectives of this project.....	4
1.3 The structure of this report.....	5
Chapter 2.....	6
Basic concepts and approaches.....	6
2.1 Digital image.....	6
2.2 Gray level and Pixel.....	6
2.3 Band.....	7
2.4 The image histogram.....	7
2.5 Edge detection.....	8
2.5.1 Filter and Edge enhancement filters.....	8
2.5.2 Edge detection.....	8
Chapter 3 Algorithms.....	11
3.1 Entropy.....	11
3.2 Standard deviation.....	11
3.3 Pixel search.....	12
Chapter 4 Text result.....	14
4.1 Text method.....	14
4.2 Test result.....	14
4.2.1 Entropy.....	14
4.2.2 Standard deviation.....	16
4.2.3 Pixel search.....	17
4.2.4 Entropy after filter.....	18
4.2.5 Standard deviation after filter.....	19
4.2.6 Pixel search after filter.....	20
4.3 Comparison of each algorithm.....	22
Chapter 5 Conclusion.....	24
Appendix.....	25
A1 References.....	25

## Executive summary

With the phenomenal growth of the Web, there is an ever-increasing volume of data and information published in numerous Web pages. The research in Web mining aims to develop new techniques to effectively extract and mine useful knowledge/information from these Web pages. Due to the heterogeneity and the lack of structure of Web data, automated discovery of targeted or unexpected knowledge/information is a challenging task. It calls for novel methods that draw from a wide range of fields spanning data mining, machine learning, natural language processing, statistics, databases, and information retrieval. In the past few years, there was already a rapid expansion of activities in the Web mining field, which consists of Web usage mining, Web structure mining, and Web content mining. Web mining is used to categorize users and pages by analyzing the users' behavior, the content of the pages, and the order of the URLs that tend to be accessed in order. Web usage mining refers to the discovery of user access patterns from Web usage logs. Web structure mining tries to discover useful knowledge from the structure of hyperlinks. Web content mining aims to extract/mine useful information or knowledge from web page contents.

For this special issue, we focused on web content mining which deals with the discovery of useful information from web content. This project leads to the making of a web crawler which retrieves and categorizes images on web pages. The images are categorized in real images, graphics and text (which include formulas).

To find the right approaches and algorithms, a test program was written. This test program tested the algorithms on 10 images from each of the categories. Three algorithms showed a big difference on the different categories. Filter technology which also used in the project makes the difference distinctly between the text and non-text, and then compare the two cases.

The best algorithm turned out to be entropy for both images after filter and original ones. It takes the least time to get the result among three algorithms and the effect of entropy is the most distinct.

A pixel search algorithm is also used. This algorithm is called line-scan, and it is not especially good, although the programming is the simplest.

Standard deviation is the worst among the three. It costs a long time to get the result and its effect is inconspicuous.

This solution can be used to reveal the bad use of graphics at web pages. It might also be used as a plug-in in browsers to give additional image information for those who cannot see them.

# Chapter 1 Introduction

## 1.1 Motivation

In recent years the growth of the World Wide Web exceeded all expectations. Today there are several billions of HTML documents, pictures and other multimedia files available via internet and the number is still rising. But considering the impressive variety of the web, retrieving interesting content has become a very difficult task. In the past few years, there was already a rapid expansion of activities in the Web mining field which is used to discover the content of the web, the users' behavior in the past, and the WebPages that the users want to view in the future. Web mining consists of web content mining, web structure mining and web usage mining. In this paper, we focused on web content mining which deals with the discovery of useful information from web content.

Web Content Mining uses the ideas and principles of data mining and knowledge discovery to screen more specific data. The use of the Web as a provider of information is unfortunately more complex than working with static databases. Because of its very dynamic nature and its vast number of documents, there is a need for new solutions that are not depending on accessing the complete data on the outset. Another important aspect is the presentation of query results. Due to its enormous size, a web query can retrieve thousands of resulting WebPages. Thus meaningful methods for presenting these large results are necessary to help a user to select the most interesting content.

## 1.2 The objectives of this project

The goal of this project is to create a crawler/classifier that downloads the images in a web page and tries to classify the content of each image into different categories, e.g., mathematical formula, logo, buttons, and so on. The focus should be on automatic detection of image usage that reduces the accessibility of a web page.

In this work we first used Harvestman to download the images in a random web page, and then our main job is to differ between text (including formulas) and non-text, which are image usage that reduces the accessibility. In addition the realistic photos (like 3D modeling that use advanced algorithms like ray-tracing) and game screen shots are very similar to the real photos category. It is not emphasized to differ this category from real photos, because it is of no real use to know the difference. Then we use filter technology to make the more difference among the images when we test the all various kinds of images. Finally, we compare the two cases which using filter and

none, and analyze third algorithms according to the test result.

## **1.3 The structure of this report**

The rest of this paper is organized as follows. Some basic concepts, for example, grey level and pixel, band, the image histogram and edge detection, are presented in this chapter. These let us understand more about content of the paper.

Three algorithms which is entropy, standard deviation and pixel search, are used to solve the task are presented in section 3.

Section 4 presents a test-bed that tests the algorithms. The algorithms are tested on images from the categories, and the result is presented in graphs so the difference of the categories easily can be seen. We use two methods to test algorithms. One is filter technology which used in the project makes the difference distinctly between the text and non-text. Then we compare the two results when using the filter technology and not, and discuss three algorithms which one is the best to classify the images.

Finally, the last section makes up the conclusion which is some finishing words about the results, the testing process and the future work.

# Chapter 2

## Basic concepts and approaches

### 2.1 Digital image

An image is a picture, photograph, display, or other form giving a visual representation of an object or scene. However, in digital image processing, it has another meaning:

A digital image is an image  $f(x,y)$  that has been discretized both in spatial coordinates and brightness.

### 2.2 Gray level and Pixel

Fig 1 is a digital image. As it shows, Each number in Fig 1 corresponds to one small area of the visual image, and the number gives the level of darkness or lightness of the area..

40	40	20	40
20	20	20	20
40	20	20	20
40	40	40	40

**Fig 1 A digital image**

We will assume that the higher the member, the lighter the area, so zero is black, the maximum value is white, and intermediate values are shades of grey. These values called **grey levels**. Each small area to which a number is assigned is called a “**pixel**”, which is short for picture element ([1]). The size of the physical area represented by a pixel is called the spatial resolution of the pixel. This varies greatly, from a few nanometers in microscope images to tens of kilometers in satellite images. Each pixel has its value, plus a line coordinate and a sample coordinate. These give its location in the image array. For example, the pixel at line 3, sample 2 has value 20.

The minimum value a pixel can have is typically 0, and the maximum depends on how the number is stored in the computer. Different formats allow different maximums. One way is store each pixel as a single bit, which means it can take only the values 0 and 1, or black and white. Another common way is to store each pixel as a byte, which is 8 bits. In this form the maximum pixel value is 255.

## 2.3 Band

When we deal with color, we may have images of the three components red, green, and blue. Multispectral scanner instruments flown on aircraft and satellites typically gather from 3 to 11 images. All images are collectively referred to as “the image”, the individual images as “**bands**” of the image, and we may speak, for example, of the red band of an image. In most cases, the bands are considered to be aligned so that they superimpose and can, for example, be displayed on the three colors of a color display monitor with no shift or displacement between them.

## 2.4 The image histogram

Often we would like to have some measure of the distribution of the pixel values in an image. We can use the mean  $\bar{v}$  and standard deviation  $\sigma$  :

$$\bar{v} = \frac{1}{n} \sum_{i,j} v(i, j) \quad \sigma = \sqrt{\frac{1}{n} \sum_{i,j} (v(i, j) - \bar{v})^2} \quad 2.4.1$$

where  $n$  is the number of pixels in the image. In addition to these, one of the most common measures of the pixel value is a (single band ) image is a table giving the number of pixels having each possible value  $v$ . This table, which is often given as a plot, is called the **image histogram** and we denote it by  $h(v)$  ([2]). The domain of the histogram is the set of possible pixel values. If the image has 8 bit pixels, this is interval 0 to 255. The histogram may be computed over the entire image, or only over a portion of the image which is of interest. If  $n_a$  is the number of pixels in the area over which the histogram is computed, it is easy to see that:

$$\sum_v h(v) = n_a \quad 2.4.2$$

Often the histogram is normalized:

$$H(v) = \frac{h(v)}{n_a} \quad 2.4.3$$

$H(v)$  is analogous to the probability density function of statistics, and may be considered as the probability of a pixel having value  $v$ . In this case,

$$\sum_v H(v) = 1 \quad 2.4.4$$

For two bands and a specified area, we may also compute a two dimensional table  $h(v,w)$  giving the number of pixel having value  $v$  in the first band and value  $w$  in the second. This is called the two dimensional histogram, or sometimes the scatter plot or scatter diagram, of the two bands. The domain of  $h(v,w)$  is the rectangular region of the plane bounded by  $(0, v_{\max})$  and  $(0, w_{\max})$ . For 8 bit pixels, this is the region

(0,255)\*(0,255). Similar to the one dimensional histogram, the two dimensional histogram satisfies:

$$\sum_v \sum_w h(v, w) = n_a \quad 2.4.5$$

for  $n_a$  the number of pixels in the area of the histogram.

## 2.5 Edge detection

### 2.5.1 Filter and Edge enhancement filters

An enhancement filter attempts to improve the quality of an image for human or machine interpretability, where quality is measured subjectively. Most enhancement filters are heuristic and problem oriented, and models of the degradation are generally not used in deriving them.

Edge enhancement filters are high pass filters and their effect is to enhance or boost edges. The term “edge detector” is also used. This may mean a simple high pass filter, but sometimes may be more general, including a threshold of the points into edge and non-edge categories, and even linking up of edge pixels into connected boundaries in the image.

### 2.5.2 Edge detection

**Edge detection** is one of the most commonly used operations in image analysis, and there are probably more algorithms in the literature for enhancing and detecting edges than for any object. An edge is the boundary between an object and the background, and indicates the boundary between overlapping objects.

Technically, edge detection is the process of locating the edge pixels, and edge enhancement will increase the contrast between the edges and the background so that the edges become more visible.

The most common method of differentiation in image processing applications is the **gradient operators ([3])**.

For a function  $f(x,y)$ , the gradient of  $f$  at coordinates  $(x,y)$  is defined as the vector

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad 2.5.1$$

The magnitude of this vector,

$$\nabla f = \text{mag}(\nabla f) = \left[ \left( \frac{\partial f}{\partial x} \right)^2 + \left( \frac{\partial f}{\partial y} \right)^2 \right]^{1/2} \quad 2.5.2$$

is the basic for various approaches to image differentiation. Consider the image region shown in Fig 2, where the z's denote the values of gray levels. Equation 2.5.2 can be approximated at point  $z_5$  in a number of ways.

$z_1$	$z_2$	$z_3$
$z_4$	$z_5$	$z_6$
$z_7$	$z_8$	$z_9$

**Fig 2**

The simplest is to use the difference  $(z_5 - z_8)$  in the x direction and  $(z_5 - z_6)$  in the y direction, combined as

$$\nabla f \approx \left[ (z_5 - z_8)^2 + (z_5 - z_6)^2 \right]^{1/2} \quad 2.5.3$$

Instead of using squares and square roots, we can obtain similar results by using absolute values:

$$\nabla f \approx |z_5 - z_8| + |z_5 - z_6| \quad 2.5.4$$

Another approach for approximating Eq.2.5.2 is to use cross differences:

$$\nabla f \approx \left[ (z_5 - z_9)^2 + (z_6 - z_8)^2 \right]^{1/2} \quad 2.5.5$$

or, using absolute values,

$$\nabla f \approx |z_5 - z_9| + |z_6 - z_8| \quad 2.5.6$$

Equations 2.5.3 ~2.5.6 can be implemented by using masks of size 2\*2. For example, Eq. 2.5.6 can be implemented by taking the absolute value of the response of the two masks shown in Fig 3(a) and summing the results. These masks are called the **Roberts cross-gradient operators**.

Masks of even sizes are awkward to implement. An approximation to Eq.2.5.2, still at point  $z_5$  but now using a 3\*3 neighborhood, is

$$\nabla f \approx \left| (z_7 + z_8 + z_9) - (z_1 + z_2 + z_3) \right| + \left| (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7) \right| \quad 2.5.7$$

The difference between the third and first row of the 3\*3 region approximates the derivative in the x direction, and the difference between the third and first column approximates the derivative in the y direction. The masks shown in Fig 3(b), called

the **Prewitt operators**, can be used to implement Eq. 2.5.7. Finally, Fig 3(c) shows yet another pair of masks, called the **Sobel operators**, for approximating the magnitude of the gradient.

The formulas in chapter 2.5.2 are referred in [4].

1	0
0	-1

0	1
-1	0

(a) Roberts

-1	-1	-1
0	0	0
1	1	1

-1	0	1
-1	0	1
-1	0	1

(b) Prewitt

-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

(c) Sobel

**Fig 3**

# Chapter 3 Algorithms

## 3.1 Entropy

The entropy ((5)) is one of the most fundamental and revealing quantities that can be associated with stochastic information sequence. The basic concept of entropy in information theory has to do with how much randomness there is in a signal or random event. An accurate estimate of the entropy provides an indication of the amount of redundancy contained in the sequence and, consequently, an upper bound on the data compression possible. This statistical redundancy, which is related to the correlation and predictability of the data, can be removed without destroying any information. In this work, we estimate the entropy of grey-level images.

Suppose that we have a  $(M \times N)$  dimensional monochrome image  $X$ , where each pixel can take one of  $2^k$  luminance values with  $k$  equal to the number of bits/pixel.

Let a luminance value  $x$  have probability of occurrence  $p_i$  in the image. The entropy,

$H(x)$ , is defined as

$$H(x) = -\sum p_i \log(p_i) \text{ Bits/pixel}$$

The entropy represents the average information rate per symbol. In a more mathematical sense and wider use entropy is any quantity having properties analogous to those of the physical quantity, especially the quantity  $-\sum x_i \log(x_i)$  of a distribution  $\{x_1, x_2, \dots, x_n\}$ . The entropy of an image will hopefully give high values to real images and low values to text and graphics.

## 3.2 Standard deviation

Deviation is a statistical term which expresses how distributed the single values are in comparison to the mean value. The standard deviation is kind of the "mean of the mean," and often can help you find the story behind the data. The formula is

$$std = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2}$$

Where  $\bar{x}$  is the mean value. The mean value is the average color value of each pixel.

The standard deviation is a statistic that tells you how tightly all the various examples are clustered around the mean in a set of data. When the examples are pretty tightly bunched together and the bell-shaped curve is steep, the standard deviation is small and it means the colors to the image are in the similar. When the examples are spread apart and the bell curve is relatively flat, that tells you have a relatively large standard deviation and it means the image has big contrasts. In (6) there is information theory about standard deviation.

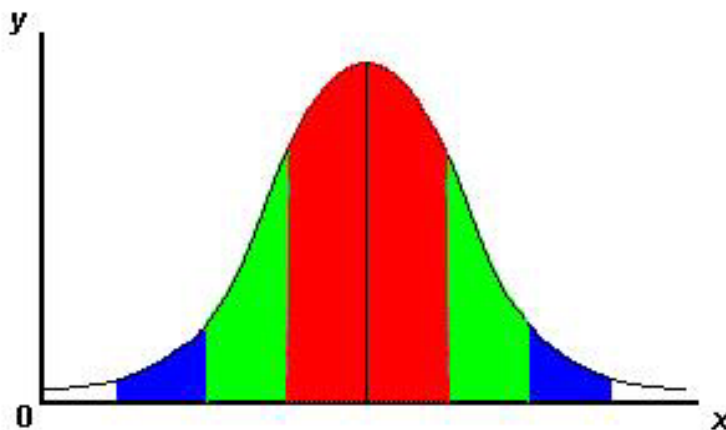


Figure 4: graph of standard deviation

One standard deviation away from the mean in either direction on the horizontal axis (the red area on the above graph) accounts for somewhere around 68 percent. Two standard deviations away from the mean (the red and green areas) account for roughly 95 percent. And three standard deviations (the red, green and blue areas) account for about 99 percent.

If this curve were flatter and more spread out, the standard deviation would have to be larger in order to account for those 68 percent or so. So that's why the standard deviation can tell you how spread out the examples in a set are from the mean

This algorithm will probably split single colored images or images with many similar colors from images which contains a broad spectrum of colors.

### 3.3 Pixel search

We have chosen to call this method line-scan, after the way it processes images. The theory is that images of text will have two colors which make up most of the image. These two colors are then scanned for the blank lines which appear in text between lines and characters/ words.

At first the image is converted to a 8-bit gray-scale image. This is done so that we don't have to process the image in 3 different color-bands. Additionally the chance

that some pixels of a different color could end up being converted to the same color decreases. But this error would be partly corrected by the other algorithms.

Thereafter the histogram of the image is searched and the two highest color values are stored. The values are marked as background color and foreground color. If these makes up less than 85% of the image this method concludes with the fact that this are not text and pass this value back. The pictures are now scanned vertically for lines that contain less than a given percentage of the background color. This scan returns a list containing top and bottom vertical position of potential strings. If there are a distinct number of these lines a percentage that the image is text is added.

Afterwards these lines are scanned horizontally one after another. We now have a 4-point value specifying each character. If there are enough of these we add a weighted chance that we are dealing with an image of text.

# Chapter 4 Text result

We use three kinds of images to test, real image, text or formula, and graphic. One thing should be mentioned, text and formula are not to be categorized in this report because in most case they will have very close result value. So, in some place, text or formula will be just mentioned as text.

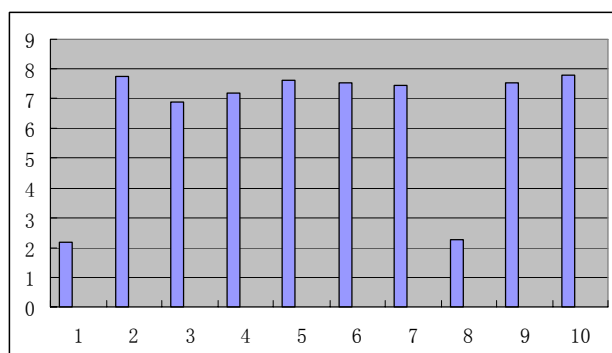
## 4.1 Text method

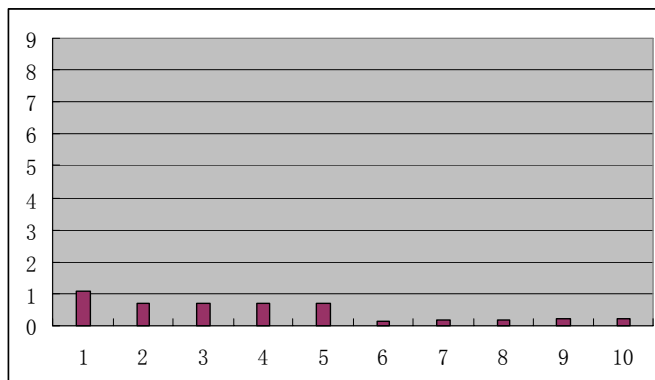
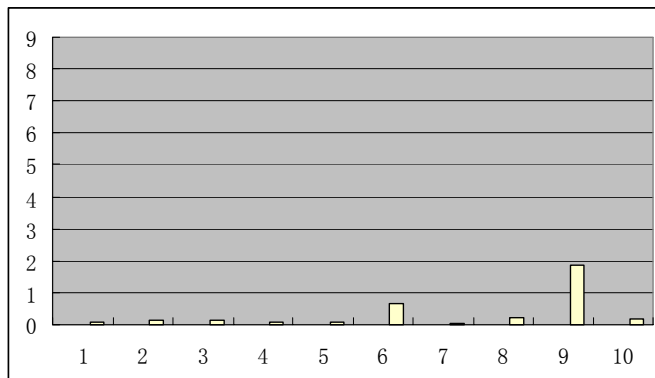
30 images are used to test. 10 is real image, 10 is text or formula (5 text and 5 formula), and the other 10 is graphic. These 30 images are in different size, different format, and manually categorized in three local folders. Three algorithms are used to analyze these 30 images. Then we add image filter technology to these three algorithms to see if we can get more clear result among each kind of images. The efficiency of the algorithm is based mainly on exactness, and the execution speed. Exactness is measured by the extent to which these three algorithms differentiate each kind of images. The execution speed is measured by the time a certain algorithm takes to get the result. The result is presented in graphic, and late more exactly in table, with the image number in X-axis and the result value in Y-axis.

## 4.2 Test result

### 4.2.1 Entropy

Real image



**Text or formula****Graphic**

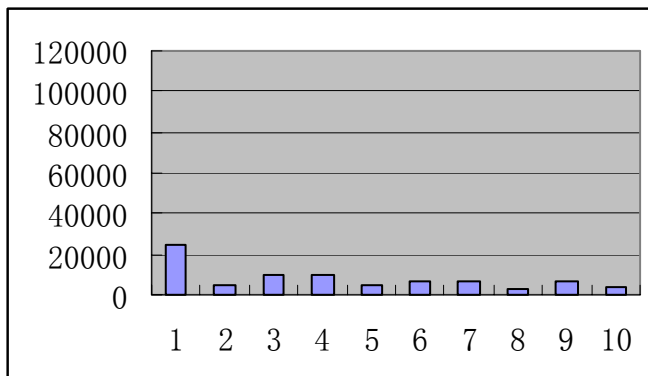
Entropy is can differentiate the real image very efficient, since real images often have high value, usually above 7, while text or graphic have lower value, usually lower than 1. Real image 1 and real image 8 have especially lower value than the other real images because they have much more pixels that have the same value than the others.

Text has low value than the real image, often lower than 1, but often higher than the graphic. However, since the not very high difference between text and graphic, it is not efficient to use entropy to directly distinguish.

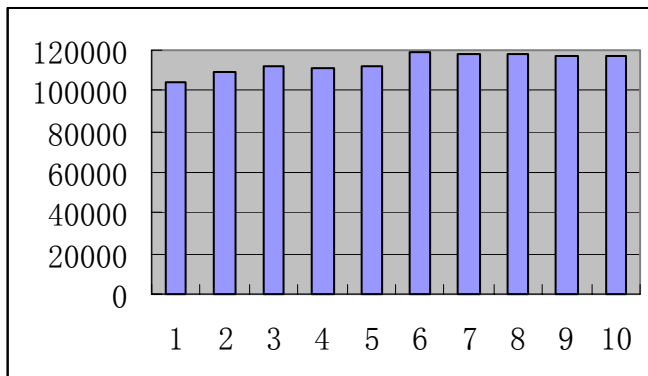
It takes 4.2 seconds for entropy to get the result value of 30 images.

## 4.2.2 Standard deviation

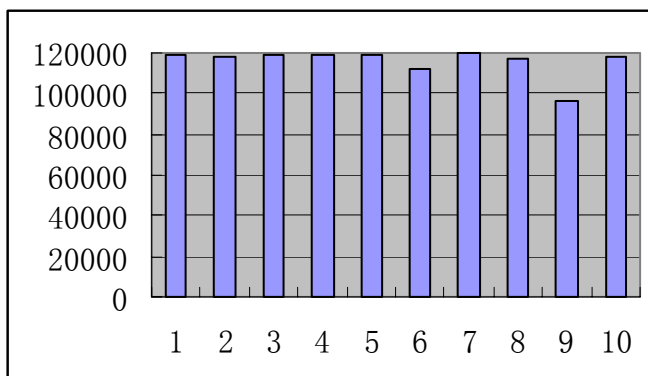
### Real image



### Text or formular



### Graphic

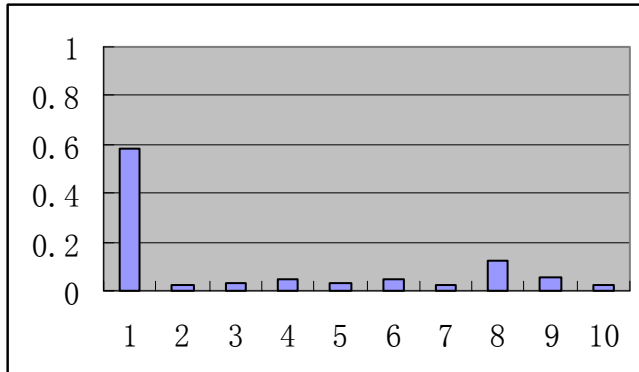


In standard deviation, real images have relative lower value than the other kinds of images, while text and graphic have much more higher value. So it is easy to differentiate the real image. However, Standard deviation also could not do well in differentiating the images between text and graphic, since they have very close value.

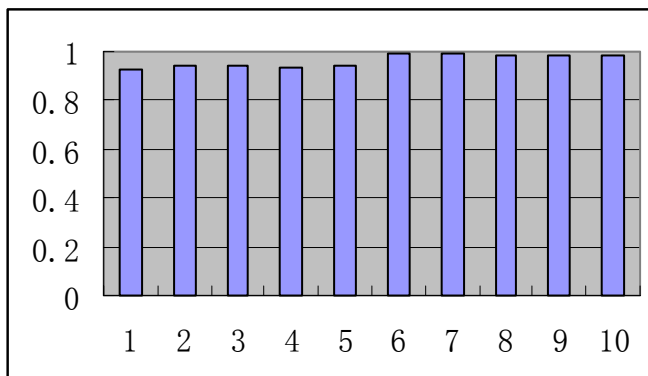
It takes 5.1 seconds to get the result of 30 images.

### 4.2.3 Pixel search

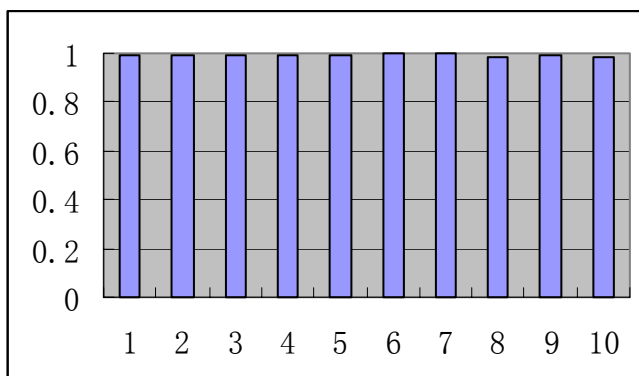
#### Real image



#### Text or formula



#### Graphics



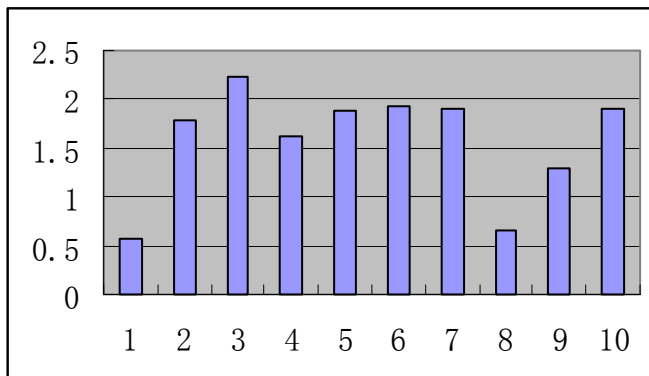
The pixel search is also good at differentiating real image, since the value of real image is usually below 0.1. Real image 1 and real image 8 have relative higher value than the other real images because these two real images have more white value pixels(255,255,255) than the other real images. With the 0.85 as the boundary between real image and the other kinds of images, it is easy to differentiate real images.

Text or graphic have a higher value, usually higher than 0.95. Although there are some difference between text and graphic, it is also difficult for standard deviation to categorize the text and graphic.

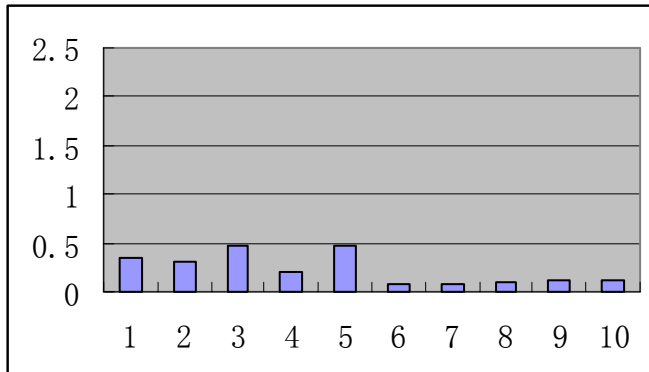
It takes 4.5 second for pixel search to get the result of 30 images.

### 4.2.4 Entropy after filter

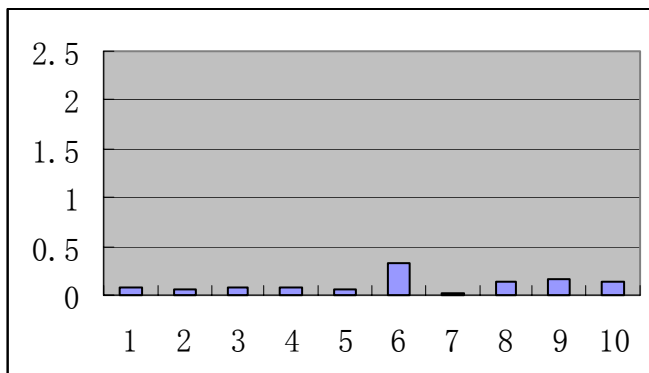
#### Real image



#### Text



#### Graphic



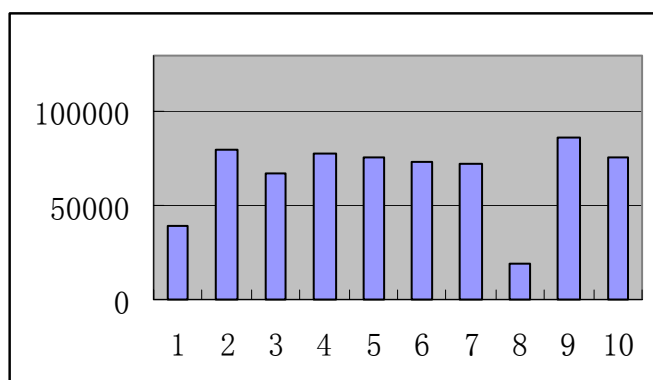
Entropy after filter can differentiate different kinds of images quite well. Real image usually have a highest value, usually 4-6 times higher than text. And the value of the text usually has 4-6 times higher than the graphic.

The disadvantage of this method is the same as the entropy above. That is do not analyze the images that have many the same value pixel, like real image 1 and real image 8.

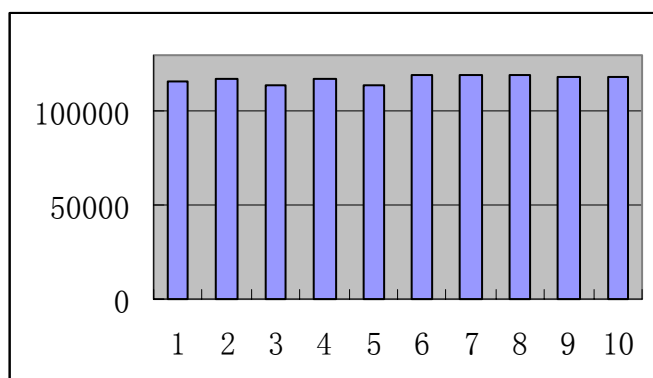
However, the exactness is at the cost of the CPU time. It cost a long time to analyze the image filter part. It takes 121 seconds to analyze 30 images and get the result.

## 4.2.5 Standard deviation after filter

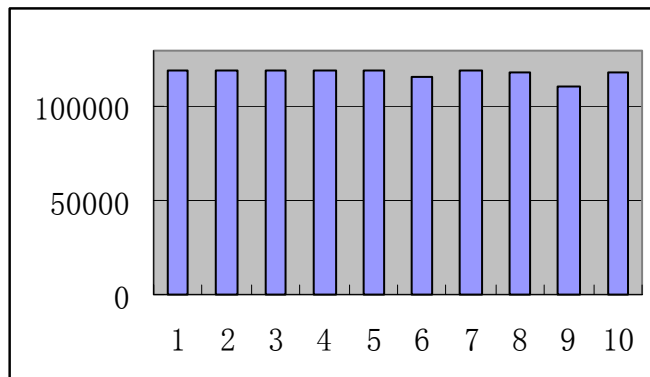
### Real image



### Text



## Graphic

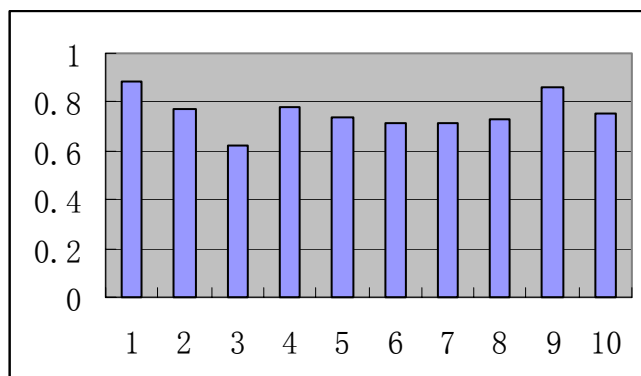


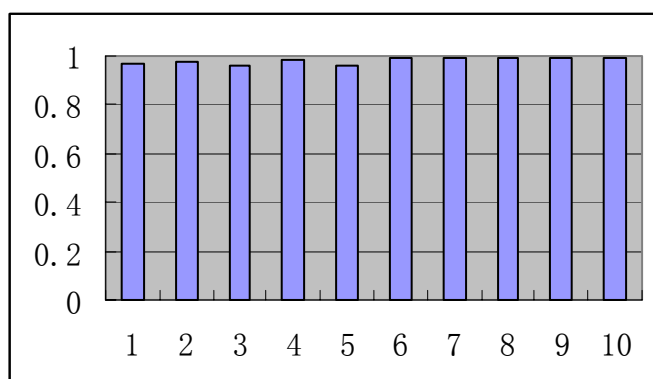
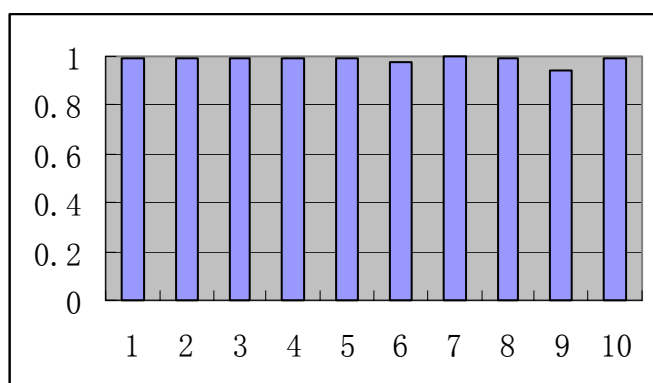
Using image filter in Standard deviation do not work better than not using image filter. From 4.2.2, the value of text and graphic is usually ten times larger than the value of real image. After using image filter technology, the value difference between real image and the other kinds of images become not larger, but even smaller. So, it is not an efficient way to analyze images in this way.

This method also consumes a lot of CPU time. I takes 126 seconds to analyze 30 images.

## 4.2.6 Pixel search after filter

### Real image



**Text or formula****Graphic**

Pixel search after filter also do not works better than not using image filter. In 4.2.3, real image usually have the very low value, lower than 0.1. After using image filter technology, the value of real image is usually above 0.6 and below 0.85. Although this method could still be used in categorizing images, it is not efficient and will cause more difficulty in distinguishing real image from different kinds of images.

It takes 124 seconds to analyze 30 images. So, in this method, much time was consumed in analyzing the image filter part, only to get the worse result.

### 4.3 Comparison of each algorithm

Mean value Algorithm	Real image	Text or formula	Graphic
<b>Entropy</b>	6.40857691	0.46686699	0.345681019
<b>Standard deviation</b>	8182.70925	113716.6774	115798.8214
<b>Pixel search</b>	0.098722001	0.962050365	0.969939635
<b>Entropy after filter</b>	1.578941532	0.228293925	0.165298649
<b>Standard deviation after filter</b>	66748.95611	117387.096	118039.6604
<b>Pixel search after filter</b>	0.757193056	0.991517188	0.98654

Entropy is the best algorithm that can distinguish real image from different kinds of images efficiently, and is the most fast among three. The shortcoming is that when it meets some real images that have many the same value pixels, entropy will have difficulty to differentiate them efficiently.

Standard deviation can also be used to distinguish real image among different kinds of images. However it is the slowest algorithm, comparing to the other two algorithms, also it is not efficient.

Pixel search is also quick algorithm, and can distinguish real image and non-real-image efficiently. However, when the image is made up of many light pixels, or the image have many pixels that have the same value, then pixel search will have the problem to distinguish.

As to filter technology, we use it to get the more obvious result among real image and text and graphic. However, in practice, it is not as good as we thought. The biggest shortcoming is the CPU time it takes. When using in the entropy algorithm, image filter technology make the result more exacter and can categorize the real image and text and graphic efficiently. To standard deviation, image filter works not as good as not using image filter technology. When using in the pixel search algorithm, the same situation as the standard deviation algorithm happens. So, image filter technology in standard deviation and pixel search just waste CPU time, while could not provide more exacter result.

When categorizing three kinds of images-----real image, text or formula, and graphic, it is better to use entropy-add-filter to get the exacter result.

We were not mean to separate text and formula as mentioned in the head of chapter 4, However, when implement, we found entropy and standard deviation could be used in categorize text and formula. From 4.2.1, the first five images are text, and the following 5 images are formula. Compare their result value in the text and formula, it is not difficult to found that the value of the text is much higher than formula (usually five times large).

## Chapter 5 Conclusion

With the rapid growth of the Web, providing relevant pages of the highest quality to the users based on their queries becomes increasingly difficult. In this project, we mainly deal with different kind of images in the web pages. The most important objective with this project was to identify image and text and graphic, which so far we have achieved to this point.

The diversity of images is huge. However, there are some certain characteristic to each category of images. We can analyze the different characters of different kind of images, and get the different result. Through this method, we can categorize different kinds of images.

We choose three algorithms to categorize different kinds of images in the web. That is: entropy, standard deviation, and pixel search. In real practice, we imply these three algorithms and use it into the real practice to see which one is the best. And find that entropy is the best, based on the effect and the speed. And image filter technology is being used to make difference between real images and text or graphic more obvious.

Our final prototype works well in processing images. However, it is not perfect yet, these algorithms could ensure 100% to categorize different kinds of images and wait for more modification.

We use Harvestman software as our crawler to download and categorize different kinds of images. This function works very well.

# Appendix

## A1 References

- [1] Wayne Niblack. An introduction to digital image processing, 1985
- [2] Wayne Niblack. An introduction to digital image processing, 1985
- [3] Rafael G. Gonzalez, Richard E. Woods. Digital image Processing, 1993
- [4] Rafael G. Gonzalez, Richard E. Woods. Digital image Processing, 1993
- (5) Salvatore D. Morgera, Jihad M. Hallik. A fast algorithm for entropy estimation of grey-level images, IEEE, 1994
- (6) Standard deviation, <http://www.robertniles.com/stats/stdev.shtml>