REPUBLIC OF TÜRKİYE

ALTINBAŞ UNIVERSITY

Institute of Graduate Studies

Information Technologies

KEYWORD BASED SEARCH ENGINE OPTIMIZATION USING BAG OF WORDS ALGORITHM

Mohammed Khudhur Jaafar JUBOORI

Master's Thesis

Supervisor

Asst. Prof. Dr. Sefer KURNAZ

Istanbul, 2023

KEYWORD BASED SEARCH ENGINE OPTIMIZATION USING BAG OF WORDS ALGORITHM

Mohammed Khudhur Jaafar JUBOORI

Information Technologies

Master's Thesis

ALTINBAŞ UNIVERSITY

2023

The thesis titled KEYWORD BASED SEARCH ENGINE OPTIMIZATION USING BAG OF WORDS ALGORITHM prepared by MOHAMMED KHUDHUR JAAFAR JUBOORI and submitted on 7/4/2022 has been **accepted unanimously** for the degree of Master of Science in Information Technologies.

Asst. Prof. Dr. Sefer KURNAZ
Supervisor

Thesis Defense Committee Members:

Asst. Prof. Dr. Sefer KURNAZ	Department of Computer Engineering, Altınbaş University	_____
Asst. Prof. Dr. Abdullahi Abdu IBRAHIM	Department of Computer Engineering, Altınbaş University	_____
Asst. Prof. Dr. Zeynep ALTAN	Department of Computer Engineering, Beykent University	_____

I hereby declare that this thesis meets all format and submission requirements of a Master's thesis.
Submission date of the thesis to Institute of Graduate Studies: ___/___/___

I hereby declare that all information and data presented in this graduation project has been obtained in full accordance with academic rules and ethical conduct. I also declare all unoriginal materials and conclusions have been cited in the text and all references mentioned in the Reference List have been cited in the text, and vice versa as required by the abovementioned rules and conduct.

Mohammed Khudhur Jaafar JUBOORI

Signature

DEDICATION

I would like to express my sincere gratitude to my advisor, Sefer KURNAZ for their invaluable guidance and support throughout my master's program. Their expertise and encouragement helped me to complete this research and write this thesis.

ABSTRACT

KEYWORD BASED SEARCH ENGINE OPTIMIZATION USING BAG OF WORDS ALGORITHM

JUBOORI, Mohammed Khudhur Jaafar
M.Sc,. Information Technologies, Altınbaş University,
Supervisor: Asst. Prof. Dr. Sefer KURNAZ
Date: 04 / 2023
Pages: 76

Text data for analysis is an essential activity in science. The diversity of scientific fields produces very heterogeneous data in terms of type and information provided. Many projects have been carried out in our laboratory to extract semantic elements from images from different backgrounds Document classification or document categorization is a problem in library science, information science and computer science. This task consists of assigning one or more classes or categories to a document. This can be done "manually" (or "intellectually") or through algorithms The main contributions of this thesis is Creating a new way of interacting with the user, much more personalized. the data we collect today allows us to see things that until recently were too big for us to see". data are categorized, that is, if keywords, which indicate the main content of the data, were associated with these data. Scientific articles, for example, when associated with keywords, help in the analysis of the profile of the conference in that year or of a journal issue.in this work we used the bag of words algorithm for keyword extraction from the image and search related content in the database. The BOW features are way better than RGB and HSV when it comes to color based extraction as it is developed as a further iteration to these two. Also the EHD proves to be a better method than Wavelet and other segmentation methods. There are certain anomalies however that arise due to the combination of these two features. Too much of a color in the background takes away the weightage of the centre object. Such problems exist in every CBIR system but with further analysis we can surely work on improvement.

Keywords: AI, BOW, EHD, CBIR, Macine Leraning.

TABLE OF CONTENTS

Pages

LIST OF TABLES

Pages

LIST OF FIGURES

Pages

ABBREVIATIONS

POS : Parts Of Speech

CBIR : Content Based Image Retrieval

HOG : Histograms Of Gradients

GLOH : Gradient Location and Orientation Histogram

CBIR : Content-Based Image Retrieval

QBIC : Query By Image Content

1. INTRODUCTION

1.1 BACKGROUND

Text data for analysis is an essential activity in science. The diversity of scientific fields produces very heterogeneous data in terms of type and information provided. Many projects have been carried out in our laboratory to extract semantic elements from images from different backgrounds, such as images of initials [1], handwritten documents [2], natural images [3], comic strips [4], digital documents [6]. As far as we are concerned, our work focuses mainly on the analysis of visual data, extracted from images. The Bag of Words Pattern is one of the approaches which consists in constructing an image description also called an image signature. Figure 1.1 illustrates part of the process of representing an image as a bag of sight words. It is based on the bag-of-words model from the field of textual document processing. This original model was proposed in 1975 [7]. The description of the images by the visual bags of words makes it possible above all, by this signature, to characterize them then to classify them or to group them in relation to each other. The larger goal is to accomplish image identification and recognition. Lewis [8] show that the bag-of-words model in the context of word processing contains only 5% of relevant information for the classification and/or clustering of textual documents. We thus wondered what was the situation for visual words, and in particular whether it is possible to reduce the size of the bags of visual words while retaining their capacity for description. In their classic version, the bags of visual words of images are vectors of frequencies of the visual words. They are presented in tabular form, with each visual word being a group (cluster) of features. These characteristics are the information contained in the image, for example color, outline, shape, etc. In the image database, the comparison between images then consists of We thus wondered what was the situation for visual words, and in particular whether it is possible to reduce the size of the bags of visual words while retaining their capacity for description. In their classic version, the bags of visual words of images are vectors of frequencies of the visual words. They are presented in tabular form, with each visual word being a group (cluster) of features. These characteristics are the information contained in the image, for example color, outline, shape, etc. In the image database, the comparison between images then consists of We thus wondered what was the situation for visual words, and in particular whether it is possible to reduce the size of the bags of visual words while retaining their

capacity for description. In their classic version, the bags of visual words of images are vectors of frequencies of the visual words. They are presented, with each visual word being a group (cluster) of features. These characteristics are the information contained in the image, for example color, outline, shape, etc. In the image database, the comparison between images then consists of the visual word bags of pictures are frequency vectors of the visual words. They are presented in tabular form, with each visual word being a group (cluster) of features. These characteristics are the information contained in the image, for example color, outline, shape, etc. In the image database, the comparison between images then consists of the visual word bags of pictures are frequency vectors of the visual words. They are presented in tabular, with each visual word being a group (cluster) of features. These characteristics are the information contained in the image, for example color, outline, shape, etc. In the image database, the comparison between images then consists of.

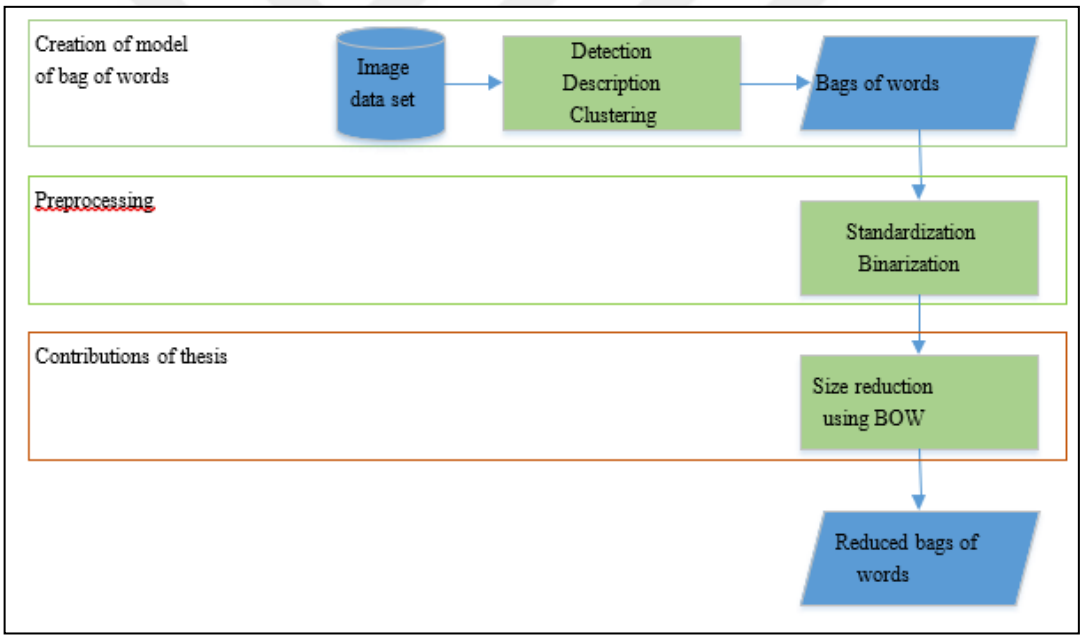

Figure 1.1: Schematic Illustration Generally Presenting the Processing Chain From an Image Database in Order to Obtain the Bags of Visual Words Reduced in Three Stages: Image Signature Extraction, Data Pre-Processing Discrete to Binary Data, Dimension Reduction of Image Signatures.

To compare their sight word vectors. The large size of the sight-word vector makes processing difficult when the sight-word bag model is applied to the large image databases encountered today. For example, the VOC2012 database contains 22,500 natural images of

different objects. Each image can be represented by a vector of 3000 visual words. In this thesis, we seek to reduce the size of these vectors using a method of dimension reduction of the vector of visual words. Many works on dimension reduction exist in several scientific fields such as statistics, computer vision or machine learning. It can be seen as the reduction of redundant and/or irrelevant information in the data description. Many dimension reduction methods are proposed in the literature, depending on the type of method (ie statistical, probabilistic, logical or algebraic) but also on the approach used for the choice of attributes (ie extraction or selection), or on the type of attributes analyzed (ie qualitative/symbolic attributes or quantitative/numerical attributes). These methods are detailed in chapter 1 of the manuscript. In general, we can categorize them into two approaches:

a. attribute extraction methods transform the initial set of attributes

Table 1.1: Illustration Generally Presenting the Representation of Images as Bags of Sight Words. The Images are Taken From the CALTECH256 Image Database.

		Words						
		1	**2**	**3**	**4**	**5**	**6**	**7**
Pictures		7	1	0	2	0	1	6
		1	7	1	3	5	0	1
		1	0	1	6	0	1	1
		2	3	7	2	6	8	3

b. attribute selection methods

The terms visual word or attribute are used equally and without distinction in this manuscript. Attribute extraction methods are very efficient in terms of reducing the number of attributes. However, Fukunaga shows that these methods are quite useful for compressing data, but not very suitable for categorizing or grouping data [7]. Moreover, in some fields,

it is preferable to keep the original attributes for their properties (ie in the field of bioinformatics, more specifically genetic selection linked to genomic expression). Attribute selection methods were therefore chosen to handle this case and have been well studied in the context of machine learning [11]. However, . In other words, these methods are commonly used in supervised learning. In the context of unsupervised learning, where the ground truth is missing, the question of the evaluation of the result of the reduction arises. Most of the dimension reduction methods in the literature rely on statistics, probabilities, and very few on algebraic or logical measures.

Formal concept analysis (FCA) has been applied in fields such as data analysis or data mining. These applications popularized the concept lattice structure which is a logical/algebraic approach. The concept lattice is constructed from an array of binary data (images × visual words) called context (objects × attributes). A concept lattice is a graph where each node is called a formal concept, which contains a maximal set of objects and their common attributes. Two formal concepts are linked by a relation of inclusion between objects and of inverse inclusion between attributes, called relation of specialization/generalization. In the most pessimistic case, this graph can have an exponential size. The rapid rise in computing power of computers as well as the democratization of these powers nowadays allows us to develop applications using this type of graphs. The concept lattice has been used in the context of feature extraction [12] by generating numeric attributes from binary attributes. It has also been applied in one step of an attribute selection method [13]. However, to our knowledge at the beginning of this thesis, it has never been applied as an attribute selection method. However, it seems to have interesting properties for its exploitation in this field. In 2014, Ikeda and Yamamoto [14] applied Formal Concept Analysis for the prediction of classes in the test base simultaneously with the selection of local attributes. The truss structure has been the subject of abundant theoretical research since the end of the 19th century. The first reference work on lattice theory gives an algebraic definition of it from the lower bound and upper bound operations In this graph, the author makes the distinction between the elements not corresponding to an upper limit (sup-irreducible) or a lower limit (inf-irreducible). The irreducible elements are gathered in the form of a binary table, the table of irreducible, the sup-irreducible in rows and the inf-irreducible in columns. The structure of the lattice and the object/attribute combinations are maintained by keeping only the attributes corresponding to the inf-

irreducibles. This is a logical reduction. In this manuscript, we propose a reduction algorithm based on this principle of selection of irreducible attributes. In this manuscript, we are interested in attribute reduction on the visual word bag model of image databases.

1.2 THESIS AIMS

The objective of this thesis is to propose and evaluate a search method to reduce the number of attributes (visual words) while keeping their ability to distinguish images. For this, we will focus on:

a. the reduction of attributes while keeping the inf-irreducibles of the lattice, which is based on the fundamental theorem of lattice theory [7], and which uses the precedence graph to perform the reduction. This algorithm is tested on several image databases;

b. attribute reduction by removing “similar” irreducible attributes, which relies on an approximate extension of the precedence graph, used similarly to the previous algorithm. This fuzzy reduction algorithm reduces the attributes by accepting a drop in performance compared to the previous reduction algorithm, by experimenting with this algorithm on a basis of the selection domain of the attributes.

1.3 PROBLEM STATEMENT

Document classification or document categorization is a problem in library science, information science and computer science. This task consists of assigning one or more classes or categories to a document. This can be done "manually" (or "intellectually") or through algorithms. Intellectual classification of documents has been the main problem of library science, while classification of documents through an algorithmic approach is the competence of information science and computer science. [5] In the context of information science, which is the part that is addressed in this work, as we have today the presence of technology almost constant in everything we use, be it the computer, tablets or even our cell phone, everything this generating a large volume of data, thus creating the need to extract information from these data in order to generate new knowledge from them. The advancement of technology has allowed the storage of these large volumes of data and the analysis of this data can be useful for different organizations. Facebook, for example, processes more than 500 TB of data daily [2], data which, when processed, can generate useful information, and which can be used in different ways, such as, for example, in order

to generate advertisements. directly to a specific user, based on the searches carried out by the user on his cell phone.

1.4 CONTRIBUTIONS

The main contributions of this thesis is Creating a new way of interacting with the user, much more personalized. the data we collect today allows us to see things that until recently were too big for us to see". data are categorized, that is, if keywords, which indicate the main content of the data, were associated with these data. Scientific articles, for example, when associated with keywords, help in the analysis of the profile of the conference in that year or of a journal issue. The association of keywords also allows a more efficient search for an article of interest, since keywords indicate the main idea of an article and thus allow to easily align the researcher's interest to that article. keywords are often maintained by the authors themselves. And most of the documents available on the Web still don't have keywords assigned to them. The manual association of keywords is impracticable, considering the large volume of data currently available. Therefore, there is a demand for approaches that can automatically discover the keywords that best describe/represent the subject of a given document, and then associate these words with them.

Considering the pointed problem of how to automatically extract the keywords associated with a certain document, the objective of this course conclusion work is precisely to propose an automatic method, for the solution in an efficient way, of this problem of text classification. Using concepts studied throughout the Information Systems course, to extract keywords from publications on a website that transmit the internal content of these publications, where the relevance of these words to the text is made through statistical analysis. To evaluate this approach, a real scenario was used, on the website, www.preadly.com. Where the organization that maintains the site contacted us with a demand, indicating that it has today, a difficulty in categorizing its publications, due to the diversity of subjects approached by them, in a way that the keywords really transmit the content present in the text, without that this is done manually. A quantitative and qualitative analysis was carried out, verifying that the approach proposed in this work is promising.

1.5 THESIS STRUCTURE

This work is structured in 5 chapters and, in addition to this introduction, it will be developed as follows:

a. Chapter 2: introduces some of the state-of-the-art method related to our work.
b. Chapter 3: introduces the materials that are fundamental to make the best use of the information treated in this work.
c. Chapter 4: deals with the practical scenario, Data and codes in which this work was applied.
d. Chapter 5: deals with results of carrying out the proposed work using everything that has been studied.
e. Chapter 6: Conclusions – Gathers the final considerations, points out the contributions of the research and suggests possibilities for further deepening.

2. LITERATURE REVIEW

Text classification (TC), a topic of basic relevance brought on by recent breakthroughs in text mining and natural language processing, is seeing a renaissance of attention (NLP). Although the term "text classification" can be used to refer to a wide variety of specific methodologies that are utilized in many domains, they all have the same objective, which is to affix a predetermined label to a specific piece of input text. This can be done in a number of different ways. There are a lot of distinct approaches one can take to achieve this goal.

Some of the first uses of TC were the retrieval of information, the tagging of themes, the analysis of feelings, and the classification of news. Although TC was initially developed for classification, its applications have now been expanded to include more intricate activities like as replying to extracted queries and delivering summaries. TC's original purpose was categorization. In this particular setting, the clearer idea of a "label" has been swapped out for a list of potential alternatives (e.g., an answer or a sentence to include in a summary).

The current rate at which new textual content is being produced substantially exceeds the capacity of manual methods to deal with these responsibilities. As a consequence, TC strategies are not only advantageous but also essential. As a consequence of this, it is of the utmost need to work toward the creation of TC systems that are reliable and objective.

2.1 TEXT CATAGORIZATION

The NLP research community has developed a number of different standard definitions for TC jobs. These concepts are being used as evaluation criteria for the various solutions that have been suggested. Within this section, we will talk about the significant representations while, for the most part, adhering to the taxonomy that Li and his colleagues constructed.

The technique of discovering and classifying the underlying emotional tone of a piece of writing in order to draw inferences about the meaning the author intended to express is referred to as "sentiment analysis" (SA) in the discipline of linguistics.

For example, the procedure of identifying the major ideas contained in a piece of writing is referred to as "topic labeling," which is an abbreviation for the full phrase (its "topics").

The process of categorizing news pieces in accordance with broad topics or reader interests is referred to as news classification (NC), which is also an alternate name for news categorization.

The process of finding an appropriate response to a question by selecting a suitable response from a broader pool of potential solutions is referred to as "question answering" (QA), which is an abbreviation for the phrase "question answering" (usually extracted from a context document). When it comes to framing work of this sort, binary classification and multiclass classification are two methods that are widely applied as different approaches.

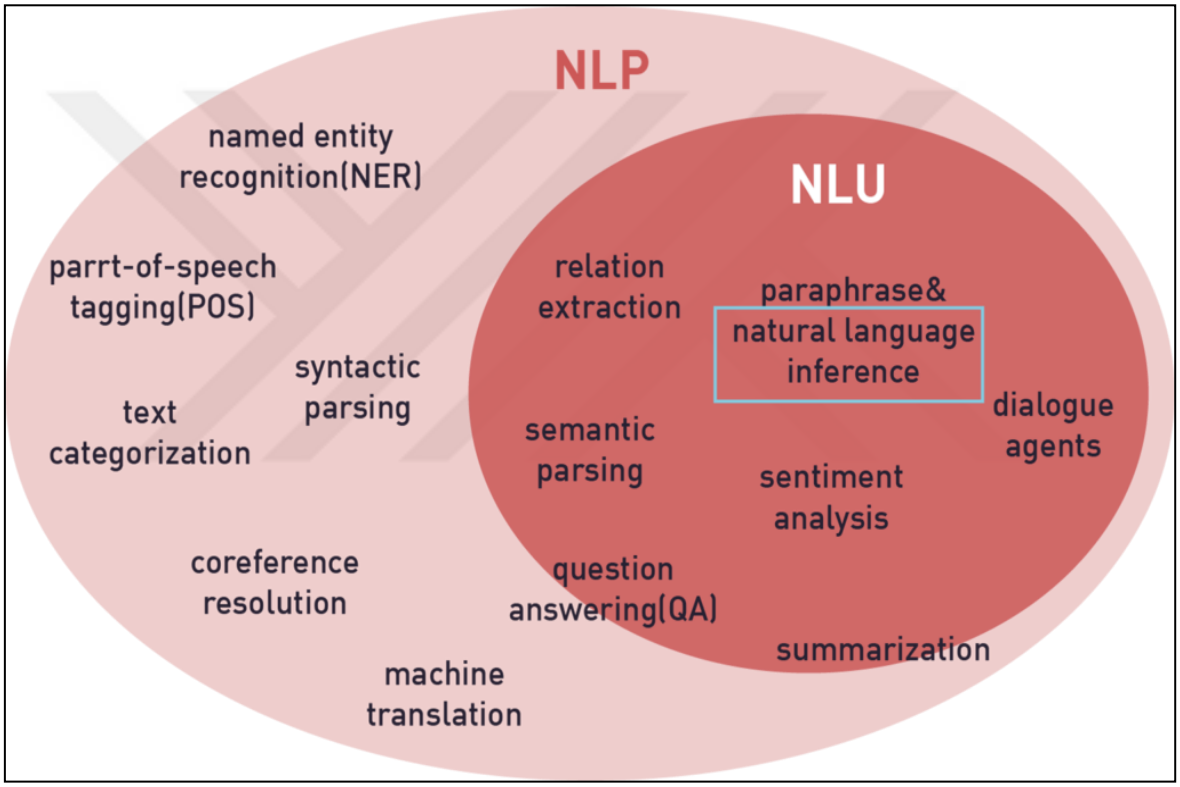

Figure 2.1: NLI and NLP Structure.

An example of natural language inference (NLI) is to determine whether or not there is a logical connection between two assertions (classify whether the entailment happens in both directions or neither), and named entity recognition is the process of locating and classifying named entities within unstructured text (NER).

In syntactic parsing, some of the tasks that need to be completed include the labeling of semantic roles, the identification of speech dependencies, and the tagging of parts of speech (PoS). These are just some examples (SP).

It is of the utmost importance to keep in mind that the formulations that have been offered here are general, and that particular tasks will seem somewhat different when implemented within the context of their respective domains.

2.2 TEXT REPRESENTATION

The projection of text characteristics into a certain feature space is one of the most crucial stages of any type of TC process. Because it is not organized in any specific fashion from a computational standpoint, it must first be subjected to a variety of different operations before it can be processed by a computer. This is because it is not structured in any particular way. In light of the fact that there is no one tactic that can be summed up in the phrase "silver bullet," the preprocessing stage of the classification pipeline needs to take into consideration the models that will be utilized in the subsequent stages.

In the first place, traditional techniques are predominantly based on a manual feature engineering process, which requires careful handling and an in-depth acquaintance with the topic at hand. In order to be successful with conventional methods, one must have a high level of expertise in the field. On the other hand, later systems based on deep learning can be distinguished from those that came before them by the automatic extraction of distinguishing traits. As we will see in a moment, preprocessing is still a crucial component of these methods, despite the fact that its application could be varied depending on the assumptions that are made by these techniques.

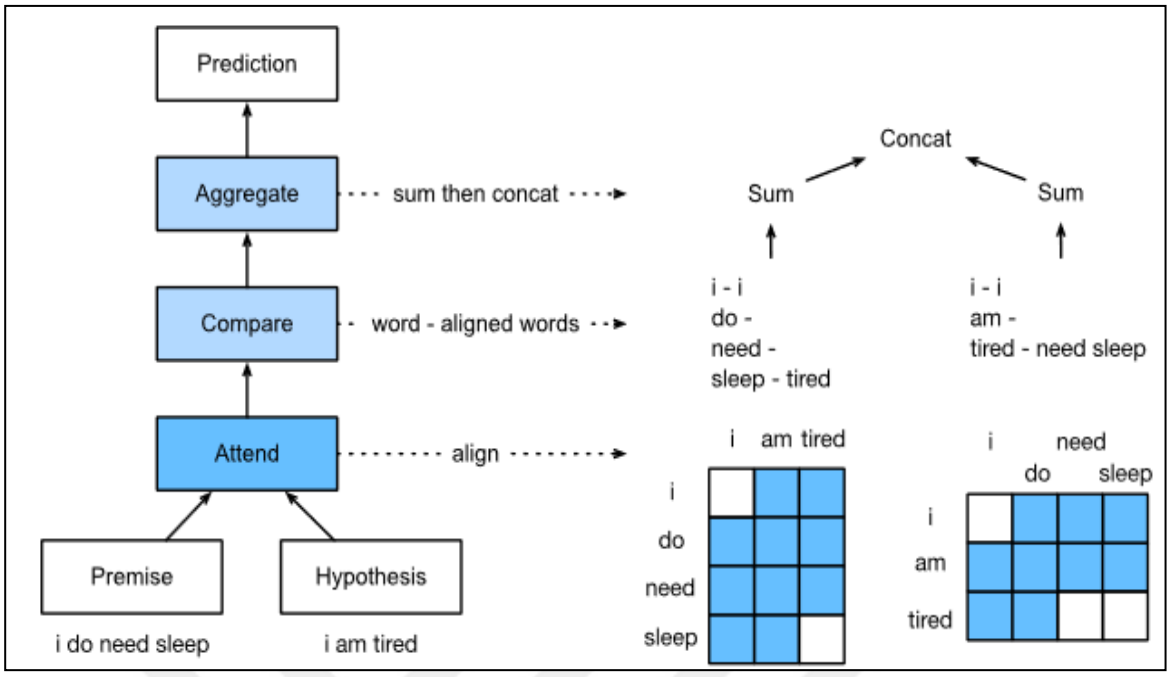

Figure 2.2: Diagram of Text Representation Workflow.

Strategies for the Categorization of Text in a More General Sense 1.3.

The ideas of "shallow" and "deep" machine learning approaches, two of the most important types used in this study, are discussed in this section. These are two of the most important types.

Methods that Make Acquiring Knowledge Easily and Quickly Available.

The methods that were utilized in the past are referred to as "shallow learning" on occasion, and there is a sound rationale behind this term. When we use this phrase, we are referring to the canonical approaches to machine learning that have been around for many years. In light of the fact that there is not a universal agreement over this definition, we would like to make it very clear that when we use this phrase, we are referring to those canonical approaches. That is to say, all of the techniques that have been utilized in the past to make data projections, such as linear regression and logistic regression, are able to be grouped together into this category because they share a common basis. This category includes linear regression and logistic regression. The exception to this rule is represented by neural networks. In addition, this category contains neural networks that are known as "shallow." These networks have only zero to two hidden layers and serve as a bridge between the methods described above and the methods that are based on deep learning. It is possible to categorize shallow neural networks as either completely connected or partially connected.

In terms of precision and consistency, rule-based methods have been outperformed by shallow learning systems. This is because rule-based methods came before shallow learning systems. In many real-world applications, the use of shallow learning methodologies is still common, particularly when serving as dependable starting points. This is especially true in situations when the information being learned is rather superficial. Despite the fact that they are inefficient when working with huge datasets, shallow strategies come into their own when there is a shortage of resources. This is the case even though they are not ideal. These more conventional approaches usually involve an expensive period of feature engineering, the precise cost of which is determined by the level of complexity of the domain that is being taken into consideration. This cost can be significant from a computational standpoint; however, getting the domain knowledge essential for the successful usage of relevant feature extraction algorithms can be more difficult in actual practice. This cost can be significant from a computational standpoint.

2.3 DEEP LEARNING BASED APPROACHES

One of the many subfields of artificial intelligence that has been impacted by the development of deep learning models is text classification. Due to the fact that these methods may duplicate delicate qualities without the need for hand engineering them, the amount of specialized experience that must be given to their implementation has been greatly decreased. Because of this, these methods have gained a lot of favor in recent years. Instead, a considerable amount of effort has been spent into the construction of neural network architectures that are capable of efficiently extracting representations for textual units. This work has taken up a major portion of the available time. Recent advancements in this sector have shown to be particularly fruitful, giving rise to representations that are useful both semantically and contextually. Because it is able to make advantage of a document's underlying linguistic structure, automatic feature extraction is very useful when it comes to modeling data based on textual sources because of this ability. Because of this capacity, selecting it is a wise choice. Once we have a firm understanding of the language, this structure may appear natural to us, but it is frequently beyond the capacity of a machine to comprehend it.

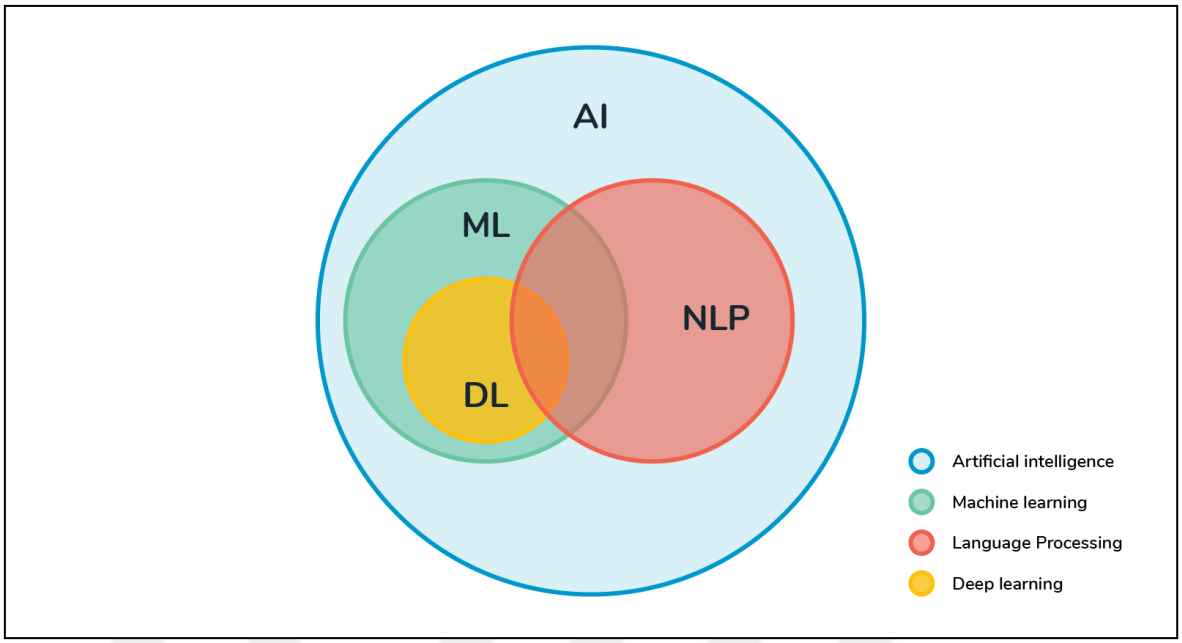

Figure 2.3: Text Mining Using Deep Learning as Part of NLP.

Recent research publications have offered an in-depth examination of a variety of text classification methods. In particular, we refer to the work that has been done by Li et al. [1], which offers a comprehensive analysis of models from a range of depths at varying angles. Kowsari et al. [2] conduct a substantial amount of research on various preprocessing techniques, such as the extraction of features and the reduction of dimensionality, for the purpose of their literature review. In contrast, the study that was carried out by Minaee et al. [3] was solely devoted to carrying out an exhaustive analysis of deep methods. This is the case despite the fact that the researchers did incorporate quantitative findings concerning traditional procedures into their experimental performance analysis.

This undertaking intends to widen the scope of existing text classification surveys by providing an in-depth description of the design process for a classifier that can be applied to textual data. The goal of this attempt is to make existing text classification surveys more comprehensive. In light of this, we provide an in-depth investigation of the primary data preparation procedures that are utilized in conjunction with text classification algorithms. Even though these phases of the TC pipeline are frequently disregarded, it is vital for us to recognize both their role and the reasoning behind any judgements that are made concerning them if we are going to devise a strategy that can be relied upon for carrying out this undertaking. After that, we provide a detailed benchmark of approaches that are deemed to be state-of-the-art in a number of subtasks, as well as an overview of significant English TC

datasets. Lastly, we conclude with some conclusions and recommendations. In addition, we discuss how to reproduce the results that were discovered in two newly synthesized multilabel TC datasets, and we publish the results that were discovered in those datasets. These results may be found in the previous sentence. We believe that this is an essential addition because the subtasks that they cover, specifically multilabel TL and NC, are not well covered by other contributors to this task.

The following is a rundown of the findings that are considered to be the most important ones from this investigation:
We provide an overview of quantitative results for the most important methods in the literature, as well as for the datasets that we presented, along with an analysis of evaluation metrics; we showcase a variety of datasets that are written in the English language; and we provide the code for synthesizing two multilabel classification datasets [17].

The remaining aspects of the survey are laid up in the manner that is displayed above. In the following section, we will discuss the preparation stages that are involved in TC pipelines. This part includes not only the usual suspects but also the tokenization approaches that are used the most extensively for cutting-edge, deep-learning-based models. These methods are described in detail in this section. In Part 3, we will explore the methods that are used to project preprocessed data into feature space. We will also discuss the word embeddings that are based on shallow neural networks and will be described in this section. In the following sections, we will first present a high-level review of generic classifiers that are often used with such data and then we will present a more in-depth description of the current landscape of deep learning-based classification algorithms. we will present a high-level review of generic classifiers that are often used with such data. will present a more in-depth description of the current landscape As we are ready to move on to the experimental portion of the survey, we are going to move on to Section 6, where we will show significant datasets (including two that have just been synthesized) and quantitative results.

If doing so is permitted by the relevant legislation, the experiment code and data sets that were utilized in this inquiry can be acquired at https://gitlab.com/distration/dsi-nlp-publib (accessed on 28 December 2021).

When working with natural language tasks like TC, which require unstructured text as input data, preprocessing is required. This is because these activities cannot function without it.

Textual information, which does not have a numerical representation that is inherent to it, must first be projected into an appropriate feature space before it can be fed into any classifier. This is because textual information does not have a number representation that is intrinsic to it. This is a crucial difference to establish in light of the fact that other types of data, such as photographs and time series, are also taken into consideration. It is vital to carry out preprocessing activities because these actions build the framework for subsequent operations such as the extraction of features and the classification of the data. It is also essential to execute preprocessing activities.

In the following, we will discuss the many preparation procedures that are incorporated into TC techniques. There are numerous of these procedures. In order to get a good start on understanding manual methods for feature extraction, we are going to look over the most typical ways that are currently being used. This will allow us to acquire a better grasp of manual methods In its stead, the second half of this section will discuss the preprocessing processes that are unique to deep models. These activities are described below. This survey aims to shed light on the ways in which state-of-the-art models differ from their predecessors and incorporate unique strategies for expressing textual data. The purpose of this survey is to shed light on the ways in which state-of-the-art models differ from their predecessors. By doing this, our objective is to shed light on the ways in which our goal is to shed light on the ways in which our goal is to shed light on the ways in which our goal is to shed light on the 2. 1 The Preprocessing Steps That Are Employed Most Typically

Preprocessing activities clean and standardize the input data in order to improve the outcomes of the feature extraction stage. These preprocessing operations can either be generated manually, as they were in older systems, or automatically taught. This is achieved by increasing the quality of the results produced during the feature extraction stage (in deep learning methods). This article provides a comprehensive overview of a selection of the preprocessing methods that are now in widespread use.

2.3.1 Tokenization

Tokenization is the initial and most fundamental stage of the process of getting the text ready to be used. The granularity of the textual data that we analyze and generate is determined, in large part, by this method, which may be summed up as "splitting a stream of text into smaller bits" (historically called tokens). The vast majority of natural language processing

models focused on words up until relatively recently; however, recently developed methods have begun to break down text into a broader variety of smaller components (such as character n-grams or even more maximal forms of decomposition, such as underlying bytes [4]).

It wasn't always the case that tokenization [5] is a straightforward procedure, despite the fact that this is now often understood to be the case. [Here's a good example:] Conventional methods involve the application of rule-based strategies such as the employment of white space, punctuation, and contractions, in addition to a number of other components. It cannot be denied that ever more intricate knowledge-based systems have been devised, all of which continue to rely on linguistic ideas. Even though tokens in these systems no longer correspond to the conventional concept of a typographic unit, recent progress has turned significantly toward data-driven tokenizers, which produce better results. This is despite the fact that tokens in these systems no longer correspond to the traditional idea of a typographic unit. Despite the fact that data-driven tokenizers provide better results, this is still the case today. The term "tokenization" is currently being used to refer to the act of dividing a sentence into components that, most importantly, do not have any grammatical justification or explanation. This usage of the term was adopted very recently.

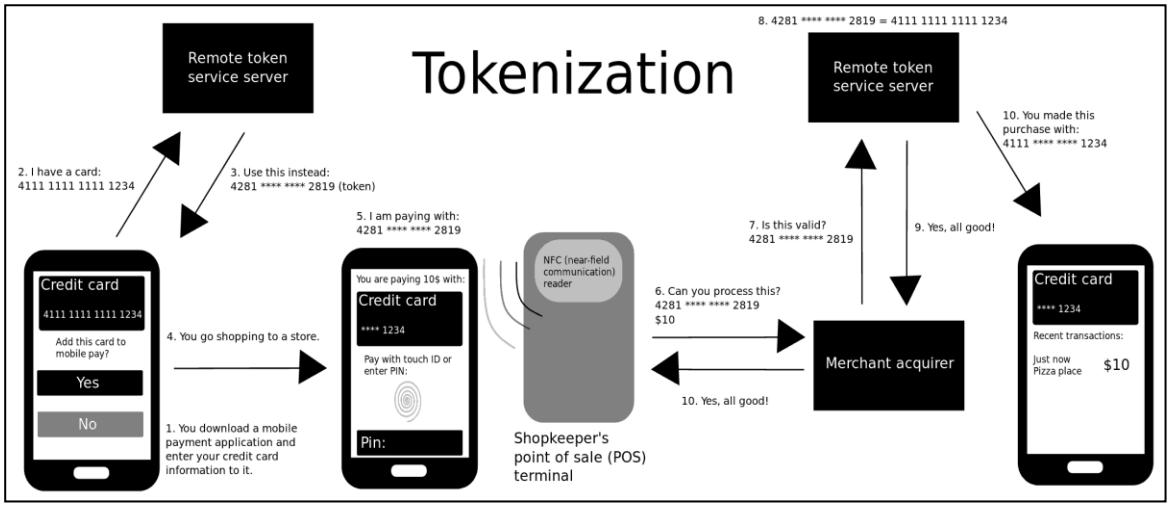

Figure 2.4: Process of Tokenization.

This is the procedure that the phrase is attempting to explain in its fullness. Traditional approaches are usually referred to as "pre-tokenizers" since they continue to use conventional tokenizers as the first stage in their process. This is the reason why traditional tokenizers are known as "tokenizers."

We endorse the work that Mielke et al.[6] have done because it provides a detailed historical assessment of the development of tokenizers in recent years. A deeper dive into this topic is outside the scope of our study, thus we will not be discussing it further here. [Further citation is required] Even though the standard technique and the tokenization approach were both developed at the same time, the standard approach is generally believed to have been the one that was developed first. Tokenization, on the other hand, is tackled in deep learning with techniques that are more modern and advanced, particularly the elimination of stopwords and the reduction of noise, as will be demonstrated in Section 2.2.

It is conceivable for the tokens that are produced as a result of the tokenization process to have information that is either misleading or not essential included in their collection. It is essential to clear the text of any superfluous symbols, letters with accents, and other types of noise that may be present. Words that are used frequently but do not add anything to the meaning of the phrase are known as stopwords [7]. Filler words are another name for these words. The utilization of lowercase letters, the repair of incorrectly spelt words, and the standardization of slang terminology and abbreviations are a few more normalizing strategies that may assist in reducing the total amount of distinct characteristics.

On the other hand, these processes have the potential to result in ambiguity in meaning (in English, "US" might be an acronym for "United States," but lowercasing it would render it indistinguishable from "us" as a pronoun) [2]. [Case in point:] In addition, the elimination of stopwords and several other textual components can be counterproductive to more modern ways of thinking. Because models of deep learning are taught to comprehend natural language, it is highly likely that the syntax of a phrase plays a significant role in the model's ability to comprehend the meaning of the text that it is being presented with. This is because natural language is the most complex form of language. In light of this, participating in activities that are destructive ought to be done with the utmost care.

Second, we are going to have to move forward with the textual standardization (2.1.3).

On the other hand, it is possible that traditional methods would benefit from further simplification of the feature space. Because of this, they are unable to tell the difference between two different inflections of the same word (such as child and children), and as a result, they will treat both of these instances as separate bits of information. As a result of this, it could be helpful to simplify nomenclature by restricting inflections to a single

standard form. This can be accomplished through lemmatization, which is the process of deriving the lemma or canonical form of a word [9], or through stemming, which is the process of generating the root form of a word [8]. Both of these processes are described in more detail below. However, despite the fact that these processes have the potential to make the categorization more effective as a whole, they are not devoid of the complexities that are typically associated with such endeavors. To underscore this point, it's possible that words with distinct meanings share the same root or lemma, which means that they can be interchanged due to their common linguistic building blocks. It is essential to keep in mind that the utilization of these strategies will facilitate the acquisition of new vocabularies.

In addition to the advantageous approaches that have been stated above, a TC algorithm might also benefit from a variety of other beneficial methods. One illustration of this is the practice of labeling words according to the parts of speech that they belong to. A group of words that share similar grammatical functions is referred to as a "part of speech." Examples of parts of speech include nouns, verbs, and adjectives. When it might be advantageous to restrict the vocabulary to one or a subset of these components of speech, this preprocessing step is often included in jobs; when this occurs, the step is sometimes implemented (for example, one might be interested in only nouns and adjectives). It is not difficult to see how incorrect classifications could lead to less noteworthy results from any models that use this strategy because tagging points of sale is, at its core, a classification activity. This kind of mistake has the potential to either allow statements that are not wanted to be heard or prohibit statements that are necessary from being heard.

2.3.2 Pre Processing

As was said before, preparation steps must never be carried out before giving careful thought to the applicable technique. In more modern models, which are based on the architecture of deep neural networks, the old methods of removing accents and apostrophes from letters and transforming them to lowercase are also included. This is done in addition to changing the letters to lowercase. To further ensure that the model receives input samples of a consistent size, the tokenized documents are trimmed or padded to a certain number of tokens. This helps to further ensure that the model receives input samples of the same size. This assists in ensuring that the model receives input samples of a consistent size (i.e., with the same number of tokens).

On the other hand, the primary concentration has been placed on the improvement of text tokenization. Tokenization techniques of the current day need to discover a way to establish a middle ground between a vocabulary that is too small to effectively portray nuance and a vocabulary that is too large to be maintained in memory. As a result of this, a number of different strategies have been considered; however, in this part of the article, we will focus on the most important of these strategies. To continue along this line of thought, a number of cutting-edge tokenizers add normalization processes into their operations [6]. There is a connection between this and the prior point.

2.3.3 DNN Based Methods

It is the job of a deep model to supply a vectorial representation for each token that is contained inside the training data. An embedding matrix is a specific kind of data structure that is used to retain previously learned vectors and to create relationships between well-known vocabulary items and the embeddings that correspond to them. Because of this, the size of the matrix is proportional to the number of vocabulary items as well as the dimension of the embedding. This is true regardless of the embedding. Due to the restricted amount of time and memory that is available for the task, processing arbitrary big vocabularies is not practical. These characteristics encourage research and development of a wide variety of tokenization algorithms with the intention of reducing the amount of vocabulary while simultaneously lowering the number of terms that are not well represented. Tokenization is an approach to reduce the amount of vocabulary while simultaneously lowering the number of terms that are not well represented. The most significant shortcoming of word-level models is their inability to account for some words, which we shall refer to as "out-of-vocabulary" (OOV) terms. In the course of testing, they generate "unknown tokens," which is a result that is inappropriate for the vast majority of NLP activities.

Pre-tokenizers interpret derivations and inflections as independent words (such as "snowboarding" and "snowboard"), which results in the model learning numerous encodings since the model learns a representation for each word in the training corpus. This causes the model to learn multiple encodings. This problem could result in requirements that are impossible to achieve in terms of space as well as time because the number of learnable parameters of such models is directly reliant on the total number of different words that are present in the corpus. If, on the other hand, we tokenize at the character level, then our

vocabulary would be comprised of all of the individual characters that are present in the corpus, which is obviously not a very huge number, at least not for alphabetic languages. If we tokenize at the token level, then our vocabulary would be comprised of all of the individual tokens. Tokenizing words at the sentence level is easier than using this technique, which makes it easier for the model to learn compact token representations. Nevertheless, this strategy is still substantially more difficult. Finding a meaningful representation for the letter "s" is a task that is significantly more difficult than, for instance, finding a meaningful symbol for the word "snow."

As a result of the inadequacy of these two tactics, the majority of models today use hybrid strategies that partition words into more manageable chunks. This is done in order to make the words easier to work with. According to the standard recommendation, the meanings of common terms should not be broken down into their component parts, while the meanings of less common phrases should be broken down in this manner.

3. MATERIALS AND METHODS

3.1 INTRODUCTION

The image is one of the most important means of communication used in search engines. Therefore, image processing is the set of methods and techniques operating in order to improve the visual aspect of the image and to extract information deemed relevant.

In recent decades, the field of research through multimedia content has experienced a real enthusiasm due to new communication techniques leading to new modes of exploration and transmission of knowledge made available to all, which has given rise to many works.

Currently the management of large volumes of data is a major problem, particularly in image databases, in many areas. For example, in the field of photography of works of art There are already solutions based on keywords, but not on visual content, which requires a search by content to allow users to do searches that cannot be expressed in the form of key words, such as a visual atmosphere, or any impression that cannot find words.

In image retrieval, we can also cite copy detection methods based on a content-based image recognition scheme. The idea is to provide image owners with tools to test whether an image comes from their stock. , and this on visual bases only. The purpose of such a system is then to be able to establish similarity links between images from the Web and those of a collection owned by an owner and, in the event that such a resemblance exists, to find what is the collection image that was copied. In such a context, content-based image recognition systems are mainly subject to three constraints: robustness of the description scheme, search efficiency and search speed.

This research theme is often called "content-based image retrieval" (Content based image retrieval CBIR). The CBIRs make it possible to find, within an image database, the images most visually similar to a request image given as an example. For this, they use image analysis methods to automatically describe the visual content of images. This involves extracting high-dimensional digital vectors called descriptors from the images and associating them with a measure of similarity.

This chapter includes a state of the art concerning the general principle of an image search system by content. To search for an image, the principle is to give an automatic

representation of image and that in order to conceive a system of search of images by contents.

There are several techniques for extracting images by content, these descriptors are classified into two types (global descriptors and local descriptors). To introduce this, a passage is obligatory towards the different ways of characterizing images in digital form.

3.2 DIGITAL IMAGE PROCESSING

Digital image processing refers to all the techniques used to modify a digital image with the aim of improving its quality and extracting information from it. The processing of digital images is a discipline which has developed rapidly thanks to the significant developments in the fields of multimedia.

Image processing is based in particular on mathematics related to information, these treatments make it possible to store images more easily and improve their quality.

3.2.1 Image Types

There are two types of digital images: the pixel image and the vector image.

a. The image in pixels

The image in pixels (Bitmap) is in the form of a grid with two axes, X and Y, made up of a multitude of points: called pixels. Behind each pixel is a color swatch.

The definition (or precision) of a raster image thus depends on the number of pixels that make it up. The more there are, the sharper the image will be: this is the resolution (Figure 3.1).

b. The vector images

The vector image comes from geometric formulas, that is to say a set of points connected by line segments, polygons, arcs of circles, and curves forming a path. It is the processor that will "translate" these shapes into information that can be interpreted by the graphics card.

The interest with this type of images is the possibility of enlarging a vector image without losing the initial quality. This type of image is used to represent simple shapes (Figure 3.2).

Figure 3.1: Example of Pixel Image.

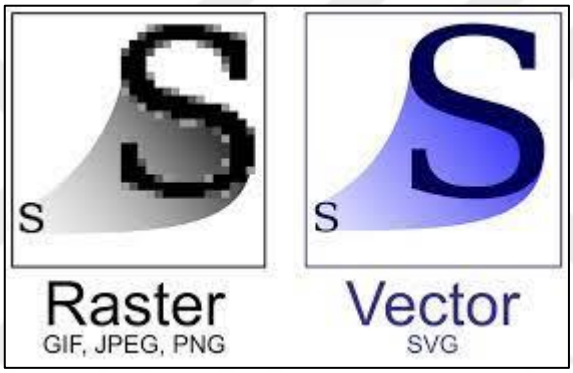

Figure 3.2: Example of A Vector Image.

3.2.2 Advantages and Disadvantages of the Two Types of Images

Pixel images are suitable for the world of photography. Of course, they can represent any other type of image such as drawings, diagrams, maps, etc. However, this type of image suffers from the "pixelation phenomenon" which means that the lower the quality of the photo, the smaller squares of pixels appear, but on the other hand, the higher the quality of the image, the more the file is bulky.

Parts of the picture. The degradation of a part of the image does not lead to the total degradation of the image.

But spatial modifications, which are simple in principle, involve significant computing time and present certain problems: loss of information by size reduction, stair-stepping effects, etc.

Conversely, vector images are rather light. They also have the property of being able to be enlarged without limit and without pixelation, each line and each shape being dynamically recalculated.
On the other hand, they are rather intended for the creation of relatively simple renderings such as diagrams, diagrams, etc. Contrary to pixel images, the spatial modifications of the image are relatively flexible because they consist of geometric operations which do not lead to the loss of information.

The disadvantage of this type of images is that they can only represent simple shapes. They are therefore not usable for photography, in particular to obtain realistic photos.

3.3 EXTRACTION AND DESCRIPTION OF IMAGE FEATURES

Describing the content of images is an essential step in a content-based image retrieval system, because the performance of CBIRs largely depends on the choice of descriptors used and the techniques associated with their extraction. A descriptor is defined as the knowledge used to characterize the information contained in the images. This step makes it possible to provide a representation of the content of the image also called signature of the image.

Many descriptors are used in search systems to describe images. Two types of descriptors can be distinguished: global descriptors and local descriptors. Global descriptors are obtained from the entire image. On the other hand, the local descriptors characterize a set of significant local regions of the image which are rich in information (figure 3.3). Recent work on content-based image indexing and retrieval systems most often focuses on extracting local descriptors that are more efficient for content-based image retrieval. Hotspots are image regions rich in local information content and stable under affine transformations and illumination variations.

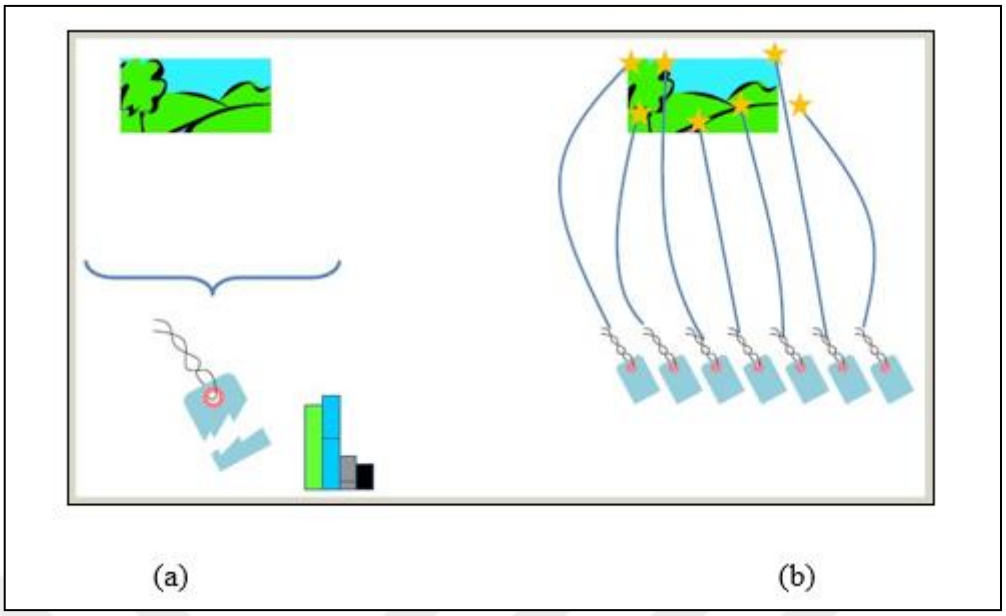

Figure 3.3: Image (A) Represents A Global Descriptor, and Image (B) Re-Presents A Local Descriptor.

3.4 IMAGE DESCRIPTORS

To describe image (a) of figure 3.3, low-level descriptors are used, also called feature vectors, such as color, texture and shape...

a. Shape: modeling techniques can be classified into two categories. The contour approach describes a region by means of the pixels located on its contour. The region approach considers a region with respect to the characteristics of the pixels that this region contains.

b. Color: the color is generally defined by means of digital triplets making it possible to code the intensity of these components. A distinction is made between color spaces defined according to properties such as RGB (Red, Green, Blue), and those based on human perception of colors such as HSV (Hue, Saturation, Value). To model the distribution of colors, we generally use a histogram indicating the intensity of a color on the abscissa, and the number of pixels on the ordinate.

c. Texture: a texture can be characterized by the attributes of contrast, regularity and periodicity of the pattern. In the context of search by content, it makes it possible to distinguish areas of similar colors, but of different semantics.

For the image (b) the extraction and the description of the regions of interest is a technique increasingly used with success in several fields of computer vision, this technique consists in highlighting zones of this image judged "interesting" for the analysis, i.e. presenting

remarkable local properties. Algorithms for detecting points of interest generally focus on particular points of the contours, selected according to a precise criterion. Thus, the corners (corners) are the points of the image where the outline.

The authors indicate that the majority of the literature assumes that a region/point of interest is equivalent to a corner in the image, or, more generally, a region characterized by an interesting value of the brightness gradient in several directions. In general, a region of interest has the following characteristics:

a. It has a formal mathematical definition,
b. It has a precise position in the image,
c. It is rich in local visual information,
d. It is stable in the face of local and global variations of the image, ie, it retains the same visual information in the event of variation.

To use the points of interest, it is necessary to characterize the region around these points. The characterization of a point of interest is calculated, on the region around this point at a chosen scale. The invariant region is defined as a stable region in an image. This means that if we transform this image with some conditions such as scale, rotation, light, this region is also detected (Figure 3.4).

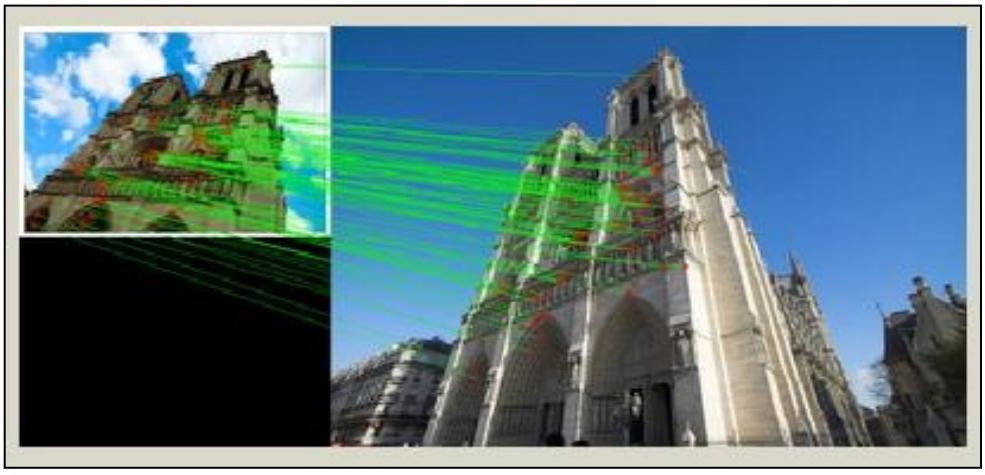

Figure 3.4: Example of Two Images: Detection of Points of Interest (Description Based on Local Information) With A Change of Scale (Different Rotations and in Different Light).

The regions of interest are located via a detector, it is an algorithm or software which has the role of describing a region by a description.

the authors have cataloged different techniques for detecting and describing regions of interest by their robustness to rotations and scale changes.

In order to be as invariant as possible to scale changes, the points of interest are extracted through a global multi-scale analysis of the image, we cite the Harris-Laplace detector, the Fast-Hessian and the Gaussian difference [Low04]. The description part is based on a local exploration of the point of interest in order to represent the characteristics of the neighbourhood.

It is demonstrated that the use of histograms of directed gradients (HOG) makes it possible to obtain good results. Among the many methods using HOGs, we will retain the SIFT (Scale Invariant Feature Transform) [Low04], because both detector and descriptor, it consists of a Gaussian difference (DoG), coupled with R-HOGs.

The detector is based on an approximation of the Gaussian Laplacian [Lin98], in order to perform a multi-scale analysis of the image, it then generates a vector of 128 elements. The creation of the vector is based on the direction factors of the region.

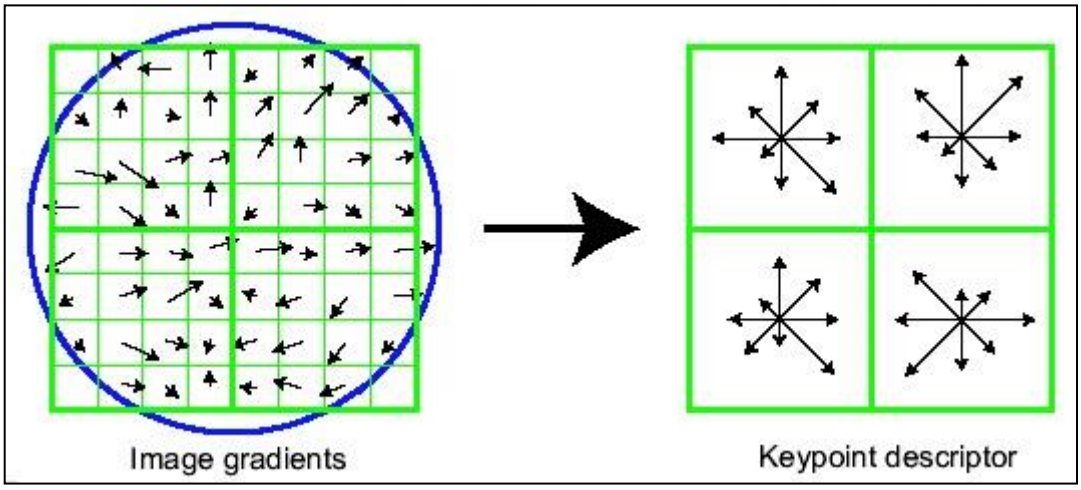

Figure 3.5: Key Point Descriptor" is Then Normalized by A Vector.

In this image, "key point descriptor" is the direction factors of the normalized region on the left SIFT descriptor example.

Figure 3.6: Original Picture.

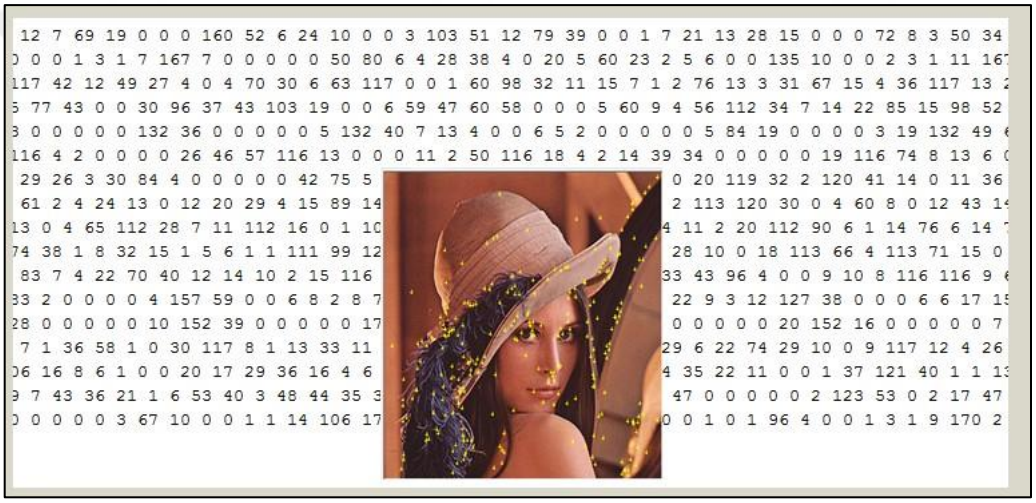

Figure 3.7: The Points of Interest of the Image Using the SIFT

Other descriptors have been proposed in the literature:

a. Shape contextthe principle of this algorithm is to extract from an image the points by describing the contours, and to obtain for each of these points the shape context by determining the relative distribution of the closest points by means of a histogram of distribution of log-polar coordinates.

b. PCA-SIFTis a vector of image gradients in the x and y directions calculated inside the support region. The gradient region is sampled at 39 × 39 positions, generating a vector of dimension 3042. This dimension is reduced to 20 by the method of principal component analysis.

c. Gradient Location and Orientation Histogram (GLOH) is an extension of the SIFT descriptor whose robustness and distinctiveness have been improved.

A comparative study on the evaluation of the performance of descriptors is presented in.

3.5 SYNTHESIS

We detail in what follows, a synthesis on the extraction and the description of images by the global characteristics, and the local characteristics.

3.5.1 Overall Characteristics

The most basic, and historically first, image search scenario by content is that of global search by example. An effective way to involve the user in the content search process is to ask him for an example close to what he is looking for, by choosing an example image, or by providing an image or by drawing an image. query. The search returns images that look like the query image.

a. Limits of the comprehensive approach

As part of the global research, many image descriptors and similarity measures have been proposed, in order to characterize color, texture and shape information.

Color (such as histograms) and texture (such as contour histograms) features are relatively trivial to extract and mostly give good results in image similarity search. However, these characteristics are global information and do not always make it possible to take into account their spatial distribution in the image. However, the search is based on a calculated similarity measure between the target images and the query. Consequently, the result images returned by the system are close to the example, in other words, similar to the request in the sense of the measure adopted.

Figure 3.6 shows the results of a query for “children's portraits”. The database used contains about 6000 varied images. The results correspond to the 12 images closest to the image given as an example. The features extracted from the images relate to color and texture. It can be seen that the semantics of the result images is very different from the semantics of the example. The system returns a single image whose semantics is close to that sought

This means that the global characteristics do not translate the notion of semantics contained in the image. Consequently, such a system only makes it possible to satisfy low-level requests or even to find images presenting certain characteristics.

Figure 3.8: Illustration of the Semantic Gap.

The limitation of these search and navigation methods exploiting the global visual description of the images, is the implicit assumption that the entire content of the image is relevant to the user's need. The image is considered as an atomic visual entity, whereas the human generally perceives an image as a composite entity of objects. The interest of the user may relate to one or more image components or he may wish to ignore the background of an image which he will not consider relevant for his request. It is, therefore, important to allow the user to explicitly designate the relevant components in the images and to carry out the search from them in order to reduce the semantic gap of the global approach.

3.5.2 Local Characteristics

When we talk about partial query, we are interested in finding particular areas of the image. It is a question of breaking away from a global description to move towards a description at the local level. The goal is to get closer to the semantics. In order to extract local characteristics, it is essential to detect the zones of interest of the image, are the salient points of the image, then to calculate in each of these zones a characteristic vector.

The detection of image components in large databases is a difficult problem, especially when it must be automatic. In the literature, for content search, different methods exist to define and extract image components. The points of interest, in an image, correspond to double

discontinuities of the intensity function. These can be caused, as for contours, by discontinuities in the reflectance function or discontinuities in depth. These are for example: corners, T-junctions (Figure 3.9).

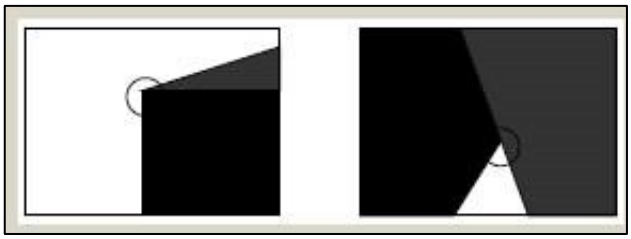

Figure 3.9: Different Types of Points of Interest: Corners, T-Junction.

a. Existing techniques for indexing images by local features:

i. Manual cropping

The manual trimming of the areas of interest of the database images (see [Smi97]) has the advantage of defining the regions according to the user's expectations. However the major disadvantage is the painful intervention of the user who is not reliable to extract the regions in large image databases.

ii. Image blocks

In contrast in terms of computational cost and precision, the systematic subdivision of images into blocks is simple and unsupervised, but very approximate: visual descriptors are calculated on the cells of a constant grid of the image. The subdivision is dependent on the spatial scale of the grid which does not adapt to the content of the image. In uniform areas, the cells will be unnecessarily numerous and will be insignificant in areas with strong local photometric variation. The search methods based on this technique make it possible to compare a combination of several squares in the query image with several squares in the database images. The combinatorial induced by this method is expensive and not precise.

iii. Histogram back projection

the histogram rear projection does not require predefining image areas a priori, as for the points of interest. It is original but little used, it proceeds as follows: from the histogram of a query zone, possibly defined dynamically for each pixel (its quantified color is denoted c), we define an intensity equal to the value of cell c of the quotient histogram between the query histogram and the global histogram of the candidate image. The image of the intensities created is averaged and the places of strong intensities correspond to the most probable position of the request histogram. This method is expensive, since it assumes a

traversal of each image for each histogram color at the time of the request. Smith was inspired by this in VisualSeek to extract regions at indexing time.

iv. Points of interest

have been used for problems of precise mapping of places between two images corresponding to different views of the same scene. The properties of stability and repetitiveness of their detection have motivated their application to content-based image retrieval. They were initially applied to the global search of grayscale images The color points of interest were then used for a search approach by parts of images The characterization of image components by points is effective to find with precision of fine details, but their implementation today remains costly at the level of similarity measurement. Moreover, they do not lend themselves to the characterization of smooth or uniform areas.

b. Point detection methods be classified into three categories:

i. Contour approaches: the idea is to detect edges in an image first.

The points of interest are then extracted along the contours by considering the points of maximum curvature as well as the intersections of contours.

ii. Intensity approaches: the idea this time is to look directly at the intensity function in the images to directly extract the discontinuity points.

iii. Model-based approaches: the points of interest are identified in the image by matching the intensity function with a theoretical model of this function of the points of interest considered.

The approaches of the second category are those generally used. The reasons are: independence with respect to the detection of contours (stability), independence with respect to the type of points of interest (more general methods). e) Regions of interest

The last family of partial image query systems is based on the unsupervised segmentation of database images. Each image is partitioned into a set of photometrically homogeneous areas, the regions, which are individually indexed by statistical visual descriptors.

In a query image, the user can select the region of interest and find the images comprising a similar region. This approach has been used, among others, in Blobworld systems presented

a method for extracting so-called dominant color regions from the image. The algorithm used is a simple extraction of connected components and the only attribute used for segmentation is color. The image is therefore composed of a set of regions characterized by their color, their size and their relative position. The signatures of the entire database are then stored in a hash table.

Other researchers have also proposed the same type of system for searching around the region, using the classic attributes of color, texture and shape to which they add characteristics of size and absolute spatial location.

a. Comparison of techniques

In the manual, block and region approaches, the image components are explicitly defined in the indexing phase (ie “off-line”). On the other hand, the notion of image components in the case of points of interest and the back-projection of primitives only makes sense at the time of the search (ie “on-line”). The dynamic definition of the components at the time of the search offers an interesting flexibility, but entails a very high computational cost of search. Moreover, the geometric characteristics (position, size, shape) are difficult to integrate into these representations.

The partial representation by regions of interest corresponds to a good compromise because it is adaptive to the visual content of the image (unlike blocks), totally automatic (unlike manual clipping) and fast for research (unlike points of interest and back-projection of primitives). It has the advantage of considerably reducing the complexity of the system, and therefore of being suitable for searching in large databases. Moreover, it naturally lends itself to high-level image modeling in which an image is considered as a structured set of objects possessing individual and relative characteristics.

b. Limits of the local approach

In the literature, the most developed approach in image search by regions, corresponds to the search paradigm by region example. The user selects a sample region in an image and the system finds images with a visually similar region.

This paradigm has the following two problems:

i. The first is non-trivial, it concerns the automatic detection of regions in an image database which must be significant for the user. The regions obtained are often

unsatisfactory or too small and therefore too homogeneous to constitute relevant query keys.

ii. The second problem is that of the description and the visual similarity of the regions which must take into account the visual specificity of the regions.

The similarity between two images is measured as a combination of the similarities between all the constituent regions of each image. Although using regions, this approach corresponds to the image-example search paradigm. Indeed, the system returns images whose overall visual appearance is similar to the example image. The user does not have the ability to explicitly designate query regions. Although this approach gives results closer to the semantics of the query, it is insufficient [Fau03].

Hence the interest of introducing other media such as text to have more semantics.

3.5.3 Global or Local Representation

Below is a table summarizing the main advantages and disadvantages of the different representations presented above:

Table 3.1: Comparison Between Different Type of Image Representations.

Representation	**Benefits**	**Disadvantages**
Local	- Some invariances are easy to obtain - Extraction for lower computational cost simpler comparison: - Entity comparison	- Not suitable for finding common parts between two images - Not robust in case of change of
	with a single vector	resolution of images or extraction of sub-images,
Global	More resistance to most transformations robust: statistics dependent on several descriptors More accurate description of an image as a whole	- points or region significant extracted less significant therefore less relevant

3.6 GENERAL ARCHITECTURE OF AN IMAGE INDEXING AND RETRIEVAL SYSTEM

Classically, an image search system by visual content includes an off-line image database indexing phase and an online search phase, which we describe in the next two sections. Fig. 3.8 represents the general architecture of a system for indexing and searching images by content, this system is executed with two stages: the indexing stage and the search stage.

The purpose of the indexing step is to organize and prepare the database. This phase is called off-line, because it is carried out before any search. It includes the following processing:

The extraction of characteristic descriptors from the images, The construction of indexes from descriptors. The goal is to set up techniques allowing to access any descriptor as quickly as possible during the search.

The search step takes a query vector as input and uses an algorithm that takes full advantage of the indexing step, thus limiting the search for similar data.

In other words; during consultation of the database, the user selects an image through a graphical interface. The indexes or signatures of the request are confronted with the indexes of the reference images, thus the system selects and presents to the user the images most similar to the request.

The visual content of base images that is extracted and described is called image signatures. The signatures of the images in the database constitute a database of signatures. To search for images, the user provides a sample image (called a query). Different methods of formulating requests exist, the treatment of these in the system depends on the way in which the information is presented to it. A search pattern specifies how the query is represented. We list below the types of image query methods that can be classified into six categories:

i. Query on a single characteristic: the selection of images is based on a single characteristic, the percentages of different values taken are determined by the user. An example of this type of query is to find images containing 10% red, 30% green and 60% blue.
ii. Simultaneous query on several characteristics: the user request is a combination of several characteristics (color, texture, shape). An example of this type of query is to find the images containing 10% red and 30% green and 60% blue of the tree texture.

iii. Feature Location Query: the user specifies the different values taken by the characteristics and the location of these characteristics in the image. An example of this type of queries is to find images where 25% of the color red is located to the left of the image.

iv. Image search by sketch (query by sketch): the user draws his query using a graphical interface to search for images that look like him in the database.

There are two types of drawing:

a. The sketch: the user describes what he wants by precisely representing the contours of the objects, generally in a single and unique color, only the form aspect carries the information.

b. The rough drawing: a colored drawing is proposed by the user. It contains a colorful representation of each object, but or (the "but or" makes the phrase heavy) the outlines are generally vague. The color information (the colors themselves but also the arrangement of these) is therefore essential, unlike the shape which is not very representative.

v. Image search by objects: the user describes the characteristics of an object of an image rather than the entire image, the goal is to find a precise object in a series of images. An example of this type of query is to find images containing a person. You can also combine several objects and specify the spatial relationships between the different objects. An example of this type of query is to find images where a person is sitting next to a tree.

vi. Search by example: In this case, the system needs to compare an example of the same type (image) with the base to produce similar images. This method is natural simple and does not require deep knowledge to handle the system. It is therefore well suited to a non-specialist user.

After formulation, the system represents the request by its signature. The measures of similarities/dissimilarities between the signature of the request and that of all the images in the database are calculated and compared. The result is most often presented in the form of a list of images of descending similarity. As all the images in the database must be examined to find images similar to the query, the cost becomes prohibitive when the size of the

database increases. To remedy this problem, most RIC systems use an indexing scheme which provides a very efficient search method. The main idea is not to go through the whole database but to examine only a small part of the database.

3.7 MEASURES OF SIMILARITY BETWEEN DESCRIPTORS

In the search for visually similar content, images are represented by their characteristics (global or local) in the form of a multidimensional vector. The measure of similarity between two images is reduced to a metric defined on a vector space. The key idea in image search is approximate search, hence the use of the concept of proximity and distance between objects. Generally, the request is expressed in the form of one or more high-dimensional vectors, we then seek the closest neighbors of the request, and this during the online phase, where the system calculates the correspondence between the descriptor of the 'image-request and each descriptor of the database images.

Generally, a similarity measure verifies the following properties:

a. Perception: a small distance in feature space indicates two similar images.
b. The calculation: the distance measurement is calculated quickly for low latency.
c. Stability: the distance calculation should not be affected by a modification of the size of the base.
d. Robustness: the measurement should be robust to changes in image acquisition conditions.

Therefore, the calculation, established by a measure of similarity, results in an ordered list of the images of the base. The first image in this list is the one considered by the system to be the most relevant, ie the one that best responds to the image query.

The complexity of calculating a distance must be reasonable because in a CBIR system, this task is executed in real time. Other parameters come into play such as the dimension of the characteristic space, the size of the base,...etc.

Many definitions of distances exist, each obviously giving different results.
Consider two feature vectors X = (x1,x2,...,xn) and Y = (y1,y2,...,yn), the similarity between them can be calculated by the different distances described in detail in the following .

4. PROPOSED METHODOLOGY

Content Based Image Retrieval has been researched for many years now as an alternative of text based image retrieval. Increase in communication bandwidth, information content and the size of the multimedia databases has given rise to the concept of content based image retrieval. Content-based image retrieval (CBIR), also known as query by image content (QBIC) and content-based visual information retrieval (CBVIR) is the application of computer vision techniques to the image retrieval problem, that is, the problem of searching for digital images in large databases. "Content-based" means that the search analyzes the contents of the image rather than the metadata such as keywords, tags, or descriptions associated with the image. The term "content" in this context might refer to colors, shapes, textures, or any other information that can be derived from the image itself. CBIR is desirable because searches that rely purely on metadata are dependent on annotation quality and completeness. Having humans manually annotate images by entering keywords or metadata in a large database can be time consuming and may not capture the keywords desired to describe the image. The evaluation of the effectiveness of keyword image search is subjective and has not been well-defined. In the same regard, CBIR systems have similar challenges in defining success.

Current research works attempt to obtain and use the semantics of image to perform better retrieval. There exist many systems for image retrieval but efficient CBIR is a challenging task. Some works focused on how to represent an image, which means how to extract the visual features of an image. Some other works focused on how to understand an image, which means how to extract the objects in an image and describe the relationship between objects. All these works emphasize on the accuracy of the retrieval, but pay very little attention on the ability of responding to a huge amount of requests. However the content based image retrieval is very time consuming due to the extraction and matching of high dimensional and complex features. The aim of this project is to achieve highly accurate results along with a reduced computation time.

4.1 EXISTING SYSTEM ANALYSIS

Initially, image retrievals used the content from an image individually. Approaches using a combination of contents then started gaining prominence a bit later. Combining shape and color using various strategies such as weighting, histogram-based, kernel based, or

invariance-based has been one of the premier combination strategies. Shape and texture using elastic energy-based approach to measure image similarity has been presented. Others include an automated extraction of color and texture information using binary set representations, a color histogram along with the texture and spatial information. Image retrieval by segmenting them had been the focus of few research papers. Despite the extensive research effort, the retrieval techniques used in CBIR systems lag behind the corresponding techniques in today's best text search engines, such as in query, Alta Vista, Lycos, etc. For CBIR the research is primarily focused on exploring various feature representations, hoping to find a "best" representation for each feature.

4.2 PROPOSED SYSTEM

In this chapter we will discuss about the design of the system, system requirements, modules present in the system, implementation methodology and all the planning that has to be done before beginning the implementation of the project.

4.2.1 Problem Addressed

The Edge histogram method has low precision in certain cases when the segmentation cannot be done properly. Hence, this system will move Edge histogram descriptor to a second level filter and thus allowing the first level filter to pass on a refined dataset for the EHD thus tackling the precision issue. The BOW (Hue Min Max Difference) is preferred over other techniques as it converts a problem based on the well-known RGB histogram to the BOW histogram table thus reducing computation time. Also this technique concentrates on the images with 32 and 128 bins only and thus filtering out the need for unnecessary scaling and conversion. For the EHD the image is scaled down by a factor of two, thus reducing computation time and not affecting the image search. These two methods have never been used in unison before and we intend to try and use them to produce efficiency in output. However, merging two completely different approaches is always a difficult task as while both are working together, the results of one should not hamper the results of other in an unwanted way.

4.2.2 Proposal

Edge Histogram Descriptor is used to perform two-dimensional segmentation in an image. The selection of the image is done by generating a matrix to bring out the area of interest.

This helps to get rid of unwanted objects and to make the search more precise. Many methods such as magic wand, intelligent scissors, knockout and graphcut are also available but the reason for choosing EHD is due to its precision and is user friendly. This technique is only recently being used for CBIR.

String based color comparison method for content-based image retrieval is based on support vector machine classifier. In this method the feature extraction is done based on the color string coding followed by string comparison. This technique transfers image retrieval problem to strings comparison.

4.2.3 Proposed Implementation

First, user builds the database using calcfeatures which gives us different tables in the features MATLAB file.

The image is scaled down by a factor of two and edge value is computed.
The RGB feature histogram is generated for the image.
The RGB to BOW model uses RGB values and returns Hue, Sum and Difference.
These values are then converted into a quant table of hue, sum and difference.
If the image does not have a histogram of 32 or 128 bins an error message is displayed on the command line.
After calculating the BOW values the histogram is created.
When the query image is input, it calls the find similar function.
The database is loaded in the background to avoid processing delays.
The distance between all the image features in terms of BOW and EHD are calculated.
The output is generated based on the ascending order of these distances.

4.2.4 Design of System

The image is given as a query to the system. After selecting the query user gets the option to perform search. Pre-processing takes places and all the required information is extracted from the image. These extracted features are then compared to the database containing features of the image dataset. The output is obtained on the same GUI window with images shown in descending order of proximity.

BOW model is used to convert images to words using HMM technique. This technique is based upon factors like hue, shade, tone, tint, brightness of the color. The colors adjacent to

a particular color are considered to be neighboring colors. In the color space the indices are considered as partial values according to the distance. This method was developed to enhance the performance of image searches based on content.

The steps involved are as follows:

Step 1: Image is obtained from training dataset.

Step 2: Calculate the Hue value from image

Step 3: Calculate the Min and Max value from image

Step 4: Calculate the BOW transform for image

Step 5: Features of image is extracted and save it in training file.

Step 6: If this image is the last image, then preprocess the training file and train the classifier, otherwise go to step 1.

Step 7: Now, image is obtained from testing data set. Step 8: Extract the feature of image. Step 9: If it is the last image then predict the class using trained classifier, otherwise go to step 4.

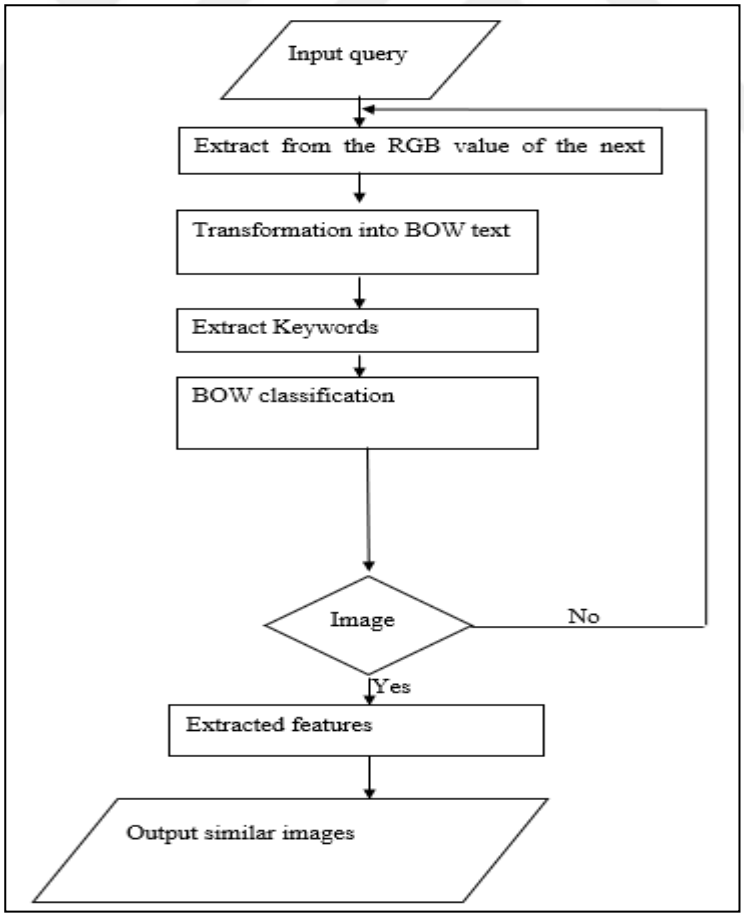

Figure 4.1: Proposed Image Search by BOVW.

The need to use this technique stems from the lack of spatial information used in the color based techniques. Edge information is used in this method to retrieve images.

Edge Histogram Descriptor (EHD) belongs to the family MPEG-7 descriptors. MPEG-7 provides different standard Multimedia Descriptors to describe and interpret visual contents for images. It offers the interoperability across different multimedia databases. Main visual features descriptors are color, shape and texture descriptors. In proposed methodology, Edge Histogram Descriptor (EHD) is applied to extract texture features of images. The EHD characterizes edges to represent spatial distribution in an image.

Feature extraction process using EHD consists of following steps:

a. First, the array of digital image is spitted into equal 4X4 subparts/ sub-images.
b. In next step, every subpart is further divided into no overlapping square blocks. Here size of blocks are depends on the resolution of input image.
c. From every block the edge is calculated and then type is identified using the filter coefficients illustrated in figure There are six types of edges. The type of edge may be vertical, horizontal, 450 diagonal, 1350 diagonal, no direction edge and no-edge. Initially first five types of edges are identified and no-edge blocks can be automatically obtained after the process of normalization.
d. By applying above steps, from every sub-image 5-bin edge histogram is obtained. Five bins are vertical, horizontal, 450 diagonal, 1350 diagonal, and no-direction
e. Then, the value of each bin in the sub-image is computed by normalizing total number of image blocks in the sub-image.
f. Finally for these normalized bins nonlinear quantization value is calculated to limit the number of bits sufficient for the descriptor. Texture features of input image and images present in database are extracted using above steps.

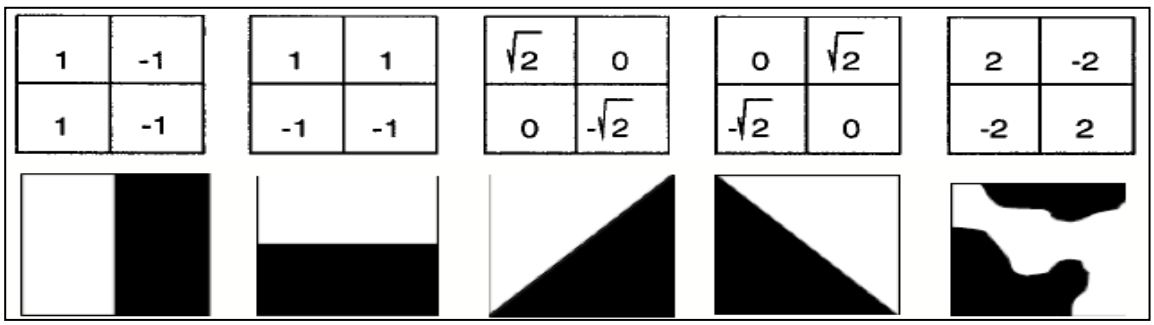

Figure 4.2: HSV Features Matrix.

4.2.5 Build Database Code Flow

The build database starts retrieving features from the images in a given folder. Depending on the size of the image the processing time per image varies. We use a 2000 image database which is a good number considering other search engines available. This database takes roughly 1.9 seconds * 2000 = 3800 seconds. i.e roughly an hour to get created.

4.2.6 GUI Code Flow

The GUI allows users to select a query image and then runs the image retrieval code on the same. The features of the query image are compared to the features stored in features.mat file and thus we get a result of about 20 similar images as the output. The precision and negative outputs are discussed in the result and analysis section.

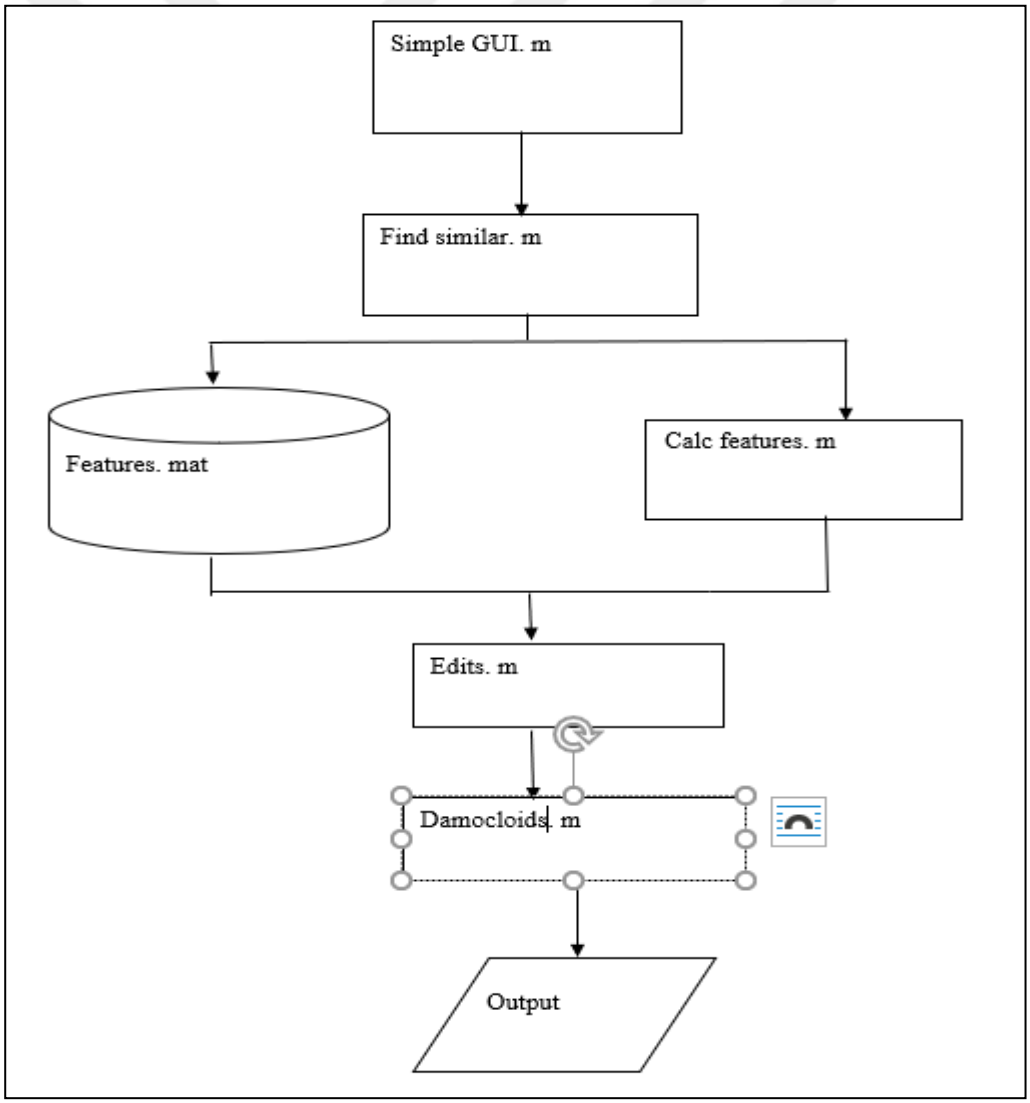

Figure 4.3: Structure of the Codes.

4.3 SYSTEM REQUIREMENTS

In this sub-topic we will be discussing about the software applications, programming languages, system configuration that was needed to develop this project.

4.3.1 Software Requirements

Matlab: The entire project was developed on the MATLAB R2015a edition. All the versions developed post 2013 are efficiently able to run this code. Matlab can be installed on any native OS and the code can be executed. Matlab was chosen over languages like Java, C++ and Python. Matlab is really efficient in dealing with mathematical formulas and there are vast number of algorithms available. Merging all of the algorithms also becomes easy because of its simple syntax. Creating a GUI is also easy with matlab.

4.3.2 Hardware Requirements

Hardware requirements of the system are:

i. RAM – Minimum 8GB, but 16GB recommended for no lag
ii. Operation System – Microsoft Windows 11 &above
iii. Processor Speed – Minimum 2GHz
iv. Hard disk – Minimum 50GB

4.3.3 Non-Functional Requirements

a. Efficiency Requirement: The processing should be quick when extracting features. The system should not lag or freeze during execution as image processing takes a lot of CPUS overhead.
b. Reliability Requirement: The results should be accurate as much as possible. It is impossible to get 100% efficiency but we should strive to reduce the number of negative results.
c. Usability Requirement: The UI is kept as simple as possible for even a layman to use. Also, the results should be legible enough to see clearly.

5. RESULT AND ANALYSIS

In this chapter, we are going to discuss about the results of the implementation of the proposed system. Modules of this system are implemented in a parallel manner and later on all the modules were merged to produce the final results. In the later part, we will analyze the system by comparing the results by using same database in another feature engines.

5.1 RESULTS AND IMPLEMENTATION

In this section, output of the implementation of each module will be shown with the help of screenshots in detailed manner which is as follows.

This will enable understanding how effectively it works as well as in what cases do we need better features to improve upon what we build.

5.2 BUILD DATABASE

We need to first generate the features.mat database of all the images in the dataset. This is done by running the builddatabase code on the command line. This will run all the modules under calfeatures and the processing time is displayed on the command line.

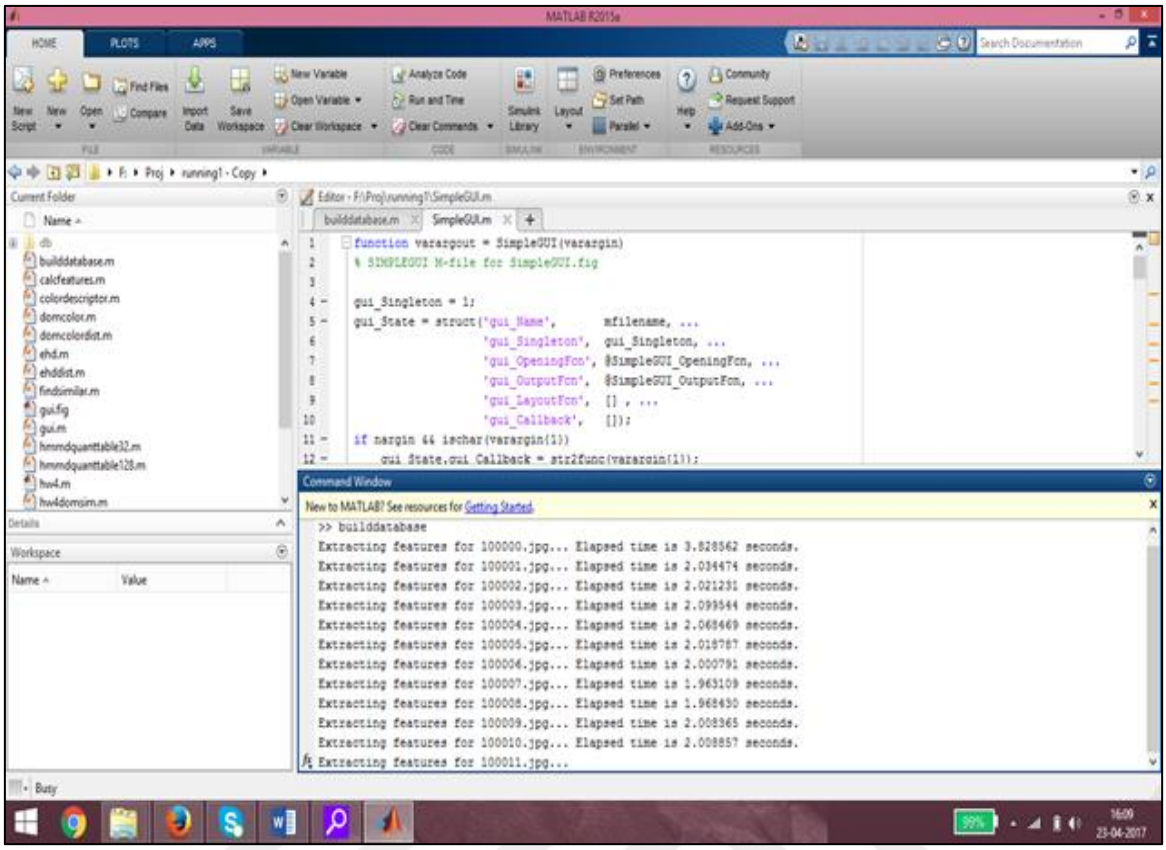

Figure 5.1: Build Database.

5.2.1 GUI Execution

To run the User interface we have to run it from the command line which will then shoot up the interface window. This is done as follows:

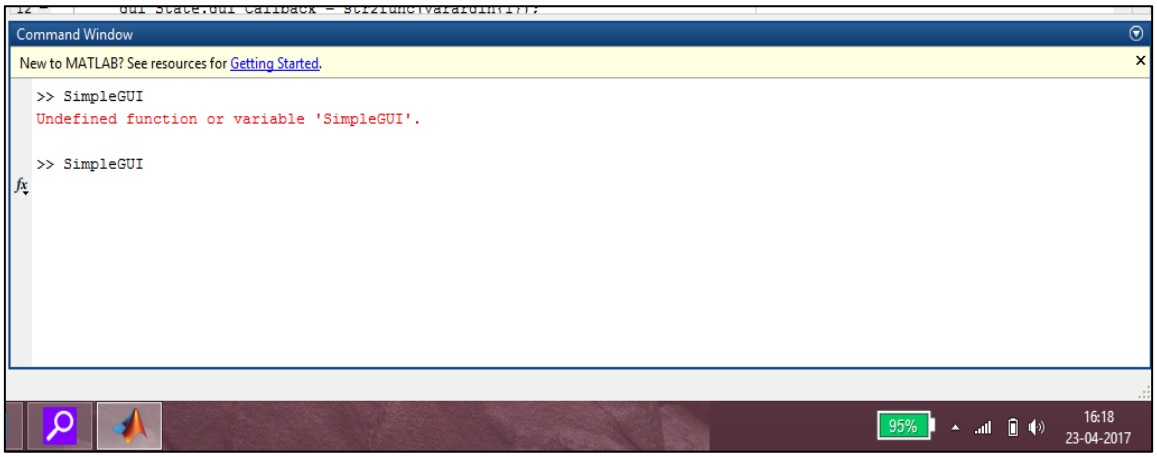

Figure 5.2: Gui Execution.

5.2.2 GUI Window

The GUI window is kept simple in order for users to understand how to use it without having any background knowledge.

This window contains the following options:

Close: To close the GUI window.

Minimize: To minimize the GUI window.

Browse: To select the image which will be used as a query.

Go! : To execute the image search.

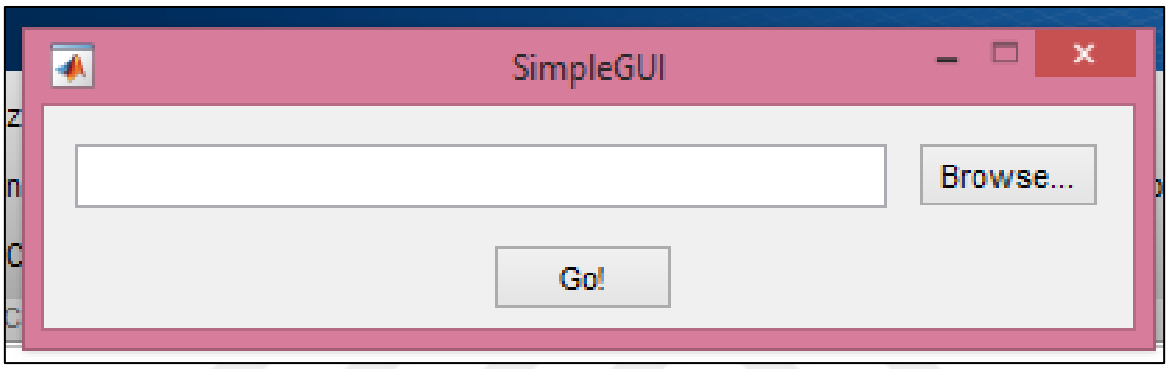

Figure 5.3: Gui Window.

5.2.3 Search Browse

The Browse commands opens up a window which allows users to search for the query image and select it. This window is similar to the selection window which we see while using general windows applications. Hence, it becomes easy to navigate through it.

Users have the options to navigate through directories, select an image, and select the type of images to search for and to cancel the selection.

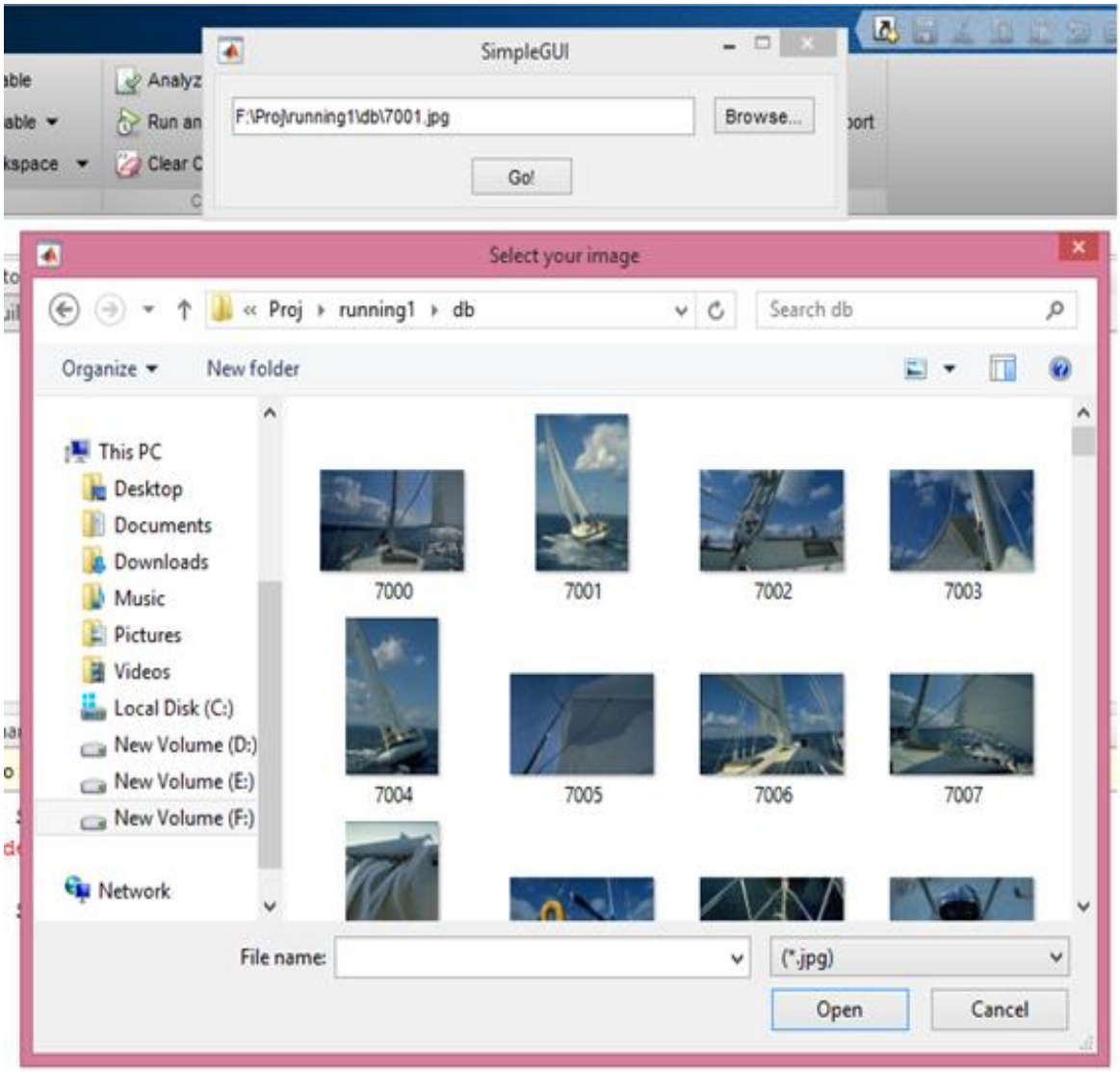

Figure 5.4: Browse.

5.2.4 Image To Keyword Search

The Command runs the image retrieval engine and finally presents the desired output. We have chosen to show 20 images in our output which are followed by the query image. You will observe that if the query is a part of the database it will be shown first in the output because it has the strongest similarity in terms of the features.

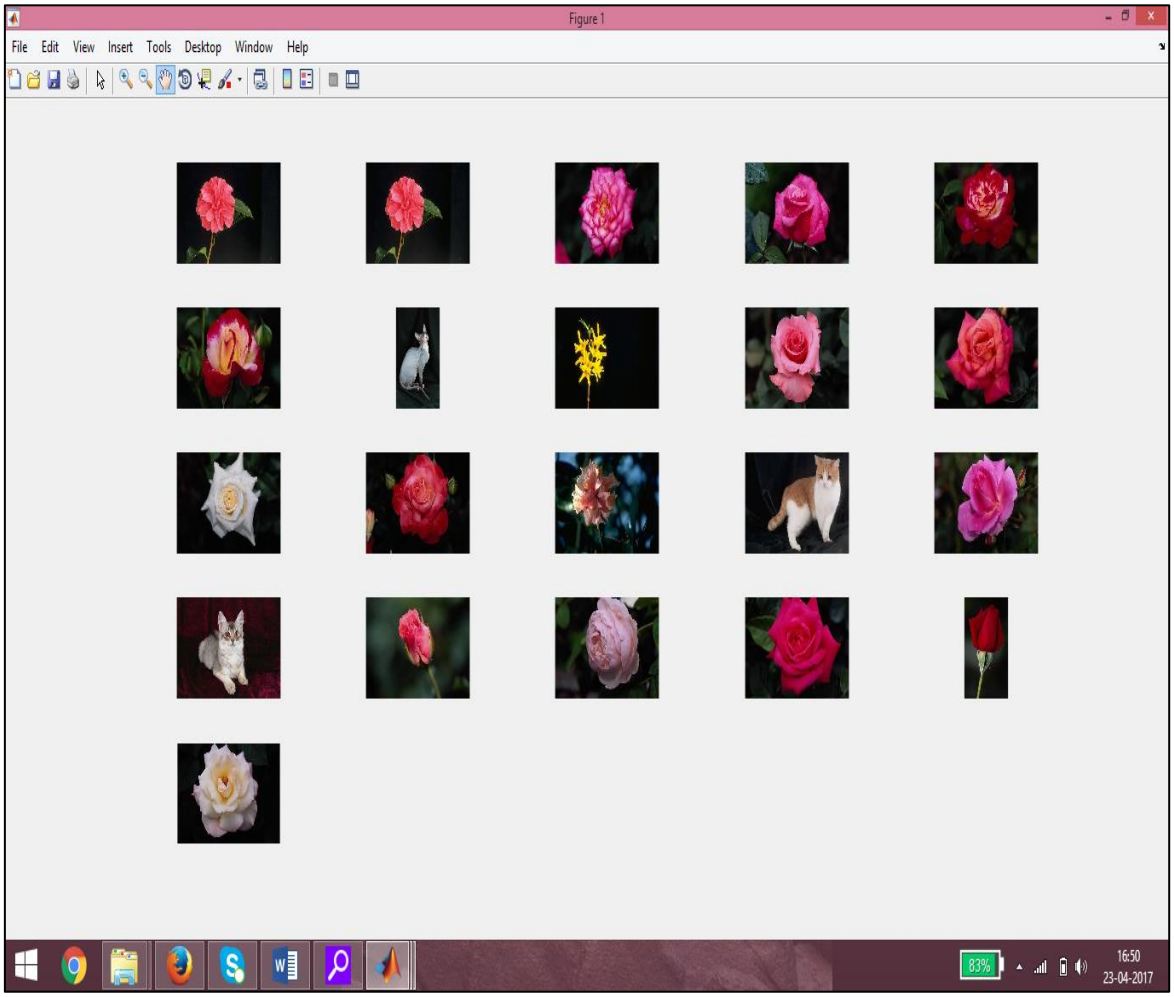

Figure 5.5: Searching Related Images from Keywords Extracted from the First Image.

5.3 COMPARISON AND ANALYSIS

5.3.1 Pre-Processing Time

As observed from the build database figure. The Average time taken for each image can be considered to be 2 seconds.

Our database consists of 2000 images.

Therefore,

Total time to preprocess

= average time per image * no. of images

= 2 seconds * 2000

= 4000 seconds

= 66.67 minutes

~ 1 hour 6 minutes.

If the images in the database are of low resolution, it takes lesser processing times. But this dataset gives us a good processing time with efficient images.

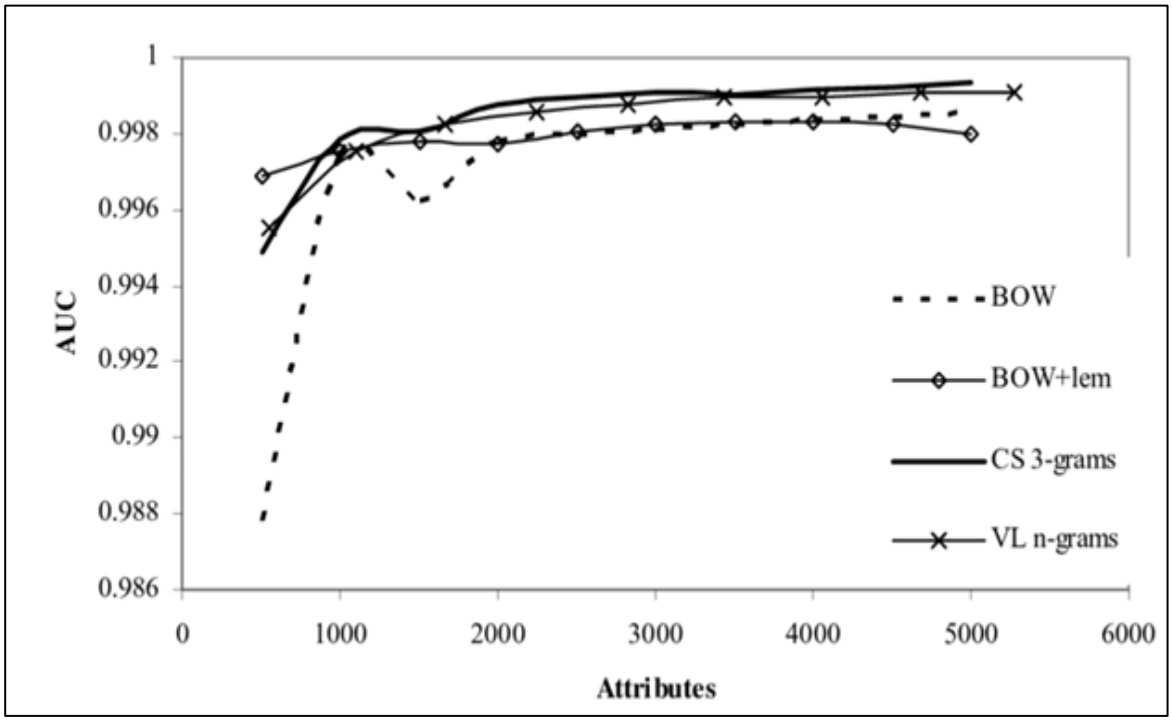

Figure 5.6: Comparing Time Results of BOW Algorithm on Different Datasets.

5.3.2 Efficiency Analysis

Positive output: This is the number of images which are expected and are in some way similar to the query image.

Negative output: This is the number of images which are a part of the output but they are in some way not similar to the query image.

On observing our output for multiple queries it is found that if there is too much of some particular color in the query image, the output will give unrelated images with a large amount of the same color.

We analyze how this system fares in comparison to other features which we could have used

by using the same database on an open-source project. We will compare the number of positive and negative outputs of both.

Our system (BOW and EHD)

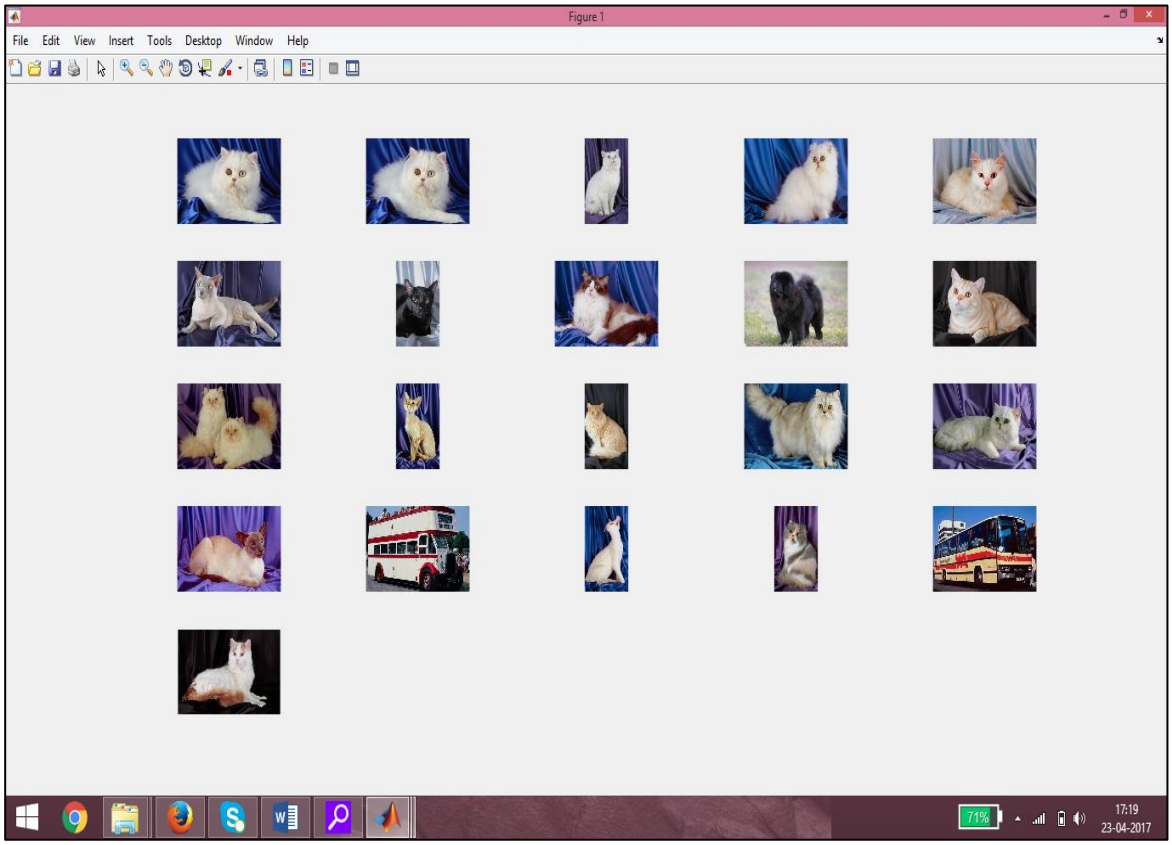

Figure 5.7: BOW and EHD.

Here we select a random image from the database and use it as a query for the engine. The returned output is then analyzed and the number of positive and negative searches are counted.

Positive images: 17/20

Negative images: 3/20

RGB and EHD

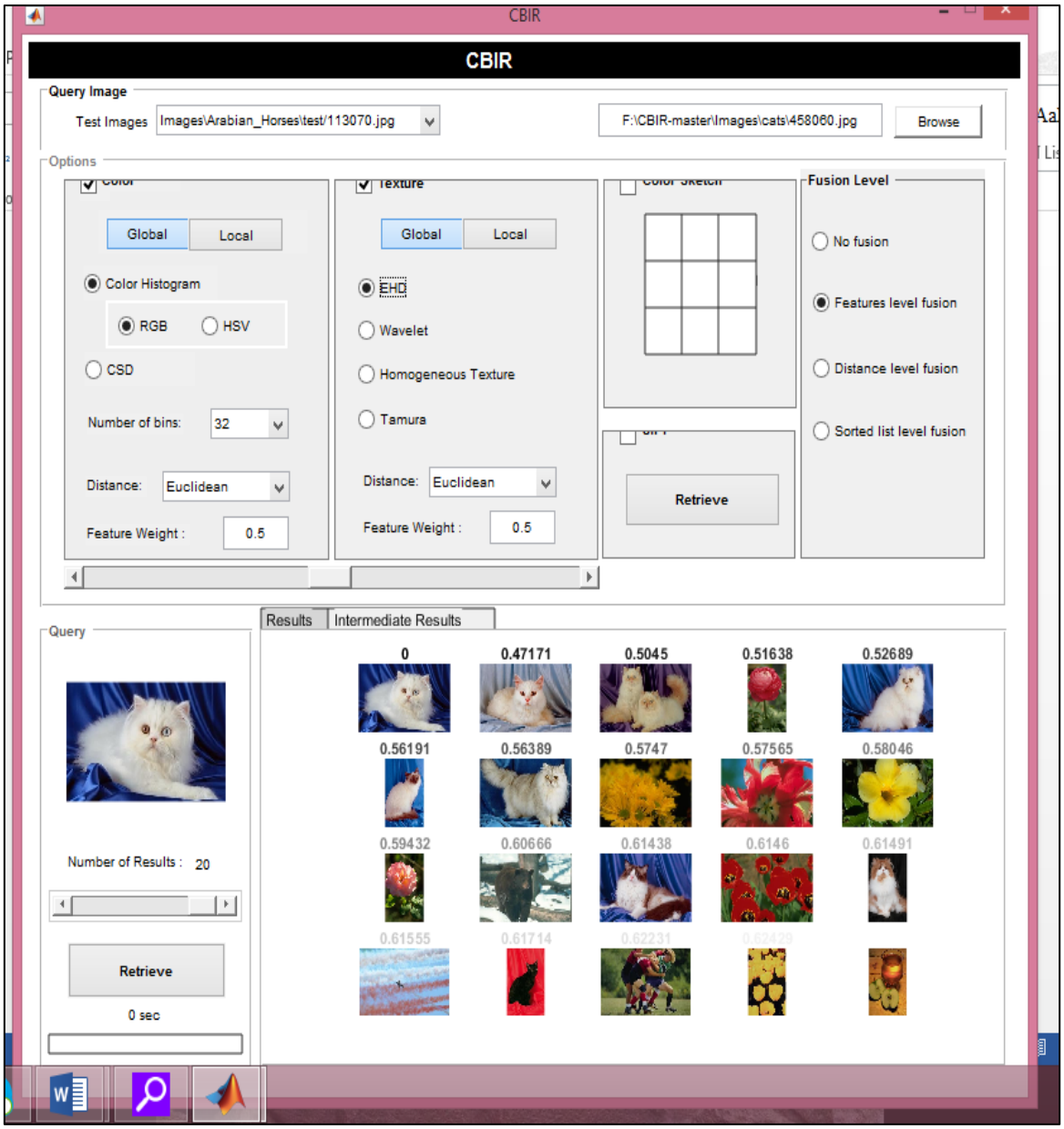

Figure 5.8: RGB and EHD.

We use another CBIR system available by selecting RGB and EHD as the extraction and comparison features. The output is then analyzed and positive and negative images are counted.

Positive Images: 9/20

Negative Images: 11/20

RGB and Wavelet

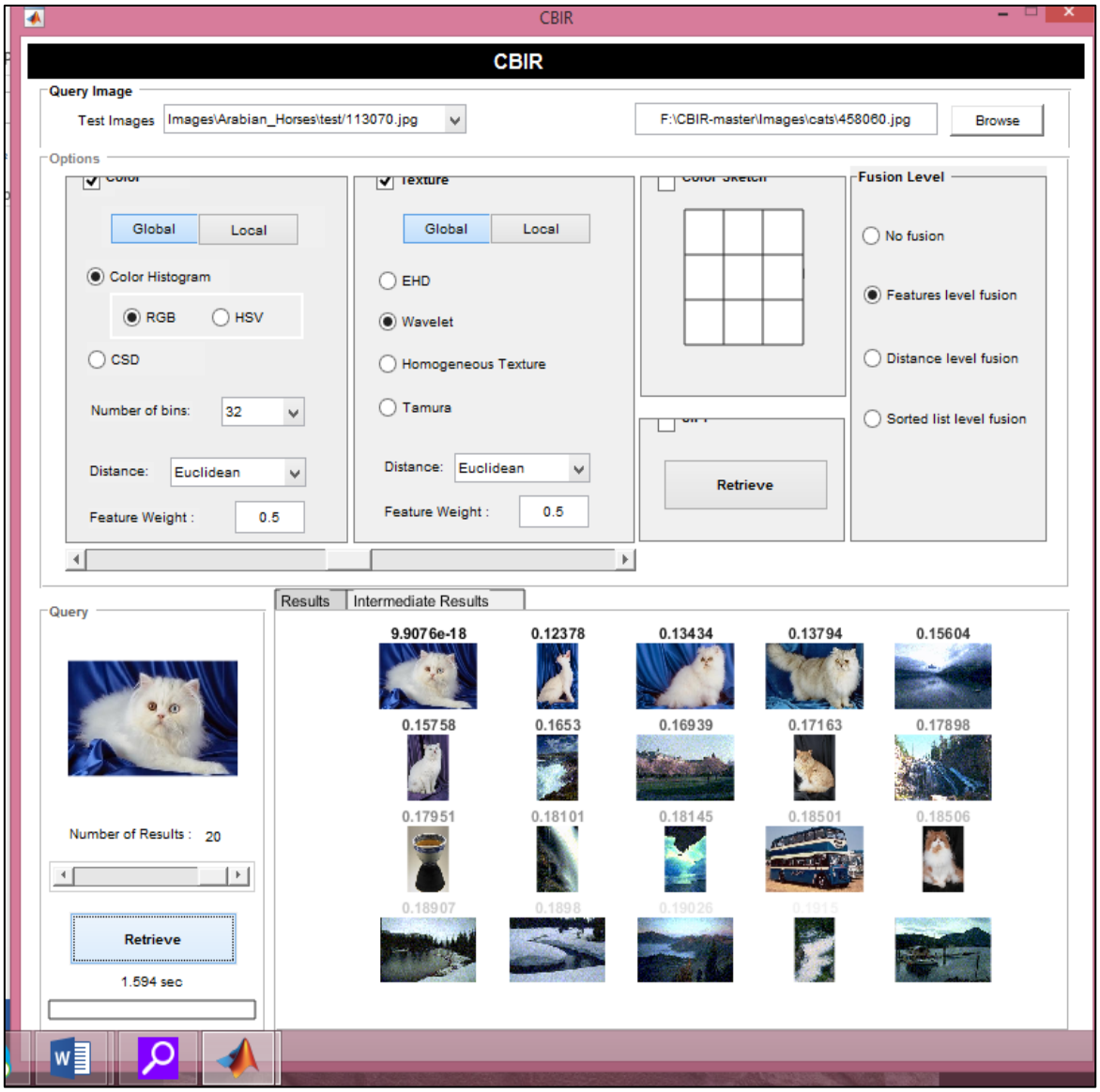

Figure 5.9: RGB and Wavelet.

We use another CBIR system available by selecting RGB and Wavelet as the extraction and comparison features. The output is then analyzed and positive and negative images are counted.

Positive Images: 7/20

Negative Images: 13/20

HSV and EHD

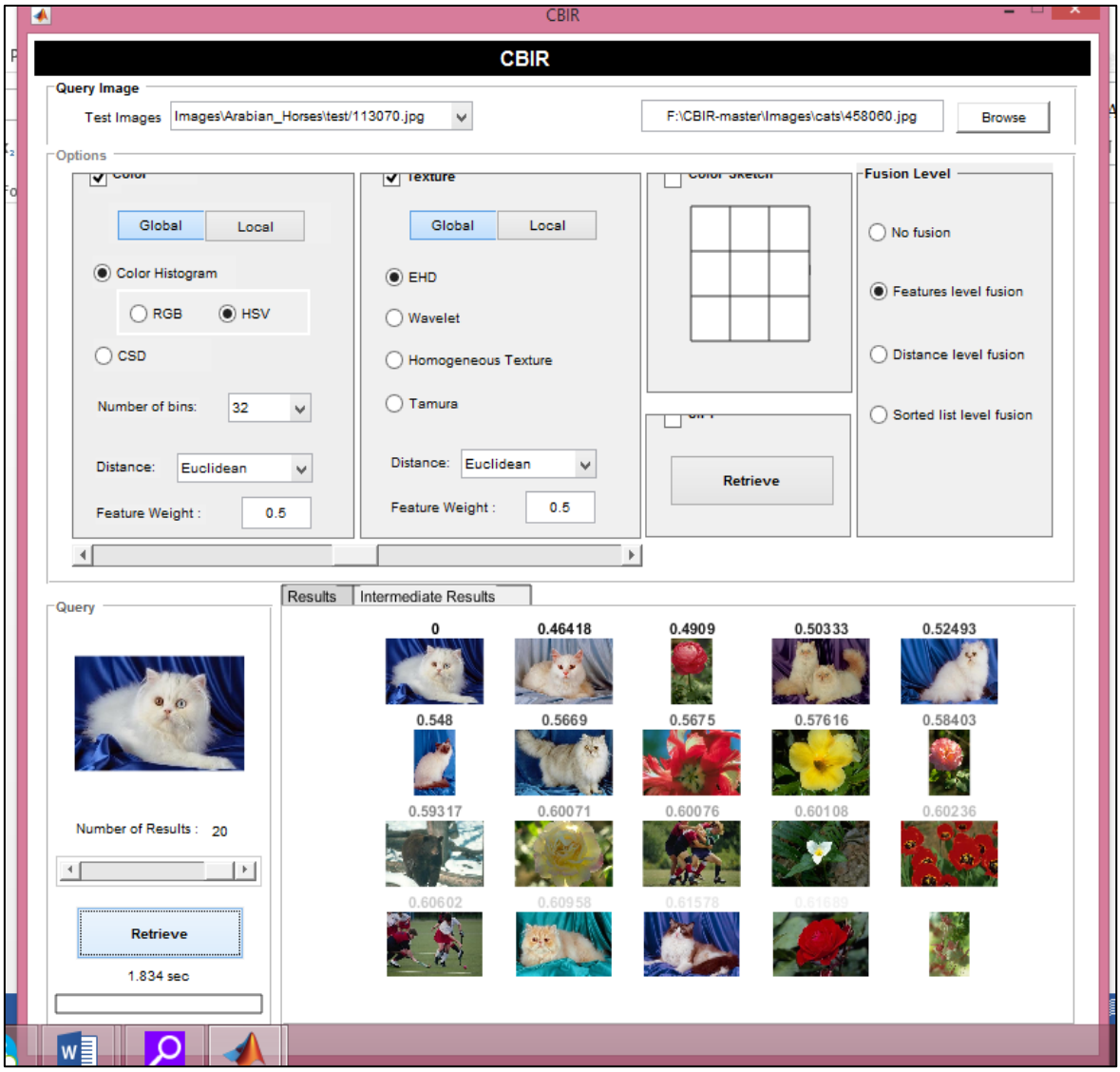

Figure 5.10: HSV and EHD.

We use another CBIR system available by selecting HSV and EHD as the extraction and comparison features. The output is then analyzed and positive and negative images are counted.

Positive Images: 8/20

Negative Images: 7/20

HSV and Wavelet

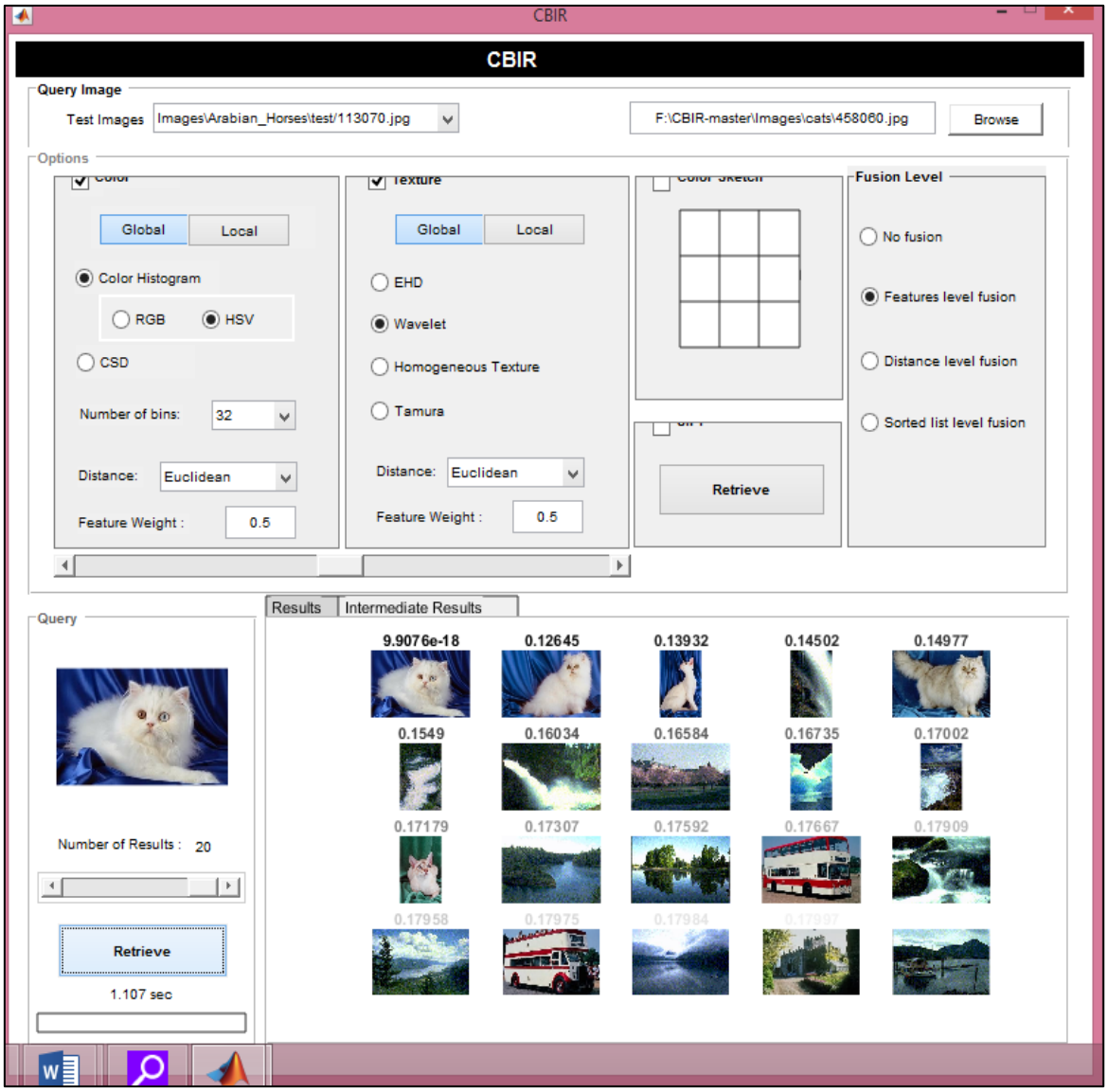

Figure 5.11: HSV and Wavelet.

We use another CBIR system available by selecting HSV and Wavelet as the extraction and comparison features. The output is then analyzed and positive and negative images are counted.

Positive Images: 5/20

Negative Images: 15/20

Case Scenarios

Here we will analyze the kind of query which will produce the worst, best and average case scenario and study them.

Worst case scenario:

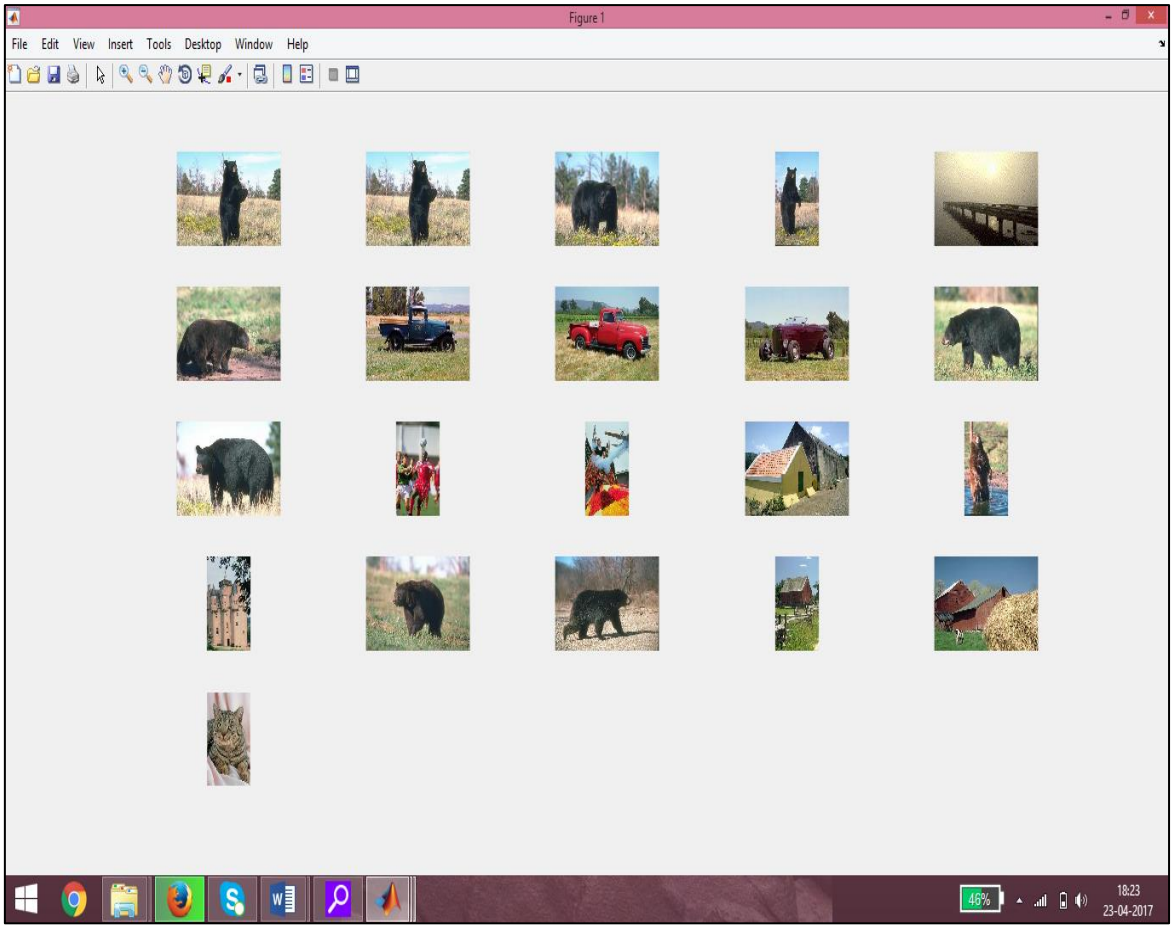

Figure 5.12: Worst Case Scenario.

Here, the query does not have a clear distinguishable foreground and the background consists of the same color shade and covers a large portion of the image. Hence we get images which have the same background in addition to images with same foreground.

Positive images: 8/20

Negative images: 12/20

Best case scenario:

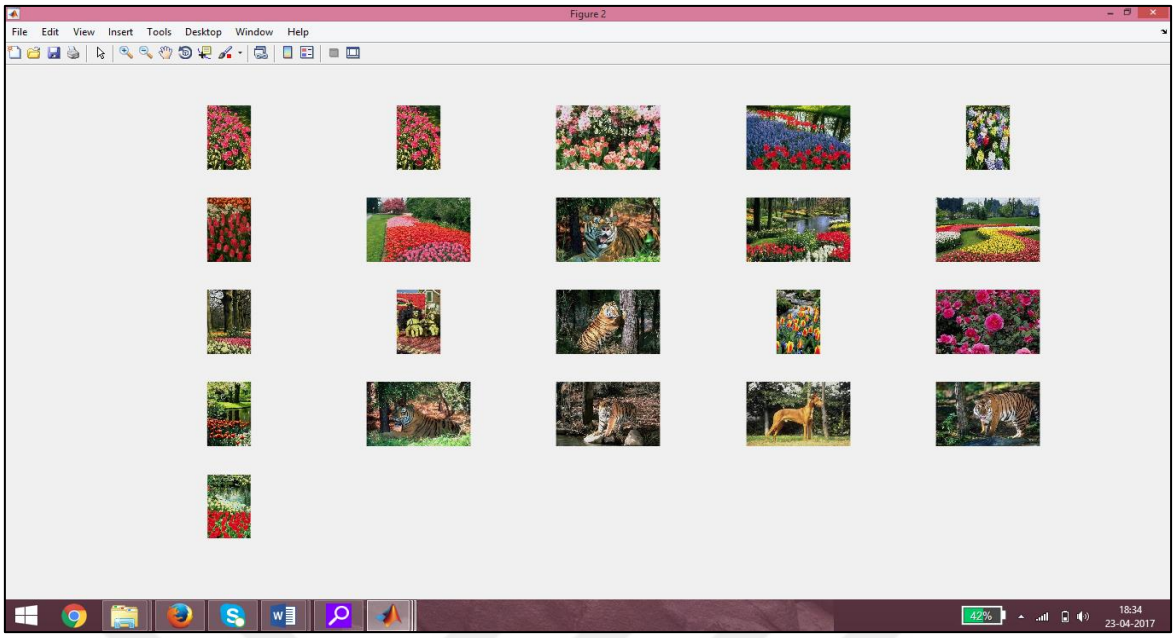

Figure 5.13: Good Case Scenario.

Images which are not similar but have the same color combinations can have a great amount of feature similarity between them. The entire image may be considered as a negative output, but some part of it does match the query.

Positive images: 14/20

Negative images: 6/20

Best case scenario:

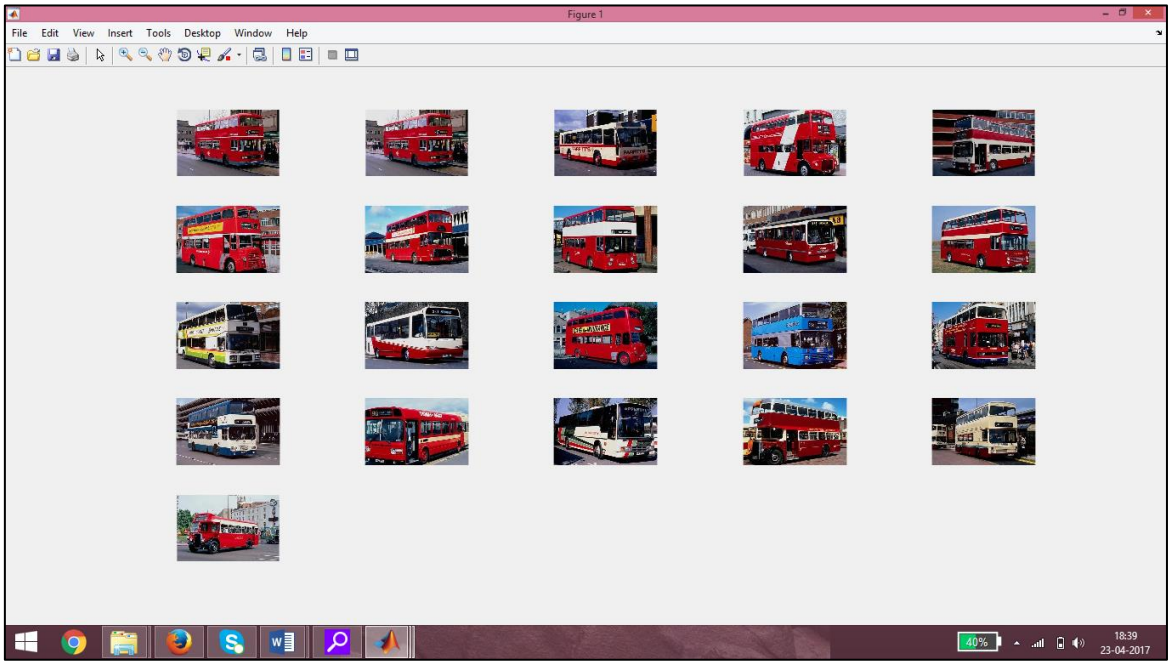

Figure 5.14: Best Case Scenario.

The query image has a well-defined shape and also the color distribution for the entire image is well balanced. In such cases it becomes very efficient to get good outputs which are completely efficient.

Positive images: 20/20

Negative images: 0/20

Tabular analysis

Table 5.1: Efficiency = Positive Images/ Total Images *100.

CBIR technique	**Positive images**	**Negative images**	**Efficiency**
BOW and EHD(cat)	17/20	3/20	85%
RGB and EHD(cat)	9/20	11/20	45%
RGB and Wavelet(cat)	7/20	13/20	35%
HSV and EHD(cat)	8/20	12/20	40%
HSV and Wavelet(cat)	5/20	15/20	25%
BOW and EHD(worst case)	8/20	12/20	40%
BOW and EHD(good case)	14/20	6/20	70%
BOW and EHD(best case)	20/20	0/20	100%

It can be observed that this technique is quite efficient. It performs way better than using some other existing features and, in a few cases, performs similar to them. It is only for a few exceptions that the performance degrades. Which should be acceptable given that image search is not as subjective as text-based search.

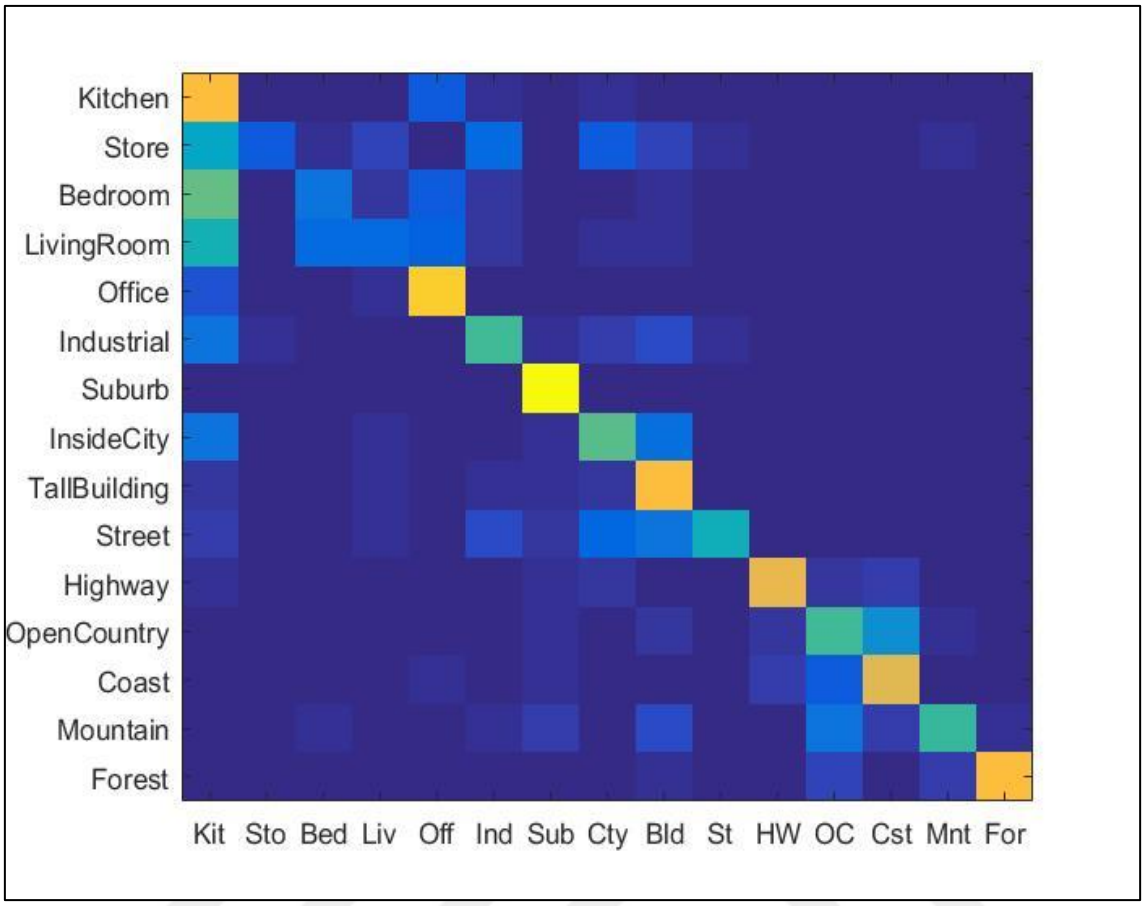

Figure 5.15: Confusion Matrix of the Keyword Extraction Efficiency.

6. CONCLUSION AND FUTURE WORK

6.1 CONCLUSIONS

Color and Shape based Image Search Engines a good way of performing CBIR with good efficiency. The system has been developed with so much care that it is free of errors and at the same time efficient and less time consuming. System is robust. The system is scalable and new filters can be added as well as new images can be added to the database. The main purpose of system is to find out whether the features to be used decided by us are competitive enough with the current systems being used. It is concluded that the system fares better than the other extraction feature systems which are available on the internet. The BOW features are way better than RGB and HSV when it comes to color based extraction as it is developed as a further iteration to these two. Also the EHD proves to be a better method than Wavelet and other segmentation methods. There are certain anomalies however that arise due to the combination of these two features. Too much of a color in the background takes away the weightage of the centre object. Such problems exist in every CBIR system but with further analysis we can surely work on improvement. This system is a research oriented system with several features mainly producing unique output of CBIR. The result indicates the potential evidence that BOW and EHD can be used and improved to create an efficient Image Search.

6.2 FUTURE WORK

Our future effort is to add the ability to allow the user searching based on the feature of his choice in addition to the hybrid method used in the project. New functionalities that can be added into this system. Image renaming feature can be added on the search panel. The output images could be renamed with a similar prefix or suffix in order to use the project as a sorting method to find only targeted images easily. Feedback for output could be asked to the used asking for satisfaction levels on using the system or to select the images he feels should not be shown. The user could provide feedback for the output he gets. Based on this feedback a dedicated learning algorithm can be used to improve the results after each use. Dedicated application could be built as a product for the market. There are tools available which can help convert a matlab code into an Android application. We could strive and make it into a dedicated product.

REFERENCES

[1] Li, Q.; Peng, H.; Li, J.; Xia, C.; Yang, R.; Sun, L.; Yu, P.S.; He, L. A Survey on Text Classification: From Shallow to Deep Learning. arXiv 2020, arXiv:2008.00364.

[2] Kowsari, K.; Jafari Meimandi, K.; Heidarysafa, M.; Mendu, S.; Barnes, L.; Brown, D. Text Classification Algorithms: A Survey. Information 2019, 10, 150. [Google Scholar] [CrossRef][Green Version]

[3] Minaee, S.; Kalchbrenner, N.; Cambria, E.; Nikzad, N.; Chenaghlu, M.; Gao, J. Deep Learning–Based Text Classification: A Comprehensive Review. Acm Comput. Surv. 2021, 54, 1–40. [Google Scholar] [CrossRef]

[4] Graves, A. Generating Sequences With Recurrent Neural Networks. arXiv 2013, arXiv:1308.0850. [Google Scholar]

[5] Manning, C.D.; Raghavan, P.; Schütze, H. Introduction to Information Retrieval; Cambridge University Press: Cambridge, UK, 2008. [Google Scholar] [CrossRef]

[6] Mielke, S.J.; Alyafeai, Z.; Salesky, E.; Raffel, C.; Dey, M.; Gallé, M.; Raja, A.; Si, C.; Lee, W.Y.; Sagot, B.; et al. Between words and characters: A Brief History of Open-Vocabulary Modeling and Tokenization in NLP. arXiv 2021, arXiv:2112.10508. [Google Scholar]

[7] Saif, H.; Fernandez, M.; He, Y.; Alani, H. On Stopwords, Filtering and Data Sparsity for Sentiment Analysis of Twitter. In Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), Reykjavik, Iceland, 26–31 May 2014; pp. 810–817. [Google Scholar]

[8] Jivani, A. A Comparative Study of Stemming Algorithms. Int. J. Comput. Technol. Appl. 2011, 2, 1930–1938. [Google Scholar]

[9] Plisson, J.; Lavrac, N.; Mladenic, D. A rule based approach to word lemmatization. In Proceedings of the 7th International Multiconference on Information Society (IS04), Ljubljana, Slovenia, 11–15 October 2004. [Google Scholar]

[10] Gage, P. A New Algorithm for Data Compression. C Users J. 1994, 12, 23–38. [Google Scholar]

[11] Sennrich, R.; Haddow, B.; Birch, A. Neural Machine Translation of Rare Words with Subword Units. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, Germany, 7–12 August 2016; pp. 1715–1725. [Google Scholar] [CrossRef]

[12] Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language Models are Unsupervised Multitask Learners. OpenAI Blog 2019, 1, 9. [Google Scholar]

[13] Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv 2019, arXiv:1907.11692. [Google Scholar]

[14] Wang, C.; Cho, K.; Gu, J. Neural Machine Translation with Byte-Level Subwords. Proc. AAAI Conf. Artif. Intell. 2020, 34, 9154–9160. [Google Scholar] [CrossRef]

[15] Schuster, M.; Nakajima, K. Japanese and Korean voice search. In Proceedings of the 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Kyoto, Japan, 25–30 March 2012; pp. 5149–5152. [Google Scholar]

[16] Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minnesota, MN, USA, 2–7 June 2019; pp. 4171–4186. [Google Scholar]

[17] Kudo, T. Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Melbourne, Australia, 15–20 July 2018; pp. 66–75. [Google Scholar] [CrossRef][Green Version]

[18] Kudo, T.; Richardson, J. SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System

Demonstrations, Brussels, Belgium, 31 October–4 November 2018; pp. 66–71. [Google Scholar] [CrossRef][Green Version]

[19] Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R.; Le, Q.V. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In Proceedings of the 33rd International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019. [Google Scholar]

[20] Conneau, A.; Khandelwal, K.; Goyal, N.; Chaudhary, V.; Wenzek, G.; Guzmán, F.; Grave, E.; Ott, M.; Zettlemoyer, L.; Stoyanov, V. Unsupervised Cross-lingual Representation Learning at Scale. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online event, 5–10 July 2020; pp. 8440–8451. [Google Scholar] [CrossRef]

[21] Rodolà, E.; Cosmo, L.; Litany, O.; Bronstein, M.M.; Bronstein, A.M.; Audebert, N.; Hamza, A.B.; Boulch, A.; Castellani, U.; Do, M.N.; et al. Deformable Shape Retrieval with Missing Parts. In Proceedings of the Eurographics Workshop on 3D Object Retrieval, Lyon, France, 23–24 April 2017; Pratikakis, I., Dupont, F., Ovsjanikov, M., Eds.; Eurographics Association: Geneva, Switzerland, 2017. [Google Scholar] [CrossRef]

[22] Gasparetto, A.; Minello, G.; Torsello, A. Non-parametric Spectral Model for Shape Retrieval. In Proceedings of the 2015 International Conference on 3D Vision, Lyon, France, 19–22 October 2015; pp. 344–352. [Google Scholar] [CrossRef]

[23] Pistellato, M.; Bergamasco, F.; Albarelli, A.; Cosmo, L.; Gasparetto, A.; Torsello, A. Robust phase unwrapping by probabilistic consensus. Opt. Lasers Eng. 2019, 121, 428–440. [Google Scholar] [CrossRef]

[24] Jones, K.S. A statistical interpretation of term specificity and its application in retrieval. J. Doc. 1972, 28, 11–21. [Google Scholar] [CrossRef]

[25] Maćkiewicz, A.; Ratajczak, W. Principal components analysis (PCA). Comput. Geosci. 1993, 19, 303–342. [Google Scholar] [CrossRef]

[26] Tharwat, A.; Gaber, T.; Ibrahim, A.; Hassanien, A.E. Linear discriminant analysis: A detailed tutorial. Ai Commun. 2017, 30, 169–190. [Google Scholar] [CrossRef][Green Version]

[27] Tsuge, S.; Shishibori, M.; Kuroiwa, S.; Kita, K. Dimensionality reduction using non-negative matrix factorization for information retrieval. In Proceedings of the 2001 IEEE International Conference on Systems, Man and Cybernetics. e-Systems and e-Man for Cybernetics in Cyberspace (Cat.No.01CH37236), Tucson, AZ, USA, 7–10 October 2001; Volume 2, pp. 960–965. [Google Scholar] [CrossRef]

[28] Rosenfeld, R. Two decades of statistical language modeling: Where do we go from here? Proc. IEEE 2000, 88, 1270–1278. [Google Scholar] [CrossRef]

[29] Jurafsky, D.; Martin, J. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, 3rd ed. (draft) Unpublished. 2021, pp. 30–35. Available online: https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf (accessed on 28 December 2021).

[30] Huang, E.H.; Socher, R.; Manning, C.D.; Ng, A.Y. Improving Word Representations via Global Context and Multiple Word Prototypes. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, Jeju Island, Korea, 8–14 July 2012; Volume 1, pp. 873–882. [Google Scholar]

[31] Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient Estimation of Word Representations in Vector Space. In Proceedings of the 1st International Conference on Learning Representations (ICLR 2013), Scottsdale, AZ, USA, 2–4 May 2013. [Google Scholar]

[32] Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed Representations of Words and Phrases and their Compositionality. In Proceedings of the 26th International Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 5–10 December 2013; Volume 2, pp. 3111–3119. [Google Scholar]

[33] Pennington, J.; Socher, R.; Manning, C. GloVe: Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543.

[34] Baroni, M.; Dinu, G.; Kruszewski, G. Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, Baltimore, MD, USA, 22–27 June 2014; Volume 1, pp. 238–247.

[35] Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T. Enriching Word Vectors with Subword Information. Trans. Assoc. Comput. Linguist. 2017, 5, 135–146. [Google Scholar] [CrossRef][Green Version]

[36] Xu, S.; Li, Y.; Wang, Z. Bayesian Multinomial Naïve Bayes Classifier to Text Classification. In Advanced Multimedia and Ubiquitous Engineering; Springer: Singapore, 2017; pp. 347–352. [Google Scholar] [CrossRef]

[37] Van den Bosch, A. Hidden Markov Models. In Encyclopedia of Machine Learning and Data Mining; Springer: Boston, MA, USA, 2017; pp. 609–611. [Google Scholar] [CrossRef]

[38] Sutton, C.; McCallum, A. An Introduction to Conditional Random Fields. Found. Trends® Mach. Learn. 2012, 4, 267–373. [Google Scholar] [CrossRef]

[39] Cover, T.; Hart, P. Nearest Neighbor pattern classification. IEEE Trans. Inf. Theory 1967, 13, 21–27. [Google Scholar] [CrossRef]

[40] Li, B.; Yu, S.; Lu, Q. An Improved k-Nearest Neighbor Algorithm for Text Categorization. arXiv 2003, arXiv:cs/0306099. [Google Scholar]

[41] Ali, N.; Neagu, D.; Trundle, P. Evaluation of k-nearest neighbour classifier performance for heterogeneous data sets. SN Appl. Sci. 2019, 1, 1–15. [Google Scholar] [CrossRef][Green Version]

[42] Bellman, R. Dynamic Programming. Science 1966, 153, 34–37. [Google Scholar] [CrossRef] [PubMed]

[43] Cortes, C.; Vapnik, V.N. Support-Vector Networks. Mach. Learn. 1995, 20, 273–297. [Google Scholar] [CrossRef]

[44] Boser, B.E.; Guyon, I.M.; Vapnik, V.N. A Training Algorithm for Optimal Margin Classifiers. In Proceedings of the Fifth Annual Workshop on Computational Learning Theory, Pittsburgh, PA, USA, 27–29 July 1992; pp. 144–152. [Google Scholar] [CrossRef]

[45] Safavian, S.; Landgrebe, D. A survey of decision tree classifier methodology. IEEE Trans. Syst. Man Cybern. 1991, 21, 660–674. [Google Scholar] [CrossRef][Green Version]

[46] Ho, T.K. Random decision forests. In Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, Canada, 14–15 August 1995; Volume 1, pp. 278–282. [Google Scholar] [CrossRef]

[47] Islam, M.Z.; Liu, J.; Li, J.; Liu, L.; Kang, W. A Semantics Aware Random Forest for Text Classification. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM '19), Beijing, China, 3–7 November 2019; pp. 1061–1070. [Google Scholar] [CrossRef][Green Version]

[48] Genkin, A.; Lewis, D.; Madigan, D. Large-Scale Bayesian Logistic Regression for Text Categorization. Technometrics 2007, 49, 291–304. [Google Scholar] [CrossRef][Green Version]

[49] Krishnapuram, B.; Carin, L.; Figueiredo, M.; Hartemink, A. Sparse multinomial logistic regression: Fast algorithms and generalization bounds. IEEE Trans. Pattern Anal. Mach. Intell. 2005, 27, 957–968. [Google Scholar] [CrossRef][Green Version]

[50] Breiman, L. Bagging predictors. Mach. Learn. 2004, 24, 123–140. [Google Scholar] [CrossRef][Green Version]

[51] Schapire, R.E. The Strength of Weak Learnability. Mach. Learn. 1990, 5, 197–227. [Google Scholar] [CrossRef][Green Version]

[52] Freund, Y.; Schapire, R.E. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. J. Comput. Syst. Sci. 1997, 55, 119–139. [Google Scholar] [CrossRef][Green Version]

[53] Joulin, A.; Grave, E.; Bojanowski, P.; Mikolov, T. Bag of Tricks for Efficient Text Classification. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, Valencia, Spain, 3–7 April 2017; Volume 2, pp. 427–431. [Google Scholar]

[54] Ibrahim, M.A.; Ghani Khan, M.U.; Mehmood, F.; Asim, M.N.; Mahmood, W. GHS-NET a generic hybridized shallow neural network for multi-label biomedical text classification. J. Biomed. Inform. 2021, 116, 103699. [Google Scholar] [CrossRef]

[55] Iyyer, M.; Manjunatha, V.; Boyd-Graber, J.; Daumé, H., III. Deep Unordered Composition Rivals Syntactic Methods for Text Classification. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, Beijing, China, 27–31 July 2015; Volume 1, pp. 1681–1691. [Google Scholar] [CrossRef]

[56] Le, Q.; Mikolov, T. Distributed Representations of Sentences and Documents. In Proceedings of the 31st International Conference on Machine Learning, Beijing, China, 21–26 June 2014; Volume 32, pp. 1188–1196. [Google Scholar]

[57] Mikolov, T.; Le, Q.V.; Sutskever, I. Exploiting Similarities among Languages for Machine Translation. arXiv 2013, arXiv:1309.4168. [Google Scholar]

[58] Hochreiter, S.; Schmidhuber, J. Long Short-term Memory. Neural Comput. 1997, 9, 1735–1780. [Google Scholar] [CrossRef] [PubMed]

[59] Tai, K.S.; Socher, R.; Manning, C.D. Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, Beijing, China, 27–31 July 2015; Volume 1, pp. 1556–1566. [Google Scholar] [CrossRef][Green Version]

[60] Dieng, A.B.; Wang, C.; Gao, J.; Paisley, J. TopicRNN: A Recurrent Neural Network with Long-Range Semantic Dependency. arXiv 2016, arXiv:1611.01702. [Google Scholar]

[61] Howard, J.; Ruder, S. Universal Language Model Fine-tuning for Text Classification. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia, 15–20 July 2018; Volume 1, pp. 328–339. [Google Scholar] [CrossRef][Green Version]

[62] Wang, B. Disconnected Recurrent Neural Networks for Text Categorization. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia, 15–20 July 2018; Volume 1, pp. 2311–2320. [Google Scholar] [CrossRef][Green Version]

[63] Cho, K.; van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1724–1734. [Google Scholar] [CrossRef]

[64] Schuster, M.; Paliwal, K. Bidirectional recurrent neural networks. IEEE Trans. Signal Process. 1997, 45, 2673–2681. [Google Scholar] [CrossRef][Green Version]

[65] Peters, M.E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L. Deep Contextualized Word Representations. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, New Orleans, LA, USA, 1–6 June 2018; Volume 1, pp. 2227–2237. [Google Scholar] [CrossRef][Green Version]

[66] Zhang, Y.; Wallace, B.C. A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification. In Proceedings of the Eighth International Joint Conference on Natural Language Processing (IJCNLP), Taipei, Taiwan, 27–30 November 2017; Volume 1, pp. 253–263.