Addis Ababa University

College of Natural Sciences

Automatic Sentence Based Image Description Generation Framework

YORDANOS SHIMELIS HAILE

A Thesis Submitted to the Department of Computer Science in Partial Fulfillment for the Degree of Master of Science in Computer Science

Addis Ababa, Ethiopia

September 2019

Addis Ababa University

College of Natural Sciences

Automatic Sentence Based Image Description Generation Framework

YORDANOS SHIMELIS HAILE

Advisor: ***Fekade Getahun (PhD)***

This is to certify that the thesis prepared by Yordanos Shimelis , titled*: Automatic Sentence Based Image Description Generation Framework* and submitted in partial fulfillment of the requirements for the Degree of Master of Science in Computer Science complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the Examining Committee:

Name	Signature	Date
Advisor: Dr. Fekade Getahun	_____	
Examiner:	_____	
Examiner:	_____	

Abstract

Sentence-based image description generation is a challenging task involving several open problems in the fields of Natural Language Processing and Computer Vision. To address this problem most of the previous efforts for this task rely on visual clues and corpus statistics. The generation approaches employ both concepts-to-text and text-to-text natural language generation methods, which generate image description by transferring text from descriptions of a similar image and generate a summary for a new image from retrieval related document but do not take much advantage of the semantic information inherent in the available image descriptions. Since these approaches have no capable of building novel descriptions.

We focus on novel descriptions generation for unseen images. Here, we present a generic approach, which benefits from two sources visual data and available descriptions simultaneously. Our approach works on syntactically and linguistically motivated phrases extracted from the human descriptions. The proposed framework has three main components, which are called Image Engine, Search Engine, and Text Engine. Image Engine does feature extraction from training image dataset and provide to indexer sub-component, after indexation is completed visual word, construction is done by clustering local descriptor. Search Engine does feature extract from unseen image and compute similarity measure between image feature in the index and unseen image. The text engine does syntactically, and linguistically motivated phrases extracted from the textual descriptions and generate linguistic model. Then each image associate with linguistic model. Finally, text engine does assemble phrases into a grammatically correct sentence. Experimental evaluations demonstrate that our design mostly generates well-spoken and semantically correct descriptions.

In order to validate the proposed approach, a Java-based prototype is developed. We used LIRe and Lucerne for low-level features extraction and indexation and for phrase extraction, we used Stanford core NLP and for sentence generation, we used SimpleNLG. Using relative metrics such as recall and precision measures were conducted using sample test images. The experimental result gives 60% recall and 75% precision.

Keyword: *Text engine; Image engine; Search engine; image index; Phrase relevance evaluation; phrase integration*

ACKNOWLEDGEMENTS

First and foremost, I would like to thank over and over again the almighty God for giving me the strength to achieve whatever I have achieved so far. Next to this, I would like to express my deepest gratitude to my advisor, Dr. Fekade Getahun, for his supervision, clever advice, patience, and invaluable support throughout this thesis work. This thesis would not be successfully completed without your supervision and critical reading, I have learned a lot from you; I thank you again.

Many thanks also to all my friends who have supported and encouraged me. Thank you, my classmate, Mr. Mebratu Teshale who contributes his constructive encouragement idea from the beginning of this work and fruitful clarifications and subsequent guidance throughout the study as well as for facilitating related tasks to achieve my research schedule effectively.

My warm appreciation goes to my friends and classmates for the family relationship we have that shape and encourages me for completing my studies. I would like to thanks all department of computer science staff members of Addis Ababa University, collage of Natural science for their support. I would like to thanks to Ethiopian Broadcasting Corporation for entitled me this opportunity.

Finally, I would like to say thanks to my beloved family who encourage and appreciate me in an innumerable way in every aspect of my work.

Table of Contents

List of Tables

List of Figures

List of Algorithms

Acronyms and Abbreviations

AI	Artificial Intelligence
API	Application Programming Interface
BoVW	Bag of Visual Word
CBIR	Content-Based Image Retrieval
CEDD	Color Edge Directivity Descriptor
CLD	Color Layout Descriptor
CRF	Conditional Random Field
DCT	Discrete Cosine Transformation
DOG	Difference Of Gaussian
EHD	Edge Histogram Descriptor
FCTHD	Fuzzy Color Histogram Descriptor
GPS	Global Position System
HMMD	hue-min-max-difference
HOG	Histogram of oriented gradient
HSV	Hue, Saturation, Value
HVS	Human Visual System
IR	Information Retrieval
JCD	Joint Composite Descriptor
LDA	Latent Dirichlet Allocation
LIRE	Lucerne Image Retrieval Framework
LM	Language Model
MRF	Markov Random Field
NLP	Natural Language Processing
NLTK	Natural Language Toolkit

NLG	Nature Language Generation
PTB	Penn Treebank tagset
PASCAL	Pattern Analysis, Statistical Modelling Computational Learning
POS	Part Of Speech
RGB	Red, Green, Blue
SCD	Scalable Color Descriptor
SIFT	Scale Invariant Feature Transformation
SUB	Stony Brook University
YCbCr	Color space Family
XML	Extensible markup Language

Chapter One: Introduction

1.1 Background

In a modern information society, digital images play an increasingly important role in today's communication and our day-to-day life. For social media, image sharing is a popular mode of communication and is used not just for communicating with friends and family, but also for citizen journalism and activism. The availability of image capturing devices such as digital cameras, image scanner The expansion of social media that allow sharing of images, around 130,000 photos are uploaded each minute on Facebook. Flickr, another the most popular images hosting website, hosts more than 6 billion images. The network bandwidth has increased, and image compression techniques improved. With the development of the Internet the amount of digital image collection is increasing rapidly [1].

Users for various domains, such as entertainment, educational issue, crime prevention, and investigation and medical-related issues, require efficient image searching, browsing, and retrieval tools. With this, indexing and searching relevant images has become a challenging task. It also has attracted a lot of attention in the research community to get the automatically generated description for an image [2].

An image can be described either by a list of keywords or by a higher-level structure such as a sentence. [3] [4] [5] Keywords based image description is useful in various applications like image indexing, image retrieval, search engine optimization, and video surveillance. The keyword-based approach is inspired by the web search engines but has its own limitations[6].For instance an image description labeled with (black, car, dog) does not convey the information. Whether attribute black is associated, with dog or car, and what are their states and relative position. Whereas the sentence "A dog is sitting on a black car or "A black dog is sitting inside a car." Indirectly encodes the relationships between words *Figure1-1* .

http://news.softpedia.com/news/Flickr-Boasts-6-Billion-Photo-Uploads-215380.shtml

(Black, dog, car)

"A black dog is sitting inside a car."

(Black, dog, car)

"A dog is sitting inside a black car."

Figure1-1 Limitations of Keyword-Based Image Description am images taken from (http://www.imageclef.org/photodata)

Hence, the automatic conversion of keywords based to sentence based description using natural language, which has stronger semantic content. *Figure1-1* importance of image descriptions sentence based descriptions can differentiate images, but independent labels cannot. Both the above images have same labels but different semantics. Automatic image description generation is a complex AI (Artificial Intelligence) task, because of the huge semantic gap between how the computers interpret an image and how humans look at image. It requires both image understanding and natural language generation [7].

Previous work for generating descriptions for unseen images rely mostly on few object detectors ,classifiers and corpus statistics Mitchell *et al.*,[8] , [7],and Farhadi *et al* [2] but do not utilize the semantic information encoded in the available descriptions of images. Either they use these descriptions to restrict the set of prepositions or verbs Farhadi *et al* [2], or pick one or more complete sentences and transfer them to a test image unaltered [9]. While the former may result in quite verbose and non-humanlike descriptions, in the latter it is very unlikely that a retrieved sentence would be as descriptive of a particular image as a generated one. This is because a retrieved sentence is constrained in terms of objects, attributes and spatial relationships between objects; whereas a generated sentence can more closely associate the semantics relevant to a given image.

The main purpose of this thesis is to propose a framework for automatic sentence based image description generation. We present a generative approach that gives emphasis to textual information rather than just relying on computer vision techniques. Instead of using object detectors, we estimate the content of new image based on its similarity with available images. To minimize the impact of encountering noisy and uncertain visual inputs, we extract the syntactical phrases from known descriptions and use only those for combining new descriptions.

Extracting dependency patterns from textual descriptions rather than using it as an n-gram language model is inspired by [1]. These patterns have a predefined structure (e.g. (subject; verb), (attribute; subject), (object; verb), (attribute; object), (subject: prep; object), (verb; prep; object)), and can easily be mapped to generate a syntactically and grammatically correct description. The main strength of our approach is that it works on these patterns which carry a bigger chunk of information, compared to predicting individual component such as objects, attributes, verb, preposition, etc. in a piece-wise manner and then combining them at a later stage as done by previous approaches.

1.1 Motivation

Describing an image with a set of labels is not sufficient to explain what is present in an image (*Figure1-1*). To address the limitations of keyword-based annotation, we need to annotate images with higher-level structure such as a phrase or a sentence. Majority of the previous approaches addressing the problem of generating image descriptions rely on trained object detectors and classifiers to determine what is present in the image but none of them use the semantic information encoded in the available image descriptions.

1.2 Statement of the Problem

The problem is formally stated as: Given a dataset of images and their corresponding textual descriptions, the task is to describe an unseen image *Figure1-2*

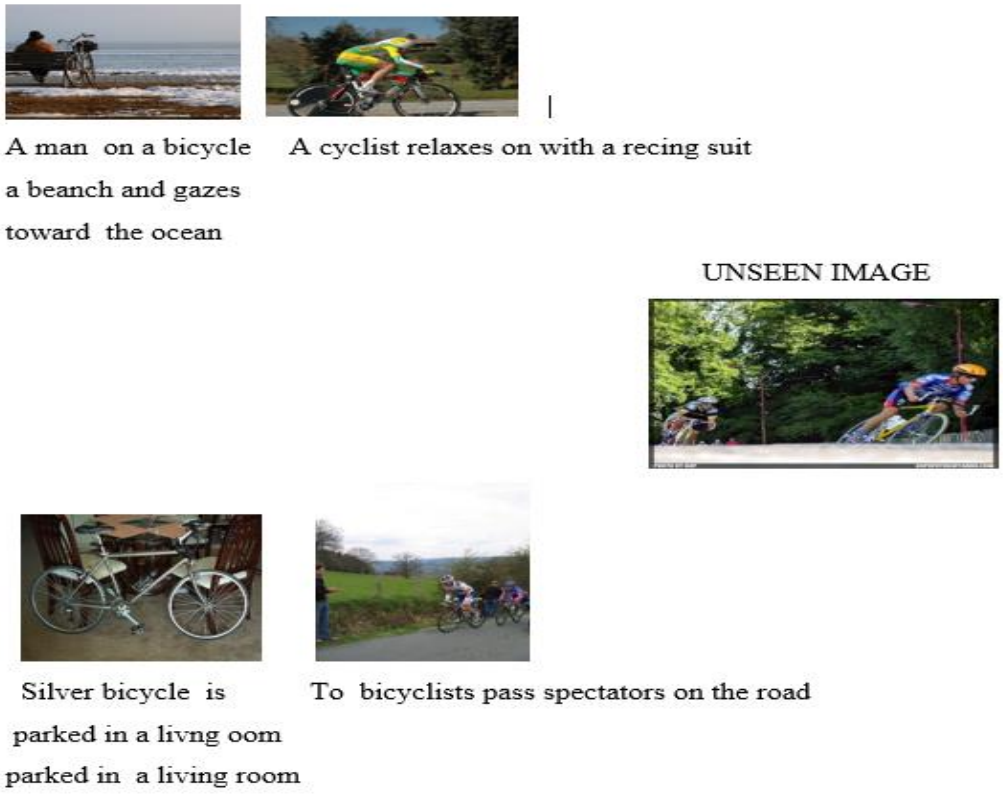


Figure 1-2 Problem Statement

Generating natural language descriptions for images is a challenging task. Identifying image content (objects, stuff, attributes, scene, and action) directly from images is often noisy and unreliable. In Yang *et al* [10] for 20% of the testing dataset, no objects are detected. Detector scores below a predefined threshold *Figure1-3,* then how these various components interact with each other i.e. the relationship between words (preposition) and finally finding the correct order of words to generate a description.

- Failure of state-of-the-art object detectors: tree, road, person
- Incorrect preposition[7]: tree under road, person under road

Figure 1-3 Challenges in Object Detectors an image from the PASCAL sentence dataset.

1.3 Objective

General objective

The main objective of this thesis is to design and implement a generic-based image description generation framework.

Specific objectives

In order to achieve the above general objective, the following specific objectives are identified:

- To review existing work in the area of image feature extraction, image similarity computation, and description generation.
- To study and analyze components of image description generation approaches and how these various components interact with each other.
- To select appropriate tools for image feature extraction, phrase relation extraction and sentence generation, prepare training dataset for both visual and textual data.
- To designing a framework, which improve the performance of CBIRs (Content Based Image Retrieval system) and minimize the semantic gap.
- To develop a prototype to validate the proposed framework

- To conduct experiments to evaluate the accuracy of the proposed framework.
- To draw conclusion and recommend future research areas.

1.4 Methodology

To realize the above objective, the following methods shall be used:

Literature review: Detail review and assessment shall be made on works, which are done in the area of image description generation. Specifically, the literature review focus on image understanding, the general architecture of image description generation, image data sources and their properties, and other related issues shall be studied.

Data Sources: Automatic image description generation requires a huge image with the corresponding human written description dataset for clear and meaningful image description. Therefore, samples of the image with description shall be collected from different multimedia sources over the web such as 2D Images, UIUC Pascal (Pattern Analysis Statistical Modelling and Computational Learning) sentence, Flickr 8k, Flickr 30k, Microsoft Coco and abstract scenes. Sampling shall be done depending on their content.

Development Tools and Programing Language: Components of the image description generation shall be designed using java programming language. Open-source libraries compatible with a java programming language shall be used. The reason is java is suitable for developing proposes prototype. LIRe (Lucerne Image Retrieval) shall be used to extract visual features. Lucene Java-based Search library shall be used to index and searches for any kind of document that is used as a high-level feature indexer [11]. Stanford core NLP shall be used to extract related phrases[12]. SimpleNLG it is a java library and used to description generation[13].

Testing and Evaluation: Proper testing shall be made and the newly proposed solution shall be evaluated in terms of its goals and contributions. The proposed work shall be evaluated using relative metrics such as recall and precision that helps to properly measure the performance of the proposed image description generation system. The proposed system also shall be evaluated based on human evaluation that will evaluate the system performance.

1.5 Scope and Limitations

The scope of this research work is focused on developing an automatic sentence- based image description generation framework and the limitations of this approach require a huge collection of images with their corresponding human-written descriptions and need a sufficient number of images for different categories. Due to resource limitation, system performance imposed on this research, because of this we are forced to put some boundaries.

1.6 Application of Results

The result of the work can apply for image search engines like, Google images in responding to a user query (Google images have a huge amount of image resources, which are not fully utilized because of poor image description embedded with an image)[6].

It helps visually impaired (blind and partially sighted) user to have better understand the content of images on the web and increasing accessibility to them. Improves the understand of their surroundings, most modern mobile phones are able to capture photographs, making it possible for the visually impaired to impact images of their environments. These images can then be used to generate a sentence that can be read out loud to the visually impaired so that they can get a better sense of what is happening around them [1].

The following are some of the areas where the result of the work can be used:

- User queries can be processed based on the context of the images so relevant results can be generated in a way of achieving efficient web multimedia resource usage.
- To make visually impaired users had better understand the content of images on the web and increasing accessibility to them [14].
- An alternate way to access information that is shared in an image, in cases where the user is unable to access or view the image directly.
- Users who access the Internet using mobile devices or limited bandwidth connections. To reduce the amount of human labor needed for organizing, retrieving, and analyzing digital media.
- Help users to make decisions about what they are seeing in an image in a situation where the computer is making a decision, used to ask a human for help or feedback.
- Searching online or personal image collections (humans often use natural language to describe images that they wish to search for).

- A testbed for image understanding, a testbed for grounded language understanding.

1.7 Organization of the Thesis

The overall structure of the thesis is organized as follows. Chapter Two describe the literature review work relevant to this thesis. This includes image understanding and natural language generation .Chapter Three, describes related work that is state of the art on description generation and image representation approaches of image description generation and related issues are covered. Both chapters present the gaps in the reviewed researches and it proposes a solution to bridge the gaps. Chapter Four presents the architecture of the proposed framework and its components. It also describes the flow of the system operation; this chapter explains our approach to generate descriptions for images from the corresponding human-written description.

The implementation technique, the development environment, and tools are described in Chapter Five. Analyses and evaluates the empirical results of the set of experiments conducted to validate the proposed approach presented in Chapter Four. It also discusses experimental setting, dataset, and issues related to the evaluation methodology. Experimental results, which determine the effectiveness of the system, also described in this chapter. This paper is ended up with the last chapter that is Chapter Six that contains conclusion about the overall system work and recommendation for future work.

Chapter Two: Literature Review

2.1 Introduction

This literature review is about the general background information in the area of our work that is relevant for understanding our framework. The main components of the image description generation processes are image understanding and language generation. Each is a very challenging problem, motivating the entire communities of researchers from the fields of NLP (Natural Language Processing) and CV (Computer Vision).

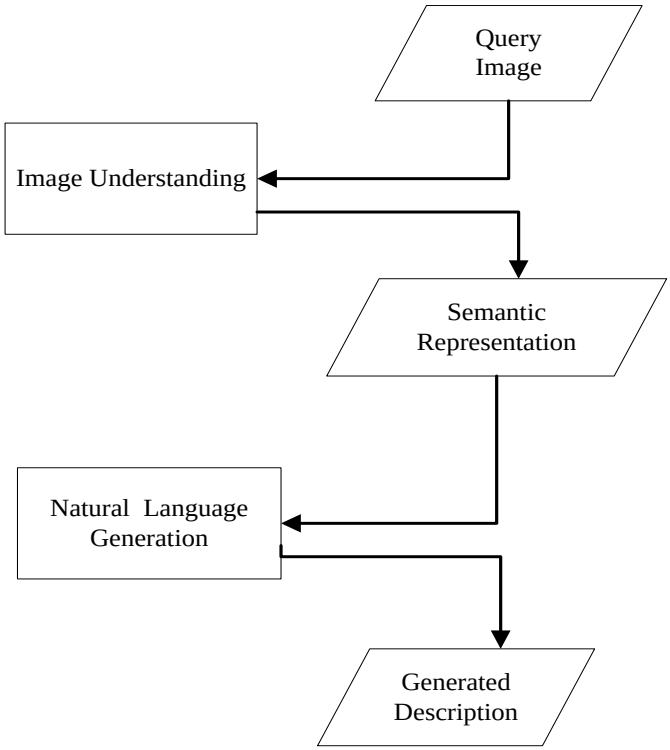


Figure 2-1 General Architecture for Image Description Generation from[15]

2.2 Image Understanding

Image understanding refers to the process of determining the content and meaning of a query image. This process is usually associated with Computer Vision, a field of research concerned with automatically reconstructing properties of the real world according to visual input [8]. However, there are some applications where this information can be recovered from alternate sources, such as meta-information, or related text. This section briefly reviews image-understanding approaches, which are relevant to automatic image descriptions generation.

2.2.1 Computer Vision

As shown in *Figure 2-2* Computer vision tasks include methods for processing, analyzing and understanding images. Image understanding can be seen as the separating of visual information from image data using models constructed with the aid machine learning. Applications of computer vision include image classification, visual object detection, image retrieval, and machine vision and traffic automation [8].

Figure 2-2 Task of Computer Vision from [8]

2.2.1.1 Visual Object Recognition

Visual object detection is one of the central problems of CV research and is one component of many automatic image description generation approaches. The core challenge is to learn basic categories of visual objects, to locate and identify new instances of these categories. The human vision system is able to perform this task with very little effort, considering the difficulty of the task. Consider, for instance, the task of recognizing a table. There are many different kinds of tables, such as a dining table, a workbench, an operating table, or a ping-pong table. However, humans are able to recognize that these all belong to the same conceptual category and infer the identity of objects that have not been seen before. Even the exact same object can vary in appearance in different images.

The automatic object detection system, which has most clearly been used for keywords based image description is the Deformable Part-Based Model [16]. Part-based models are particularly helpful for recognizing object categories such as humans, which appear in different poses. It represents images using low-level HOG(histogram of gradients) features [17], which measure the direction of the change of intensity at different parts of the image. To train this object detector, match the movable parts of the object in the training image, such as wheels on a bicycle, or limbs on a person. Then use SVM to discriminatively learn the different objects. Supervised models such as the Deformable Part-based Model require images with labeled instances of objects for training. Typically, each label corresponds to a bounding box that indicates the location of the object in the image.

2.2.1.2 Scene Recognition

Another fundamental challenge in Computer Vision is scene recognition. Many scenes can be described by their global spatial properties. [18] The well-known GIST feature is a global image descriptor related to perceptual dimensions such as “naturalness”, “roughness’, and “ruggedness”. These features are coarsely localized in order to describe the structure of the image. Another well-known global image descriptor is the Tiny Image descriptor [19], which resizes the image to a 32x32 thumbnail so that the structure of the scene can be described using the overall layout of the colors in the thumbnail image. Both GIST and Tiny Image descriptors can be used for classifying types of scenes, such as of different kinds of scenes: beach, forest, City Street, and so on. They can also help in recognizing different attributes of

scenes, such as man-made vs natural environments, or indoor lighting vs outdoor lighting. Finally, scene-level image descriptors can also be used as a measure for comparing images in many data-driven Computer Vision applications. These methods reduce an inference problem for an unknown image to finding an existing labeled image that is similar.

2.2.1.3 Feature Recognition

Finally, our thesis works broadly focus on feature recognition for image understanding approach. Feature recognition is the process of generating descriptors that represent the visual content of images in a certain manner. Because of the high dimensionality of an image's visual feature space, its representation is in the form of one or more visual features [20].

A descriptor defines the syntax and semantics of the feature representation. Examples of visual features include a low-level feature such as color, shape, spatial and texture and high-level features such as metadata (keywords and text associated with images). A feature is a function of one or more measurements, which specifies some quantifiable property of an object and quantifies its significant characteristics. Features can be obtained from the whole image (global features), or from blobs, which are segmented parts of the image (local features). [6] Classifies the visual features as follows:

General features: Application-independent features, such as color, shape, and texture and are further divided into:

- Pixel-level features: the computed features at every single pixel, e.g. color, location. For example low-level image processing *Figure 2-2*, which involves manipulating visual information on the pixel level.
- Local features: the computed features over the segmented regions or blocks obtained by the subdivision of the image.
- Global features: the computed features over the entire image or the regular sub-area of an image.

Domain-specific features: Application-dependent features; they are often a synthesis of low-level features for a specific domain such as human faces, fingerprints, and conceptual features. On the other hand, all features can be coarsely classified into low-level and high-level features. While low-level features can be extracted directly from the image, high-level feature

extraction bases on low-level features or from associated textual description [21]. In most cases image features are represented using low-level (global and local) features or with high-level features (metadata and description).

2.2.1.4 Global Feature

Global features quantify characteristics of the whole image to be measured. An example would be a color histogram of an image, which gives important information about the whole image [21]. The commonly used feature representation is based on a global feature set extracted from images. Many sophisticated global feature extraction algorithms have been designed and good surveys are available [22].

A color feature is one of the most widely used visual features and it is proven a very discriminate feature for image similarity search. Colors are defined in selected color space. Color spaces shown to be closer to human perception and used widely in IR systems include, RGB, HSV, YCrCb and the HMMD (Hue-Min-Max-Difference). Common color descriptors CBIR systems include color-covariance matrix, color histogram, color moments and color coherence vector [23]. The color histogram is the most commonly used representation technique, statistically, it describes the combined probabilistic of the three-color channels RGB. However, those color features cannot adequately describe the high-level concept of an image because two semantically different images may have the same color distribution.

A texture feature refers to the patterns in an image that present the properties of homogeneity that do not result from the presence of a single color or intensity value. Texture provides important information in image retrieval as it describes the content of many real-world images such as fruits, clouds, trees, bricks, and fabrics. Hence, texture is key feature in defining high-level semantics for image retrieval purposes. However, it is almost impossible to describe texture in words, because it is more a statistical and structural property. The commonly used texture features in image retrieval systems include features obtained using Gabor filtering or wavelet transformation[24], statistical features obtained using local statistical measures such as the six Tamura texture features. Among the six Tamura features coarseness, directionality, regularity, contrast, line likeness, contrast and roughness, the first three are more significant. The remaining three are related to the first three and do not add much to the effectiveness of texture description.

2.2.1.5 Local Feature

Local features quantify characteristics of a particular region of the object to be measured. Local features are computed at multiple points in the image and are consequently more robust to occlusion and clutter. Advantageous to choose local image features over global features for the following reasons [25].

- Locality: Small spatial neighborhoods are less sensitive to image deformations. Other parts of the object can be occluded.
- Pose Invariance: The interest point detector can select a canonical position, scale, and orientation. Matches are then made with respect to these canonical coordinates.
- Applicability: One view of a textured object is sufficient for training in order to recognize an object from nearby 3D viewing directions (but possibly different scales, image locations, and image orientations).

2.2.2 Non-Visual Approaches

There are also non-CV approaches to image understanding, which are used for image descriptions generation, retrieval, and annotation. Image search engines on the web, such as *images.google.com* normally use text that is related to the image in order to decide which images to retrieve for a query. Previous work has used related text and meta data such as an article related to a news image [26], or the GPS (global position system) coordinates where the image was taken [1].

2.3 Natural Language Generation

Natural language generation is an area of NLP that deals with the automatic production of text or speech according to a certain input [27]. Generation methods are often categorized as either concept-to-text methods, which produce textual output from non-linguistic input or as text-to-text methods that produce textual output using input text that is linguistic input from human-authored sources [28]. Generating description is a NLG (natural language generation) problem. The task of NLG is to turn a non-linguistic representation into human-readable text. Typically, the non-linguistic representation is a logical form (semantic, not syntactic form) or a set of numbers.

In image description, the input is an image representation, which the NLG model has to turn into sentences. Generating text involves a series of steps [27] .we need to decide which aspects of the input to talk about (content selection), then we need to organize the content (text planning) and verbalize it (surface realization). Surface realization, in turn, requires choosing the right words (lexicalization).Hence; automatic image description requires not only full image understanding but also a sophisticated natural language generation.

2.3.1 Concept-to-Text Generation

The most basic tasks of this concept-to-text generation techniques are a selection of content to be in the output text and realization of the natural language output. Content selection is determined by the input data such as the output of a visual detection system as well as the communication objective for the output, and a set of constraints capturing linguistic or other knowledge. This objective may be reached using various AI planning algorithms [29].

Surface realization is a linguistic process of constructing a sentence using the choices of words and syntactic structures found in the content selection stage [29]. It involves applying morphological and syntactic rules so that the output text sounds natural and correct. The rules governing this process are relatively well understood, and there are several software systems available for realization [13]. However, understanding which concepts are important, and selecting words and phrases to describe that content, is still an open research question.

2.3.2 Text-to-text Generation

In a text-to-text generation, content is typically specified by some textual input source. Some examples of text-to-text generation are:

Summarization, generating a summary that contains only the most important information in a document or group of documents.

- Extractive summarization methods select relevant sentences from the original documents and using that text as the summary [30].
- Abstractive summarization methods generate new sentences to describe relevant content from the source documents [30].

Compression: decreasing the length of an input sentence by deleting words that are not relevant, without making the sentence ungrammatical [31].

Paraphrasing :rewording and rearranging phrases or sentences in a different way from the original [32].

Simplification, rewriting a sentence to make it easier to understand[28]. Synthesis, combining the relevant content of two sentences into one single sentence.

Text-to-text generation approaches are typically guided by some concept of relevance. In some cases, relevance is determined using basic qualities of the input text, such as the frequency of a word in a document, or the positions of noun phrases in the grammatical structure of the text. Outside Sources, including non-linguistic information can also be used to guide the selection of relevant content.

2.4 Generated Image Description

There is a variety of motivating research at the intersection between CV and NLP. We are tried to maintain the background on two main tasks in describing images, from generating a keyword-based description to generating a summary based image description.

2.4.1 Keyword-based Image Description

Keywords based image description is the task of taking in an image and generating relevant descriptive keywords that describe the visual content of the image [3].It is an important area of research with applications such as tagging, indexing, and retrieval. Keyword-based description can be used to approximate the content of the query image and as a source of content words for generating an output summary based description. The Computer Vision approaches to this task, using image-understanding techniques to select descriptive keywords for a query image. A survey of methods for keywords based image description generation is found in [6].

In addition to Computer Vision approaches, there is research using Natural Language Processing to discover visually descriptive keywords. Text processing is computationally less expensive than image processing and may provide information that is difficult to learn visually. Instead of selecting descriptive keywords according to a visual-to-textual representation dictionary, descriptive keywords can be extracted from human write description text that is corresponding with the image. Most commercial image search websites

use the surrounding text as a source of information for understanding the content of an image [1].

2.4.2 Summary based Image Description

Describe images using natural language descriptions rather than keyword length descriptions. Which describes the relevant content in the image. Natural language descriptions are helpful for describing the relationships between objects in images, or for describing images to humans. The image description is the output of a complex process, which involves understanding the query image, grounding the visual representation to a semantic representation of what is relevant in the image, and the natural language generation of the output description. *Figure 2-1* shows an image description system. However, the exact formulations of the image description task vary across image description approaches.

2.5 Image Description Generation Tools

2.5.1 Lucene

Lucene is an open-source, highly scalable text search-engine library available from the Apache Software Foundation implemented in Java [11]. Documents are the primary items to be indexed and retrieved in Lucene. It is different from the notion of the document as a file. Each document object is made up of one or more field objects. Each field object is a name and value pair. Since Lucene is mainly for text document indexing and searching, we used LIRe library to wrap the image's visual features like text and create Lucene document.

2.5.2 LIRE

lucene Image Retrieval library is a lightweight and easy to use Java library for visual information retrieval systems [11]. It builds on the well-established text search engine Lucene. LIRe creates a Lucene index of image features for content-based image retrieval that is (color and texture characteristics) using local and global state-of-the-art methods. LIRe is actively used for research, teaching, and commercial applications. Due to its modular nature, it can be used on process level (e.g. index images and search) as well as on image feature level. Developers and researchers can easily extend and modify LIRe to adapt it to their needs. LIRe will provide us with global and local image feature extraction that we will use for similarity computation.

2.5.3 Stanford Core

Stanford CoreNLP [12] is a popular Natural Language Processing toolkit supporting many-core NLP tasks. That provides a set of human language technology tools. Documents are the primary items to be annotated by Stanford core NLP annotator. The general system works, raw text is put into an Annotation object and then a sequence of Annotators add information in an analysis pipeline. The resulting annotation, containing all the analysis information added by the annotators, can be output in XML or plain text forms.

2.5.4 SimpleNLG

SimpleNLG is an NLG system that allows the user to specify a sentence by giving its content words and its grammatical roles (such as subject or verb). The specification can be presented at different levels of detail. It is a surface realizer for a simple grammar and has significant coverage of English syntax and morphology. It automates several tasks, such as orthography, morphology, and grammatical realization[13]

Chapter Three: Related Work

3.1 Introduction

This chapter contains, review of related research works in the area of automatic image description generation. Different approaches used to solve these areas are also reviewed in this chapter.

3.2 Image Descriptions using manually created Database

Patrick Hède *et al.*[33] Presented an approach description generation for images without problems of occlusion and images of objects shot in a uniform background. Their system relies on a manually created database of objects indexed by an image signature (from color and texture characteristics) and two keywords (name and category). For input image, a signature is created and similar images are retrieved from the database by comparing the signatures. For the generation of descriptions in the natural language needs interpretation supplied by the indexing system. The descriptions are systematic, built using the information from the image processing. They have three categories of descriptors; the first category is the object names .the second category is the color information, the third category, find spatial information. Lastly, the number of objects and the general subject of the image are added based on object size. However, this approach is practical for only binary images or artificial images, no shadow, no reflected or refracted lights with a homogeneous background.

Moreover, generation is not a human-like description, not syntactically and linguistically generation. Full of noisy during content selection, word ordering and surface realization. Systematic construction of sentences often leads to errors. For instance, the structure color plus noun usually gives good results as in the “yellow banana” but sometimes leads to phrases that should be avoided as in “the orange orange”. The main object is chosen according to its surface, construction like “The table is under the bottle” which should be “The bottle is on the table. Ignore the vast amount of information available in the image, because they use uniform background.

3.3 Image Descriptions with Surrounding Text Documents

A few approaches were presented to combine knowledge in the form of text documents into the image. For each query image, the authors assumed that they are able to retrieve a related text document. The output description should contain a summary of information in the document that is relevant to the query image. This task formulation is similar to query-focused automatic summarization [30], but with an image serving as the focus for the output summary.

Ahmet *et al*. [1] Present image description generation methods for geotagged images that model image content using GPS coordinates of where the image was taken information, which is often recorded by cameras in mobile phones. Scene type (subject type) and place names can be obtained automatically given GPS coordinates. These systems retrieve text documents related to scene type, place name, and then generate brief summaries of those text documents.

The summarizer creates image descriptions in a three-step process. First, it applies text preprocessing, the given input documents. Then it applies some relevance representation to have qualities input for summarizer, to extracts features from the document sentences. The features used by the summarizer to score sentences in the documents are query similarity, cosine similarity over the vector representation of the sentence and the query. Centroid similarity, cosine similarity over the vector representation of the sentence and the centroid. Sentence Position, the first sentence in the document gets score 1 and the nth sentence as 1/n. Starter Similarity: If the sentence starts with a query term or the object type, it gets a score of 1, otherwise 0. LM Sim, the probability of generating a sentence using n-gram language model LM. Finally, it combines the features using a linear weighting scheme to compute the final score for each sentence and to create the final summary.

The strength of the system, it requires only GPS information associated with the image in order to generate descriptions. Other efforts towards the automatic generation of image descriptions based on the surrounding textual context of the image without consideration of image-related features such as color, shape or texture. However, their domain is limited to static objects such as buildings and mountains that are objects with determined geo-

coordinates. The approach cannot be applied to dynamic objects in daily life like people, cars, etc. Information graphics such as line graphs and plots exist in many documents, because documents may contain not only textual information but also visual data. However, the information contained in them is often not described in the document, and inaccessible to users such as the visually impaired. In these systems, the content of the image is determined by directly accessing the data used to generate the information graphic, as well as analyzing the associated text document.

Feng *et al*.[26] Developed descriptions generate system for news images using both extractive and abstractive summarization on the news articles that perform with each image. The authors use a joint model of visual and textual information to select relevant content. They integrate these models using a topic model based on Latent Dirichlet Allocation [5].A Latent Dirichlet Allocation topic model is trained on articles, images, and descriptions from the training set. Keywords are generated for an unseen image and article pair by estimating the distribution of topics that generates the test instance, then multiplying them with the word distributions in each topic to find the probability of textual keywords for the image. Once the relevant content is identified, they present methods for description generation using both extractive and abstractive summarization.

- Extractive summarization generation, to compute the similarity between the sentences and the description keywords generated by the annotation model, the following measures are used, word overlap, cosine similarity and probabilistic similarity [30].
- Abstractive summarization generation, word-based model and phrase-based model. The abstractive model defined over phrases gives better results than extractive methods[27]

To test their system they used BBC dataset created by [26]. The annotation keywords for the training set are generated by selecting descriptive words from the image descriptions. To address the problem of converting natural language into annotations, a large amount of preprocessing is performed. However, not all image contents accurately and obviously described by keywords. For instance, changes may also exist between query keywords and tag keywords that are query “car” would not return images annotated with “automobile”. Moreover, these approaches to image description are not all of the text in the related document

will be related to the visual content in the image. The approaches depend on extra data outside of the content of the image. These approaches are most applicable for specific domains (e.g., news, travel, financial reports) for which it can be assumed that these documents exist and can be retrieved in a structured way.

3.4 Image Description by Description Transfer from Similar Image

Ordonez *et al*. [9] presented the Im2Text model built using web-scale approach to image description generation. Retrieves an image, which is the closest visual match to the query image, and transfers its description to the query image. Visual matches are computed using a combination of visual object detectors (Deformable Part-Based Model [16]) this model represents each object by a collection of movable part like and scene-based descriptors [18]. For find, a description for an image using GIST [18] is a commonly used feature in Computer Vision, which coarsely localizes perceptual attributes (e.g. rough vs smooth, natural vs manmade). Thus, the extraction stage of their description generation process selects a sentence from the GIST nearest neighbor to the query image, to find the most similar image from a database of description images.

The approach is data-driven matching methods have shown to be effective for a variety of complex problems in the Computer Vision approach. These approaches reduce an inference problem for an unknown image to finding an existing labeled image, which is semantically similar. When transferring descriptions for query images do not have a corresponding text document available. One can reduce the description problem to finding a semantically similar description image, and transferring the existing description to the query image. However the approach for description generation without using natural language processing effort. In addition, they do not effort to compose a new description.

The authors present a new corpus, the SBU-Flickr dataset, which is made of 1 million images and human-authored descriptions uploaded by users of the website *flickr.com.*

. Due to its size, the SBU-Flickr dataset has enabled notable research in both Computer Vision and Natural Language Processing. However, the SBU-Flickr dataset is known to have many misalignments between images and description content, because Flickr users often use descriptions to describe background information about the image.

Farhadi *et al.* [2] present related approach with the above approach models the text more directly but is more restrictive about the source and quality of the human-written training data. Learn joint representations for images and descriptions, but can only be trained on data with very strong alignment between images and descriptions (because descriptions written by Mechanical Turkers). Both images and descriptions are mapped to an intermediate meaning space (object, action, and scene) and the results are compared. The triples (object, action, scène) of an image is predicted by solving MRF (Markov Random Field).The node potentials are computed as a linear combination of scores from several attributes and classifiers and the edge potentials are estimated by frequencies of node labels.

The approach build meanings space that comes between the space of sentences and the space of images. Similarity between a sentence and an image by mapping each to the meaning space (building scoring procedure) then comparing the results. However, retrieved sentences would be as descriptive of a test image as a generated one. This is because a retrieved sentence is constrained in terms of objects, action and spatial relationship between objects; whereas a generated sentence can, more closely associate the semantics relevant to a given image.

They build their own dataset of images and sentences around the PASCAL 2008 images that are selecting different images from each category that is the contribution of the approach is PASCAL sentence dataset. Which has been used by most of the subsequent approaches including our work.

3.5 Image Description by Integrating Visual and Linguistic Models

Kulkarni *et al.*[7] Uses CRF (conditional random field) based model to predict labeling for an input image, whose nodes correspond to image entities (such as objects, attributes, and prepositions) is used to predict the best labeling for an image of objects in a scene. The CRF integrates two types of potentials, image-based potentials, including object detectors, stuff detectors, attribute classifiers, and preposition function. Text-based potentials, a combination of parsing based potentials (from Flickr image descriptions) with the Google potentials. Joining both the detection scores with text-based potentials computed from large text corpora. Predicted labels are used to complete sentence templates, which provide a form for the generated descriptions, who use an HMM (Hidden Markov Model) based approach. In

addition to correcting noisy initial detections, the linguistic model can also be used to predict verbs and preposition words.

They make use of language models to predict function words that put together words in the meaning representation. However, this approach has one main limitation in the case of description generation, they use statistical language model to assigns a probability to a sequence of m words by means of a probability distribution. An n-gram language model uses only n-1 previous words to predict the next word. The n-gram language models is that they only capture local information about short-term sequences and cannot model long-distance dependencies between terms. Difficult to produce grammatically correct sentences using language models alone. This approach combines a separate sentence for each triple, independently from all other triples. Hence, it is unaware of discourse structure that means coherency among sentences.

3.6 Summary

This chapter presented research works in the area of image description generation that is important from our point of view. For these works, the limitation of the above work in comparison to the proposed system. Previous related approaches depend mostly on few object detectors, classifiers and corpus statistics. However, do not utilize the semantic information encoded in the available descriptions of images. Either they use these descriptions to restrict the set of prepositions or verbs, or pick one or more sentences that are complete and transfer them to a test image unaltered .So that might be result in quite verbose and non-humanlike descriptions. Moreover, it is very unlikely that a retrieved sentence would be as descriptive of a particular image as a generated one. This is because a retrieved sentence is constrained in terms of objects, attributes and spatial relationships between objects; whereas a generated sentence can more closely associate the semantics relevant to a given image.

Chapter Four: Automatic Image Description Generation Framework

4.1 Introduction

This Chapter describes the overall design of the proposed framework. As discussed in Section 2.2.1, 2.3.1 and 3.5, there are research gaps in both image understanding and description generation method. The underlining image description generation using object detector and classifier based is an issue need to be addressed. Therefore, an approach is proposed in this thesis to minimize the semantic gap is by integrating automatic image description generation to describe an image using natural language generation.

4.2 Framework

The main goal of designing this framework is to exploit to estimate the content of a new image based on its similarity with available images. To minimize the impact of encountering noisy and uncertain visual inputs, we extract syntactically motivated patterns from known descriptions and use only those for composing new descriptions. Where much relevant visual information cannot be captured by computer vision machine learning approaches like object detectors and classifiers.

In Computer Science, a framework is a conceptual structure indicating what kind of programs can or should be built and how they would interrelate and communicate [20].A framework generally provides some basic functionality that can be used and extend to make the more complex application. The main driving reason for proposing a framework in this thesis is to show fundament architectural structure and functionalities that should be included in image description generation systems for increasing the performance and bridging the semantic gap.

Our framework is composed of three major components, which is presented in *Figure 4-1* associated visual and textual features with similarity computation metric and language model. This is achieved through given dataset of images and their corresponding human-generated descriptions and our task is to describe an unseen image. We extract linguistically motivated phrases from these descriptions and given a new image, these phrases are then integrated to get structure of the form (((attribute1; object1); verb); (verb; prep; (attribute2; object2));

(object1; prep; object2))), that is the image is represented by using phrases (((attribute1; object1); verb); (verb; prep; (attribute2; object2)); (object1; prep; object2))). In the end, the description is generated using the phrases based on a fixed template. This framework component reasonably minimizes the semantic gap in CBIR systems. Designing this framework is for the achievement of human-like description generation for the unseen image.

4.2.1 High-Level View of Proposed Framework

High-level description of the proposed framework to increase the effectiveness of existing image retrieval systems and the proposed method for semantic gap bridging, which is automatic image description generation . The major components of the proposed framework are:

1. Image Engine
2. Search Engine
3. Text Engine.

The components are shown in *Figure 4-1* and are described in more detail below.

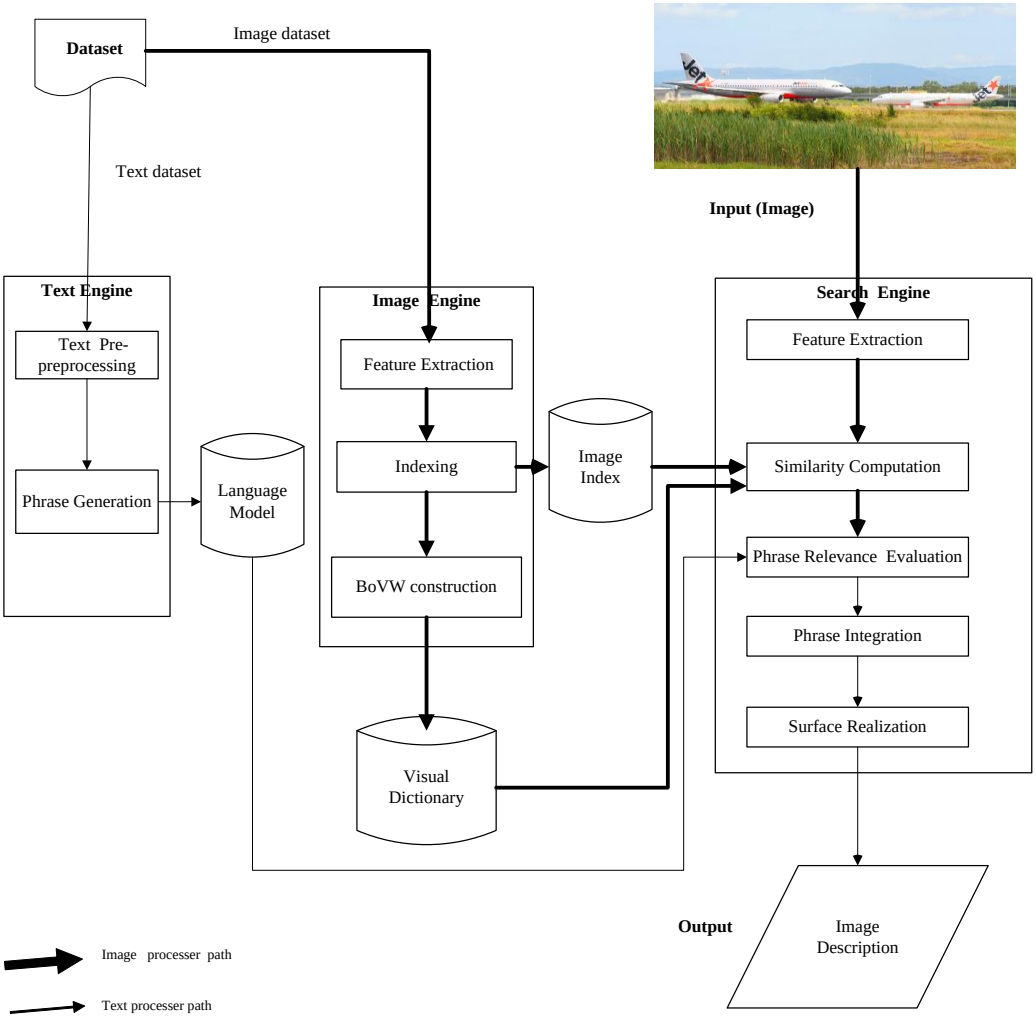


Figure 4-1 The High-Level Architecture of the proposed Framework

4.3 Image Engine

As described in the *Figure 4-1* the system architecture start with image engine it is the main component it uses visual information of images from feature extractor sub-component to build index and to generate bag of the visual word so, to get visual information. Initially, images are processed to extract certain types of global and local feature from each image then it will be converted to Lucene document format and this document are used to build the index. After images indexation is completed Visual dictionary are construction is done. In the next sub-sections, details of this component are presented.

4.3.1 Feature Extraction

This subcomponent extracts visual features from an image and represents it by its descriptors. In this work, five types of visual descriptors are used, color descriptors, texture descriptors, hybrid descriptors (a combination of color and texture) and visual words. A color feature is proven a very discriminative feature for image similarity search, and texture provides important information in image representation as it describes the content of many real-world images. Even though, they are often inconsistent due to variations in camera angle, orientation, camera viewpoint or change in illumination. Image's visual word is included in image representation in this framework because it is relatively resistant to such factors, variations in camera angle, orientation, camera viewpoint or change in illumination.

4.3.1.1 Low-Level Features

In this work, we use both visual descriptors (global descriptors and local descriptor) to take the advantage of global features and local features, since the sentence description should talk about all image content. *Table 4-1* shows the visual features used to index and search. These features are not randomly selected instead with detail study and experiment in order to incorporate different domains, which is a relatively better representative of an image [34]. In addition, features presented in the next sub-sections are used.

Table 4-1 Extracted Low-level features

Features	Feature Type	Feature Descriptor
Scalable color	Global feature	Color
Color layout	Global feature	Color
Edge Histogram	Global feature	Texture
JCD	Global feature	Texture and color
SIFT	Local feature	Visual word

Scalable Color Descriptor

The SCD (Scalable Color Descriptor) is color descriptor it is derived from a color histogram defined in the HSV (Hue, Saturation, and Value) color space with fixed color space quantization. It uses a Haar transform coefficient encoding[34] allowing scalable representation of description, as well as complexity scalability of feature extraction and matching procedures.

Its binary representation is scalable in terms of bin numbers and bit representation accuracy over a broad range of data rates. The SCD is useful for image-to-image matching and retrieval based on the color feature. Retrieval accuracy increases with the number of bits used in the representation. The histogram values are extracted, normalized and non-linearly mapped into a 4-bit integer representation, giving higher significance to small values. In order to use this descriptor to perform similarity retrieval, a matching function based on L_1 metric is used *Table 4-2*

Color Layout Descriptor

The CLD (Color Layout Descriptor) is also a color descriptor it captures the spatial layout of the representative colors on the image. Representation is based on coefficients of DCT (Discrete Cosine Transform). This is very compact and highly efficient in fast browsing and search applications. It provides image-to-image matching.

The Color Layout uses an array of representative colors for the image, expressed in the YCbCr color space. It is based on generating a tiny 8x8 thumbnail of an image, which is encoded

through DCT and quantized. The size of the array is fixed to 8X8 elements (block) to ensure scale invariance of the descriptors. The array obtained in this way is then transformed using the DCT, which is followed by zigzag re-ordering to group non-zero entries. A representation color was chosen for each block by averaging the values of the pixels in each block [4]. As well as efficient image matching, this also offers a quick way to visualize the appearance of an image, by reconstructing an approximation of the thumbnail, by inverting the DCT.

Edge Histogram Descriptor

Edge Histogram Descriptor captures the spatial distribution of edges, somewhat in the same spirit as the color layout descriptor. To compute the EHD (Edge Histogram Descriptor), a given image is first subdivided into 4 × 4 sub-images and local edge histograms for each of these sub-images are computed. Edges are broadly grouped into five categories: vertical, horizontal, 45°, 135° and neutral. Thus, each local histogram has 5 bins corresponding to the above 5 categories. The image partitioned into 16 sub-images results in 80 bins. These bins are non-uniformly quantized using 3 bits/bin, resulting in a descriptor of size 240 bits. Since edges play an important role for image perception, it can retrieve images with similar semantic meaning. Thus, it primarily targets image-to- image matching, especially for natural images with non-uniform edge distribution. However, EHD can be very sensitive to objects or scene distortions [34]. In order to use this descriptor to perform similarity retrieval, a matching function compute based on L_1 metric is used between two edge histogram *Table 4-2*

Hybrid Descriptors

Hybrid descriptors can be formulated by incorporating color and texture to a new descriptor. In this work, JCD (Joint Composite Descriptor) is used. A combined vector contains color and texture information at the same time. JCD successfully combines (CEDD) Color Edge Directivity Descriptor and (FCTHD) Fuzzy Color Texture Histogram Descriptor [34], which are both hybrid of color and texture descriptors.

SIFT Descriptor

As mention in *Table 4-1* the above descriptors are global descriptors. In this work local descriptor also used. A new approach proposed for extracting distinctive invariant features from images that can be used to perform reliable matching between different views of an object or scene [24]. This method has been called the SIFT (Scale Invariant Feature Transform), we adopted the work in [20].Extracting the visual word feature from images involves the following steps as shown in *Figure 4-2*[35] :

- Automatically detect key points (regions or points) of interest.
- Compute local descriptors
- Quantize the descriptor into visual words to form the visual vocabulary

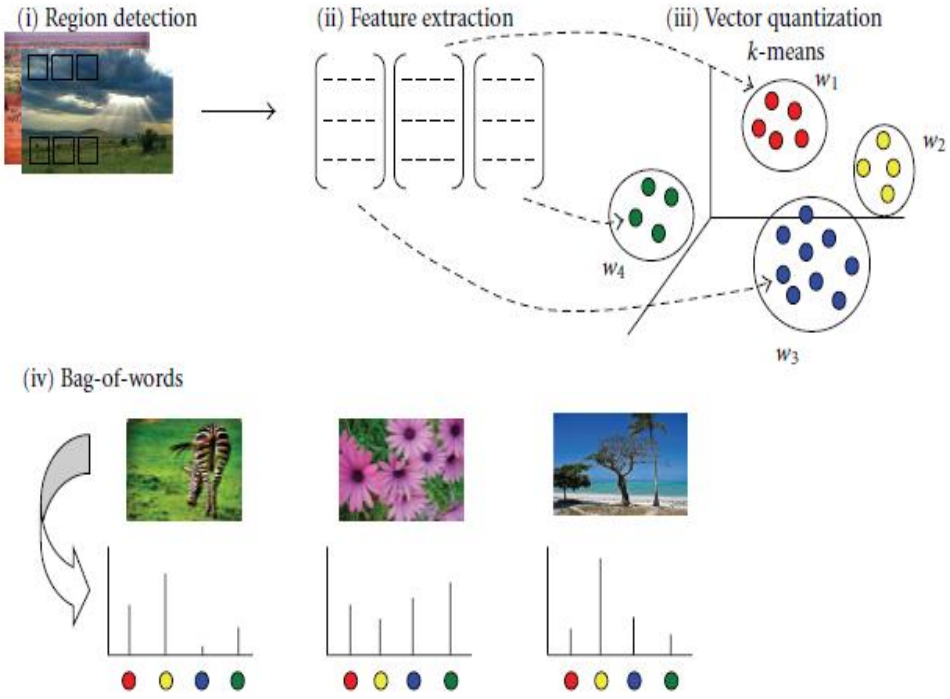


Figure 4-2 Visual Word Construction Steps

These steps are presented in detail in the next sub-section.

Interest Point Detection

The first step in the above *Figure 4-2* methodology is to detect local interest key points. This is done automatically by using an interest operator to extract information-rich patches from each image. Interest point detectors detect the "keypoints", the salient patches in an image. In this thesis, the scale-space extrema of DoG (Differences-of-Gaussians) [20] is used for the automatic detection of key points from images. The DoG algorithm searches over all scales and image locations to identify potential points of interest that are invariant to scale and rotation within a Difference-of-Gaussians pyramid.

A Gaussian pyramid is constructed from the input image by repeated smoothing and subsampling, and a DoG pyramid is computed from the differences between the adjacent levels in the Gaussian pyramid [35]. Then, Interest points are identified as local maxima or minima of the DoG images across scales. Each pixel in the DoG images is compared to its 8 neighbors at the same scale, plus the 9 corresponding neighbors at neighboring scales. If the pixel is a local maximum or minimum, it is selected as a candidate keypoint [20].

For each candidate key point:

- Interpolation of nearby data is used to accurately determine its position.
- Key points with low contrast are removed.
- Responses along edges are eliminated.
- The key point is assigned an orientation.

To determine the keypoint orientation, a HOG is computed in the neighborhood of the key point (using the Gaussian image at the closest scale to the keypoint's scale). The contribution of each neighboring pixel is weighted by the gradient magnitude and a Gaussian window times the scale of the keypoint. Peaks in the histogram correspond to dominant orientations. A separate keypoint is created for the direction corresponding to the histogram [35].

Local Descriptors

Feature representation methods deal with how to represent the patches as numerical vectors. These methods are called feature descriptors. A SIFT descriptor is deployed in this framework. SIFT is an algorithm for extracting local feature descriptors that are reasonably invariant to changes in illumination, image noise, rotation, scaling, and small changes in viewpoint [35].

Interest points for SIFT features correspond to local extreme an of DoG filters at different scale. Once a key point/interest orientation has been selected, the local feature descriptor is computed as a set of orientation histograms on 4×4 pixel neighborhoods. The orientation histograms are relative to the keypoint orientation, the orientation data comes from Gaussian image closest in scale to the keypoint's scale. Histograms contain 8 bins each, and each descriptor contains an array of 4 histograms around the key point. This leads to a SIFT feature vector with $4 \times 4 \times 8 = 128$ elements.

Describing the keypoint as a high dimensional vector. This vector is normalized to enhance invariance to changes in illumination [35]. After this step, each visual content is a collection of vectors of the same dimension (128 of SIFT) where the order of different vectors is of no importance.

Figure 4-3 [36] shows the overall SIFT based interest point detection and descriptors extraction method from an image.

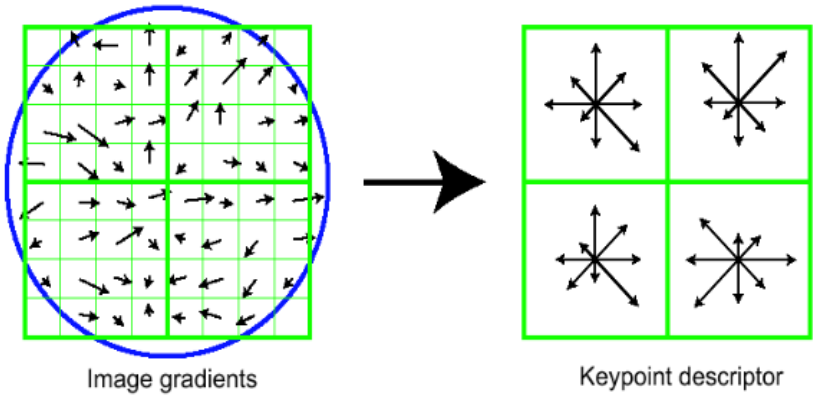


Figure 4-3 SIFT Descriptor

The general idea behind the algorithm is to first sample an image with the difference-of-Gaussians point detector at different scales and locations. Importantly, this detector is, to some extent, invariant to translation, scale, rotation and illumination changes. Each detected region is represented with a SIFT descriptor which is a histogram of edge directions at different locations. Subsequently SIFT descriptors are quantized into a discrete set of visual terms via a clustering algorithm such as K-means.

4.3.2 Image Representation

For an efficient combination of content-based (global and local features) image retrieval, we propose here an original image data model able to integrate different types of low-level features. Our approach for image representation considers both global and local representation models and Lucene Document representation structure.

Lucene is an open-source, highly scalable text search-engine library available from the Apache Software Foundation [37] . Documents are the primary items to be indexed and retrieved in Lucene. It is different from the notion of the document as a file. Each Document object is made up of one or more field objects. Each field object is a name and value pair. Given a document A represented by $A = (F1, F2, \dots, Fm)$, where F is the extracted visual information that describes A [38].

Since Lucene is mainly for text document indexing and searching, we used LIRe to wrap the image's visual features like text and create Lucene document [11].

We define the following representation structure for image indexing and searching

Image Model: An image is an object that has a visual component. It is formally represented in a model having four components *M (id, I, O, F).*

Where:

Id: is a unique identifier the image,

I: is a file path for identifying an image location,

O: is BoVW representation of the image object itself,

F: is a list of local and global feature vectors,

We define the contents of components F and O of *M* as F (Descriptor, Value):

- Descriptor: is the kind of representation such as Scalable color, Color layout, Edge Histogram, and JCD.
- Value: is a feature vector representation. It captures the content of the global features of an image that is required to perform the similarity measuring operations.

O (Descriptor, Value):

- Descriptor: is the kind of representation such as BoVWs.
- Value**:** is the feature vector extracted to represent the local features of the salient object that is a BoVW representation. It is used for similarity measuring operation on the salient objects.

With this schema, the model describes an image data in several levels of abstraction. An instance image *M* represented as: *M (id, O, F),*

Where,

- F is the extracted low-level feature vector. Specifically, Color layout, scalable color, edge histogram and JCD *Table 4-1*.
- O is the feature vector extracted to represent the low-level features of the salient object (i.e. bag-of-visual word vector).

4.3.3 Image Indexing

Once visual features are extracted, the next process is indexing the extracted features it is the initial part of all search applications. Its goal is to process the original data into a highly efficient cross-reference lookup in order to facilitate rapid searching. The indexing structure in this work is based on an image's visual feature. The effectiveness of the image similarity measuring system relies on images representation structure in the underline data source.

The indexer subcomponent uses the suggested image representation method to organize images in an index (Section4.3.1). The indexing structure we used is an inverted index [37]. *Figure 4-4* shows each step we have employed to build an index.

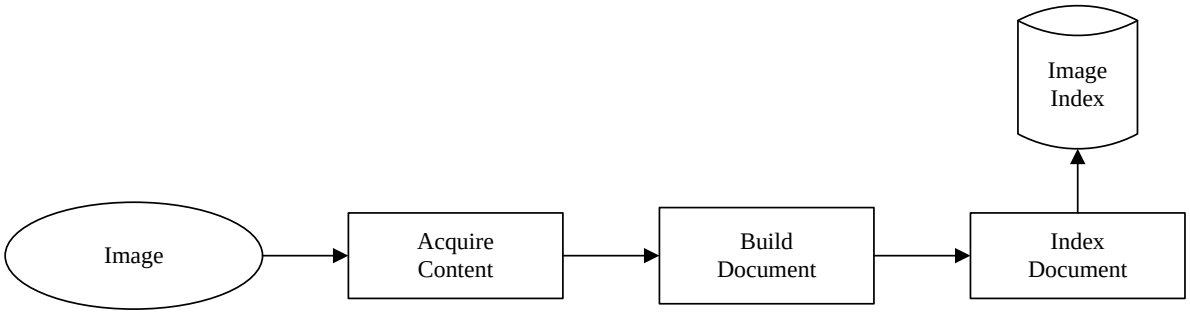


Figure 4-4 Indexing Step Employed

The first step of indexing is acquiring the content (i.e. extracting image feature). This process gathers and scopes the content that needs to be indexed. In this thesis work, the content of the image that needs to be extracted by Feature Extractor sub-component Section4.3.1 is a global and local feature. Both global features and local features provide one basic functionality to the searching visually similar features form unseen image features for estimating the content of the query image. The local features provide silent object-based similarity searching, which is constructed using SIFT and clustered using the K-Means algorithm.

The next step is to build documents out of the content that is image representation .The extracted features that need to be indexed has to be translated into the units (documents) used by the image retrieval system. Hence, the image representation model that we propose above is employed to build a document out of the extracted image features.

The final step is to index the document. During the indexing step, the document is added to the index. Once the index is created, BoVW dictionary construction is followed by processing bag-of-visual word field values of each.

4.3.4 Bag of Visual Words Construction

Once the indexing processes is completed, the next step is constructing Bag of visual words using BoVW constructor sub component of image engine that detects identifiable objects from a given image, using its visual descriptors. The main advantage of the BoVW model is its invariance to camera angle, image scale and orientation, as well as, occlusion, and lighting. In this work, we adopted the work in [20] .

Bag of Visual Words methodology was first proposed in the text retrieval domain problem for text document analysis, and it was further adapted for computer vision applications [23]. For image content analysis, a visual similarity of a word is used in the bag of visual word model, which is based on the vector quantization process by clustering low-level visual features of local region or points, such as color, texture, and so forth.

As shown *Figure 4-2*. After the key-points are detected and their local feature descriptors are extracted as we discussed above in Section 4.3.1 and Section 4.3.1.8 from all images in the training dataset .The final steps are visual words construction converts vectors representing patches (or local descriptors) to visual words that produces a BoVW represented in the form of vector (histogram).

In this thesis, the *K-Means* algorithm [39] is exploited to cluster the vectors. Each cluster is considered as a visual word that represents a specific local pattern shared by the keypoints in that cluster. The number of clusters is the bag of visual words' size. This representation is analogous to the bag-of-words document representation in terms of form and semantics because a BoVW representation can be converted into a visual word vector similar to the term vector of a text document.

In general, the *K-Means* clustering aims to partition or group *n* descriptors from all the training images into *k* clusters in which each descriptor belongs to the cluster with the nearest mean and the center of each cluster corresponding to a different visual word.

Suppose that we have *n* descriptors $X = (X_1, X_2 \ldots X_n)$, where each descriptor is a *d* - dimensional real vector. *K-Means* clustering aims to partition the *n* descriptors into *k* sets ($k \leq n$) $M = \{M_1 M_{2\ldots} M_k\}$ and find *k* centers (descriptors) in *M*. It operates as follows: it starts with randomly chosen descriptor *x* assigning to the value of cluster *m* that has the smallest Euclidean distance to the center. Then each cluster center is repeatedly recomputed until a steady state has been established or maximum number of iteration steps has been reached.

4.4 Search Engine

This component accepts unseen images as an input; it extracts relevant features of the unseen image using feature extraction processes. Then, the image representations are matched against the unseen image's features by performing similarity operation, and the matching images are returned. The main problem is to obtain a list of images, which are most "similar" in some aspects to the query image from a list of images indexed with their visual features. Solving this problem requires two things, feature extraction and similarity computation.

4.4.1 Similarity Computation

For retrieval of matching image, a similarity function (distance function) will be computed between the formulated unseen image and the images representative features in index. The most widely used distance functions in image similarity computation are those of the Minkowski family (or norm), which is which is employed over vector spaces. Such similarity measurement algorithms are distance similarity (such as, Manhattan Distance, Euclidean Distance) can be used depending on feature type [39].

For instance let us consider two low-level feature vectors $X = \{x_1, \ldots, x_n\}$ and $Y = \{y_{1,\ldots,} y_n\}$.In a vector space, the images are identified with *n* real-valued coordinate $X = \{x_1, \ldots x_n\}$ .Thus, the l_p distance function between two feature vectors X and Y is defined as

$$\text{Distance Function } (X,Y) = \sqrt[p]{\sum_{i=1}^{n} |xi - yi|p} \qquad (1)$$

According to the value assigned to *p,* we obtain family variations. The distances are additive, in the sense that each feature contributes independently to the measure of distance [40]. The formulas for three –Norms are shown in *Table 4-2*[39] .

Table 4-2 Lp –Norms or Minkowski Family

Family	Name	Equation
$P = 1$	Manhattan – Norm	D $(X, Y) = \sqrt[1]{\sum_{i=1}^{n} \lvert Xi - Yi \rvert 2}$
$P = 2$	Euclidean-Norm	D$(X, Y) = \sqrt[2]{\sum_{i=1}^{n} \lvert Xi - Yi \rvert 2}$
$P = \infty$	Maximum-Norm	D $(X, Y) = max_i^n = 1 \lvert Xi - Yi \rvert$

Given the unseen image and one image *I* in the index, having feature vectors f^U and f^I respectively. Once the image content is captured by the defined feature vector, the important thing is to determine the similarity measure between two images. Their visual similarity denoted as *VisualSim* $(U, I) \in [0,1]$is computed using distance function *Table 4-2* between their corresponding feature vectors. Value 0 means no similarity and value 1 identically. A higher distance value corresponds to a less similarity and a small distance results in a higher similarity. The overall searching performed following the steps presented in *Algorithm4-2*.The similarity measure formalized as:

$$VisualSim\ (U, I) = \sqrt{\sum_{i=1}^{n} \lvert fU - fI \rvert 2} \qquad (2)$$

This searching algorithm (*Algorithm4-2*) accepts unseen image visual descriptors, to perform similarity operation as an input and returns the most similar images accordingly their score values. Searching is being with by reading the indexed document as shown in line 2, which indicates the initial entry point of the index file that will be used by the linear searcher as shown in line 3. The linear searcher then reads documents from the index sequentially, compares them to the unseen image using the distance function presented in Equation 2, and returns a list of images.

```
Algorithm 2: Pseudo Code of Similarity Searching
Input U: Visual Descriptors//unseen image visual descriptor
      L: Integer // number of image to be retrieved
      F: Feature type // feature type for similarity
measure
Output: R[ ]:image
1.Begin
2.Index = Index Reader(I)// returns the initial index
entry point of indexed document
3.R =Linear Search(U, Index, f)
4.Return R
5.End of Algorithm
```

Algorithm 4-1 Pseudo Code of Similarity Searching

4.5 Text engine

The text engine component used textual information, which is obtained from text preprocessing task. After applying some text-preprocessing task like tokenization, part of speech tagging and parsing the next step is grammatical analysis. In this work we use Stanford Core NLP [12] ,which is not randomly selected instead with a detail study and experiment in order to incorporate different text preprocessing and grammatical analysis, which is relatively better.

4.5.1 Text Preprocessing

The text preprocessor component is depending on the application and type of document we use to apply some important text preprocessing tasks. Tasks are commonly decomposed into subtasks, chained together to form processing pipelines. The remaining error produced in these subtasks propagates, adversely affecting the end objectives. The tasks that we consider to apply tokenization, part of speech tagging and parsing.

4.5.1.1 Tokenization

The tokenizer component breaks of categorize stream of character to produce token. This component uses a different method like whitespace to make tokens. [12] Uses tokenizer that

was started as a PTB (Penn Treebank) tag set style tokenizer [18] .The tokenizer saves the character offsets of each token in the input text, as Character Offset Begin Annotation and Character Offset End Annotation. These tasks are the basis for other analyses. When text is divided on tokens, we can do further analysis of each token, specifically, part-of-speech tagging and parsing.

4.5.1.2 Part-of-speech Tagging

The part-of-speech tagger component perform determination of the part of speech for each token. The English taggers use the Penn Treebank tag set [18].The POS(Part-Of-Speech) Tagger assigns parts of speech to each token, or labels tokens with their POS tag such as noun, verb, adjective, etc., though generally computational applications use more fine-grained POS tags like 'noun-plural'. Part-of-speech tagger most approaches to sequence problems [27].POS model we use in this work is set to the English left3words POS model[12].

4.5.1.3 Parser

The parser component determines the parse tree (grammatical analysis) of a given tokens. It uses the output of former analyses. The grammar for natural languages is ambiguous and typical sentences have multiple possible analyses .In other words, it shows relations between words in the sentence. There are two primary types of parsing, dependency parsing, and constituency parsing (phrase structure tree) parsing. Dependency parsing focuses on the relationships between words in sentence making things like primary objects and predicates. Whereas constituency parsing focuses on building out the parse, tree using PCFG (Probabilistic Context-Free Grammar) [4]. The constituency has long dominated in the computational linguistics community it represents all description relationships uniformly.

The sample sentence when preprocessed through Stanford CoreNLP gives the following output. *Figure 4-5* shows that text preprocessing for the sample sentence, "A black and white dog is looking at the camera." (Snipping code from our text preprocessing generated NLP model).

```
[A 1, black 2, and 3, white 4, dog 5, is 6, looking 7, at 8, the 9, camera 10, . 11]
A/DT black/JJ and/CC white/JJ dog/NN looking/VBG at/IN the/DT camera/NN ./.

(ROOT
(S
(NP (DT A)
(ADJP (JJ black)
(CC and)
(JJ white))
(NN dag))
(VP (VBZ is)
(VP (VBG looking)
(PP (IN at)
(NP (DT the) (NN camera)))))
(. .)))
```

Figure 4-5 Text Preprocessing for the Sample Sentence

In brief, our model divided the sentence on tokens then analyzes POS of the tokens in the sentence. A POS tag is assigned for each token. Next, the syntactic phrases (chunks) in the sentence are identified the next token-level task can switch to syntactically motivated phrases.

4.5.2 Phrase Generator

The second sub component in text engine is phrase generator used to effectively use the semantic information encoded in the available descriptions. Rather than considering objects, attributes, verb, preposition from a sentence in a progressive manner. The syntactic relationships between the tokens are specified by using the previously analyzed results. In *Figure 4-6* dependencies are written as an abbreviated relation name (governor, dependent) where the governor and the dependent are phrases in the sentence to which a number indicating the position of the word in the sentence is attached.

For instance, the symbol "ROOT" to the token "looking" means that the token "looking " plays the central role when representing the sentence, and the subject (nsubj) of the action is dog and the object (dobj) of the action is a camera. (Snipping code for training dataset)

```
root(ROOT-0, looking-6) det(dog-5, A-1) amod(dog-5, black-2) cc(black-2, and-3)
conj:and(black-2, white-4) amod(dog-5, white-4) nsubj(dog-5, looking-6)
case(camera-9, at-7) det(camera-9, the-8) nmod:at(looking-6, camera-9) punct(dog-5, .-10)
```

Figure 4-6 Plain Text Representation of Typed Dependency

4.6 Phrase Representation Approaches

There are three alternatives to the phrase representation approaches (syntactic relationships) [12].The representations follow the same format in the plain text format *Figure 4-6* and in an XML format (Appendix II), which captures the same information.

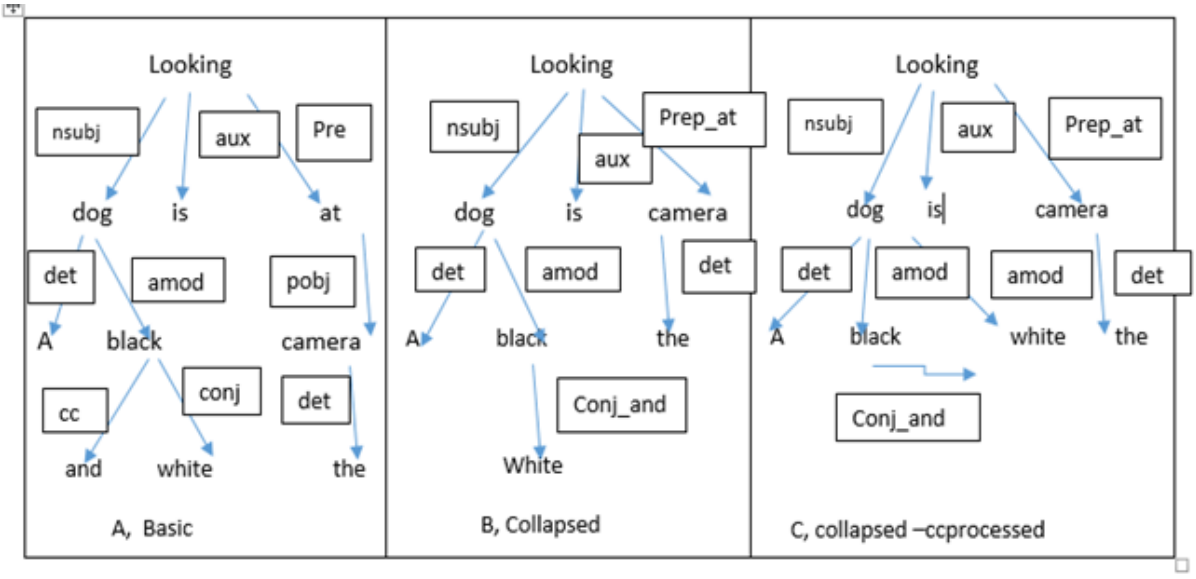


Figure 4-7 Graphical Representation of Phrases for Sample Sentence

The first column in *Figure 4-7* the *basic representation* approach forms a tree structure and there are no crossing syntactical relationship. Every token in the sentence except the head of the sentence is the dependent of some other token. The example sentence, looking is the head of the sentence.

In the second columns in *Figure 4-7* the *collapsed representation*, information about prepositions and conjuncts are combined to get a direct relationship between content words. This helps in the task of pattern extraction. For example, the syntactical relationships involving the preposition "at" in the above example is collapsed into one single relation:

Prep (looking, at) pobj (at, camera) Becomes prep_at (looking, camera)

The same rules are applied for syntactical relationships involving conjunction:

cc (black, and), conj (black, white) Becomes conj_and (black, white)

In the third column in *Figure 4-7 collapsed-ccprocessed representation*, syntactical relationship involving the conjuncts are propagated.

amod (dog, black), conj_and (black, white) Becomes amod(dog, black), amod(dog, white), conj_and (black, white).

In our work, we use *"collapsed-ccprocessed-dependencies"* that are useful for relation extraction tasks [12], as dependencies, function words are collapsed to reflect direct relation between content words. We extract relation phrases that carry bigger chunk of information compared to individual components of a sentence. To extract phrases first the available descriptions are preprocessed as shown in *Figure 4-5*.

We extract syntactically motivated phrases and represent each sentence to a list of phrases like (subject; verb), (object; verb), (verb; prep; object), (subject; prep; object). We look at an image as a collection of such phrases in the visual domain and hypothesize that similar-appearing images have identical phrases. Previous approaches obtain relations of the form (object; action; scene) [2], (object1 ; object2; verb; scene; prep) [10], or ((attribute1; object1); prep; (attribute2; object2)) [7] by combining the outputs of individual detectors with some corpus statistics to predict the complicated action and preposition. However, such predictions can be quite noisy *Figure 1*- (person; under; road), resulting in unsound sentences. In contrast, our phrases indirectly test ordering preference information, and hence generate semantically meaningful descriptions.

In practice, we extract seven types of phrases, (attribute, subject), (attribute, object), (subject, verb), (object, verb), (subject, prep, object), (object, prep, object), and (verb, prep, object).

We process various syntactical relationship types in the collapsed-ccprocessed-representation and consider the POS tags of words to generate phrases. For all phrase constituents except 'attribute', we consider the root form (lemma) of words while generating phrases. The following types of relational phrases are extracted using various tags.

4.6.1 Extracting Subject - Verb

Majority of these phrases are extracted from 'nsubj' (nominal subject) and 'nsubjpass' (passive nominal subject).

For image description, "A tour boat docked next to flowers." Sometimes 'amod' (adjectival modifier) it also gives this relation.

Gives Subject Verb Boat dock.

For the image description, "A father and a daughter looking at a horse through a chain link fence." amod (father, looking), and conj and (father, daughter)

Gives Subject verb father look daughter look

4.6.2 Extracting Object - Verb

Such relations are extracted using 'nsubjpass' (passive nominal subject) to handle Passive sentences. For the image description, "A Lemonaide stand is manned by a blonde child with a cookie. "Gives nsubjpass (manned, stand)

Object verb stand man

4.6.3 Extracting Subject-Prep-Object

Hence, we combine the information, given by nsubj and nmod next to relations as the governor of both these relations is same. From the image description, "A tour boat docked next to flowers." we also get the information that, the boat is next to flowers in the image.

nsubj (docked, boat), nmod next to (docked, flowers)

Gives subject prep object Boat next to the flower

4.6.4 Extracting Object-Prep-Object

Hence, we combine the information, given by dobj and nmod with relations, as the governor of both these relations is the same. For image description "Gray racing horse with green rider", dobj (racing, horse) and nmod with (racing, rider)

Gives Object prep object horse with rider

4.6.5 Extracting Attribute-Subject

Most of such phrases are extracted using 'amod' (adjectival modifier) tags. For the image description "A brown and white dog sits on a floral-patterned chair", amod (dog, brown) and amod (dog, white)

Giving Attribute subject brown dog

Attribute subject white dog

4.6.6 Extracting Attribute-Object

For the image description, "A brown and white dog sits on a floral-patterned chair." We Get relation. From amod (chair, floral-patterned)

Gives Attribute-object floral-patterned chair

For description without any verb, "A small green and yellow plane in the sky." we have following syntactical relations: amod (plane, small), amod (plane, green) and amod (plane, yellow)

Gives the following phrases

Attribute object small plane

Attribute object green plane

Attribute object yellow plane

4.6.7 Extracting Verb-Prep-Object

The majority of such phrases are extracted using, 'nmod' (prepositional modifier) relation. For description, "A tour boat docked next to flowers." nmod next to (docked, flowers)

Gives Verb prep object dock next to flower

4.7 Phrases Relevance Computing for Unseen Image

For any unseen image, our goal is associating phrases with it. *Figure 4-8* according to our hypothesis, an image inherits the characteristics of images similar to it. As we mentioned in Section 4.6 and *Algorithm 4-2*, image similarity is done based on the distance of the unseen image from any other images in the image training dataset.

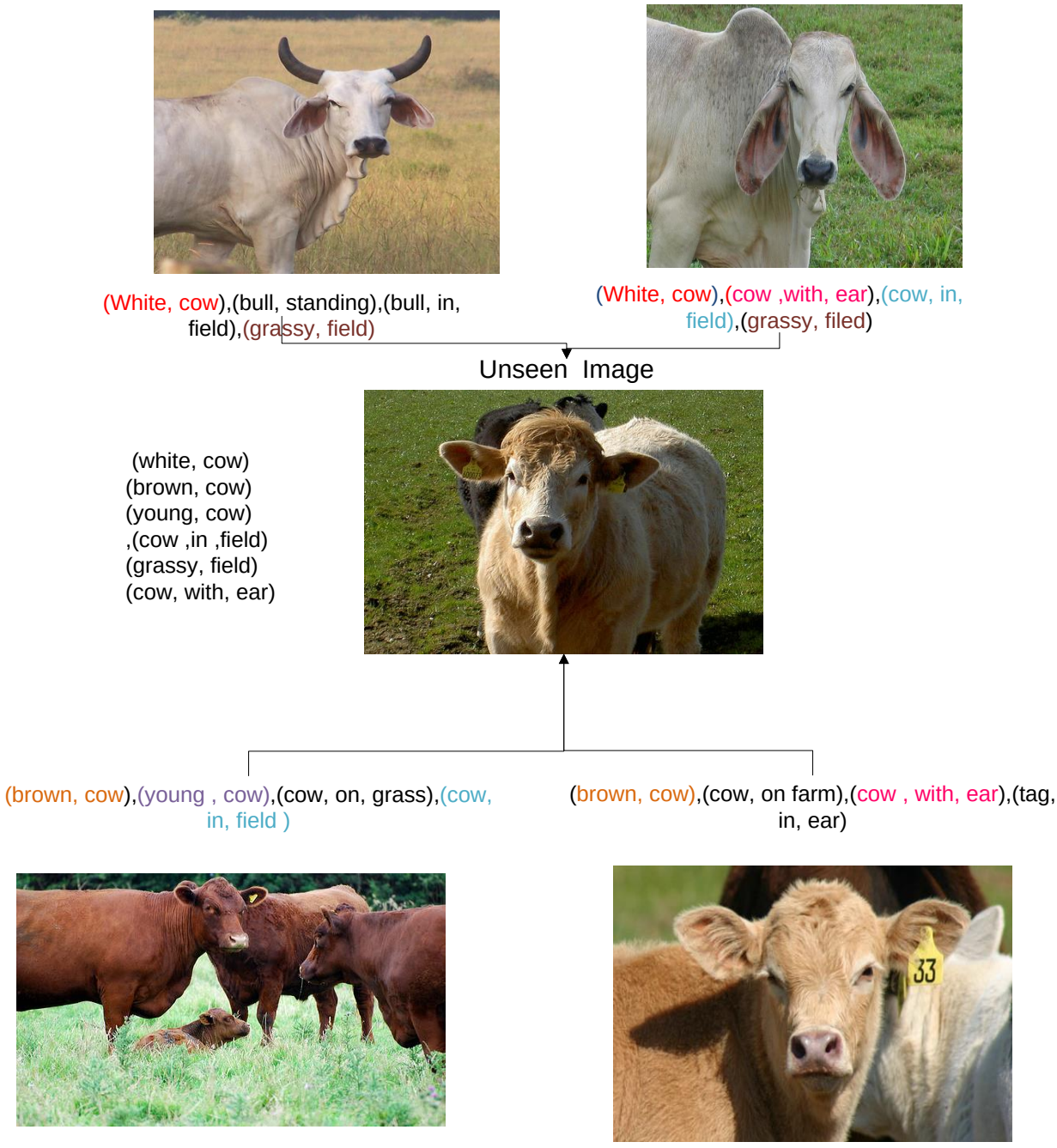


Figure 4-8 Image Inherits Characteristics of Similar Images

Algorithm 4: Pseudo Code of phrase relevance model

Input: phrases [ ]

Output: Relevant Phrases [ ]

1. **Begin**
2. **For each** phrase pattern
3. Counting phrases
4. **If** relevant phrase is equal to zero or one **break**
5. **For**(i==0; i++; i==5);
6. **For**(j==0;j++;j==4);
7. **If** relevant phrase count greater than or equal to two
8. Add new key phrase
9. **End For**
10. **End For**
11. **Return** phrases
12. **End of Algorithm**

Algorithm 4-2 Pseudo Code of Phrase Relevance Computing for Unseen Image

Computing the number of occurrence of phrases, we used the probability of a phrase in an similar images; it is recognizable that the high frequency phrases those with high probability in its relevance that will assign to the unseen images. The relevance based on phrase scoring approach, count the number of times each phrase appears across extracted phrases. It then reports the frequencies using a cutoff supplied that limits the output to just those phrases with a certain minimum frequency. Returns a collection of relevant phrases, defined as relevant phrases such as subject, verb, attributes, preposition, and object used for construction of the sentence.

4.8 Phrase Integration

This component used for phrases integration, during this step we look for phrase ordering elements in different phrases, as it was shown, for instance. {((subject), (attribute1, subject), (subject, verb), and (subject, prep, attribute2, object))}. These phrases are used for description generation in the next step. Unlike previous approaches, which first calculate the scores of different one-grams for an image (such as individual objects, then verbs and prepositions, scene.) in a progressive manner and then integrate them at a later stage to form a description. Our framework directly extracts phrases, which carry a bigger chunk of information. The design of the phrase integration process for forming a phrase ordering, which is an intermediate step of the image description propose framework.

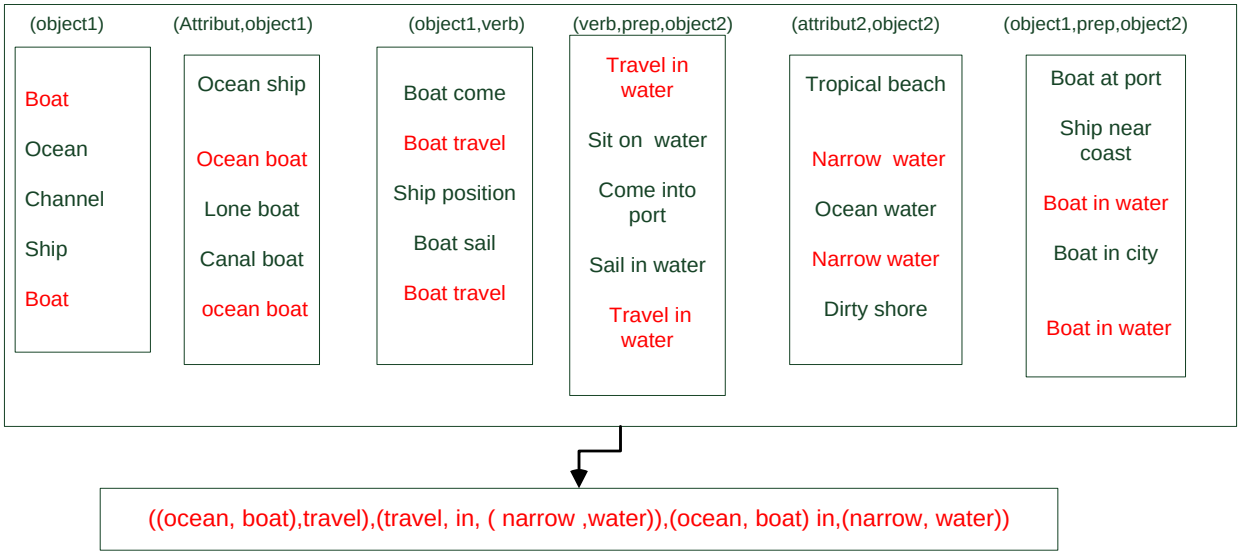


Figure 4-9 Graphical Design of the Phrase Ordering Process

4.9 Sentence Generation

This is the last component used for description generation. A text generator engine that converts syntactical structural representations into human-readable natural language descriptions. It is a linguistic process of constructing a description based on syntactical representation. It involves applying morphological and syntactic rules so that the output description sounds natural and correct. Moreover, the main function of this component is structure realization and Linguistic realization. Structure realization converts abstract

structures such as phrases in the format of (XML) to plain text format. Linguistic realization is the process of applying the rules of grammar to produce a description, which is syntactically, morphologically, and orthographically correct.

4.9.1 Surface Realization

The output of our phrase integration step is a structure of sentence constituents. Now the task is to generate a sentence. One major challenge in the generation is to determine the appropriate content. While [10] perform content selection(text planning) to deal with noisy inputs from detectors, [7] use n-gram frequencies for correct ordering preference of words. It is more natural to say “person in green” or “person wearing green” instead of saying “green person”. In our approach, phrases extracted from the human-generated descriptions. Hence, all such phrases are likely to be clean and relevant, and so we require no explicit content selection or word re-ordering. [7] Gets the wrong triple (person; under; road), whereas we get the correct spatial relationship between these two objects (person; on; road). Once we have determined the content of our sentences, the task of generation is to frame it into a grammatical form and output the result that verbalizes it (surface realization). Surface realization, in turn, requires choosing the right words (lexicalization).

For Linguistic Realization, we use SimpleNLG [13]. It is a surface realizer for simple grammar and has significant coverage of English syntax and morphology. SimpleNLG is a simple Java API, which gives users direct control over the realization process, designed to facilitate the generation of Natural Language. SimlpeNLG generate natural language string from a fully specified input.

A sample of the lexical entries from the lexicon used by SimpleNLG:

Word Element: base=give, category=VERB, realization=null, category=VERB, {features=is Ditransitive=true, present participle=giving, present3s=gives, intransitive=true, transitive=true, Past participle=given, past=gave.}

Word Element: base=John, category=NOUN, realization=null, category=NOUN, {features=proper= True, non-Count=false.}

Word Element: base=big, category=ADJECTIVE, realization=null, category=ADJECTIVE,

{features= is Classifying Adj=false, comparative=bigger, predicative=true, superlative=biggest, Is Color Adjective=false, is Qualitative Adjective=true}

4.9.2 Orthography

- Inserting appropriate whitespace in sentences,
- Implementing the rule that if the last word in the sentence ends with a period, do not follow it with another period. For example, generating the sentence "There are 52 states in U.S.A." instead of "There are 52 states in U.S.A..."
- Fit text into rows of fixed length by inserting line breaks between words (rather than in the Middle of a word)
- Format lists such as "penguins, ducks, and swan"

4.9.3 Morphology

Morphological synthesis is a process of returning one or more surface forms from a lexical from components that could be stored separately in different databases (lexicons). Changes the morphological form of a word to fit a syntactic context. For instance:

- Gender: This feature represents the gender of the subject of a noun phrase, or the object or subject of a verb phrase. It is most commonly used with personal pronouns. Example: he vs. she. It takes the following a set of values: FEMININE, MASCULINE, and NEUTER.
- Number: This feature is used to determine if the element is to be represented in singular or plural form. Example: dog vs. dogs, child vs. children.
- Tense: This feature represents the tense of a word by taking one of the following values: PAST, PRESENT, and FUTURE. Example: write vs. wrote vs. will write
- Person: This feature represents the first-person, second-person or third-person nature of the phrase. This predominantly affects pronouns such as I, you and they but some verbs will also be modified depending on the person of reference. For example, teaching is used as the present tense for the first and second person (I teach John and you teach Yosef) while teaches is used for the third person (he teaches Robel).

- Comparative and Superlative: This flag determines if the Adjective or Adverb should be inflected into the comparative or superlative form. Example: big vs. bigger vs. biggest.

4.9.4 Grammar

- Noun-verb agreement: Nouns and verbs need to agree on a number. They will be either singular or plural. Example: 'The dog chase the cat' is wrong, whereas 'The dog chases the cat' is correct. Similarly, 'I is a student' is incorrect. The correct sentence is 'I am a student'. SimpleNLG automatically changes the ending of the verb so that it agrees with the subject(s) of the sentence.
- Creating well-formed verb groups such as "does not like".
- Allow users to specify different parts of a sentence or phrase and set their features. SimpleNLG composes these parts into an appropriate syntactic structure.

As our sentence structure have a syntactically and linguistically motivated structure, their mapping to a sentence is straightforward using SimpeNLG. Its classes allow us to specify the subject of a sentence, the exact verb we want to appear in the sentence and the object. SimpleNLG methods are used to indicate the tense (present) and the aspect (progressive) of the verb. Once we have expressed the information about the content of a sentence in SimpleNLG terms, it assembles this information into a grammatically correct sentence.

A sentence is defined in terms of its syntactic constituents using SimpleNLG class Called **SPhraseSpec.**

A 'noun phrase'.is represented by class **NPPhraseSpec** represents. Methods are delivered for setting and getting the following constituents:

- Specifier (e.g. 'the')
- Pre Modifier (e.g. 'brown')
- Noun (e.g. 'horse')
- Post Modifier (e.g. 'on the sofa')

Class **PPPhraseSpec** defines prepositional phrase**.** Methods provided are:

- Preposition (e.g. 'on')
- Object (e.g. 'the sofa')

VPPhraseSpec defines a verb phrase. Methods used for this class are:

- PreModifier (e.g. ‘quickly’)
- Verb (e.g. ‘gave’)
- Indirect Object (e.g. ‘John’)
- Object (e.g. ‘a pen’)
- Post Modifier (e.g. ‘before exam’)

Some of the examples of realization from integrated phrases using SimpleNLG are:

- Phrases: ((continental; liner); park); (park; at; (small; airport)); ((continental; liner); at; (small; airport))

  Description: A continental liner is parked at a small airport.
- Phrases: ((several; man); sail); (sail; in; (small; sailboat)); ((several; man); in; (small; sailboat))

  Description: Several men are sailing in a small sailboat.
- Phrases: ((American; eagle); perch); (perch; on; (thick; rope)); ((American; eagle); on; (thick; rope))

  Description: An american eagle is perching on a thick rope.

Here, SimpleNLG has:

- Capitalized the first letter of the sentence
- Added the auxiliary ‘is’ and made it agree with the subject (In the second example, subject is plural, therefore SimpleNLG automatically converts ‘is’ to ‘are’)
- added -ing to the end of the verb (because the progressive aspect of the verb is desired) or constructed .passive form (when asked, as in first example)
- Put all the words together in a grammatical form
- Inserted the appropriate whitespace between the words of the sentence
- Put a period at the end of the sentence
- Extracting plural form of the word from the lexicon (In addition to the noun-verb agreement, an adjective must also agree with its noun or pronoun by matching its number. Singular nouns take singular adjectives, while plural nouns take plural adjectives. In the second example, the attribute of the subject (‘several’) is plural,

therefore we set Feature. A NUMBER of the subject (‘man’) as PLURAL and We get the correct plural form ‘men’ in the realization).

Though the descriptions generated in our case follow a template-like structure, the use of SimpleNLG saves the manual effort of writing individual templates. Our description of generation approach is domain -independent.

Algorithm 5: Pseudo Code of sentence generation

Input: linguistic constituents of the sentence (verb, subject) and also linguistic features (plural subjects from this information the realizer has constructed the actual sentence//

Output: generated sentence

1. **Begin**
2. **For each** linguistic constituent of the sentence
3. **Syntactic realization** //using grammatical knowledge to choose inflections, add function words and also the order of component for example in English the subject usually precedes the verb//
4. **Morphological realization** //computing inflected forms, for example, the plural form of woman is women not womans //
5. **Orthographical realization** //dealing with casing punctuation and formatting
6. **End For**
7. **Return** generated sentence
8. **End of Algorithm**

Algorithm 4-3 Pseudo Code for Sentence Generation

Chapter Five: Experiment and Evaluation

5.1 Introduction

This chapter presents the prototype developed to validate the proposal made in this work. The ideas and algorithms previously presented in Chapter 4 were implemented in a prototype system. The system's user interface allows non-expert and expert users to more easily interact with the system and to make the generation process more semantically. This chapter begins by describing the development environment employed to implement the prototype system and main features. Then describes the dataset used for testing, experimental result, and evaluation of the system performance. The chapter ends with running example of the system for different unseen images.

5.2 Development Environment

The implementation is carried on the Microsoft Windows operating system (i.e. Windows 8.1 with Service Pack 1). NetBeans IDE (version 8.2), an IDE (integrated development environment) is used for writing, compiling, testing, and debugging applications on the JDK (Java SE Development Kit) 8.1 platform. Lucerne (version 3.4.0) is used to organize the extracted image's features in an index and to facilitate searching with query image only. LIRe(version 9.0) for image's feature extracting and wrapping it to textual document, and indexing and searching images with it . For phrase extraction, we use StanfordcoreNLP (version3.9.1). SimpleNLG (4.4.8) for sentence generation. The selected programming language for implementing the system in Java. The following are major reasons why java is selected for developing the system. Many open-source reusable components are available (open-source Java libraries).It is a platform- independent and experience of the language.

5.3 Prototype

Components of the prototype are implemented as classes in the Java programming language. Some of the major components are discussed below:

Feature Extractor: As indicated in Section 4.3.1 this component responsible for extracting an image's visual features. The extracted feature is used by indexer component for organizing the extracted feature in an index Section 4.3.3

Image Indexer: This component aimed to build an index for keeping all the necessary information about the images in the dataset. This includes both local and global features Section 4.3.3

Similarity Searcher: This component aimed to search similar images from the index for given an unseen image; searched to retrieve the images most similar to it. Image matching is implemented using content descriptor (color, texture, hybrid and local features).The similarity measures such as distance function computed for image similarity. Finally, the result images are retrieved Section4.4

Text Preprocessor: This component aimed to text preprocessing that is breaking the textual data into tokens and performs operations on them, like tokenization, lemmatization, part of speech tagging and parse Section 4.4.1

Phrase Generator: This component aimed to be extract syntactical motivated phrases from text preprocessor **(**Section 4.5.2) .The syntactic relationships (dependencies) between the tokens are specified by using the previously analyzed results that is from Text Preprocessor component.

Sentence Generator :This component is aimed to construct the sentence using the phrases in phrase integration stage this linguistic process is done by applying surface realization processes (Section4.7)

5.4 Dataset

We use the UIUC PASCAL Sentence dataset to test the performance of our approach. Pascal Sentence dataset [41] it contains 1000 images taken from a subset of the Pascal-VOC (Visual Object Classes) 2008 challenge image dataset and is hand-annotated with sentences that describe the image by professional human annotators using Amazon Mechanical Turk [42].

(http://vision.cs.stonybrook.edu/~vicente/sbucaptions/)

Figure 5-1 shows some sample images with their descriptions. There are five descriptions per image, and each description is usually short around 15 words long. The dataset is available at *http://vision.cs.uiuc.edu/pascal-sentences/.*

We partition the dataset into image dataset and textual dataset. Also the image dataset partition in to two test dataset and training data set from training image dataset, we extract local and global features using 5-feature descriptors then index those features. From the textual dataset, we apply important text preprocessing tasks; extract all possible phrases from the preprocessed tokens.

	One-jet lands at an airport while another takes off next to it. Two airplanes parked in an airport. Two jets taxi past each other. Two parked jet airplanes facing opposite directions. two passenger planes on a grassy plain
	A man on a dirt bike jumping very high. A man on a motorbike jumping with the sky behind him A motocross bike with rider flying through the air. Motocross rider in mid-air Person on motorbike in midair at sunset.
	A bright red couch in a room with mostly wooden furniture. A living room with a large red sofa behind a small wooden table. A red couch with plaid pillows rests against a white wall. Red sofa in den or living room. The old red couch looks handsome in the den.

Figure 5-1 Sample Images with Corresponding Description from PASCAL Dataset

5.5 Evaluation of Experimental Results

This section introduces and discusses the experimental results using PASCAL dataset [41].The training set is composed of 900 images along with their text description. The test set for evaluating system performance is composed of 100 images (5images per class). Training images are processed and extracted features are described using color layout, edge histogram, joint composite descriptor, and scalable color. Then organized the extracted features in the index. For local feature descriptor, we build BoVW descriptor from index, feature descriptors using SIFT algorithm and a K-means algorithm is utilized to cluster the visual words based on the extracted key points. For image similarity computation and retrieval matching images for unseen image we use 5 methods of searching and based on threshold value (search more similar image having top score value).

PASCAL dataset has the following classes: *"aero plane ", "bicycle ", "bird " ,"boat", "bottle" ,"bus , "car", "cat", "chair", "cow",* "*Dining table*" *, "dog", " horse", "Motorbike", "Person" " Potted plant" ," sheep ", " sofa" , "Train"* and *"Tv monitor" Table 5-2.*

The performance of the prototype system is evaluated using statistical precision and recall method on the generated confusion matrix. The confusion matrix gives the full picture of our system performance in case of similar images retrieval. We adopt the guideline used by [20] for instance, a three class problem with the classes A, B, and C. A prototype may result in the following confusion matrix when tested on independent test data as shown in *Table 5-1*.

Table 5-1 Three Class Confusion Matrix

	A	B	C
A	Tp_A	e_{AB}	e_{AC}
B	e_{BA}	TP_B	e_{BC}
C	e_{CA}	e_{CB}	TP_C

The confusion matrix shows how similarity between unseen image and image training dataset is made in the prototype. The confusion matrix shows how estimations are made in the prototype. The rows correspond to the known class of the data; the columns correspond to the estimations made by the prototype. The value of each of element in the matrix is the number of estimates made with the class corresponding to the column for examples with the correct value as represented by the row. Thus, the diagonal elements show the number of correct categories made for each class, and the off-diagonal elements show the errors made.

Precision and recall computation using the confusion matrix is described as follows. Precision is a measure of accuracy if a specific class has been estimated. It is defined by:

$$Precision = \frac{\text{TP}}{\text{TP+FP}} \tag{2}$$

Where *TP* and *FP* are the numbers of true positive and false-positive results for the considered class. In the confusion matrix above *Table 5-1*, the precision for the class *A* would be calculated as:

$$Precision = \frac{\text{TPA}}{\text{TPA} + e_{BA+e_{CA}}} \tag{3}$$

Recall is a measure of the ability of a system to select instances of certain classes from a data set. It is commonly also called sensitivity and corresponds to the true positive rate. It is followed in as follows:

$$Recall = \frac{\text{TP}}{\text{TP+FN}} \tag{4}$$

Where *TP* and *FN* are the numbers of true positive and false negative calculations for the considered class. *TP +FN* is the total number of test examples of the considered. For class A in the matrix *Table 5-2*, the recall would be:

$$Recall = \frac{\text{TPA}}{\text{TPA}+e_{AB+}\, e_{AC}} \quad (5)$$

We have constructed a confusion matrix for each class used in test data *Table 5-2* this is performed by providing separately each class of test data to the measurement and recording the calculation result of each instance from the trained prototype.

Table 5-2 Confusion Matrix on test Images

	Aer	Bic	Bir	Bot	Bos	Bur	Car	Cat	Ch	Co	Di	Do	H	M	Pe	Po	Sh	So	Tr	Tv
Aeroplane	3	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bicycle	0	3	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0
Bird	0	0	4	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0
Boat	1	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bottle	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
Bus	0	0	0	0	0	4	0	1	0	0	0	0	0	0	0	0	0	0	1	0
Car	0	0	0	0	0	1	4	0	0	0	0	0	0	0	0	0	0	0	0	0
Cat	0	0	0	0	0	0	0	3	0	0	0	1	0	0	0	0	0	0	0	0
Chair	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	2	0	0
Cow	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	1	0	0	0
Diningtable	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	1	0	0	0	0
Dog	0	0	0	0	0	1	0	0	0	0	0	4	0	0	0	0	0	0	0	0
Horse	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	1	0	0	0
Motorbike	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	1	0	0
Person	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	1	0	0
Pottedplant	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0
Sheep	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	4	0	0	0
Sofa	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0	0	3	0	0
Train	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	1	0	0	3	0
Tvmonitor	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4

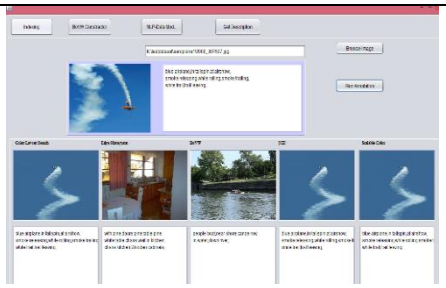
As explained above the diagonal elements of the confusion matrix represent the necessary similarity values and have the highest value. For the remaining columns, the similarity value is less and that shows wrong similarity measurement.

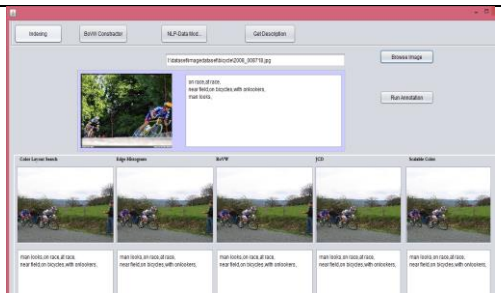
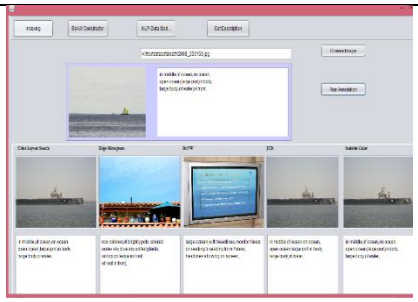
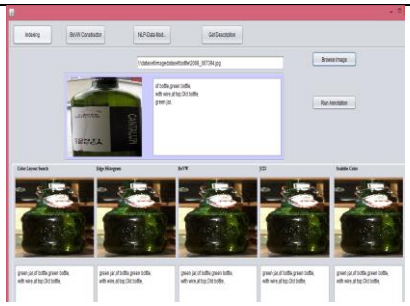
Precision and Recall values computed on each class considered for testing it is computed using equation 4 and 3.Generally, high precision is achieved, with up to 75% for the "*aeroplane, bird ,boat, bus, chair ,cat, car, cow, dog, sheep, sofa, person Tvmonitor, motorbike, dinning table,"* .The image retrieval produced fewer similarities for the classes "*bottle, bicycle, ,horse, , train, potedplant"* .This is mainly because of high number false positive estimated by the similarity computation system.

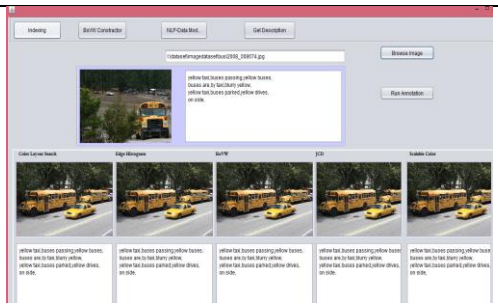
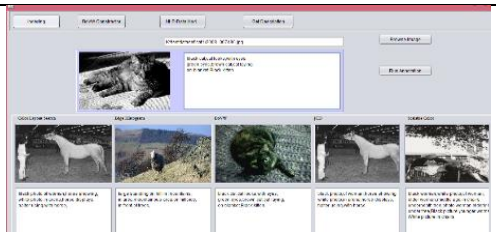
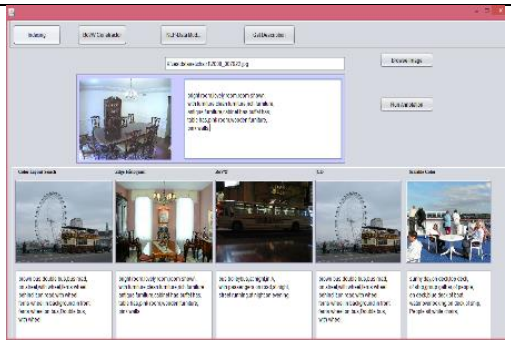
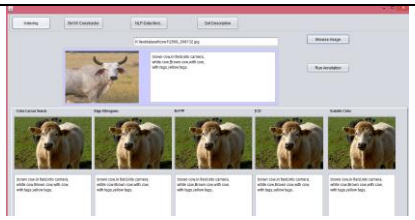
Moreover, we compute the overall accuracy of the similar images retrieval system by taking the separate test class's data together as single file. The overall accuracy of the retrieval result is computed as the ratio of combined precision value over the considered test data as shown in Equation 6 .The overall accuracy is 75%, which indicates 75% of test data from each class instance are correctly measured by the prototype and the rest 25 % are wrongly measured.

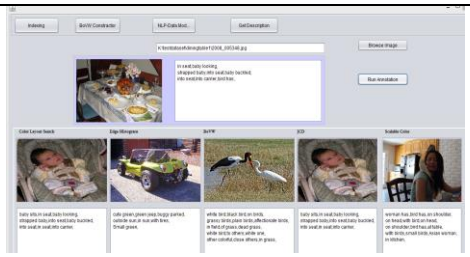
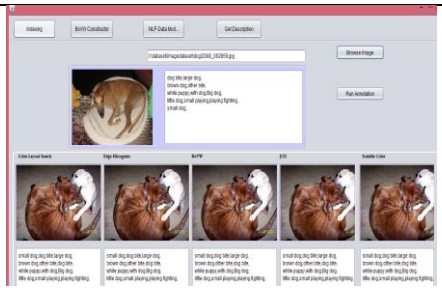
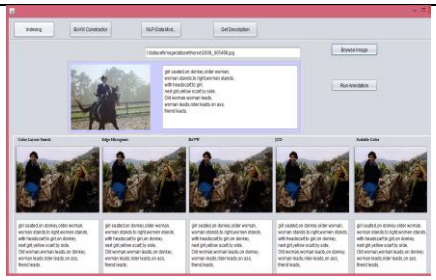
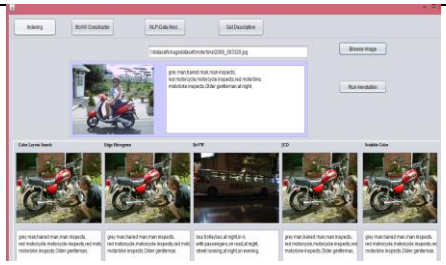
$$Accuracy = \frac{\#\text{correctly estimated data}}{\#\text{Total testing data}} * 100\% \quad (6)$$

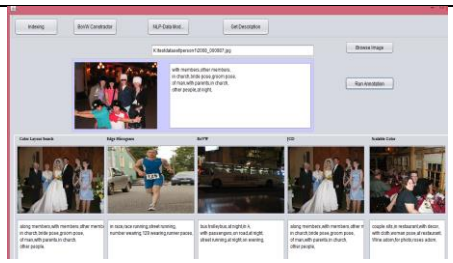
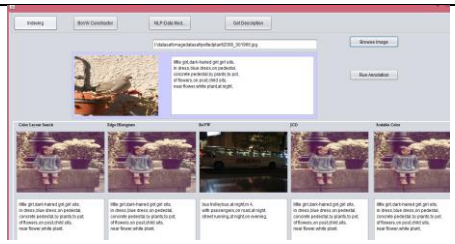
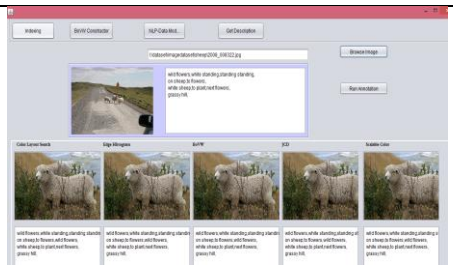
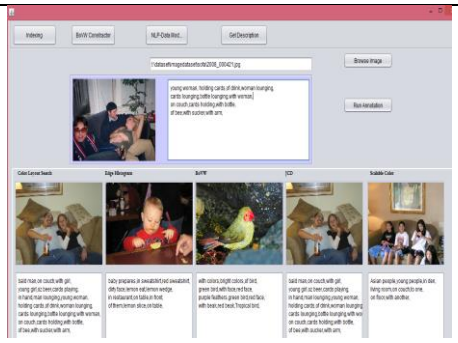
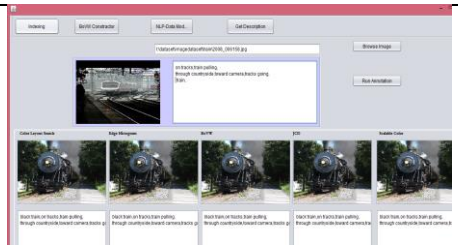
Table 5-3 Similarity Matrix on Sample Images

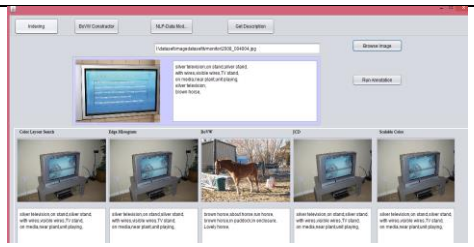
Categories Names	**Methods of Similarity Searching**	**Sample Images Retrieved for Unseen Image (Similarity Matrix)**	**Precision**	**Recall**
Aeroplane	CLD EH BOVW JCD		0.75	0.8

	Scalable color			
Bicycle	CLD EH BOVW JCD Scalable color		0.8	0.67
Bird	CLD EH BOVW JCD Scalable color		0.75	0.8
Boat	CLD EH BOVW JCD Scalable color		0.8	0.8
Bottle	CLD EH BOVW JCD Scalable color		0.8	1

Bus	CLD EH BOVW JCD Scalable color		1	0.67
Car	CLD EH BOVW JCD Scalable color		1	0.8
Cat	CLD EH BOVW JCD Scalable color		0.6	0.4
Chair	CLD EH BOVW JCD Scalable color		0.6	06
Cow	CLD EH BOVW		0.8	0.8

	JCD Scalable color			
Dining table	CLD EH BOVW JCD Scalable color		0.75	0.67
Dog	CLD EH BOVW JCD Scalable color		1	0.8
Horse	CLD EH BOVW JCD Scalable color		0.8	0.8
Motorbike	CLD EH BOVW JCD Scalable color		1	0.8

Person	CLD EH BOVW JCD Scalable color		1	0.8
Potted plant	CLD EH BOVW JCD Scalable color		1	0.8
Sheep	CLD EH BOVW JCD Scalable color		1	0.8
Sofa	CLD EH BOVW JCD Scalable color		1	0.7
Train	CLD EH BOVW JCD		0.7	0.8

	Scalable color			
Tvmonitor	CLD EH JCD Scalable color		0.8	0.8
		Mean	0.8	0.75

5.4.1 Human Evaluation

To get a better understanding of the performance, we manually evaluate the approach at various stages. We provide human evaluators with a set of images and phrases generated from description, extracted attribute-object pairs, subject-verb pairs and object -preposition. The results on 50 images from each class of test dataset are shown in *Table 5-4*.

Table 5-4 Precision of Attribute-Object, Object- Verb and Object-Preposition Generation

Dataset	Attribute-Object	Object –Verb	Subject-Verb	Object-Preposition
PASCAL	80%	70%	70%	80%

To highlight the effect of attribute and object extraction on generation accuracy *Table 5-4.* This is a hopeful result as it highlights the importance of the correct object and attribute detection *Figure1-1* and for correct object and preposition *Figure 1-* the image description generation task. In addition, *Table 5-4* shows that incorrect object- verb and subject –verb phrase generation have least accuracy.

In the description generation system, in addition to evaluating our system using confusion matrix for image similarity computation and accuracy of phrase extraction generated description performance measure, human evaluation also becomes necessary. We collect human judgments on 50 images from the PASCAL dataset verified in human evaluation. Two aspects are verified in human evaluation: Readability of descriptions and Relevance of (generated) text with a given image. For simplicity, human evaluators assign one set of scores on a likert scale of {1, 2, 3}for each aspect per image, where 1 is good, 2 is ok and 3 is bad. We adopt the definition and guideline used by [43]:

Readability: How grammatically correct is the generated sentence?

1. Mostly perfect English phrase or sentence.
2. There are some errors, but mostly understandable.
3. Terrible.

Relevance: How relevant is the generated description of the given image?

1. Very relevant.
2. Reasonably relevant.
3. Totally off.

Before we develop the scale we need a range of numerical values, which could range from 1-3 it is range gives the weight of the responses. For 50 test images that is total respondents is 50 and the scale range is 1= good, 2= ok, and 3= bad. From our test data 30 images descriptions readability is good (30*1=30), 15 images descriptions readability is ok (15*2=30), 5 image description bad (5*3=15). Total score=30+30+15+75.points=75/50=1.5 .We can conclude that the respondents that is readability is ok because it falls within the range of 1 -2 is ok. Lower score means better performance .The score value shows *Table 5-5* below.

Table 5-5 summarizes our human evaluation results. The scores given by two human evaluators were identical on 78% of the instances on the test sets.

Table 5-5 Summarizes Human Evaluation Results for Sentence Generation

Dataset	**Readability**	**Relevance**
PASCAL	1.5	1.5

Table 5-5 shows the human evaluation results. In terms of readability and relevance, our goal in this work is to consider syntactical motivated phrases from human generation description approaches for generating image descriptions that can address limitations of key words based image description generation *Figure1-1*. That is phrase generation to sentence generation performs best than key words to sentence generation.

In our work, there are different sources of errors. Some errors are due to mistakes in the original visual similarity measure to estimate content of input image. For instance, *Table 5-3* in category name *"cat"* the similarity measure using CV approaches is good in BoVW descriptor but our phrase relevance computing is depending on counting of similar phrases so misalignment happen between CV approach and NLP approach.

5.6 Experimental Results

Figure 5-2 to *Figure 5-4* shows the interfaces of all functionality of Indexing, Bovw, NLP data model and get description using the proposed approach. The overall image description generation is performed following the approach presented in Section4.1

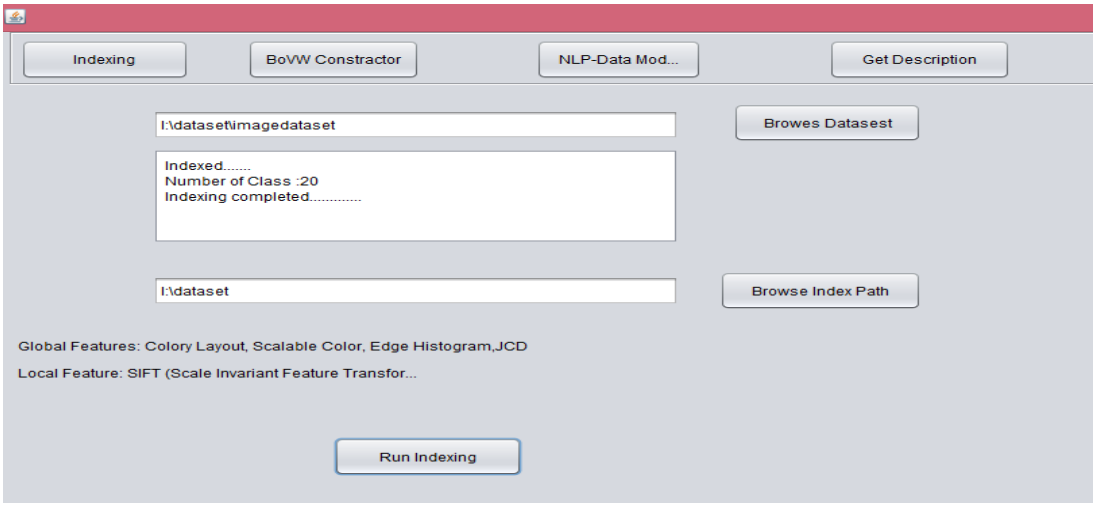


Figure 5-2 System Interface for Image Description Generation and Indexing Processes

The *Figure 5-2* Shows image description generation and indexing processes. The interface allows the user to select dataset used for indexing; directory path the indexed file put and used this indexed file for next processing step that is BoVW construction.

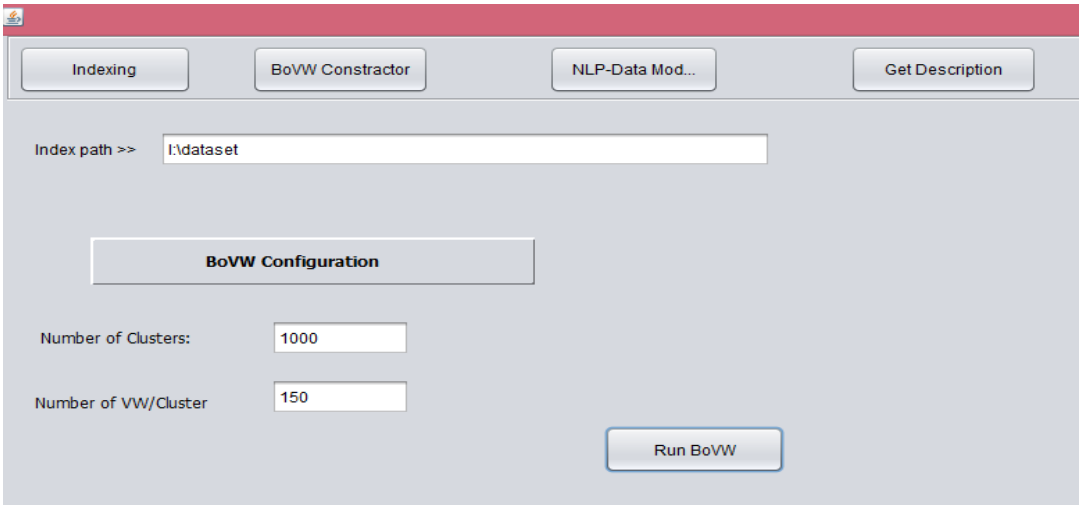


Figure 5-3 BOVW construction Interface for Image Description Generation

The *Figure 5-3* shows the system interface for BoVW construction .The interface allows the user to accept the indexed file directory from indexing processes and Bovw configuration these are number of clusters and number of visual word per cluster then run BoVW button is clicked automatically BoVW construction is performed based on Section 4.3.3

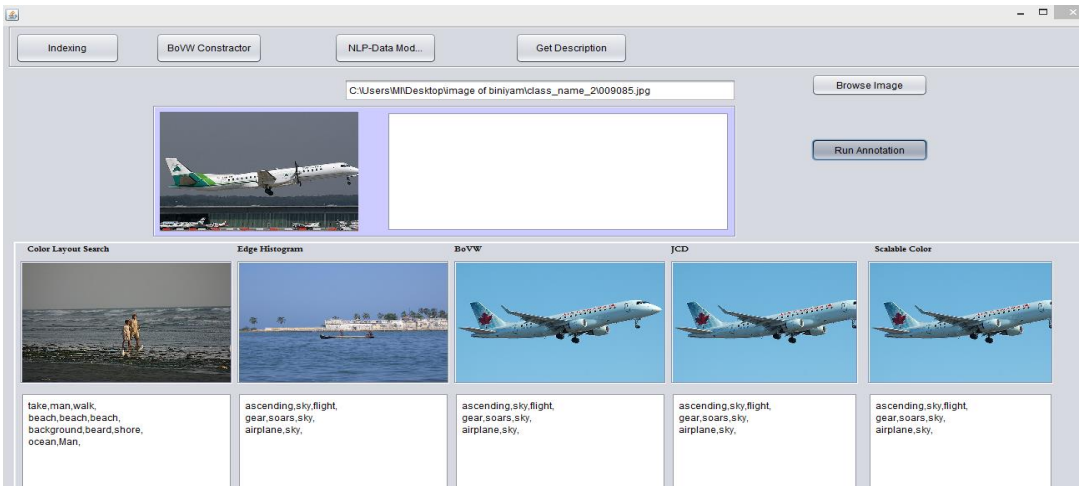


Figure 5-4 Interface for Similarity Searching for an Unseen image, Phrases Extraction

The *Figure 5-4* shows the system interface for an example image, airplane, used as input for similar images are retrieved automatically based on image representation structure using both global and local features descriptors and the associated phrases are extracted based on phrase extraction representation structure presented in Section4.5

Once Get Description button is clicked, color layout feature, scalable color feature, edge feature JCD features and BoVW features are extracted from the example image and the query engine component computes the similarity between the example image and images in the underlying index, using their mentioned features of both.

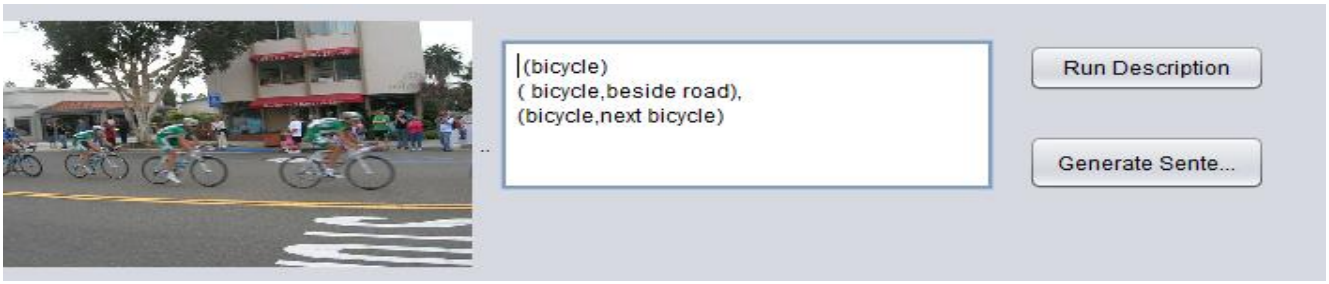


Figure 5-5 Screenshot of Sample Phrase Extraction Evaluation Correct Object, Attribute-Object, and Preposition Extraction

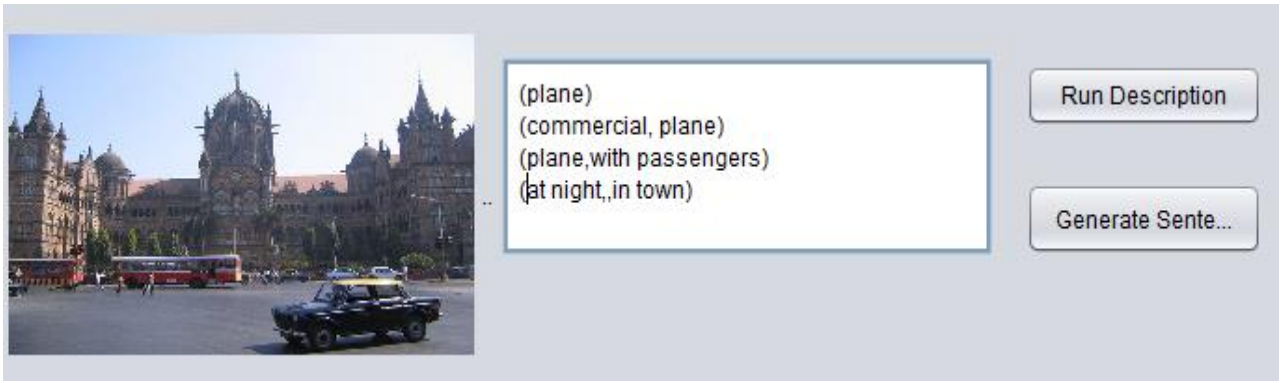


Figure 5-6 Screenshot of Sample Phrase Extraction Evaluation. Incorrect Object and, Attribute-Object Extraction

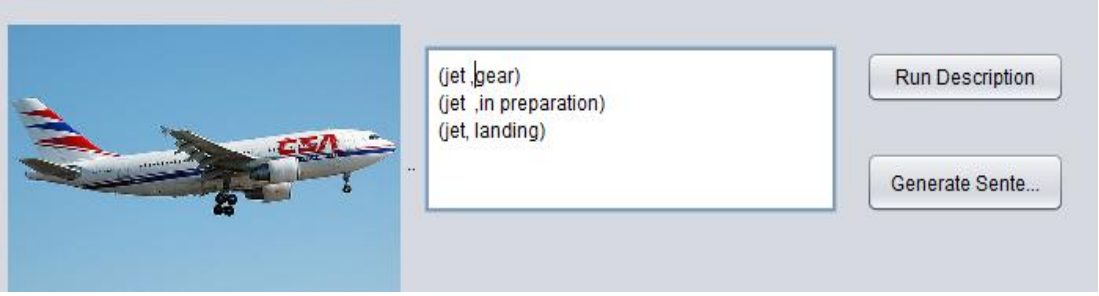


Figure 5-7 Screenshot of Phrase Extraction Evaluation. Correct Object, Attribute-Object Extraction, correct Verb and Preposition

Figure 5-8 Screenshot of Sample Phrase Extraction Evaluation. Correct Object, Attribute-Object, Verb and Preposition Extraction

Chapter Six: Conclusion and Recommendation

6.1 Conclusion

Generating of human-like description for unseen image is not easy to overcome. In order to overcome the well-known problem in semantic gap, sentence- based image description generation is the solution. Phrase extraction from human written description based on syntactical motivated structure and integrating those phrases composing new description for unseen image is the solution. However, difficulty is to make computer vision understand similar images for unseen images for estimating the content of unseen images and able to describe image using human language.

We proposed a novel approach for generating relevant, fluent and human-like descriptions for images without relying on any object detectors, and classifiers. We achieved significantly better results than state-of-the-art by analyzing and efficiently extracting the semantic information encoded in the image descriptions. Experimental results demonstrate that the quality of the generated descriptions is highly sensitive to the generation precision of objects and attributes present in an image. Human evaluation of each stage shows that generating verb has the least accuracy. One direction of future work would be to use some vision-based inputs in addition to corpus statistics (like Google. n-gram statistics) to generate the verb.

6.2 Contributions of the Work

To summarize, the following are the main contribution of this research work:

- Design general architecture and develop a sentence-based image description generation framework.
- Provide the way to develop automatic image description generation using natural language generation for other researchers.
- Producing state-of-the-art performance on the PASCAL sentence data set.
- This research paper is useful for anyone who has the interest to know and wants to work on the area of sentence-based image description generation.

- Use of semantic information encoded in the available image descriptions:

Previous approaches ignore the vast amount of information available in the image descriptions like the content of the image (objects, attributes, action, and scene). Image descriptions also tells us about the spatial relationship between various objects (preposition). Even for complex images, this information can be easily extracted from the descriptions.

6.3 Recommendations

As the research area includes Computer Vision, Natural Language Processing and Natural Language Generation together, developing such sentence based image description generation framework spends lots of time and efforts. However, to develop more efficient and improved effective Sentence based image description generation framework, additional features should be incorporated on this work. It is recommended that this research work can also further extend and enhanced by adding the following features.

- To search the possibilities of generating varied and interesting descriptions.by expanding each noun (subject/object) up to at most three hyponym levels using WordNet synsets.
- Adding some vision-based inputs in addition to corpus statistics (like Google. n-gram statistics) to generate the verb.
- Using deep neural networks for the task of image description [44], further progress in this area of research will likely lead to models, which provide richer representations of visual scenes.

References

[1] Ahmet and R. Gaizauskas, "Generating image descriptions using dependency relational patterns," *Proc. 48th Annu. Meet. Assoc. Comput. Linguist.*, no. July, pp. 1250–1258, 2012.

[2] A. Farhadi *et al.*, "Every Picture Tells a Story : Generating Sentences from Images," *Proc. 11th Eur. Conf. Comput. Vis.*, pp. 15–29, 2014.

[3] S. L. Feng, R. Manmatha, and V. Lavrenko, "Multiple Bernoulli Relevance Models for Image and Video Annotation," *Proc. Comput. Vis. Pattern Recognit.*, vol. 2, pp. 1002–1009, 2008.

[4] M. Guillaumin, T. Mensink, J. Verbeek, C. Schmid, and L. J. Kuntzmann, "TagProp : Discriminative Metric Learning in Nearest Neighbor Models for Image Auto-Annotation," *Proc. 12th Int. Conf. Comput. Vis.*, pp. 309–316, 2013.

[5] D. M. Blei and M. I. Jordan, "Modeling Annotated Data," *SIGIR Forum (ACM Spec. Interes. Gr. Inf. Retrieval)*, no. SPEC. ISS., pp. 127–134, 2009.

[6] A. Makadia, V. Pavlovic, and S. Kumar, "A New Baseline for Image Annotation," *ECCV*, pp. 1–14, 2014.

[7] G. Kulkarni and A. C. Berg, "Baby Talk : Understanding and Generating Image Descriptions," *Proc. Conf. Comput. Vis. Pattern Recognit.*, pp. 1601–1608, 2015.

[8] M. Mitchell *et al.*, "Midge: Generating Image Descriptions From Computer Vision Detections," *Proc. 13th Conf. Eur. Chapter Assoc. Comput. Linguist.*, pp. 747–756, 2014.

[9] Vicente Ordonez Girish Kulkarni Tamara L Berg, "Im2Text : Describing Images Using 1 Million Captioned Photographs," *Proc. Neural Inf. Process. Syst.*, pp. 1143–1151, 2014.

[10] Y. Yang, C. L. Teo, H. Daumé III, and Y. Aloimonos, "Corpus-Guided Sentence Generation of Natural Images," *2011 Conf. Empir. Methods Nat. Lang. Process.*, no. January, pp. 444–454, 2013.

[11] M. A. Riegler, K. Pogorelov, M. Lux, and M. Riegler, "LIRE - Open Source Visual Information Retrieval," *IEEE Trans. Image Process* ., no. May, 2016.

http://www.semanticmetadata.net/lire/ last accessed June 2017.

[12] M. De Marneffe and C. D. Manning, "The stanford typed dependencies representation," *Proc. Work. Cross-Framework Cross- Domain Parser Eval.*, no. ii, pp. 1–8, 2015.

http://nlp.stanford.edu/software/corenlp.shtml

[13] Albert Gatt and Ehud Reiter, "Simplenlg: A realisation engine for practical applications.," *Proc. 12th Eur. Work. Nat. Lang. Gener.*, no. October, pp. 90–93, 2014.

http://nlp.simplenlg.edu/software/nlg.shtml

[14] P. D. Rebecca Mason, "Data-driven Image Descriptions," *Proc. 12th Eur. Work. Nat. Lang. Gener. (ENLG)*, no. May 2016, pp. 90–93, 2016.

[15] V. Ordonez, G. Kulkarni, and T. L. Berg, "Im2text: Describing images using 1 million captioned photographs," *Adv. Neural Inf. Process. Syst.*, pp. 1–9, 2011.

[16] P. Felzenszwalb and D. Mcallester, "Object detection with discriminatively trained part based models," *Pattern Anal. Mach. Intell. (PAMI), IEEE Trans. on*, *32*, pp. 1627–1645, 2013.

[17] Y. Arabyat, "Histograms of Oriented Gradients for Human Detection," *Comput. Vis. Pattern Recognition,CVPR . IEEE Comput. Soc. Conf.*, vol. 5, no. 8, pp. 886–893, 2009.

[18] A. Oliva and A. Torralba, "A scene-centered representation of gist Remember the pictures The classical RSVP task," *Prog. Brain Res.*, vol. 155 B, pp. 23–36, 2010.

[19] G. Carneiro, N. P. Da Silva, A. Del Bue, and J. P. Costeira, "Artistic image classification: An analysis on the PRINTART database," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, pp. 143–157, 2014.

[20] K. Biniyam and G. Fekade, "Bag-of-visual words based automatic image annotation," *IEEE AFRICON Conf.*, 2015.

[21] C.-F. Tsai, "Bag-of-Words Representation in Image Annotation: A Review," *ISRN Artif. Intell.*, vol. 4, pp. 1–19, 2014.

[22] Y. Liu, D. Zhang, G. Lu, and W. Y. Ma, "A survey of content-based image retrieval with high-level semantics," *Pattern Recognit.*, vol. 40, no. 1, pp. 262–282, 2009.

[23] Kraisak.k, “Multi modal multi-semantic image retrieval,” *Sch. Electron. Eng. Comput. Sci. Queen Mary, Univ. London*, 2013.

[24] K. Kesorn and S. Poslad, “Semantic restructuring of natural language image captions to enhance image retrieval,” *J. Multimed.*, vol. 4, no. 5, pp. 284–297, 2010.

[25] S. Nowozin, H. Nickisch, M. Deisenroth, and B. Huhle, “Object Classification using Local Image Features},” *Tech. Univ. Berlin*, vol. 5, p. 8, 2009.

[26] Y. Feng and M. Lapata, “How Many Words is a Picture Worth ? Automatic Caption Generation for News Images,” *Proc. 48th Annu. Meet. Assoc. Comput. Linguist.*, no. July, pp. 1239–1249, 2013.

[27] M. Harper, “Introducing Speech and Language Processing,” *Comput. Linguist.*, vol. 32, no. 1, pp. 137–142, 2010.

[28] E. Reiter and R. Dale, “Building Natural Language Generation Systems,” *MIT Press.*, vol. 33, no. 1, 2009.

[29] M. Zock, “Natural Language Processing and Cognitive Science,” *NAACL-2012 Main Proceedings, Montr. Canada. Assoc. Comput. Linguist.*, no. October, 2014.

[30] A. Nenkova and L. Vanderwende, “The impact of frequency on summarization,” *Microsoft Res. Redmond Washingt. Tech Rep MSRTR2005101*, 2010.

[31] J. Clarke and M. Lapata, “Global Inference for Sentence Compression An Integer Linear Programming Approach,” *J. Artif. Int. Res.*, vol. 31, pp. 399–429, 2010.

[32] S. Wubben, A. Van Den Bosch, and E. Krahmer, “Learning to Paraphrase: An Unsupervised Approach Using Multiple-Sequence Alignment,” *Proc. 2009 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol.*, vol. 1, pp. 4292–4299, 2009.

[33] P. Hède, P. Moëllic, J. Bourgeoys, M. Joint, and C E A Fontenay-aux-roses Scri Licm E S I E, “Automatic generation of natural language descriptions for images .,” *Proc. 58 Comput. Inf. Retr. (Recherche dInformation ses Appl. Ordinateur) (RIAO)*, no. 1, pp. 306–313, 2008.

[34] M. Lagoudakis and M. Zervakis, “DOGi : An Automatic Image Annotation Tool for Images of Dog Breeds Adonis Dimas Dissertation thesis,” *Tech. Univ. Crete Dep. Electron. Comput. Eng. Chania*, 2013.

[35] C.-F. Tsai, “Bag-of-Words Representation in Image Annotation: A Review,” *ISRN Artif. Intell.*, pp. 1–19, 2014.

[36] A. J. &and D. F. F. Estrada, “SIFTtutorial pp.12,” vol. 60, no. 2, pp. 91–110, 2008.

[37] M. Balipa and B. R., “Search Engine using Apache Lucene,” *Int. J. Comput. Appl.*, vol. 127, no. 9, pp. 27–30, 2015.

[38] A. Sonawane, “Using Apache Lucene to search text,” *IBM Dev. Work.*, p. 1, 2010.

[39] P. H. Bugatti, A. J. M. Traina, and C. Traina, “Assessing the best integration between distance-function and image-feature to answer similarity queries,” *Dep. Comput. Sci. – ICMC, Univ. São Paulo São Carlos – USP, Brazil*, no. January, p. 1225, 2010.

[40] P. H. Bugatti, A. J. M. Traina, and C. Traina, “Assessing the best integration between distance-function and image-feature to answer similarity queries,” no. January, p. 1225, 2010.

[41] C. Rashtchian, P. Young, M. Hodosh, and J. Hockenmaier, “Collecting Image Annotations Using Amazon’s Mechanical Turk,” *NAACL HLT Work. Creat. Speech Lang. Data with Amaz. Mech. Turk*, no. June, pp. 139–147, 2012.

[42] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (VOC) challenge,” *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010. http://vision.cs.uiuc.edu/pascal-sentences/

[43] S. Li, G. Kulkarni, T. L. Berg, A. C. Berg, and Y. Choi, “Composing simple image descriptions using web-scale N-grams,” in *Fifteenth Conference on Computational Natural Language Learning, Proceedings of the Conference*, 2013, no. June, pp. 220–228.

[44] O. Vinyals and A. Toshev, “Show and Tell: A Neural Image Caption Generator,” *arXiv:1411.4555v2 [cs.CV]* , Apr 2016

[45] http://natlib.govt.nz/librarians/digital-library-tools last accessed June 2017.

Appendices

Appendix A

Stanford typed dependencies the current representation contains approximately 50 grammatical relations. The dependencies are all binary relations: a grammatical relation holds between a governor (also known as a regent or a head) and a dependent. The grammatical relations are defined below, in alphabetical order according to the dependency's abbreviated name (which appears in the parser output). The definitions make use of the Penn Treebank part-of-speech tags and phrasal labels [12]. Some of the Stanford dependencies we use in our in work are listed.

acomp: An adjectival complement of a verb is an adjectival phrase, which functions as the complement (like an object of the verb).

Birds fly low over water. acomp (fly low)

agent: An agent is the complement of a passive verb, which is introduced by the preposition "By" and does the action.

A small child being held by a woman. agent (held ,woman)

amod: An adjectival modifier of a noun phrase is any adjectival phrase, which modifies the meaning of the noun phrase.

An army green plane flying in the sky. amod (plane ,green)

cc: A coordination is a relation between a conjunct and the content word.

A small green and yellow plane in the sky. cc (green ,and)

conj: It is the relation between two (content) words connected by a coordinating conjunction, such as "and", "or", etc. The head of the relation is the first conjunct.

A small green and yellow plane in the sky. conj (green ,yellow)

dobj: The direct object of a verb phrase is the noun phrase, which is the (accusative) object of the verb.

The kitty bus offers the riders a pleasant atmosphere. dobj (offers ,atmosphere)

iobj: The indirect object of a verb phrase is the noun phrase, which is the object of the verb.

The kitty bus offers the riders a pleasant atmosphere iobj (offers ,riders)

nn: A noun compound modifier of a noun phrase is any noun that modifies the head noun

The kitty bus offers the riders a pleasant atmosphere. nn (bus ,kitty)

nsubj: A nominal subject is a noun phrase, which is the syntactic subject of a clause.

The kitty bus offers the riders a pleasant atmosphere. nsubj (offers ,bus)

nsubjpass: A passive nominal subject is a noun phrase which is the syntactic subject of a Passive clause.

A Lemonaide stand is manned by a blonde child with a cookie. nsubjpass (manned ,stand)

amod: A participial modifier of a noun phrase or verb phrase or sentence is a participial verb form that serves to modify the meaning of a noun phrase or sentence.

A small child being held by a woman. amod (child ,held)

prep: A prepositional modifier of a verb, adjective, or noun is any Prepositional phrase that serves to modify the meaning of the verb, adjective, noun, or even another preposition.

A blonde child with a cookie man a Lemonaide stand.

Prep (manned, by) and prep (child, with)

Appendix B

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet href="CoreNLP-to-HTML.xsl" type="text/xsl"?>
<root>
  <document>
    <sentences>
      <sentence id="1">
        <tokens>
          <token id="1">
            <word>A</word>
            <lemma>a</lemma>
            <CharacterOffsetBegin>0</CharacterOffsetBegin>
            <CharacterOffsetEnd>1</CharacterOffsetEnd>
            <POS>DT</POS>
          </token>
          <token id="2">
            <word>black</word>
            <lemma>black</lemma>
            <CharacterOffsetBegin>2</CharacterOffsetBegin>
            <CharacterOffsetEnd>7</CharacterOffsetEnd>
            <POS>JJ</POS>
          </token>
          <token id="3">
            <word>and</word>
            <lemma>and</lemma>
            <CharacterOffsetBegin>8</CharacterOffsetBegin>
            <CharacterOffsetEnd>11</CharacterOffsetEnd>
            <POS>CC</POS>
          </token>
          <token id="4">
            <word>white</word>
            <lemma>white</lemma>
            <CharacterOffsetBegin>12</CharacterOffsetBegin>
            <CharacterOffsetEnd>17</CharacterOffsetEnd>
            <POS>JJ</POS>
          </token>
          <token id="5">
            <word>dag</word>
```

```
<token id="5">
  <word>dag</word>
  <lemma>dag</lemma>
  <CharacterOffsetBegin>18</CharacterOffsetBegin>
  <CharacterOffsetEnd>21</CharacterOffsetEnd>
  <POS>NN</POS>
</token>
<token id="6">
  <word>is</word>
  <lemma>be</lemma>
  <CharacterOffsetBegin>22</CharacterOffsetBegin>
  <CharacterOffsetEnd>24</CharacterOffsetEnd>
  <POS>VBZ</POS>
</token>
<token id="7">
  <word>looking</word>
  <lemma>look</lemma>
  <CharacterOffsetBegin>25</CharacterOffsetBegin>
  <CharacterOffsetEnd>32</CharacterOffsetEnd>
  <POS>VBG</POS>
</token>
<token id="8">
  <word>at</word>
  <lemma>at</lemma>
  <CharacterOffsetBegin>33</CharacterOffsetBegin>
  <CharacterOffsetEnd>35</CharacterOffsetEnd>
  <POS>IN</POS>
</token>
<token id="9">
  <word>the</word>
  <lemma>the</lemma>
  <CharacterOffsetBegin>36</CharacterOffsetBegin>
  <CharacterOffsetEnd>39</CharacterOffsetEnd>
  <POS>DT</POS>
</token>
<token id="10">
  <word>camera</word>
```

```
    </token>
    <token id="9">
      <word>the</word>
      <lemma>the</lemma>
      <CharacterOffsetBegin>36</CharacterOffsetBegin>
      <CharacterOffsetEnd>39</CharacterOffsetEnd>
      <POS>DT</POS>
    </token>
    <token id="10">
      <word>camera</word>
      <lemma>camera</lemma>
      <CharacterOffsetBegin>40</CharacterOffsetBegin>
      <CharacterOffsetEnd>46</CharacterOffsetEnd>
      <POS>NN</POS>
    </token>
    <token id="11">
      <word>.</word>
      <lemma>.</lemma>
      <CharacterOffsetBegin>46</CharacterOffsetBegin>
      <CharacterOffsetEnd>47</CharacterOffsetEnd>
      <POS>.</POS>
    </token>
  </tokens>
  <parse>(ROOT
    (S
    (NP (DT A)
    (ADJP (JJ black)
    (CC and)
    (JJ white))
    (NN dag))
    (VP (VBZ is)
    (VP (VBG looking)
    (PP (IN at)
    (NP (DT the) (NN camera)))))
    (. .)))

    </parse>
```

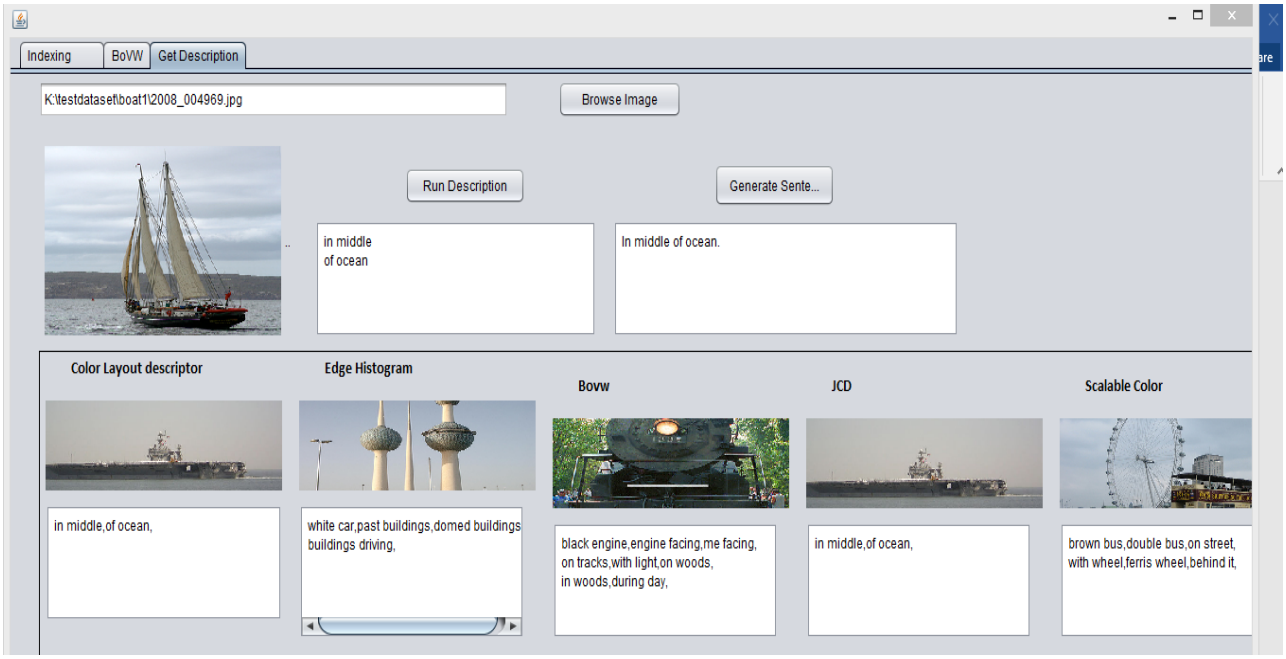
```
  </parse>
<dependencies type="basic-dependencies">
  <dep type="root">
    <governor idx="0">ROOT</governor>
    <dependent idx="7">looking</dependent>
  </dep>
  <dep type="det">
    <governor idx="5">dag</governor>
    <dependent idx="1">A</dependent>
  </dep>
  <dep type="amod">
    <governor idx="5">dag</governor>
    <dependent idx="2">black</dependent>
  </dep>
  <dep type="cc">
    <governor idx="2">black</governor>
    <dependent idx="3">and</dependent>
  </dep>
  <dep type="conj">
    <governor idx="2">black</governor>
    <dependent idx="4">white</dependent>
  </dep>
  <dep type="nsubj">
    <governor idx="7">looking</governor>
    <dependent idx="5">dag</dependent>
  </dep>
  <dep type="aux">
    <governor idx="7">looking</governor>
    <dependent idx="6">is</dependent>
  </dep>
  <dep type="case">
    <governor idx="10">camera</governor>
    <dependent idx="8">at</dependent>
  </dep>
  <dep type="det">
    <governor idx="10">camera</governor>
    <dependent idx="9">the</dependent>
```

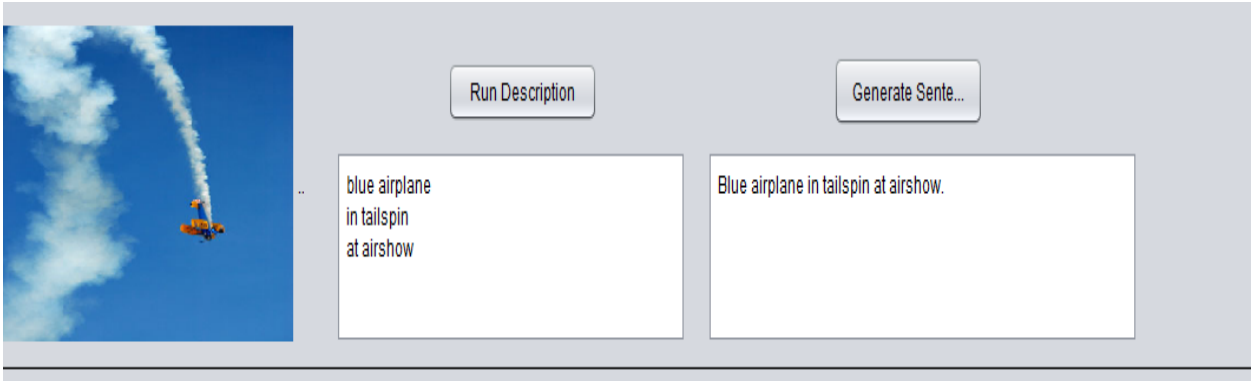
```
      <dependent idx="9">the</dependent>
    </dep>
    <dep type="nmod">
      <governor idx="7">looking</governor>
      <dependent idx="10">camera</dependent>
    </dep>
    <dep type="punct">
      <governor idx="7">looking</governor>
      <dependent idx="11">.</dependent>
    </dep>
  </dependencies>
  <dependencies type="collapsed-dependencies">
    <dep type="root">
      <governor idx="0">ROOT</governor>
      <dependent idx="7">looking</dependent>
    </dep>
    <dep type="det">
      <governor idx="5">dag</governor>
      <dependent idx="1">A</dependent>
    </dep>
    <dep type="amod">
      <governor idx="5">dag</governor>
      <dependent idx="2">black</dependent>
    </dep>
    <dep type="cc">
      <governor idx="2">black</governor>
      <dependent idx="3">and</dependent>
    </dep>
    <dep type="conj:and">
      <governor idx="2">black</governor>
      <dependent idx="4">white</dependent>
    </dep>
    <dep type="nsubj">
      <governor idx="7">looking</governor>
      <dependent idx="5">dag</dependent>
    </dep>
    <dep type="aux">
```

```
    </dep>
    <dep type="aux">
      <governor idx="7">looking</governor>
      <dependent idx="6">is</dependent>
    </dep>
    <dep type="case">
      <governor idx="10">camera</governor>
      <dependent idx="8">at</dependent>
    </dep>
    <dep type="det">
      <governor idx="10">camera</governor>
      <dependent idx="9">the</dependent>
    </dep>
    <dep type="nmod:at">
      <governor idx="7">looking</governor>
      <dependent idx="10">camera</dependent>
    </dep>
    <dep type="punct">
      <governor idx="7">looking</governor>
      <dependent idx="11">.</dependent>
    </dep>
  </dependencies>
  <dependencies type="collapsed-ccprocessed-dependencies">
    <dep type="root">
      <governor idx="0">ROOT</governor>
      <dependent idx="7">looking</dependent>
    </dep>
    <dep type="det">
      <governor idx="5">dag</governor>
      <dependent idx="1">A</dependent>
    </dep>
    <dep type="amod">
      <governor idx="5">dag</governor>
      <dependent idx="2">black</dependent>
    </dep>
    <dep type="cc">
      <governor idx="2">black</governor>
```

```
    <dep type="conj:and">
      <governor idx="2">black</governor>
      <dependent idx="4">white</dependent>
    </dep>
    <dep type="amod" extra="true">
      <governor idx="5">dag</governor>
      <dependent idx="4">white</dependent>
    </dep>
    <dep type="nsubj">
      <governor idx="7">looking</governor>
      <dependent idx="5">dag</dependent>
    </dep>
    <dep type="aux">
      <governor idx="7">looking</governor>
      <dependent idx="6">is</dependent>
    </dep>
    <dep type="case">
      <governor idx="10">camera</governor>
      <dependent idx="8">at</dependent>
    </dep>
    <dep type="det">
      <governor idx="10">camera</governor>
      <dependent idx="9">the</dependent>
    </dep>
    <dep type="nmod:at">
      <governor idx="7">looking</governor>
      <dependent idx="10">camera</dependent>
    </dep>
    <dep type="punct">
      <governor idx="7">looking</governor>
      <dependent idx="11">.</dependent>
    </dep>
  </dependencies>
  <dependencies type="enhanced-dependencies">
    <dep type="root">
      <governor idx="0">ROOT</governor>
      <dependent idx="7">looking</dependent>
```

Appendix C


Indexing
BoVW
Get Description
K:\testdataset\boat1\2008_004969.jpg
Browse Image
Run Description
Generate Sente...
in middle
of ocean
In middle of ocean.
Color Layout descriptor
Edge Histogram
Bovw
JCD
Scalable Color
in middle,of ocean,
white car,past buildings,domed buildings
buildings driving,
black engine,engine facing,me facing,
on tracks,with light,on woods,
in woods,during day,
in middle,of ocean,
brown bus,double bus,on street,
with wheel,ferris wheel,behind it,


Run Description
Generate Sente...
blue airplane
in tailspin
at airshow
Blue airplane in tailspin at airshow.

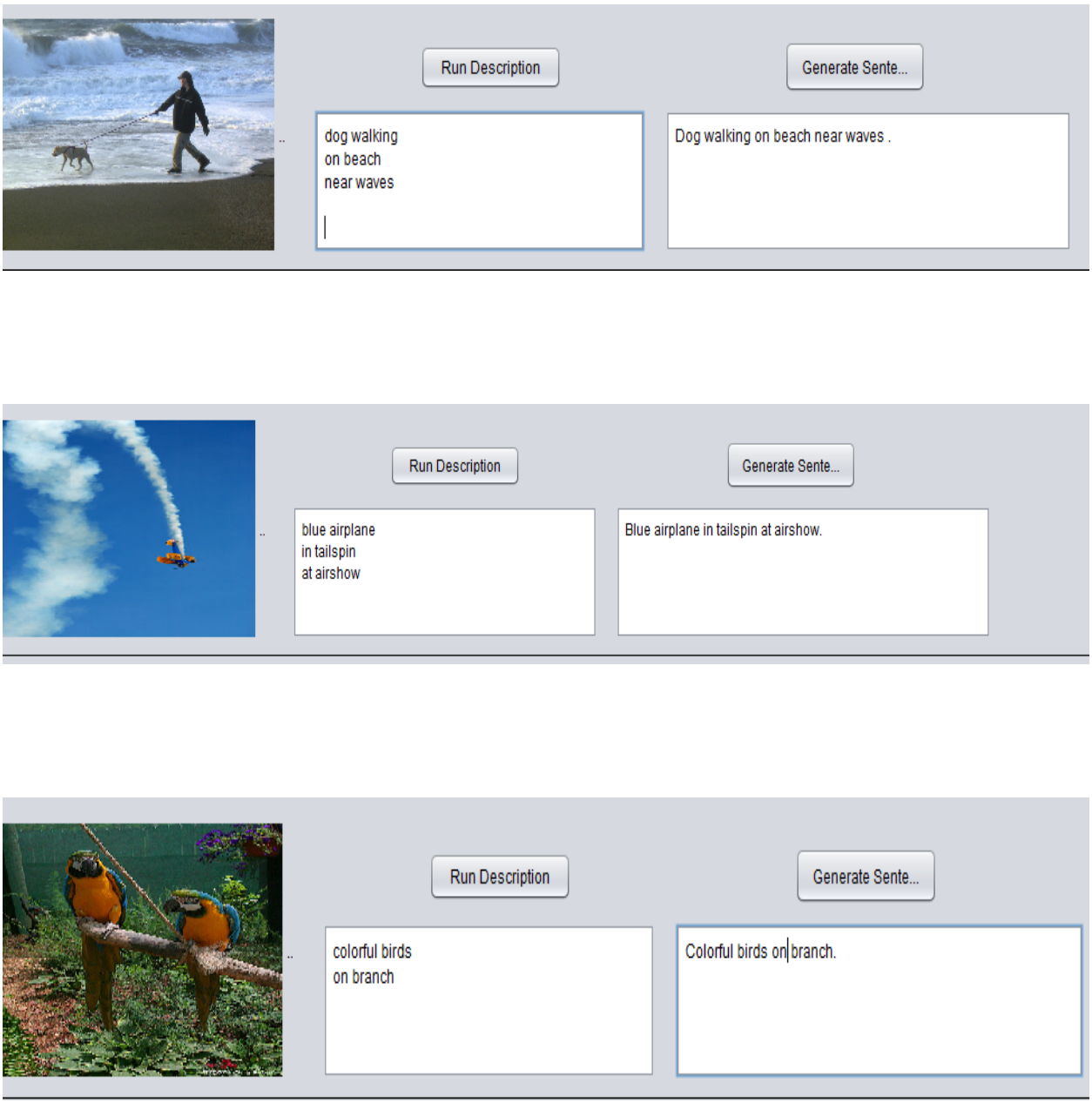


Figure 5-8 Examples of image description generated in our approach images from the PASCAL dataset

Table 5-7 Phrases Relevance Computing for association Phrases with Image

Images	Attribute-subjects Counts Attribute-objects Counts Subj-verb Counts	Integration
	Yellow aero plane Blue Sky Aeroplane-fly Fly-in-sky Sit-in -sky	yellow aero plane blue sky aeroplane fly in sky
	overweight person person-sit sit-on-sofa	overweight person, sit-on-sofa ,green sofa
	brown horse grass field stand-in-field horse-stand	brown horse, stand-in grass field
	white cat cat-sit cat-look	white cat look at camera
	fishing boat fishing anchor boat-docked –anchor lie-at-anchor	fishing boat lie at anchor