Visual Objects in the Global Graph

by

Joseph Dung

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Arts

Humanities Computing,

University of Alberta

Abstract

Here, we use the premise of an actual image-retrieval application to examine how various approaches and techniques in computer vision help to bridge the much talked about semantic gap. A lot of cross-fertilization of ideas from the world of text processing has found its way into image processing as well. When images are processed for various purposes, their low-level features usually bear no resemblance with the types of concepts used in describing them. When tasked with developing a useful image model, most strategies take a data-driven or ontological route, or a combination of both. Whatever strategy is adopted, we observe how different image contexts are used to derive some type of semantic knowledge. In this study, we provide an analysis of how an ontological model can be derived from the structural composition of clustered features that result from an image-retrieval task, especially focusing on error pairs. In other words, we explore the additional contexts in which the semantic gap can be narrowed, when the search context for images relative to a large database of features, is also narrowed. We use a small sample of games set to train and eventually test how effective our image-retrieval task can find and match an image based on its low-level features. In so doing, we had wanted to create the basis for potentially pairing these unique low-level features to a higher-level concept based on scene class, for instance. But ultimately for each image-retrieval task, we keenly recognize when errors do occur, under different object descriptor and search strategies, and particularly look out for consistent error patterns across these descriptors, based on the retrieved results from an image search. We discover an additional context for deriving semantic knowledge about the query image, providing for the basis to develop another data-driven ontological model.

Acknowledgements

This learning process has been an especially exciting but difficult process, but the gains of all the insights are so well worth it. I am so grateful for the warm support from my family, who well know that I am the type of person who takes the red pill and stumbles down the rabbit hole. I am especially thankful for my mother's prayers and Suzie's kind concern and encouragement. Their prayers is truly appreciated and treasured.

A lot of others have helped challenged and nudged me to try harder and do better in the course of this research. I want to thank Professor Geoffrey Rockwell and Janey Kennedy for this persistent nudge. Thank you so much! In like manner, I am most grateful for Tanya Marin's constant encouragement and support, and so many like her that I have met who have shown interest in my work with the desire that I do better. I also appreciate the support I got from the Graduate Student's Association and the International Center of the University of Alberta.

I am thankful to God most of all, who does sustain me and loves me…, most times I do not know the reason why. But I am so very grateful because he seems not to have given up on me, knowing that with his help I can always learn from mistakes and do better. Indeed, learning *never* stops!

Contents

List of Figures

List of Tables

Chapter 1 Introduction

Perhaps, an almost alarmist tone from a *New York Times* article by Markoff (2012), may provide some perspective in ascertaining some of the goals of this study:

> But even as images and video rapidly come to dominate the Web, search engines can ordinarily find a given image only if the text entered by a searcher matches the text with which it was labeled. And the labels can be unreliable, unhelpful ("fuzzy" instead of "rabbit") or simply nonexistent. To eliminate those limits, scientists will need to create a new generation of visual search technologies — or else, as the Stanford computer scientist Fei-Fei Li recently put it, the Web will be in danger of 'going dark'.

We explore the same problem here as well, by seeking to understand how those limits as stated in the quote above could be eliminated in image-retrieval contexts. Another term for the elimination of those limits is essentially the narrowing of the *semantic gap*, which will be a persistent premise underlying all our discussion.

1.1 Motivation

Computer vision is a budding field of study that has emerged from the need to enable computers and other types of machines to have the ability to perceive the world as humans would. This process depends on so many types of image processing techniques that are always constantly being improved upon. Regardless of the approach taken, most methods will include the various procedures for acquiring, processing, analyzing, and understanding images, in general, high-dimensional data from the real world in order to produce numerical or symbolic information, so as to make informed decisions, (Klette, 2014). This ability to emulate human perception forms an important step in designing world-facing systems that can perform certain intelligent tasks. We are inundated with rich visual information in the real world and interpreting this vast amount of data can be a challenging process. Vision-based systems rely on the extraction of information from the images captured to carry out certain tasks. Normally, the eventual goal is to use this information to gain an understanding of different objects present in an environment along with their physical and geometrical attributes. The type of information extracted and its analysis depends upon the application to be performed. This is evident in the many visual object

recognition tasks devoted to understanding the basic conceptual categories a recognized object may belong to, including their scene classes. Our study aims to take this a step further, restricting our focus to a basic image-retrieval task where each image is assumed to represent a distinct scene class. We forward an image-retrieval tool using OpenCV[1] libraries within the Visual Studio 2010 framework. However, we had earlier tested *hundreds* of scripts and applications using both the OpenCV and Matlab vision tools with the hopes of properly evaluating our games' real-world image dataset. But we encountered a huge problem during this study: most computer vision applications that have resulted from academic research or otherwise are strictly tied to the types of dataset used.

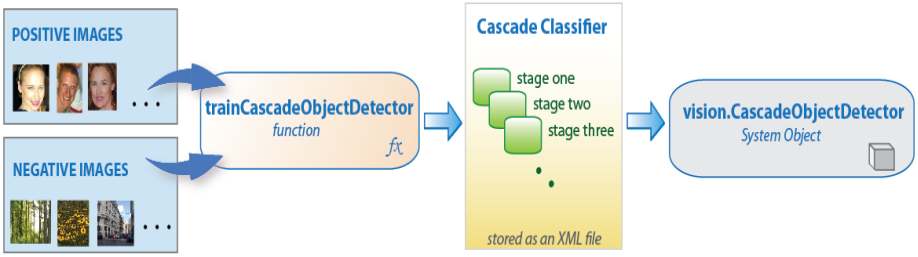

Figure 1.1 OpenCV Haar Cascade pipe

In other words, an object or facial recognition algorithm is designed around the dataset used to train and test for it, meaning that it is not always possible to basically plug in a games' image dataset without seriously tweaking the source code, a process which costs more time. For example, using the OpenCV Haar classifiers will not easily work for our chosen games dataset. This is because most of the trained datasets in use were designed to recognize real-world faces and objects and not drawn or modelled characters. The initial direction of this thesis was to find a way to build a Haar classifier from our image dataset using the exact same process OpenCV uses to make their own Haar Classifiers which are basically XML files containing object descriptor vectors. While OpenCV uses the Haar classifier as an approach, we are not limited to using its technique since we can easily use its libraries in our own application. We had use an image-retrieval tool that is not limited to using XML files to store image vectors. We instead use an Inverted File for storing the features we had extracted; and we also use an Inverted Search process to retrieve them. When a query image is used to discover a match, the descriptive features extracted from the query image is compared against those in our database, usually for

[1] OpenCV (Open Computer Vision) is an open source library of computer vision tools.

the purpose of object recognition. Inverted matching was described by Sivic and Zisserman (2003), where they state that "an inverted file is structured like an ideal book index. It has an entry for each word in the corpus followed by a list of all the documents (and position in that document) in which the word occurs." However, XML file vectors are not usually the ideal structures for making inverted file storages, thus limiting the broad utility of OpenCV Haar Cascades. Object Recognition underlies the core problem of learning visual categories as well as identifying new instances of those categories. Most vision recognition tasks fundamentally rely on the ability to recognize faces, objects, scenes, in *specific* cases besides the resolution of general categories which in turn depends on a lot of other factors (which we will explore in the study), including the annotation and eventual training of a dataset.

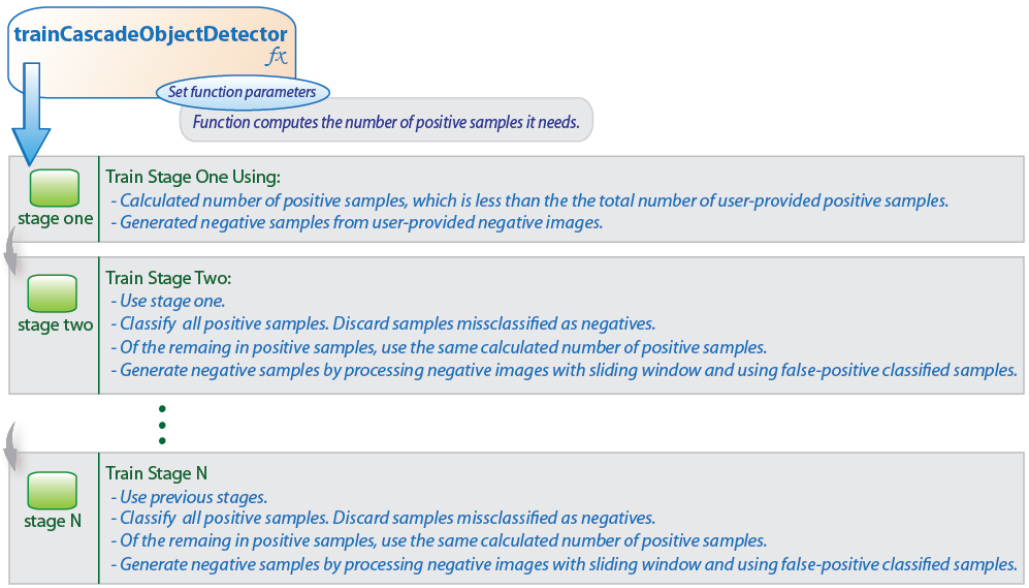

Figure 1.2 Training stages for Haar Cascade

In fact we did build a simple Object Marker annotation tool to be able to mark out objects and particular regions of interest from our images that do contain the object or faces we want to train and recognize.[2] This is from recognizing that the concept of a face or monster from a games' character point of view is so very different from what normal facial recognizers have been trained to expect. The following figures are actual screen shots of our annotating script, using a bounding box to mark out the object of interest for training. But we soon discovered in the course of this research that the OpenCV Haar Classifiers do work best as bounding box object

[2] This was the initial thrust of this thesis project before being forced to change direction due to time.

detectors, and the accuracy of our own classifier will depend on the quantity of the training images used and the ratio of the negative to positive image samples we intend to use. Depending on the object descriptor and clustering strategy used, it takes *weeks* to actually train image samples in order to get any level of accuracy at all, during testing. The general rule of thumb has been that the larger the training set used, the more accurate the recognizer will get in identifying new unseen instances of the same objects, faces, events or scenes. Nonetheless, as stated before, much of the current research effort has been based on the recognition of objects and scenes in the *real* world, and this is reflected in the types of datasets already annotated and trained, with annotated games content, completely absent.

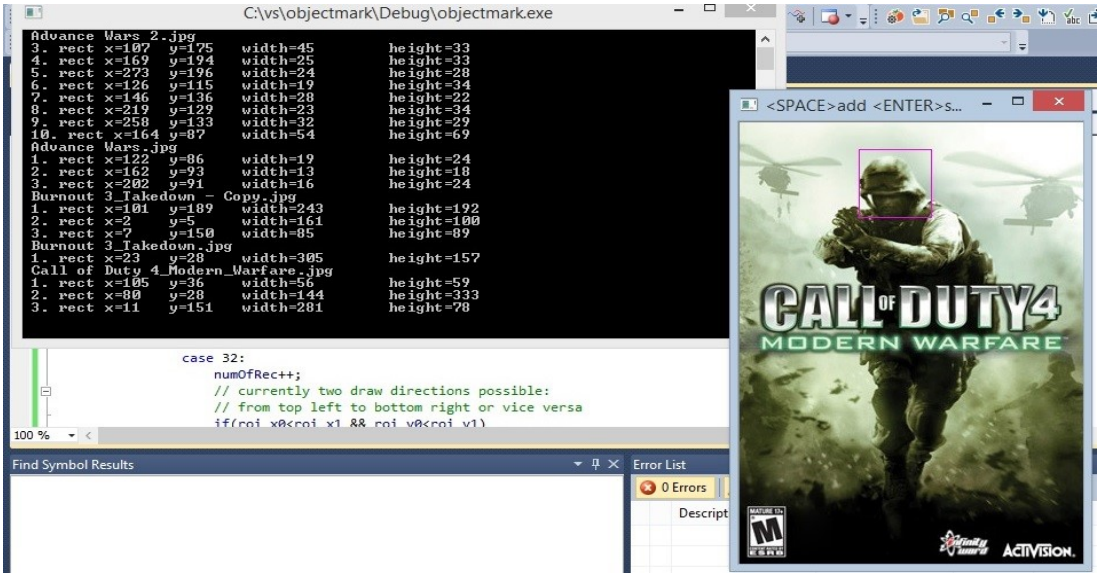

Figure 1.3 Object marker tool for annotating images

In the figure above, notice that there are no distinctive facial features on the character to indicate a normal face exists. We had to manually mark out the head/face area of the character for the purpose of creating a unique Haar-based classifier that recognizes such visual instances above as a face. This will also hint to the great difficulty any current facial recognition system will likely encounter when trying to discover a face in the game's image. The same problem can be extended to finding a monster in an image. Currently, there are no recognition systems dedicated to finding monsters, even though monsters may share some vague facial traits with humans or animals. Ordinary facial recognition is still a very hard and unsolved problem. Developing a future *monster* recognition system is not a trivial task. We predict that this is going to be an even

harder problem than ordinary facial recognition. Which unique features constitute a monster is not a simple question after all. Consequently, an exercise that saw this study initially annotating game's content with faces and monsters including other objects for eventual training had to inevitably stop because we were running out of time. The decision was taken to focus on an image-retrieval task where each image is likened to a unique scene class and to search for visually similar images with our tool can be likened to searching for a collection of similar scene classes—or not, since we decided that we will be very particular about the significance of retrieval mismatches or errors and how the same errors can contain significant semantic information about the query image itself.

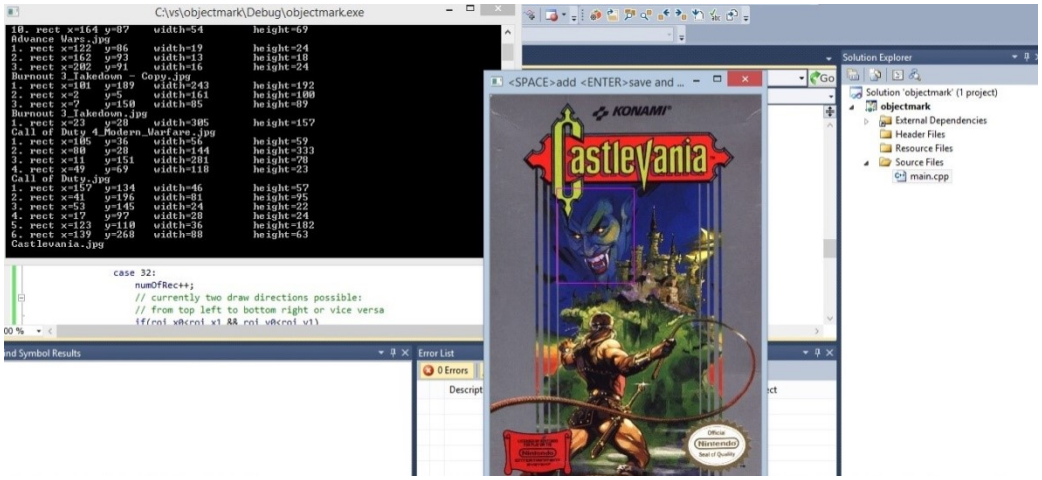

Figure 1.4 Object Marker tool designating a monster's face

1.2 Problem definition and Contribution

We decided to focus on the basic task of image-retrieval in this study with these objectives in mind: i. We treat each image as a unique scene class since we did not have the time to annotate individual faces and objects in the image for eventual recognition ii. By viewing each game's image as a unique scene class, we had expected the possible mapping of the image's low-level features to a higher abstract description, or the bridging of the semantic gap, iii. But this mapping will also depend on how efficiently the chosen object descriptors we used for this study were able to repeatedly locate and match the images based on their features; thus, we will measure their performance iv. Unfortunately, in most image-retrieval tasks there will always be errors and mismatches based on a host of factors. This study investigates the possible additional semantic

input gotten from these same errors, or as we describe them, *error pairs*. This thesis' unique contribution emerges from viewing these errors derived from several image-retrieval tasks, as possible signals for the probe image, in so doing, generating new sources of learnable semantic knowledge about the probe images. We also propose a new form of image search.

1.3 The Semantic Gap

Evidently most of the central object recognition problems in vision have come a long way in the classification of objects into different conceptual categories. The introduction of better techniques and data has seen confidence scores risen for many such recognition tasks. Basically it is no longer enough to simply visually recognize an object into a category, even though so much of this task is yet to be solved, since the occurrences of many unique objects seem limitless. Even though the classification layers of this field of study have barely been exhausted with so many types of tasks and categories still considered, (as seen by the many ImageNet, PASCAL and Voc tasks carried yearly), the overall trend has shifted towards the fine-granular examination of objects (as in a *flower*, what *type* of flower--based on the analysis of petals and leaves, as seen in many ImageNet tasks), or even the process of visually comparing the features of one these objects to those of an entirely unrelated one based on their sub-similarities. With this observation, one cannot deny the underlying importance of a semantic layer of sorts, to assist in the in-depth disambiguation of objects. However, the bulk of the techniques in computer vision have so far relied on low-level image analysis techniques which have no bearing to the overall cognitive depiction of the same object. Hence, a semantic gap is created as a wide gulf between these low-level features the object has and its high-level cognitive description. A semantic layer that links local surface features with other visual cues from a related vocabulary to broaden the understanding of a given object being processed and recognized is always welcomed. There is a distance between the descriptions obtained by automatic methods for image analysis and their real content, as explained in the semantic gap.

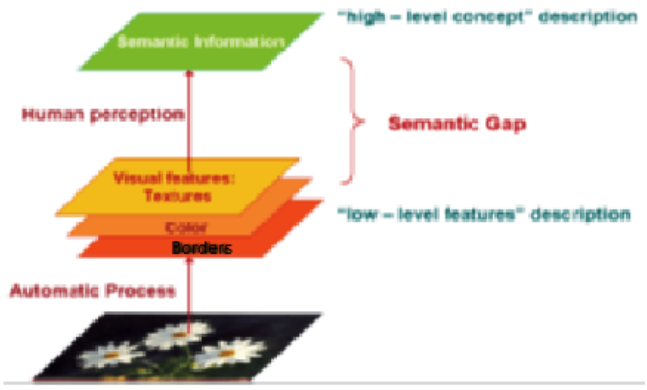

Figure1.5 The Semantic Gap between the image and the concept for it

This gap is further explained by the absence of a meaningful concurrence between the information extracted automatically by computers and the human perception of the real image resource, based on high-level concepts (Hare et al. 2006). Low-level features of auditory streams as well as video sequences and static images have received a lot of attention because features at this level can support automated indexing of content as opposed to high-level descriptions which currently map poorly to scenes, and also require an arduous manual annotation process, prone to errors. Automated indexing of content can be exemplified by attempts to generate descriptions of broadcasted sports events (e.g. identification of goals in a soccer match) based on (combinations of) low-level features. Low-level feature extraction using attributes such as color, texture parameters, borders, etc., is typical for most current feature extraction, but there are also higher abstractions of these features that aim to correspond to higher concepts which we will discuss in the next chapter.

1.4 Local Features

Local feature descriptors are designed to find local image structures in a repeatable manner and to represent them in robust ways that are invariant to typical image transformations, such as translation, rotation, scaling, and affine deformation. Indeed, the local features constitute the basis of approaches developed to automatically recognize specific objects or scene classes, as pointed out by Grauman and Leibe, (2011). The most popular local feature extraction method is the Scale Invariant Feature Transform (SIFT), introduced by Lowe (2004) which we will further discuss in the next chapter, and also adopt as our object descriptor baseline for our thesis experiment.

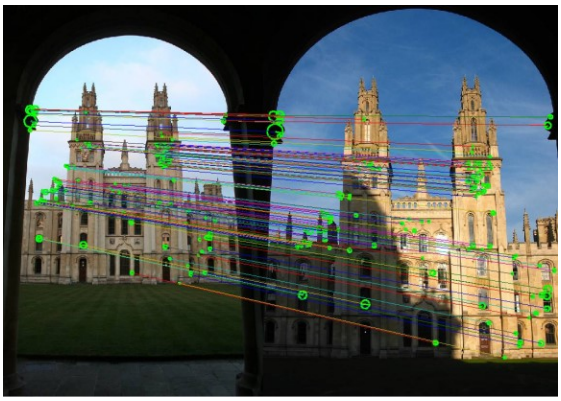

Figure 1.6 Locating corresponding local features

The ability to identify whole scene classes, particular locations or even particular objects like buildings usually depends on how accurately our object descriptors were able to detect the unique local features in an image, and eventually mapping them to a similar image. This is exactly the basis for the recognition of object instances. The goal of instance-level recognition is to match (recognize) a specific object or scene, for any image-retrieval task. Possible examples include recognizing a specific building, such as the Notre Dame, or a specific painting, such as *Starry Night* by Van Gogh. The challenging objective is to also recognize these images despite changes in scale, viewpoint, illumination conditions and partial occlusion. The application of feature descriptors—like SIFT, and its equivalents— has had an impact on image-retrieval research which relies on corresponding descriptors when searching from an image of an object of interest (the query or probe), to obtain (or retrieve) those images that share similar traits with the target object.

1.5 Overview of chapters

In Chapter 2, the theoretical background of this thesis's work is explained using concrete examples from related research. Chapter 3 describes our experiment with an image-retrieval application, where we also focus on the error pairs generated. There we discuss the types of signals these have for a probe image. Chapter 4 provides a brief conclusion. The Appendix aims to visually summarize the key ideas discussed in this thesis, with the hopes of providing a better understanding of the various strategies in use to algorithmically resolving the meaning of images.

Chapter 2 Theoretical Background/Related Research

2.1 Appreciating the Problem

A student whose core research was in Computer Vision describes his parents' initial simplistic understanding of the field when he tried to explain his research.[3] He stated that it was mostly about making computers "see" and recognize objects in an image. Naturally, just as many others, his parents were confused about why that was *difficult*. Given that, when scrutinizing from afar, it only takes looking at a picture and, *say*, "*see* a chair in it. It is so effortless!" (for humans at least). What is the problem? The problem, as the student goes on to explain, is that the programs that exist and that are written do not come with a magical Visual Cortex. To best illustrate the problem, the student proceeds to plainly reveal what the computer has to work with, in order to accomplish any recognition task: a huge array of every pixel's Red, Green and Blue component that together give the color of that pixel. The following is a *much*-truncated listing of one such array that stores an image:

> **…** (17, 16, 11), (124, 120, 85), (120, 112, 76), (122, 114, 78), (122, 124, 87), (118, 114, 79), (126, 118, 81), (122, 114, 78), (123, 115, 79), (103, 110, 69), (123, 119, 84), (124, 116, 77), (122, 114, 75), (126, 118, 82), (115, 112, 79), (121, 117, 82), (124, 116, 80), (125, 117, 81), (50, 55, 23), (121, 116, 78), (119, 111, 74), (121, 113, 74), (126, 117, 84), (20, 17, 12), (121, 117, 82), (119, 111, 75), (120, 112, 73), (121, 115, 79), (115, 111, 76), (117, 109, 70), (118, 110, 74), (123, 115, 78), (105, 111, 85), (122, 116, 82), (115, 110, 70), (117, 107, 71), (120, 114, 80), (112, 109, 74), (121, 112, 81), (102, 101, 57), (117, 110, 66), (60, 63, 36), (112, 106, 72), (106, 101, 63), (106, 98, 62), (115, 111, 66), (18, 18, 18), (109, 107, 66), (34, 34, 10), (110, 101, 62), (115, 110, 72), (116, 108, 72), (102, 94, 55), (94, 91, 58), (110, 103, 61), (24, 27, 10), (108, 100, 63), (74, 73, 42), (100, 91, 58), (114, 109, 67), (111, 109, 70), (95, 86, 53), (27, 24, 17), (94, 87, 43), (87, 88, 57), (99, 94, 62), (76, 73, 42), (77, 70, 41), (78, 73, 44), (20, 19, 17), (81, 79, 54), (18, 18, 18), (75, 70, 38), (107, 103, 66), (102, 93, 60), (72, 66, 40), (59, 57, 34), (14, 10, 1), (88, 92, 65), (72, 64, 51), (52, 53, 35), (42, 37, 17), (104, 95, 66), (101, 99, 60), (53, 46, 30), (21, 22, 17), (62, 61, 43), (61, 62, 28), (75, 70, 38), (33, 31, 16), (29, 27, 15), (50, 44, 30), (15, 16, 11), (6, 5, 1), (25, 27, 22), (47, 44, 35), (91, 85, 49), …

The student points at the massive array of numbers and says, "There. Can you tell me if there is a chair in the above image? It just so happens that there is." After few moments of shocked gaze, and a minute spent clarifying the problem, his parents proclaimed the task impossible. Observing

[3] Based on actual events read in a forgotten Vision blog a long time ago.

the cluster of numbers in triples, indeed, the task looks much more daunting but this is exactly what computers and the human researchers who analyze these images have to deal with. However it is worth noting too that so much progress has been done, after poring over the lower features of an image. There is nothing inherent in the lower-level features to suggest that a human or an object may even exist since the machine only sees numbers. But a great deal of effort has been made towards abstracting away an image and representing them beyond just pixels, so as to better study and train them. An image representation beyond the pixel is welcomed. Describing the semantic gap, (Agarwal and Roth, 2002) state it this way: "…we suggest that in order to extract high-level, conceptual information such as the presence of an object in an image, it is essential to transform the raw, low-level input (in this case, the pixel gray scale values) to a higher-level, more 'meaningful' representation that can support the detection process." (1) As can be noticed in the figure below, image representations that get mapped into a high-level function are possible, and also indicative of what happens to images when they are processed with a learning model.

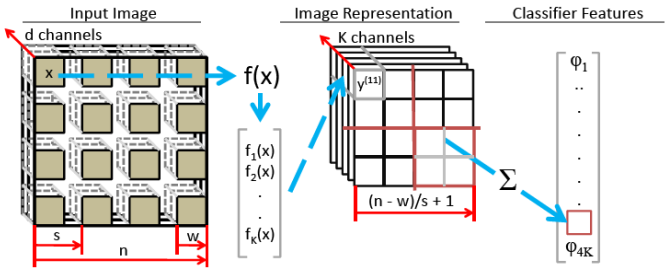

Figure 2.1 The Image as a Matrix Mapping function

Obviously, unlike the language world, images do not readily have the equivalent of a part of speech, to enable the prediction of their distributional patterns. Parker (2011) also reflected on this core problem: "… at its heart, computer vision is about making measurements on images and/or determining what objects appear within those images. Many people have difficulty understanding why this is a hard problem. After all, people recognize complex objects with apparent ease, and quickly. Why is this hard for computers? The answer is that computers use pixels to represent objects rather than some more natural representation that has more structure." Indeed, the pattern in which these pixels appear are as arbitrary and varied as it can be. The fundamental elements of any language have a lot more predictable structure when compared to the elements of an image. Rizoiu et al. (2014) further clarifies this when scrutinizing the degrading data quality problem:

> The difficulty when analyzing images comes from the fact that digital image numerical formats poorly embed the needed semantic information. For example, images acquired using a digital photo camera is most often stored in raster format, based on pixels. A pixel is an atomic image element, which has several characteristics, the most important being the size (as small as possible) and its color. Other information can be color coding, alpha channel etc. Therefore, an image is stored numerically as a matrix of pixels. The difficulty raises from the fact that low-level features, such as position and color of individual pixels, do not capture too much information about the semantic content of the image (e.g., shapes, objects). This problem is also known as the semantic gap between the numerical representation of the image and its intended semantics. To address this issue, multiple representation paradigms have been proposed… (2)

One of the proposed paradigms that builds on layers and layers of previous work is the Bag of Words model which we will be discussing much later in the chapter.

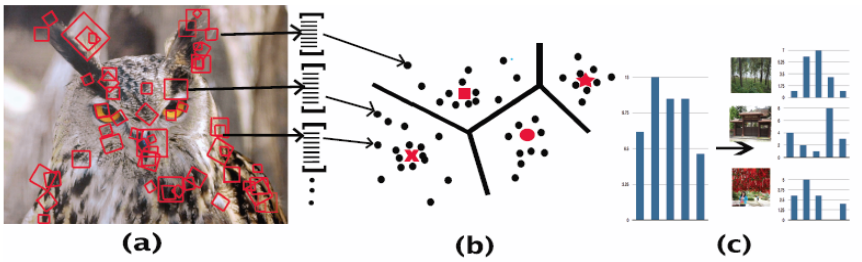

Figure 2.2 Illustration of bag of visual words procedure: (a) detect and represent local interest points as descriptor vectors (b) quantize vectors (c) histogram computation to form Bag of Visual Words vector for the image.

In the above model, an image not only consists of a large matrix of pixels made up of numerical values along its rows and columns, but also allows for normal arithmetic operations to be applied to it. Hence we find particular regions in the image detected for some importance, enabling computations for a histogram count of features. Each image region has a frequency count for a particular feature. Most image-processing algorithms execute matrix operations at this basic level, when applying operators and learning models on the raw matrix. We could liken the properties of these pixels to types of semantic attributes that could be further processed by an algorithm. A pixel may also have a certain type of colour, position, luminance and texture, also

captured by numerical values inside its matrix representation (pixel colour, position, luminance, location and texture are all types of semantic attributes as well). We equally discover that different regions of the matrix may contain numeric values that are similarly based on colour or luminance. Eventually, we understand that the regions with particular contrasting numeric values can as well define the shades of an "edge" or a line. If we follow the contours of these lines we eventually can determine a unique segment or shape which denotes a semantic value as well, but at a higher level of abstraction. While humans can easily make out what these shapes mean, it is still not an obvious process for a machine. It is the task of image *detectors* to determine the most interesting regions of an image and the task of image *descriptors* to transform those regions into a consistent feature vector. (Nixon, 2008) describes the process:

> Objects are represented as a collection of pixels in an image. Thus, for purposes of recognition we need to describe the properties of groups of pixels. The description is often just a set of numbers – the object's descriptors. From these, we can compare and recognise objects by simply matching the descriptors of objects in an image against the descriptors of known objects. However, in order to be useful for recognition, descriptors should have four important properties. First, they should define a complete set. That is, two objects must have the same descriptors if and only if they have the same shape. Secondly, they should be congruent. As such, we should be able to recognise similar objects when they have similar descriptors. Thirdly, it is convenient that they have invariant properties. (281)

We will further discuss feature descriptors in detail in the next chapter where we get to compare how accurate several common descriptors accurately process and match images from a number of games cover content using the OpenCV based tool we developed for this project. Feature descriptors have emerged to be a very essential step in the processing of images for varied tasks and with it, also carry semantic importance because of their abstractions. In this study, we cluster various image descriptors using a bag of words model and carry out an inverted search to retrieve corresponding images, especially looking out in particular for errors when mis-matches occur. We also developed a tool to measure the relative distance candidate test images have in comparison to the other images in the dataset. The significance of this being that we can proceed to set a thresh-hold where closer images could belong to the same visual family, based on the similarity of their feature descriptors. However, in the remainder of this chapter, we will explore

some background detail around the extensive amount of work done so far to narrow the semantic gap, noting sadly, that a chunk that will be discussed here in this chapter has not been incorporated in the experiments done for this thesis because of time constraints.

2.2 Analyzing the Semantic Gap

If the numbers behind pixels did represent the atomic elements behind an image, any form of abstraction that attempts to configure an image around color, shape, texture, or location, embodies noteworthy but simple attempts to bridge the semantic gap. Indeed, color, as well as shape, texture and location, in an image context are *semantic* descriptions of images themselves—nonetheless, still at a lower level of abstraction. Those depictions, while extremely useful, do not go far enough in helping the machine *really* understand the contents of an image. Improved forms of image representation that utilize both the low-level features and a high-level structure approximating the content in a reasonable time frame, we not only required, but have also captured a lot of ongoing research. While it is true that most techniques in computer vision use hard language models that encapsulate statistical tools for visual object recognition, some attention is also paid to the critical role semantic models can also bring.

An extremely high-level problem to illustrate this is the concept of *scene* and *action* recognition (relevant because each game's cover content could also depict a type of scene or action as well). What type of scene or action does an image or frame depict? In this regard, we realize how useful the location of individual objects in relation to others is, as well as their positions or place in a wider spatial semantic network. This problem has consequence for the analysis of game image content since most covers attempt to depict a very focal type of event that also describes the title. Understanding colors, shapes and texture may help in building up the layers of semantics that eventually help determine what type of action or scene a games' image cover portrays, but then, as explained earlier, this is not a straightforward interpretation for the computer. If the machine really knew what all those basic elements sum up to, it could possibly identify scenes where "characters sky-diving and shooting" or where "characters evading arrows and rocks" are—and easily index them for a content-based search engine. This is far from a solved problem and it represents another core challenge of bridging the semantic gap. (Rizoiu et al. 2014) describes their strategy in solving an aspect of the problem:

> One of the privileged tracks to closing the semantic gap is to take into account additional information stored in other types of data (e.g., text, labels, ontologies of concepts) associated with the images. This raises the difficulty of object co-occurrence. For example, a picnic scene is defined by the simultaneous presence of "people", "trees", "grass" and "food". In terms of labels, this translates into label *co-occurrence*. Our approach can be scaled to image classification by addressing the label co-occurrence issue. (25)

Influences from textual distributional semantics seem to have also influenced image inferences in the instance of label co-occurrence. Bruni et al. (2011) explains *distributional semantic models* as: "the use of large text corpora to derive estimates of semantic similarities between words. The basis of these procedures lie in the hypothesis that semantically similar words tend to appear in similar contexts", something also noted by Miller and Charles (1991) and Wittgenstein (1953). In clarifying their distributional semantics from text and images, Bruni et al. also asserts that, "the meaning of spinach (primarily) becomes the result of statistical computations based on the association between spinach and words like plant, green, iron, Popeye, muscles. Alongside their applications in NLP areas such as information retrieval or word sense disambiguation (Turney and Pantel, 2010)." (1) Analogously, when objects in an image appear in a clutter, it is possible to make informed predictions about their co-occurrence with other objects and likely predict scene classes based on their distributional patterns in the real world (as long as it is *not* the chaotic image of a *garbage* scene). When an image contains books, a bed, a lampstand, a pillow, and a drawer it could possibly infer what other types of real world objects which may appear in the image as well—since these are typically objects found in the bedroom. The bedroom becomes the scene class based on the objects that co-occur in the image. In other words, the existence of real world objects in their natural or man-made order project a meaningful distributional pattern with reasonable semantic properties. Hence it is likely odd, finding a tree or a boat inside a bedroom; therefore, processed images that utilize semantic models can infer probable objects for a scene class in addition to scoring very low, those that are incongruent.

An additional higher level pattern with interesting semantic properties hinted by Acharya and Ajoy (2005) is the process of image segmentation. They illuminate on the concept:

> Segmentation is the process that subdivides an image into a number of uniformly homogeneous regions. Each homogeneous region is a constituent part or object in the

> entire scene. In other words, segmentation of an image is defined by a set of regions that are connected and non-overlapping, so that each pixel in a segment in the image acquires a unique region label that indicates the region it belongs to. Segmentation is one of the most important elements in automated image analysis, mainly because at this step the objects or other entities of interest are extracted from an image for subsequent processing, such as description and recognition…. After extracting each segment, the next task is to extract a set of meaningful features such as texture, color, and shape. (2)

Notice how the assigning of semantic labels carefully follows the segmentation process. There is clear topological structure that guides any segmentation process given the fact that they are also a key component in scene analysis. (Shotton et al. 2008),[4] identified how little has been done in *event recognition* so far as static images are concerned, even proceeding to define an event to be a semantically meaningful human activity taking place within a selected environment and containing a number of necessary objects. Shotton et al. proceeded to set the goal of "achieving an event categorization by as much semantic level image interpretation as possible". They describe how this is somewhat like what a school child does when learning to write a descriptive sentence of an event. It is taught that one should pay attention to the **five W's**: *who*, *where*, *what*, *when* and *how*. In Shotton et al.'s system, they attempted to answer three out of the five W's: *what* (the event label), *where* (the scene environment label) and *who* (a list of the object categories). They defined their goal as classifying an event in the image as well as providing a number of semantic labels to the objects and scene environment within the image. For example, given a rowing scene, their algorithm recognizes the event as "rowing" by classifying the environment as a lake and recognizing the critical objects in the image as athletes, rowing boat, water, etc. They were able to achieve this integrative and holistic recognition through a generative graphical model. Observed from an engineering viewpoint, event classification is a useful task for a number of applications. It is part of the ongoing effort in providing effective tools to retrieve and search semantically meaningful visual data. Such algorithms are at the core of the large-scale search engines in addition to some digital library organizational tools. Event

[4] Actually, Shotton et. al, used a Semantic Texton Forests for image segmentation, reviving interests in the potential for textons as a result. Textons use a completely different approach from normal descriptors.

classification is also particularly useful for automatic annotation of images, as well as descriptive interpretation of the visual world for visually impaired patients.

MIT Professor Patrick Winston in a class on Visual Object Recognition proceeded to write down these verbs on the board and challenge his students.[5] "How do you visually determine what's happening? If you could write a program that would reliably determine when these verbs are happening in the field of view, I will *sign* your…!" He utters after writing the following verbs on the board:

> Approach Carry Dig Fall Give Hit Lilt Push
> Run Touch Arrive Catch Deep Flee Go Hold Move Put
> Down Snatch Turn Attach Chase Enter Fly Hand Kick Open
> Raise Stop Walk Bounce Close Exchange Follow Haul Jump
> Pass Receive Take Bury Collide Exit Get Have Leave Pick up
> Replace Throw

Winston was attempting to demonstrate something that goes beyond basic conceptual categories. Each of these verbs would likely show a human being *doing* something in relation to another human or another object. In this regard, our object recognizer is tasked with not only "identifying" the primary target object or human, but also resolving how this same human is connected with other secondary objects in the field of view. In the case of *drinking*, the object recognizer must not only detect the human but also detect the object in the human's hand or in close proximity with the human and then make clever inferences as to what may be going on based on this connection. However, of all the techniques Winston discussed when analysing the problem, he may have overlooked a key contribution a graph-based approach might mean for understanding the main *verb* in the image by the simple detection of co-occurring objects in the image context. We assume that when certain objects co-occur near each other, an obvious but latent action (or verb) is inferred (as in when images contain *human*, *tomatoes* and *knife*, the verb "cutting" is elicited; or when *human*, *paper* and *pen* co-occur, the elicited verb is "writing"). But there are some other difficult types of verbs for the machine to understand. How does the system understand the concept of *Pick up* or *Replace*; *Put* or *Give*; *Fall* or *Lift*? –just to mention a few.

[5] Patrick Winston's lecture can be found on MIT OpenCourseWare's Youtube Channel: http://youtu.be/gvmfbePC2pc

Recognizing these actions is still a very high-level of abstraction and is a crucial part of scene or event recognition of images. We mention this because any serious future studies of images using both games and movie content will not only have to go beyond the recognition of individual objects in an image, but also go towards identifying what type of action(s) the image is depicting and possibly the emotional valence of the identified scene class. This type of research is also very useful for the analysis of image-retrieval using games image content. On the other hand, *reality* is a domain much of the current research in image analysis has resided and so applying any existing technique or reality-based data to fantasy worlds –as found in some movie genres or video game content— may prove problematic. This is because the image content of these other worlds attempt to provide additional meanings to what may pass as familiar objects or scenarios, or even provide unfamiliar objects and scenarios entirely.

2.3 The Event Triples

In their analysis of language models for visual recognition, (Le et al. 2013) forwarded their thoughts on what may be seen as an increasingly conflating dichotomy between the textual and image worlds. They had argued that "Computational linguistics has created many tools for automatic knowledge acquisition which have been successfully applied in many tasks inside the language domain, such as question answering, machine translation, semantic web, etc." (1) Interestingly, in their paper they had asked whether such knowledge generalizes to the observed reality outside the language domain, where well-known image datasets can serve as a proxy for observed reality: "In particular, we aim to determine which language model yields knowledge that is most suitable for use in Computer Vision." They had set out to test a number of language models and a linguistically minded knowledge base in the context of *Human action recognition* and *Objects in context*, using the premise of the semantic triple. Their task of human action recognition is to determine if a human exists in an image and then proceed to recognize the <subject, verb, object> triples based on objects (e.g., car, horse) and scenes (the place that the actions occur, e.g., countryside, forest, office) recognized in images. In this scenario, they only consider images with human actions so the "human" subject is always present. The second task, *Objects in context* involved the prediction of the most likely identity of an object given its context as expressed in terms of co-occurring objects in the same way we had discussed distributional semantic models above. Le et al. tested their language models in two ways: 1. By

directly comparing the statistics of the linguistic models with statistics extracted from the visual domain. 2. By comparing the linguistic models inside the two computer vision applications, leading to a direct estimation of their usefulness. They had hoped to address these research questions: 1. Is the knowledge from language compatible with the knowledge from vision? 2. Can the knowledge extracted from language help in computer vision scenarios?

Kläser (2010), in his dissertation, focuses on the problem of action recognition in realistic video material, such as movies, internet and surveillance videos. In order to be more precise about his goal, he had to clarify the meaning of an action and action recognition by an analogy to languages. Although he did not explicitly mention the triple in his proposal, he does concede to the fact that every action will compose of a subject, verb and object element as a means to decomposing the meaning of a parsed action. Those are in turn triples inevitably.

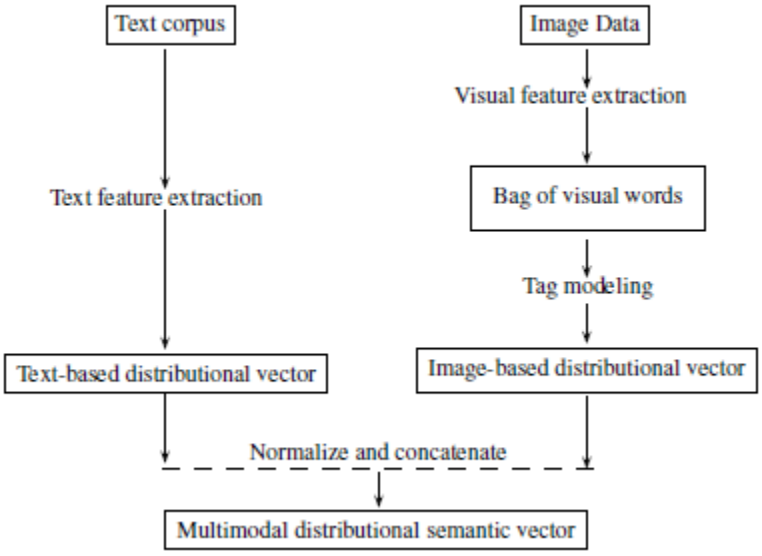

Figure 2.3 Texts and Images as a Multimodal semantic unit

Similar to Le et al., Kläser also took inspiration for his image model from natural language models. He explains, “Human language is composed of sentences which are themselves structured with subjects, verbs, and objects. In order to describe the visual content of a video in an automatic fashion, a structure similar to that of a language is necessary. From an algorithmic point of view, this translates to the detection of (a) subjects (or actors) which most commonly are humans; (b) objects which can be other humans, they can be objects, and they also include environments in which the subject is operating; (c) verbs which describe actions of the subject as

well as interactions between subjects and objects. In this sense, an action can be precisely localized in a short interval in time, yet it can also refer to an event that lasts for a rather long time period." (5) While we may not be analyzing video or internet content in this study, we mention Kläser's work here because it has relevance for the different types of scene or action recognition which can be applied to the analysis of image content from relevant games covers and videos. Understanding why the formation of triples assists in event recognition is key in any in-depth future study in this area. Beyond the triples, an understanding of action taxonomy was surveyed in (Poppe, 2010) and defined in (Moeslund et al. 2006) where an action primitive (or movement), can denote an action, and activity. An action primitive describes a basic and atomic motion entity out of which actions are built. An activity is a set of several actions. Activities can be understood as larger scale events that often depend on the context and the environment in which the action happens.

Interestingly, Le et al. were not the only ones keen on applying their knowledge of language models in the visual world. (Bruni et al. 2013) also proceeded to create a Visual Semantics Toolkit (or VSEM)[6] to assist computational linguists with cross-over research interests to explore Computer Vision within an accessible framework. The VSEM's toolkit's all-in-one integrative approach had hoped to provide investigators with a clearly recognizable processing pipeline for representing and working on image data. Within the Visual Semantics premise, an *image* is regarded as a *document* and described by general features kept in a *dictionary*. In the same way, a text document can be described by a related bag of words; an image *document* can also be described by a corresponding bag of words also, or features of visual words.

The basic pipeline for image representation is as follows: First, interesting local patches of an image are found by what we call a *detector*. These are subsequently described by *descriptor* vectors and mapped to their respective visual words from a pre-made visual dictionary. A visual word can be regarded as a cluster of similar descriptor vectors. In this manner, the whole image can be described by a visual word histogram. To arrive at a concept representation, histograms of images tagged with the same concept are aggregated. As a result, one of the ways in which object and human detection is possible in images is through the process of grouping these classes of images in a training process and then comparing the resulting histogram with unknowns. A

[6] VSEM is based on Matlab. We tested it for this study but we always ran out of memory when processing. Depending on the size of the data it can be a resource hog, crashing as a result.

pipeline for visual representation may likely be divided into these steps: i. vocabulary creation, and ii. image representation, where a common vocabulary of visual words is created by clustering lower level image features from a training set. After creating the vocabulary, the system proceeds to represent images in terms of bag-of-visual-words histograms using the following steps: i. the extraction of local image features, ii. mapping of local features to higher-level visual words contained in the vocabulary, iii. creation of bag-of-visual-words histograms, based on the mapping obtained in the previous step, and iv. spatial binning.

Differing settings can be adjusted based on the experimental objective inside the VSEM toolkit, (Bruni et al., 2013). Tentatively, a visual pipeline in VSEM could have the following settings: *Descriptors*: SIFT descriptors with gray colour scale settings.

Dictionary: k-means dictionary.

Encoding: Hard quantization.

Spatial binning: 2 square divisions, 3 horizontal divisions, giving rise to a feature vector eight times the size of the number of visual words.

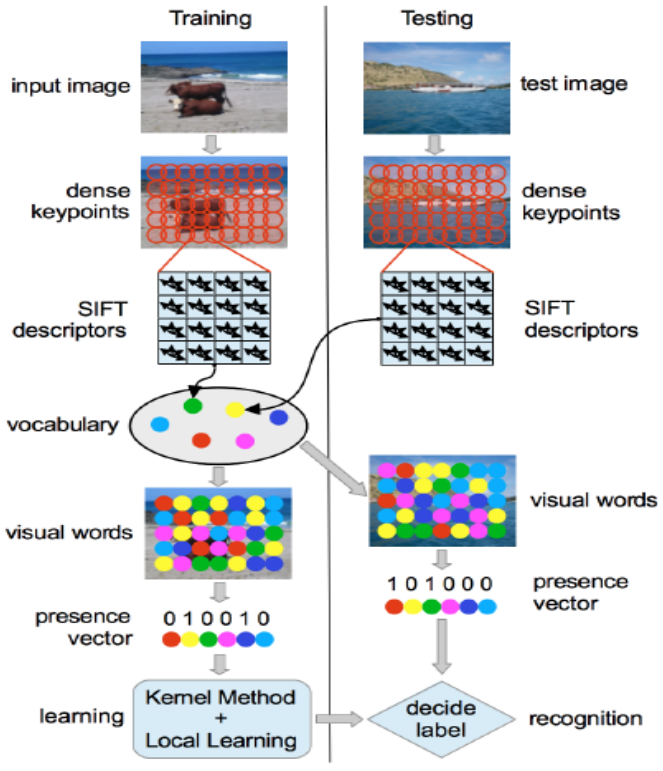

Figure 2.4 Training and testing visual words

Descriptor settings can be adjusted accordingly depending on experimental objectives. (Tomasik et al. 2009) describe their pipeline using a similar SIFT-based descriptor as above (SIFT is the Scale Invariant Feature Transform). They had used a standard "bag of visual words" image classifier, as implemented in Vedaldi's open-source Matlab package, (Vedaldi and Fulkerson, 2010). To be able to extract these bags of words, their work involved the extraction of 10,000 SIFT features from *each* image. They proceeded to collect a subset of these features from each training image and applied the hierarchical k-means clustering to construct a tree of cluster centers in SIFT-feature space. Each of these vectors can be thought of as a "visual word" that characterizes an image in some way. It is from this tree that they transformed each image into a "bag of words" by associating each of the image's SIFT vectors with the words in the tree to which it is closest. The result is a histogram of frequency counts for each word, subsequently applied to standard information-retrieval techniques like term frequency-inverse document frequency (TF-IDF) weighting and cosine similarity. We use the latter distance measure for our images in this study. On the other hand, Tomasik et al. classified their test images using a distance-weighted variant of k-nearest neighbour, in which each training image "votes" for its own category label in proportion to how much closer it is to the test image than the average training image, an insight we will also build on when discussing error pairs.

Obviously, we now notice how so much of the pipeline for image-processing contains some similarities with text-processing pipelines with the former borrowing a lot of concepts and procedures from the latter, even though texts do not understandably have a SIFT in their pipelines. In developing tools for image processing, a lot of the concepts used to analyze texts filtered into the realm of images as well: with visual words and vocabulary assuming an entirely new meaning. Sift-based bag of words models can be used to achieve so many varied image recognition tasks, including the usual facial/object detection and recognition, as well as higher level concepts like event recognition. However, one of the drawbacks when using the bag of words model is the noted loss of spatial information which does provide an additional semantic context in the analysis of an image. How useful is a visual sentence if it is not able to make sense of the regions within the image? (Tirilly et al. 2008) point this out when discussing language models for image categorization:

> Two shortcomings of this representation are the loss of the spatial information of visual words and the presence of noisy visual words due to the coarseness of the

vocabulary building process. On the one hand, we propose a new representation of images that goes further in the analogy with textual data: visual sentences, that allows us to “read” visual words in a certain order, as in the case of text. We can therefore consider simple spatial relations between words. We also present a new image classification scheme that exploits these relations. It is based on the use of language models, a very popular tool from speech and text analysis communities. On the other hand, we propose new techniques to eliminate useless words, one based on geometric properties of the keypoints, the other on the use of probabilistic Latent Semantic Analysis (pLSA). (1)

The use of spatial information for bag of words models has seen resurgence, where the Spatial Pyramid Matching technique is the most notable. One could argue that for the most part, languages are just coded images with different levels of abstraction and vice-versa. In this regard we note how the progress of text processing techniques have been reconceptualised in image analysis, producing what is already identifiable, like visual words, vocabulary and visual sentences as mentioned above. Bae and Juang (2010) applied the idea of linguistic parsing to generate the Bag of Words feature for image annotation. Specifically, images are represented by a number of variable-size patches in a multidimensional incremental parsing algorithm.

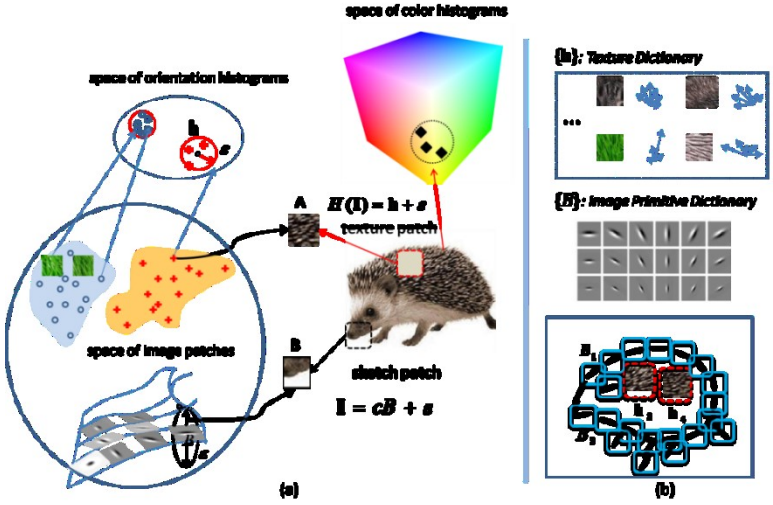

Figure 2.5 The left figure illustrates the image space and its composition. A hedgehog image may be seen as a collection of local image patches which are from different subspaces (primitive, texture, color, etc.) of varying dimensions and complexities. The right figure shows a few automatically learned hybrid image templates learned by composing the four types of patch prototypes. For each object/scene category, four example images are shown, followed by four bands of the hybrid templates.

Since each image patch represents a concept akin to how words represent a concept, they are parsed accordingly. Each patch may correspond to different properties around the image's segment (color, texture, location etc.,). Then, the occurrence pattern of these parsed visual patches is fed into a LSA (Latent Semantic Analysis) framework.

Other techniques use a multiple windowing system over the image segment and are designed to detect the same low-level features and then map them to a high-level concept inside the window. Consequently, instead of having a single window detecting only a single concept, perhaps, a cow, we instead have patches of variable sizes located on different segments of said cow being able to resolve which parts of the cow they are (as in, the cow's head, eyes, tail, and legs etc.,). Other techniques take account of the accretion of completely different visual words from completely different image datasets in order to build completely different vocabularies, and then later merge them in a larger codebook. (Lopez-Sastre et al. 2013) describe their Visual Word Aggregation (VWA) process:

> …recent category-level object and activity recognition systems work with visual words, i.e. vector-quantized local descriptors. These visual vocabularies are usually built by using a local feature, such as SIFT, and a single clustering algorithm, such as K-means. However, very different clustering algorithms are at our disposal, each of them discovering different structures in the data. (1)

Combining different vocabularies into one has the effect of countering the spatial loss problem that comes from building and using a single vocabulary. In Lopez-Sastre et al.'s approach, viewing each visual vocabulary as one unit, they offered a Visual Word Aggregation methodology to learn a common codebook, where the strength of the visual vocabulary construction process is increased, and the size of the codebook is determined in an unsupervised integration, and where more discriminative representations are obtained. They also added a contextual component to their visual words by incorporating the spatial neighbouring relation between the local descriptors in the VWA process, culminating in the Contextual-VWA (C-VWA) approach. In their own words: "We integrate over segmentation algorithms and spatial grids into the aggregation process to obtain a visual vocabulary that narrows the semantic gap between visual words and visual concepts." (1) They had also used a distance metric on clustered features to measure how unlike or similar certain objects are and cast that as a basis to classify these same objects and events.

However, while the aforementioned techniques and pipelines have been enlightening, we are still left wondering how exactly the creation of low-level dictionary features and their distances from each other actually map to a higher level understanding of human action recognitions, objects in context prediction or even scene classifications especially in the view of closing the semantic gap? Is the semantic gap merely the gap or distances between high-level features? Or even low-level descriptor patterns? Do we still need an intermediate process between these features and the concepts they might also represent? This is an important question since active research is slowly evolving beyond the mere identification of individual objects to analyzing how these objects connect or relate with one another in an image— either in a hierarchical action- based taxonomy or a loose graphical structure. Regardless of strategy, it is noteworthy pointing out that the triple as a concept could be used to understand an image either as a lower level feature (as in the numeric colour values of a pixel or shape, colour and texture) to the extremely high-level descriptions of an event, scene or human action that could possibly emerge from a higher level analysis. All the same, the lingering problem of the semantic gap still remains since there is no clear linkage between the lower level triples and the higher level ones.

2.4 Problem Background

Studies able to initially identify with the challenges of the semantic gap from an image content analysis viewpoint were those in the image-retrieval realm, according to (Smeulders, et al. 2004). Content-based image-retrieval (CBIR), which was suggested in the early 1990s, is a technique for automatically indexing images by extracting their (low-level) visual features, such as color, texture, and shape, with the retrieval of images based entirely on the indexed image features, (Kherfi et al. 2004). It was theorized that relevant images can be retrieved by calculating the similarity between the low-level visual contents, (Datta et al. 2008). However, because of the existing semantic gap between those low-level visual features and formulated user queries, that approach tended to provide unsatisfactory results. As a result, improved image annotations were suggested. The objective of image annotation is to automatically assign keywords to images, enabling image-retrieval based on aligned query images by keywords. As explained by (Tsai, 2012), “Image annotation can be regarded as the image classification problem: that images are represented by some low-level features and some supervised learning techniques are used to learn the mapping between low-level features and high-level concepts (i.e., class labels). One of

the most widely used feature representation methods is Bag of Words (BoW)." (1) Although there are merits to annotating images that are assigned keywords, it still suffers from some performance errors and also the "human-in-the-loop" problem since the amount of visual and video data has since exploded, making manual annotation impractical on such a large scale (though we still see efforts like LabelMe, Amazon Mechanical Turk, and the ESP Game, that attempt to encourage annotation with a *game* or *reward* element to it—those are still not practical for web scale image indexes). Imagine assigning a keyword to every single static image frame in a movie or video game. Perhaps the intention is to train a few annotated images and then test it on the un-annotated remaining. But therein exists another problem: No two scenes are intended to be exactly alike, informing the need to re-appraise the annotation strategy towards a visual understanding of image fundamentals and a lesser need for a human annotator in the loop. Tsai explains this further, "Typically, images are represented as points in high dimensional feature space. Then, a metric is used to measure similarity or dissimilarity between images on this space. Thus, images close to the query are similar to the query and retrieved. Although CBIR introduced automated image feature extraction and indexation, it does *not* overcome the so-called semantic gap...." (1)

The semantic gap problem still lingers on despite the massive amounts of human annotation and semi-supervised learning approaches taken to classify images. "The notation of similarity in the user's mind is typically based on high-level abstractions, such as activities, entities/objects, events, or some evoked emotions, among others. Consequently, retrieval by similarity using low-level features like color or shape will not be very effective. In other words, human similarity judgments do not obey the requirements of the similarity metric used in CBIR systems." Tsai (2012:1). We could argue on the contrary that perhaps images with darker visuals or some deep amounts of red may likely correlate with the emotions of fear, sadness, gore or even death, and those with some brightness may signify hope, joy peace and so on—such connotations do suggest that low-level features may have some direct input to capturing higher level emotional types contrary to Tsai's arguments; however, this is not always the case (the cover art image of *Silence of the Lambs* being a classic example and many gothic images that employ bright luminance).

On the other hand, say, we have games images and need to classify them based on action and emotion. How do we go about that? Some type of annotation may be required at the start, since

we need to take some positive example images that depict the classes of fear, courage, pain, joy, fun and so on, and then proceed to train our images as seeds to discover other similar types of images in the same group. It is still a very high-level of abstraction for the machine. On the other hand we could allow the machine to recognize the actions in the scene of an image and then proceed to make inferences as to what type of emotion those actions may elicit. *If* the first recognition part is solved, *then* the inference part is made easier. But both are very difficult problems to date because this process goes beyond just assigning keywords to images. For instance, when humans observe a scene, three simultaneous actions happen: i. High-level recognition (classification) ii. Identification of specific items in the scene (annotation). iii. Localization of scene components (segmentation).

All of these individual tasks contribute to understanding what the whole scene is all about and what class of emotion could be applied to it. When machines do the same, current methods at different levels of abstraction may tend to: i. Provide a single label to an image. ii. Provide multiple labels to an image without localization. iii. Separate imagery between background clutter and foreground objects, all of which can be computationally expensive tasks. This has not deterred some recently proposed models which have attempted to capture the simultaneous occurrence of multiple objects in an image along with their high-level scene classes. This strategy not only limits the entire dependence of human annotators in labelling images, but has also resulted in more accurate *semantic* representations of the observed images. Once the visual object recognition algorithm recognizes the individual objects in the image, it can then proceed to make inferences as to what class of scene or action the image belongs to or what type of emotional value it evokes, based on the nature of the co-occurring objects it has discovered in the image.

2.5 Image-retrieval with Semantic Features

A few techniques have been proffered to deal with the semantic gap presented in images and their textual descriptions. A key difference between content-based and text-based retrieval systems is the fact that the human interaction is a crucial part of the latter system. Humans tend to use high-level features (concepts), such as keywords, text descriptors, to interpret images and measure their similarity. Obviously, the features automatically extracted using computer vision

techniques are mostly low-level ones (colour, texture, shape, spatial layout, etc.), and consequently, there is no direct link between the high-level concepts and the low-level features.

Even though many sophisticated algorithms have been developed to describe colour, shape, and texture features, (depicting progress in the right direction), on the whole, these algorithms cannot yet adequately model image semantics and do have numerous limitations when dealing with broad content image databases. Furthermore, so much of the available image datasets currently under active study largely concern the analysis of images around everyday objects (cars, planes, faces, chairs etc.), exposing the fact that there has been so little study of image content around reality replicas or imaginary concepts that may appear as modelled figures in drawings or games. Applying a generic facial recognition algorithm that has been trained on normal images to images games' covers is likely to reveal how much failure can result from the system attempting to grasp what constitutes an approximation or exaggeration of reality. For instance, the images encountered for this study either showed a lot of blurry, small, blank or almost empty faces; or revealed caricatures that depicted monsters. A generic facial recognition system will likely fail in detecting those types of anomalies unless a dedicated dataset with the target content has been properly marked out and trained for the purpose of detecting the types of peculiar faces or monsters that may appear in video games. (As noted earlier, a small script was written for this study to clearly mark out regions of interest in the games image content for a Haar-cascade classifier). Delineating a region of interest in an image is a semantic task that enables the algorithm to take special notice of particular regions of image contents during the training process.

However, there is a difference between *marking* regions of interest for training possibly using a tool like the Object Marker and the dedicated process of image segmentation. When we simply mark out regions of interest around objects in images we only desire that the end result be the simple detection of similar objects in varied images usually with a descriptive bounding box around the object. Segmentations tend to outline the objects in their own field.

There are two types of segmentations as well. Normal segmentation models that only depend on the pixel values of the affected regions, and Landmark-based segmentation models where a lot more attention is paid to the entire surface as well as the sub-surface of the affected regions. Object-delineated annotation for segmentation in the latter case is a usually more involved process when compared to a simple box-based or pixel-value marking since it generally involves

the use of more complex landmark annotations and precise measurements that subdivides an object surface into many smaller sub-parts: see (Cootes, 1995) research into Active Shape Models (ASM) and Active Appearance Models (AAM).

However, we will only focus on regional markings as training pre-process for images in this study since we had also used box-based regional markers to determine image areas and thus objects of interest. For instance, if we wanted to make our training algorithm recognize a shoe in an image, we would have to manually mark out many training images containing different types of shoes. During the testing phase when the unseen data has to recognize shoes, the areas occupied by shoes will have a bounding box to signify a resolved recognition or could be segmented in the displayed result. Nonetheless, as stated before, ASM-based segmentation is a more involved process both in the annotation and training phases given that its recognition accuracy is expected to be a lot higher. ASM-based segmentation models are used in tasks that demand precise recognition, like facial recognition. While, on the other hand, normal pixel value based segmentations are used in simple object detection processes. This is also reflected by (Nalina and Muthukannan, 2013):

> … dividing an image into sub partitions on the basis of some similar characteristics like color, intensity and texture is called image segmentation. The goal of segmentation is to change the representation of an image into something more meaningful and easier to analyze. Image segmentation is normally used to locate objects and boundaries that are lines, curves, etc. in images. Segmentation can be done by detecting edges or points or line in the image. When we detect the points in an image then on the basis of similarities between any two points we can make them into separate regions. (1)

Consequently, we are now observing a higher form abstraction beyond colour, location, and texture to that of a set of regions in an image. Because of *segmentation* a set of regions is produced. Each region in turn is a set of image elements belonging to that set. The grouping of these elements into regions states a relationship between them: they are believed to belong to the same object. That is, when the image elements in one region share a set of properties, they are said to be similar. Between different adjacent regions there is discontinuity. Most often, all elements in each region have to be connected with each other. This is one constraint which can be applied on each region, the connectivity constraint. Often, each region also has to fulfill

certain regularity; for instance being smooth to some degree or have a fixed topology. Typically, segmented regions are made to represent a single object or concept. Their spatial layout in the image can be represented in a graphical tree structure with the familiar root and branching leaf nodes that can be found in some in textual analysis (like trees in context-free grammars). Segmented regions with different colors can also represent the canonical *ground truth*[7] for a class of images. Johnson et al. (2006) describes how semantic labels affect the understanding and representation of an image since the process encompasses the problems of object detection, recognition, and segmentation, therefore expanding the range of relevant semantic labels. In other words, a segment constitutes a semantic label and the automated regional segmentation of an image region is also a type of annotation since semantic labels are assigned. Johnson et al. also highlight how the newest algorithms tend to consider image regions in the context of the rest of the image, counting other clever approaches for retrieving images from automatically classified image libraries. (Li et al. 2011) seem to concur with this approach in their work:

> Our proposed model captures the co-occurrences of object and high-level scene classes. Recognition becomes more accurate when different semantic components of an image are simultaneously recognized, allowing each component to provide contextual constraints to facilitate the recognition of the others. In addition, both object recognition within a scene as well as scene classification can benefit from understanding the spatial extents of each semantic concept. Our model can recognize and segment multiple objects as well as classify scenes in one coherent framework. (1)

Accordingly, multiple object segmentations and labelling are said to actually feed the event or scene recognition process. Their spatial structure in relation to one another in the image captures another important semantic property, spatial contexts for objects (the sky region is always up, the ground or water is always below, the clouds in between, and anything else, just above the ground). This is also not to say that the fast detection of multiple objects in an image is already a solved problem. It is still a hard problem. So many familiar objects may appear different when viewed from different angles, or under poor illumination, and this tends to disturb the accuracy of the detection process. Furthermore, occlusions, deformations and the great variety some objects can take, can also affect how they can be detected. This is why so much work has been

[7] Ground truth refers to the accuracy of the training set's classification for supervised learning techniques.

done in perfecting image descriptors that can consistently locate and describe images using invariant properties. This was highlighted by (Nixon, 2008):

> Other important invariance properties naturally include scale and position and also invariance to affine and perspective changes. These last two properties are very important when recognising objects observed from different viewpoints. In addition to these three properties, the descriptors should be a compact set. Namely, a descriptor should represent the essence of an object in an efficient way. That is, it should only contain information about what makes an object *unique*, or different from the other objects. The quantity of information used to describe this characterisation should be less than the information necessary to have a complete description of the object itself. Unfortunately, there is *no* set of complete and compact descriptors to characterise general objects. Thus, the best recognition performance is obtained by carefully selected properties. As such, the process of recognition is strongly related to each particular application with a particular type of object. (281)

The last point is important for us because it helps to explain the reason why so many of the tools and algorithms studied for this thesis research had solutions that were simply optimized for a particular dataset, making it either impossible to suddenly use a game-cover image data for processing; or if it were even possible to use them, they resulted in so many errors because the studied algorithms could not generalize their learning models to new domains. Of course, this might mean the need to tweak a lot from the settings of the utilized descriptors or overhauling the entire code behind the tools themselves in which time did not just permit. Essentially, descriptors that have been used to build a visual vocabulary around facial recognition; gender or emotional recognition, cannot suddenly be used to analyze a chair for instance. There are not that many applications that are generalized. And because there is no set of complete and compact descriptors that can characterize general objects, it remains an open area of research. An integrated framework may have large and diverse datasets in addition to self-contained tools of varied kinds in a single framework, and is able to select the appropriate algorithm and descriptors for processing the appropriate testing image. In the course of this study it was discovered that tens of these types of tools are still in a developmental or proof of concept stage and were not so practical for the analysis of games image content.

2.6 Levels of Retrieval

Extensive experiments on various visual object recognition techniques and CBIR systems show that low-level contents often fail to describe the high-level semantic concepts in a user's mind, hinting that so much remains to be done to match the accuracy levels of text-based retrieval systems. This is informed by the fact that Computer Vision is a far more difficult problem and initial user anticipations usually run high. While progress is still being made, the performances of these systems still fall short of current user's expectations. In the context of CBIR, there are three query levels according to (Min and Yang, 2010), also marking the levels of difficulty surrounding the problem:

> *Level 1*: Retrieval by primitive features such as colour, texture, shape or the spatial location of image elements. Typical query is query by example, 'find pictures like this'.
>
> *Level 2*: Retrieval of objects of given type identified by derived features, with some degree of logical inference. For example, 'find a picture of a flower'.
>
> *Level 3*: Retrieval by abstract attributes, involving a significant amount of high-level reasoning about the purpose of the objects or scenes depicted. This includes retrieval of named events, of pictures with emotional or religious significance, etc. Query example, 'find pictures of joyful crowd'. Levels 2 and 3 together are referred to as semantic image-retrieval, and the gap between Levels 1 and 2 as the semantic gap. (2)

(Min and Yang, 2010) also point out the discrepancy between the limited descriptive power of low-level image features and the richness of user semantics as a 'semantic gap'. Users in Level 1 retrieval are usually required to submit an example image or sketch as a query. But what if the user does not have an example image at hand? This results in the reliance of semantic image-retrieval as a more convenient means for users since it supports query by keywords or by texture. Conversely, in high-level semantic-based image-retrieval, low-level image features can be related with high-level semantic features as a way of reducing the 'semantic gap'. Low-level image features and high-level semantic features exist in two completely different contexts hinting of an even wider gap than realized. Some have stated the need for ontologically-driven approaches as the appropriate to model as these take additional contexts into account as a strategy to narrowing this gap. (Bannour and Hudelot, 2011) had also forwarded three types of

hierarchies for computer vision: i. A language-based hierarchy, ii. visual hierarchy and iii. a semantic hierarchy: based on both semantic and visual features, which encompasses an ontological-driven process, unlike other techniques, and allows for a semantic description of images. They argue that this is best suited for image-retrieval systems because they model the semantics of images through relationships that help reasoning about it and understanding its meaning. In addition, they observed that an appropriate ontology can make explicit the relationships between the labels and concepts. Tu et al. (2005) seem to have taken the language-based hierarchy approach quite literarily. Theirs is a graphical approach that uses a type of hierarchy inspired by the parsing graph of languages. In their work that attempts unifying segmentation, detection, and recognition, they explain:

> The parsing algorithm optimizes the posterior probability and outputs a scene representation in a "parsing graph", in a spirit similar to parsing sentences in speech and natural language. The algorithm constructs the parsing graph and re-configures it dynamically using a set of reversible Markov chain jumps. This computational framework integrates two popular inference approaches – generative (top-down) methods and discriminative (bottom-up) methods. The former formulates the posterior probability in terms of generative models for images defined by likelihood functions and priors. The latter computes discriminative probabilities based on a sequence (cascade) of bottom-up tests/filters. (1)

Being a set of reversible Markov chain jumps, their image parsing algorithm is not rigid (as in a series of pre-determined fixed templates) and must construct the parsing graph on the fly. Hence, making their ontological framework a statistically driven one with each type of chain jump corresponding to an operator for reconfiguring the parsing graph. Generated templates are a lot more dynamic in this type scene interpretation, following the values and outcomes from priors and posteriors on successive nodes. This may find uses in the parsing of games image content that require re-adapting reality. However, this purely data-driven approach in truth is a little harder to achieve requiring more sophisticated algorithms. On the other hand, (Min and Yang 2010) tend to agree more with Bannour and Hudelot's line of thinking regarding ontology models, but theirs is a hybrid approach. They proceed to suggest five techniques that can connect the different contexts: i. Using object ontology to define high-level concepts, ii. Using machine learning tools to associate low-level features with query concepts, iii. Introducing relevance

feedback (RF) into the retrieval loop for continuous learning of users' intention. iv. Generating semantic templates (ST) to support high-level image-retrieval, v. Making use of both the visual content of images and the textual information obtained from the Web in situations where WWW (the Web) images and pages are retrieved as a type of context for the images. Indeed, Min and Yang's opting for a more synthetic approach that embraces the semantic model as well as the machine learning model, including the human model thrown in the mix for a relevance feedback is an attempt to ascertain a feasible fusion solution for web-based image-retrieval. Evidently, in more practical web usage contexts, people will prefer to search for images using textual descriptions. The current state-of-the-art on the web still finds texts and textual-based metadata controlling the way searches are made and web items found. On the other hand, with the increasing use of different mobile devices, there is a growing interest in applications that enable a *Level-1* retrieval type visual task, especially with faster image processing, because at the moment, most image applications in this area are slow and far from perfect. Graphical approaches that parse images on the fly should be welcomed. But there is still a problem of processor and network bottlenecks and speed to contend with. When images are captured on a mobile device, should the processing be done on the same device, or should the app extract features and send them to a remote server? Transmitting whole images to a faster remote server is not practical (images can run in the megabytes, costing more to transmit from a phone). Processing the images locally on the phone, with current techniques, is not so practical as well (processor speeds and storage is limited). A compromise will likely involve extracting features from the images on the mobile device and transmitting these features to a remote server. But even this can be an involved process for the mobile devices, delaying the much needed instantaneous response for the image processing tasks. This is why so much research focus has gone the direction of developing faster and more accurate feature descriptors to take advantage of these types of contexts. Graphical approaches to parsing images have been viewed as a faster class of techniques that can be utilized in contexts where response times are critical. It is anticipated that real-time feature-driven visual tasks will likely surge when recognition accuracies increase for various types of search contexts that are meaningful for the user. This underscores the key importance of feature descriptors in conflating the semantic gap, bridging input signals to learnable features. It also highlights the reason why our scope for this study will

focus on these descriptors as described in the next Chapter. We will examine how accurate these feature descriptors are in different image-retrieval contexts.

2.7 Object-ontology

In systems utilizing simple semantics, different intervals may be defined for low-level image features, with each interval corresponding to an intermediate-level descriptor of images. For example: 'light green, medium green, dark green'. These descriptors form a simple vocabulary, the so-called 'object-ontology' which provides a qualitative definition of high-level query concepts (notice that the descriptor also appears as a *triple*). Database images can be classified into different categories by mapping such descriptors to high-level semantics (keywords) based on our knowledge. For example, 'sky' can be defined as region of 'light blue' (colour), 'uniform' (texture), and 'upper' (spatial location) -- again, notice the triple defining an object or region's meaning in terms of *colour*, *texture* and *location* in the image.

Hare et al. (2006) conducted a survey around content-based image-retrieval focusing on the lack of coincidence between the information that one can extract from the visual data and the interpretation that the same data has for a user in a given situation. At the end of the survey the authors concluded that:

> A critical point in the advancement of content-based retrieval is the semantic gap, where the meaning of an image is rarely self-evident. The aim of content-based retrieval systems must be to provide maximum support in bridging the semantic gap between the simplicity of available visual features and the richness of the user semantics. (2)

Some techniques which aim at bridging the semantic gap in retrieval systems have mostly used an auto-annotation approach, where keyword annotations are applied to unlabelled images. Hare et al. discuss some short-comings of auto-annotation due to their lack of richness when compared to real image annotations in archival collections. They go on to suggest that perhaps a way forward is to combine shareable ontologies to make explicit the relationships between the keyword labels and the concepts they represent. For example, a number of auto-annotation techniques directly associate descriptors with labels, without any concept of objects. The first attempt at automatic annotation was perhaps the work of (Mori et al. 1999) which applied a co-occurrence model to keywords and low-level features. (Duygulu et al. 2002) will later use that

insight to learn a lexicon and apply it to a fixed image vocabulary. (Zhao and Grosky, 2002) proposed an approach to bridging the semantic gap using Latent Semantic Indexing (LSI). This necessitated the creation of a semantic-space involving the use of linear algebra to associate images and terms, thus, avoiding the need for words or labels (the Latent Semantic Indexing is a technique originally developed for textual information retrieval). Nonetheless, the surging interest amongst some researchers on the use of ontologies and semantic web tools with multimedia collections seems to be on the rise were also on the rise on par with data driven approaches. "Semantic descriptions of non-textual media can be used to facilitate retrieval and presentation of media assets and documents containing them. Existing multimedia metadata standards, such as MPEG-7, provide a means of associating semantics with particular sections of audio-visual material. While technologies for multimedia semantic descriptions already exist, there is as yet no formal description of a high quality multimedia ontology that is compatible with existing (semantic) web technologies. " (Hardman, 2008).

In a study of annotations explain their observations: "Due to the well-known semantic gap problem, a wide number of approaches have been proposed during the last decade for automatic image annotation, i.e. the textual description of images. Since these approaches are still not sufficiently efficient, a new trend is to use semantic hierarchies of concepts or ontologies to improve the image annotation process. This paper presents an overview and an analysis of the use of semantic hierarchies and ontologies to provide a deeper image understanding and a better image annotation." (Bannour and Hudelot, 2011:1) How practical or feasible these are in the context of the semantic gap problem remains to be seen. Earlier work on semantically describing images using ontologies as a tool for annotating and searching images more intelligently was described by (Schreiber et al. 2001). A number of authors like (Hare et al. 2006) have also described efforts to move the MPEG-7 description of multimedia information closer to ontology languages such as RDF and OWL. However, in their work, Bannor and Hudelot also fixated on the role of context as a key driver in disambiguating images:

> Always in the quest for models that could help to map successfully low-level features into high-level semantic concepts, some approaches make use of "contextual knowledge" by building semantic hierarchies or integrating a priori knowledge to improve image analysis and interpretation. Indeed, objects in the real world are

> always seen embedded in a specific context, and the representation of that context is essential for the analysis and the understanding of images. (1)

Again, their observation merely echoes what we have previously discussed on label co-occurrence and distributional semantic models all of which tended to use the cluster of recognized objects in an image as a context. (Bannour and Hudelot, 2011) on the other hand seem to be hinged on topology as an additional context for a resolution constraint:

> Contextual knowledge for image interpretation may stem from multiple sources of information, including knowledge about the expected identity, size, position and relative depth of an object within a scene. For example, topological knowledge can provide information about which objects are most likely to appear within a specific visual setting, e.g. an office typically contains a desk, a phone, and a computer, but it is unlikely that it contains a bed. Spatial information can also provide information about which locations within a visual setting are most likely to contain objects, e.g. in a beach scene, the sky is usually placed at the top, while the sea is below. Given a specific context, this kind of knowledge can help reasoning on data to improve image annotation. (1)

Graph theory and a graphical approach to the analysis of objects as a unit and objects in relation to others in an image is key. We observe how much the topological analysis of objects and regions in images has informed the work of the likes of (Cootes, 1995), who has used both regional segmentations and graphical analysis to recognize faces and objects, in his Active Shape Models proposal. However, it is likely that Bannour and Hudelot's idea of spatial analysis and topology may not be in the mould of entirely data-driven templates like Coote's, which includes additional sub-segmentation of object surfaces beyond the identified regional segment as an object unit. Topology in this regard is utilized differently by Cootes. Bannour and Hudelot's much simpler position on the other hand is the notion that ontologies are a means of capturing the relevant knowledge of a domain, thereby providing a common understanding of this domain knowledge. Furthermore, it includes the determination of the acknowledged vocabulary of this domain, as well as giving the explicit definition of the vocabulary (terms) and the relations between these vocabularies in formal models at different levels. Their notion of spatial analysis does not zero in on sub-surface segmentation, but instead view the concept from a broader position of inter-object structure and relationship that is peculiar to certain types of scene classes.

Thus, when observing an image of a sunset, seen here as a domain, it is assumed from a spatial and topological standpoint that the sun, clouds, grass or trees, occupy very strategic positions in a picture that will be replicated in many varied images of the same type. If a busy street is another domain, we can also extend the same idea of fixed segmented domains in the image space's ground truth. A street sign will occupy a position below a cloud and besides a building or car for instance. A graphical structure modelling this ontology can also be depicted from this analysis. However, when we apply this understanding to the study of games content or cover images, all the regular notions of a domain simply stops. This is because our ontological models from reality may not necessarily map to the structures that are in a games world in many cases. What kind of ontological model does a cover art like the *Super Mario Galaxy* depict? The natural world and its ontologies are simply reversed in so many fantasy realms. It could vary with genre but it is highly likely that models from the fantasy genre will likely produce very inconsistent and varied ontological representations since their goal is rewriting the rules of reality in their own self-contained worlds, in most cases. Other than that, it is possible sharing a visual vocabulary of games image content across different genres and clustering them into various families. But there is no existing games vocabulary that aims to encode the structure of these worlds at the moment, for eventual object, facial and scene recognition unique to that world.

In image-retrieval, the application of ontologies usually targets the following objectives, as identified by (Bannour and Hudelot, 2011): i. A unified description of low-level features: where ontologies are used to provide a standard description of low-level features. ii. A visual description ontology: where ontologies are used to represent the different types of relations among image features such as edges, lines and regions. iii. Knowledge description: ontologies are used to model the concepts (objects) and relations among them. Typically, these approaches use *reasoning* on concepts or on contextual information, (i.e. after the image analysis or visual object recognition process) tackling the problem of image interpretation along the way.
iv. Semantic mapping: ontologies are used to help the mapping between the visual level and the semantic level. With regards to the last point, one could argue that machine learning techniques that learn a visual model can identify obvious and hidden patterns between the visual and semantic features and appropriately map them. But if we define our expectations from ontology, how can an algorithm use machine learning models to discover hidden patterns? How consistent can a semantic hierarchy work with a learning model?

Bannour and Hudelot argue that the use of semantic hierarchies, which are based on visual and semantic information, is more convenient as it cares about perceptual and conceptual semantics. In other words, when analyzing images semantic hierarchies are important because they try to connect low-level features with higher level concepts. Conceptual semantics alone (as observed in texts) may not correspond to image semantics, but a "semantic-visual" representation attempts to tie the two worlds together in a hierarchy that could help reasoning on both images and concepts. However, building and using concept hierarchies for image analysis constrains the reasoning to the inheritance relationships, i.e. "is-a" relationship in a top-down model. While it should enrich the types of relationships used to reason about images (counting composition relationships, spatial, topological, etc.) they do benefit from the strong reasoning power on contextual knowledge but are less useful in generalizing to newer contexts because of the rigidly inherent top-down *is-a* ontology. Contrariwise, building a "semantic network" for image analysis instead of semantic hierarchies was suggested by the researchers as a good way to narrowing the semantic gap and to improving image semantics modeling, since it would certainly allow for the free association of concepts easily, in a flexible bottom-up approach. Yes, fishes and sharks and buildings can appear and locate in the sky too (referencing the cover art of *Sharknado* and *Inception*). When structural scene rules are re-written this way, image parsers that work with natural world examples will likely fail here.

Approaches that simply provide a latent correlation between the low-level features and their descriptive tags are types of semantic hierarchies (hierarchies of concepts) and are also a particular type of ontology. It would seem as if the idea of correlations put in this hierarchy, might be potentially evoking the property of conceptual inheritance for detected objects. This idea, subtly suggested, can be defended or argued for. If objects in an image co-occur based on their relative positions in the image parsing graph, do they necessarily inherit some global property? Obviously, yes. It is a possibility especially when we make inferences. The presence of objects like a pot, a stove, and assorted ingredients may signal and elicit the verb *cooking* shared by each of the co-occurring objects in the scene, thereby enabling the possible inheritance of that global inferred property, even though these individual objects may share little or no other linking property with each other (based on size, colour, texture and so on). But in a semantic hierarchy, that single verb connects them together loosely based on location or based on their functionality triggered by the dominant inferred verb (A pot on a stove is used to cook food). These, according

to Bannour and Hudelot appear as collections of classes ordered by the transitive closure of explicitly declared subclass or subtype relations. Their clarifying example was: Given that **A** is a subclass of **B**, captures the fact that the state and the behaviour of the elements of A are coherent with the intended meaning of B, while disregarding the additional features and functionalities that characterize the subclass. Meaning that we could grasp an object's intent when it is contained in a context. This can be explained in another location- based example. In scene recognition for instance, co-occurring objects in A can describe a scene or event class B, as in: a knife, along with pots, cookers and tables may denote a *kitchen* scene, but a knife could also appear with guns, blood, bombs, ropes and helmets to denote an entirely different class. In both scenarios, the type of event or class B describes the exact functionality of what a knife will be used for. The fact that an object like knife can co-occur with different objects to predict entirely different scenes or events suitably highlights the weakness of rigid semantic hierarchies. However, semantic hierarchies are still being used in image-retrieval as a framework for hierarchical image classification, to consequently provide multi-level image annotation, usually when automated segmentations of these objects occur. Nonetheless, a rigid semantic hierarchy may not have the predictive power to detect how objects may appear in new contexts, hence the suggestion for semantic *networks* instead. Semantic networks are likely to discover or allow for the prediction of new, out-of-context objects and scene classes. At the moment, much of the research in semantic hierarchies and networks is for enabling more accurate object and scene recognition in cluttered images. It should be stated that there is a clear difference between object recognition/categorization and scene classification although the two processes can be combined into one. This has been discussed by (Quel et al. 2005).

At the other extreme end of the image analysis spectrum, some methods have taken the opposite approach tending to avoid words or direct labelling altogether when making similarity judgments around diverse contents or bridging the semantic gap. For instance, by opting for a mathematical representation using LSI, therefore avoiding the need for labels as the primary descriptor, a high-dimensional space for both lower-level features and descriptions can be actualized. Some efforts generalise CLIR (Cross Language - Latent Semantic Indexing), where any *document* (be it text, image, or even video) can be described by a series of observations made about its content, as explained by (Hare et al. 2006). We refer to each of these observations as *terms*. In order to create a semantic-space for searching images, we first create a 'training'

matrix of terms and documents that describe observations about a set of annotated training images; these observations consist of low-level *descriptors* and observations with keywords that occur in each of the images. LSI is then applied to this training term-document matrix. The final stage in building the semantic-space is to 'fold-in' the corpus of un-annotated images, using purely *visual observations*. The result of this process is two matrices; one representing the coordinates of the terms in the semantic space, and the other representing the coordinates of documents in the space. Similarity of terms and documents can be assessed by calculating the angle between the respective coordinate vectors. This approach is based on the actual feature properties and their relative distances. The resulting semantic spaces are quite similar to applications that use the vector space models. According to Liska (2013), "vector space models have proven to be successful in many applications, tending to represent the meanings of concepts or words as points in high-dimensional arithmetic vector spaces, also referred to as semantic spaces. There are at least two good reasons to use vector spaces. First, individual vector components can stand for specific features (such as size, *animacy* or context), which is a natural way to characterize a concept. Secondly, the notion of "distance" or "similarity" between concepts reduces to the distance between representation vectors in the vector space." (2)

Meaning that features that cluster closer together or towards common centres called centroids, may represent concepts that are quite similar, but those farther away from these centres or features in the semantic space will likely have little or nothing in common. With regards to the generalisation of these terms to unknowns, machine learning techniques have become the norm in most computer vision research. Lillywhite (2013) elucidate on this:

> One of the main goals of computer vision is to take raw sensor data, the input signal, and create a set of symbols that represents the data. In object recognition these symbols are referred to as features. Machine learning techniques are commonly used to take these features and then classify them as either belonging to the object of interest or not. In general, machine learning algorithms take in symbols, find patterns in the symbols, and use mathematical methods to separate the symbols into classes. (1)

Learning models free the annotator from having to identify or create rules for classification based on unique image patterns and are in general more accurate than human experts (thus, lessening the need for further annotators). These machine learning models require that the set of

features uniquely describes the object of interest in order to be more accurate in finding those unique patterns. This ability is in turn dependent on the quality and quantity of data used for training, in addition to the quality of the object descriptor used to extract the features. Keypoint descriptors like Sift employ complex pattern-finding methods in order to detect and extract features from an image at the lowest level. Consequently, learning models are used both in the extraction process, and also at a higher level when these features are mapped into a higher dimension. Nevertheless, it is noteworthy pointing out that there are actually levels in the complexity of the type of processing and learning that can be applied on images. Using feature descriptors to mark out the areas of interests on the image is a basic start. Clustering them and creating a type of vocabulary will depend on the strategy and objective. Learning models and distance functions are usually applied on features that have been encoded into a vocabulary. We are simply distinguishing between pattern finding at the raw pixel level and pattern finding that mostly uses various distance functions to learn more about the hidden structures and relations of high-dimensional features in the images. Finding these unique patterns will always depend on the type of training data used. Even as the image processing used to create a higher-level representation of the input bridges the semantic gap that exists between the raw input signal and what is needed by the machine learning algorithm, the gap still remains when we view it from a user's perspective, since the abstract descriptions likely utilized by the user is still far removed from the possible distances or combinations of these image representations.

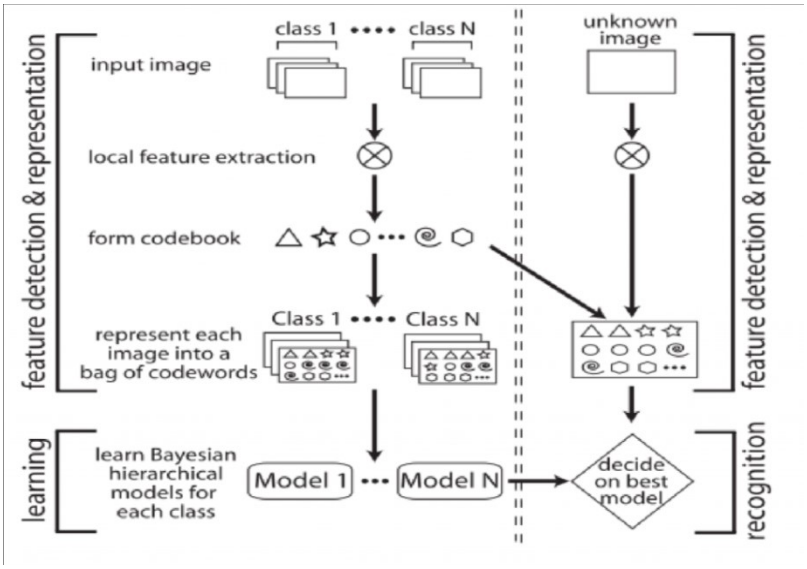

Figure 2.6 One of the hardest goals for a learning model is to find the representative prototypes from the codebook image representation.

Merging the concepts of ontology and machine learning for images is still a nascent field.

2.8 The Sift Descriptor

The SIFT descriptor (Scale Invariant Feature Transform) was proposed by Lowe (1999) and further described in Lowe (2004). It includes both a feature detector and descriptor. This algorithm converts an image into a large set of local feature vectors that describe key regions in the image called SIFT keys. These descriptors are invariant to scaling, rotation, and translation, and partially invariant to illumination changes and affine projection to enable them to extract low-level features in images regardless of the slight changes. Sift locates key points on the same image consistently regardless of the aforementioned changes. The concept of *repeatability* is important for object descriptors, and that is: a useful and efficient descriptor must be able to consistently detect and describe the key features in an image no matter how many times it is processed with that objective from varying angles or positions. Sift has become one of the most widely used descriptors to date. For this study we will be using it as our baseline and comparing it against other descriptors that have been proposed to replace or complement it. A survey conducted by Mikolajczyk and Schmid (2005) to compare the performances of different descriptors showed that Sift performed better than all other local descriptors. Sift is capable of efficiently identifying stable key locations using the following steps to extract a set of descriptors from an image: i. Scale-space extrema detection, ii. Key point localisation, iii. Orientation assignment, iv. Key point description.

The Sift descriptor is based on the idea of using the local gradient patch around a point to build a representation for the point. This representation is built by generating multiple orientation histograms for the patch. Given a feature point in the image and a square patch around it which has been appropriately scaled and rotated, the gradient magnitudes and orientations in the patch are used to generate orientation histograms over a 4x4 region. For each orientation histogram 8 bins are used. The final descriptor is computed by concatenating the outputs of 16 orientation histograms which results in a 128 element feature descriptor. A number of processes are integrated into the construction of the orientation histograms so as to improve the robustness of the SIFT descriptor. One such process involves weighting the magnitude of gradient samples by a Gaussian function centered on the keypoint before the samples are added to the orientation

histograms. This ensures that samples which are closer to the center are more significant and are assigned more weight.

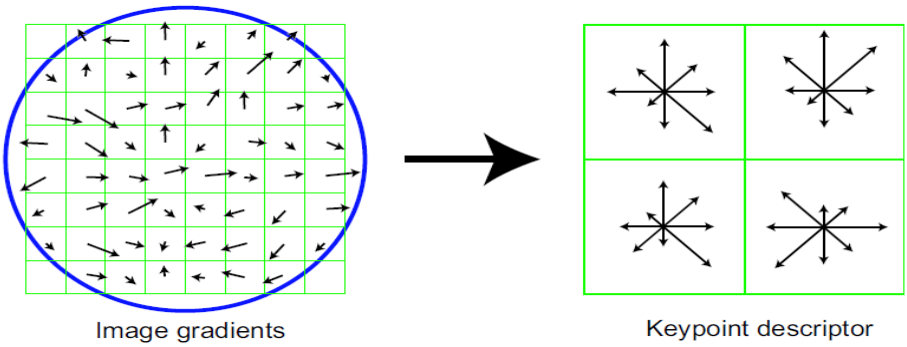

Figure 2.7 A keypoint descriptor is created by first computing the gradient magnitude and orientation at each image sample point, as shown on the left. These are weighted by a Gaussian window, indicated by the overlayed circle. These samples are then accumulated into orientation histograms summarizing the contents over larger regions, as shown on the right, with the length of each arrow corresponding to the sum of the gradient magnitudes near that direction within the region.

It also makes the descriptor less sensitive to small positional shifts in the local patch. In the final transformation stages, Sift requires its descriptor be invariant to changes in image illumination. Normalizing the descriptor to unit length helps in overcoming any brightness or contrast changes that may have occurred in the image. But, this does not dispose of the non-linear illumination changes that may have taken place. Therefore, to overcome such changes, a thresholding operation is performed to restrict the maximum gradient magnitude of the descriptor followed by a normalization operation. Sift uses a similarity measure based on the ratio of Euclidean distances, which is a measure calculated by computing the ratio of distances to the closest and the second closest neighbour for a given descriptor. It is assumed that the nearest neighbour is a correct match while the second nearest is an incorrect match. It has been shown that it is easier to differentiate between correct and incorrect matches using this measure based on ratio of distances rather than using the distance of nearest neighbour alone. Sift performs better when the similarity measure used is distance ratio matching, whereas the PCA-SIFT (Principal Component-Sift) descriptor gives better results for nearest neighbour matching; this has been experimentally confirmed by Mikolajczyk and Schmid (2005). This applies when comparing the performances of different descriptors for both the nearest neighbour similarity measure and the distance ratio measure which we will demonstrate in the next Chapter. The Sift computing steps is explained by Lowe (1999):

i. First, detect keypoints using the SIFT detector, which also detects scale and orientation of the keypoint.

ii. Next, for a given keypoint, warp the region around it to canonical orientation and scale and resize the region to 16X16 pixels

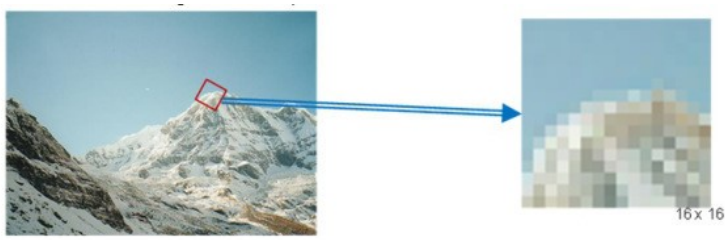

Figure 2.8 An interest point warped

iii. Compute the gradients for each pixels (orientation and magnitude).

iv. Divide the pixels into 16, 4X4 squares.

v. For each square, compute gradient direction histogram over 8 directions

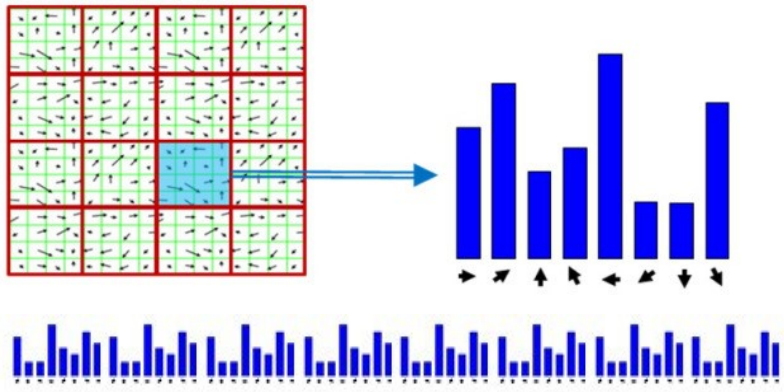

Figure 2.9 Concatenate the histograms to obtain a 128 (16*8) dimensional feature vector

There are other types of descriptors besides Sift. We will be comparing some of them in the next chapter. Some object description techniques forgo a histogram of gradient techniques altogether and choose to reconceptualise patches of pixels as textons. According to Malik et al. (1999), "Julesz introduced the term *texton*, analogous to a phoneme in speech recognition, more than 20 years ago as the putative units of pre-attentive human texture perception." Textons are defined qualitatively for simple binary line segment stimuli using oriented segments, crossings and terminators as primitives, even though they did not provide an operational definition for gray-level images. Consequently, texton theory fell into disfavor as a model of human texture discrimination when models like Sift became popular. However, novel techniques based on the

work of Shotton et al. (2008) uses a more advanced concept of the texton for cutting-edge event categorization and image segmentation.

2.9 Images as Visual Words

As indicated earlier, it is possible to encode an image in terms of *visual words*. Before that is done, we need to first encode these images as a *dictionary*. This dictionary is derived from a large set of training images that are unlabeled, but known to contain examples of all of the scenes or objects that will ultimately be classified. Usually similar points in the image are grouped into one visual word using algorithms like k-means. Agglomerative techniques are used to cluster features and to identify similar points from different images which are then more easily compared. Csurka et al. (2004) built a generative bag of visual words model to distinguish between examples of books, bicycles, people, buildings, cars, trees, and phones. Despite the wide variety of visual appearances within each class, they achieved a 72% correct classification. By applying a discriminative approach to the same problem, they managed to improve performance further. However, there are a number of drawbacks to the generative bag of words model as it:
i. assumes that the words are generated independently, although this is not necessarily true. The presence of a particular visual word tells us about the likelihood of observing other words. ii. But it ignores *spatial* information. Consequently, when applied to object recognition, it cannot tell us *where* the object is in the image. This is solved with spatial pyramid approaches among others.

2.10 Bag of Words

The bag-of-words methodology was first advocated within the problem area of the text retrieval domain for document analysis, but has since been adapted for computer vision applications. It is because the BoW model is a more accessible type of technique that has also enabled an extensive range of application successes that has led to a wider adoption by different researchers working on different types of image analysis tasks. For image analysis, a visual equivalent of a word is used in the BoW model, which is based on the vector quantization process by clustering low-level visual features of local regions or points, such as color, texture, and so forth. In explaining the bag of words for action recognition, Klaser (2010) clarifies that "it originates from document retrieval applications where orderless methods are a popular choice for representing textual data. The bag-of-words model describes text documents as frequency distributions over words and has

been applied extensively in this domain." (28) Its processes can be further defined as follows: Given a training dataset containing images of a domain, an image represented based on interest regions is detected and marked out. Visual features are extracted from these regions using a specific unsupervised learning algorithm, such as k-means. These features get tokenized as keypoint-based features, with the scale-invariant feature transform as one example. K-means is used to group these features based on a fixed number of visual words (or categories) using a cluster number within the generated visual-word vocabulary (or codebook). The vector of an image contains the presence or absence of information of **each** visual word in the image. For example, the number of keypoints in the corresponding clusters is the visual word. A co-occurrence table finally summarizes the data in a co-occurrence table of counts.

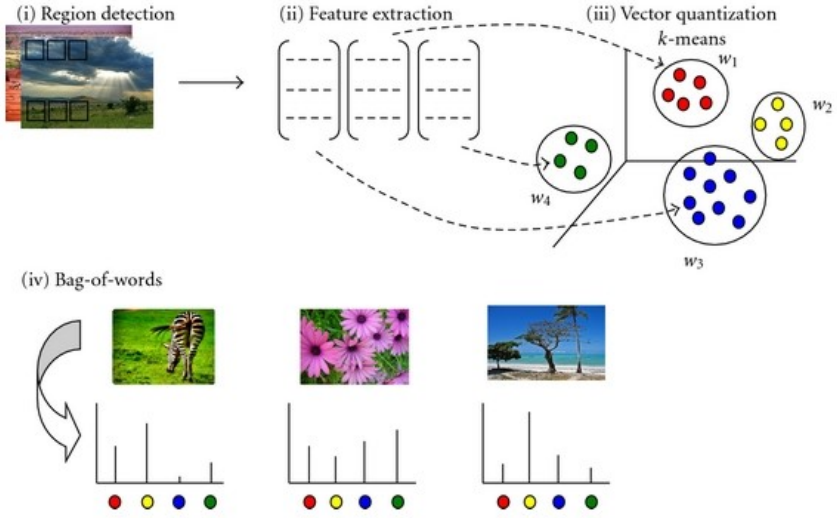

Figure 2.10 To extract the BoW feature from images involves the following steps: (i) automatically detect regions/points of interest, (ii) compute local descriptors over those regions/points, (iii) quantize the descriptors into words to form the visual vocabulary, and (iv) find the occurrences in the image of each specific word in the vocabulary for constructing the BoW feature (or a histogram of word frequencies

Noted challenges with the bag of words model include the fact that while the model is quite effective for object and scene recognition, it can be improved upon by a number of techniques: a. modeling the co-occurrence of visual words (creating the latent *Dirichlet allocation* model). b. This model can be extended to describe the relative positions of different parts of the object (creating a constellation model) and c. Extended again to describe the relative position of objects in the scene (creating a scene model). The goal when matching local features is to find those

descriptors from any previously seen model (exemplar) that are near in the feature space to those local features in a novel image. Since each exemplar image may easily contain on the order of hundreds to thousands of interest points, the database of descriptors quickly becomes very large; consequently making searching for matches impractical. The database must be *mapped* to data structures for efficient similarity search. The very constituent nature of this data structure that it must be mapped to is of important note to us as it seems to also hint to us what may be useful bridge between the raw image and feature descriptors as well as the higher level description for that image. In this study, we had proposed the notion of error pairs and reversed search as one possible strategy that can be used to effectively reduce or limit the number of features that can be searched in the database. This is further discussed in the next chapter on *Experiments*.

Despite these drawbacks however, the bag of words model is still being used for a wide array of applications ranging from annotation and image-retrieval; object, event and action recognition, among others. (Sivic and Zisserman, 2003) further applied the image feature representation method for document representation, in the area of image and video retrieval, with promising results. (Aly et al., 2011)[8] decided to use the Bag of Words model for large scale object recognition using a massive 11,000 games cover dataset. They explain their objective: "In this setting, the goal is to find the matching image in the collection given a probe image containing the same object. In this work we explore the different possible parameters of the bag of words (BoW) approach in terms of their recognition performance and computational cost. We make the following contributions: 1) we provide a comprehensive benchmark of the two leading methods for BoW: inverted file and min-hash; and 2) we explore the effect of the different parameters on their recognition performance and run time, using four diverse real world datasets." (1). In this study, we also adopt some of the experimental objectives of Aly et al. with the aim of finding a matching image from a collection using a probe image using an inverted search approach.

However, in this regard, our images are not exclusive to games data image, since we are particular about finding matches from a diverse set as well. Image datasets are usually the main sources of semantic knowledge in the bag of words model. They contain a meaningful semantic

[8] We also did test Aly's Matlab-based tool and sadly found the use of linux g++ files for certain tasks alienating for us Windows folks. That can be annoying since it limits us from using his project as his is the ONLY vision project seriously focused on games content.

representation of types of the visual inputs typical of a particular domain. For this study we used a lot of games-related image content, hence, our bag of words derived from the training set will likely signal the key semantic concepts encountered in this domain. They are offered under the form of a collection *G* images, with *m* labels:

$$G = \{m_i | \, i = 1, 2 \ldots k\}$$

The notion behind this type of representation suggests that visual words are also expected to encode the semantics of the images patterning to a domain. The quality of the visual words (and their predictive power) would certainly be enhanced if they are constructed with features from the respective objects or scene classes (more training data equals better recognition, right?). But if the learning model only uses images from its own domain, the predictive power is made weaker, because it would hardly generalize to new contexts. Or in our case, comparable images from an equivalent games domain will likely learn visual patterns unique to the genre. Choosing only features from the same domain will likely eliminate the background noises that also generate extraneous features in the mix, or with our example, features belonging to other objects or other domains. Selecting the right mixture of images certainly increases the relevant/noise feature ratio. Consequently, the resulting visual words are more accurate descriptions of the objects denoted by the labels or domain type. On the contrary, in this study, we perform both types of experiments by proposing to construct a dedicated visual vocabulary for domains that include not only games covers; where $m_i \in G$ contains only features extracted from images peculiar to games, m_i—but, we also include sets that are not part of the games domain, and as such, those sets become our hard negatives to further strengthen the predictive power of our model. Recognizing different types of character actions on the game covers using semantic triples would have been an enriching research experience; however, we are severely constrained to basic image-retrieval for the discovery of unique features from our images that could potentially map to scene classes.

(Klaser, 2010) had used a bag of words model for action classification using a random sampling approach. He describes it as a bag-of-features representation for a video sequence which contains a loose representation of a set of local space-time features obtained in a sparse set of spatio-temporal interest points. These points are gotten by applying the space-time extension

of the Harris operator, (Laptev, 2005). Klaser's work had used the bag of words model for human action recognition.

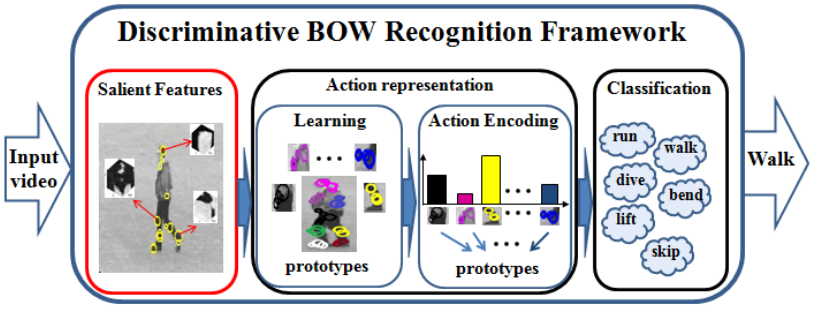

Figure 2.11 Bag of words for action recognition

Faced with a massive amount of video data, Klaser (2010) was still able to retrieve the most relevant features from a very large pool, by repeatedly applying a clustering algorithm to the extracted features to reduce the margin of errors. In further developing his visual vocabulary, he explains he had to:

> …apply either random sampling or k-means on the set of training features. Random sampling has the advantage that it is very fast since only a subset of V random training features needs to be computed. For results using k-means, we cluster a subset of 100,000 randomly selected training features in order to limit computational complexity. We increase precision by initializing k-means 8 times and keeping the result with the lowest error. Features are assigned to their closest vocabulary word using Euclidean distance. The resulting histograms of visual word occurrences are used as video sequence representations. (Appendix A1).

Although Klaser does not mention semantic triples, we can still grasp that his resulting histograms were all targeted towards the identification of a human *subject*, an *object* (which can be another human or an actual object), and the connecting verb or action that defines their relationship. On the whole, the bag of words model has found varied uses for a range of diverse tasks and has become an important link in bridging the semantic gap between the low-level features and the high-level concepts surrounding those features. It is ubiquitous in so many applications: from general object recognition and scene classification, to human action recognition, image segmentation, query expansion, and similarity measures, not forgetting the annotation of large scale image databases, just to name a few.

Chapter 3 Experiment

3.1 Our Scope

In this study, we develop and test the accuracy of a simple image-retrieval tool which in turn uses libraries from the OpenCV framework. Due to constraints (time and resources), we have limited our experiment to the comparison of various image descriptors based on their ability to accurately describe features and retrieve images from games cover content. We group the image features using k-means clustering to form our bag of words and then proceed to measure the relative distances of these features using different distance functions. We utilize an inverted search process to retrieve similar or equivalent images as a way of ascertaining how precisely our descriptors were able to uniquely differentiate one image from another based on the detected image features. We also measure the relative distances of these features from one another as a way of defining image similarity and possibly grouping a family of visually-related images. The accuracy of a relevant image descriptor is important because so much of an algorithm's ability to recognize objects, events or actions depends on this preceding task. If we had trained our algorithm to recognize certain types of faces or objects commonly found in games, we would want our image-retrieval tool to accurately match the correct object when performing facial or object detection on the images. However, since we used a global-feature approach for the automated analysis of our images, by means of a descriptor's mapping out of the whole image as a single field for detection and description (for this image-retrieval task, we did not mark out specific regions of interest in the image for objects to enable individual character or object recognition). Our objective here is not to recognize individual objects or define specific scene classes or scene actions for this thesis, even though our algorithm does make a good effort in detecting and describing image regions that have particularly interesting features, easily creating the possibility of mapping out game characters and objects within the same images. We instead treat each image as a unique semantic unit or its own scene class. We want to understand how unique they are based on the visual features. The significance being that if we could identify what makes the image unique based on its features alone, the same low-level features can easily be later mapped to any type of metadata describing the same image (be it a text, a link etc.,).

As previously stated, we are simply interested in observing how the scenes in our images structure as a single semantic unit, where they are uniquely described as *distinctly* from one another as possible, based on their low-level features. Incidentally, our work almost resembles the automated large scale discovery of image families, (Aly et al. 2009), in that we also used a small portion of their games image dataset with some of their objectives. However, unlike Aly et al., our objective is not the discovery of large image families, but an inquiry into how one image is uniquely described from another. This is not to say finding image similarities the way Aly et al. had done does not have its merits. In fact we were inspired by (Aly et. al's 2009) work as a starting point by adopting the notion of comparing two approaches for measuring image similarity and assessing their performance on datasets:

> This work focuses on the problem of automatically identifying image families in unprocessed image collections. We compare two broad approaches for measuring similarity between images: global descriptors vs. a set of local descriptors. The global approach represents each image by one feature descriptor computed from the whole image. The local approach represents each image by a set of local feature descriptors computed at some interesting points in the image…1) We use the term family to indicate groups of images having high visual similarity with possible change in color, view-point, scale ... etc. (1)

Once more, unlike Aly et al. we hinge on a global-based image *distinction* as our focus because we are interested in further building the premise of having data-driven visual links based on natural images that have to be as different from a similarly-looking image as possible. The opposite of family clusters is lone distinction, but can both concepts help each other? We hope to also explore this possibility with our experiments. Certainly, this process is also important for image-retrieval, and so we will be sensitive to errors. A possible near future experiment will be to use the same images to annotate the types of scenes or actions each games cover depicts, train them accordingly and then apply learning algorithms to discover similar scenes or events in unseen images. In this particular scenario, we can tolerate having "a family of images" that share the same visual characteristics, thereby adopting Aly et al.'s approach. But we will be doing this with the key difference being that we are not merely clustering images into families based on the simplistic concepts of colour or texture. When considering an event or action recognition in image contexts, colours and texture do play a small part at the lower levels. However, a lot also

depends on how objects are recognized in addition to the structural relationship each individual object has in the image including how they connect to a more abstract theme (as in, how thematically different is *character holding a knife* from *character holding a ball*—why?). A future study into the image families of games and other natural scenes will concentrate on these types of Event Triples using a graphical representation. The same concept can be applied to sequences of video frame images, not just static images, enabling people to search for scenes within movies and games using a high-level of abstraction (as in, perform a search for all scenes where "character gazes at stars" or "character reads the Bible"). Connecting the semantic triple in a textual search to an equivalent visual component, where image features have also been clustered as semantic triples in high-dimensional vectors will be an exciting area of study, but is beyond the scope of this study. We will limit our focus to basic retrieval for now.

3.2 Application Description

We mentioned earlier that a simple image-retrieval tool was used for this study and will be the framework by which we train, test and evaluate our image data based on different descriptor settings. We track for errors in the retrieval results based on these settings. It will also be the basis by which we aim to understand some of the arcane concepts in the visual recognition field. The screenshots below give a clearer description of our tool, showing the images from our vocabulary on the left and the query image alongside image statistics on the right. To use the tool, we load a number of images into the application constituting our training set. For this project we used three images, most of which come from games content and a few from non-games content. Based on the settings chosen, unique feature descriptors are extracted from this set. There are different routes to take to determine how these features are clustered (k-means among others being a common strategy). There are also different settings for measuring distances between extracted features using the nearest neighbour concept. There are two strategies to choose when inputting our test image for comparison or retrieval. The application has two methods for adding test images: i. by using a webcam to capture and detect objects from a live stream (hence, the camera panel in the middle of the application). ii. Or by adding an image file or a list of files from a directory of static images. These images get displayed within the camera viewer panel, and can be similarly "played" as sequences of images in the panel.

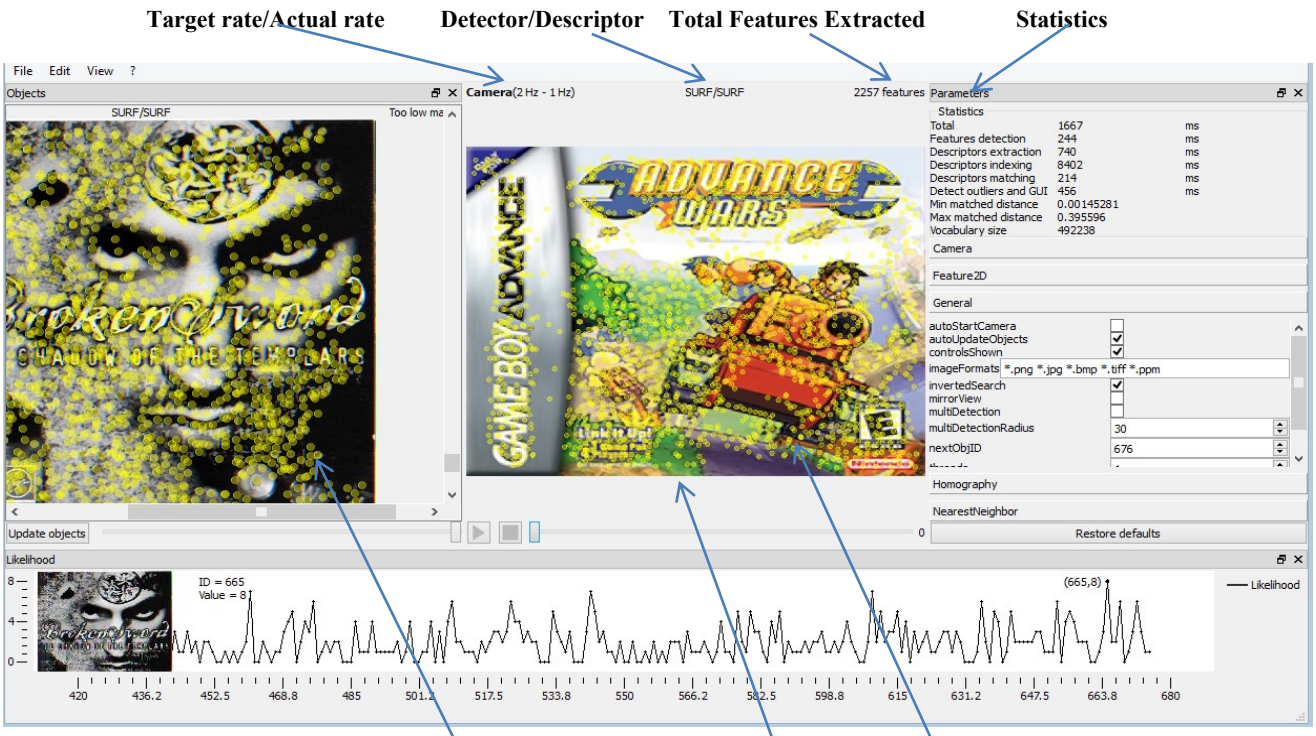

Figure 3.1 Image-retrieval App. Possible matching image

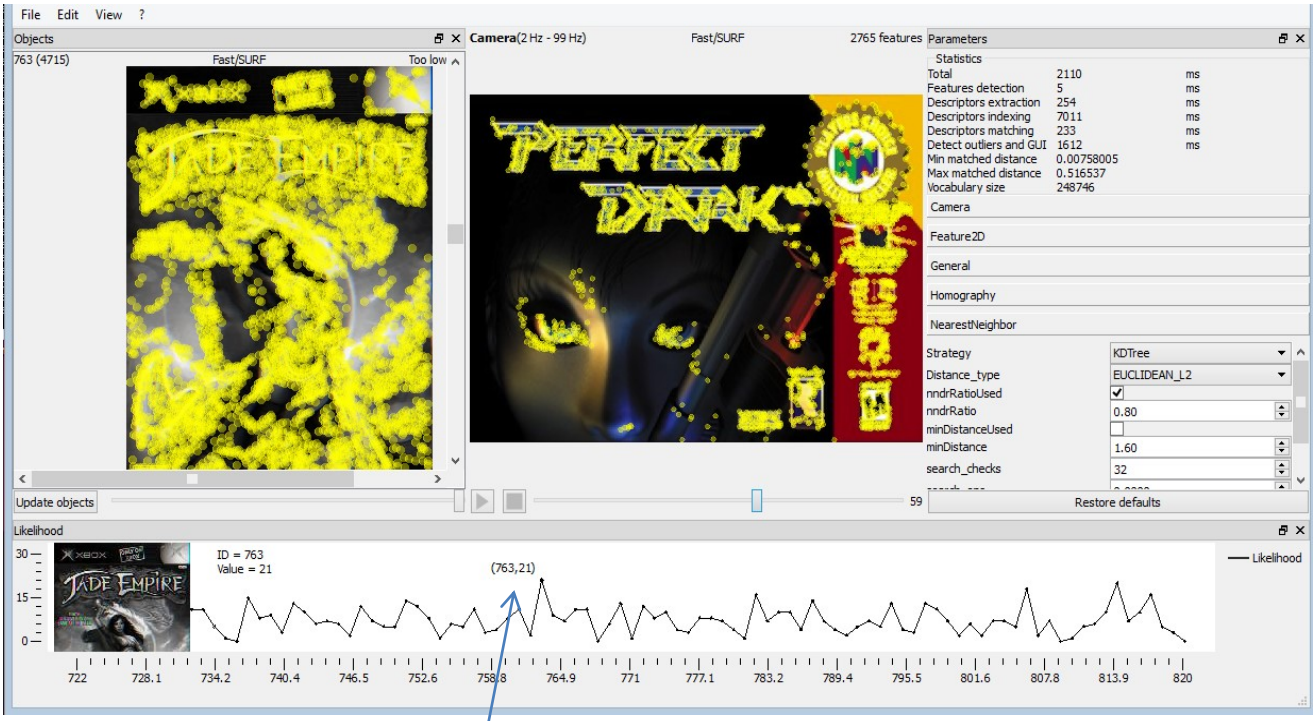

Figure 3.2 The *likelihood* panel showing a graph of the inverted search results. Each point on the graph corresponds to an image's proximity to the query image with the highest match peaking the most in the graph (in this case with 763 objects). On the likelihood plot, you can mouse over the dots to show the related image. The image in the graph is actually an error-match.

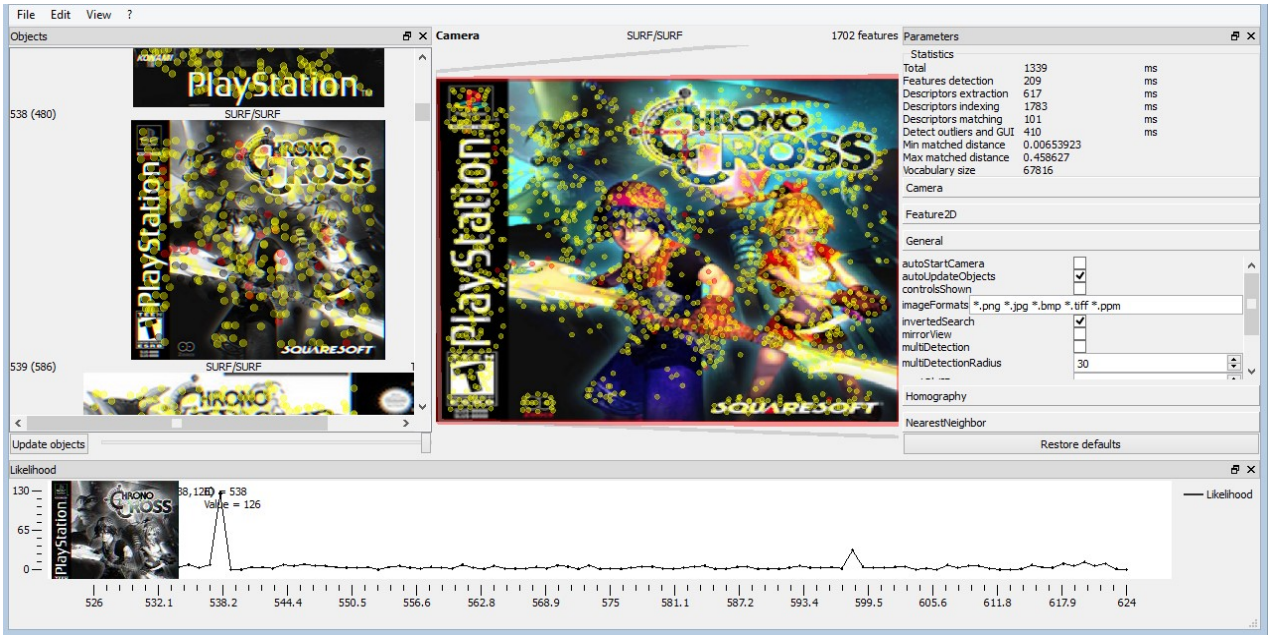

Figure 3.3 showing *matching* image between Query image and highest result, from the inverted search. You can change any parameters at runtime, making it easier to test feature detectors and descriptors without always recompiling.

In this study we chose as a second option of loading our test images containing a hundred games cover content and executing an inverted search for each image in the test set against our earlier features extracted from our dataset (the 100 test images are also part of the 300 earlier loaded). It is sadly a small sample and is not an actual true test for the accuracy of our image-retrieval tool, but in order to train the 11,000 games content images from Aly et al. that we already had for this study will cost us a lot more valuable time which we did not have. Thus, each of the 100 games images in our test set also has features extracted and compared against our trained images using a distance function. Whenever we observe a particular image on the camera panel, we observe how its described features compares against a candidate match to the right. Based on the settings as well as the descriptor chosen to process the image, the number of features detected, described and extracted, not forgetting the vocabulary size for each image will vary greatly even for the same image. These are all shown on the statistics panel with a sample shown below. The statistics panel displays how many features were detected and descriptors described, indexed and matched showing how fast those tasks were performed in milliseconds (*ms*). The panel also reveals the distances between the indexed features and the total vocabulary for both the trained and probe images.

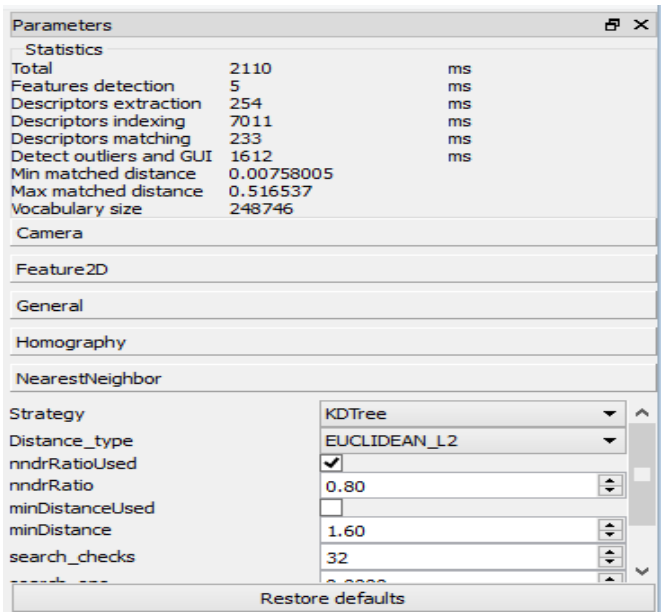

Figure 3.4 The Statistics panel

3.3 Basic Procedure

The following is a basic description on how to use the tool. We had acknowledged two methods in loading and testing images in the application. We can test using webcam mode or using an inverted search mode. To find objects using a webcam, here are the steps:

1. Go "Edit" -> "Add object...",
2. Present an object,
3. Select the features extracted from the object, return to main screen,
4. Play ("Edit" -> "Start") and
5. See highlighted features corresponding to the object.

As indicated earlier, you can change the parameters at run time, thus making it easier to test feature detectors and descriptors without recompiling. The tool's detectors and descriptors use the same libraries found in OpenCV. Here are a few supported: BRIEF, Dense, FAST, GoodFeaturesToTrack, MSER, ORB, SIFT, STAR, and SURF. We briefly encountered the concept SIFT in the previous chapter, but there are other competing descriptors claiming various advantages. We will use SIFT as our baseline and compare its performance against other selected descriptors.

The other way to load training/and testing data in our tool is by using the inverted search mode. Here is a basic description of its procedure:

1. Download your training and evaluation dataset
2. Open the application and reset all settings to default (menu "Edit->Restore all default settings").
3. Open Parameters panel (menu "View->Parameters") and go to "Detector_Descriptor" section. From there scroll down to SURF parameters. Uncheck "SURF_extended" parameter and set "SURF_hessianThreshold" to 150.
4. In the "General" section of the parameters, check "controlsShown" and "invertedSearch".
5. Open the likelihood panel (menu "View->Likelihood"). When detecting, this will show the likelihood score of the scene with all the objects.
6. Load objects from the training dataset previously downloaded by using the menu "Edit->Add objects from files...." Select all the files in the directory of the training dataset. This may take a while until all images are processed (extracting features and creating the vocabulary).
7. You can resize the objects size by moving the scroll bar next to "Update objects".
8. Setup the camera to use a directory of images by using menu "Edit->Camera from directory of images...", and then select the evaluation dataset.
9. Press "Space" or action "Play" and observe what is going on. This would look like the image panel in the figures above. Note that the "Objects" panel scrolls automatically to object with the highest likelihood score. A rectangle is shown if the homography can be computed.
10. On the likelihood plot, you can mouse over the dots to show the related image.

With regards to data collection and training with our application, the process is similar to the outline in the figure below. The figure helps to capture the sequences of actions for the entire experimental process from a learning model viewpoint.

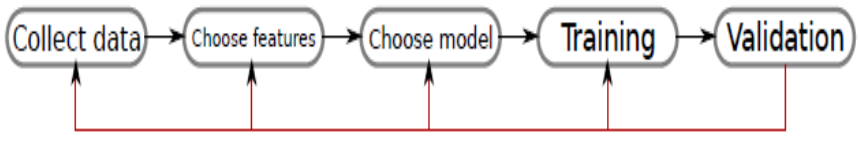

Figure 3.5 A learning model life-cycle

Interestingly, the process of actually collecting images is getting easier at the moment with the advent of image sharing websites. But then again, according to Aly et al. (2009) such collections

contain duplicates and highly similar images or what they referred to as *image families*. Our application can handle tens of thousands of unprocessed images, but time-constraints forces us to use just a few. Still for a second set of experiments we used another tool to calculate the distances of a probe image against the rest of the images in the games dataset. For this second tool we also used a pre-computed visual codebook or dictionary containing 10, 000 visual words unrelated to games. Aly et al. (2009) explains how the automatic discovery and cataloguing of similar images in large collections is important for many applications, e.g. image search, image collection visualization, and research purposes among others. In their work they get to assess their performance as the image collection scales up to over 11,000 images with over 6,300 families presenting an algorithm to automatically determine the number of families in the collection. As specified earlier, we only used 300 of the 11,000 images to analyze the opposite of image families—image uniqueness.

3.4 Image Matching

Our goal is the implementation of basic image-retrieval where we need to match the exact probe image with a similar one found in the cluttered image database using only low-level features. In order to do this we need to choose an object descriptor that allows for a highly distinctive description. As Sift is our baseline, it has been described by Lowe (2004):

> The keypoint descriptors are highly distinctive, which allows a single feature to find its correct match with good probability in a large database of features. However, in a cluttered image, many features will not have any correct match in the database, giving rise to many false matches in addition to the correct ones. The correct matches can be filtered from the full set of matches by identifying subsets of keypoints that agree on the object and its location, scale, and orientation in the new image. The probability that several features will agree on these parameters by chance is much lower than the probability that any individual feature match will be in error. The determination of these consistent clusters can be done rapidly by using an efficient hash table implementation of the generalized Hough transform. (2)

The best candidate match for each keypoint is resolved by detecting its nearest neighbour in the database of keypoints from training images. The nearest neighbour is defined as the

keypoint with the minimum Euclidean distance for the invariant descriptor vector. Other types of distance functions can also be used depending on strategy. The basis for image matching uses these types of distance metrics, to evaluate key-point features relative to a point. Image matching is actually the appraisal of various distance functions applied on high-dimensional features. It is one of the fundamental tasks in computer vision since a good set of correspondences between images is vital in order to carry out certain tasks down the processing pipeline. But, numerous features from an image will have no reliable match in the training database because they arise from background clutter or have ambiguous matches. For that reason it would be useful to discover a way to measure the constancy of each individual feature match. This cannot be done based just on individual feature distance, as with some descriptors. This is why some techniques utilize additional learning models to find these, taking into thought other types of distance functions like the second closest nearest neighbour. We consider the second-closest match as an estimate of the density of false matches within this portion of the feature space and at the same time detecting specific instances of ambiguous features. In other words, can an image's uniqueness in a clutter of images be discovered based on the types of errors or mismatches it produces? (Errors can also indicate a pattern that can be learnable too). Unfortunately, there are no efficient algorithms to ascertain the exact nearest neighbour of a point in high dimensional spaces. Our keypoint descriptor has a 128-dimensional feature vector, and the best algorithms, such as the k-d tree, Friedman et al. (1977), provides almost no speedup over exhaustive search for such a high number of dimensions, resulting in a lot of approximations to enable matches.

Evidently, precision is important in image-retrieval applications or tasks that seek to apply high-level classes or categories to the images. In this work we want to see if a probe image will find itself among the background clutter. As in, we are interested in retrieving both the exact image as well as images similar to a query image that constitutes our closest neighbours. The image's distinctive features as identified and constructed by the object detectors and descriptors should maintain the concept of repeatability when finding a match based on these features. We analyze the matches obtained for different techniques and evaluate their performance for different evaluation metrics. The objective at this point is to find the best matching strategy amongst all the different configurations. Given a query image, we use several matching strategies developed in this inquiry to retrieve an image from the database which is most similar to the query.

Figure 3.6 Mismatches in Retrieval results reveal the concept of Error Pairs

The figures above depict different image-retrieval results from the application we had described earlier. Most of the images are actually error pairs, and not actual matches. We explained in the previous chapter that we will be sensitive to errors in this study. The images in the figures illustrate the end result of an inverted search process after matching image descriptors. The results can either show *precise* or *error* matches. Usually, the image that provides the maximum number of corresponding visual word matches with the query image is regarded as a match even though they might be so semantically dissimilar. The objective of performing such a retrieval application is to find images which correspond to the intersection points or are closest to the intersection points between different games scene sequences. The matches obtained are refined using the Random Sample Consensus (RANSAC) method before deciding on the closest image. We forward the argument in this thesis that while we desire consistent matches to mark image distinction from a clutter, consistent errors emitted from image-retrieval from a particular *probe* is indicative of a type of distinction marker for the probe image as well. Objects in our probe image are matched against those in a vocabulary created with descriptors. Our vocabulary was built mostly from game content. The colourful spots on the screenshot images are actually a visualization of the visual words from our vocabulary. In performing this task, we utilize an inverted search. As indicated earlier, inverted matching (or likelihood computation) as defined by Sivic and Zisserman (2003) are feature reference files structured like an ideal book index.

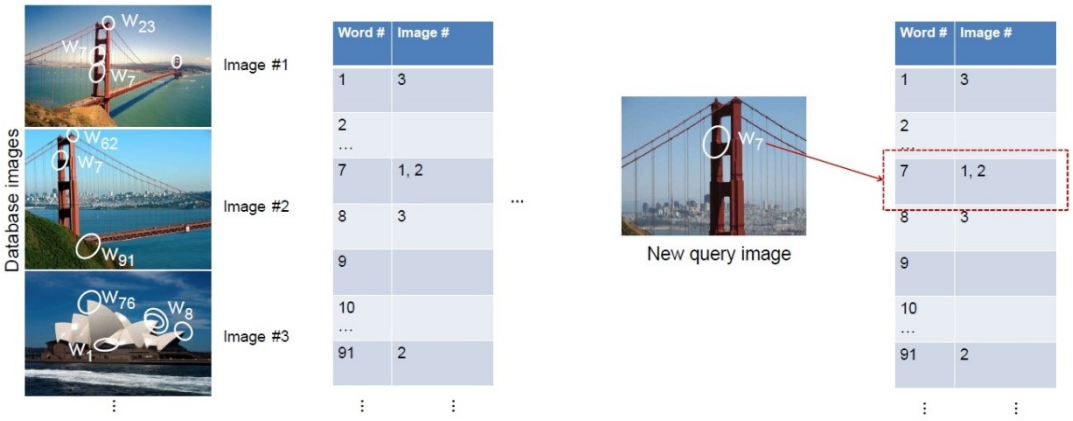

Figure 3.7 a. All database images are loaded into the index mapping words to image numbers. **b**. new query image is mapped to indices of database images that share a word.

But instead of actual textual words, we have visual words in its place that has been derived from the respective images' object descriptors and their indexes.

3.5 Distance Metrics

Here are a few distance metrics used in the image-retrieval experiment in this study. Depending on the objective and strategy, different distance metrics are used to discover features in high-dimensional space. Each type of applied metric elicits its own strength and weaknesses with the consequence that the overall performance of the bag of words model usually depends on the metric strategy employed in discovering hidden features. A distance metric learning may discover a 'true' similarity function that respects a set of constraints. Given a set of pairwise constraints, which must-link constraints ***M*** and cannot-link constraints ***C***, the task is to find a distance metric ***D*** that minimizes the total distance between must-linked pairs.

$$\sum_{(x,y)\in M} D(x,y)$$

And maximizes total distance between cannot-linked pairs:

$$\sum_{(x,y)\in C} D(x,y)$$

We use the Euclidean *l2* distance type for the first type of must-link pairs:

$$d\,(x,\,y) = \sqrt{\sum (x_i - y_i)^2})$$

However we also note that the *l1* metric is quite often used in computer vision for feature distance calculation.

$$d_1(\mathrm{p},\mathrm{q}) = \|\,\mathrm{p}-\mathrm{q}\,\|_1 = \sum_{i=1} |\; p_i - q_i \;|,$$

We had also used a cosine similarity function as a type of similarity metric between the images. The cosine measure works as follows:

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$

Where the scalar product is defined as,

$$x \circ y = \Sigma\, xi * yi$$

And the length of vectors is the following:

$$|| \text{x} ||= \sqrt{\sum x_i^2} ; || \, y \, ||= \sqrt{\sum y_i^2}$$

When measuring for similarity, two points *a* and *b* are similar if the distances between their descriptors remains below an arbitrary threshold: $d_M(D_a - D_b) < t.$ The figure below shows the values of the distance measures of a probe image compared against other games covers in the dataset. The cosine similarity measure was used to compute the relative distances of the images to each other based on their extracted visual words. The figure below is just a truncated sample of the overall results.

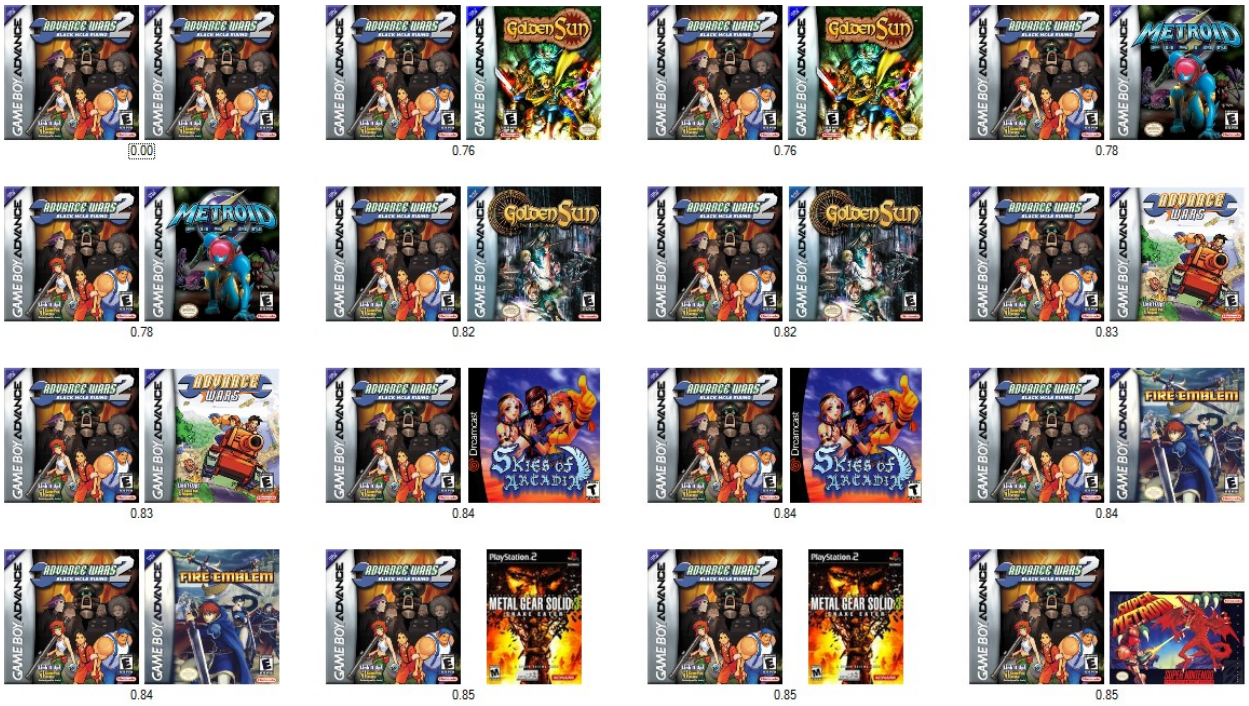

Figure 3.8 In the figure above, we take a single query image "Advanced Wars 2" measure its similarity against every other image in the dataset. Due to space constraints we only show a small sample of the results.

When comparing the descriptors of the several images in the figure above we use the cosine normalized difference, which ranges between 0 (identical) and 1 (completely different). For this experiment we also used a pre-computed dictionary containing ten thousand pre-computed visual words. After obtaining initial candidate image matches, either from an IF (inverted file) or MH

(min-hashing), we may well also opt to re-ranking these images by using the consistency of corresponding features.

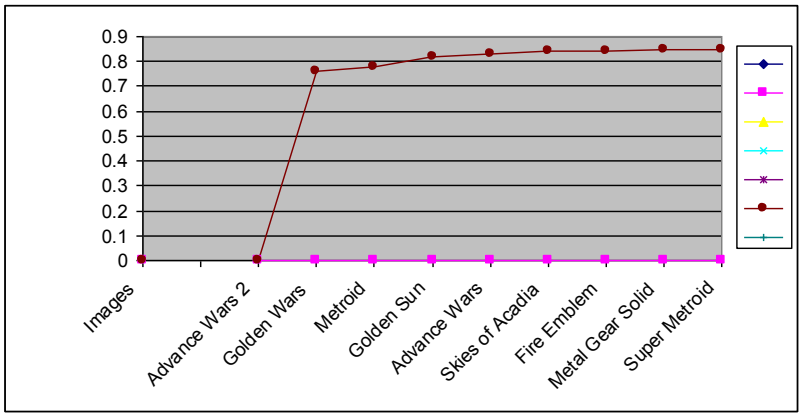

Table 3.1 Small sample showing the distances from *Advanced Wars 2* as probe based on the figure 3.8.

We could also visualize the corresponding visual words or descriptors on the probe and their equivalent images, including the value of their cosine distances.

Figure 3.9 Visual words on images and relative cosine distances

As stated before, we could use the cosine measures as a means to not only understand visual thresholds but also as a basis for grouping the images into various image families based on visual description. We had actually used feature matching crudely using the visual words, where

features that have the same visual word are simply considered matched. Other more advanced techniques can be used, for example Jégou et al., (2008), but we had used the simplest approach here. These *feature matches* are then used to fit an affine transformation between the probe image and the candidate images using RANSAC, Forsyth and Ponce (2002). Basically, affine transformations are a **relation** between two images, based on their rotations, translations and scale. It basically means that we could take an image, make a copy of it, and try to invert the copy, reduce its size or rotate it along an axis, but an ideal object descriptor will still be able to notice the visual similarities between the images.

Nevertheless, we are not interested in merely clustering the images based on visual appearances alone as hinted earlier. Our semantic objective is a lot more sophisticated than that. We understand that using the similarity measure defined above, a 0.0 score equates an identical image. This precise match prompts us to take a closer look at its unique visual descriptor. What makes the image's features exactly distinctive from another in the dataset? If these visual patterns are exactly unique to that image, could these features be easily mapped to a high-level concept representative of the core image scene, equating to a descriptive scene class? It is a way of inquiring into the possibility of having low-level features peculiarly identify an image scene and directly link same to any additional meta-data or semantic concepts surrounding that image.

3.6 Error Pairs

While we are concerned about precision, we are also sensitive about errors or image mismatches our application generated during the various image-retrieval tasks. The whole objective of any learning model is to minimize errors in the discovery and analysis of features. The types of errors that results, is usually dependent on the binary descriptor as well as settings that was applied in a retrieval task. We compare the types of errors each descriptor produces when tested with our game's cover content. We selected a few descriptors from an array currently used in the OpenCV framework and integrate these libraries into our application. The following is a very broad overview of the other object descriptors that were tested (we omit Sift because we have discussed it earlier as our baseline): i. BRISK: Binary Robust Invariant Scalable Keypoints-- is a novel method for keypoint detection, description and matching. The key to speed lies in the application of a creative scale-space FAST-based detector in combination with the assembly of a bit-string descriptor from intensity comparisons retrieved by dedicated sampling of each

keypoint neighbourhood, Leutenegger(2011). ii. SURF: Speeded Up Robust Features-- is a robust local feature detector, first presented by Bay et al. (2006). As the name suggests, it is a speeded-up version of SIFT taking its inspiration from it. If Lowe's SIFT approximated the Laplacian of Gaussian (LoG) with Difference of Gaussian for finding scale-space, SURF goes a little further and approximates LoG with Box Filter. Essentially, for purposes of clarity, the Laplacian of an image highpoints regions of rapid intensity change, and is therefore often used for edge detection. The same Laplacian is often applied to an image that has first been approximated with a Gaussian smoothing filter in order to reduce its sensitivity to noise, and hence the term LoG. SURF leverages a common image analysis approach for regions-of-interest detection that is called blob detection. The typical approach for blob detection is a difference of Gaussians. There are several reasons for this, the first one being to mimic what happens in the visual cortex of the human brains. The drawback to difference of Gaussians (DoG) is the computation time that is too expensive to be applied to large image areas. In order to bypass this issue, SURF takes a simple approach. A DoG is simply the computation of two Gaussian averages (or a Gaussian blur) followed by taking their difference. An approximation is to estimate the Gaussian blur by a *box blur*. A box blur is the average value of all the image values in a given rectangle which can be computed efficiently via *integral images*. SURF also uses a wavelet response for orientation assignment and feature description. This makes Surf suitable for vision tasks like object recognition or 3D reconstruction. iii. ORB: Oriented FAST and Rotated BRIEF[9]—it is a very fast binary descriptor based on another descriptor called BRIEF, but has the advantage of being sensitive to image orientation, Rublee et al. (2011). ORB is mostly the merging of the FAST[10] keypoint detector and the BRIEF descriptor with many modifications to enhance performance. While it uses FAST to find keypoints we understand that FAST does not compute orientation. So what about rotation invariance? The developers simply modified the FAST and BRIEF fusion to accommodate such invariances. Orb similarly applies the Harris corner measure to find top N points among them and uses a pyramid to produce multiscale-features. The following tables show how the various object descriptors performed when compared against Sift using the *l1* and *l2* distance measures for the features. The tables indicate

[9] BRIEF: Binary Robust Independent Elementary Features
[10] FAST: Features from Accelerated Segment Test

errors from the various descriptors in failing to match a query image from our 300 games dataset sample.

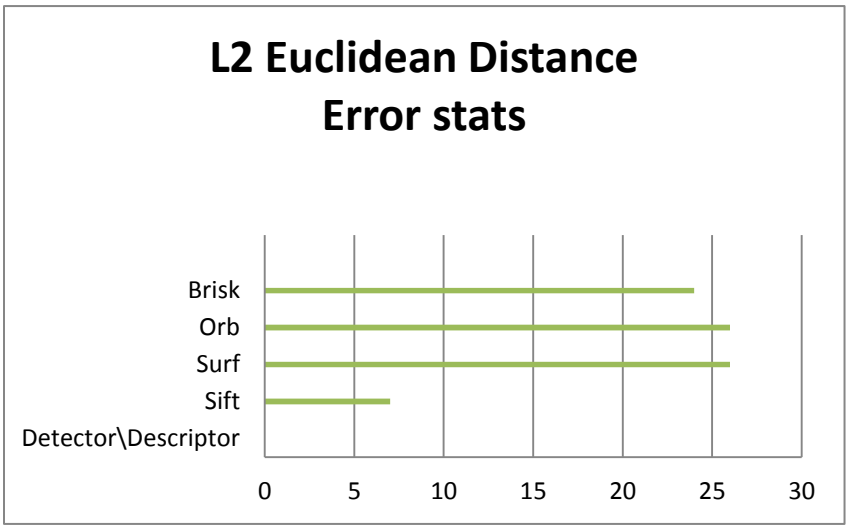

Table 3.2 Detector/Descriptor errors using l2 distance

What the table shows is that Sift performs better from our small sample set when compared against the other descriptor techniques, especially when using the Euclidean distance measure for the image features. However, its performance suffers considerably when the l1 distance is used.

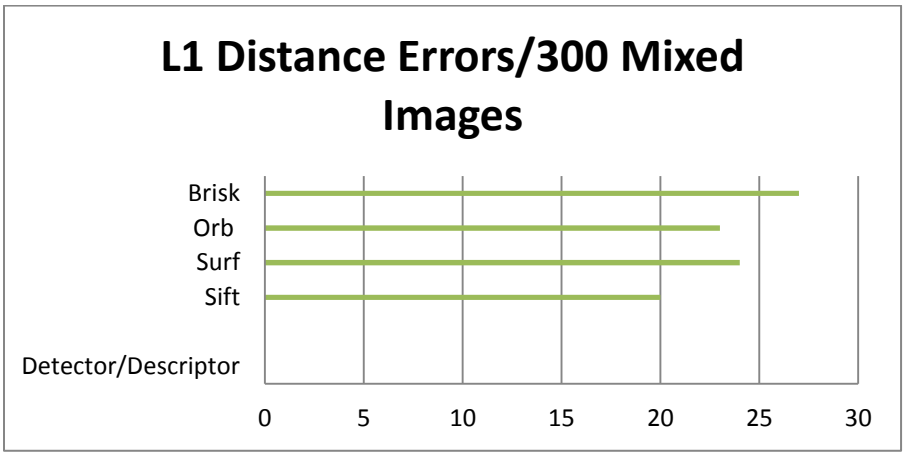

Table 3.3 Sift performs poorly here with *l1* mostly because of background clutter.

We also decided to test our object descriptors using a smaller dataset –our original 100 games image sample without mixing them with additional image content from domains completely outside of games.

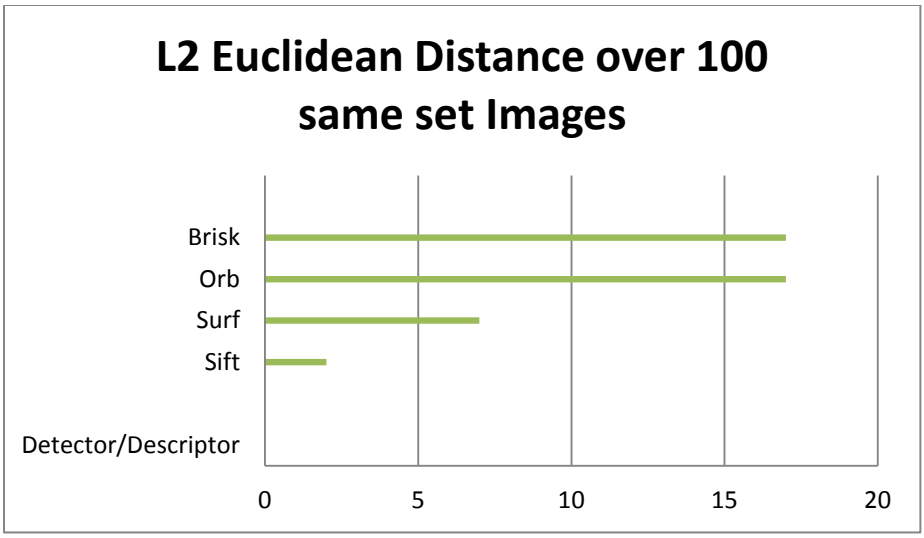

Table 3.4 The relative proportion of errors remain the same with Surf improving

Curiously, we discover that the pattern of errors relatively held with Sift performing a lot better in our tests. However, Sift had no errors at all when the l1 distance measure was used again for this smaller sample.

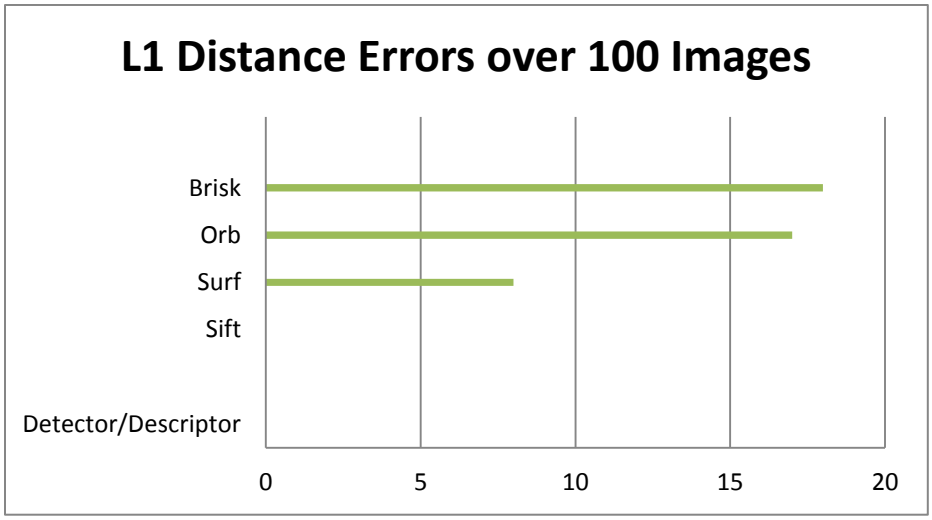

Table 3.5 Sift had no errors here for *l1*, 100 images

When we used our baseline descriptor on 300 games cover set under the l2 Euclidean distance, we got 8 errors. When we mixed the images with non –games images during training under the l1 distance, the errors degraded to over 20. When we made the training sample size to just 100 games cover set, under the l2 distance, the errors shrunk to just three. When we used the same set under the l1 distance, our baseline Sift recorded no errors at all. The other descriptors showed far worst performance under each parameter and types of data set when compared to Sift.

Evidently this could mean that Sift is merely good at verifying an image match when no other competing pattern from a background clutter is there to confuse it. This smaller sample will hardly constitute a true learning model for our descriptors since errors tend to be artificially minimized especially when we have a smaller sample set and when there are no other tough choices to make, regarding the analysis of the image's features. Incidentally we are simply stating that image-retrieval errors are dependent on the types of descriptors and learning models used. But when we find a *consistent* pattern of the types of errors or image mis-matches *across* different object descriptors for a *particular* query image, it does tell us something. To put it another way, if we used the Brisk, Surf, Orb and Sift descriptors in different instances to invert search our image database and the results consistently produce the same type of errors or image mis-match pair (the second closest nearest neighbours for that image), based on the concept of descriptor repeatability, it is also suggesting to us that a particular image can also be identified by its mis-match patterns. If Sift or Orb descriptors produce a feature mismatch, and we are able to observe 10 other similar images or errors in the likelihood panel, the 10 images in that neighbourhood can become error-patterns for our probe. In other words, while the second nearest neighbour approach can be used to find images nearest to the probe image, or sample the density of errors in that neighbourhood, we soon discover that for each probe image, every consistent error resulting from the probe is also a distinctive marker indicative of the probe image as well. This essentially means that while a perfect and precise match for a query image shows an example of image distinctiveness, the types of consistent errors an image produces when there is a mismatch, shows a type of distinctiveness too, that could be traced back to the probe image. *By their errors, we shall know them*. The only drawback to this hypothesis being that if we suddenly mixed our original 100 image dataset to a larger and newer dataset, the pattern of errors suddenly changes for our probe images because of the new background clutter. But then again even the new dataset will likely create a new set of consistent error patterns for each query image to enable us mark out that image based on the errors or nearest-neighbour cluster they had elicited in the likelihood graph and panel. It should be noted that the images in the graph of our likelihood panel are all images in the neighbourhood of our query image, each having a relatively close distance from the probe. Not only can they be analyzed into distance-based family clusters, with additional semantic descriptions to highlight them, but in the context of our study, they are all hypothetically error-candidates for the query image no matter how many similar object

features they share. Image-retrieval errors in our context are a new type of meta-data by themselves, designed to recognize the target image.

3.7 Local Descriptor Metrics

The tables in the previous section essentially show the individual performances of the various local descriptors used in this study. Because our sample size was really small and we lacked the time to proceed with further tests, we did not perform further evaluation of these descriptors. We had instead focused on the analysis of error-pairs since they hinted at a new semantic knowledge that has previously been untouched. However, in discussing the usual metrics for accuracy for these types of experiments we henceforth list the various descriptor metrics starting with the general *Descriptor Success* rate based on repeatability:

$$Descriptor\ Success\ Rate\ ,\ s\ =\ \frac{Correctly\ Matched\ Interest\ Points}{Repeated\ Interest\ Points}$$

The more common evaluation metrics includes a matching score for the image-retrieval task:

$$\text{matching score} = \frac{\text{number of correct matches}}{\text{number of detected regions}}$$

The values for the matching score can all be lifted from our application's statistics panel. However we also use our error-pair tables in addition to the information in our statistics panel to get additional values for precision and recall for each object descriptor:

$$\text{recall} = \frac{\text{number of correct positives}}{\text{total number of positives in the data set}}$$

And,

$$\text{precison} = \frac{\text{number of correct positives}}{\text{number of correct positives} + \text{number of false positives}}$$

The precision parameter describes the number of correct detections relative to the total number of detections. On the other hand, the threshold is usually set below the value of a false positive rate otherwise the number of false matches becomes too high to provide reliable scene recognition. For our experiment, we are more interested in knowing the number of false detections relative to the total number of detections, and so we make use of the 1-precision parameter:

$$1 - precision = \frac{\text{number of false positives}}{\text{number of correct positives + number of false positives}}$$

In our study context, a correct positive corresponds to a correct match between the two images while a false positive refers to a false match. The total number of positives in the dataset refers to the total number of correct matches that can be found. The number of correspondences is obtained using *repeatability*. Using the above definitions we redefine *recall* and *1- precision* as:

$$\text{recall} = \frac{\text{number of correct matches}}{\text{number of correspondences}}$$

and,

$$1 - precision = \frac{\text{number of false matches}}{\text{total number of matches}}$$

What about *possible* but not actualized matches? Given two images representing the same scene or image, the detection rate is the number of correctly matched points with respect to the number of possible matches:

$$Pcorrect = \frac{\#correct-matches}{\#possible-matches}$$

The false positive rate is the probability of a false match in a database of descriptors. Each descriptor of the query image is compared against each descriptor of the database counting the number of false matches. The probability of false positives is the total number of false matches with respect to the product of the number of database points and the number of image points:

$$P_{false} = \frac{\#\ of\ \ false\ matches}{(\#database\ points)(\#query\ image\ points)}$$

3.8 Reversed Search

A consequence of treating error pairs as a signal has important ramifications for reversed search, utilized with the inverted index approach. In this thesis we had forwarded the premise that any mismatches, errors or false positives can also be *learned* using an algorithm, to be indicative of the probe image. Therefore, our assumption purports that when these errors relative to a probe are consistent, in the context of our error-pairs, they are not quite unusable, since they can also be used to predict the particular probe image especially when an exact match is not found, in a

sort of reversed search process. The actual implication for real world search contexts occurs when our learning model defines a cluster of images to be most representative of a particular probe. Whenever these clusters occur as error pairs, they trigger the probe. Other contexts include situations where a user has to use an abstract description or even several image examples to target an actual scene. A hypothetical search context would be facial recognition and scene class retrieval: *Find the scene where X drives into the sunset* or *Find a scene where a mother holds her baby in Church* or *Find the Characters that look like X*. When the user locates the scene or character of interest, we propose that the cluster of images that were produced as a result of the search can also be a proxy for the same target image. This different approach to image-retrieval as forwarded by this thesis simply asserts the modest assumption that, probe or query images do not always have to be a one-against-all comparison; a cluster can also target one. In a future application along this line, we would want to have *several* images forwarded at once as our probes or query images, to enable us locate just *one* target image. Basically, what we are doing is reversing the image-retrieval process so that a cluster of features (that in the present appear in our Likelihood graph panel as a neighbourhood) can become a representative prototype for a single candidate image. It is somewhat like condensing the features of a neighbourhood cluster into one super candidate. As stated before, the search context could be text or image driven. To be able to locate the image features similar to what the user has in mind, or map the search texts to corresponding high-dimensional feature vectors, will mean simultaneously treating the combined features of several images as one, helping to also narrow down the search context. If the user had *one* particular scene or person in mind, and wants to retrieve similar scenes to be able to locate the target scene, this reversed approach could be a start. The effectiveness of this approach will almost always be influenced by the dictionary size of the relevant images.

In more conventional approaches, the dictionary size (that is, the number of visual words in the codebook) significantly affects the recognition performance and run time of the BoW approach. It has been demonstrated that using larger dictionaries, in the order of hundreds of thousands, improves performance and reduces search time in the inverted file. Dictionaries can be generated using the Approximate K-Means technique (AKM), Philbin et al., (2007), and by using random Kd-trees, Arya et al., (1998) to perform an approximate nearest neighbour search.

The table below is gotten from Aly et al. (2009). Notice how the increasing the dictionary size generally increases the recognition performance, especially with harder scenarios like 2 and 4.

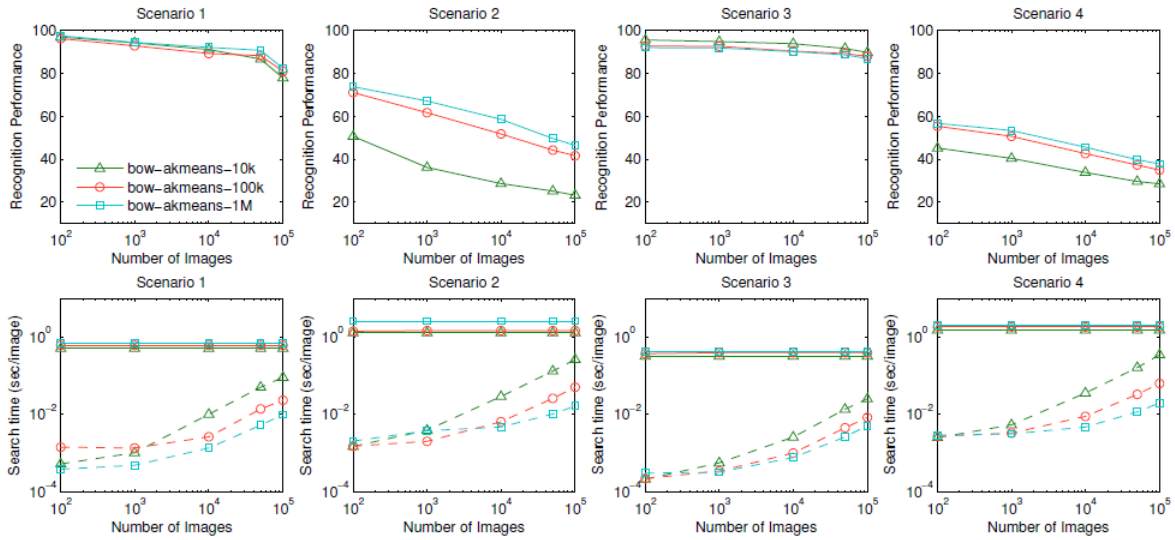

Table 3.6 The effect of dictionary size. Results for {*none*, l_1, l_1} combination with different dictionary sizes: 10K, 100K, and 1M visual words built with AKM. In the bottom row, solid lines represent time to compute visual words, while dashed lines show time to search the inverted file.

With larger dictionaries, the time to compute visual words for features increases slightly (since we are using Kd-trees), however, the time to search the IF decreases. This is intuitive since the number of images with similar words goes down as the number of words increases. This suggests that using larger dictionaries is generally the way to go. On the other hand, using the premise of our reversed search where combined multiple probe images are used to search for a single candidate image, will also have the effect of not only narrowing the search context, but also improving retrieval speeds accordingly.

Recall that every type of match or false positive ultimately depends on the type of descriptor used including the clustering strategy for the particular image-retrieval task.

$$D(X,M) = \sum_{\text{cluster } k} \sum_{\substack{\text{point } i \text{ in} \\ \text{cluster } k}} (\mathbf{x}_i - \mathbf{m}_k)^2$$

If we used K-means clustering to group comparable features from our dataset, and we also used a measure to minimize the sum of squared Euclidean distances between points $\mathbf{x}_i$ and their nearest cluster centers $\mathbf{m}_k$, there is a possibility of tweaking the algorithm to make centers, *seekers*:

i. Randomly initialize K cluster centers for our training images

ii. Iterate until convergence:

a) Assign each data point to the nearest center

b) Recompute each cluster center as the mean of all points assigned to it

Our argument for error pairs re-examines the figure below and appreciates it in a new way: Let these errors become new centres around the probe image. Let the green dots not only represent feature centres but also the key features from our probe image. Let the blue neighbourhood clusters around the green dots become all candidate features for an image match.

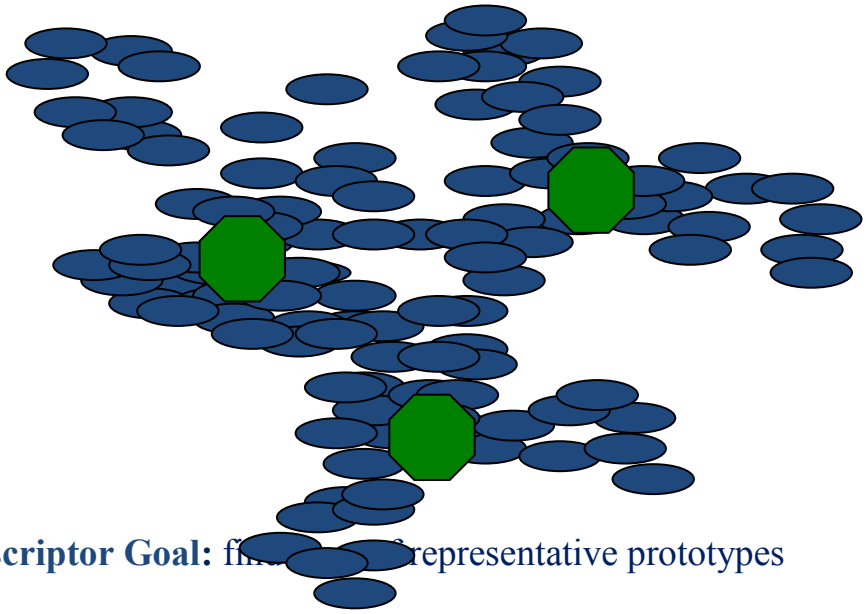

Figure 3.10 Centers amidst a neighbourhood feature cluster.

If our probe image consistently produces the same types of error clusters, then we can assume that errors themselves constitute a set of representative prototypes for our probe image, where we could take these set of errors in a reversed situation, and they would all point to the query image as a match. This also leads to newer insights for reversed searches. In normal searches, every blue dot above tries to cluster near the green centre based on some similarity measure. But using the error pairs approach within a reversed search context, the error-pairs are faced with two scenarios: i. Have no initial centre, but vote in the query image as a prototype centre ii. Combine their features to search against the remaining database for a new centre, relative to the representative features of the probe image. In other words, we make the cluster of blue dots attract the green centre instead. This can be done through a voting process where the features in the neighbourhood cluster may well define their relative distances from each, or from features in

the database, relative to a probe. The candidate centre with the most votes becomes the archetypal exemplar for that cluster.

In summary, when an image search occurs and we get a series of images that are expected to be visually similar to our query image but may not be in reality, what we are actually seeing are the results of the nearest neighbour features for that query image. Since we did not find an exact match, we could either dismiss the entire result as a mismatch, or we could decide to further train a learning model to associate the exact or approximate series of image clusters with the original query image we had in mind. In this regard, when trying to make approximate matches for the probe image, the error pairs can easily find its target. Thus, even errors are not wasted. The semantic gap is lessened when these types of additional contexts are also included in the retrieval task. In other situations, we may have several example images as our query image(s) combined to retrieve a single target image. This forms not only our desired result but also serves as a prototype for that cluster. The proposed reversed search model is also designed to narrow the search context in larger sized dictionaries where object descriptors can be very large. In our premise, we go beyond using features from a single query image, by expanding the context of features from multiple images now formed into a group, to search against the remaining database. Because each image provides a unique aspect to what is being searched for, it is predicted that the types of faces, objects, scene classes, or actions that can be predicted from this approach, will be more accurate than if the search only came from a single probe image. This also has the consequence of narrowing the search context, and the semantic gap during retrieval.

Chapter 4 Conclusion

4.1 Study Outcome and Contributions

Firstly, the amount of knowledge gained and generated in the course of this project has been staggering. It is a result of a deep desire to master machine learning techniques over the years, emerging from a deep need to learn something practical and challenging. Computer vision is hard and challenging, especially when one wishes to push the envelope. In this thesis we had investigated the concept of the semantic gap by exploring the various approaches that have already been adopted by a number of researchers using both ontological and data-driven techniques. We had examined instances of the semantic gap in our study and the various strategies that were adopted by various researchers to narrow that gap for image processing. We took particular note of ontological models as reflected in the thoughts of Bannour and Hudelot (2011):

> Ontologically-driven approaches are widely accepted now as very appropriate to model and take contexts into account. Thereby, unlike other techniques that allow a semantic description of images, ontologically driven approaches are best suited for image-retrieval systems as they model the semantics of images through relationships that help reasoning about it and understanding its meaning. (2)

However, in our own particular case, using the premise of several image-retrieval tasks, we had sought to expand the context by which additional semantic knowledge from low-level features can be incorporated to bridge the semantic gap for image-retrieval tasks. Evidently our goal is not the mere evaluation of the performances of different object descriptors. That type of evaluation has already been done with copious literature to that effect. We were very interested in discovering additional contexts for our image search. We had evaluated the types of mismatch errors generated by these different object descriptors under different parameters, for an added purpose: To help provide the keen insight into how these error patterns occur across the different descriptors. We were eager to understand if the notion of descriptor repeatability has been maintained under different parameters, sample sizes, and distance functions. The contribution of our image analysis comes from determining the relative importance of error-pairs. Although we were also very interested in discovering how each games cover possessed features that were

distinctive enough to be matched during an image search; which indeed, is something significant since an image's feature distinction effectively means that the global field of that image can easily be mapped directly to a high-level concept, since those features were unique enough (low-level feature uniqueness, equates the narrowing of the semantic gap by assigning high-level concepts to describe the global field of that image). But we were also interested in image mismatches that resulted from the many image-retrieval exercises that we made with our games content. We soon realized that these groups of images that cluster in the graph of our likelihood panel can also be a type of ontological representation of the probe image relative to the neighbouring cluster of images in that panel. In other words, what we have just done is expand the ontological contexts by which we could re-explain the query image by the types of results it generates from a learned dataset. This is a novel contribution of this thesis, which we hope to further investigate. Another significant contribution emerges from the enabling of a reversed image search process, where image results can now be used to determine a single prototype candidate. In other words, we had suggested the concept of multiple, simultaneous image searches, to contribute diverse features to what is being searched for and to also narrow the context of the search at the same time in a large database.

4.2 The Image-retrieval Tool

The image-retrieval tool used for this study will also be further developed to accommodate new clustering and search contexts. One of our goals is to enable the selection of a group of image clusters in the likelihood panel to be able to retrieve a target image that is a representative prototype of that cluster. The current application as it is does not have those features. Ultimately, another goal will be to develop a web and mobile-based equivalent of the image-retrieval tool to enable web-scale, and real world search context for images.

Works Cited

Addis, M., Boniface, M., Goodall, S., Grimwood, P., Kim, S., Lewis, P., Martinez, K., Steveson, A. *SCULPTEUR: Towards a New Paradigm for Multimedia Museum Information Handling*, In: *International Semantic Web Conference* (ISWC 2003), Florida, USA (2003) 582–596.

Agarwal, S., D.Roth, *Learning a sparse representation for object detection*, in: Proceedings of the 7th European Conference on Computer Vision Part IV, Springer-Verlag, 2002, pp.113–130.

Ahn, Luis von and Laura Dabbish. *Labeling Images with a Computer Game*. School of Computer Science Carnegie Mellon University Pittsburgh, PA, USA 2004.

Alahi A., Ortiz R., Vandergheynst P. *FREAK: Fast Retina Keypoint*. IEEE Conference on Computer Vision and Pattern Recognition, Rhode Island, Providence, USA. 2012

Aly, Mohamed, et al. *Towards automated large scale discovery of image families*. Computer Vision and Pattern Recognition Workshops, 2009. CVPR, Workshops 2009. IEEE Computer Society Conference on. IEEE, 2009.

Aly, Mario Munich, Pietro Perona, *Bag of Words for Large Scale Object Recognition*. Properties and Benchmark. Mohamed 1 Computational Vision Lab, Caltech, Pasadena, CA, USA.

Aly, Mohamed. Peter Welinder, Mario Munich Pietro *Perona Scaling Object Recognition: Benchmark of Current State of the Art Techniques*. Computational Vision Group, Caltech Pasadena, CA 91106.

Andoni, A. and P. Indyk. *E2lsh: Exact Euclidian locality-sensitive hashing*, 2004. http://web.mit.edu/andoni/www/LSH

Anuja Khodaskar and Dr. S.A. Ladke. *Content Based Image-retrieval with Semantic Features using Object Ontology*. *International Journal of Engineering Research & Technology* (IJERT), Vol. 1 Issue 4, June – 2012.

Artiemjew, Piotr., Przemyslaw Górecki, Krzysztof Sopyla. *Categorizaton of Similar Objects Using Bag of Visual Words and k-Nearest Neighbour Classifier*, Technical Sciences, Abbrev: Techn. Sc., No 15(2), 2012.

Acharya, Tinku, and Ajoy K. Ray. *Image processing: principles and applications*. John Wiley & Sons, 2005.

Arya, S., Mount, D., Netanyahu, N., Silverman, R., and Wu, A. *An optimal algorithm for approximate nearest neighbour searching*. Journal of the ACM, 45:891–923. 1998.

Arya, Sunil, et al. *An optimal algorithm for approximate nearest neighbour searching fixed dimensions*. *Journal of the ACM* (JACM) 45.6 (1998): 891-923.

Armitage, L.H., Enser, P.G.B.: *Analysis of user need in image archives*. *Journal of Information Sciences* 23, (1997) 287–299.

Bae, Soo Hyun, and Biing-Hwang Juang. *IPSILON: incremental parsing for semantic indexing of latent concepts*. *Image Processing*, IEEE Transactions on 19.7 (2010): 1933-1947.

Ballard, D. H. *Generalizing the Hough transform to detect arbitrary patterns*. Pattern Recognition, 13(2):111–122, 1981.

Bannour, Hichem, and Céline Hudelot. *Towards ontologies for image interpretation and annotation*. *Content-Based Multimedia Indexing* (CBMI), 2011, 9th International Workshop on. IEEE, 2011.

Bay H., Tuytelaars T., Van Gool L. *Surf: Speeded up robust features*. ECCV, p. 404–417. 2006.

Baeza-Yates, R. and Ribeiro-Neto, B. *Modern Information Retrieval*. ACM Press. 1999.

Blank, M. L. Gorelick, E. Shechtman, M. Irani, and R. Basri. *Actions as space-time shapes*. In ICCV, 2005.

Bosch, A., A. Zisserman, and X. Munoz, *Representing shape with a spatial pyramid kernel*, in Proceedings of the 6th ACM international conference on Image and video retrieval. ACM, 2007, pp. 401–408.

Bosch, A., A. Zisserman, and X. Munoz. *Scene classification using a hybrid generative/discriminative approach*. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 30, April 2008.

Boureau, Y., F. Bach, Y. LeCun, and J. Ponce. *Learning mid-level features for recognition*. In CVPR, 2010.

Brian Tomasik, Phyo Thiha, and Douglas Turnbull. *Tagging Products using Image Classification*. Dept. of Computer Science, Swarthmore College Swarthmore, PA 19081

Bruni, Elia., Ulisse Bordignon, Adam Liska, Jasper Uijlings and Irina Sergienya. *VSEM: An open library for visual semantics representation*. Proceedings of the ACL 2013. (System demonstration)

Bruni, Elia., Marco Baroni, Giang Binh Tran. *Distributional semantics from text and images*. GEMS '11: Proceedings of the GEMS 2011 Workshop on *Geometrical Models of Natural Language Semantics*. Association for Computational Linguistics 2011.

Broder, A. *On the resemblance and containment of documents*. In Proc. Compression and Complexity of Sequences 1997, pages 21–29. 1997.

Broder, A., Charikar, M., and Mizenmacher, M. *Min-wise independent permutations*. *Journal of Computer and System Sciences*, 60:630–659. 2000.

Broder, A. Z., Glassman, S. C., Manasse, M. S., and Zweig, G. *Syntactic clustering of the web*. *Computer Networks and ISDN Systems*, 29:8–13. 1997.

Brown, Matthew, and David G. Lowe. *Invariant Features from Interest Point Groups*. BMVC. No. s 1. 2002.

Brown, Matthew, and David G. Lowe. *Recognising panoramas*. ICCV. Vol. 3. 2003.

Cao, G., Nie, J.-Y.; and Bai, J. 2007. *Using markov chains to exploit word relationships in information retrieval*. In Evans, D.; Furui, S.; and Soul-Dupuy, C., eds., RIAO. CID. 2007.

Cameron Schaeffer. *A Comparison of Keypoint Descriptors in the Context of Pedestrian Detection: FREAK vs. SURF vs. BRISK*. Stanford University CS Department.

Chatfield, K. V. Lempitsky, A. Vedaldi, and A. Zisserman, *The devil is in the details: an evaluation of recent feature encoding methods*, in British Machine Vision Conference, 2011.

Chen, Yi, Umamahesh Srinivas, Vishal Monga and, Trac D. Tran. *Sparsity-based Face Recognition using Discriminative Graphical Models*. *Signals, Systems and Computers* (ASILOMAR), 2011 Conference Record of the Forty Fifth Asilomar Conference, 1204 - 1208, Nov. 2011.

Chen, Y., Zhou, X., and Huang, T. S. *One-class svm for learning in image-retrieval*. ICIP. 2001.

Chen,Y., T. T. Do, and T. D. Tran, *Robust face recognition using locally adaptive sparse representation*, in Proc. IEEE Intl. Conf. Image Processing, 2010, pp. 1657–1660.

Chen, T., Cheng, M.-M., Tan, P., Shamir, A., and Hu, S.M. *Sketch2photo: internet image montage*. ACM Trans. Graph. 28. 2009.

Chen, Z., Z. Song, Q. Huang, Y. Hua, and S. Yan, *Contextualizing object detection and classification*, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2011, pp. 1585–1592.

Chum, O., J. Philbin, M. Isard, and A. Zisserman. *Scalable near identical image and shot detection*. In CIVR, pages 549-556, 2007.

Chum, O., Philbin, J., Sivic, J., Isard, M., and Zisserman, A. *Total recall: Automatic query expansion with a generative feature model for object retrieval*. In ICCV. 2007.

Chum, O., Perdoch, M., and Matas, J. *Geometric min-hashing: Finding a (thick) needle in a haystack*. In CVPR, 2009.

Craw, I., D. Tock, and A. Beautt, *Finding Face Features*, in proc.2nd European Conf. on Computer Vision, pp 92-96, 1992.

Csurka, Gabriella, et al. *Visual categorization with bags of keypoints*. Workshop on statistical learning in computer vision, ECCV. Vol. 1. No.1-22. 2004.

Csurka, Gabriella, Christopher R. Dance, Lixin Fan, Jutta Willamowski, Cédric Bray. *Visual categorization with bags of keypoints*, In Workshop on *Statistical Learning in Computer Vision*, ECCV. 2004.

Cootes, Timothy F., et al. *Active shape models-their training and application*. *Computer vision and image understanding,* 61.1 (1995): 38-59.

Dalal, N. and B. Triggs. *Histograms of oriented gradients for human detection*. In CVPR, 1:886–893, 2005.

Datta, R., D. Joshi, J. Li, and J. Z. Wang, *Image-retrieval: ideas, influences, and trends of the new age*, ACM Computing Surveys, vol. 40, no. 2, article 5, 2008.

Delaitre, Vincent, Ivan Laptev, and Josef Sivic. *Recognizing human actions in still images: a study of bag-of-features and part-based representations*. In Frédéric Labrosse, Reyer Zwiggelaar, Yonghuai Liu, and Bernie Tiddeman, editors, *Proceedings of the British Machine Vision Conference*, pages 97.1-97.11. BMVA Press, September 2010.

Deng, Jia, et al. *Imagenet: A large-scale hierarchical image database*. *Computer Vision and Pattern Recognition*, 2009. CVPR 2009. IEEE Conference on. IEEE, 2009.

Deng, Jia and Wei Dong, Richard Socher, Li-Jia Li, Kai Li and Li Fei-Fei. *ImageNet: A Large-Scale Hierarchical Image Database*. *Computer Vision and Pattern Recognition*, IEEE Conference, CVPR 2009.

Deza E., Deza M. *Encyclopedia of Distances*, Springer. 2009.

Diakopoulos N., Essa I. A., Jain, and R.: *Content based image synthesis*. In CIVR (2004), pp. 299–307.

Duda, R. O. and Hart, P. E. *Use of the Hough transform to detect lines and curves in pictures*. *Communications of the ACM*, 15(1):11–15. 1972.

Duygulu, Pinar, et al. *Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary*. *Computer Vision*—ECCV 2002. Springer Berlin Heidelberg, 2002. 97-112.

Eitz, M., Hildebrand, K., Boubekeur, T., and Alexa, M. *Sketch-based image-retrieval: benchmark and bag of features descriptors*. IEEE TVCG. 2010.

Erik B. Sudderth. *Graphical Models for Visual Object Recognition and Tracking*, Department of Electrical Engineering and Computer Science, MIT, May, 2006.

Fabian, Junior, Ramon Pires, and Anderson Rocha. *Searching for people through textual and visual attributes*. Graphics, Patterns and Images (SIBGRAPI), 2012 25th SIBGRAPI Conference on. IEEE, 2012.

Fauqueur J., Boujemaa N. *Logical query composition from local visual feature thesaurus*. In Third International Workshop on Content-Based Multimedia Indexing (CBMI'03) 2003.

Fischler, Martin A., Jay M. Tenenbaum, and Hans Christoph Wolf. *Detection of roads and linear structures in low-resolution aerial imagery using a multisource knowledge integration technique*. *Computer Graphics and Image Processing* 15.3 (1981): 201-223.

Foo, J. , J. Zobel, R. Sinha, and S. Tahaghoghi. *Detection of near-duplicate images for web search*. In CIVR, pages 557-564, 2007.

Forsyth, David A., and Jean Ponce. *Computer vision: a modern approach*. Prentice Hall Professional Technical Reference, 2002.

Freeman, W. T., E. C. Pasztor, and O. T. Carmichael. *Learning low–level vision*. International Journal of Computer Vision, 40(1):25–47, 2000.

Freemany, William T. *Where computer vision needs help from computer science*. ACM-SIAM Symposium on Discrete Algorithms (SODA), January, 2011, invited talk.

Friedman, J.H., Bentley, J.L. and Finkel, R.A. 1977. *An algorithm for finding best matches in logarithmic expected time*. ACM Transactions on Mathematical Software, 3(3):209-226.

Friedman, Jerome H., Jon Louis Bentley, and Raphael Ari Finkel. *An algorithm for finding best matches in logarithmic expected time*. ACM Transactions on Mathematical Software (TOMS) 3.3 (1977): 209-226.

Frey, B. J. and N. Jojic. *A comparison of algorithms for inference and learning in probabilistic graphical models.* IEEE Transactions on *Pattern Analysis and Machine Intelligence*, 27(9):1392–1416, September 2005.

Gammeter, S., Bossard, L., Quack, T. & Van Gool, L. I Know *What You Did Last Summer: Object-Level Auto-Annotation of Holiday Snaps*, in '*Proceedings of the IEEE International Conference on Computer Vision*', 2009.

Gao, B. T. Liu, T. Qin, X. Zheng, Q. Cheng, and W. Ma *Web image clustering by consistent utilization of visual features and surrounding texts*. In *MULTIMEDIA*, pages 112 121, New York, NY, USA, ACM. 2005.

Geman, S. and D. Geman. *Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images*. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, November 1984.

Goodall, S., Lewis, P.H., Martinez, K., Sinclair, P., Addis, M., Lahanier, C., Stevenson, and J.: *Knowledge-based exploration of multimedia museum collections*. In: *Proceedings of European Workshop on the Integration of Knowledge, Semantics and Digital Media Technology* (EWIMT), London, U.K. (2004).

Gordon, S., H. Greenspan, and J. Goldberger. *Applying the information bottleneck principle to unsupervised clustering of discrete and continuous image representations*. In ICCV, 2003.

Grauman, Kristen, and Bastian Leibe. *Visual object recognition*. No. 11. Morgan & Claypool Publishers, 2011.

Grauman, K. and T. Darrell, *The pyramid match kernel: Discriminative classification with sets of image features*, in *IEEE International Conference on Computer Vision* (ICCV), 2005, vol. 2, pp. 1458–1465.

Green, P. J., *Reversible Jump Markov Chain Monte Carlo Computation and Bayesian Model Determination*. *Biometrika*, vol. 82, no. 4, pp. 711-732, 1995.

Grzeszick, Rene., Leonard Rothacker, Gernot A. Fink. *Bag-of-Features Representations using Spatial Visual Vocabularies for Object Classification*, In Proc. IEEE Intl. Conf. on Image Processing, Melbourne, Australia, 2013.

Hardman, Lynda. *Knowledge Space of semantic inference for automatic annotation and retrieval of multimedia content*. http://homepages.cwi.nl/~media/samt08/ , 2008.

Hare, Jonathon S., Patrick A.S. Sinclair1, Paul H. Lewis, Kirk Martinez, Peter G. B. Enser, and Christine J. Sandom. *Bridging the Semantic Gap in Multimedia Information Retrieval Top-down and Bottom-up Approaches*, Enser, P.G.B.: *Pictorial information retrieval*. Journal of Documentation (1995) 126–170.

Hare, Jonathon S., et al. *Bridging the semantic gap in multimedia information retrieval: Top-down and bottom-up approaches*. (2006).

Harris, Chris, and Mike Stephens. *A combined corner and edge detector*. Alvey vision conference. Vol. 15. 1988.

Harris, C. *Geometry from visual motion*. In *Active Vision*, A. Blake and A. Yuille (Eds.), MIT Press, pp. 263-284. 1992.

Heisele, Bernd, Thomas Serre, and Tomaso Poggio. *A component-based framework for face detection and identification*. *International Journal of Computer Vision* 74.2 (2007): 167-181.

Hollink, L., Schreiber, A.T., Wielinga, B.J., Worring, M.: *Classification of user image descriptions*. Int. J. Hum.-Comput. Stud. 61 (2004) 601–626.

Hough, P. V. C. *Method and means for recognizing complex patterns*. U. S. Patent, 3,069,654. 1962.

Hu, B., Dasmahapatra, S., Lewis, P., Shadbolt, N. *Ontology-based medical image annotation with description logics*. In: Proceedings of The 15th *IEEE International Conference on Tools with Artificial Intelligence*, IEEE Computer Society Press (2003) 77–82.

Hunter, J.: *Adding multimedia to the semantic web*: *Building an mpeg-7 ontology*. In Cruz, I.F., Decker, S., Euzenat, J., McGuinness, D.L., eds.: SWWS. (2001) 261–283.

Illingworth, J. and Kittler, J. *A survey of the Hough transforms*. *Computer Vision, Graphics, and Image Processing*, 44:87–116. 1988.

Indyk, P., and R. Motwani. *Approximate nearest neighbour: Towards removing the curse of dimensionality*. In *Symposium on Theory of Computing*, 1998.

Ionescu, Radu Tudor, Marius Popescu, and Cristian Grozea. *Local learning to improve bag of visual words model for facial expression recognition*. Workshop on *Challenges in Representation Learning*, ICML. 2013.

Jassim T.Sarsoh, Kadhem M.Hashem & Mohammed A.Al-Hadi. *Classifying of Human Face Images Based on the Graph Theory Concepts*. Global Journals Inc. Volume 12 Issue 13 Version 1.0 Year 2012.

Jain, A. K ., M. N. Murty, and P. J. Flynn, *Data Clustering Review*, ACM Computing Survey, 3, 31, 264, 1999.

Jain, Anil K., Robert P. W. Duin, and Jianchang Mao. *Statistical pattern recognition: A review*. Pattern Analysis and Machine Intelligence, IEEE Transactions on 22.1 (2000): 4-37.

Jégou, H., Douze, M., and Schmid, C. *Hamming embedding and weak geometric consistency for large scale image search*. In ECCV. 2008.

Jégou, H., Douze, M., and Schmid, C., *Packing bag-of-features*. In ICCV. 2009.

Jégou, Hervé, Matthijs Douze, and Cordelia Schmid. *Improving bag-of-features for large scale image search*. International Journal of Computer, Vision 87.3 (2010): 316-336.

Jiang, Y.-G., Ngo, C.-W.: *Visual word proximity and linguistics for semantic video indexing and near-duplicate retrieval*. *Computer Vision and Image Understanding*. 113(3), 405–414, 2009.

Jianxiong Xiao, James Hays, Krista Ehinger, Aude Oliva, and Antonio Torralba. *Sun database: Large-scale scene recognition from abbey to zoo*. In Proc. Conf. on *Computer Vision and Patter Recognition*, USA, 2010.

Johnson, Matthew, et al. *Semantic photo synthesis*. Computer Graphics Forum. Vol. 25. No. 3. Blackwell Publishing, Inc, 2006.

Jordan, M. I. *Graphical models*. Statistical Science, 19(1):140–155, 2004.

Jurie, F. and B. Triggs, *Creating efficient codebooks for visual recognition*, in CVPR, 2005.

Kande, T, *Computer Recognition of Human Face*, Based and Styttgrat, Birkhausar, 1977.

Ke, Y., R. Sukthankar, and L. Huston. *An efficient parts-based near-duplicate and sub-image-retrieval system*. In Multimedia, pages 869-876, 2004.

Kemelmacher-Shlizerman, I., Shechtman, E., Garg, R., and Seitz, S. M. *Exploring photobios*. SIGGRAPH. 2011.

Kherfi, M. L., D. Ziou, and A. Bernardi. *Image-retrieval from the World Wide Web: issues, techniques, and systems.* ACM Computing Surveys, vol. 36, no. 1, pp. 35–67, 2004.

Kläser, Alexander. *Learning human actions in video*. Dissertation. PhD thesis, Université de Grenoble, 2010.

Klette, Reinhard. *Concise Computer Vision*. (2014).

Kosowsky, J. J., and Alan L. Yuille. *The invisible hand algorithm: Solving the assignment problem with statistical physics*. *Neural networks* 7.3 (1994): 477-490.

Kristen Grauman, Bastian Leibe. *Visual Object Recognition*, Morgan & Claypool Publishers, 2011.

Krüger, Norbert, Gabriele Peters, and Christoph Von Der Malsburg. *Object recognition with a sparse and autonomously learned representation based on banana wavelets*. *Learned Representation based on Banana Wavelets*, Technical Report IR-INI 96-11. Institut Fur Neuroinformattik, Rurh-Universitat Bochum. 1996.

Kirt Lillywhite, Dah-JyeLee, BeauTippetts, and JamesArchibald. *A feature construction method for general object recognition*, in *Pattern Recognition*, Published by Elsevier Ltd. 2013.

Kumar S., Hebert M.: *A hierarchical field framework for unified context-based classification*. In Proc. ICCV, October 2005.

Lampert, Christoph H., Hannes Nickisch and Stefan Harmeling. *Learning To Detect Unseen Object Classes by Between-Class Attribute Transfer*. *Computer Vision and Pattern Recognition*, CVPR, IEEE Conference on June 2009.

Laptev, Ivan. *On space-time interest points*. *International Journal of Computer Vision* 64.2-3 (2005): 107-123.

Lauritzen, S. L., *Graphical Models*. Oxford University Press, NY, 1996.

Lazebnik, S., C. Schmid, and J. Ponce, *Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories*, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2006, vol. 2, pp. 2169–2178.

Lazebnik, S., Schmid, C., and Ponce, J. *Spatial pyramid matching*. In *Object Categorization: Computer and Human Vision Perspectives*. Cambridge University Press. 2009.

Le, D.T., J.R.R. Uijlings, R. Bernardi, *Exploiting Language Models for Visual Recognition*, In EMNLP, 2013.

Leordeanu, M. Hebert, M. Sukthankar, R. *Beyond Local Appearance: Category Recognition from Pairwise Interactions of Simple Features*. *Computer Vision and Pattern Recognition*, CVPR, 2007.

Leibe, B., A. Ettlin, and B. Schiele, *Learning semantic object parts for object categorization*. *Image and Vision Computing*, vol. 26, no. 1, pp. 15–26, 2008.

Leung, Thomas K., Michael C. Burl, and Pietro Perona. *Finding faces in cluttered scenes using random labeled graph matching*. *Computer Vision*, 1995. Proceedings., Fifth International Conference on. IEEE, 1995.

Leuttengger S., Chli M., Siegwart R. 2011. *BRISK: Binary Robust Invariant Scalable Keypoints*. *Proceedings of the IEEE International Conference on Computer Vision* (ICCV), pp. 2548–2555.

Li., Fei-Fei and Perona, P. *A bayesian hierarchical model for learning natural scene categories*. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2, 524 – 531 vol. 2. May 2005.

Li, Li-Jia, and Li Fei-Fei. *What, where and who? classifying events by scene and object recognition*. *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*. IEEE, 2007.

Li, L.-J., Socher, R., Fei-Fei, and L.: *Towards total scene understanding: classification, annotation and segmentation in an automatic framework*. In: Proc. *IEEE Computer Vision and Pattern Recognition*, CVPR 2009.

Li, Fei-Fei and P. Perona. *A bayesian hierarchical model for learning natural scene categories*. In CVPR, volume 2, pages 524-531, 2005.

Li, Li-Jia, Richard Socher, and Li Fei-Fei. *Towards total scene understanding: Classification, annotation and segmentation in an automatic framework*. *Computer Vision and Pattern Recognition*, 2009. CVPR 2009. IEEE Conference on. IEEE, 2009.

Li, T., T. Mei, I.-S. Kweon, and X.-S. Hua, *Contextual bag-of-words for visual categorization*, IEEE TCSVT, vol. 21, no. 4, pp. 381–392, 2011.

Lillywhite, Kirt, et al. *A feature construction method for general object recognition*. *Pattern Recognition* 46.12 (2013): 3300-3314.

Liska, Adam. *Semantic Spaces from Images*. Department of Computer Science, University of Melbourne. Master thesis, September, 2013.

Lopez-Sastre, Roberto J., et al. *Heterogeneous Visual Codebook Integration via Consensus Clustering for Visual Categorization*. *IEEE transactions on circuits and systems for video technology* 23.8 (2013): 1358-1368.

Lowe, D.G., *Object recognition from local scale invariant features*, in *IEEE International Conference on Computer Vision* (ICCV), 1999, vol. 2, pp. 1150–1157.

Lowe, D.G., *Distinctive image features from scale invariant keypoints*, *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

Lowe, D. G, and Muja, M. *Fast approximate nearest neighbours with automatic algorithm configuration*. In Intl. Conf. on *Computer Vision Theory and Applications* (VISAPP'09), 2009.

Malik, Jitendra, et al. *Textons, contours and regions: Cue integration in image segmentation*. Computer Vision, 1999. *The Proceedings of the Seventh IEEE International Conference* on. Vol. 2. IEEE, 1999.

Marian-Andrei Rizoiui, Julien Velchin and Stephane Lalich. *Semantic-enriched Visual Vocabulary Construction in a Weakly Supervised Context*. *Intelligent Data Analysis* 19, 2014.

Markoff, John. *Seeking a Better Way to Find Web Images*. *The New York Times*. November 19, 2012. http://www.nytimes.com/2012/11/20/science/for-web-images-creating-new-technology-to-seek-and-find.html?_r=5&

Mehdi Mirza-Mohammadi, Sergio Escalera1, and Petia Radeva1. *Contextual-Guided Bag-of-Visual-Words Model for Multi-class Object Categorization*. Dept. Matematica Aplicada i Analisi, Gran Via 585, 08007, Barcelona, Spain, Computer Vision Center, Campus UAB, Edifici O, 08193, Bellaterra, Barcelona.

Miller, George A., and Walter G. Charles. *Contextual correlates of semantic similarity*. Language and cognitive processes 6.1 (1991): 1-28.

Mikolajczyk, K. and C. Schmid. *Scale and affine invariant interest point detectors*. IJCV, 60:63-86, 2004.

Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Kadir, T., Van Gool, L.: *A comparison of affine region detectors*. IJCV 65(1-2), 43–72, 2005.

Mikolajczyk, Krystian, and Cordelia Schmid. *A performance evaluation of local descriptors. Pattern Analysis and Machine Intelligence*. IEEE Transactions on 27.10 (2005): 1615-1630.

Mikolajczyk, K., Leibe, B., and B. Schiele, *Efficient clustering and matching for object class recognition*, in *BMVC*, 2006.

Mikolajczyk, Krystian, and Cordelia Schmid. *A performance evaluation of local descriptors*. *Pattern Analysis and Machine Intelligence*, IEEE Transactions on 27.10 (2005): 1615-1630.

Min, Hu, and Yang Shuangyuan. *Overview of content-based image-retrieval with high-level semantics*. *Advanced Computer Theory and Engineering* (ICACTE), 2010 3rd International Conference on. Vol. 6. IEEE, 2010.

Moeslund, Thomas B., Adrian Hilton, and Volker Krüger. *A survey of advances in vision-based human motion capture and analysis*. *Computer vision and image understanding* 104.2 (2006): 90-126.

Moghaddam, Baback, and Alex Pentland. *Probabilistic visual learning for object representation*. *Pattern Analysis and Machine Intelligence*, IEEE Transactions on 19.7 (1997): 696-710.

Moosmann, F., E. Nowak, and F. Jurie, *Randomized clustering forests for image classification*, IEEE PAMI, vol. 30, no. 9, pp. 1632–1646, 2008.

Mori, Yahuside T. H., Ryuichi O., *Image-to-Word transformation Based on dividing and vector quantizating image with words* », In MISRM'99 First international workshop on multimedia intelligent storage and retrieval management, 1999.

Muja, M. and D. G. Lowe. *Fast approximate nearest neighbours with automatic algorithm configuration*. In Int. Conf. on Comp. Vision Theory and Applications, 2009.

Nalina, P., Muthukannan, K., *Survey on Image Segmentation Using Graph Based Methods*. International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-6, January 2013.

Nister, D. and H. Stewenius. *Scalable recognition with a vocabulary tree*. In CVPR '06, pages 2161-2168.

Nixon, Mark. *Feature extraction & image processing*. Academic Press, 2008.

Parker, J. R., *Algorithms for Image Processing and Computer Vision*. 2nd Edition. Wiley Publishing, Inc. 2011.

Penev, Penio S., and Joseph J. Atick. *Local feature analysis: A general statistical theory for object representation*. *Network: computation in neural systems* 7.3 (1996): 477-500.

Perronnin, Florent, et al. *Adapted vocabularies for generic visual categorization*. Computer Vision–ECCV 2006. Springer Berlin Heidelberg, 2006. 464-475.

Philbin, J., Chum, O., Isard, M., Sivic, J., and Zisserman, A. *Object retrieval with large vocabularies and fast spatial matching*. CVPR. 2007.

Philbin, J., Chum, O., Isard, M., Sivic, J., and Zisserman, A. *Lost in quantization: Improving particular object retrieval in large scale image databases*. In CVPR. 2008.

Poppe, Ronald. *A survey on vision-based human action recognition*. *Image and vision computing* 28.6 (2010): 976-990.

Olivia, A., and Torralba, A. *Building the gist of a scene: the role of global image features in recognition*. *Progress in Brain Research*. 2006.

Quack, T., Leibe, B. & Van Gool, L., *World-Scale Mining of Objects and Events from Community Photo Collections*, in '*ACM International Conference on Image and Video Retrieval*'. 2006.

Quelhas, P., F. Monay, J.-M. Odobez, D. Gatica-Perez, T. Tuytelaars, and L. Van Gool, *Modeling scenes with local descriptors and latent aspects*, in ICCV, 2005.

Quelhas, Pedro, et al. *Modeling scenes with local descriptors and latent aspects*. Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on. Vol. 1. IEEE, 2005.

Quoc V. Le , Rajat Monga , Matthieu Devin , Kai Chen , Greg S. Corrado , Jeff Dean , Andrew Y. Ng. *Building High-level Features Using Large Scale Unsupervised Learning*. In *International Conference on Machine Learning*, 2012.

Ren, Reede, John Collomosse, and Joemon Jose. *A BOVW based query generative model*. *Advances in Multimedia Modeling*. Springer Berlin Heidelberg, 2011. 118-128.

Rizoiu, Marian-Andrei, Julien Velcin, and Stéphane Lallich. *Semantic-enriched Visual Vocabulary Construction in a Weakly Supervised Context*. Intelligent Data Analysis 19.1 (2014).

Rublee, Ethan, et al. *ORB: an efficient alternative to SIFT or SURF*. *Computer Vision* (ICCV), 2011 IEEE International Conference on. IEEE, 2011.

Salton, G. *Automatic Information Organization and Retrieval*. McGraw Hill Text, 1968.

Sande, K. van de, T. Gevers, and C. Snoek, *Evaluating color descriptors for object and scene recognition*, PAMI, vol. 32, pp. 1582–1596, 2010.

Schmid, Cordelia, and Roger Mohr. *Local gray value invariants for image-retrieval*. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 19.5 (1997): 530-534.

Schneiderman, Henry, and Takeo Kanade. *Object detection using the statistics of parts*. *International Journal of Computer Vision* 56.3 (2004): 151-177.

Schreiber, A.T.G., Dubbeldam, B., Wielemaker, J., Wielinga, B.: *Ontology-based photo annotation*. *IEEE Intelligent Systems* 16 (2001) 66–74.

Schreiber, A. Th Guus, et al. *Ontology-based photo annotation*. *IEEE Intelligent Systems* 16.3 (2001): 66-74.

Se, Stephen, David Lowe, and Jim Little. *Global localization using distinctive visual features*. *Intelligent Robots and Systems*, 2002. IEEE/RSJ International Conference on. Vol. 1. IEEE, 2002.

Shi, J. and Tomasi, C. *Good features to track*. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (CVPR'94), pp. 593–600, Seattle. 1994.

Shotton, J., M. Johnson, , and R. Cipolla. *Semantic texton forests for image categorization and segmentation*. In CVPR, 1:1–8, 2008.

Shotton J., Winn J., Rother C., and Criminisi A. *Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation*. In ECCV, 2006.

Sivic, Josef, and Andrew Zisserman. *Video Google: A text retrieval approach to object matching in videos*. *Computer Vision*, 2003. Proceedings. Ninth IEEE International Conference on. IEEE, 2003.

Sivic, and A. Zisserman, J. Philbin, O. Chum, M. Isard, J.. *Object retrieval with large vocabularies and fast spatial matching*. In CVPR, 2007.

Sivic, Josef., Frederik Schaffalitzky, and Andrew Zisserman. *Efficient object retrieval from videos*. In Proc. European Signal Processing Conference, Austria, 2004.

Sivic, J., Russell, B. C., Ponce, J., and Dessales, H. *Automatic alignment of paintings and photographs depicting a 3d scene*. In *3D Representation and Recognition* (3dRR). 2011.

Smeulders, A. W. M., M. Worring, S. Santini, A. Gupta, and R. Jain, *Content-based image-retrieval at the end of the early years*, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 12, pp. 1349–1380, 2000.

Snoek, C.G.M., van de Sande, K.E.A., de Rooij, O., Huurnink, B., van Gemert, J., Uijlings, J.R.R., He, J., Li, X., Everts, I., Nedovic, V., van Liempt, M., van Balen, R., de Rijke, M., Geusebroek, J.-M., Gevers, T., Worring, M., Smeulders, A.W.M., Koelma, D., Yan, F., Tahir, M.A., Mikolajczyk, K., Kittler, J.: *The media mill* TRECVID 2009 Semantic video search engine. In: *TRECVID*, 2009.

Starovoitov, V., Samal, and Sankur, *Matching of Face in a Camera Image and Document Photographs*, Institute of Engineering Cybernetic Suraganora, Min, Belarus, (1997).

Steggink, Jeroen., and Cees GM Snoek. *Adding semantics to image-region annotations with the name-it-game*. Multimedia systems 17.5 (2011): 367-378.

Su, Y. and F. Jurie, *Visual word disambiguation by semantic contexts*, in ICCV, 2011.

Szeliski, Richard. *Computer vision: algorithms and applications*. Springer, 2010.

Szeliski, R., 2010. Personal communication.

Szeliski, R., *Computer Vision: Algorithms and Applications*. Springer, 2010

Thorsten J. *Text categorization with support vector machines: learning with many relevant features*. Proceedings of ECML-98, 10th European Conference on Machine Learning, 1398: 137–142. Eds. C. N'edellec, C. Rouveirol. Springer Verlag, Heidelberg, DE. 1998.

Tirilly, Pierre, Vincent Claveau, and Patrick Gros. *Language modeling for bag-of-visual words image categorization*. Proceedings of the 2008 international conference on, *Content-based image and video retrieval*. ACM, 2008.

Tomasik, Brian, Phyo Thiha, and Douglas Turnbull. *Tagging products using image classification*. Proceedings of the 32nd international ACM SIGIR conference on *Research and development in information retrieval*. ACM, 2009.

Triggs, B. *Detecting keypoints with stable position, orientation, and scale under illumination changes*. In Eighth European Conference on *Computer Vision* (ECCV 2004), pp. 100–113, Prague, 2004.

Tsai, Chih-Fong. *Bag-of-words representation in image annotation: A review*. ISRN Artificial Intelligence, 2012.

Tsinaraki, C., Polydoros, P., Moumoutzis, N., Christodoulakis, and S., *Coupling owl with mpeg-7 and tv-anytime for domain-specific multimedia information integration and retrieval*. In: Proceedings of RIAO 2004, Avignon, France (2004).

Tu, Zhuowen, et al. *Image parsing: Unifying segmentation, detection, and recognition*. *International Journal of Computer Vision* 63.2 (2005): 113-140.

Tu, Z., and S.C. Zhu, *Image segmentation by Data-driven Markov chain Monte Carlo*, IEEE Trans. PAMI, vol. 24, no.5, pp. 657-673, 2002.

Tu, Z.W., and S.C. Zhu, *Parsing images into regions, curves and curve groups*. *Int'l Journal of Computer Vision*, (Under review), A short version appeared in the Proc. of ECCV, 2002.

Turney, Peter D., and Patrick Pantel. *From frequency to meaning: Vector space models of semantics*. *Journal of artificial intelligence research* 37.1 (2010): 141-188.

Tuytelaars T., Gool L. V. *Matching widely separated views based on affine invariant regions*. *International Journal of Computer Vision* 59, 1, 61–85. 2004.

Uijlings., J.R.R., Smeulders, A.W.M., Scha, R.J.H.: *Real-time bag of words*, *approximately*. In: CIVR 2009, Santorini, Fira, Greece, pp. 1–8. ACM, New York 2009

Vedaldi, Andrea, and Brian Fulkerson. *VLFeat: An open and portable library of computer vision algorithms*. Proceedings of the international conference on Multimedia. ACM, 2010.

Vedaldi, A., Gulshan, V., Varma, M., and Zisserman, A. *Multiple kernels for object detection*. ICCV. 2009.

Wainwright, M. J. and M. I. Jordan, *Graphical models, exponential families and variational inference*. *Foundations and Trends in Machine Learning*, vol. 1, no. 1-2, pp. 1–305, 2008.

Wang, H. , J. Yuan, and Y.-P. Tan, *Combining feature context and spatial context for image pattern discovery*, in IEEE ICDM, 2011.

Winn, J., A. Criminisi, and A. Minka, *Object categorization by learned universal visual dictionary*, in ICCV, 2005.

Wittgenstein, Ludwig. *Preliminary Studies for the Philosophical Investigations*. (1969).

Wu, J. and J.M. Rehg, *Centrist: A visual descriptor for scene categorization*, IEEE Transactions on *Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1489–1501, 2011.

Yuan, J. and Y. Wu, *Context-aware clustering*, in CVPR, 2008.

Yang, Ming-Hsuan, David Kriegman, and Narendra Ahuja. *Detecting faces in images: A survey*. *Pattern Analysis and Machine Intelligence*, IEEE Transactions on 24.1 (2002): 34-58.

Yang, L. *Distance Metric Learning: A Comprehensive Survey*, Technical report, Michigan State Univ. 2006.

Yang, L., R. Jin, R. Sukthankar, and F. Jurie, *Unifying discriminative visual codebook generation with classifier training for object category recognition*, in CVPR, 2008.

Yang, J., K. Yu, and T. Huang. *Efficient highly over-complete sparse coding using a mixture model*. In ECCV, 2010.

Zhang, J. M. Marszalek, S. Lazebnik, and C. Schmid. *Local features and kernels for classificaiton of texture and object categories: A comprehensive study*. *Int. Journal of Computer Vision*, 73(2):213–238, 2007.

Zhang, S., Q. Tian, G. Hua, W. Zhou, Q. Huang, H. Li, and W. Gao, *Modeling spatial and semantic cues for large-scale near-duplicated image-retrieval*, CVIU, vol. 115, pp. 403–414, 2011.

Zhao, Rong, and William I. Grosky. *Narrowing the semantic gap-improved text-based web document retrieval using visual features*. Multimedia, IEEE Transactions on 4.2 (2002): 189-200.

Zhao, R., Grosky, W.I., *From features to semantics: Some preliminary results*. In*: IEEE International Conference on Multimedia and Expo (II)*. (2000) 679–682.

Zhu, S.C., R. Zhang, and Z.W. Tu, *Integrating top-down/bottom-up for object recognition by data-driven Markov chain Monte Carlo*. Proc. of *IEEE Conf. on Computer Vision and Pattern Recognition*, Hilton Head, SC. 2000.

Zhu, Song-Chun, et al. *What are textons?*. *International Journal of Computer Vision* 62.1-2 (2005): 121-143.

Zisserman, A. M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and*, The PASCAL Visual Object Classes (VOC) challenge*, IJCV, vol. 88, no. 2, pp. 303–338, 2010.

Zobel, J. and Moffat, A. *Inverted files for text search engines*. ACM Computational Survey. 2006.

APPENDIX A

More Screenshots from application showing mismatches

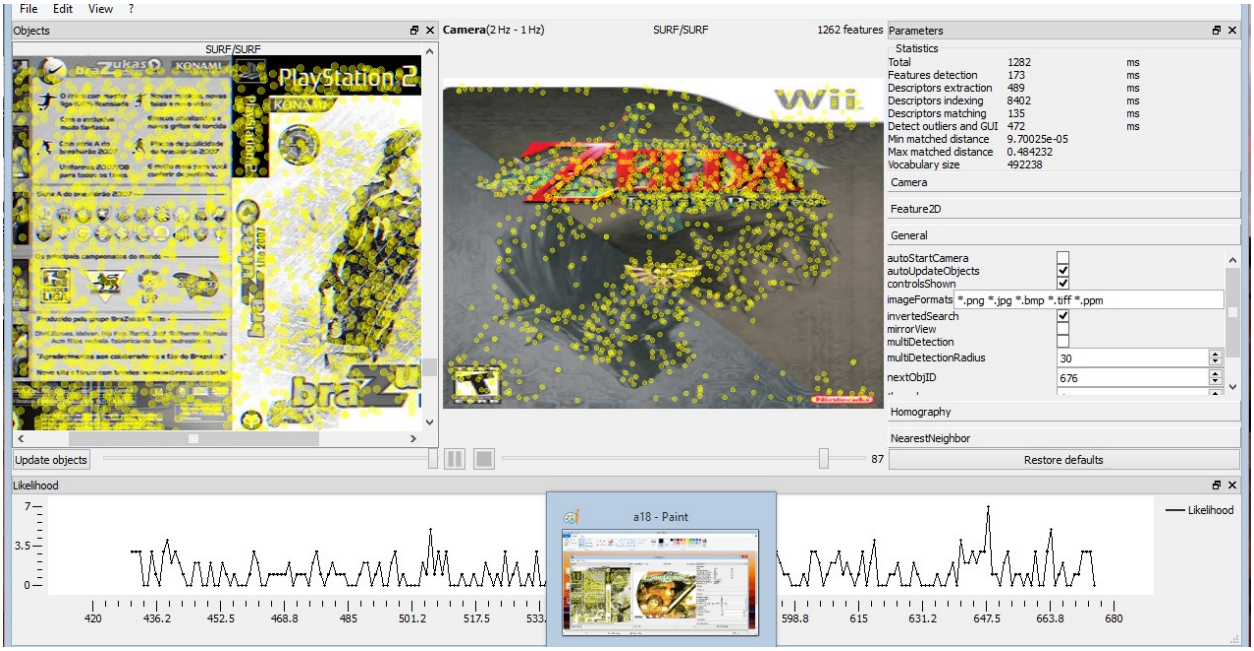

Figure A.1 General color templates and image texture seem to resemble, although they are thematically unlike

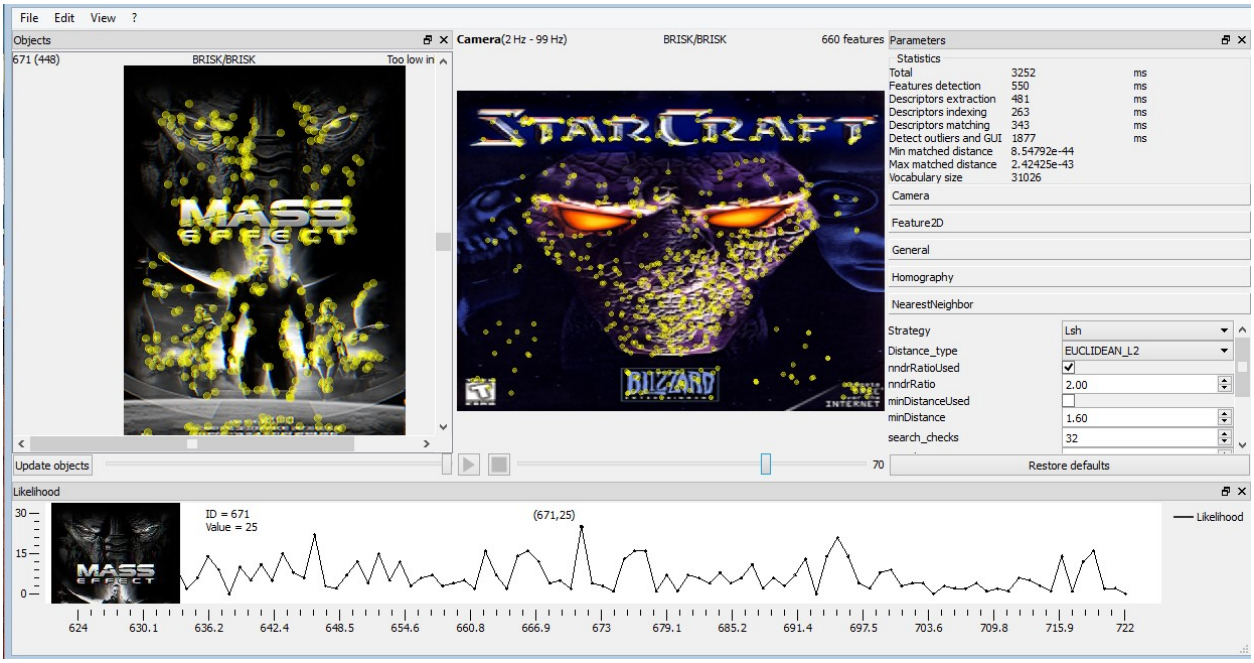

Figure A.2 The darker colors here may also coincide with a darker theme from both games.

APPENDIX B

The Inverted Search Process

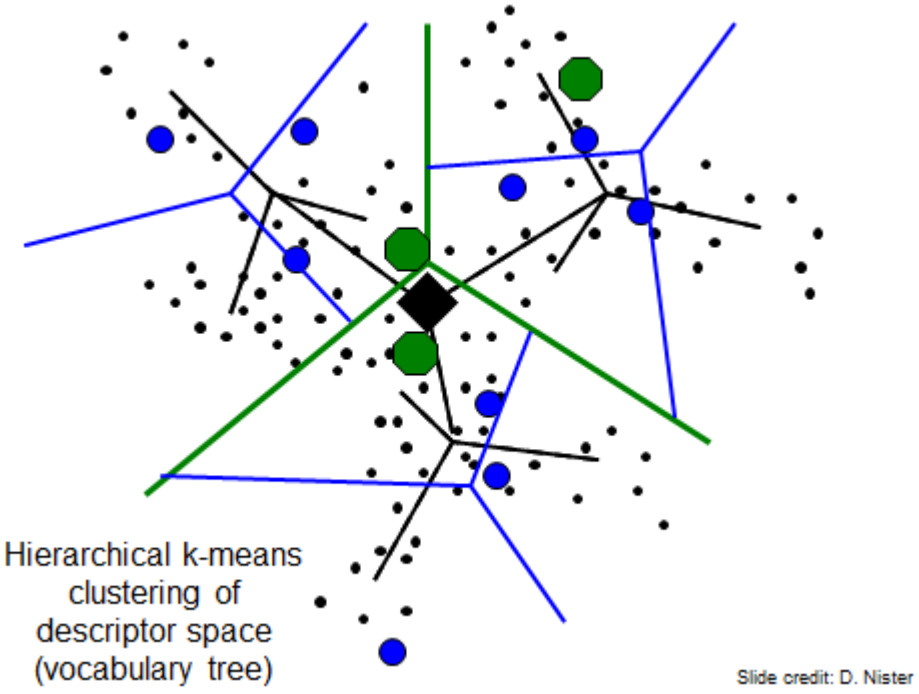

Figure B.1 Clusters and centers

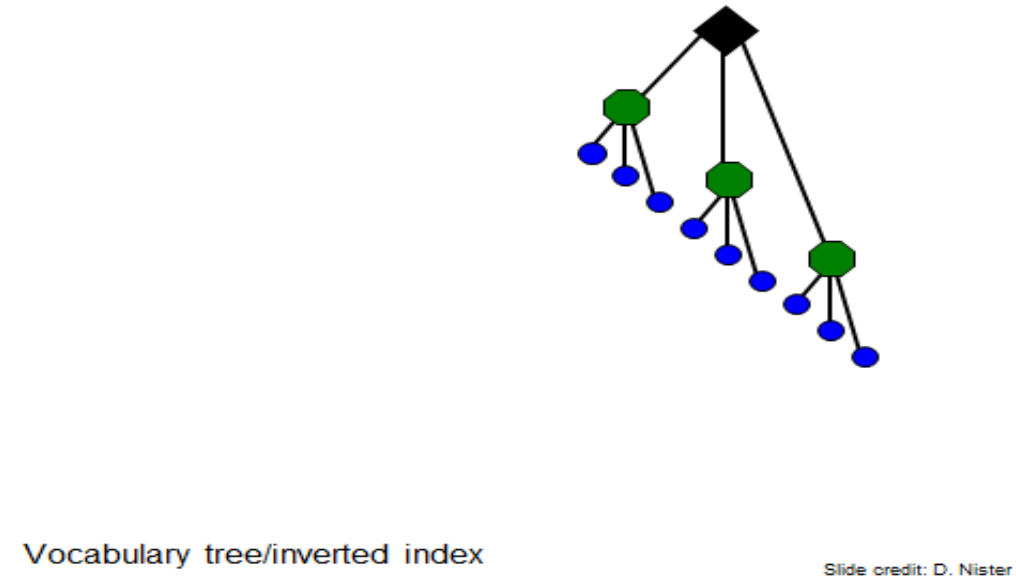

Figure B.2 Inverted Index

APPENDIX B

The Inverted Search Process

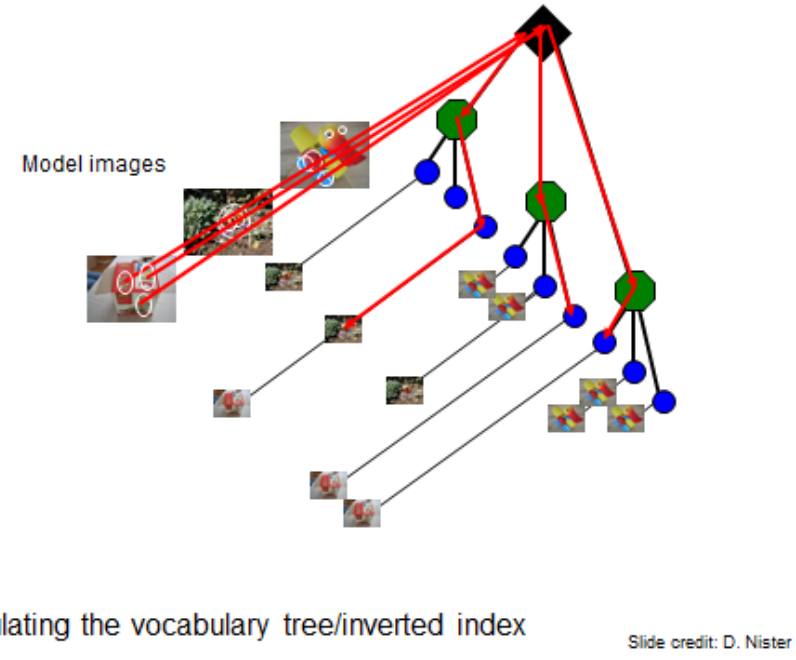

Figure B.3 Populating the Vocabulary Tree

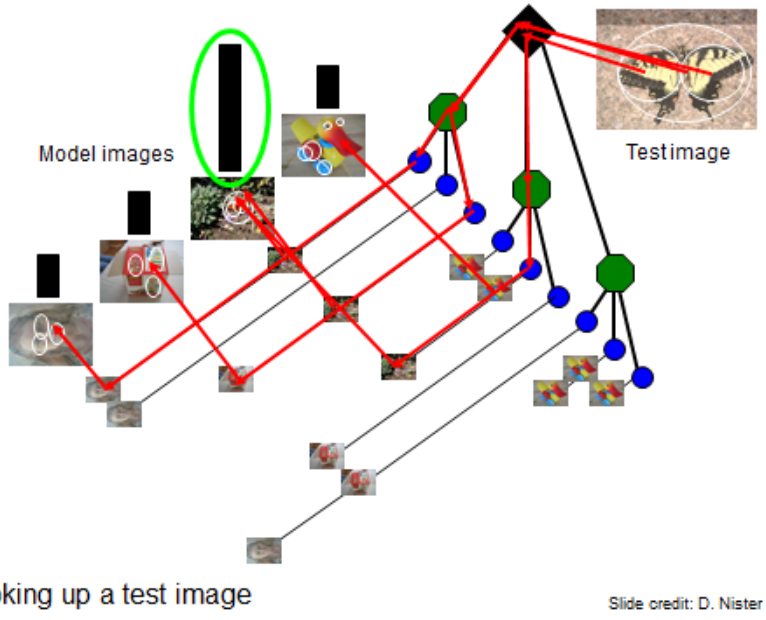

Figure B.4 Test image against model images

APPENDIX C

Aly et al.'s (2009) Discovery of Large Image Families

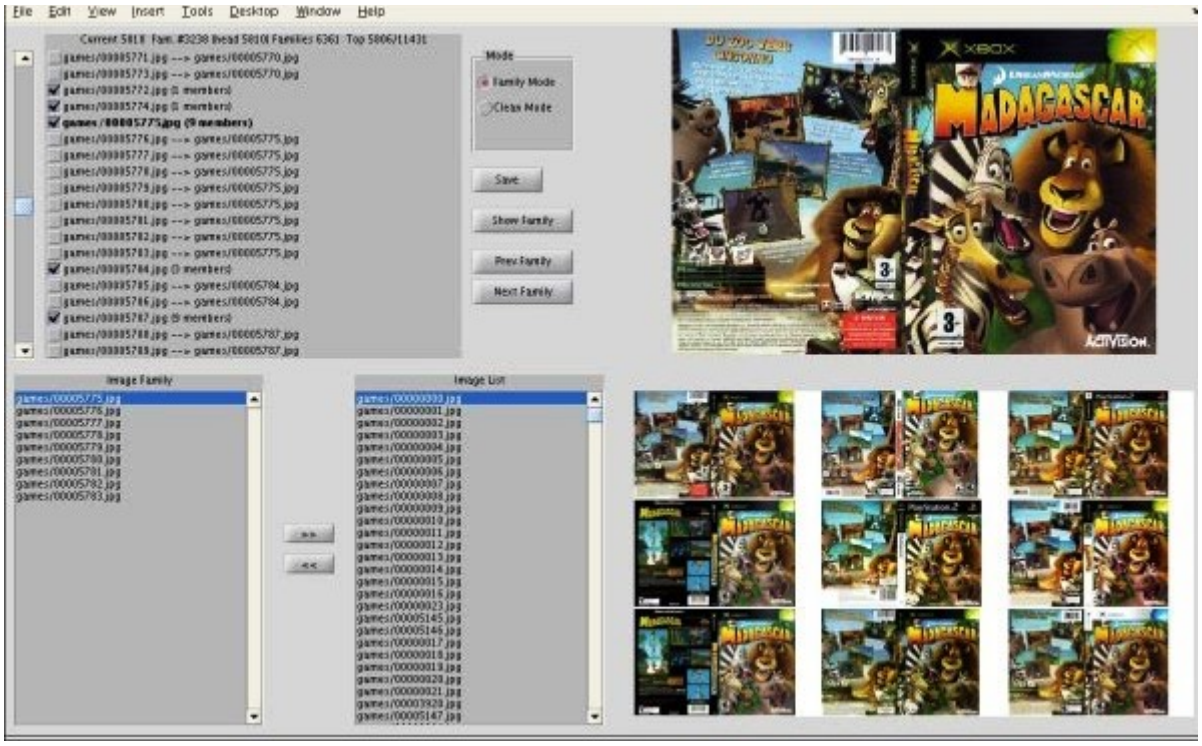

Figure C.1 The simple Matlab based image family viewer. (we tried it for this study, sadly, Matlab had problems on Windows, with the framework relying on Linux-based scripts/compilers for certain critical vision tasks)

Figure C.2 They clustered duplicate images

APPENDIX D

Semantic space and Segmentations

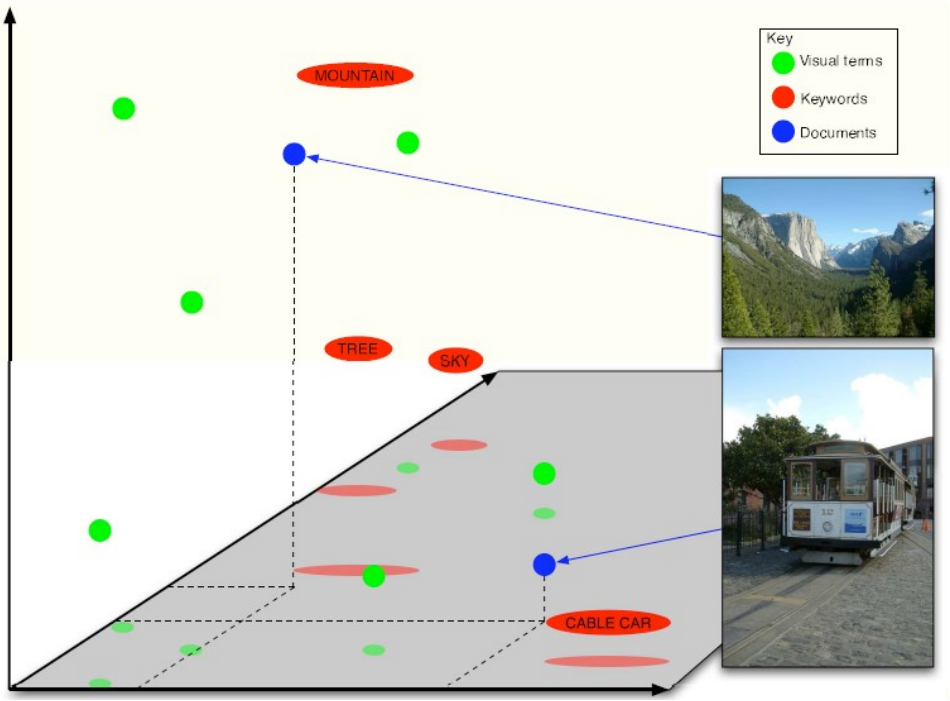

Figure D.1 An illustration of the semantic space

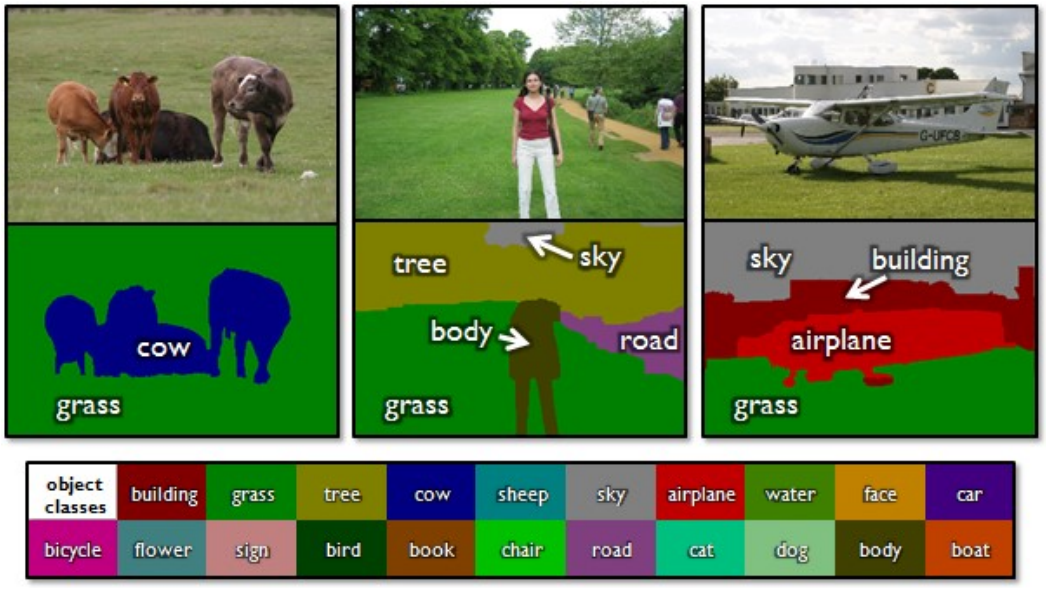

Figure D.2 Segmentations with labelling of co-occurring objects as Ground Truth for scenes.

APPENDIX E

The Parsing Graph

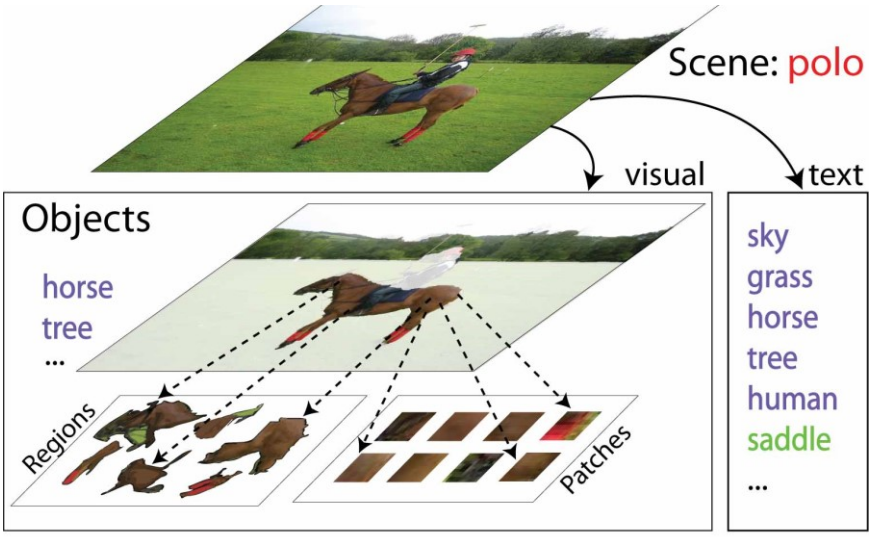

Figure E.1 Image parsing example. The parsing graph is hierarchical and combines generative models (downward arrows) with horizontal connections (dashed lines), which specify spatial relationship between the visual patterns.

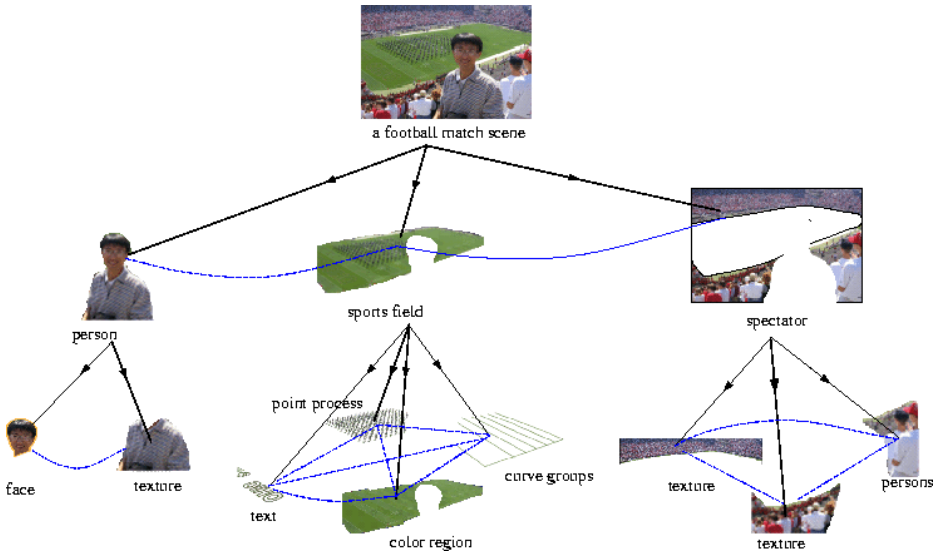

Figure E.2 The decomposition of a scene class using a parsing graph.

APPENDIX F

Human Actions and Scene Classes

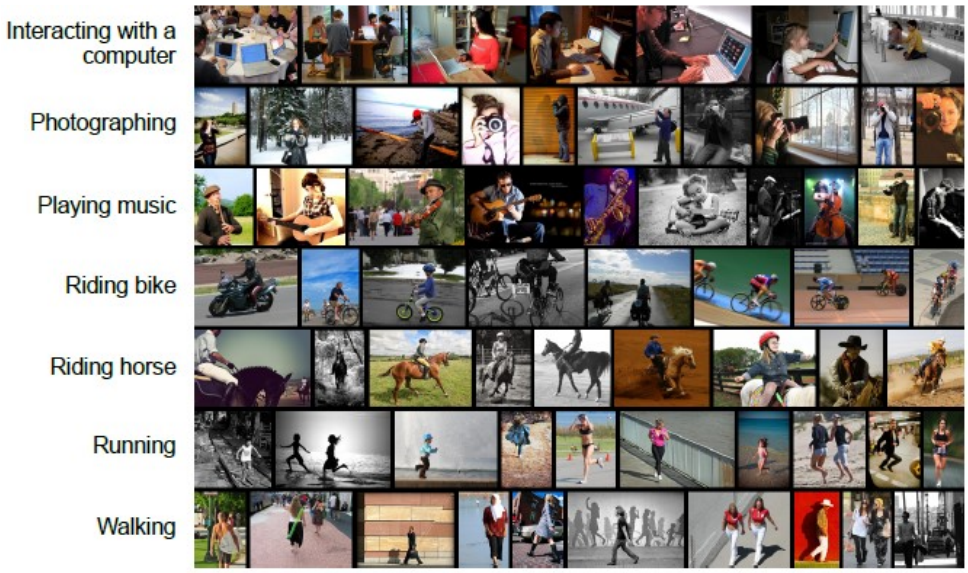

Figure F.1 Images showing the various types of actions or verbs that can be recognized.

Figure F.2 Images showing different scene classes

APPENDIX G

Textons

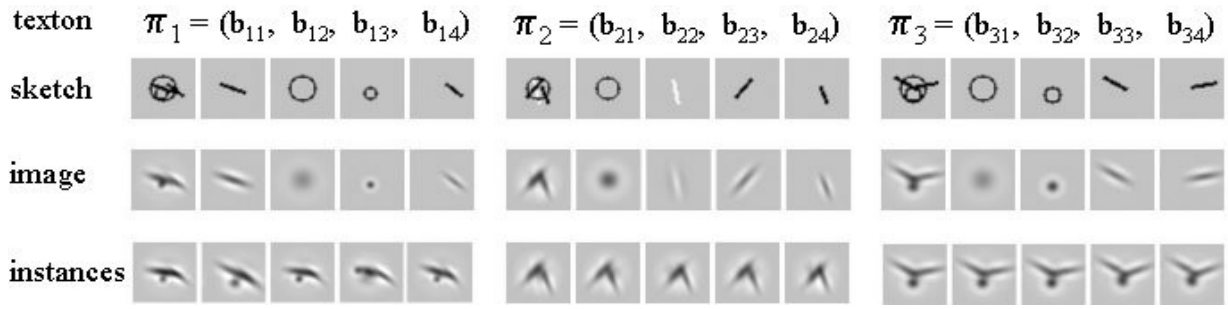

Figure G.1 The texton representation of a flying bird

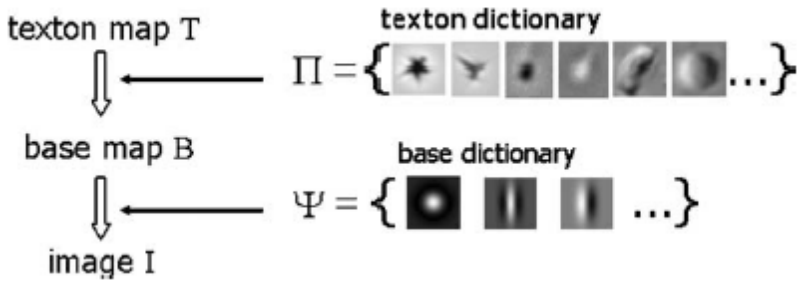

Figure G.2 A three-level generative model: an image **I** (pixels) is a linear addition of some image bases selected from a base dictionary. The base map is further generated by a smaller number of textons selected from a texton dictionary. Each texton consists of a number of bases in certain deformable configurations, for example, star, bird, cheetah blob, snowflake, bean, etc.