UNDERSTANDING IMAGES AND VIDEOS USING CONTEXT

by

GONZALO VACA-CASTANO
B.S. Pontificia Universidad Javeriana, 2003
M.S. University of Puerto Rico, 2010

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the Department of Electrical and Computer Engineering
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Fall Term
2017

Major Professor: Niels D. Lobo & Mubarak Shah

ABSTRACT

In computer vision, context refers to any information that may influence how visual media are understood. Traditionally, researchers have studied the influence of several sources of context in relation to the object detection problem in images. In this dissertation, we present a multifaceted review of the problem of context. Context is analyzed as a source of improvement in the object detection problem, not only in images but also in videos. In the case of images, we also investigate the influence of the semantic context, determined by objects, relationships, locations, and global composition, to achieve a general understanding of the image content as a whole. In our research, we also attempt to solve the related problem of finding the context associated with visual media. Given a set of visual elements (images), we want to extract the context that can be commonly associated with these images in order to remove ambiguity.

The first part of this dissertation concentrates on achieving image understanding using semantic context. In spite of the recent success in tasks such as image classication, object detection, image segmentation, and the progress on scene understanding, researchers still lack clarity about computer comprehension of the content of the image as a whole. Hence, we propose a Top-Down Visual Tree (TDVT) image representation that allows the encoding of the content of the image as a hierarchy of objects capturing their importance, co-occurrences, and type of relations. A novel Top-Down Tree LSTM network is presented to learn about the image composition from the training images and their TDVT representations. Given a test image, our algorithm detects objects and determine the hierarchical structure that they form, encoded as a TDVT representation of the image.

A single image could have multiple interpretations that may lead to ambiguity about the intentionality of an image. What if instead of having only a single image to be interpreted, we have multiple images that represent the same topic. The second part of this dissertation covers how to extract the context information shared by multiple images. We present a method to determine the topic that these images represent. We accomplish this task by transferring tags from an image retrieval database, and by performing operations in the textual space of these tags. As an application, we also present a new image retrieval method that uses multiple images as input. Unlike earlier works that focus either on using just a single query image or using multiple query images with views of the same instance, the new image search paradigm retrieves images based on the underlying concepts that the input images represent.

Finally, in the third part of this dissertation, we analyze the influence of context in videos. In this case, the temporal context is utilized to improve scene identification and object detection. We focus on egocentric videos, where agents require some time to change from one location to another. Therefore, we propose a Conditional Random Field (CRF) formulation, which penalizes short-term changes of the scene identity to improve the scene identity accuracy. We also show how to improve the object detection outcome by re-scoring the results based on the scene identity of the tested frame. We present a Support Vector Regression (SVR) formulation in the case that explicit knowledge of the scene identity is available during training time. In the case that explicit scene

labeling is not available, we propose an LSTM formulation that considers the general appearance of the frame to re-score the object detectors.

To my girls Andrea and Catalina, my parents Gonzalo and Dora, and my sister Paola.

ACKNOWLEDGMENTS

Thanks to Dr. Shah for letting me be part of his Lab. Thanks to Dr. Lobo for showing the light in the darkest moments. Thanks to all the lab–mates and ex–lab mates, I learned a lot during these years with you. Thanks to my parents for all that they did for me (now I understand everything). Thanks to my girls, you are everything in my life. I am very grateful to Dr. Linwood Jones, Dr. Wasfy Mikhael, and Dr. Paul Wiegand for serving on my committee.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

One of the main computer vision goals is to emulate the behavior of the human visual system. Given an image or video, it is relatively easier for a human to determine the type of scene, the objects that are part of the image, how they are related, and to have a general understanding of the visual content. Consequently, a lot of research efforts in computer vision have focused on the detection of objects and classification of the scene, and recognition of actions and activities.

With the arrival of deep learning methods, the object detection and scene classification problems achieved high levels of performance, showing impressive results on large datasets. In spite of the success of object detection and scene classification, researchers are still struggling with developing computers' comprehension of the content of the image or a video as a whole.

In computer vision, context refers to any information that may influence the way that images and videos are understood. Several sources of context have been identified by researchers, especially in relation to object detection in images. Local pixel content is likely the most common source of context; it deals with the information of surrounding pixels in the windows or Regions of Interest (ROI) of the target objects, to capture evidence about the background typically associated with the objects. 2D scene GIST is another source of context commonly used. It captures image statistics that gives a hint about the global scene, which could be a good cue to the existence of some type of objects.

Other examples of context used in image analysis are: the weather conditions including several measurable factors like sun direction, cloud cover, sky color, precipitation, or season; the photogrammetric context given by camera parameters like focal length, lens distortion, radiometric function; photographer cultural bias in generating the image ; geographic context such as GPS location (from image tags), terrain type, elevation, or population density. In these cases, the extraction of context is performed beforehand, therefore these sources of context are typically employed to build case specific applications.

In this dissertation we focus on two important sources of context. They are semantic context and temporal context. Semantic context is associated with: scene category, objects present in the scene and their spatial extents, keywords, and activity being depicted. Temporal context is related to temporal proximity of images, or nearby frames in the case of videos.

Making use of semantic context, we present a framework that allows modeling the image content, and learning about the image structure and its parts. The framework performs object detection using thousands of categories while simultaneously understanding the overall composition of the image by determining the relative importance of the objects in the image and identifying their hierarchies.

We also study the context from an opposite perspective. Given several images, the goal is the extraction of the shared context by these images. An image can have multiple meanings, but when several images are used in the same context, the intersected meaning of the group of images can clearly emerge as the common context. We present a framework where the semantic information of each input image can be leveraged to infer the topic that the visual information reveals.

As further exploration of context, this dissertation examines the case of video as the input source. Specifically, we deal with egocentric videos because they have smooth changes of the scene context and few changes in the collection of objects. The semantic and temporal context are used to improve the quality of the object detection, and scene identification.

1.1 Contributions

Firstly, we apply context to process images. To achieve this we introduce the following: a new image representation that encodes the location of the elements in the image and the hierarchical structure that they form; a parsing method to obtain the image representation from an annotated dataset; an algorithm to train and learn the visual structure of the images in the dataset; and an inference algorithm that outputs the proposed image representation including the detected objects.

Secondly, we present a general purpose tool to determine the common topic describing multiple images that are contextually related. This type of media source (multiple images) has not been used with this purpose before. We also present a new paradigm for image retrieval, where multiple images are used as input, the underlying common context is searched, to finally retrieve images that could relate the concepts being expressed by the input query images.

Lastly, we introduce mechanisms to exploit the temporal context in the case of videos. Concretely, we focus on egocentric videos performing activities of daily living (ADL), where we demonstrate that the contextual information leads to improvements in the results of scene identification and object detection.

In the following subsections, we provide details of our specific contributions employing context information to improve the semantic understanding from single images, multiple images and video.

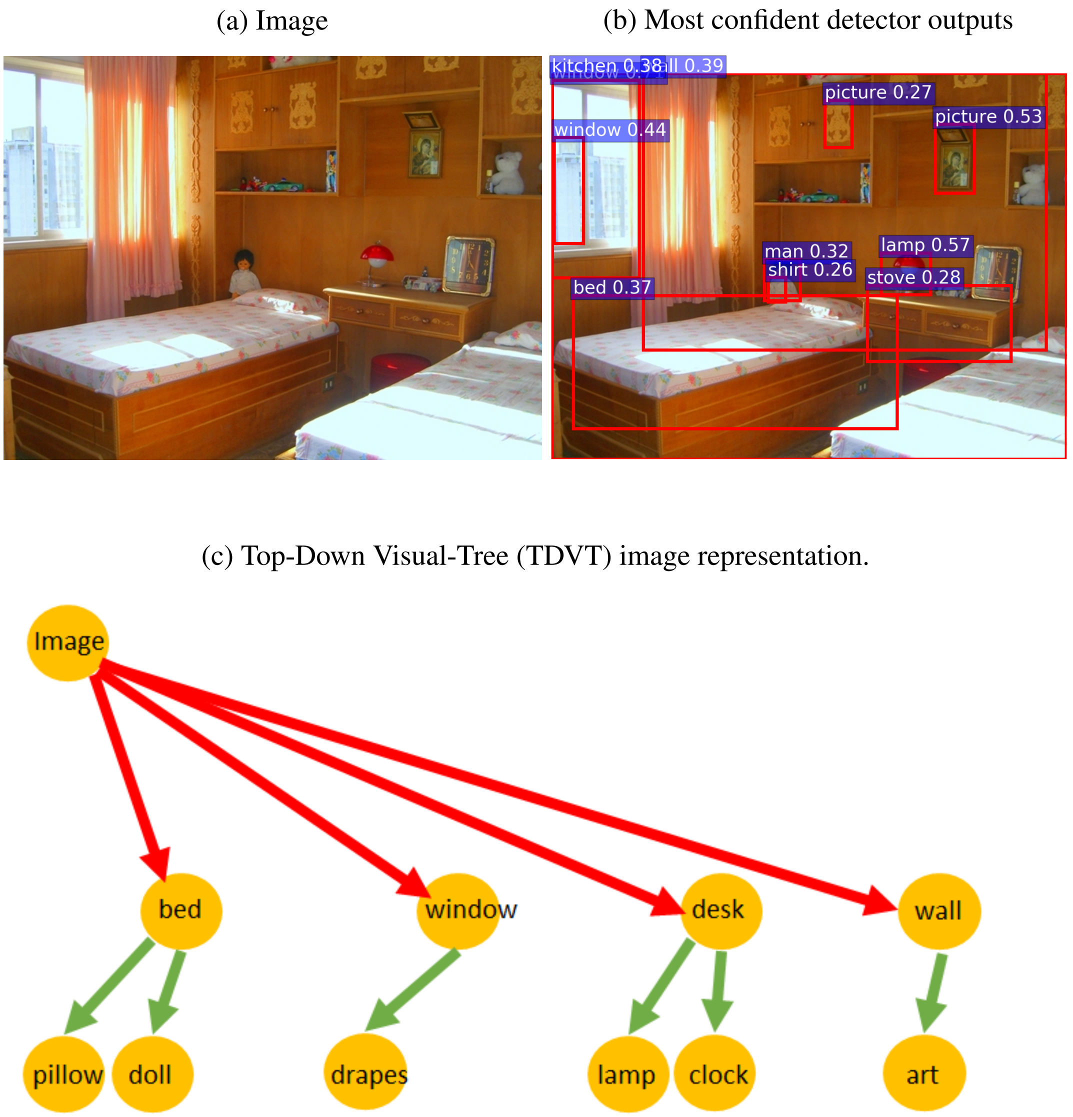
1.1.1 Context in Single Images

Humans derive a great deal of information about the world through their visual sense. Vision accounts for two–thirds of the electrical activity of the brain when the eyes are open [1]. In spite of the recent success in computer vision on individual tasks such as image classification, object detection, image segmentation, and the progress on scene understanding, researchers still lack clarity about computer comprehension of the content of the image as a whole. While most recent efforts have involved the use of language to achieve comprehension [2, 3, 4, 5, 6, 7, 8, 9, 3, 10], this dissertation presents a pure visual representation of the image content that allows an understanding of the image through a unified framework for object/stuff detection, and representation of the discovered objects, their relations, and their relative importance.

The object detection problem focuses on identifying a particular object from multiple categories. Given a set of window proposals, a multi–label classifier determines scores for the different types of objects and the background in the selected region. Each category competes with others for the highest score that determines the label of the region. In scenarios constrained to a few categories (e.g., less than 200) the described approach works reasonably well; however, in more realistic scenarios, a larger number of categories is necessary in order to have a better understanding of the diverse content of images. This increase in the number of categories results in many simultaneous possible valid detections, which makes scene interpretation more difficult. In addition, many categories with weak detectors do not generate high scores, even in scenes where the global and local context strongly suggest the opposite. An example is shown in Figure 1.1b where object detections for most prominent classes are displayed. Neither the “drape” nor the “pillow” are detected directly, however these objects could be detected using the context information from the detection scores of the nearby objects “window”, and “bed”, which are generally highly correlated with the un-detected objects. Hence, the object detection problem with multiple categories will benefit from the incorporation of contextual cues such as object co–occurrence, and overall image consistency.

Images contain more information than just objects. A typical image has a hierarchical structure that relates objects to each other. Starting from the higher level that represents the full image, there are elements that are more prominent or meaningful from a visual and/or semantic point of view. Similarly, each one of these elements may also be related with other elements that have lower relative importance in the image.

In the case of Figure 1.1a, the elements in the scene in order of their importance are “bed”, “window”, “desk”, and “wall”. Each one of these objects has associated other less relevant elements. The “pillow” and the “doll” are associated with “bed”. The “drape” is associated with the “window”. Similarly, there are some “pictures” associated with the “wall”. Figure 1.1c depicts a proposed tree that captures the inherent hierarchical structure of Figure 1.1a. In this dissertation, we propose a Top-Down Visual-Tree (TDVT) image representation that encodes the location of the elements in the image and their hierarchical structure. The proposed representation also encodes the order of importance of the elements and defines their type of relations.


(a) Image

(b) Most confident detector outputs

(c) Top-Down Visual-Tree (TDVT) image representation.

Figure 1.1: a) An image , b) the most confident detector outputs, and c) the TDVT image representation of its content.

As a pre-processing step, we show how to automatically obtain a Top-Down Visual-Tree (TDVT) image representation from object annotations through an image parsing process that operates on available image datasets (with annotations). Then, in the training process, we learn models for object detection and use a novel Top-Down Tree LSTM network to learn the hierarchical structures associated with images from their TDVT image representation. During testing, given an input image, our method generates a rich set of outputs, beyond the object locations. In particular, we are also able to predict the structure of the image, and identify its main salient elements; recognize which objects are parts of others, and identify relationships between objects.

The inference process finds un-detected elements that traditional object detection is unable to capture, since our approach incorporates the image context during object detection.

The contributions of the chapter 3 are summarized as follows: 1) a new TDVT image representation that captures hierarchical structure, importance of objects and type of dependencies, 2) a parsing method to obtain the image representation from an annotated dataset, 3) an algorithm to train and learn the visual structure of the images in the dataset, 4) an algorithm to perform inference from an image to detect the objects, their relative importance, and image's hierarchical structure.

1.1.2 Context Discovery From Multiple Images

In Chapter 4, we present a solution to the inverse problem of using multiple images to identify the context shared by these images. In the remaining of the thesis, we refer to the shared context as the topic. A section is dedicated to solve the problem of finding the topic, where the topic is represented as a set of words that properly described such topic. Later, we propose a new paradigm of image retrieval where multiple images are used as a query, and present a solution based on the discovery of topics associated to these images. The details of the two sections are presented next.

1.1.2.1 Finding the Context Shared by Multiple Images

We propose a novel method to infer the shared context that a group of semantically related query images may potentially be related to. Given a set of images that are related to a particular topic (shared context), we are interested in finding a set of words that properly describes such a topic. Knowing the topic in an unconstrained, unsupervised scenario is important because it permits a system to delimit the search space for locations, objects, intentions, and context knowledge. A good application of this is illustrated in the following example. Consider someone looking for ideas for a gift, while they are walking through the mall. This person may take pictures of related gift ideas; then, based on the provided images, semantic concepts are found and used to retrieve a textual set of gift suggestions.

A topic is represented as a collection of the top ten most descriptive words associated with the subject depicting the images. The main goal of the novel proposed topic discovery method is to select the best set of words that represents the main topic. In order to achieve this goal, visual information of each query image is used to retrieve similar images with text labels (tags) from an image database. We consider a scenario where the tags are noisy and diverse. The words or tags associated to each query are processed jointly in a word selection algorithm that refines the search topic, rejecting words that are not part of the topic and producing a set of words that appropriately describe the topic.

This work makes the following contributions: 1) a new problem of determining the topic described by a set of images, 2) a word selection algorithm that chooses a group of words that are strongly conceptually related, and 3) a novel mapping algorithm that maps words semantically related to a different set of words from a smaller vocabulary, preserving the semantic relation of the transformed words.

1.1.2.2 Image Retrieval From Multiple Images

We present a new image retrieval paradigm that uses multiple images as input to perform semantic search. Most of previous work in image retrieval concentrates on using a single query image. In the few cases where multiple query images are used as input in image retrieval, these query images corresponds to different views of the same content, then the retrieved images are simply new instances of the input.

We explore a different approach for dealing with multiple images as input, since the input images are employed jointly to extract underlying concepts common to the input images.

The retrieved images have two desirable properties. Firstly, the images are *conceptually* related to the query images, and secondly the images are also related *visually* to the query images since a re-ranking of the retrieved images is performed based on visual features.

This new image retrieval paradigm has the following novelties: 1) Multiple image queries are used as inputs. 2) The input images are used to retrieve the concepts that images represent, instead of merely visual similarities. 3) Text descriptors are used as operands to discover underlying concepts common to the input images. The retrieved images capture semantic similarity from knowledge gained via text concepts and visual similarity.

1.1.3 Improving Understanding of Videos Using Context

In chapter 5, we concentrate on the use of context to improve scene and object detection from videos. We study the temporal context to exploit inherent egocentric constraints of individuals performing daily activities. In the case of activities of daily living, the actions typically are performed in common places associated with human residences such as bathroom, corridor, patio, kitchen, among others, which will be referred as the scenes.

We are interested in the frame level scene identification problem, where the goal is to find the correct scene identity for all the frames of the egocentric video. We note that temporal constraints can be exploited to improve frame level scene identification performance. The location where an activity is performed remains consistent for several frames until the user changes his/her current location. Given a frame, several trained scene classifiers are evaluated and a decision about the identity is made based on the classification scores. However, the scores obtained for individual frames can lead to wrong scene identification, since these scores are agnostic with respect to the temporal constraints associated with egocentric vision. Therefore, we propose a Conditional

Random Field (CRF) formulation that uses the scene identification scores of temporally adjacent frames to improve the scene identity accuracy.

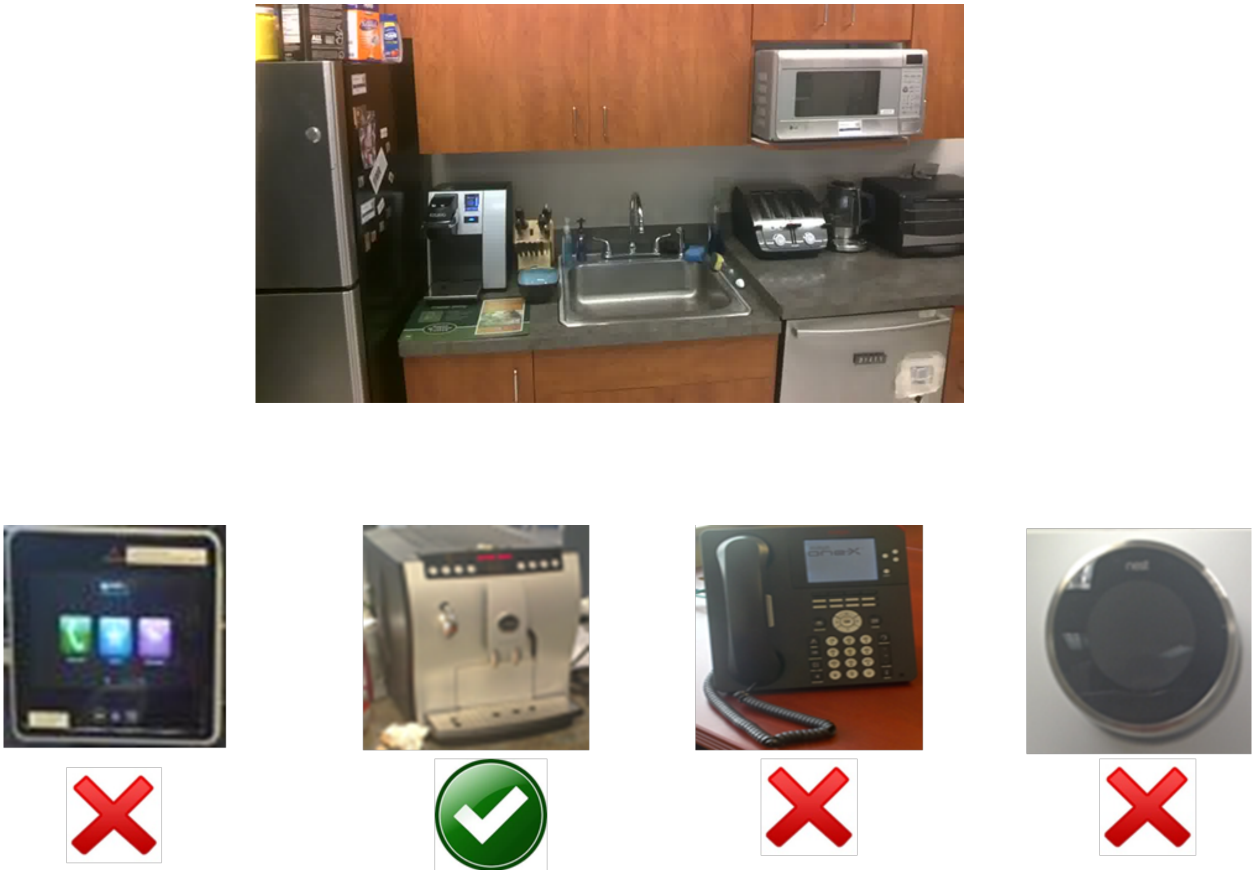


Figure 1.2: Example of how object detection is influenced by the scene context. Figure on top contains an image taken in a kitchen. Figures on bottom show a list of possible objects that could be detected. From the list, only the coffeemaker makes sense in the observed context.

In addition, we propose a method to improve object detection. The object detection task attempts to find the location of objects in a frame. Traditional approaches use human labeled bounding boxes of objects as positive training data, while visual features not included in the positive training bounding box are part of the negative data. However, in the real world, the objects are part of a scene. Consider, for example, the top image in Figure 1.2 of a kitchen. Images at bottom of Figure 1.2 show a list of possible objects that could be interesting to detect. It is obvious for humans that some types of objects are unlikely to be found in the observed scene, while a coffeemaker is an object that most likely can be found in this type of scene.

The previous observation is used as a constraint in our problem formulation to improve the quality of object detectors. We concentrate on Activities of Daily Living (ADL), where most of the egocentric activities are performed in few prototypical scenes that are common to all the actors. ADLs are an extremely challenging scenario for object detection, since the objects suffer from notable changes in appearance due to radial distortion, pose change and actor influence over the object. We do not focus on direct improvements in the object detector. Instead, the results of object detection are improved after re-scoring the outcome of the object detection method. Objects that are most probably present in a type of scene get higher scores, while objects that are unusual in a type of scene get lower scores. We present formulations. Firstly, we address the case where the labels of the test videos are explicitly predicted from scene models learned in training data. For this problem, we initially present a conceptually simple algorithm, a greedy formulation; then as an improvement we propose a more accurate Support Vector Regression (SVR) based algorithm to solve the same problem. Secondly, we address the case where we do not have an explicit knowledge of the label of the scenes. For this case, we propose a formulation based on Long Short-Term Memory (LSTM), that directly infers the probability of having a type of object in a sequence.

The main contributions of the chapter are summarized as follows: 1) we propose the use of temporal consistency constraint to improve scene identification accuracy in egocentric videos. 2)

We present two algorithms to improve the object detection results, that relies on updating the object detection scores according to the scene content of the frame currently tested.

1.2 Organization of the Thesis

The thesis is structured as follows: Chapter 2 reviews existing literature, and focuses on analysis of context in computer vision, as well as related problems like object detection and image retrieval. Chapter 3 proposes a new image representation and a complete framework to infer such representation from images. Chapter 4 presents our method to discover the common topic in multiple images, and the proposed application that uses the context to perform image retrieval. Chapter 5 shows a method to use temporal context to improve scene identification, and two algorithms that use semantic context to improve object detection in egocentric videos.

CHAPTER 2
LITERATURE REVIEW

An important unsolved problem in computer vision remains image and video understanding. Given an image or a video of the real world, the final goal of a computer vision system is to determine what visual elements and structures are presented, how these elements are related to each other, and to have a complete understanding of what is happening in the visual input. Visual understanding is difficult to define and evaluate, therefore researchers have concentrated on solving more focused, specialized, low-level problems like object detection or scene identification. Object recognition does not occur as an isolated process since, it can be influenced by the presence of other objects as well as by the overall context of the scene. Scene context provides a rich source of information that can help to improve the performance of the recognition task. The main objective of this dissertation is to study the influence of context (semantic and temporal) on scene understanding in images and videos, as well as the implicit context shared by multiple images.

In this chapter we present a literature review of the major topics associated with this dissertation. We have divided this literature review into three major sections resembling the main chapters of this thesis. The first section contains a detailed description of previous attempts to use visual context for image understanding. Most research efforts in image understanding has been focused on developing algorithms for object detection. Therefore, a subsection is dedicated to review the literature on object detection. We also devoted a subsection to review existing image representations, since they are a key element to model and understand the visual content.

The second section concentrates on finding common context from multiple images. The use of multiple input query images has been relatively un-explored by the computer vision community. The small amount of work that has been done, has been limited to multiple views of same scenes or objects. Since this dissertation presents a new paradigm for image retrieval from multiple images, a short review of image retrieval is presented as a subsection.

Finally, a short literature review is provided about analysis of egocentric videos of Activities of Daily Living (ADL) and the use of temporal context to improve object detection. Since, we also improve the scene detection results at frame level, we dedicated a subsection to cover literature review for scene identification.

2.1 Visual Context and Image Understanding

The role of context in object recognition has been analyzed from a cognitive science perspective by several authors[11, 12, 13, 14], leading to a general consensus that objects in consistent or familiar background are detected more accurately and processed faster than objects appearing in an inconsistent scene. The subject has also been explored from a computer vision perspective [15, 16, 17, 18, 19, 20, 18, 21, 22, 23], specially associated to the object detection problem and prior to the emergence of neural networks as the dominant computer vision tool. Galleguillos and Belongie [15] presented a survey that identifies three types of context: semantic, spatial, and size context;

this acts at two levels: global and local; and this shows two mechanisms of integration of context information. They demonstrated that contextual information can help to successfully disambiguate appearance inputs in recognition tasks. Heitz and Koller [20] used a terminology coined by Forsyth et al. [24] known as TAS, “thing” and “stuff”, linking discriminative detection of objects with unsupervised clustering of image regions. In the work of Torralba et al. [19], the global scene context and its influence over object recognition is considered by representing the scene as a low-dimensional global image representation (GIST), and this is used as contextual information to introduce strong priors that simplify object recognition. Torralba [22] also employed a probabilistic framework for encoding the relationships between context and object properties.

Divvala et. al [16] presented an evaluation of the role context plays in the object detection task using the Pascal VOC 2008 dataset. Understanding the context as any information source that may influence the way a scene and the objects within it are perceived, ten different sources of context were identified, and experiments were performed with a subset of context sources: local pixel context, 2D scene gist , 3D geometric, semantic, geographic, photogrammetric, and cultural cues. A Bayesian formulation for the object location, size, and presence, and their combinations, was used to improve the object detection based on these cues.

The work of Choi et al. [17] introduced the SUN dataset, and presented a Tree-structured contextual model. Their model is a single graph that represents positive and negative co-ocurrences per dataset, capturing generalities about each dataset. Song et al. [18] proposed an iterative contextualization system, with the objective of boosting object detection and image categorization by taking the outputs from one task as the context of the other one using a SVM formulation and dynamically adjusting the classification hyperplane. The hyperplane formulation is only used for ambiguous samples where context is helpful.

The role of local context in object detection using deep neural networks seems less important than in traditional object detection, presumably due to longer field of view covered by the convolutions. Only very recently, context was shown to have a positive impact for object detection using deep neural networks [25, 26, 27, 28]. Gidaris and Komodakis [26] presented a multi-region deep CNN that selects different regions around the ROI to obtain a more robust feature vector. They improved object detection results on the Pascal VOC dataset. Similarly, Bell et al. [25] form a descriptor that includes vectors from other multiple layers. They used spatial Recurrent Neural Networks (RNNs) to gather local contextual information from above, below, left, and right of the object. Similarly, Li et al. [27] proposed a network that exploits contextual information for object detection using two sub-networks. An attention-based sub-network allows to use a global context view, while a multi-scale sub-network captures local context by pooling descriptors from three bounding boxes scaled around the ROI. Their results were reported on the PASCAL VOC dataset.

Until now, we have shown different approaches to exploit the context to improve the object detection on images. In the next subsections, firstly we will review the literature for object detection. Later, we will review previous attempts to represent images.

2.1.1 Object Detection

The object detection problem has been one of the core problems in computer vision because objects are a key building block of any image. A decade ago, object detection technology, started to solve real problems with the finding of visual features like the Histogram of Gradient (HOG) that, in conjunction with Support Vector Machines (SVM) provided the reference method [29] for the period from 2005 to 2008. The Pascal VOC challenge [30] was the venue for object detection research from 2005 to 2012. An extensive analysis of the results of the different competitions on PASCAL VOC challenge during years 2008 to 2012 was published [31] by their organizers. With a dataset of twenty different object classes, and up to approximately five thousand images, the competition was dominated by methods based on Deformable Part Model (DPM) [32]. The DPM model uses HOG to describe a coarse scale 'root' filter and a set of finer-scale part templates that can move relative to the root. In testing time, the model is applied everywhere in the image (and different scales) using sliding windows. The leading method in 2010 by National Laboratory of Pattern Recognition from the Chinese Academy of Sciences, used a combination of Local binary patterns (LBP) and HOG as features over the DPM model. In 2012, the leading method from University of Amsterdam [33] used a combination of more complex features and selective window search to improve performance.

The availability of massive image datasets [34, 35] and increase computational speed up, permitted deep learning techniques to emerge, becoming the current dominant paradigm used by object detection methods[36, 37]. The number of available training images in the new datasets increased to hundreds of thousands, and the number of categories to detect was also increased. In the Large Scale Visual Recognition Challenge (ILSVRC) 2016 [1] the classes went up to 200 categories, while the COCO dataset [2] contains 80 categories. A huge gain in performance was achieved in 2014 by Girshick et al. [38] using a combination of selective search and Convolutional Neural Networks (CNN). In that work, the Convolutional Neural Network trained by Krizhevsky et al. [39] for the ImageNet (ILSVRC) classification challenge was used, but a fine tuning in the fully connected layers of the network was performed in order to adapt domain to the PASCAL VOC dataset. A comprehensive evaluation of the ILSVRC challenge can be found at [40]. The report describes the huge improvements obtained in classification and detection challenges (almost double in less than 2 years) after the generalized used of Convolutional Neural Networks (CNN).

The leading team of the ILSVRC 2013 challenge in single-object localization was OverFeat [41], which was based on an integrated framework for using convolutional networks for classication, localization and detection with a multiscale sliding window approach. The winner of object detection task was UvA team, which utilized a new way of efficient encoding [42] of densely sampled color descriptors [43] pooled using, a multilevel spatial pyramid in a selective window search [33]. The detection results were re-scored using a full-image convolutional network classifier. By 2014, all the teams were using CNNs including: GoogLeNet, the winning team of the classification and detection with extra data task using a deep convolutional Neural Network that uses a new multiscale network named inception; VGG, the winner of the single–object localization with

[1] http://image-net.org/challenges/LSVRC/2016/

[2] http://mscoco.org/

provided data track (using a very deep CNN with 16 –19 layers); and NUS, the winner of object detection with provided data track (using RCNN [38] with the network-in-network method [44]).

The core idea of the current dominant algorithms for object detection is the use of Regions of Interest (ROI) that could contain an object, and evaluation of each region to determine if it belongs to a class or to the background. Hence, a detection network typically contains a stack of Convolutional Neural layers (computed once per image), an ROI pooling method that extracts features for each ROI, and a classification network that determines the label of the region. An ROI can be generated externally by a generic object proposal method [33, 45]. Most recently, the ROI generation process was integrated in the system [37], saving extra computations by re-utilizing the computed CNN features. Further performance improvements has been achieved mainly by changes in the network structure toward the use of deeper networks [46, 47].

2.1.2 Image representation

A representation is needed in order to interpret the overall image content . Image representations in computer vision are typically based on numerical vector representations. Some representations ignore the spatial layout information like Bag of Words (BoW) [48] or Fisher vectors[49], and others use grid structure representations such as GIST-based representation [50, 51] and spatial pyramid matching (SPM) [52]. Vector representation is appealing since it can be processed using machine learning to solve specific task such as image retrieval, image classification, object detection, action detection, among others. However, these type of representations lack the understanding at the semantic level. Approaches for scene modeling like [53, 54] attempt to learn distinct scene configurations. They have been successful only on small type of scenes and label categories since they rely on scene templates. For instance, the authors in [53] use only 8 types of outdoor categories and 33 semantic labels that are finally merged to 12. Hence, these scene modeling initiatives are hard to scale up. The progress in object detection technology have shifted the challenge from tens of possible object categories to even thousands of them.

Recently, image representations were enriched with the use of Natural Language Processing(NLP), achiving impressive results in problems like image captioning [2, 3, 4, 5, 6, 7, 8] and visual questioning answering[9, 3, 10]. The common trend in these methods is that they are based on directly using language to fill the visual semantic gap. Studies about the way people think in terms of their "inner voice" conducted in people born completely deaf [55] , have revealed that, if these persons only learned sign language, they will think in sign language. There is no doubt that language is needed in order to have an abstract thinking and self-awareness. We believe, however that a pure visual level representation of the image content is still missing in the current NLP trend, because it jumps directly from objects and global descriptors to the language, without a notion of the overall content of the image. Hence, in this dissertation, we propose a novel and purely visual hierarchical representation that allows modeling the image content, and learning the image structure from massive datasets without the use of spoken language.

2.2 Context Discovery from Multiple Images

The use of multiple images as queries has been relatively unexplored by computer vision researchers. Some authors [56, 57, 58] have focused on the image retrieval problem from multiple query images, where query images correspond to instances of the same object under different viewing conditions, and the retrieved images are simply the most similar instances to the queried object.

In [56], text is used as an input to retrieve a set of images of a specific object that are used to retrieve other images of the same object using five different methods. The five methods are based on re-ranking of the images retrieved from a database using an online trained model. To train the model, the set of images initially retrieved is labeled as positive, while a set of randomly selected images of the database is labeled as negative. Basura et. al [57] also deal with the problem of object retrieval starting from multiple query images. In their formulation, they derive the most suitable set of patterns to describe the query object, where patterns correspond to local feature configuration. Hsiao et. al [59] selected the retrieved images using a Pareto front method.

In all of the above cases, there is no notion of semantics or topic selection as in this dissertation. We believe that our work is the first to show how to use multiple query images to find the shared context. Since we query images to obtain text labels, there is some relation with research efforts in the area of automatic image annotation [60, 61]. However, these efforts have been conducted on datasets with only a couple thousand images, and less than 300 human labeled tags, compared to our approach where we employ more than a million images and an open vocabulary, which is not limited to a few hundred tags.

In the next subsection, we review the literature on Image Retrieval, since in this thesis, we also propose a new paradigm for image retrieval.

2.2.1 Image Retrieval (IR)

The goal of any Image Retrieval system is to retrieve images from a large visual corpus similar to the input query. Image retrieval systems are commonly characterized by two types: Concept-based image indexing, or Content-based image retrieval (CBIR).

Concept-based image indexing, sometimes named "text-based" or "description-based" image indexing/retrieval, refers to retrieval from text-based indexing of images like tags, keywords, captions, or natural language text . Getting annotations of training images is time-consuming, expensive, and subjective. For that reason, most of the research efforts in this area have focused on automatic image annotation [60, 61]. These efforts have been conducted in datasets with only a few thousand images, and less than 300 tags. Given the limited number of tags and the difficulty to scale this number up, concept-based image indexing is less popular than CBIR.

Content-based image retrieval (CBIR) retrieves images directly from visual features of the image [62] such as colors, shapes, textures , rather than metadata like tags, keywords, or descriptions associated with them. CBIR can be roughly classified into two kinds of algorithms. In the first near-duplicate images are searched either by means of local feature indexing [63, 64, 65, 66] or hashing of global features like GIST [67, 50]. In the second category, images of the same class are retrieved by using multiclass classifiers of objects or attributes [68, 69, 70, 71, 72, 73]. Recently, the approach by Zhang et.al [73] improved the near-duplicate image retrieval by encoding these two distinct cues, local features and attributes, together in the inverted indexes.

To date, most Image Retrieval systems (including commercial search engines like Google [3]) base their search on a single image input query. Recently in some works, multiple images as input queries have been proposed for image retrieval [57, 56]. In these cases, the objective of the multiple image inputs is to acquire different viewing conditions of the unique object or concept that the user is searching for.

2.3 Scene Identification and Object Detection on Videos

Recent efforts [74, 75, 76] in egocentric vision have focused on object recognition, activity detection/recognition and video summarization, however none of these efforts have focused on scene identification and its relation with object detection.

Ren and Philipose [74] collected a video dataset of 42 objects manipulated by hands in an object-specific way, and quantified the accuracy drop of object detectors after simulating background clutter and occlusion on clean exemplars. Fathi et al. [75] observed that objects of interest tend to be centered and cover a large space of the image frame. Based on that observation they perform unsupervised bottom up segmentation and divide each frame into hand, object, and back-

[3] http://images.google.com/

ground categories. A list of objects that are part of the video is provided, and an appearance model for them is learn from the training dataset. Objects become part of the background after manipulation is completed.

In [76], a new dataset of activities of daily living (ADL) in egocentric videos is presented. The dataset contains annotations for every second of the object bounding boxes of 42 different objects, and also the results of a DPM based object detector for some of the objects. Models trained in standard datasets like Imagenet produced poor detection results, because conventional datasets contains only iconic view of the objects compared to the more challenging appearance of objects from egocentric videos. The object detection models were trained using egocentric videos from a subset of the dataset. Many of the classes with available ground truth were not reported because the success rate was negligible.

In spite of the significant performance gains of these methods for single image object detection, these methods under-perform on video object detection due to multiple factors such as motion blur, temporary occlusions, objects out of focus, among others. In this thesis, we are interested in improving the results of object detectors on sampled frames using scene context. If better object detectors are available, the approach of using tracking by detection of the Multiple Object Tracking (MOT) problem, could be incorporated to obtain better tracks and handle long-term temporal relations. Different MOT algorithms [77, 78, 79, 80] use object detections on the input video frames and generate target tracks by connecting the detection outputs corresponding to identical objects across frames. The main difference among MOT trackers is that they employed detection-association mechanism. MOT does not overlap with the proposed mechanism in this dissertation to improve object detection, but it is in fact, complementary. Therefore, we will not focus on the MOT problem in this dissertation.

Recently, Han et al. [81] proposed a heuristic method for re-ranking bounding boxes in video sequences, linking bounding boxes temporally that have a high overlap from frame to frame. They achieved the third place in the video object detection (VID) task of the ImageNet Large Scale Visual Recognition Challenge 2015 (ILSVRC2015). Unfortunately, this approach assumes the availability of detections that overlap in every frame. This condition can only be achieved with high sampling frame rates that are prohibitive in long sequences, such as ours. In contrast, we process frames sampled approximately every second. Additionally, the mentioned approach does not consider the scene context associated with the objects in the frame.

2.3.1 Scene Identification

The scene identification problem is essentially an image classification problem with a domain specific type of images. During many years, approaches based on Bag of Words paradigm [82, 83] were the dominant state of the art. Further improvement was achieved by including spatial information using pyramids [84, 52] in association with new types of encoding [85, 86, 87, 88]. Huge improvements have been obtained in classification and detection after the generalized use of Convolutional Neural Networks (CNN). Most of these new approaches are based on extension of the CNN trained by Krizhevsky et al. [39] for the ILSVRC classification challenge. A number of recent works [41, 38, 89, 90] have shown that CNN features trained on sufficiently large and diverse datasets, can be successfully transferred to other visual recognition tasks such as scene classification and object localization, with a only limited amount of task-specific training data. The work of Gong et al.[91] is used as the reference method in this dissertation for scene classification. This method concatenates global CNN features and multiple scale levels CNN features pooled by orderless VLAD.

CHAPTER 3
IMAGE REPRESENTATION AND UNDERSTANDING USING TOP-DOWN TREE LSTM

Image-based scene understanding is the task of getting a computer to automatically interpret the contents of an image. The standard framework for scene understanding in computer vision models where the ground and/or support planes are, where the objects are, and the nature of the geometrical/spatial relations between the objects and planes. An open research question is, how to provide a hollistic description that includes the presence of the elements seen, and their visual and structural relationships.

In this chapter, we take steps in that direction. We present an image representation that encodes the visual content as a collection of objects/elements that are organized in a hierarchical manner. We first give details about this representation and the intuition behind it. Our Top-Down Visual Tree (TDVT) image representation allows the encoding of the content of the image as a hierarchy of objects capturing their importance, co-occurrences, and types of relation. A detailed description of the representation and several examples are presented in section 3.1.

Figure 3.1 shows how the important parts of this chapter fit together.

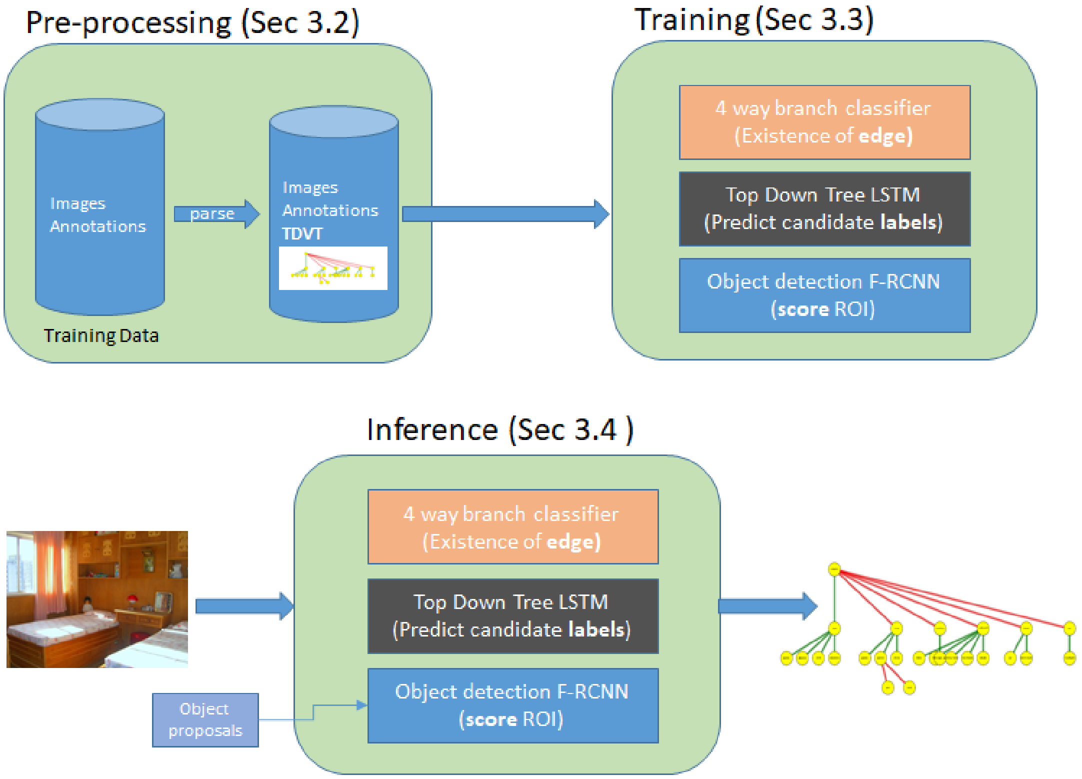


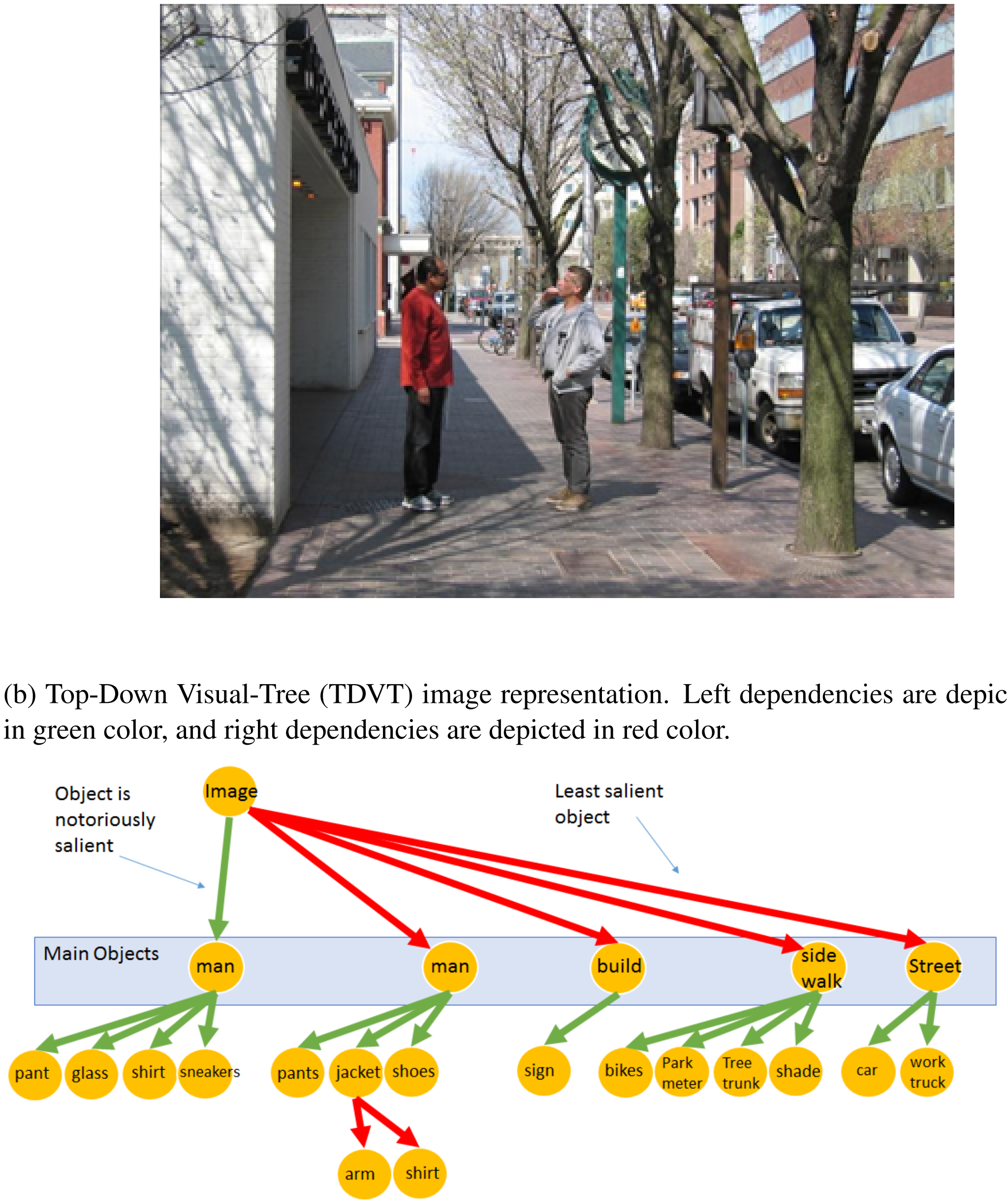
Figure 3.1: Graphical description of the content of the chapter.

Section 3.2 is dedicated to pre-processing images and their annotations from the training dataset to obtain TDVT representations that can be used in the training phase. Then, in section 3.3 we describe our training process which learns the hierarchical structure of the images. The training process utilizes training images and the TDVT representations obtained from the procedure described on section 3.2. Subsection 3.3.1 includes details about the Top-Down Tree LSTM network which predicts the most likely label of a node associated to an existing node and an edge. Subsection 3.3.2 explains the 4-way branch classifiers required to predict the possible existence of edges given the current state of the tree. In section 3.4 we provide details about the inference process that detects objects and the hierarchical structure that they form in images. In section 3.5 we present several qualitative examples and a comprehensive evaluation using the visual genome dataset. We also show that the presented framework is general and it improves object detection consistently across datasets.

3.1 Image Representation

Representations are important in order to model information about the world in a form that an automated system can utilize to solve complex tasks. Consider the image in Figure 3.2a. If someone is asked to describe the content of the image, a reasonable description would be something like this: “there are two men on a sidewalk next to a building and the street ”. This description is an example of a general idea which is that images can be described by the objects present and their relationships. Beyond this, each one of the described entities can also be further described by their constituent parts and relations with other entities. For example, the man on the left side wears pants, a shirt, glasses and sneakers. Hence, we observe that an image can be described as a hierarchical structure of related objects.

A natural choice for a data structure to represent hierarchical constructions is the tree. A tree is a collection of nodes starting at a root node, where the nodes are connected to other nodes through edges, without having any cycle (sub-tree children have only one parent node). Every node contains associated information. In the case of an image, we represent every node as a possible object/stuff/attribute and its associated information. The information associated to each node includes the bounding box location within the image, and label (class) of the object . The tree's root node corresponds to the full image.



(a) Image

(b) Top-Down Visual-Tree (TDVT) image representation. Left dependencies are depicted in green color, and right dependencies are depicted in red color.

Figure 3.2: Image and its proposed TDVT image representation obtained from its annotations.

In order to describe the image visually, it is important to encode the relative importance of the objects. Not every object in an image is equally important. In case of Figure 3.2a, the most prominent object in the image is the man in the red shirt. After that, the man wearing a jacket captures our attention next, followed by the building, sidewalk, and street. Visual information has an inherent sequential ordering in the sense that some objects capture our attention more quickly than others [92]. Our image representation captures this property through the ordering of the siblings of any sub-tree. As can be appreciated in Figure 3.2b, the child branches that are more important are listed first (closest to the left), while less salient children are listed at the end. As we will show further, the ordering is important since most prominent objects are discovered first in the inference process, exactly in the same way as humans interpret the images.

In our TDVT representation, we distinguish between two types of child edge relations. The first one, depicted as green edges in Figure 3.2b, corresponds to relations that imply clear ownership dependencies. This applies to the case of object and parts, or parts and some inherent properties such as color or shape. The second type of childhood relation, depicted as red edges, model weaker relations between the nodes where some dependence exists, but there is no clear ownership implied. In the rest of this chapter, we will denote the first type of relations as Left dependencies, and the second type of relations as Right dependencies. Examples of Left dependencies in the figure are man-jacket, man-pants, man-shoes where all the siblings correspond to parts associated with the man. An example of Right dependence is jacket-arm, where ownership association between jacket and arm is not so evident. A special case in the definition of Left and Right branches is at root level. In this case, a Left dependence represents an object that is notoriously salient with respect to other objects in the image. In the example, the man in red is considered a Left dependence, and all the remaining main objects are part of the Right dependence.

3.2 Pre-processing Images and Annotations from Training Dataset

The proposed TDVT image representation describes the content of the image in a hierarchical manner, encoding the importance of objects, and relations between the objects. The TDVT representation becomes powerfully useful when we are able to represent a large number of images which can then be used for training. Annotation of a complete dataset is a titanic task due the size of the current datasets (more than 100k images). We are interested in avoiding a special annotation process of the training images in the dataset, due to the associated high costs. This will also help us to keep our approach as independent as possible from the annotation process. Additionally, obtaining consensus from different annotators could be cumbersome, since there is no unique way to build the proposed tree representation. Therefore, it makes more sense to create the tree image representation from existing annotations and set up a set of parsing rules to build such representations in an unified manner.

Assuming that the training dataset contains bounding box annotations of the possible objects/stuff present in the image, we describe below the five steps involved in parsing a large image dataset to obtain TDVT representations. Figures 3.3 and 3.4 show the parsing obtained for two images randomly selected from the training dataset.

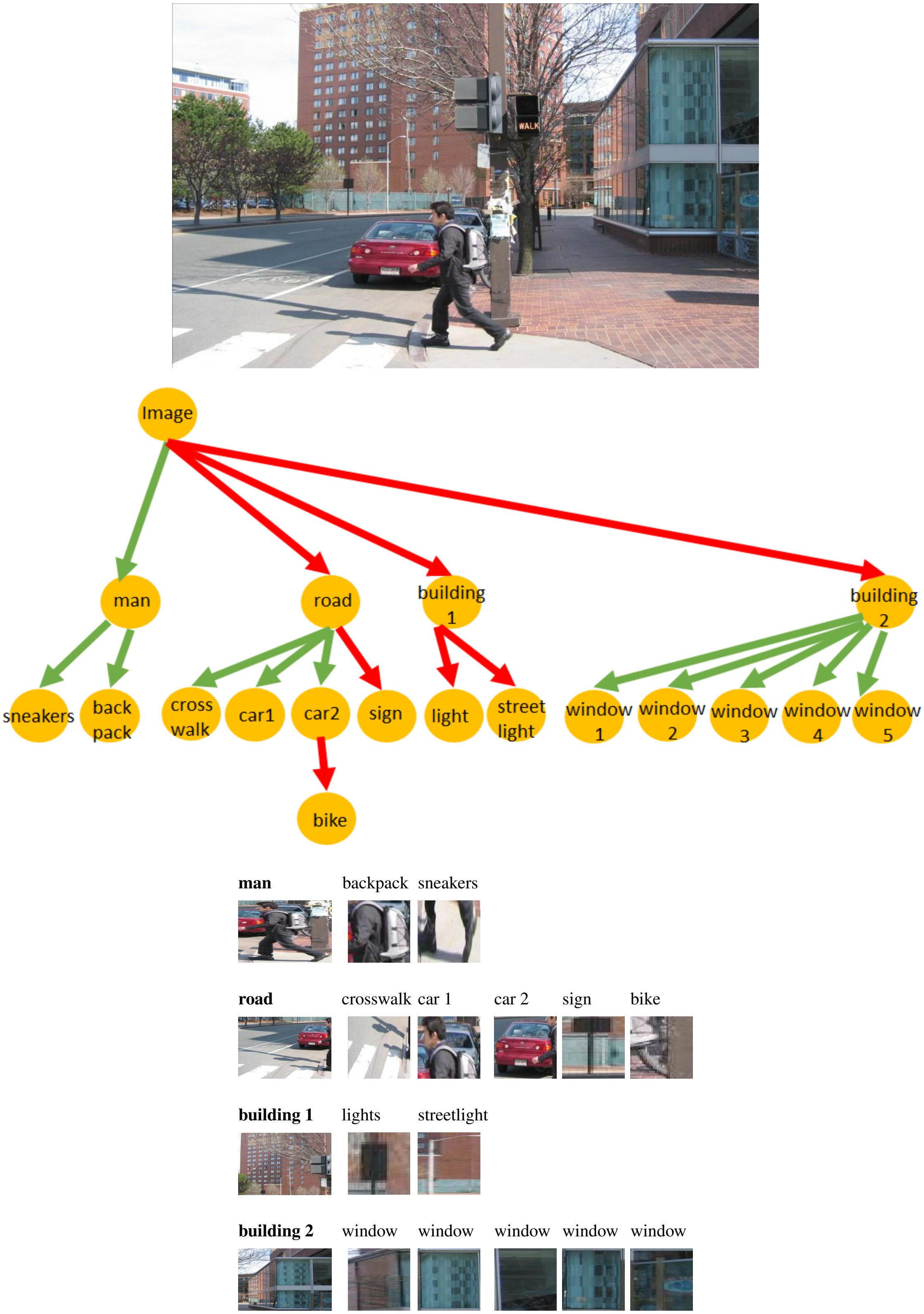


Figure 3.3: Example of an image parsed from the dataset. The TDVT image representation is depicted in the middle, while the images on the bottom are the different objects annotated that are part of the final Tree representation. Left dependencies are depicted in green color, and right dependencies are depicted in red color.

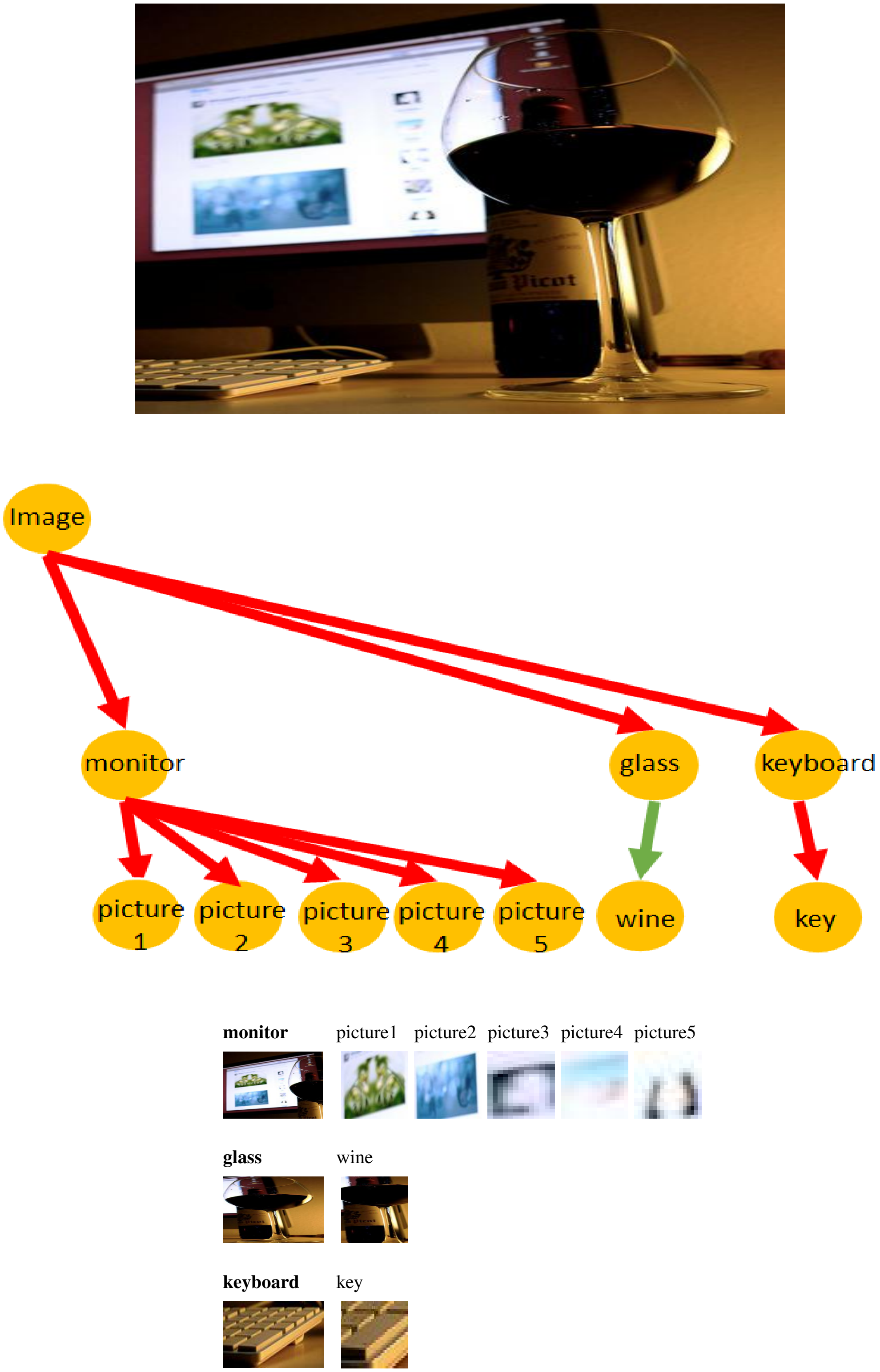


Figure 3.4: Example of an image parsed from the dataset. The TDVT image representation is depicted in the middle, while the images on the bottom are the different objects annotated that are part of the final Tree representation. Left dependencies are depicted in green color, and right dependencies are depicted in red color.

3.2.1 Merging duplicate instances from annotations.

We are dealing with a large number of classes, annotated by humans (typically turkers). Annotation corresponds to a bounding box b_i that encloses an object i, and has a label l_i. Typically, multiple individuals work on the same image, and it is quite common to find bounding boxes approximately enclosing the same object with different labels. In this step, we eliminate the duplicated annotations by considering the overlap between all the annotated bounding boxes for the image. We use the Jaccard index as a measure of the similarity between two bounding boxes b_i and b_j. Jaccard index between bounding box b_i and b_j is defined as:

$$J(b_i, b_j) = \frac{|b_i \cap b_j|}{|b_i \cup b_j|}, \tag{3.1}$$

where the numerator is the area of the common region enclosed by the bounding boxes b_i and b_j, and the denominator is the total area of the union of the regions enclosed by bounding boxes b_i and b_j. The bounding boxes with Jaccard index close to one,

$$J(b_i, b_j) > 1 - \gamma, \tag{3.2}$$

enclose approximately the same region of the image.

Annotations that satisfy the last condition are then examined semantically to determine if the annotations are variants of the same object. In order to examine the semantic similarity, we use the word distance between the labels of the annotated objects. The distance is computed using a path-based distance measure for words from the wordnet [93] hierarchy. Two labels might describe the same object if they have high semantic similarity,

$$\left|l_i, l_j\right|_s > 1 - \varepsilon, \tag{3.3}$$

where the score for a perfect match $l_i = l_j$ has a score equal to 1. Instances with high semantic similarity (Equation 3.3) and similar bounding boxes (Equation 3.2) are merged into a single node; they are considered noisy annotations of the same object.

3.2.2 Connecting related nodes

Our objective is to build a tree from the bounding boxes and labels. Every annotation available after merging duplicates (previous step) is a possible node of the tree $\mathbf{V} = \{V_1, V_2, ..., V_j, ..., V_N\}$. In this step, we define the mechanism to create edge connections, between the available nodes, $E = \{\{V_j, V_k\}, ..., \{V_l, V_m\}\}$. The main assumption here, is that nodes that are related must have at least some degree of visual overlapping. Hence, we only consider connections between nodes with Jaccard index greater than zero.

$$\{V_j, V_k\} \in E, \quad \text{if } J(b_{V_j}, b_{V_k}) > 0. \tag{3.4}$$

Each one of these edges are analyzed individually to determine the type of dependence (Left or Right) between the related nodes.

An obvious Left dependence exists when one of the nodes, the child, is inside the other node, the parent; and there is a close semantic similarity between the nodes.

$$\{V_j, V_k\} \in E^L, \text{if} \quad b_{V_j} \cap b_{V_k} > \zeta \cdot |b_{V_k}| \quad \wedge \quad \left| l_{V_j}, l_{V_k} \right|_s < 1 - \varepsilon_2, \tag{3.5}$$

where E^L is the set of Left dependencies, $|b_{V_k}|$ is the area of the bounding box b_{V_k}, ζ is a tolerance value for the area intersection, $\left| l_{V_j}, l_{V_k} \right|_s$ is the semantic similarity between the nodes V_j and V_k, ε_2 is a tolerance value for the word similarity, V_j is the parent node, and V_k is the child node.

There are datasets that have additional annotations that can be utilized to define dependencies. For example, the visual genome dataset [94] provides relations between nodes through word connections that can be used to define edges between nodes. The existence of left dependencies can be established by matching word connectors against a list of key words that denote ownership relations such as “in”, “wears”, and “have”.

Any edge that belongs to E, which is not a Left dependence E^L is considered to be part of the set of right dependencies E^R,

$$E^R = E \setminus E^L. \tag{3.6}$$

3.2.3 Generating sub-trees

The obtained list of edges E is used to identify formed disjoint sub-graphs. In this step, each one of these sub-graphs are converted into a tree structure. A tree's edges cannot have cycles. However, the disjoint graphs obtained so far, do not have this property. A spanning tree of a graph is just a subgraph that contains all the nodes without cycles. A graph may have many spanning trees, the Minimum Spanning Tree (MST) algorithm can be used to connect all the nodes together with the minimal total weighting for the edges. This way, any existing loop in each graph is removed through a Minimum Spanning Tree (MST) algorithm, where the weights of the edges in the graph are defined using the path-based distance measure for word similarity from the wordnet hierarchy,

$$w(\{V_j, V_k\}) = \left| l_{V_j}, l_{V_k} \right|_s. \tag{3.7}$$

Each one of the obtained group of nodes that do not form loops are analyzed to identify their root nodes, and edge directions are defined using parenthood hierarchical relations. The parenthood relations from Left dependencies $E^L = \left\{ \left\{ V_{\text{parent}_i}, V_{\text{child}_j} \right\}, \ldots, \left\{ V_{\text{parent}_n}, V_{\text{child}_o} \right\} \right\}$ automatically determines their edge directions.

Any node in a tree must have at most one parent. In cases where the existing Left dependencies indicate that a node has multiple parents, only one parent edge is selected. Formally,

$$E^L = \{\{V_i, V_j\}, \{V_k, V_j\}, \{V_l, V_m\}, ..., \{V_x, V_y\}\} \Rightarrow E^L = \{\{V_i, V_j\}, ..., \{V_x, V_y\}\}. \quad (3.8)$$

The largest area overlap between the node and its possible parents will determine the choice of parenthood relation, shown as

$$V_i \cap V_j > V_i \cap V_k, ..., V_i \cap V_z. \quad (3.9)$$

A list of root node candidates is created from the existing Left and Right dependencies. A node with only one connection and small bounding box size is a leaf, and this causes it to be removed from the list of root node candidates. Nodes are discarded one by one starting from known leaves until only one root node remains in the list. The criteria used for discarding nodes are a) consistency in the sizes of the bounding boxes to identify parenthood directions, and b) each node can have at most one parent. The last condition implies that if a parent exists for a node, any other edges connecting the same node are children connections and are discarded from the root node candidate list.

3.2.4 Building a preliminary tree

The set of resulting sub-trees are gathered to form a preliminary tree. The root node is defined as the node covering the whole image. Each one of the root nodes from the generated sub-trees becomes a child node of the root node; these represent the main objects in the image (See Figure 3.2b). Nodes that are isolated are considered irrelevant, and are not included in this preliminary visual tree representation.

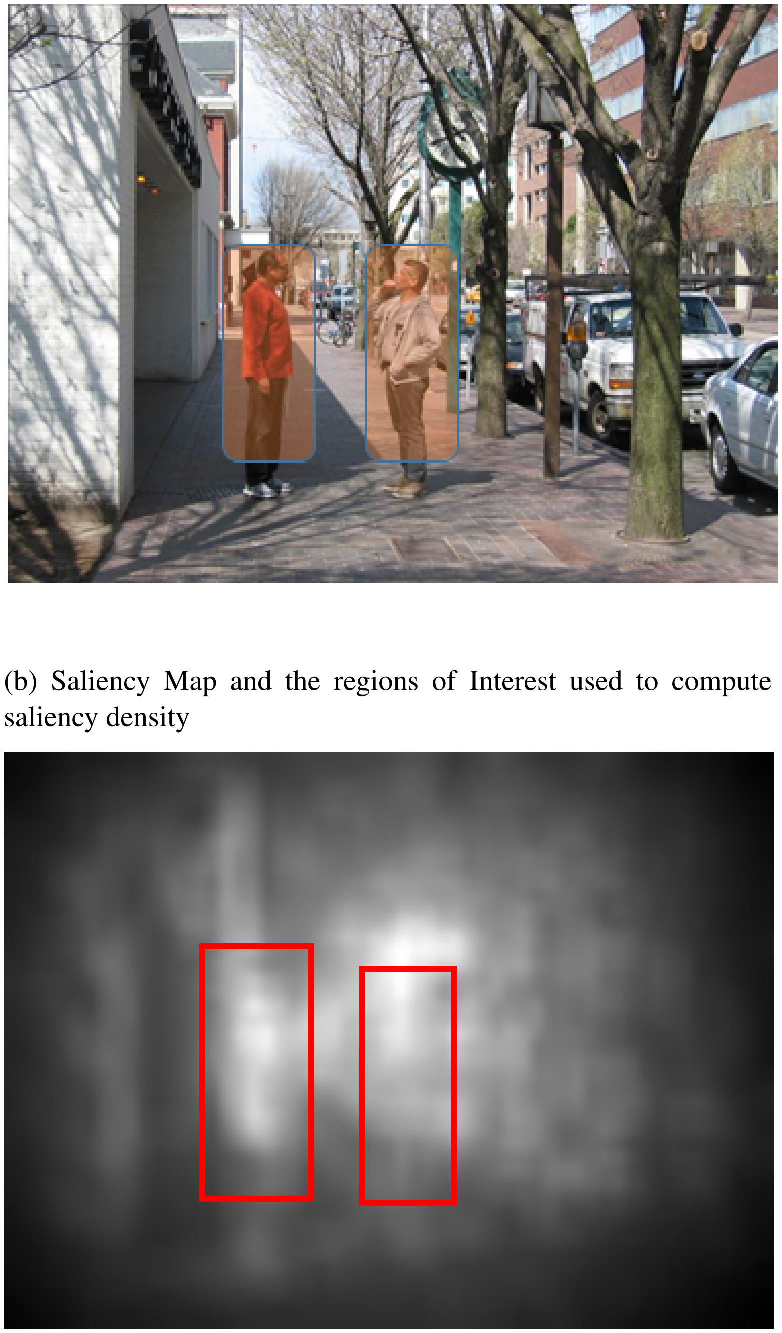
3.2.5 Re-ordering branches

The preliminary visual tree was built without any consideration about the importance of the nodes of the tree. The sorting of a group of siblings is done independently of other groups of siblings. The siblings are sorted according to their visual importance.

A saliency map is a function that gives a subjective perceptual value about the property that "makes some items in the world stand out from their neighbors and immediately grab our attention"[1]. The saliency map was designed as input to the control mechanism for covert selective attention. It is also used to predict eye movements. Given a saliency map, we define the saliency density for a region of interest as the sum of the saliency values enclosed by the associated object's bounding box divided by its area. Saliency density is computed for every node of the tree.

Every sibling is then independently processed. The siblings are sorted in a descending order according to their saliency density values. Finally, a special procedure is performed over the children of the root node to identify if the nodes qualify as Left or Right dependencies. Nodes with a very high saliency density values are interpreted as Left dependencies and the rest of the nodes becomes Right dependencies.

[1] Laurent Itti (2007), Scholarpedia, 2(9):3327



(a) Image and two of its regions of interest

(b) Saliency Map and the regions of Interest used to compute saliency density

Figure 3.5: Image and saliency map used to compute the visual importance.

3.3 Training

The training process to learn about the image composition and internal hierarchies is presented in this section. Images are processed sequentially starting from the top node that represents the full image and analyzed edge by edge to systematically analyze all the nodes that are part of the TDVT representation of a training image.

When a node is analyzed in a TDVT representation, there are some key elements that need to be learned. Firstly, it is essential to learn if a particular type of edge exists given the current state of the node. A 4-way branch classifier learns about the possible existence of an edge. Secondly, given the existence of an edge, the label of the node linked to such an edge needs to be determined. A Top-Down Tree LSTM (Long Short-Term Memory) is the proposed network to learn to predict likely node labels for the edge currently analyzed.

At this stage, a method to extract visual features, and obtain scores for regions of interest is needed. Fast R-CNN architecture [36] takes an image as input into a Convolutional Network Network(CNN). Multiple Regions of Interest (ROI) are evaluated one by one using an ROI pooling layer that maps the outcome of the CNN spanning the ROI into a fixed-size vector. Each one of these vectors pass through a multi-layer Perceptron (MLP) that outputs the scores for any possible object label. We use the layers and weights of this network to obtain visual features, and obtain scores of regions of interest (ROI) for a reduced set of labels determined by the predicted output of the Top-Down Tree LSTM network.

3.3.1 Label Predictions Using Top-Down Tree LSTM network

We propose an adaptation of the Top-Down Tree LSTM network [95], which was originally proposed in the Natural Language Processing (NLP) area, for solving our vision problem. The traditional LSTM operates on sequences, whereas the Top-Down Tree LSTM was developed to extend the power of LSTMs to tree structures. The Top-Down Tree LSTM preserves the main advantages of the LSTM architecture such as the capacity to process sequences of different sizes and long term memory. We use this network to predict the object category (label) of the node linked by the examined edge, given the current state of the tree.

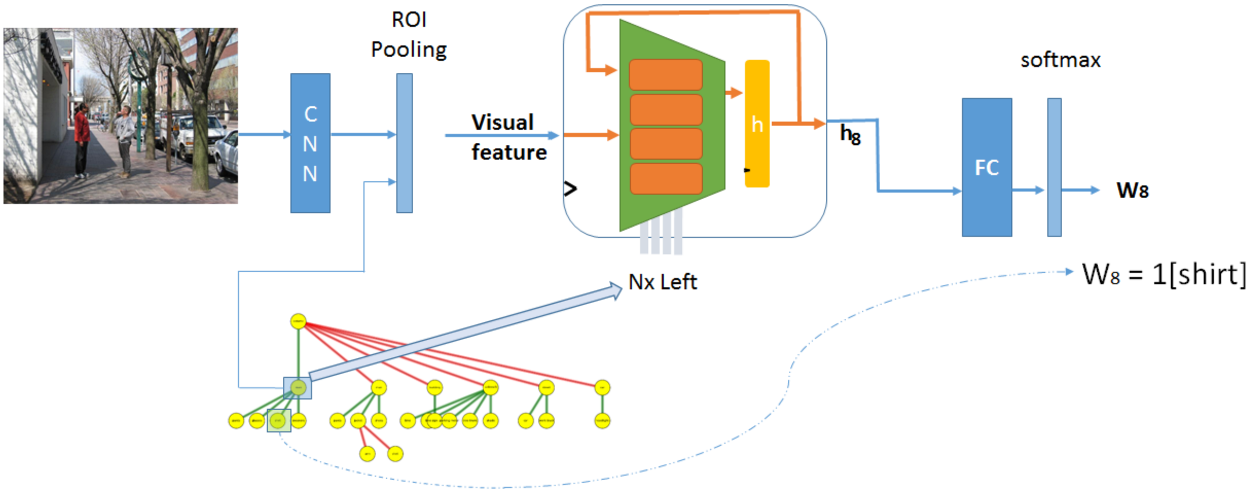


Figure 3.6: Proposed training model utilized for the prediction of the associated object according to a hierarchical structure. The input image passes once through a Convolutional Neural Network, and visual features are extracted for each node of the tree structure representation. The network is trained with the goal of predicting the label of the node to the other side of the edge being trained. In the figure, a Next-Left dependence is trained, and used to select the Next-Left LSTM unit from the Top-Down Tree LSTM. h_8 represents the hidden state. The predicted label, W_8, is a one-hot vector that represents the label 'shirt'.

Figure 3.6 presents the model used for training our image representation. The network uses as input an image and its TDVT image representation. Each node of the TDVT representation has associated bounding box coordinates that are used as ROI to pool visual features. The network is trained edge by edge. Given a node, and an edge, the network is trained to predict the label of the other node that is connected to the edge. The output is encoded as a one-hot vector. The vector consists of 0s in all components with the exception of a single 1 in a component used uniquely to identify the label. Every tree is processed sequentially in a Breadth-first order with the help of the Top-Down Tree LSTM block, which will be briefly explained in the sub-section 3.3.1.1.

3.3.1.1 Top-Down Tree LSTM

The core block of a Top-Down Tree LSTM is explained in this sub-section. A total of four LSTM networks are used by this block. Our TDVT image representation has two type of edge dependencies, Left and Right dependencies. Left dependencies are modeled by two LSTMs networks, and Right dependencies are modeled by another two LSTMs networks. The reason for having two LSTMs units for Left dependencies is to distinguish the first Left dependent from the other left dependents having the same parent. Hence, Left dependencies are modeled by a first LSTM that controls the initial Left dependence (Init-Left), and a second LSTM unit that controls the connections between the remaining nodes of the same parent (Next-Left). Clearly, the nature of the information that the LSTM units handle is different. While Init-Left dependencies represent transitions from a higher level to a lower level in the tree hierarchy, Next-Left dependencies handle transitions in the same hierarchical level. The same reasoning is elaborated for Right dependencies. The Right dependencies are controlled by two LSTMs: one to control initial right connections (Init-Right), and other to control consecutive right dependencies of the same parent node (Next-Right).

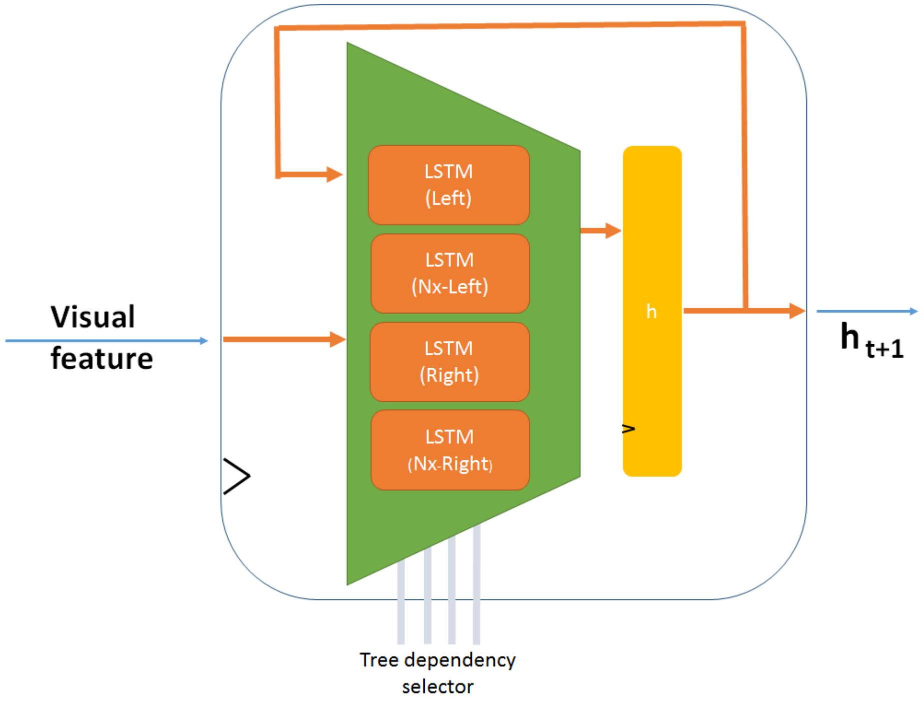


Figure 3.7: Details of Top-Down Tree LSTM unit. The tree dependency selector activates the LSTM unit according to the edge of the tree that is currently analyzed. The hidden vector h is shared by all the units, and updated for every edge evaluation.

Figure 3.7 presents a diagram of a Top-Down Tree LSTM. It contains four LSTM units, a shared hidden state vector h, and a tree dependency selector that controls which LSTM unit is activated.

Edges are processed starting from the root node. A dependency path $D(w)$ is defined as the path between the root and w, consisting of the nodes on the path and the edges connecting them. $D(w)$ is described by a sequence of tuples $\langle node, edge\text{-}type \rangle$. The nodes to be processed are visited according to its breadth-first search order. Let $< w_{t'}, z_t >$ denote the last tuple in $D(w_t)$, and $z_t \in \{$*Init-Left, Init-Right, Next-Left, Next-Right*$\}$. Let x_t denote the visual feature pooled from the node $w_{t'}$. Let $H \in \Re^{d \times (n+1)}$ denote the shared hidden states of all the existing nodes, where d is the hidden unit dimension size, and n the number of nodes in the image representation. Every time an edge is processed, only one of the four LSTMs is activated based on edge type z_t. The hidden state is updated as:

$$h_t = LSTM^{z_t}(x_t, H[:, t']), \tag{3.10}$$

$$H[:, t] = h_t. \tag{3.11}$$

It is evident that the hidden state $H[:, t]$ represents the dependency path $D(w)$. The probability of w_t given its dependency path $D(w_t)$ is estimated by a softmax function:

$$P(w_t | D(w_t)) = \frac{\exp(y_{t,w_t})}{\sum_{k'=1}^{|V|} \exp(y_{t,k'})}, \tag{3.12}$$

where $|V|$ is the number of trained object detectors, and y_t is computed with a fully connected FC layer using the hidden state as input. Figure 3.6 shows the complete model including the FC layer and softmax.

3.3.2 4-way branch classifier

A Top-Down Tree LSTM network predicts the most likely output object to be found given the sub-tree processed until that point (controlled by the shared hidden vector h), and the type of dependence that is currently processed. Then, an explicit assumption of the framework is that the edge dependence is known. However, during testing, the input is just an image, there is no predefined tree. Therefore, every time a node is found, it is necessary to determine if the node has an edge, and what type of edge the node has.

Four different classifiers are trained in order to learn to build a tree. Each classifier models a different type of dependence (Init-Left, Next-Left, Init-Right, Next-Right). During training, the classifiers use the visual feature of the node analyzed, and the current hidden vector h_t of the Top-Down Tree LSTM as inputs to predict if there exists an edge of that type or not. The learned model predicts the existence of an edge while the image is analyzed.

3.4 Inference

This section describes the procedure to obtain the detected objects and the associated TDVT image representation from a test image. We assume the availability of object proposals for the image, which can be obtained using one of the existing object proposal methods such as Edgebox [45], selective search [33], or Bing [96] .

The TDVT image representation is built gradually edge by edge. The inference process starts from the root node which represents the full image. The convolutional layers from the model trained for object detection are used to extract visual features. Using the visual feature for the full image and the hidden vector h at its initialization value, the classifier for Init-Left dependence is evaluated to determine the existence of a Left dependence. In the case that the output is positive, the Top-Down Tree LSTM network for the Init-Left dependence is utilized to predict a label vector. Each element of the vector is a score that indicates whether or not an object category exists in the image. An object detection search is performed in the image for a limited set of object categories determined by the label vector.

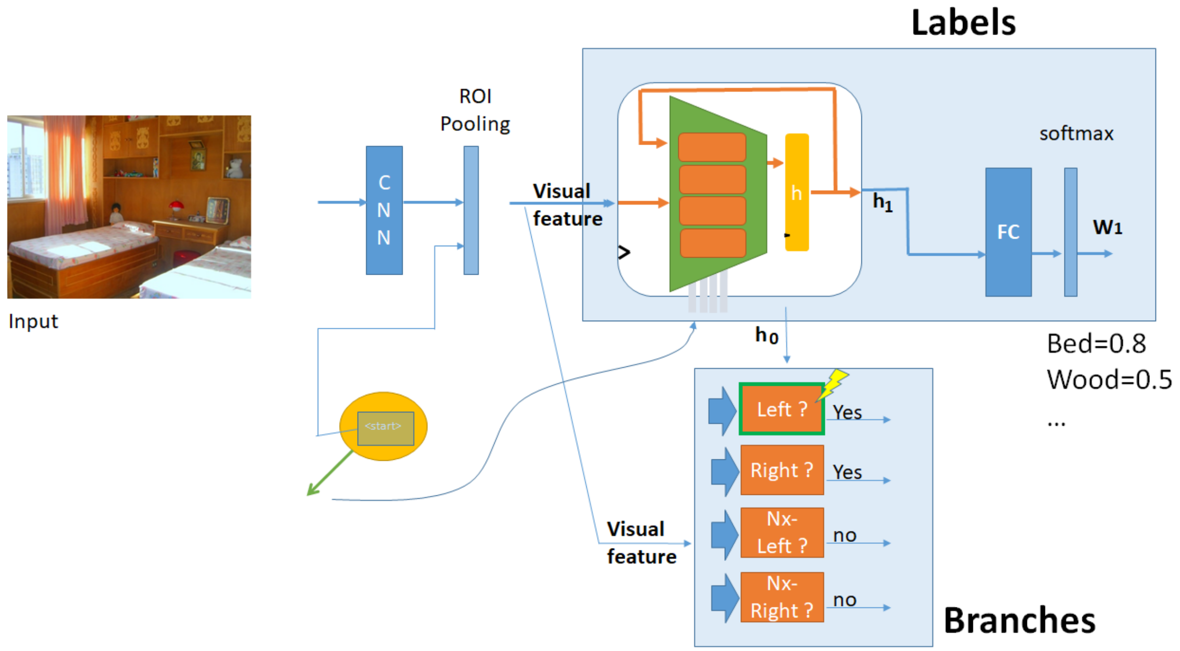


Figure 3.8: Details of the inference process. The 4-way branch classifier determines the existence of a edge. In this case, the starting node is the full image, and a Left and a Right branch exist for the node. The Top-Down Tree LSTM predicts labels of the node linked by the edge of type Left. The predicted labels are used to limit the number of classes used for object detection.

The search is performed by computing scores using the classification layers of the trained object detector, having as input the visual features pooled from the bounding boxes provided by the object proposal method. The detection with the highest score among most likely objects, is declared as the node connected by the Left edge. Subsequently, the existence of other Left dependencies for the examined node are determined by the classifier for Next-Left dependence. Visual features extracted from the latest found node and the corresponding updated hidden vector h_t are used as inputs to the classifier. Every time a Next-Left dependence is found to exist, the Top-Down Tree LSTM determines the most likely labels for the corresponding edge. The object search is performed in a reduced set of categories, in the same way that the Init-Left dependence is computed. After the Left branch is completely processed (or the root node does not have a Left dependence), identical process is followed for the Right branch.

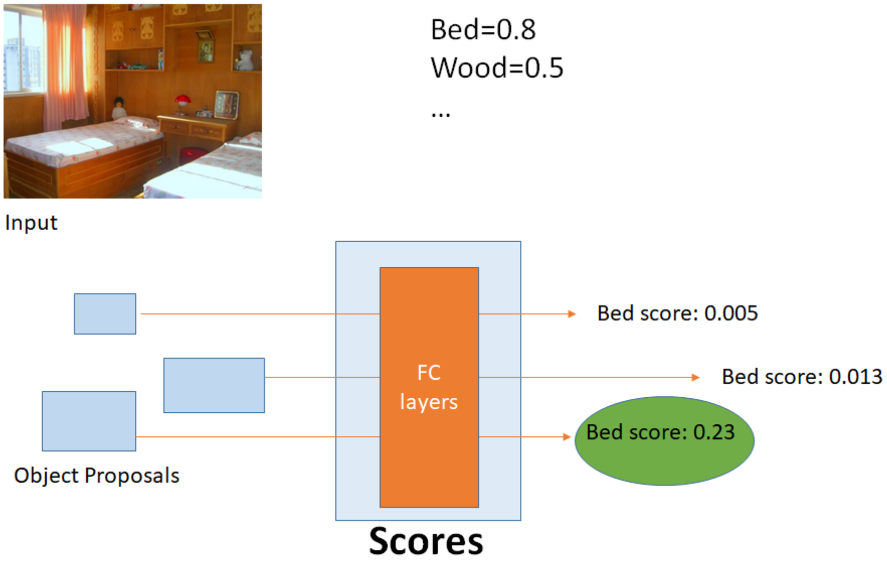


Figure 3.9: Details of the inference process. The best score among the object proposals for a reduced set of categories determined by the Top-Down Tree LSTM is used to pick the node associated with the edge that is inferred.

A classifier for the Init-Right dependence determines the existence of a Right branch. If a Right dependence exists, then the Top-Down Tree LSTM determines the possible categories of objects related to the edge. The object search procedure is performed only among the found possible categories. When the classifier for Next-Right dependencies indicates the existence of an additional Right dependency, the same process of category prediction and object search in a reduced set of categories, is repeated until there are no more Next-Right dependencies.

Once the children for the root node are computed, we proceed to infer edges and nodes of lower levels of the tree using the aforementioned object search process used for Left and Right dependencies in the root node. Left dependencies deal with parts, therefore it is reasonable to process only the bounding boxes with some level of overlap with respect to the parent node. The search process in the Right dependence is more flexible and only requires that the parent node has at least one pixel of overlap with respect to the parent node. The result of this operation is a tree that represents the image, with nodes that contain object detections results (bounding box, label, and score).

The inference process used to build a TDVT representation is summarized in the Algorithm 1. Two lists are used to control which node will be processed next. A list named NextList contains information about nodes that have edges type Next-Left or Next-Right, and a list named InitList contains information about nodes with edges type Init-Left or Init-Right.

3.5 Experiments

We perform experiments on the visual genome dataset [94]. The visual genome dataset has 108,077 images from the intersection of the YFCC100M [97] and MS-COCO [44] datasets. The annotation

```
input  :
    • Image to be processed
    • Object proposals (Edge Box)
output: TDVT representation of the image
Initialization;
    Initialize NextList to empty;
    Initialize $h_0$;
    Get visual feature (full image);
    Compute 4-way branches classifier (section 3.3.2);
    If initLeft then
    | Fill up InitList indicating the node and their type of edge is InitLeft;
    If initRight then
    | Fill up InitList indicating the node and their type of edge is InitRight;
while NextList is not empty OR InitList is not empty do
    while NextList is not empty do
        Get visual feature (ROI of analyzed node), Get $h_i$;
        Compute 4-way branches classifier (section 3.3.2);
            Update NextList if there is a NextLeft or NextRight;
            Update InitList if there is a InitLeft or InitRight;
        Compute Top-Down Tree LSTM (section 3.3.1) → set {label name, Label score};
        Pick the top N labels with highest label score;
        From the N selected labels:
            Pick object (proposal, label) with highest det. score (Fast-RCNN) ;
        Add the object as a new node of the tree;
    end
    if InitList is not empty then
        Get visual feature (ROI of analyzed node), Get $h_i$;
        Compute 4-way branches classifier (section 3.3.2);
            Update NextList if there is a NextLeft or NextRight;
            Update InitList if there is a InitLeft or InitRight;
        Compute Top-Down Tree LSTM (section 3.3.1) → set {label name, Label score};
        Pick the top N labels with highest label score;
        From the N selected labels:
            Pick object (proposal, label) with highest det. score (Fast-RCNN) ;
        Add the object as a new node of the tree;
    end
end
```

Algorithm 1: Algorithm to infer a TDVT representation from a test image.

includes 5.4 million region descriptions and 2.3 million pair-wise relationships, which are used as possible nodes of the TDVT image representation.

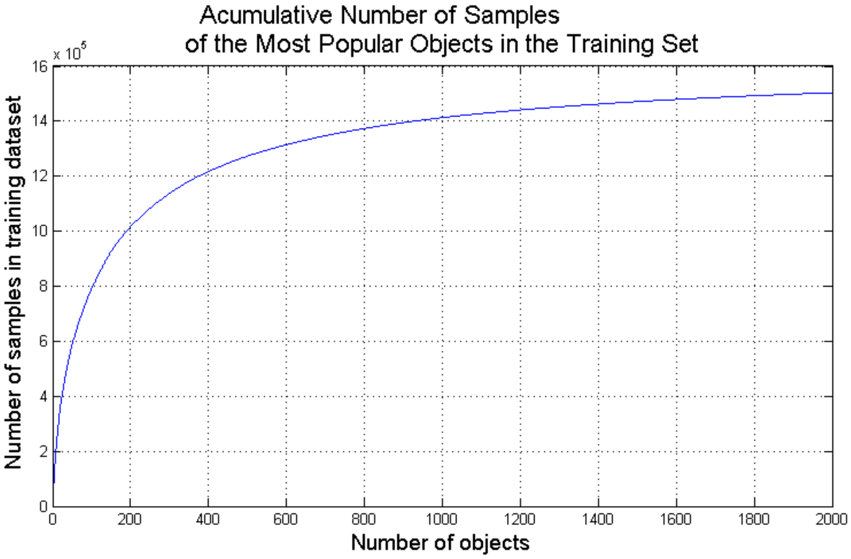


Figure 3.10: Accumulative number of samples for the most common objects in the dataset.

Using the training data, the number of instances of each object category after obtaining TDVT representations are sorted in descending order to establish the most common object categories. We selected the first two thousand popular labels as the classes to be used in the experiments. The most common class of the training dataset is ‘man’ with 37,292 training samples, while the class ranked 2000 is ‘desert’ and only has 52 samples.

Fine-tuning of a Fast R-CNN object detection model is performed using the full annotations of the selected two thousand classes. The network used to train the object is based on the VGG 16 network [46] and EdgeBox [45] is used as the object proposal method.

We now describe the choice of parameters and implementation details for the image parsing process of the dataset. We selected the parameters for merging similar objects in an image as

$\gamma = 0.2$, and $\varepsilon = 0.5$. Edges are considered Left Branches if the intersection of the nodes is more than 80% the area of the smallest bounding box ($\zeta = 0.8$), and path similarity is over 0.3 ($\varepsilon_2 = 0.7$). We use Kruskal's Algorithm to solve the Minimum Spanning Tree (MST) and generate trees from graphs. The saliency method by Duan et al. [98] was used to determine the order within the siblings. A density saliency over 0.8 defines a Left dependence in the root level.

The training details of our network are described next. The Top-Down Tree LSTM of our experiments has a hidden vector dimension of 300, was trained with Stochastic Gradient Descent(SGD), using batches of 64 images. The four classifiers to learn to build the tree was implemented as a Multi Layer Perceptron (MLP) with two hidden units of 300 dimensions.

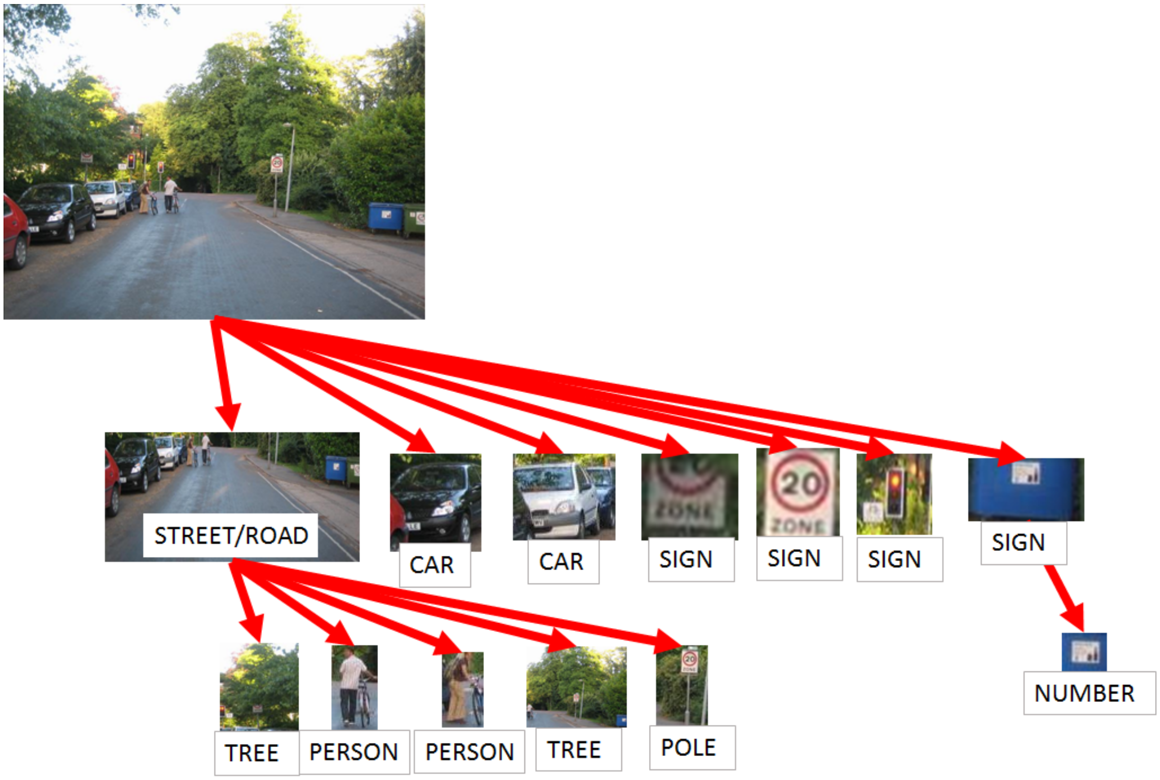


Figure 3.11: TDVT image representation for a test image. The main objects in the image in descending order of importance are: road, car 1, car 2, and different signs. The road has associated a pair of trees, a pole, and the two persons at the end of the road.

3.5.1 Qualitative Results

In this section we showed some qualitative results of the proposed approach to perform simultaneously object detection and image understanding. Test images are feed to infer a TDVT representation that express the visual content of the image.

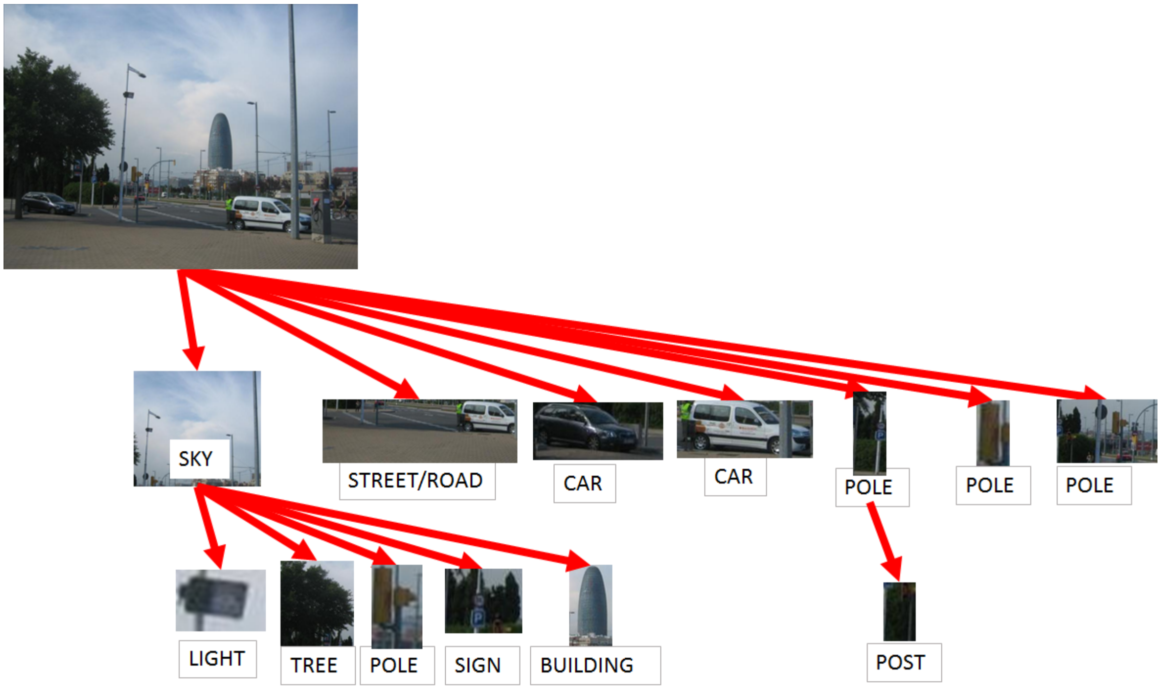


Figure 3.12: TDVT image representation for a test image. The main objects in the image in descending order of importance are: sky, street, car 1, car 2, and different poles. The sky has associated objects such as a tree, a pole, a light, a sign, and a building.

Figures 3.11 and 3.12 shows the results obtained for two randomly selected test images. While traditional object detection produces a set of bounding boxes and scores with the most likely objects, our method produces a reduced set of objects that are essential to describe the visual content, and the possible relations between these objects. For example, Figure 3.11 shows the street/road as the most important objects, followed by two cars, and some signs. Associated to the street/road, in a lower level of importance the image contains two trees, a pole, and two persons.

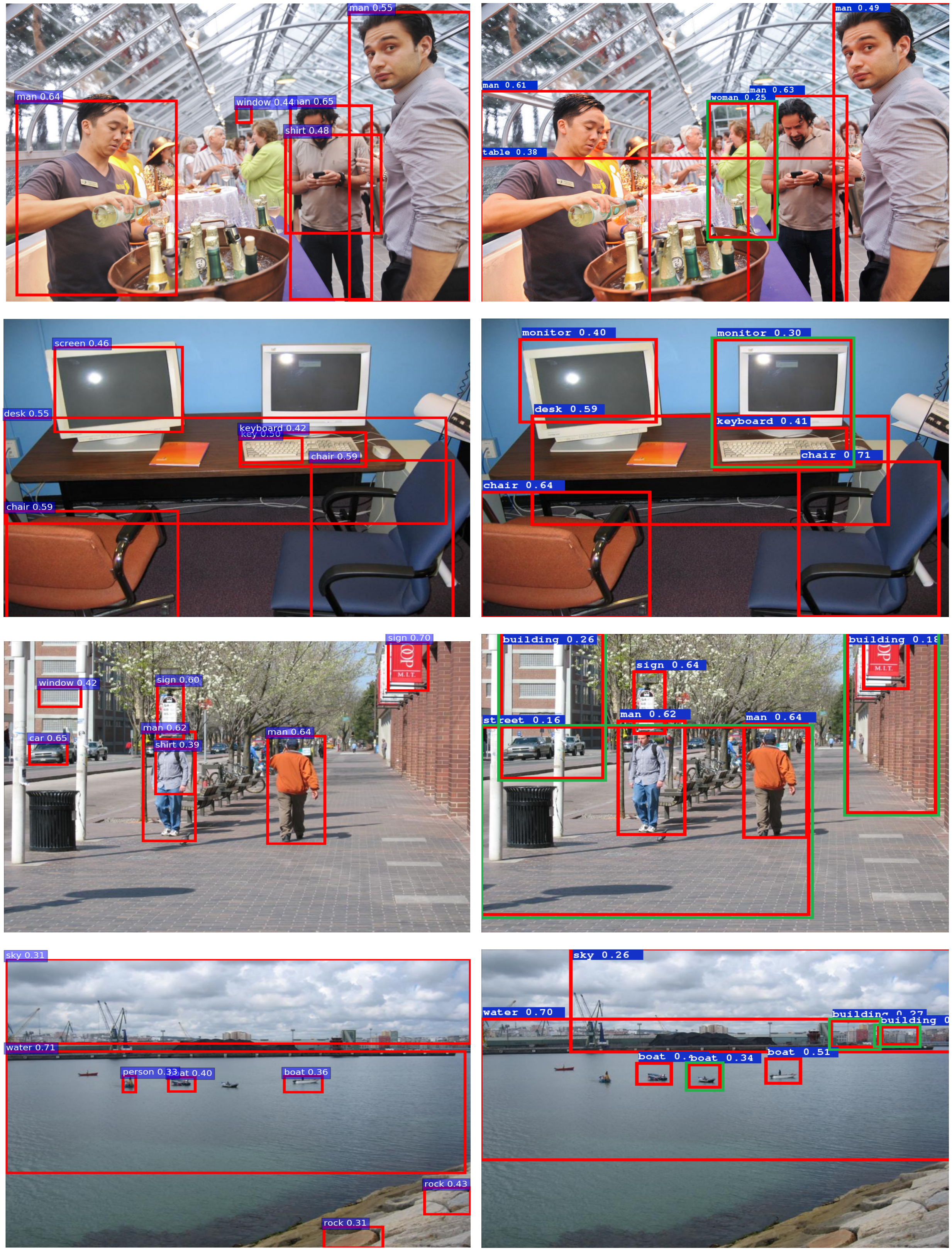


Figure 3.13: Examples of images showing the most confident detections using Fast R-CNN object detection (left side) and the proposed method (right side).

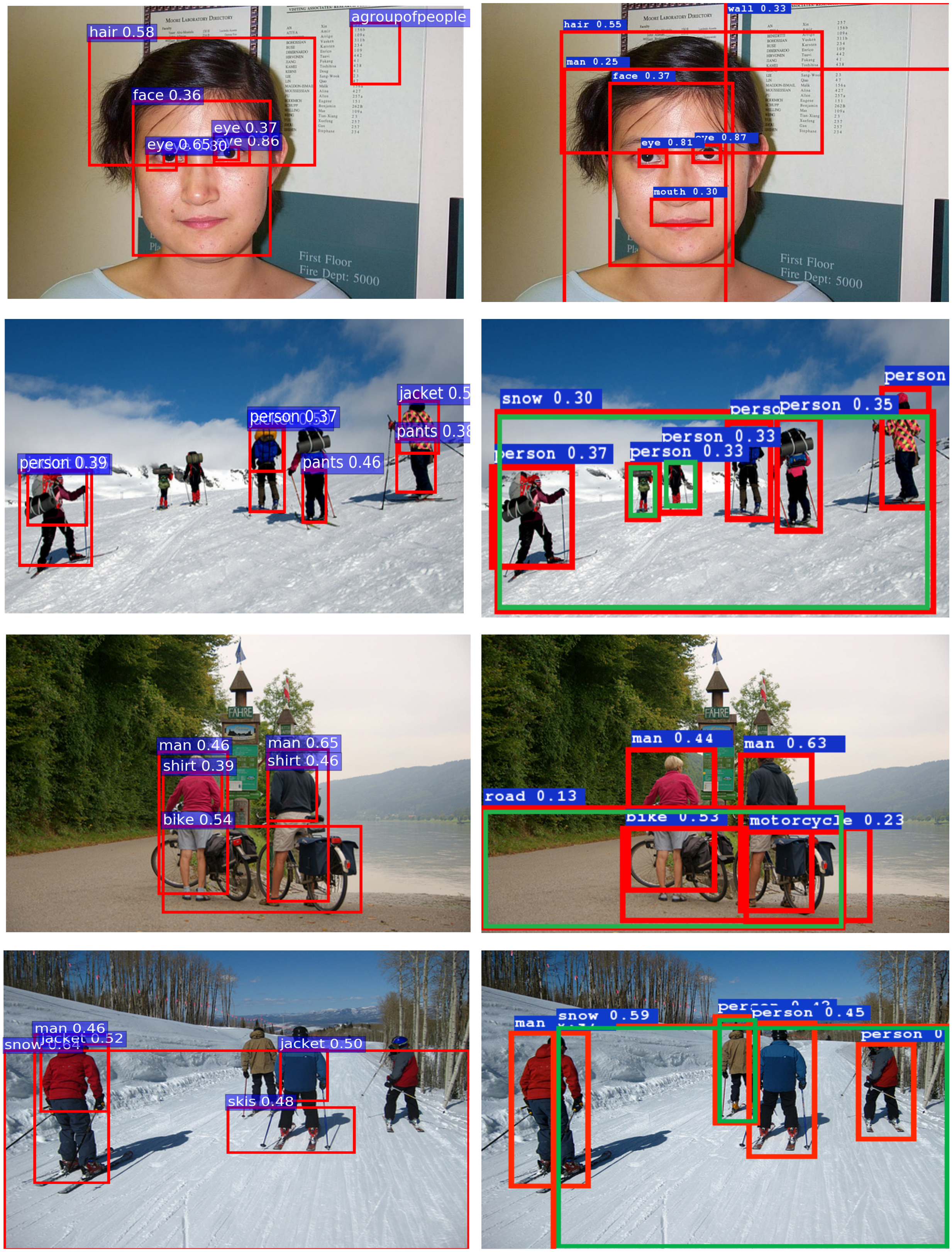


Figure 3.14: Examples of images showing the most confident detections using Fast R-CNN object detection (left side) and the proposed method (right side).

Figure 3.13 shows the object detection results of some testing images. For the proposed method (right side), we show only the detected objects from the children of the root node, which determines the most important objects of the image. For the Fast-RCNN (left side) we show the most confident results according to the object detection score. Our method allows to find un-spotted objects like the 'woman' in the first set of images, or the 'monitor' screen in the second set of images. It also, allows to find the most important elements of the images such as the 'street' and the 'building' in the third set of images. Finally, our method can get rid of inconsistent categories such as the 'person' in the last set of images. More examples can be found in the figure 3.14.

3.5.2 Dataset Parsing

We perform a user study to evaluate the quality of the visual tree representations obtained after parsing the images and annotations of the visual genome dataset. Amazon mechanical turkers were used for this evaluation. Every image of the visual genome dataset was presented next to the visual tree representation obtained from the annotations. A total of 156 turkers were asked to tell if the visual tree properly represents the content of the image. Each image was rated by at least seven different turkers.

Table 3.1: Evaluation of the parsing of images from annotations. 86.83 % of the users considers that the obtained tree represents properly the content of the image.

Yes	No
86.83 %	13.17 %

The results are presented in Table 3.1. Most turkers consider that the parsed image representations represent properly the content of the images of the dataset.

3.5.3 Category Prediction

The trained network predicts the category of the object linked by the edge currently examined, given the sub-tree processed until that point. Hence, a prediction is performed for each existing edge during inference time. The test dataset of the visual genome dataset is parsed to obtain TDVT representations of the test images (previous subsection). We use these representations as ground truth and evaluate the performance of the category prediction task for each edge. We evaluate the quality of the predictions as a rank problem using a Cumulative Match Characteristic (CMC) curve. Figure 3.15 shows the results.

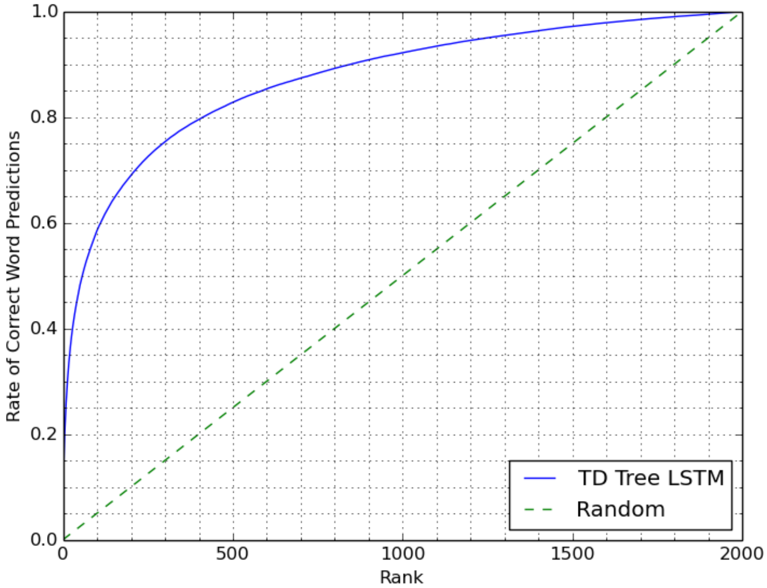


Figure 3.15: Rate of correct prediction of categories for different ranks.

Table 3.2: Comparison of the Average Precision-Recall (APR) of Fast R-CNN detector and the proposed approach for the forty four most popular classes of the dataset. The mean Average Precision (mAP) of the two thousand classes is shown in the last column of the table. Our method improves the detection in 38 out of the main 44 classes.

	man	window	person	shirt	wall	ground	building	sign	tree	woman	light	head
FCNN	42.91	31.21	17.39	42.69	18.99	28.02	35.28	47.43	35.82	**36.02**	17.03	40.75
Ours	**50.38**	**42.75**	**24.23**	**45.47**	**22.93**	**29.0**3	**44.04**	**49.93**	**44.13**	31.28	**19.58**	**42.60**

	pole	grass	hair	hand	sky	table	water	leg	car	people	pants	clouds
FCNN	15.82	21.84	30.56	30.26	36.12	**34.82**	42.33	20.91	45.75	24.59	**39.04**	6.77
Ours	**20.09**	**30.42**	**32.97**	**33.26**	**36.19**	33.09	**48.18**	**31.18**	**47.77**	**32.92**	38.21	**18.11**

	wheel	eye	ear	door	hat	trees	floor	plate	line	shadow	leaves	snow
FCNN	25.02	19.04	22.17	18.47	**21.22**	16.60	26.20	**45.42**	11.93	4.36	10.95	24.03
Ours	**29.68**	**19.39**	**26.35**	**21.95**	20.89	**21.93**	**31.69**	43.27	**17.00**	**6.31**	**20.48**	**27.14**

	nose	shoe	jacket	chair	tail	fence	windows	letter	**Mean**
FCNN	16.45	21.93	**33.43**	41.58	23.51	15.62	3.80	15.23	11.12
Ours	**20.15**	**26.85**	32.66	**42.20**	**25.20**	**20.19**	**7.27**	**24.26**	**22.53**

The prediction process allows to guide the search process. Our framework predicts the right category in 60% of the cases, when the top 100 most likely categories are considered. Clearly, our framework is a lot better than giving to all the categories the same chance as in traditional object detection.

3.5.4 Improved Object Detection

We present quantitative results performed over the 5,000 images from the original test split of the visual genome dataset. We compare our method against the object detection results of the Fast-RCNN object detector.

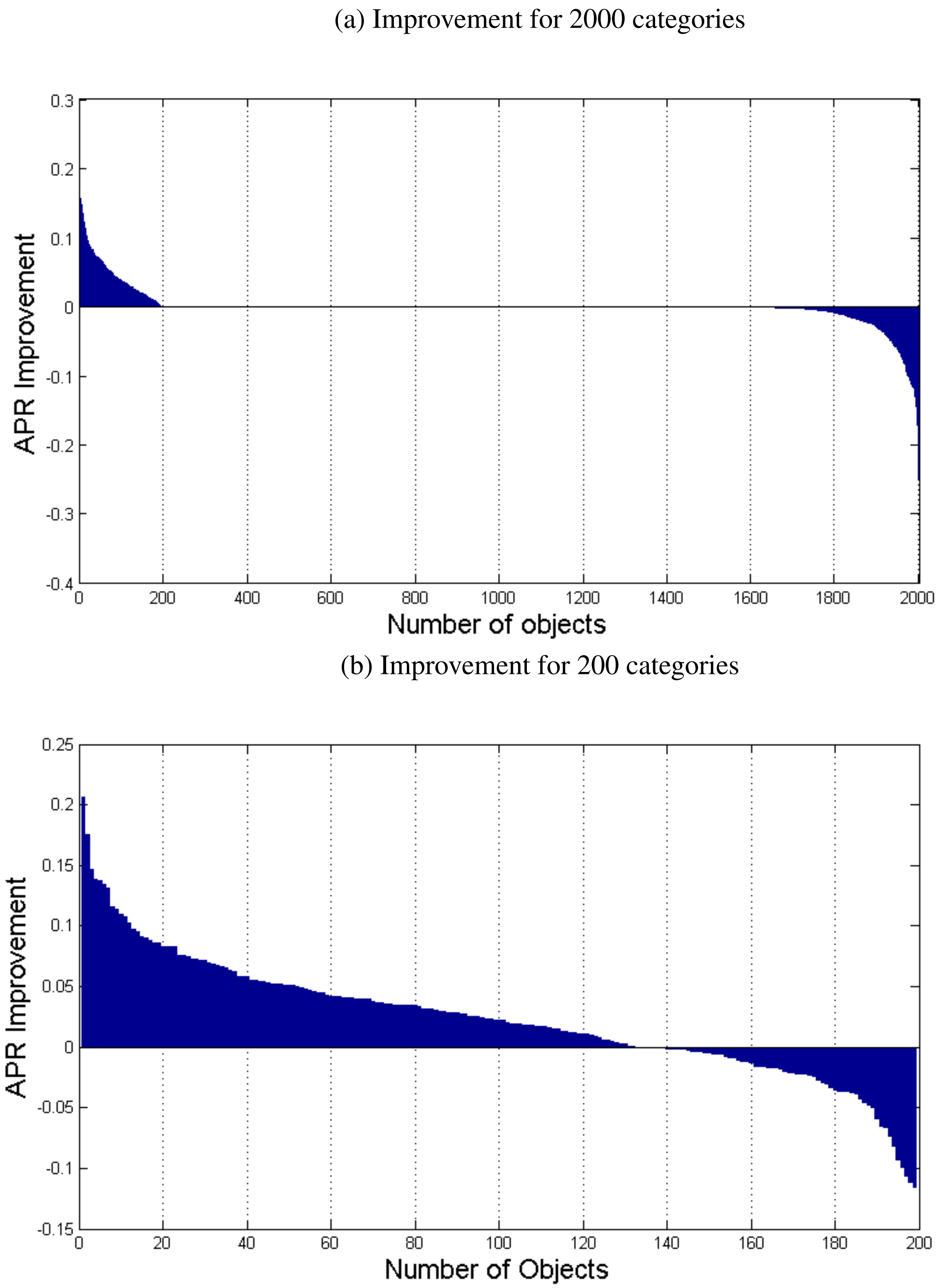
Table 3.2 shows the Average Precision-Recall (APR) for the forty four most common categories in the training dataset. Our method improves the detection in 38 out of these 44 classes. The average improvement of this large set of categories is +4.89 APR per category, while the loss in performance of the remaining six categories is lower (-1.75 APR per category). The loss in performance in the six classes was small compared to the gain on most of them.

The last column of Table 3.2 is the mean Average Precision (mAP) computed considering all the two thousand object categories. The mAP for the Fast-RCNN detector with two thousand categories was 11.12, and it increased to 22.53 using the proposed approach.

Figure 3.16 shows the relative improvements of the two thousand classes that are detected sorted by the improvement in the APR. Figure 3.16a shows that the number of improved categories is lower than the number of categories that decrease their APRs when all the two thousand categories are considered. If we only consider the two hundreds most popular categories from the training dataset (Figure 3.16b), we notice that approximately 70% of the object categories improve their APRs. These results indicate that our approach learn better models from elements that are popular in the images of the training dataset. The overall improvement in the mAP is achieved mainly for the categories that are very common in both training and test images.

3.5.4.1 Cross-dataset evaluation

We tested the ability of our model to improve the object detection results across datasets. The model previously trained with the TVDT representations and the 2000 categories selected from the visual genome dataset were used to test images of the SUN2012 test dataset [99]. We have not used any annotations from the training or validation sets. Hence, we have not performed fine tunning or re-training of the networks. We also use the network weights of the Fast R-CNN learned from



(a) Improvement for 2000 categories

(b) Improvement for 200 categories

Figure 3.16: Improvement of our model over the baseline Fast R-CNN. Object categories are sorted by the improvements. Figure a) shows the full 2000 categories, Figure b) shows the 200 most common categories. The overall improvement is achieved by having high improvement in the most popular categories in the training dataset.

the visual genome. Our goal is to observe if the learned model generalizes to any image, instead of just learning a particular dataset.

The SUN2012 dataset has annotations for 3819 object categories, which are different from the 2.000 categories from our trained dataset. Only 950 categories from the 2000 categories were mapped to the existing ground truth categories of the SUN2012 dataset. The results for object detection using Fast R-CNN and our framework are reported in Table 3.3.

Table 3.3: Evaluation of the object detection results in the test set of SUN2012 dataset trained with the data of the visual genome dataset. Metric used is mAP.

Fast R-CNN	Ours
3.40	**4.12**

The performance of the Fast R-CNN model trained on the visual-genome dataset is 3.40 mAP, considering all the 950 available categories. The apparent low performance results in this dataset are due in part to the discrepancy between the categories that the model can predict and the available groundtruth for this dataset. The experiment shows that our framework improves the results of object detection compared to the baseline Fast R-CNN. The resulting mAP is 4.12, which represents an improvement of 21.17% over Fast-RCNN object detection.

3.5.5 Obtained TDVT representation

The quality of the obtained TDVT representation is evaluated in the same way as we evaluate the visual representation obtained from parsing the dataset (section 3.5.2). Again, we asked turkers to tell if the obtained TDVT visual tree properly represents the content of the image. A total of 119 participated in this experiment. The results are shown in Table 3.4.

Table 3.4: Evaluation of the obtained TDVT image representation. 85.48 % of the users considers that the obtained tree represents properly the content of the image.

Yes	No
85.48%	14.51%

According to the turkers, 85.48% of the images are represented correctly by the obtained TDVT output. These values are close to the results obtained for the parsing process, but the obtained TVDT representation was obtained automatically from images without annotations. The presented values are the average result over many images. This helps to reduce the bias from users towards any of the options.

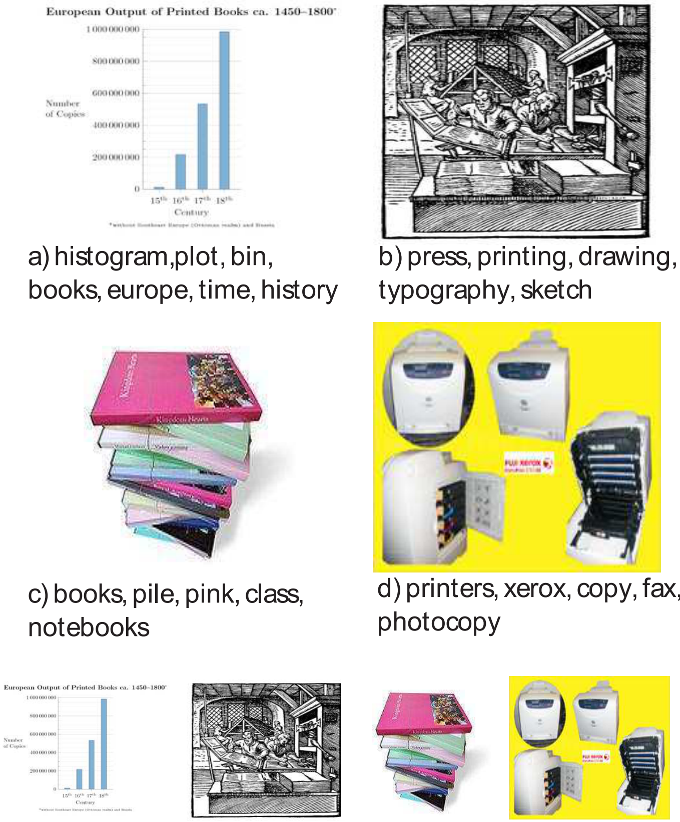
3.6 Summary

We proposed a new image representation that captures the structure of the image content, a methodology to extract the representation from existing object detection datasets, and algorithms for learning and inference of the proposed representation that allows to improve the results of object detection when a large number of categories are included. We performed our experiments in the challenging Visual Genome dataset obtaining large improvements, as measured by the mean Average Precision. The improvement is even kept on images from another dataset, as showed in the test set of the SUN2012 dataset. We showed that the proposed approach also allows to capture which ones are the most important objects, and obtain a model for the overall content of the image.

CHAPTER 4
CONTEXT DISCOVERY FROM MULTIPLE IMAGES

Humans are able to extract information and offer an interpretation from images. The interpretation of an image is based on the perception and previous knowledge of the person that is looking at the image. For this reason, a single image could have multiple interpretations that may lead to ambiguity. What if instead of having only a single image to be interpreted, we have multiple images that represent the same topic? In that case, we should expect that the information extracted from multiple query images can be leveraged to resolve the ambiguity, determine the right context and have a better description of the set of images.

In the previous chapter, we investigated the use of semantic context for single images using a powerful representation to model the image composition, and learn and infer about element co-occurrences and typical structures that these elements form. In that case, the context was used as a tool to gain new information from the image. We now attempt to solve the opposite problem. Given a set of visual elements (images), we want to find the shared context (a topic) that can be commonly associated with these multiple images.



a) histogram,plot, bin, books, europe, time, history

b) press, printing, drawing, typography, sketch

c) books, pile, pink, class, notebooks

d) printers, xerox, copy, fax, photocopy

e) printing, books, print, press, color, type, image, prints, lines, paper

Figure 4.1: Problem description. Subfigures a) to d) show four different query images and possible text associated to the images. e) shows the expected labels that describe the topic represented by the images.

Figure 4.1, from a) to d), shows four possible query images and a set of words that could describe them individually. Many of those words are ambiguous. For example, figure a) could be related to a histogram or a plot, instead of a printing process. Since we know all these images describe the same topic, our goal is to use the tags of each query image to find a more precise set of words that describe the topic. Text in e) shows a set of words that properly describes the topic of the four images.

In our approach, each query image is sent to a traditional image retrieval system that produces a group of retrieved images. Each retrieved image contains noisy (possibly incorrect) tags. These tags are consolidated into one word list per query image. These individual word lists are merged and processed by a word selection algorithm, with the aim of removing noisy labels. The output is a set of words that properly describes the topic of the queried images.

We believe that this type of approach, where multiple related images are used as queries to find the topic that is described, has a lot of potential for several practical applications, where images can be passively captured without direct interaction of the user like in the case of wearable devices and cellphones. Examples of possible applications include: context discovery, reduction of the search space for object detection, enabling semantic search by improving search accuracy through understanding searcher's intent according to the visual context, video summarization, among others. This chapter describes in detail an application on image retrieval from multiple images. Making use of the main ideas presented to find the context shared from multiple images, the proposed application allows to retrieve images that capture the semantic concept that is common between query images, but also are visually related to them.

The rest of this chapter is divided as follows. The first section explains in detail the proposed framework to find a set of words that properly describes the topic that a set of query images is representing. We then present an application for semantic image retrieval from multiple query images. Finally, we perform experiments to demonstrate the validity of the proposed framework in a dataset of 300 selected topics with up to 20 images per topic, and results of the proposed image retrieval application.

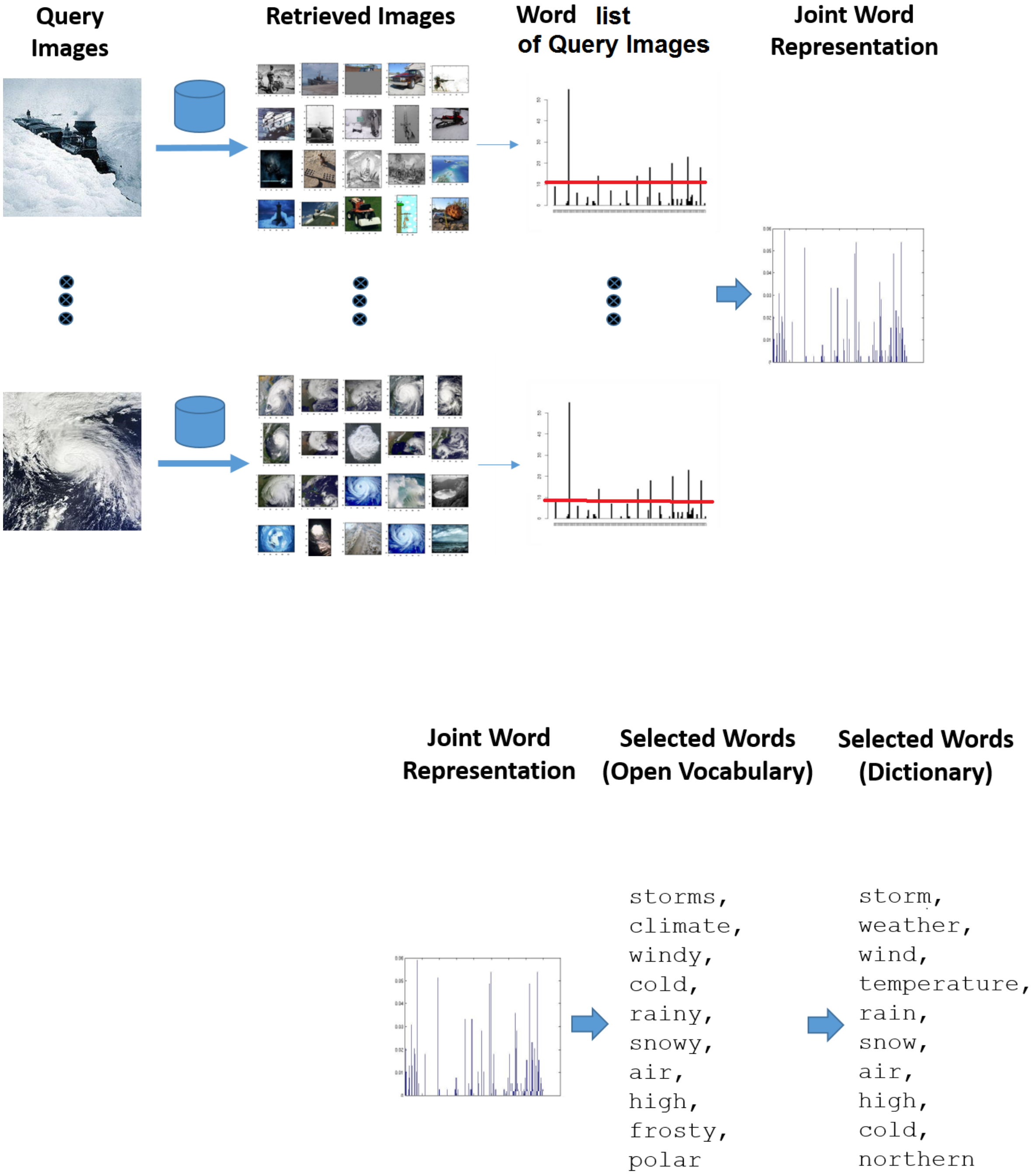

Query Images
Retrieved Images
Word list of Query Images
Joint Word Representation
Joint Word Representation
Selected Words (Open Vocabulary)
Selected Words (Dictionary)
storms, climate, windy, cold, rainy, snowy, air, high, frosty, polar
storm, weather, wind, temperature, rain, snow, air, high, cold, northern

Figure 4.2: General Framework of the proposed approach. A database that contains images with weakly labeled text descriptions is used to retrieve images similar to query images. The labels obtained from the retrieved images are weighted for each query image. A list of candidates words is produced after thresholding. A histogram is created from the lists of words of each image query. This histogram is the joint word representation. After that, a word selection algorithm determines a set of words that best describes the topic of the query images. Finally, the resulting words can be mapped to a reduced vocabulary of a dictionary.

4.1 Finding the Context Shared by Multiple Images

In this section we present our approach to find the context (topic) shared by a set of images. We accomplish this task by a combination of transferring tags from retrieved images, and performing operations in the verbal space with the tags.

Figure 4.2 presents the general framework of our approach. Images that are visually similar to each one of the query images, are retrieved from a database. Every retrieved image is accompanied by a set of tags. In previous approaches for image auto-tagging, the tags are restricted to a pre-set vocabulary with less than several hundreds words. In contrast, our retrieved images are accompanied by tags from the whole English language. Each query image generates a set of retrieved images. Each retrieved image has an associated set of tags. Each of the tags in the retrieved images list receives a weight that is dependent on the similarity between retrieved image and query image. At this point, each retrieved image is accompanied by a set of weighted tags. Next, for this same query, its retrieved images' tag list are merged into a word list. This merger of separated tag list (one tag list per retrieved image), will contain duplicates tags possibly with different weights. Duplicates are combined into one unique entry in the word list by adding up their weights. This result in a word list, each word has a weight that came from prior weights from the tag lists. A threshold is used to discard words whose accumulated weight is weak. The surviving words constitute the final word list for that query image. We discarded the weight of the surviving words, since these weights are no longer needed. We merge the word list of each query image into a joint word list that records the frequency of the words.

4.1.1 Image Retrieval

Our approach relies on Content-based image retrieval (CBIR) to retrieve images using visual features. As was suggested by Krizhevsky, and later confirmed by Babenko *et al*. [100], the features emerging in the upper layers of the CNN, learned to classify images, serve as good descriptors for image retrieval. Although global CNN features were not enough to improve the state of the art for image retrieval, the reported performance is very competitive compared against the local features alternatives that require more time to compute. We use features extracted from a Convolutional Neural Network (CNN) for image retrieval. In particular, we use the Krizhevsky network trained over ImageNet dataset, and removed the last layer of the network of fully connected units. Features are then computed by forward propagation of the mean subtracted and re-scaled to 224×224 RGB images. The result is a 4,096 dimension vector that represents the image as a global descriptor. The retrieval results were ordered using Euclidean distance.

4.1.2 Final Word List of a Query Image

Each image in the retrieval dataset has a visual representation given by a global image descriptor and an associated text that serves to link the visual representation with the searched topic description. Some texts contain mistyped words, words in languages different from English, or even texts in other alphabets that are discarded. There are also English words like pronouns, determiners, and alike that are extremely frequent over the whole dataset, but do not help to identify the topic.

A stop list is created to ignore these type of words from the text associated to the images of the dataset.

Retrieved images that are visually closer to the query image have a higher impact on the word list that represents the query image. Hence, the top retrieved candidates are weighted using an inverse exponential function of their visual distance to the query image,

$$a_{I_k} = \mathrm{e}^{-\|Q-I_k\|_2 \tau}, \tag{4.1}$$

where a_{I_k} is the weight for the words associated to the image I_k, $\|Q-I_k\|_2$ is the Euclidean distance between the visual features of images I_k and query Q, and τ is the exponential time decay constant. We have defined the exponential time decay constant τ as a linear function of the mean Euclidean distance of the best top 20 retrieved images of each query image. Due to this, the τ value varies according to the quality of the image retrievals for different query images.

An entry of the word list representation of a query image is computed as the weighted sum of all the word repetitions among the retrieved images, where the weight a_{I_k} is obtained from the exponential function described in equation 4.1. Words that contribute less than 0.1% of the total weight are considered noisy and discarded. The surviving words are the final word list of a query image. Each word of this list is a reasonable candidate to represent the topic, hence the individual weights of these words are not longer used.

4.1.3 Word Selection Algorithm

A joint representation is generated from the individual word list associated to the image queries. A natural way to choose the best set of words that represents a topic is to use the word frequency. Words that are more frequent among candidate words are selected as descriptors of the topic. In our experiments, we use this word selection method as our baseline algorithm.

An inherent limitation of this approach is that every word is examined separately, ignoring that words describing a topic should have high semantic similarity among them. In order to overcome this limitation, we consider a mechanism to determine the semantic similarity between two words. A word can be represented in a continuous dense vector space that captures semantic knowledge learned in the text domain. The skip-gram and the Continuous Bag of Words (CBOW) model architectures proposed by Mikolov *et al.* [101, 102] efficiently learn the semantically-meaningful decimal valued representations of words from very large text datasets. The intuition behind these models is that given a big corpus of text words, words that are semantically connected tend to appear close to each other in the corpus. The Continuous Bag of Words (CBOW) architecture model has V inputs corresponding to the vocabulary size of the corpus, and the same number of outputs. Each input is connected to a second layer of dimension D, that is shared by all the words of the vocabulary by means of a linear projection. The input projections are averaged and connected to the output layer. The objective is to predict a word given the immediately preceding and following words. A hierarchical softmax function encoded as a Huffman binary tree is used to efficiently reduce the amount of processing. Each word is a leaf of that tree, that has a path from the root to itself. In this way, N-way normalization of the softmax is replaced by a shorter sequence of $O(logN)$ local (binary) normalizations. All of the text of the corpus is serially used to train this network. Input words that are activated (words around the one that wants to predict) are labeled as one, and their projection weights are updated doing forward and backward propagation. The resulting projection of the word into the D-dimensional space becomes its vector representation. The skipgram model is similar to CBOW model, but in this case, an input word is used to predict the word before and after it. Both of these models, quantify the semantic similarities between two words W_i and W_j as the cosine distance between the two vector representations,

$$d(W_i, W_j) = \frac{W_i \cdot W_j}{||W_i|| \, ||W_j||}. \tag{4.2}$$

All the candidate words from the joint final word lists are considered equally important. However, words that are part of a topic, are expected to show a high semantic similarity among them. Therefore, we use Random Walks to score individual words, and discover a reliable subset of words considering the semantic relations between them.

We define a graph $G = (\mathbf{N}, E)$, where $\mathbf{N}$ represents the nodes, and E the edges. Each node represents a candidate word $\mathbf{N} = \{W_1, W_2, \ldots, W_V\}$. The initial score of the node is given by the frequency of the word using the text representation of the query, and there is an edge between each pairs of nodes $E = \{(W_i, W_j), i \neq j\}$.

We perform random walks on the constructed graph and update the scores of the nodes using their pairwise similarity given by equation 4.2. As a result, semantically related nodes tend to obtain higher scores. If a random node mistakenly gets a high score due to noisy image retrievals, its score will decrease because of its lack of semantic similarity with other candidate words.

Each random walk iteration will update the scores vector X using the following equation

$$X^{t+1} = \alpha X^{t} + (1-\alpha)X^{0}, \tag{4.3}$$

where α is a constant between zero and one, and is set to specify the contribution of the initial score versus the pair-wise similarity. The words with higher scores are selected to represent the topic.

4.1.4 Mapping to a Space Spanned by a Dictionary

In this step, we perform a mapping of the obtained words in the previous step to words that are part of a dictionary provided by the system. The previous step may generate uncommon or obscure words that are inappropriate as an output for the user. By using this dictionary, the system can choose what words to deliver to the user. The dictionary is also helpful to delimit the output words, in cases where the system has a previous knowledge about the user areas of interest.

Given a target dictionary $\Gamma = \{w_1, w_2, \ldots, w_M\}$, the objective is to find a mapping $\delta : \mathbb{O} \rightarrow \Gamma$ that maps the words from an open vocabulary $\mathbb{O} = \{o_1, o_2, \ldots, o_N\}$ to the words in the dictionary Γ.

Mapping a word from an input domain to the target domain has to preserve the meaning of the original word. A possible method to measure the distances between words from a semantic point of view was described in section 4.1.3. We define the baseline mapping $\delta_{base} : \mathbb{O} \to \Gamma$ as the transformation that for any input word, produce its closest word in the target dictionary. The distance between words are computed using the cosine distance in the vector space projection of the words (equation 4.2).

The aforementioned transformation maps individually every input word to a new word from the dictionary. It is important to notice that the input words have strong semantic similarity, since these words are describing a topic. However, in the baseline mapping, there is no guarantee that the resulting words have strong semantic similarity among them.

Hence, we propose a new mapping, that in addition to finding similarities between input and output words, also enforces strong semantic similarity among mapped words in the output space. The presented mapping is based on a Conditional Random Field (CRF) formulation.

Let L be the cardinality of the set of words that we want to map, we define a node for each of these L words. Each node has M possible scores corresponding to the distances from the node to each one of the M words of the output dictionary Γ. The edges between the nodes determine which nodes have the semantic relations. Given that all the words belong to the same topic, these nodes are fully connected.

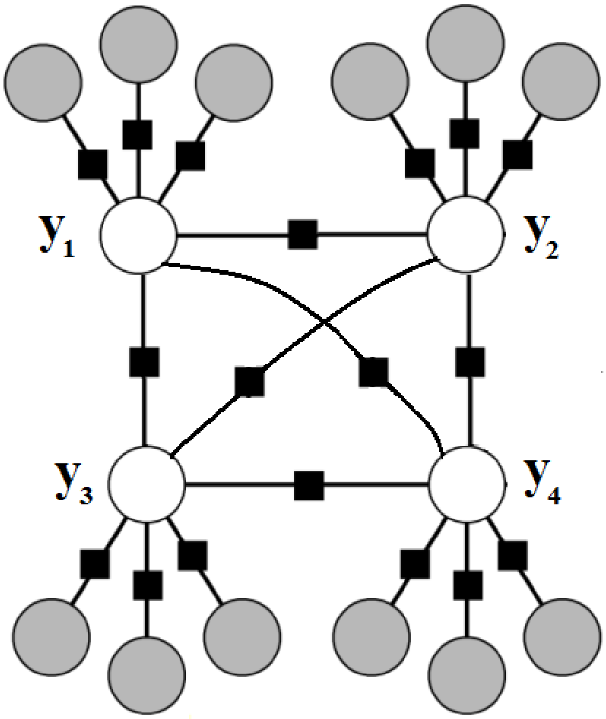


Figure 4.3: An example of a Graphical Model representation used to map the obtained set of words to a defined dictionary. The figure shows the scores of the possible word assignments as shaded nodes and label assignments as the white nodes. In this example, the size of the output dictionary M is 3, and the number of words mapped, L, is 4.

Figure 4.3 shows the graphical model used in our formulation when a set of four words is mapped and the dictionary size is three words. The white nodes represent the final word label assignments for the set, and the shadowed nodes represent any of the M word possible assignments for the particular node.

Let $Pr(\mathbf{y}|G;\lambda)$ be the conditional probability of the word label assignments $\mathbf{y}$ given the graph $G(S_L, Edge)$ and a weight λ, we need to minimize the energy equation

$$log(Pr(\mathbf{y}|G;\omega)) = \sum_{s_i \in S_L} \psi(y_i|s_i) + \lambda \sum_{s_i, s_j \in Edge} \phi(y_i, y_j|s_i, s_j), \quad (4.4)$$

where ψ are the unary potentials, and ϕ are the pairwise edge potentials. In our problem the unary potential is computed from S_i, the cosine distance between input word and a word in the output dictionary w_i, as

$$\psi(i) = 1 - S_i, \tag{4.5}$$

which privileges word labels with high similarity to the original word.

The pairwise edge potential is given by a matrix, $V(y_p, y_q)$, that determines the distances between the words that belongs to the dictionary Γ as measured by relation $1 - d(W_i, W_j)$. The matrix attempts to penalize words that are not related semantically (assigning a penalty), enforcing the global similarity of the labeled words.

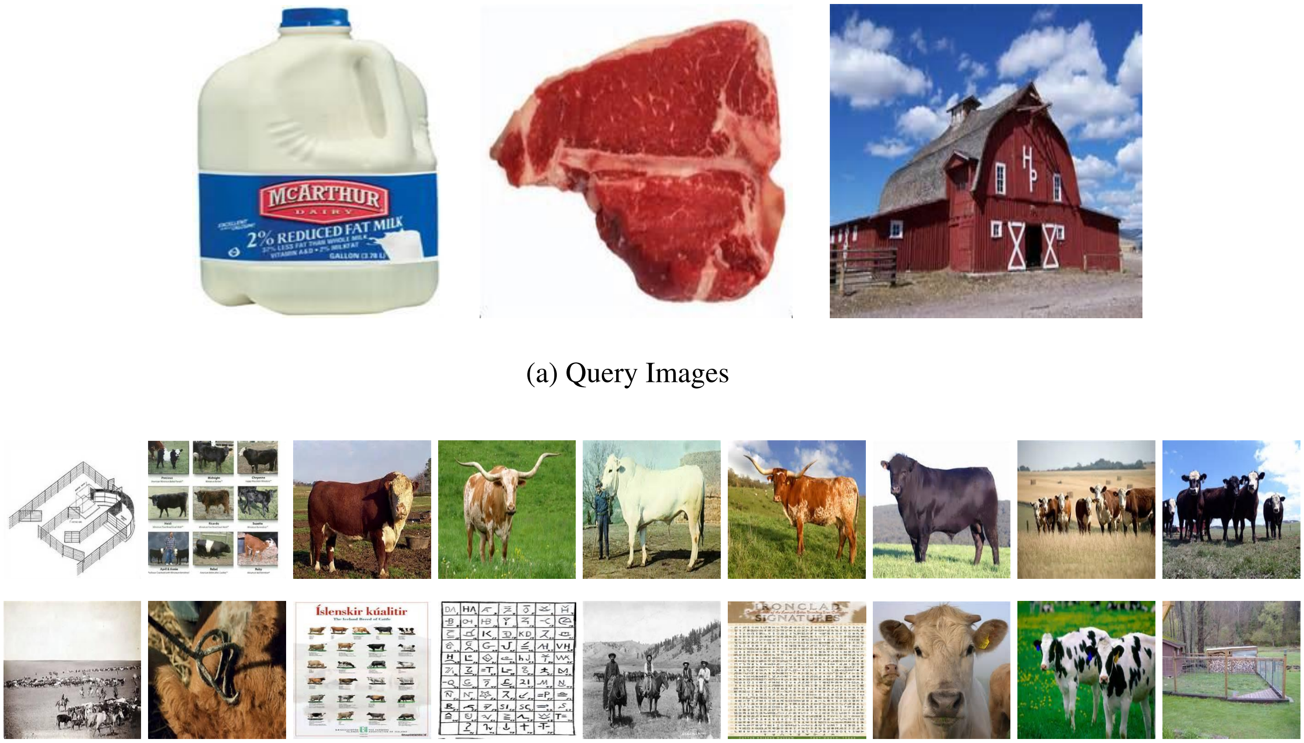
The energy function to minimize can be represented as:

$$E(\mathbf{y}) = \sum_{p=1\cdots N} \psi(p, y_p) + \sum_{p=1\cdots N, q=1\cdots N} \lambda_{p,q} V(y_p, y_q), \tag{4.6}$$

where $\lambda_{p,q}$ is a weighted adjacency matrix, with weights equal to $1/L$. We use the graph-cuts based minimization method in [103, 104, 105] to obtain the optimal solution for equation 4.6.

4.2 Application: Semantic Image Retrieval from Multiple Query Images

In this section, we propose a new application that uses the context extracted from multiple input images for retrieving images. The goal of any Image Retrieval system is to retrieve images from a large visual corpus that are similar to the input query. To date most Image Retrieval systems base their search on this paradigm for a single query image input. In the few cases where multiple query images have been used as input, the query images corresponds to the same object, scene or concept but with different viewing conditions such as pose or seasonal time.



(a) Query Images

(b) Retrieved Images

Figure 4.4: Example of the proposed paradigm with three input images. a) Input images. b) Top retrieved images.

In this work, we follow a completely different approach since the query images are employed jointly to extract underlying concepts common to the input images. It means that the query images do not need to be part of the same concept. In fact, different query images either enriches, amplifies, or reduces the importance of one descriptive aspect of the multiple significances that any image inherently has. Consider the example in Figure 4.4. Three input images are used to query the system. The first query image is a milk bottle, the second contains a piece of meat, and the third contains a farm. Apparently, these three images do not have anything in common from a visual point of view. However, conceptually they could be linked by the underlying concept “cattle”, since farmers obtain milk and meat from cattle. These types of knowledge based conceptual relations are the ones that we propose to capture in this new search paradigm. Figure 4.4b shows the top retrieved images by our system.

The utility of the proposed search paradigm is enhanced in cases where the user is not clear about what is being looked for or the user does not have the knowledge to describe it; but the user is able to provide some ideas in terms of related images. Consider for instance, a user looking ideas for a gift. The user add some pictures of vague ideas for the gift while walk thru the mall; then, based on the provided images, semantic concepts are found and used to retrieve images of gift suggestions.

Figure 4.5 presents a high level view of the steps used in the proposed method to retrieve images from multiple query input images. The approach is a variation of the framework presented in this chapter to find a topic from multiple images. Initially, each one of the query images I_i is processed individually to retrieve candidate images, based to the visual similarity, from the retrieval dataset. Each query image I_i produces a set of k nearest neighbor candidates, denoted as C_{ij}, where j goes from $1 \ldots k$.

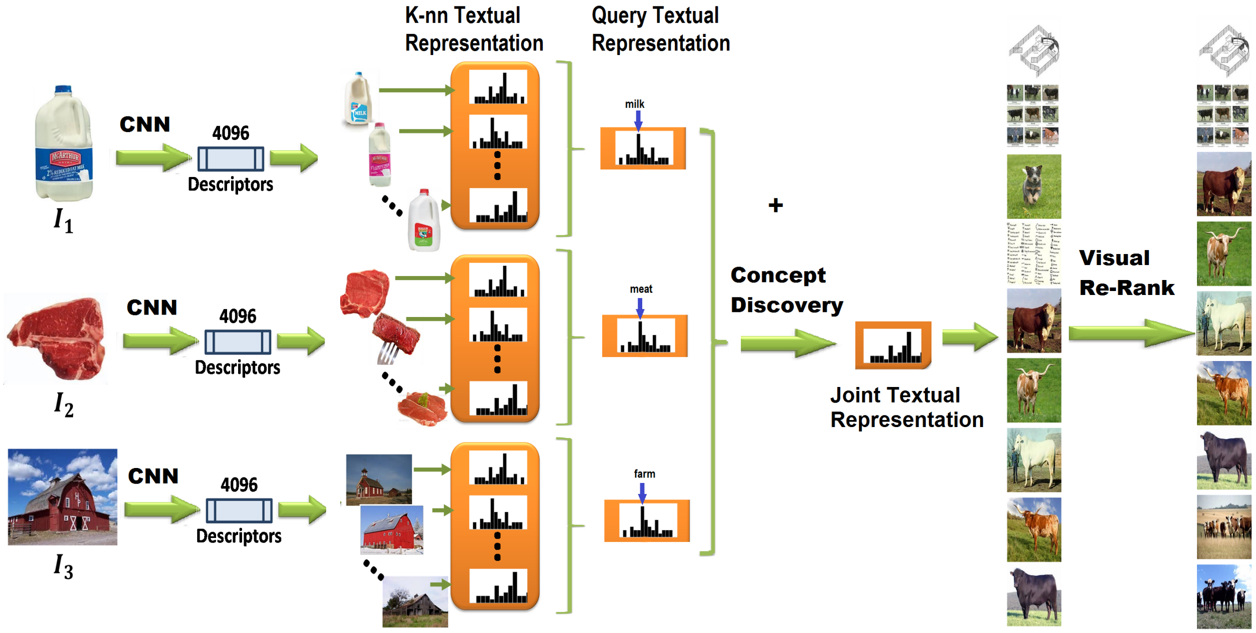


Figure 4.5: The proposed method to retrieve images from multiple input query images.

All of the images from the retrieval dataset have a dual representation: a visual representation given by a global image descriptor and a textual descriptor represented by a histogram of all of the texts describing the image. Hence, every candidate image C_{ij} has an associated textual descriptor represented by a word histogram that serves to link the visual representation with the conceptual representation.

Candidate images that are visually closer to the query image have a higher impact in the text representation of the query image. The most representative text words that describes the query images, are processed using Natural Language Processing (NLP) techniques to discover new words that share conceptual similarity. After that, the histogram summation of the weighted word representations of the candidate images C_{ij} and the aggregation of bins for the discovered words, produces a histogram that represents the queries of the search jointly. Later, cosine distance between the word representation of the database images and the joint word representation is performed to retrieve images. Finally a re-ranking is performed to privilege images with high visual similarity to any of the query images.

We use Convolutional Neural Network (CNN) to generate a high level visual representation of the images. Features are calculated using the Krizhevsky network [39]. The result is a 4096 dimension vector that represents the image as a global descriptor. Since our image descriptor is global, we perform Locality-Sensitive Hashing (LSH) to have fast approximated nearest neighbor retrieval of candidate images. The presented scheme produces retrievals that are similar in appearance, but also that accounts for the meaning of the image.

The input to our retrieval system is a set of query images without tags, labels, text description, or metadata. In order to operate in the conceptual level, textual representation of the content of the query images must be generated. Textual representations of the top retrieved images from visual features are transferred to obtain a textual representation of each query image. The top retrieved candidates are weighted using a decreasing function of their visual distance to the query image as in Equation 4.1.

In the proposed multi-query input image retrieval system, we combine the text descriptor from each input image represented by a small set of words, to infer new words that capture the common meaning of inputs.

Discrete words are represented in a continuous dense vector space that captures semantic knowledge learned in the text domain. The skipgram and the Continuous Bag of Words (CBOW) model architectures proposed by Mikolov et al [101, 102] efficiently learn semantically-meaningful float point representations of words from very large text datasets. The cosine distance from equation 4.2 can be used in order to find the semantic similarity between two words of the vocabulary. A kernel K_{ij} that measures the similarity between any word i and any word j of the vocabulary, is calculated beforehand. The purpose of the kernel K_{ij} is to quickly find the similarity of any pair of words.

```
input :
    • BoW representation of the $N$ input images
    • Index of the sorted columns of kernel $K_{ij}$
output: List of new words conceptually shared by the N input images
initialization;
while $size(ListNewWords) == 0$ do
    increase number T of words used to create N-tuples;
    increase number of sorted terms to be intercepted;
    for All N-tuples from top ranked words do
        $V \longleftarrow$ Intersection(current N-tuple);
        if $size(V) \neq 0$ then
            add $V$ to ListNewWords
        end
    end
end
```

Algorithm 2: Algorithm to discover words conceptually shared by the input images from their textual representations.

The representation of a query image is typically formed by only tens of words. Let N be the number of query images. If one word is selected from each query image, an N-tuple of words is formed. We want to examine the N words of the N-tuple to discover new words that related them conceptually.

A word W_l is considered as a new word concept, when W_l has simultaneously high similarity to all the words of the N-tuple measured in the vector space. The kernel K_{ij} allows to find the similarity between any pair of words of the vocabulary. A row (or column) q of the kernel matrix K_{ij} is the distance of the word indexed by q to any other word of the vocabulary. Performing a descend sort operation and taking their indices in the selected column q, gives the word indices which are conceptually closer to the word W_q.

Given an N-tuple, we can find the closest words for all the words that are part of the N-tuple. If a word W_l is found in the top positions of all the sorted lists of the N-tuple words, then the word W_l is declared as a new word conceptually shared by the N-tuple. We call this procedure "intersection".

There are many combinations of N-tuples that could be formed, however the most interesting ones are the N-tuples created from words that have highest weights in the textual representation of each query image.

A small number T of words with higher weights are used to create the N-tuples. In case that no word shared is found in the given set of N-tuples; the number of words used to create N-tuples and the number of sorted terms to be intercepted is increased until at least one common word is found. Algorithm 2 summarizes this procedure.

Image retrieval is performed from a unique textual representation that accounts for all input images and the list of words conceptually shared by the input images. The textual representation of the joint query is the summation of the individual query input representations and the addition of new bins indexed by the list of words shared by the input images. Image retrieval is performed based on the ranking score produced by the cosine scoring algorithm [106] between the joint search representation and the word representation of the images of the dataset.

Instead of comparing with the entire set of images of the dataset, we use a shorter number of possible outputs to calculate the score of the images. This subset of images is formed by all the images of the dataset that contain at least one word of either the list of words conceptually shared by the inputs or the most representative words of each query input. Hence, the number of images to evaluate reduces from 1 million to just a couple of thousand candidate images in the case of the Bing dataset. The retrieved images are based on their conceptual ranking only. Therefore, a re-scoring of the retrieved images is performed based on the visual similarity to the input images. Images that are visually inconsistent with all the input images are penalized, and re-ranked to lower positions. The value of penalization is calculated using an inverse exponential function of the Euclidean norm of the difference between descriptors of the retrieved image and its most visual similar input image.

The resulting text representation of the images is very sparse since only a few tens of words describe each image. For this reason, text representation can be saved in a very compact way. In fact, the full dataset representation of 1 million images can be fully loaded in 51 Mb of memory .

4.3 Implementation Details

We use the dataset provided for the 2013 MSR-Bing Image Retrieval Challenge [107], which was sampled from one-year click logs of the Microsoft Bing image search engine. It consists of

1 million images. Click logs are produced in the following manner. Some user enters a query search, Q_j; Bing displays several images, $I_1, I_2, ..., I_N$; suppose the user clicks I_i; at this point, Bing increments the click count T_{ij} in a triad $< I_i, Q_j, click\, count\, T_{ij} >$. The dataset includes 23 millions such triads (A single triad may represent the multiple clicks of different users at different times during the year). A total of 11.7 million different query terms are available, and the number of clicks T_{ij} is always more than zero.

The dataset is very suitable for image retrieval since the texts associated to the images are fairly accurate, because they are built from user's search criteria and their click preferences. The most important advantage is that the image labeling does not require humans dedicated to this activity, and is the product of implicit crowdsourcing. Examining the click log files is evident that every image is linked to about 23 queries; consequently there is a significant amount of text describing each image. We used this dataset as our image retrieval dataset.

Our text representation of the images from the dataset, is a histogram of the words of all texts associated to an image (related by the 23 million triads). After removing words in languages different from English, words written in alphabets different from Modern Latin alphabet, and removing pronouns, determiners and extremely frequent words, we end up with a vocabulary of 77,488 words for the Bing Image Dataset.

We use the public domain implementation from caffe library [108] and the provided AlexNet trained model to compute the CNN visual features that represent the images of the database. We use as descriptor the $4,096$ dimension vector obtained by running forward the network and removing the last layer with 1,000 units. Visual feature extraction is performed very efficiently; image resizing and feature extraction of the 1 million images of the database can be performed in less than 24 hours on a regular Quad core personal computer. In order to have a fast approximated nearest neighbor retrieval implementation, we use Locality Sensitive Hashing (LSH), which is an algorithm for solving the approximate Near Neighbor Search in high dimensional spaces. LSH hashes input items so that similar items map to the same "buckets" with a high probability. The distance computation is only performed within the elements of the same "bucket" , increasing the retrieval speed. In all of our experiments we use 10 bits to generate the "buckets".

Once the nearest neighbors of the query image are computed, we weight each one of the words associated to the nearest neighbors images using an exponential decay function described in equation 4.1. Experimentally we found that a good value for τ is given by $\tau = \mathbb{E}_{1 \cdots 20}[\left\|Q_j - I_k\right\|_2]/3$, where $\mathbb{E}_{1 \cdots 20}[\left\|Q_j - I_k\right\|_2]$ is the mean distance of the 20 nearest neighbors to the query Q_j in the visual space.

Finally, the edges of the graph for a random walk, are computed using the equation 4.2, and the damp factor is fixed to 0.9. The results were obtained on the random walk corresponds to single runs. The mapping between any word and its vector representation is given by the projection matrix trained on a part of Google News dataset, which contains about 100 billion words[1]. The vocabulary size D of this model is 300.

[1] https://code.google.com/p/word2vec/

4.3.1 Evaluation Dataset

We use the dataset [2] provided in [109] to evaluate our experiments. The dataset consist of 300 topics represented by the top 10 most representative words for the topic. The topics were found by performing Latent Dirichlet Allocation LDA in a corpus of the "New York Times published between May and December 2010, and by randomly selecting Wikipedia categories from a hierarchy in a breadth-first-search manner starting from a few seed categories (e.g. sports, politics, computing) that have more than 80 articles".

A set of 20 images is provided for each topic. These images correspond to the top 20 images under the Creative Commons license from English Wikipedia, retrieved from a search of the top-5 terms from a topic using Google Search.

Additionally, the dataset provide human scores that judge how appropriate the image was as a representation of the main subject of the topic. The score allows to rank which of the twenty images are more representative of the topic according to human criteria. As we will see later, the order in which the query images are presented have an influence on the quality of the topic description.

[2] http://staffwww.dcs.shef.ac.uk/people/N.Aletras/resources.html

4.3.1.1 Jaccard Index as a Performance Metric

Given a set of query images of the same topic used as input, the result is a set of words that describes the topic. The ground truth for one topic of our experiments is also a set of words. Hence, for each topic, we need to compare the similarity of two sets of words. The Jaccard index, also known as the Jaccard similarity coefficient, is chosen as metric for the evaluation of our algorithm. The Jaccard index measures the similarity between finite sample sets, and is defined as the size of the intersection divided by the size of the union of the sample sets:

$$J(G,O) = \frac{|G \cap O|}{|G \cup O|}, \tag{4.7}$$

where G is the set of groundtruth words that describes the topic, and O is the output set of words found by the algorithm.

4.3.2 Experiments

Figure 4.6 presents an example of the proposed approach. In that case, fifteen images are used as query images. After performing the image retrieval, and weighting their associated words, we end up with a huge set of words to describe each query image. In the Figure 4.6, only the top scored words are shown. Most of these words are written in red. A word written in red indicates that it is not part of the groundtruth for the topic. The joint word representation, has five matching words with the groundtruth among their top 20 obtained words. However, the joint word list contains some elements, which are semantically incoherent like “vietnam” or “fishing”. The random walk algorithm takes care of unrelated words, increasing the number of matching words from the top-10 list to four. Finally, in the last step, we complete a mapping of the found words to a smaller dictionary. In all of our experiments, we used a dictionary of 1,653 words, created by concatenating all the groundtruth words available in the list of 300 topics. As a result of the mapping step, the word “texas” is relabeled as “missouri”, increasing the number of correctly matched words to five, while other retrieved word like “area” is mapped to “north” that was already found as topic descriptor, decreasing the number of false positives.

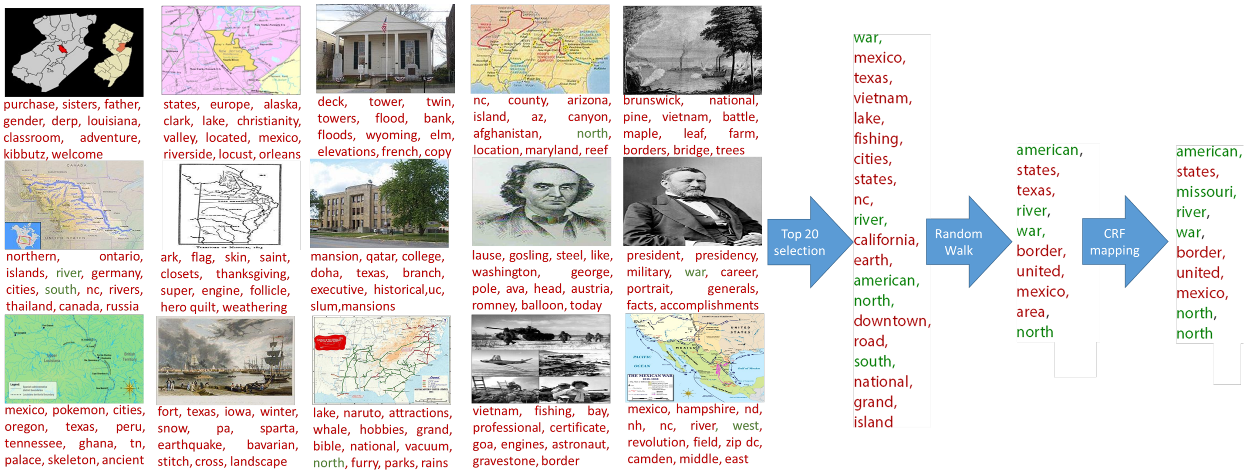
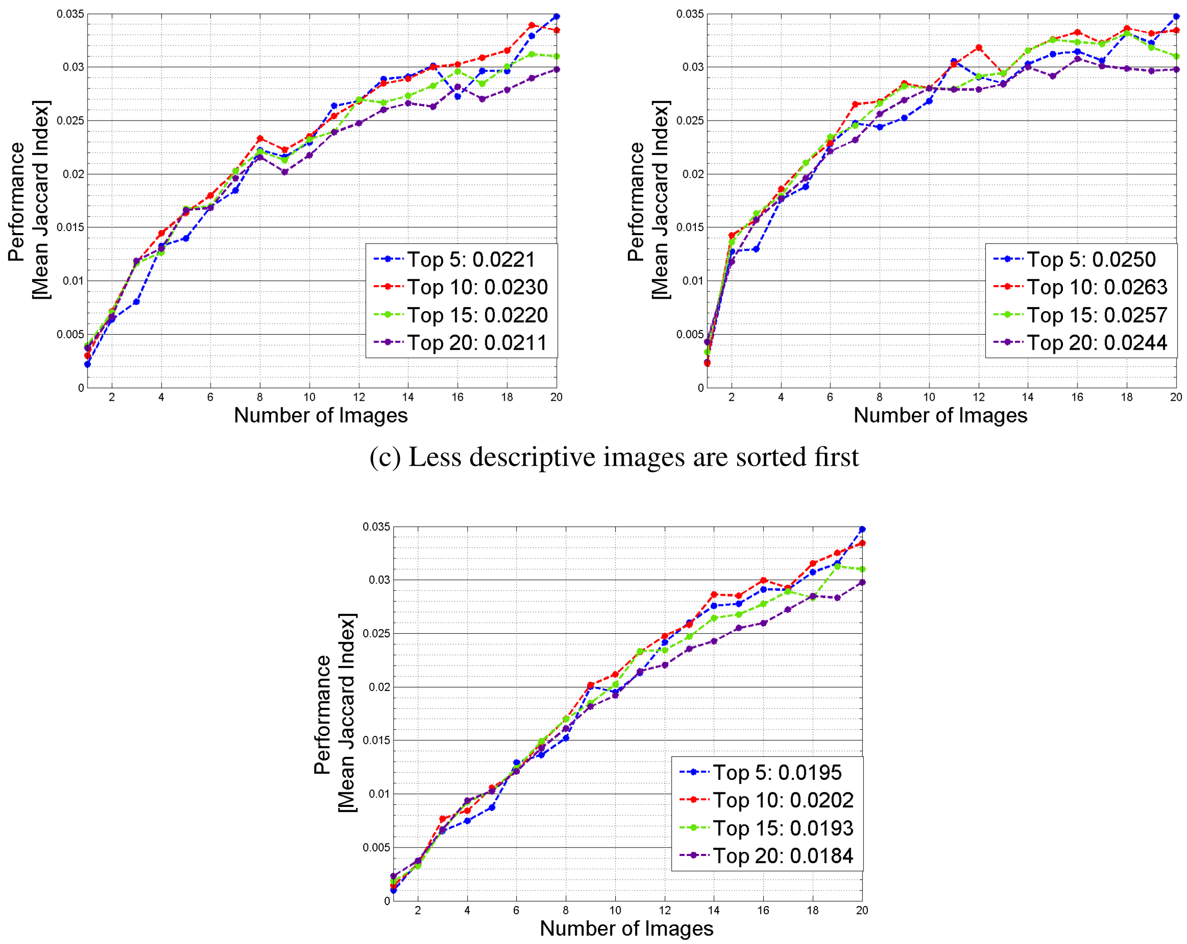

purchase, sisters, father, gender, derp, louisiana, classroom, adventure, kibbutz, welcome
states, europe, alaska, clark, lake, christianity, valley, located, mexico, riverside, locust, orleans
deck, tower, twin, towers, flood, bank, floods, wyoming, elm, elevations, french, copy
nc, county, arizona, island, az, canyon, afghanistan, north, location, maryland, reef
brunswick, national, pine, vietnam, battle, maple, leaf, farm, borders, bridge, trees
northern, ontario, islands, river, germany, cities, south, nc, rivers, thailand, canada, russia
ark, flag, skin, saint, closets, thanksgiving, super, engine, follicle, hero quilt, weathering
mansion, qatar, college, doha, texas, branch, executive, historical,uc, slum,mansions
lause, gosling, steel, like, washington, george, pole, ava, head, austria, romney, balloon, today
president, presidency, military, war, career, portrait, generals, facts, accomplishments
mexico, pokemon, cities, oregon, texas, peru, tennessee, ghana, tn, palace, skeleton, ancient
fort, texas, iowa, winter, snow, pa, sparta, earthquake, bavarian, stitch, cross, landscape
lake, naruto, attractions, whale, hobbies, grand, bible, national, vacuum, north, furry, parks, rains
vietnam, fishing, bay, professional, certificate, goa, engines, astronaut, gravestone, border
mexico, hampshire, nd, nh, nc, river, west, revolution, field, zip dc, camden, middle, east
Top 20 selection
war, mexico, texas, vietnam, lake, fishing, cities, states, nc, river, california, earth, american, north, downtown, road, south, national, grand, island
Random Walk
american, states, texas, river, war, border, united, mexico, area, north
CRF mapping
american, states, missouri, river, war, border, united, mexico, north, north

Figure 4.6: An example of qualitative results obtained by the proposed approach. Words written in green indicate words that match with the groundtruth, while words written in red indicate the words that do not match with any word in the groundtruth. The fifteen images on the left are used as query images to search for the topic. Each query image includes a text showing the top retrieved words associated to the query image. The top 20 highest scored words of the joint representation are shown next. After that, re-ranking is performed using random walk, removing words with low semantic coherence like “vietnam”, “earth”, and “island”. Finally, a mapping to a closed vocabulary is conducted which allows us to transform the word “texas” to “missouri”.

For the quantitative experiments, the performance is reported as a plot of the average Jaccard index of the 300 topics as a function of the number of input images. The order in which the images are presented has an impact on the performance, since some images are more significant than others in describing the topics.



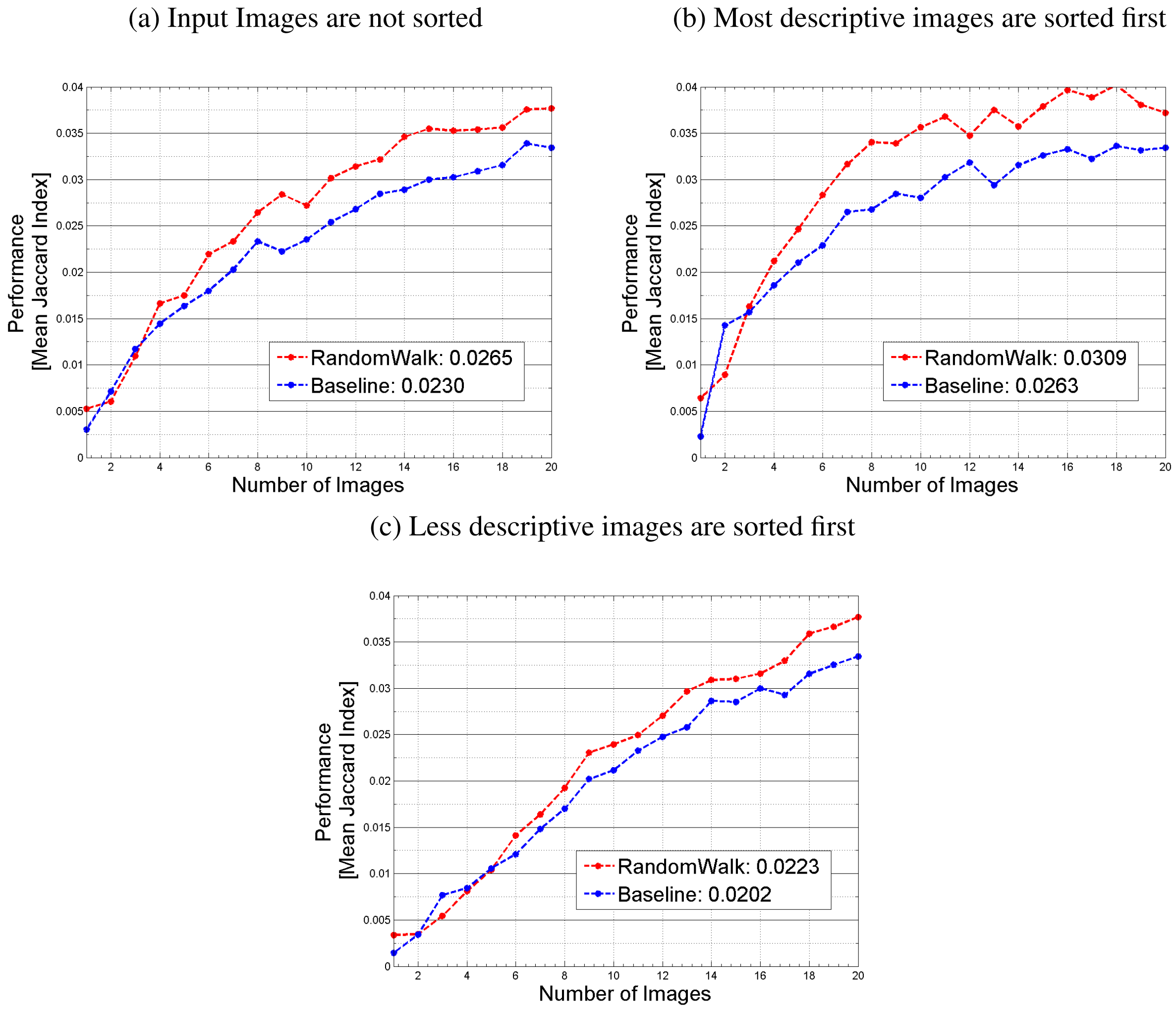
(a) Input Images are not sorted

(b) Most descriptive images are sorted first

(c) Less descriptive images are sorted first

Figure 4.7: Results obtained using the joint word representation when the top 5, top 10 , top 15, and top 20 retrieved words are used to represent the topic. The displayed plots show the performance (Y axis) computed as the mean Jaccard index of the 300 topics, as a function of the number of images used as query. The three plots represent different sorting of the input images, according to how representative the images are for the topic. Retrieving top 10 words produce the best results for any type of sorting.

We use the provided human annotations in the groundtruth, that describes the relevance of an image for the topic, to generate three different sorting of the input images. They are: a) images preserve the original dataset order; b) Images are sorted in decreasing order, using the most descriptive images of the topic first, and c) Images are sorted in increasing order, using the least descriptive images of the topic first.

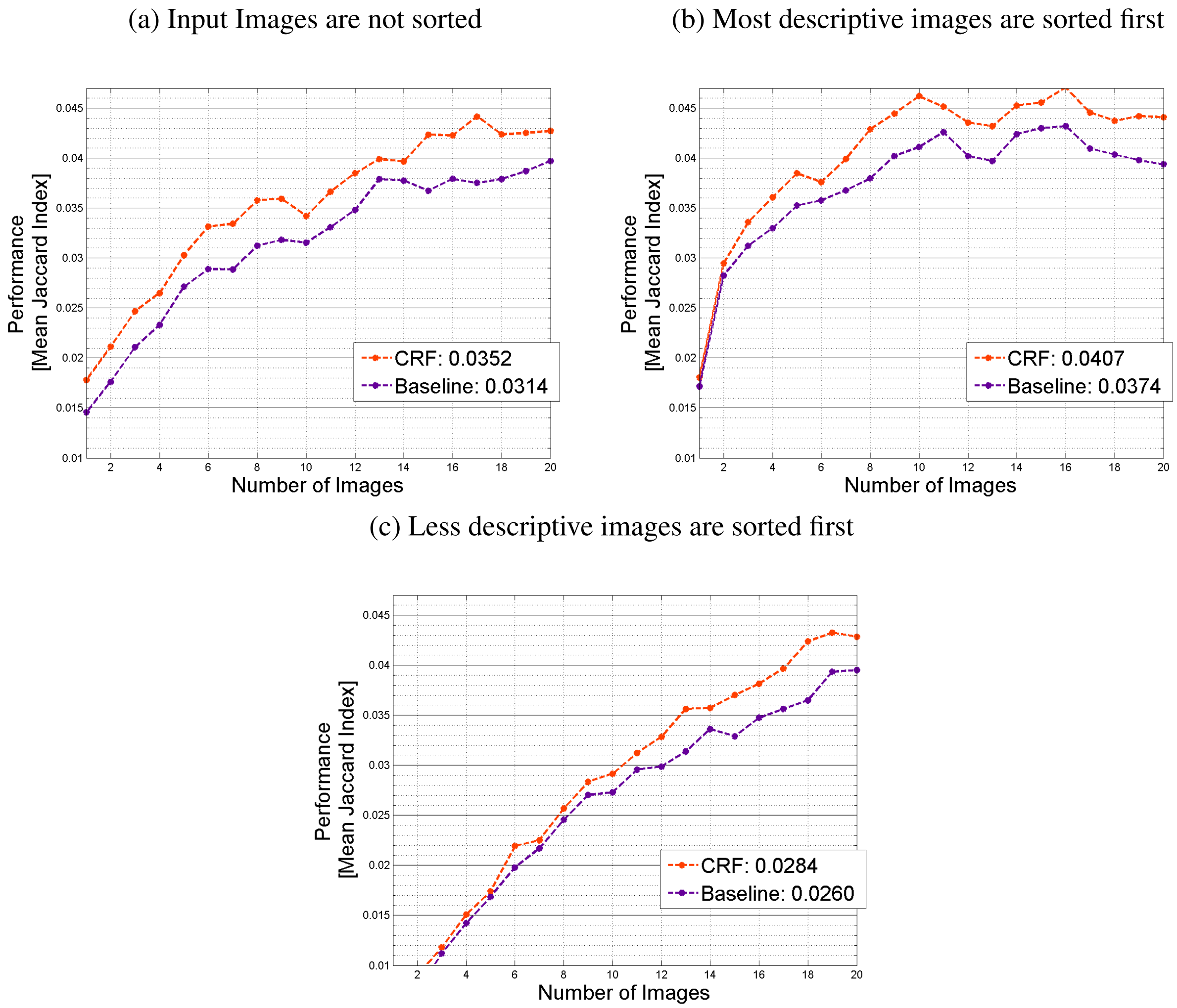

(a) Input Images are not sorted

(b) Most descriptive images are sorted first

(c) Less descriptive images are sorted first

Figure 4.8: Comparison between the baseline method (joint word representation using top 10 retrieved words) and our random walk algorithm, to select the best words that describe the topic. The three plots represent different sorting of the input image based on how representative the images are for the topic. The random walk algorithm outperforms the joint text representation under any type of sorting, and any number of input images.

As is evident from Figures 4.7, 4.8, and 4.9 the order in which images are presented to the system has an impact on the performance. For the same number of images, sorting the most descriptive images first have better results. Also, as is expected, better topic descriptions are achieved when a larger number of query images are used.


(a) Input Images are not sorted

(b) Most descriptive images are sorted first

(c) Less descriptive images are sorted first

Figure 4.9: Evaluation of the presented algorithm to map a set of words from an open vocabulary to a dictionary. The output of the random walk algorithm is mapped using two algorithms: the baseline and the proposed CRF algorithm. The baseline maps every output word to its closest word as it is measured by the cosine distance in the vector space of words. The CRF algorithm outperforms the baseline under different sorting conditions and the number of query images.

Figure 4.7 shows the results obtained extracting the output words from the joint word representation. Each plot contains results varying the number of retrieved words that describe the topic from 5 to 20 in intervals of 5 words. For the three types of sorting, retrieving the top 10 words produce the best results according to the mean of the average Jaccard index for different number of input images. We consider 10 retrieved words for the remaining experiments. Figure 4.8 compares the results of our method before and after applying the random walk algorithm on the joint word representation. As is evident from the figure, the random walk algorithm outperforms the joint word representation under any type of sorting, and the number of input images. Finally, Figure 4.9, evaluates the performance of the algorithm to map a set of words from an open vocabulary to words of a dictionary. We used as starting point the output of the random walk algorithm of the Figure 4.8, which is mapped using a baseline mapping method and the CRF based mapping algorithm. The baseline maps every output word to its closest significant word in the dictionary according to the cosine distance in the vector space of words. The CRF mapping algorithm preserves the semantic similarity under different sorting and amount of query images, outperforming the results of the baseline mapping method.

4.3.3 Application: Multiple Query Image Retrieval

We performed experiments to evaluate the proposed image retrieval application using a set of 101 pairs of image inputs. The definition of the input pairs of images was performed with the help of semantics maps downloaded from the internet [3].

[3] http://www.bing.com/images/search?q=semantic+mapping

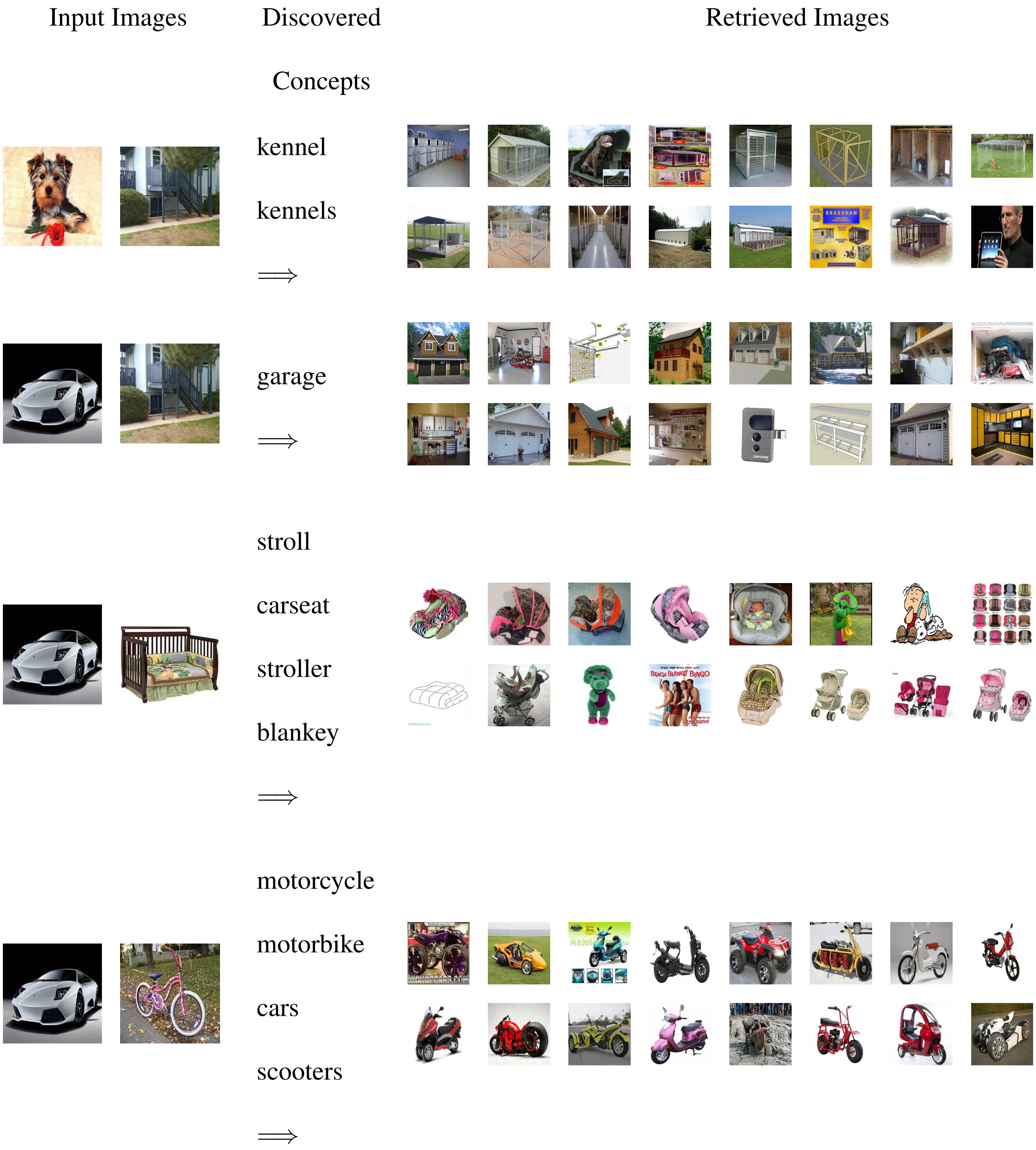


Figure 4.10: Examples of image retrieval using two input images. First column shows the input images, second column shows the discovered words conceptually shared by the input images, and third column shows the retrieval after visual re-raking.

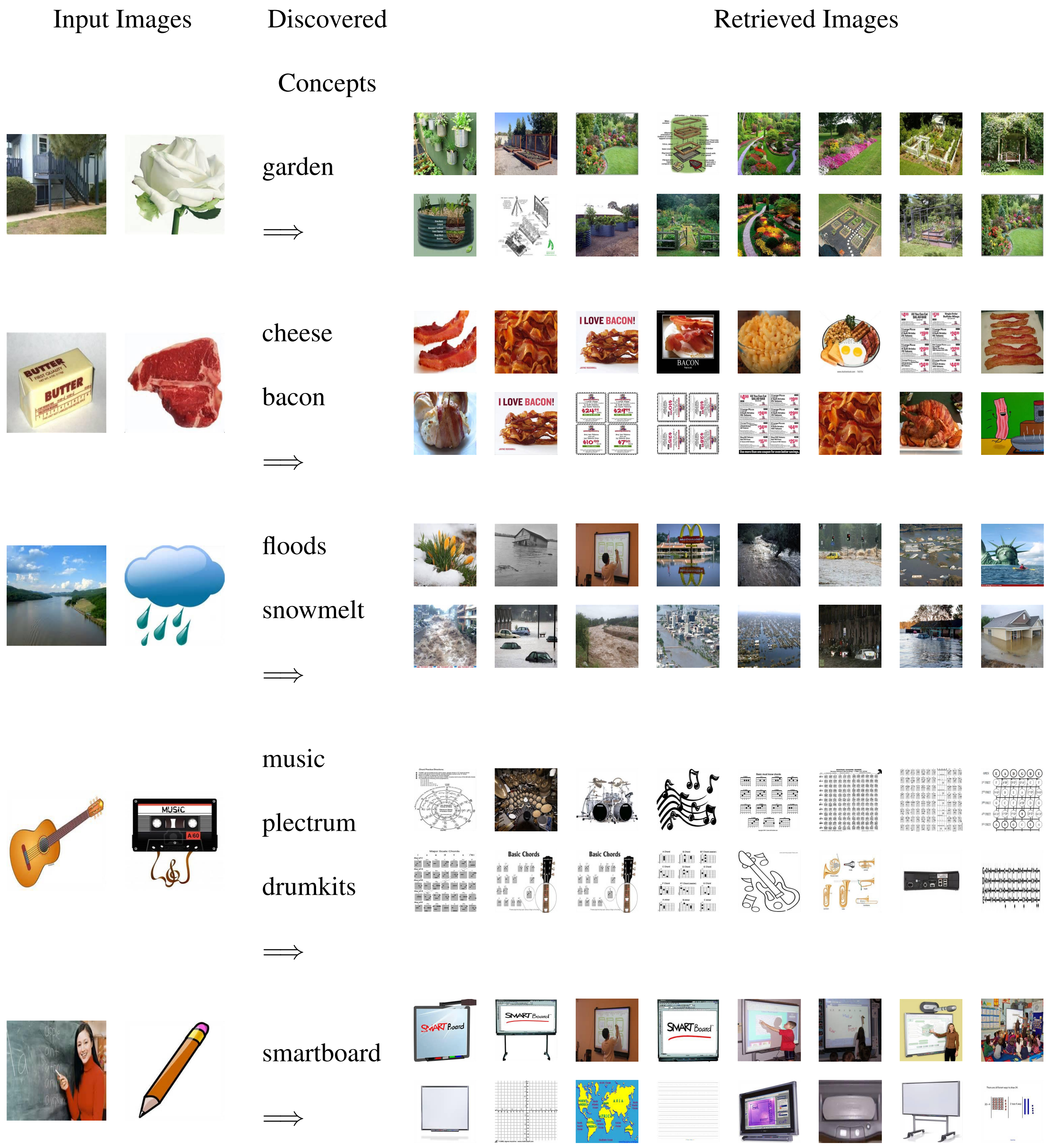


Figure 4.11: Examples of image retrieval using two input images. First column shows the input images, second column shows the discovered words conceptually shared by the input images, and third column shows the retrieval after visual re-raking.

A semantic map is a visual strategy for vocabulary expansion and extension of knowledge by displaying words and their relations with other words[110]. Any person can define a semantic map about any topic; therefore there is no “correct” semantic map. We chose several sets of pairs of words that were related according to any of the semantic maps found. Words that were not easily represented pictorially were discarded, and the remaining words were used to download example images and form query pairs of images conceptually related.

Figures 4.10 and 4.11 shows pairs of input query images defined in this way, and the top images retrieved by our system.

Table 4.1: Mean accuracy of the retrieved images acording to user ratings in 101 pairs of query images. Results are showed at different top retrieval levels.

Method	Top 5	Top 10	Top 15	Top 20	Top 25
Baseline (Before visual re-ranking)	0.5058	0.4843	0.4823	0.4774	0.4784
Baseline (After visual re-ranking)	0.5549	0.5294	0.5124	0.5093	0.5011
Ours (Before visual re-ranking)	0.7509	0.7450	0.7327	0.7172	0.7098
Ours (After visual re-ranking)	**0.7765**	**0.7579**	**0.7490**	**0.7358**	**0.7243**

For each pair of available query images, we asked several users to rate the top 25 retrieved images of a pair of query images. They were then asked to provide a binary answer to the following question: Is the retrieved image conceptually similar to both input images or not ?

Based on the users ratings, we calculated the mean accuracy of the retrieved images from the 101 pairs of inputs. Accuracy is reported for the top *X* retrieved images, with *X* ranging from five to twenty five in intervals of 5.

The baseline method is defined as the image retrieval performed from a joint search representation given by the summation of the individual input textual representations without the addition of words conceptually shared by the input images. Table 4.1 presents the results of accuracy of our method and the baseline. For both methods we include the results before applying visual re-ranking. Our method clearly outperforms the baseline results by more than 22% in the proposed task. The visual re-ranking also helps to improve the ratings of the users. The improvement is more significant in the baseline case, where the retrieved images are worse in the task of retrieving shared concepts. The previous observation is an indication that some users tend to rate positively near identical images when the conceptual meaning of the query images cannot be clearly established.

4.4 Summary

In this chapter, we presented the problem of identifying the topic that is being depicted by a group of images. We proposed an algorithm to obtain a set of words that properly describe the topic without human intervention. The approach relies on a combination of image retrieval, auto-tagging, and a random walk that takes into account the semantic similarity among the words. We also proposed a CRF based algorithm to map a set of words from an open vocabulary to a closed dictionary preserving the semantic similarity. The results indicates that the proposed algorithm clearly improves the proposed baselines.

We introduced a new search paradigm, where the representations of multiple input images are leveraged to infer concepts that are shared by all the input images. The retrieved images are conceptually and visually meaningful. We presented a complete solution to the aforementioned problem. The proposed approach achieved mean accuracy of 77.65% on top 5 retrievals and 72.43% on top 25, according to user ratings. The proposed approach outperforms the baseline by more than 22% showing that the method works very well in the proposed problem.

CHAPTER 5
IMPROVING UNDERSTANDING OF VIDEOS USING CONTEXT

Video understanding is a task even more complex than image understanding. Video presents new constraints that gives opportunities to improve the understanding. Typically in a video the scene identity remains stable for some frames, and this effect is even more pronounced in egocentric videos. Hence, we are interested in egocentric videos because they commonly contains smooth changes of the collection of objects and scene identities. In this chapter, we focus on two important building blocks of scene understanding in egocentric videos: improving scene identification by using temporal information, and improving object-detection through the utilization of the visual appearance of the scene (either scene identity or global context).

A property of egocentric videos is that the scene identity remains constant for several frames until the camera continuously moves to a different location. We use this egocentric video temporal consistency constraint to improve scene identification accuracy, employing a Conditional Random Field (CRF) formulation, which penalizes short-term changes of the scene identity. This formulation is covered in detail in section 5.1.

Assuming that we have a method for object detection that provides bounding boxes and their confidence scores, we show that it is possible to increase the performance of the detector by incorporating the information about the particular type of the scene for the frame that is being tested.

We learn from the training data, to modify the confidence scores of the object detectors according to the type of scene identity. Detection scores for objects that are unlikely to appear in a particular kind of scene are re-scored with lower values, while the scores of categories commonly associated with the type of scene are increased. Section 5.2 covers the details of improving object detection by incorporating information about the scene to re-score the original object detection results. We propose two approaches. The first one is a greedy algorithm, and the second is an algorithm based on Support Vector Regression (SVR).

Section 5.3 presents a framework for improving object detection scores that simultaneously considers the temporal information and the global context, with the additional benefit of not requiring an explicit scene labeling of the video frames.

Finally, section 5.4 describes experiments to demonstrate the impact of the proposed methods for improving the object detection and scene identification on videos. Quantitative and qualitative results are presented.

5.1 Improving Scene Identification

Given a set of training videos containing N_s type of scene identities, one scene classifier is trained for each type of scene. Under the assumption that other frames do not influence the scene identity of the current frame, each sampled frame is evaluated independently to determine the scene identity by comparing the scores of each one of the trained scene classifiers, and selecting the classifier with maximum score. However, we are dealing with egocentric camera videos where the scene identity of a frame is influenced by the identities of previous frames. It is evident that a person requires some time to move from one scene to another, therefore, if a person is known to be in a particular scene, it is very likely that the individual will remain on the same scene during some additional frames. We use a Conditional Random Field (CRF) formulation to implement the described temporal constraint of scene identities associated with egocentric videos. The goal of the formulation is to find the scene labels $\mathbf{y} = y_1, y_2, \cdots, y_N$ for a video sequence with N frames, that best fit the scores of the scene classifiers while enforcing the temporal constraint.

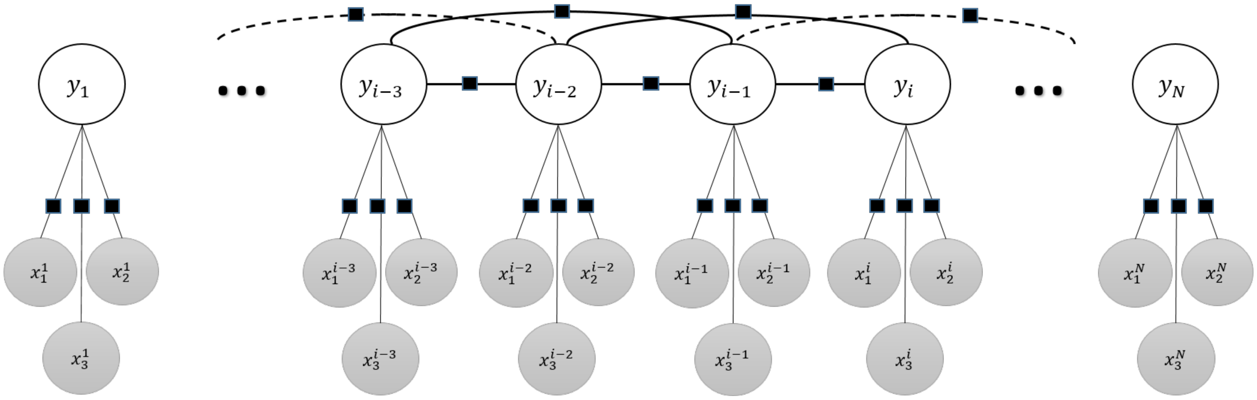


Figure 5.1: Example of a graphical model representing temporal dependencies for scene labeling in an egocentric camera video. A total of $r = 2$ previous observations and three possible scene identities are represented in the figure. The figure shows the observations (scene scoring) as shadowed nodes $x_{y_i}^i$ and label assignments as white nodes y_i. Experiments in section 5.4.1 were performed with $r = 7$.

We define a graph with scene label nodes y_i for each frame of the video, which are connected temporally through edges with their r neighbors frame labels. Each frame label has a number of possible observations (scene classifiers) associated $x_{j \in [1 \cdots N_s]}^i$. Figure 5.1 presents a particular case, where the two previous frames are connected. Let $Pr(\mathbf{y}|G; \omega)$ be the conditional probability of the scene label assignments $\mathbf{y}$ given the graph $G(S_p, Edge)$ and a weight ω, we need to minimize the energy equation 4.4, described in the previous chapter.

Similarly, the energy function to minimize can be represented as

$$E(\mathbf{y}) = \sum_{p=1\cdots N} \psi(y_p) + \sum_{q=1\cdots N} \sum_{p=1\cdots N} w_{p,q} V(y_p, y_q), \tag{5.1}$$

where $w_{p,q}$ is an adjacency matrix, that indicates which nodes are connected by edges and how much influence any of the r neighbor frames has on the current frame.

In our problem the unary potential is determined by a normalized scene classification score $\mathbf{x}^i_{y_i}$ as

$$\psi(y_i) = 1 - \mathbf{x}^i_{y_i}, \tag{5.2}$$

which privileges scene labels with high scores.

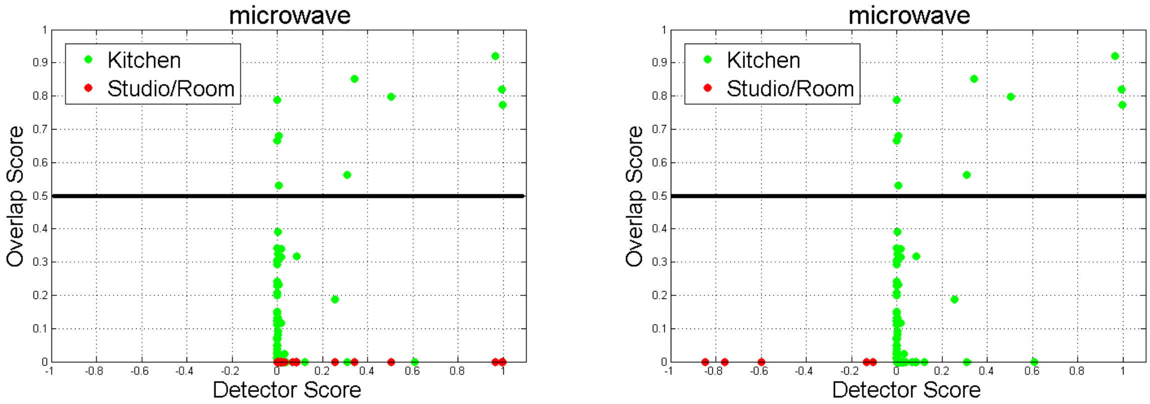
The pairwise edge potential is given by a matrix $V(y_p, y_q)$. The matrix $V(y_p, y_q)$ is defined with zeros in its diagonal, implying that the energy is not affected if the scene identity remains the same, and with positive values in positions off diagonal of the matrix to penalize changes in the scene identity. This enforces the temporal continuity of scene identities for frames linked by edge potentials in the graph.

We will discuss choices for matrix $V(y_p, y_q)$ and adjacency matrix $w_{p,q}$ in the experimental section.

5.2 Improving object detection

Object detection is the process of finding a set of bounding boxes that delimits the regions which contain the objects of interest. After running object detectors, the detection scores signify the matching between the visual model and the testing bounding box content. Typically, object detectors consider at most only the local context, which corresponds to the surrounding regions of the bounding box where the object is localized, but rarely examine global information about the scene. Consider a typical object used in ADL video, for instance, a microwave. A microwave is commonly found in the kitchen but is very unusual in other locations such as bedroom, bathroom or a laundry room. Consequently, in cases where it is possible to obtain information about the identity of the scene of the current frame, we could re-score the results of the object detector to penalize detections in scenes that typically do not contain the object that we are looking for. Overall, it is possible to increase the performance of the detector by incorporating the information about the particular type of the scene for the frame that is being tested.

The objective is to learn from the training data how much the detection score should be increased or decreased to account for the chances of having the object in a type of scene. The scene identity and the localization of the objects in every frame from the training videos of the Activities of Daily Living (ADL) dataset [76] are known in advance. Assuming that an object detector is available, we obtain bounding boxes and their associated detection scores. We also determine how much overlap exists between the candidate bounding box and the ground-truth bounding box of the searched object. The resulting measurement is called overlap score.


(a) Before Re-scoring

(b) After Re-scoring

Figure 5.2: Explanation of the main idea behind our method to improve object detection based on scene identity using training data of ADL dataset. Figures are generated from microwave detector, and show the detection score versus ground-truth match score. Figure a) shows the detections for the kitchen in green and the results for a bedroom in red. Figure b) shows a re-scoring that improves the object detection.

Figure 5.2 clarifies the concept behind our method. We focus on the microwave object in this discussion, but it applies to any other object such as, refrigerator, tv, bed, computer, etc. In all the subfigures, the X–axis represents the detection scores produced for the different candidate bounding boxes, and the Y–axis represents the overlap score on the ground-truth bounding boxes measured using the same criteria as PASCAL VOC challenge (Area Overlap / Area Total). A detection is considered valid when the bounding box overlap score exceeds 0.5. Each dot in any of the figures represents a candidate bounding box. They are computed from object detectors trained using Fast R-CNN framework [36]. The color represents the scene identity. In this example, green color represents kitchen, while red color accounts for a bedroom. From Figure 5.2(a), it is clear that many valid detections (i.e., overlap score (Area Overlap / Area Total) is over 0.5) can be found in the kitchen scenes. The figure also shows that there is not a single valid microwave detection in bedroom scenes for the training dataset, which is consistent with our common sense understanding.

If we select a threshold for the object detection score that captures most of the valid detections in the kitchen, then such a threshold produces lots of false microwave detections in the bedroom scene; but if we set up a high threshold for microwave detection (in order to avoid adding invalid detection of the bedroom scenes), then a lot of correct detections from the kitchen will be ignored. Figure 5.2(b) shows a possible re–scoring for the object detection scores based on the scene identity that deals with the fact that microwaves rarely appear in a bedroom. As can be appreciated from the figure, we have performed a simple shifting of the detection scores appearing in bedroom scenes. As a result, the detections from the bedroom scenes do not add any false positives which allows improving the results of object detection.

5.2.1 Greedy Algorithm

The goal of our algorithm is to find the optimal value to be added to the initial object detection score for each scene identity from the training data. If N_o is the number of different object detectors, there are a total of $N_s * N_o$ values to be learned. The values are saved in a matrix $C_{N_s \times N_o}$, which contains the corrections that need to be added to the detection scores according to the type of scene and object detector. We fix the object detector and fill out the rows of the matrix $C_{N_s \times N_o}$ applying the procedure that is described below. Once the correction matrix is filled for the different scenes of a particular object detector, we repeat the same procedure with every object detector.

The procedure uses as input the detection scores and their corresponding overlap scores of the candidate object bounding boxes. The candidates are grouped according to the type of scene of the frame. The first step is to select a scene identity to be used as a reference by the other types of scenes to compute their corrections. We calculate the mean Average Precision (mAP) score of the object detector for the candidates in each type of scene and save them in a sorted list. The scene identity that has the highest mAP value is selected as the reference. Once the reference scene identity is selected, we process all the scenes that do not contain any valid detection according to the PASCAL–overlap criteria. This is the same case presented in Figure 5.2(b). The magnitude of the correction is given by the difference between the lowest detection score value of a valid bounding box in the reference scene, and the value of the highest score of the new type of scene being processed. In practice, we also add a small fixed tolerance value ε, that ensure all the samples of the processed scene have scores lower than the lowest valid detection in the reference scene.

The remaining types of scenes are processed one by one starting from the scene with higher mAP in the sorted list of scenes computed in the first step that has not been processed yet. The intuition behind this choice is to assure that we adjust first the corrections of the type of scenes that need less adjustment in the correction value. At this point, we conduct a grid search of the correction value for the currently processed scene identity. The objective function is to maximize the mAP computed using the conjunction of candidates bounding boxes from previously processed scene identities and the currently processed scene identity. All the detection scores of the candidates involved in the computation of the mAP are adjusted according to their scene identities.

5.2.2 Support Vector Regression (SVR) Algorithm

In this section, we present an algorithm to learn to re-rank the object detection scores depending of the scene identity of the tested frame. The algorithm is based on a Support Vector Regressor (SVR). The problem of regression is equivalent to finding a function which approximately maps from an input domain to the real numbers based on a training sample.

Our goal is to map the object detection score to a new score value considering the scene identity. Then, the input data must encode the current scene identity and also include the detection score. The scene identity is encoded as one-hot vector of scene identities i.e. a vector with dimension equal to the number of scenes, with an entry equal to one in the dimension representing the actual scene identity, and zeroes in all the others dimensions. Hence, the input data $x^i \in \Re^{N_s+1}$ is represented by the concatenation of the one-hot scene identity vector and the detection score of the candidate bounding box.

The output data $y^i \in \Re$ contains the target detection scores. With y^i having any one of these possible values:

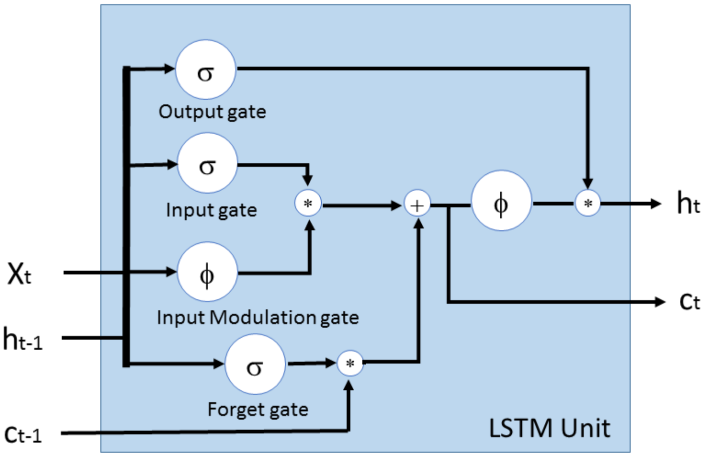
$$y^i = \begin{cases} 1 & \text{if overlap score} \geq 0.5 \\ J^i & \text{otherwise} \end{cases} \tag{5.3}$$

where J^i represents the overlap score of the candidate detection.

A separated regressor is trained for every type of object in the dataset. During testing, the detection score and the output of the scene classifiers are used to encode the input vector. The regression output of the regressor associated to the type of object is used as the new score for the bounding box.

5.3 Improving object detection without scene identity labeling.

In this section, we present a framework to use the general visual information of the frame sequences, and impose temporal constraints with the purpose of estimating how likely certain types of objects are present in the frame (without using a specific object detection method). Such information is employed to improve the results of the existing object detectors.



$$
\begin{aligned}
i_t &= \sigma(W_i X_t + U_i h_{t-1} + b_i) \\
f_t &= \sigma(W_f X_t + U_f h_{t-1} + b_f) \\
g_t &= \phi(W_g X_t + U_g h_{t-1} + b_g) \\
c_t &= f_t \odot c_{t-1} + i_t \odot g_t \\
o_t &= \sigma(W_o X_t + U_o h_{t-1} + b_o) \\
h_t &= o_t \odot \phi(c_t)
\end{aligned}
$$

Figure 5.3: Internal representation of an LSTM unit.

Our framework is based on a feedback network called Long Short-Term Memory (LSTM) [111]. LSTM is a type of neural network that allows connections from units in the same layer, creating loops that enable the network to use information from previous passes, acting as memory. LSTM can actively maintain self-connecting loops without degrading associated information. Figure 5.3 depicts the internal structure and the associated equations of the LSTM unit selected in our implementation. The LSTM unit takes an input vector X_t at each time step t and predicts an output h_t. In contrast to a simple Recurrent Neural Network (RNN) unit, the LSTM unit additionally maintains a memory cell c, which allows it to learn longer term dynamics. As a consequence, LSTM is a very effective technique to capture contextual information when mapping between input and output sequences.

Figure 5.4 depicts the proposed framework. Every frame is preprocessed to obtain a visual image descriptor which feeds the Long Short-Term Memory (LSTM) network. The system is trained to produce the correct answer to the question: which objects are visible in the image?

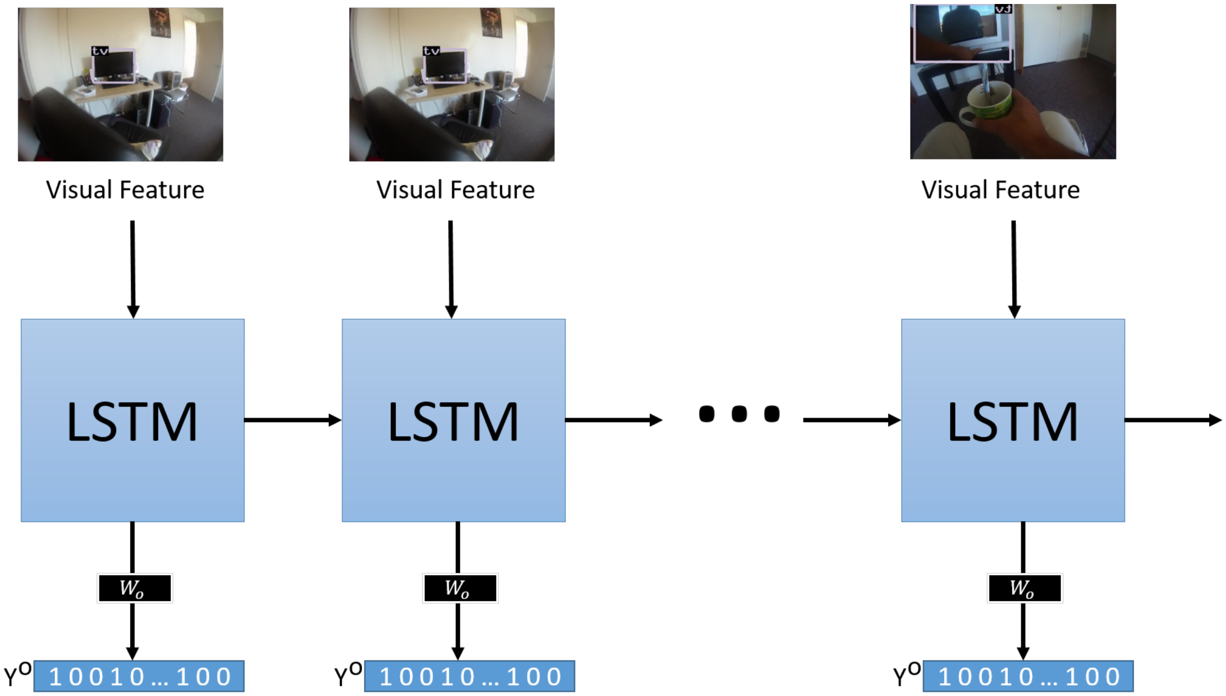


Figure 5.4: Our framework to obtain the most likely objects from scene descriptor in a frame sequence. Visual features are used as inputs, while the target vector $Y^o = [y_1^o, y_2^o, \cdots, y_{N_o}^o]$ encodes the presence or absence of an object class in the frame.

The answer to the question is encoded also as a vector $Y^o = [y^o_1, y^o_2, \cdots, y^o_{N_o}]$, where N_o is the number of possible objects to be considered, and $y^o \in \{0,1\}$. The vector Y_o has non-zero entries at the positions that indicate the indexes of existing objects in the frame. In training time, we use the information of every frame to fill out the vector Y^o, and the image descriptor X.

During testing, for each frame descriptor, we obtain a N_o dimensional output vector Y^o with values in the range $[0,1]$. The N_o dimensions of the vector indicate how likely is to find a type of object given the visual information of the frame and its history.The output layer after the LSTM unit is shared across the time.

In practice, we use this likelihood as a way to re-score the results of object detectors, according to the general information of the scene by means of the simple re-scoring function

$$S^{new}_{p_j} = S_{p_j} + k * Y^o_p, \tag{5.4}$$

where $S^{new}_{p_j}$ is the new score for the instance j of object type p, S_{p_j} is the score of the object detector j for object type p , Y^o_p is the output after the LSTM that indicates the likelihood of having the object p in the scene, and k is a constant that indicates the importance of the scene information in the final score. The value of k is determined from a small validation set containing ADL egocentric videos.

5.4 Experiments

We conduct our experiments in the Activities of Daily Living (ADL) dataset [76]. ADL dataset contains High Definition (HD) quality video from 18 daily indoor activities such as washing dishes, brushing teeth, or watching television, performed by 20 different persons in their apartments. Each video is of approximately 30 minutes length, and the frames are annotated every second with object bounding boxes of 42 different object classes. From the 42 annotated object classes, results of a trained Deformable Part-based Model (DPM) [32] are provided for 17 of them. In addition to the provided DPM models, we trained object detectors using the Fast R-CNN framework [36] and show that the proposed algorithms consistently achieve improvements independent of the type of object detector used.

The ADL dataset provides splits for separating training and testing data. From the twenty videos of the dataset, the first six of them were used as training data for object detection by the authors of the dataset. We followed the same splits on the data, then the first six videos were used to train scene classifiers, object detectors using deep networks, and the LSTM network for improving object detection without scene labels.

We performed scene identity annotations for all the video frames of the dataset. We identify eight types of scenes in the dataset. They are the kitchen, bedroom, bathroom, living room, laundry room, corridor, outdoor, and none of them (blurred frames, or non-identified place).

To evaluate the object detectors, we use the standard mean Average Precision (mAP) evaluation metric. We use the classical PASCAL VOC criteria, which establishes that at least a value of 0.5

on the overlap/union ratio among ground-truth and detection bounding box is needed to declare the bounding box as a valid detection.

5.4.1 Scene Identification

In this subsection, we show experiments on frame level scene identification and the improvements achieved by using the temporal information.

We performed frame scene identification on the video frames of the test dataset. The first baseline in our experiments is simply the results of the scene identification methods without considering the time constraint. A second more challenging benchmark considers the temporal constraint by using a moving average filter across the temporal domain. A third baseline examines a Hidden Markov Model (HMM). Finally, we show that the overall accuracy of scene identification methods is largely improved using the proposed CRF formulation.

We use four different frame level scene identification approaches in our experiments to show that the proposed formulation works well independent of the selected scene identification method. One approach is the traditional Bag of Words (BoW) representation, encoding CNN features computed over object proposals selected by using the selective search window technique by Cheng et al. [96]. We also performed experiments with the Multi-Scale Orderless Pooling of Deep Convolutional Activation Features (MOPCNN) [91] and the two additional variants described below.

Multi-Scale Orderless Pooling of Deep Convolutional Activation Features (MOPCNN) [91] is, to the best of our knowledge, the current state of the art for scene classification. MOPCNN operates at 3 scales, all of them using the sixth fully connected layer output of the Krizhevsky's convolutional network. At the full image scale, the descriptor is directly the output of the sixth layer, while the descriptor for the other two scales is created by VLAD encoding of periodically sampled CNN features at different scales followed by dimensional reduction.

The complete MOPCNN method is used as one of the tested scene identification methods, but also two variants of the method are also examined: a) the full scale of the MOPCNN method (MOPCNN-L1) i.e., the global CNN descriptor, and b) the third scale of the MOPCNN (MOPCNN-L3), which uses VLAD encoding in the 64x64 pixels scale. These two variants complete our four methods used for scene identification.

We use Caffe [108] to implement CNN feature extraction. For the Bag of Words implementation, a total of 200 object proposals are used, and the dictionary size is fixed in 5000 words. For all the scene identification methods, we use a linear SVM as the classifier. We use the graph-cuts based minimization procedure in [103, 104, 105] to obtain the optimal solution for the equation 5.1.

Table 5.1 shows the overall accuracies for the three baselines and the proposed CRF method. The baseline 1 in the table corresponds to the direct output of the scene classifiers. The baseline 2 corresponds to the moving average filtering in the time domain of the scene scores. The filter size is in some way a measure of how fast the person changes from the current scene to other scene. In our experiments, the sampling is one frame per second (1 *fps*). We examined different filter sizes, finding that considering the $r = 7$ previous sampled frames on the currently tested frame produced best accuracies. These are the results reported in the second row of the table. The baseline 3 is a Hidden Markov Model (HMM) that predicts the sequence output of the scene identities.

The results of the proposed CRF method depend on the choice of the matrices $V(y_p, y_q)$ and $\omega_{p,q}$. Following the findings of the baseline 2, the presented results assume that information of the previous seven frames influences the current frame label.

We first consider the case where any of the seven previous frames have the same impact on the current frame label i.e., $\omega_{p,p-1} = \omega_{p,p-2} = \cdots = \omega_{p,p-7}$, and penalty is the same for any pair of scene identities, i.e., the $V(y_p, y_q)$ value is the same for any position off diagonal. The fourth row of Table 5.1 reports results for this uniform choice of V and ω.

We also consider the case where the influence of the most recent frames is stronger than the previous ones. Hence, we assume that for each row of the matrix ω, its weights follow a Gaussian function with origin in the current frame. We also consider alternatives for matrix $V(y_p, y_q)$, where pairs of scene labels with more frequent transitions are penalized less severely than others pairs that rarely occurs. We use the ground-truth data to count for the possible transitions between scene identities which are normalized and represented as T_{y_p,y_q}. Values for the $V(y_p, y_q)$ entries are defined as $V(y_p, y_q) = 1 - T_{y_p,y_q}$. The last row of Table 5.1 shows the best results achieved with the selected non-uniform V and ω matrices.

Table 5.1: Comparison of the overall accuracy of four scene identification methods. The baseline 1 does not consider any temporal constrain, the baseline 2 uses a moving average filter in the time domain to decide the frame identity, and baseline 3 considers a HMM model. The proposed CRF is examined under two different choices of pairwise terms.

	BoW	MOP	CNN	CNN
	CNN	CNN	L1	L3
Baseline 1. No Time	50.45	64.53	64.08	63.87
Baseline 2. Moving Average	58.54	67.95	69.38	67.66
Baseline 3. HMM	61.21	68.97	70.92	69.79
Proposed CRF - 1. Uniform V,ω	**65.52**	68.53	71.85	69.88
Proposed CRF - 2. Non-uniform V,ω	62.27	**72.09**	**74.21**	**72.15**

In all the four scene classifiers, there is a visible improvement in the accuracy using the proposed CRF with respect to the baselines. The relative increase is more significant for the weakest scene classifier, the Bag of CNN features. As is expected, the state of the art method (MOPCNN) has the best accuracy between the scene classifiers before using any temporal constraint. However, after considering the temporal information, the improvement is superior in the scene detectors that only use one scale CNN as a classifier. As a result, the two variants of the MOPCNN method produce better accuracies than the complete MOPCNN method. This surprising result, indicates that in real life applications, a weaker but less computationally intense scene classifier can be used in place of expensive computational methods as long as the temporal constraint is exploited.

We also note that the CRF defined with a more complex pairwise relation (Non-uniform V and ω), that weights the importance of the closest frames to the tested frame, and considers the likelihood of scene transitions, produces better results when the best scene classifiers (MOPCNN and their two variants) are used. The increase was a bit lower than the uniform V and ω CRF with the weakest scene classifier (BoW), but still considerably better than any of the baselines. We attribute this effect to the stochastic nature of the output generated by the noisiest BoW classifier that converts the output in a less predictable event.

5.4.2 Improving Object Detection

We perform experiments to demonstrate that the methods presented in this chapter to improve object detection results, generalize to different kinds of object detectors. In this section, we use the DPM object detection results provided with the ADL dataset and also the object detection outputs of models trained using the Fast R-CNN framework.

The DPM models themselves are not provided, only the bounding boxes, and scores of the detections obtained by their models in the training and testing videos of the ADL dataset. A total of 17 types of objects is provided.

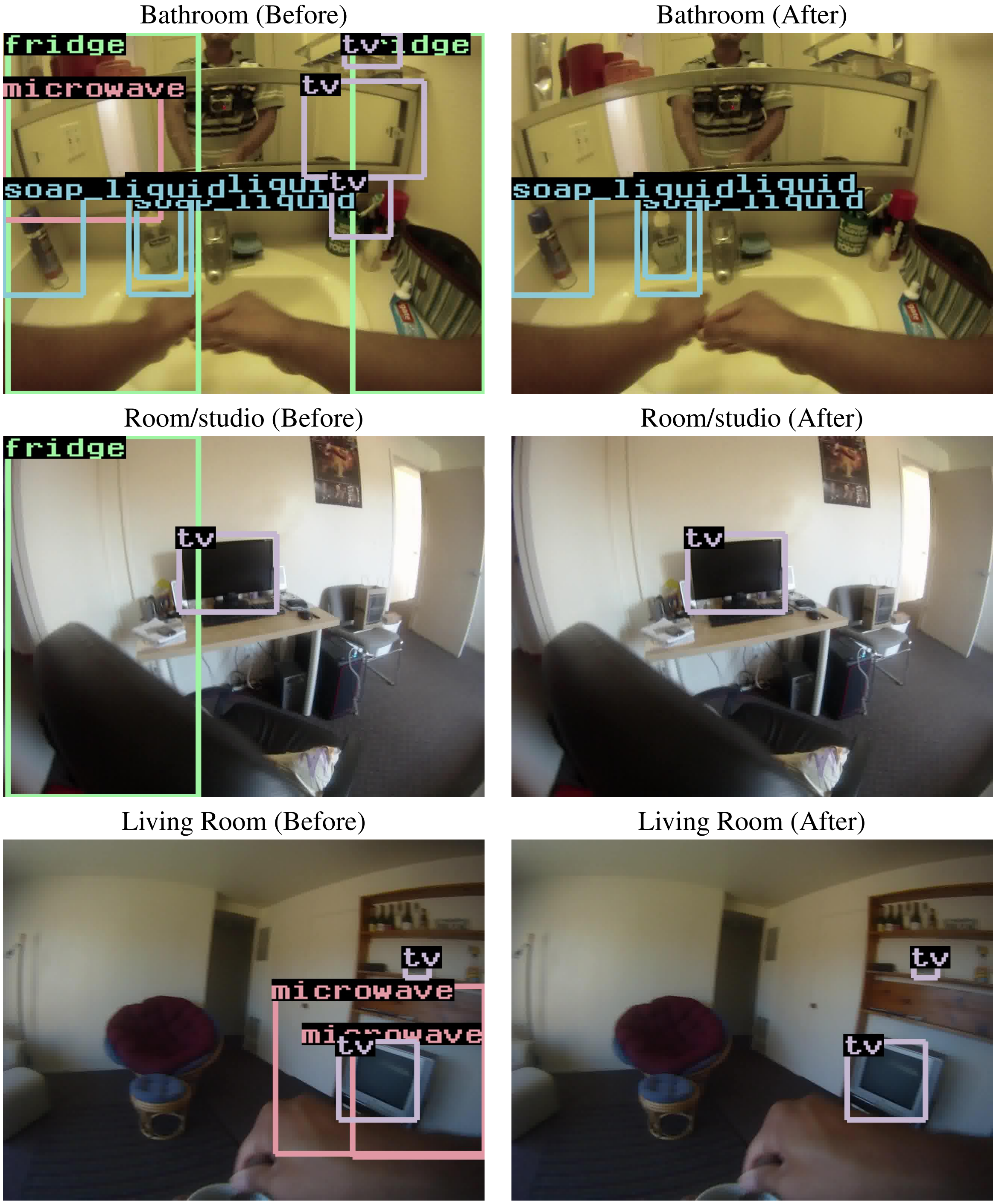
The Fast R-CNN models are trained using the VGG16 network [46] employing object proposals computed using EdgeBox [45]. We trained models for the 42 annotated objects. However, we only consider objects with an mAP of at least 5.00 %. A total of 20 object detectors satisfies this condition.

We learn different matrices of corrections $C_{N_s \times N_o}$ and re-scoring functions for the DPM and the Fast R-CNN detectors following the procedures described in section 5.2. In the case of the greedy algorithm, the parameter ε is set to 0.05 for all the experiments. In the case of the SVR algorithm, we use a Radial Basis Function (RBF) as kernel. The parameters of the SVR are $C = 0.01$, and $\gamma = 0.1$, in order to have a smooth regression function.

The first six videos of the ADL dataset are used to train the LSTM network. These videos contain information about which objects are shown in each one of the sampled frames. The Y^o vectors are generated by forming groups with duration of 20 seconds and an overlap of 8 seconds. We use the scene descriptor of the MOPCNN method to feed the network. The training is performed in batches of 16 groups executing a total of 30,000 epochs.

In testing phase, we feed each frame with the scene descriptor, and obtain a vector that indicates the likelihood of having the object (indexed in each dimension of the vector) given the general scene content. We use the equation 5.4 to re-score the object detection. The value of k in our validation set is 0.11 for both set of object detectors, the DPM and Fast R-CNN models.

The figures 5.5 and 5.6 show some qualitative results of five object detectors with a detection threshold fixed on -0.7 using the DPM object detector, for some random frames covering different scenes. In both figures, the first columns shows the detection results without using scene information, while the second columns show the obtained detection after performing re–scoring considering the scene identity. The number of false microwave detections is reduced for the scenes in the bedroom, living room, and bathroom. In the same way, false positives objects such as tv are removed from the scenes in the kitchen, and bathroom.



Bathroom (Before)

Bathroom (After)

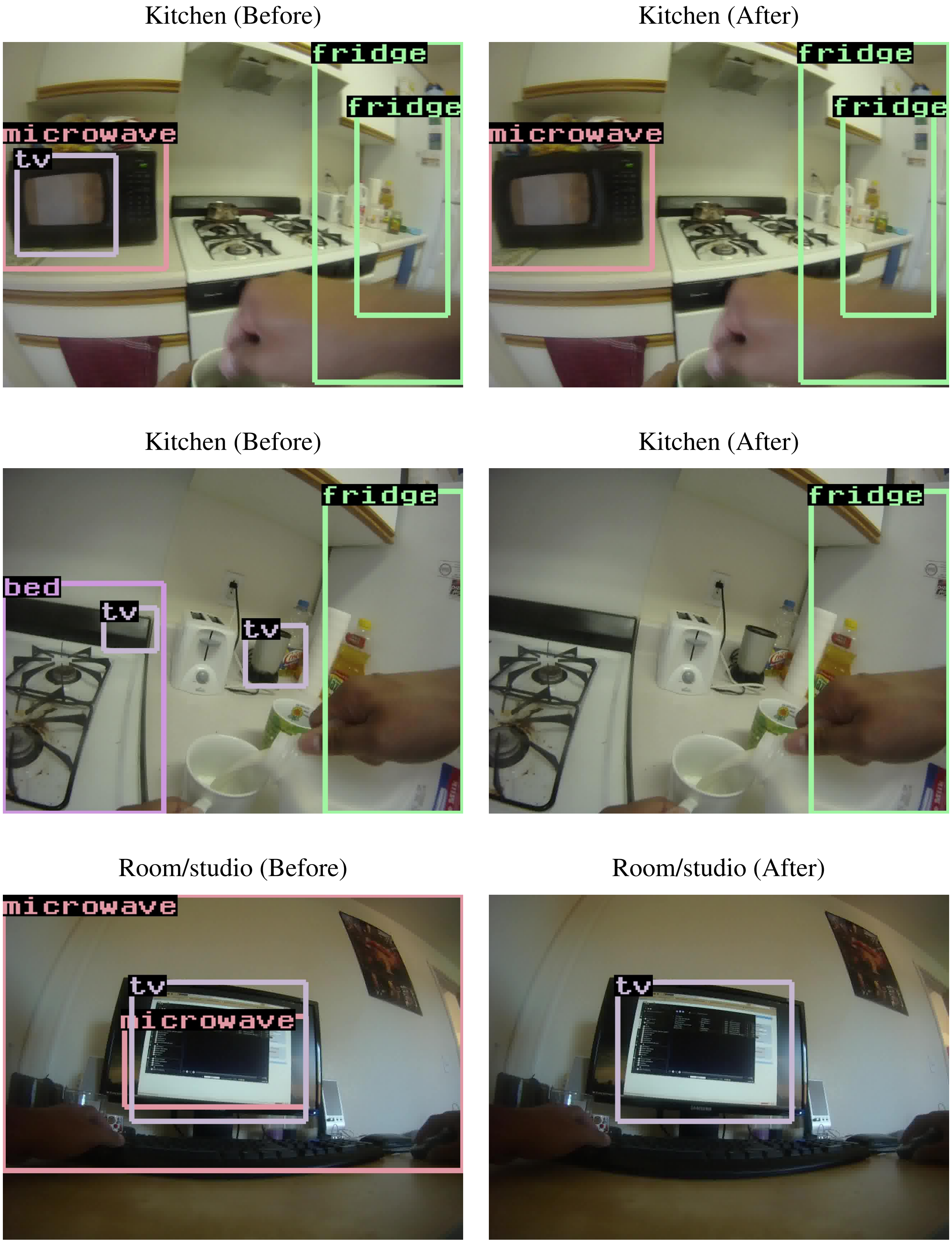
Room/studio (Before)

Room/studio (After)

Living Room (Before)

Living Room (After)

Figure 5.5: Qualitative results of the object detection before and after re–scoring the detections based on the scene. Many false positives are removed after the proposed re-scoring.



Kitchen (Before)

Kitchen (After)

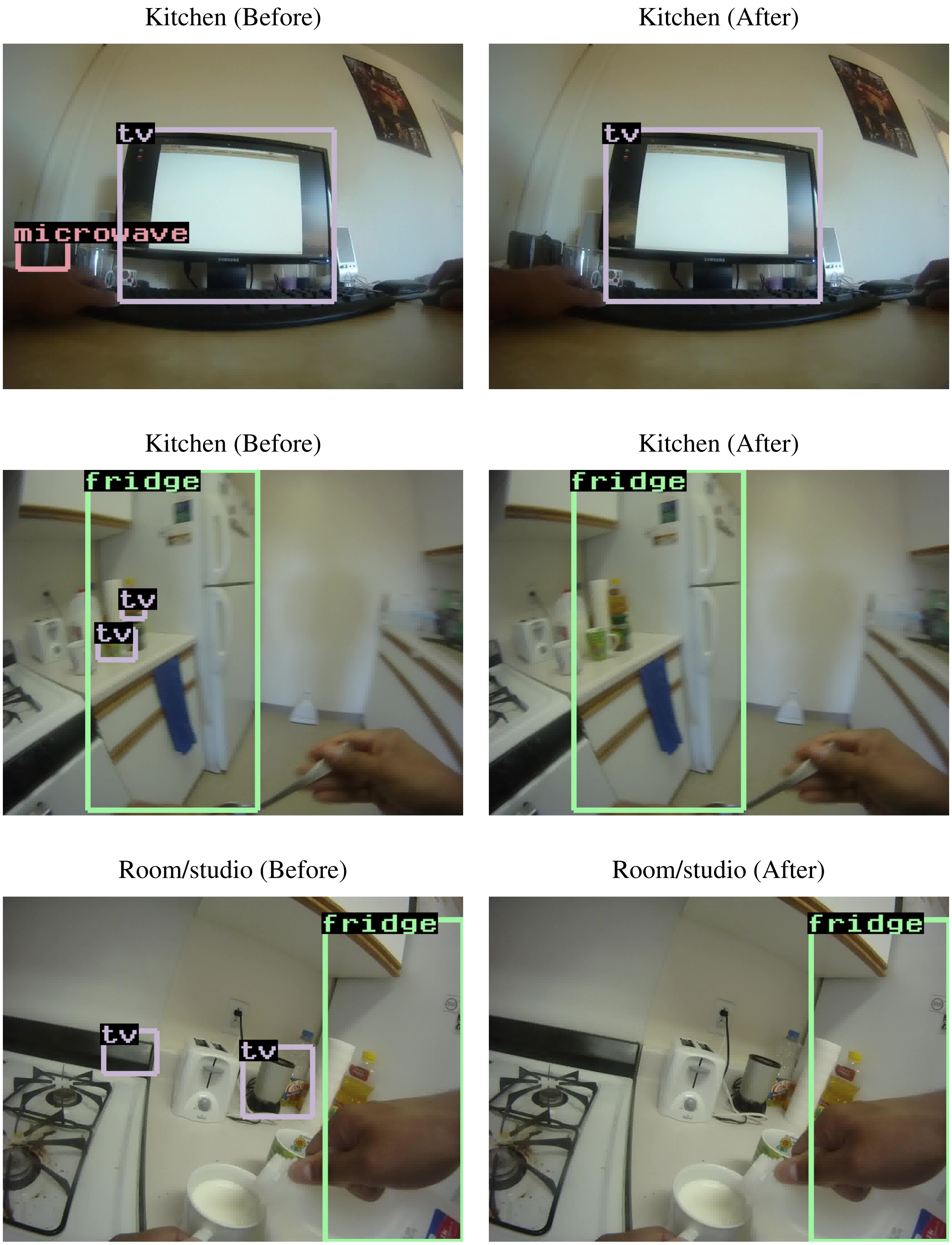
Kitchen (Before)

Kitchen (After)

Room/studio (Before)

Room/studio (After)

Figure 5.6: More qualitative results of the object detection before and after re–scoring the detections based on the scene. Many false positives are removed after the proposed re-scoring.


Kitchen (Before)

Kitchen (After)

Kitchen (Before)

Kitchen (After)

Room/studio (Before)

Room/studio (After)

Figure 5.7: More qualitative results of the object detection before and after re–scoring the detections based on the scene. Many false positives are removed after the proposed re-scoring.

Table 5.2 presents the results associated with the DPM object detectors and Table 5.3 displays the results related to the Fast R-CNN object detector. Tables 5.2 and 5.3 share the same structure. Each column of the tables presents the detection results in different scenarios. The columns of the tables present such scenarios. The first column contains the results of the selected object detector applied on the sampled frames of the ADL dataset without considering any information about the scene. The second and third columns present the results of improved object detection assuming the scene identities are known. Each column shows a different technique. They are the greedy algorithm and the SVR algorithm.

Instead of assuming the scene identities of the frames are known, the next two columns present the outcome of the greedy and the SVR algorithms, but this time using the best scene identification method that was obtained from the experiments of the previous section. This method is the model trained using the CNN features in full scales (L1) in conjunction with CRF. In the case of the greedy algorithm, the corrections values for a frame are computed as a weighted sum of corrections associated with the normalized scene identities scores for each type of object. The correction values are extracted from a column of the matrix $C_{N_s \times N_o}$, and the weights from the normalized scores of the scene identity classifiers. For the SVR algorithm, we follow a similar weighting strategy to estimate the new score values but using the scores obtained from the regression functions. Finally, the last column shows the results of the proposed LSTM method to improve object detection without explicitly using scene labeling.

Table 5.2: Results for the DPM object detection of the ADL dataset using mAP metric (as a percentage). The use of scene information increases the mAP for most of object categories. The best improvements are obtained when the scene identity is known. LSTM method performs better in comparison to the cases where the scene identity is estimated from scene classification.

		Scene Known		***CNN-L1 Scene***		
	DPM	**Greedy**	**SVR**	**Greedy**	**SVR**	**LSTM**
bed	8.74	10.32	9.28	9.01	9.34	9.37
book	11.93	11.12	10.98	12.11	11.21	12.54
bottle	1.76	1.83	2.05	1.73	2.01	1.69
cell	0.19	0.35	0.29	0.18	0.32	0.19
detergent	3.90	4.64	5.12	4.02	4.87	3.96
dish	1.26	0.98	1.35	1.53	1.04	1.38
door	12.60	7.82	8.64	12.83	9.79	14.24
fridge	24.80	28.45	29.18	25.95	26.05	26.36
kettle	12.16	13.02	12.67	11.43	12.56	13.01
laptop	38.52	40.41	37.81	38.99	32.93	39.81
microwave	17.76	21.37	22.13	18.88	21.86	19.57
pan	6.15	6.70	7.02	6.23	6.58	6.58
pitcher	1.37	1.69	1.65	0.68	1.79	1.27
soap	5.12	6.34	6.48	5.43	5.72	6.00
tap	30.15	32.40	33.38	30.19	31.84	29.59
remote	4.88	6.28	5.91	5.14	6.31	6.12
tv	44.09	46.88	48.21	45.70	47.19	45.12
Total	13.25	14.15	14.24	13.53	13.61	13.93

Table 5.3: Results for the Fast R-CNN object detectors on the ADL dataset using mAP metric (as a percentage). The LSTM method produces higher improvements compared to any of the other methods to re-score the object detection results.

		Scene Known		***CNN-L1 Scene***		
	Fast R-CNN	**Greedy**	**SVR**	**Greedy**	**SVR**	**LSTM**
book	12.83	13.62	13.88	13.12	14.14	13.33
bottle	11.28	12.32	9.96	8.70	9.81	11.71
cell	8.65	2.21	3.30	4.51	6.31	8.65
detergent	9.13	11.23	7.50	8.75	8.99	9.14
dish	11.19	13.03	13.85	12.01	12.96	11.95
door	5.59	5.69	5.85	5.61	5.24	5.74
fridge	24.95	27.54	26.25	25.07	25.41	26.75
kettle	23.83	31.11	26.79	27.12	27.20	27.28
laptop	37.46	41.17	33.16	43.91	37.37	48.84
microwave	32.35	36.85	36.78	33.62	34.53	32.37
mug/cup	13.24	14.67	14.21	12.51	12.90	14.29
oven/stove	43.02	47.73	54.58	49.54	52.66	52.54
pan	10.99	13.90	13.83	10.78	11.31	11.00
person	25.74	43.66	66.63	64.97	63.49	71.64
soap	18.77	19.09	20.53	17.05	16.94	18.62
tap	39.55	48.78	46.00	47.64	46.25	47.90
thermostat	9.01	9.63	6.27	6.00	7.83	8.99
remote	32.88	43.91	47.98	43.79	45.20	41.34
washer/dryer	38.86	47.17	45.09	39.09	40.42	40.52
tv	57.58	61.60	66.07	61.96	63.57	67.75
Total	23.35	27.24	27.91	26.79	27.15	28.49

Trained Fast R-CNN object detectors produce better detector models than the provided DPM results. The performance is almost the double, as can be appreciated from comparing the first column of the tables. Besides, Fast R-CNN models can generate a longer number of good models, 20, compared to the 17 models provided using DPM. When we include the information about the scene, we observe consistent improvements for all the presented scenarios independent of the detector model utilized. The gains are more notorious in the case of the Fast R-CNN detector models than for DPM. As is expected, using the exact information about the scene identity (columns 2 and 3) outperforms the results obtained when the scene identity is estimated (column 4 and 5) for both types of detector models. The increases are considerable in the case of Fast R-CNN models. SVR algorithm have slightly better overall performances compared to the greedy algorithm in all the tested scenarios. In general, a valid observation of the experiment is that when the object detectors have good models (mAP over 20 %), the improvements of the results by using the scene information are consistently higher than for weaker object detectors.

Finally, we highlight the results of the improved object detection without explicitly using the label of the scene. Besides reducing the labeling effort, we note that the performance achieved using the proposed LSTM formulation outperform the results reached when we estimate scene labels from scene classifiers. In fact, for the case of the Fast R-CNN detectors, the results are superior to the ones obtained using the knowledge about the scene identity directly.

5.5 Summary

In this chapter, we presented algorithms for leveraging inherent constraints of egocentric vision towards improved scene identification and object detection capabilities. Firstly, we notice that the scene identity of an egocentric video remains consistent for several frames. Subsequently, we presented a CRF formulation that improves the frame level scene identification results of different methods. Secondly, we identified the association between some type objects with some scene locations and proposed two re-scoring algorithms to improve the object detection according to the scene content. For the case where an explicit scene labeling is not available, we proposed a LSTM formulation that directly estimates the likelihoods of having some objects given a sequence of scene descriptors. Such formulation was used to improve the object detection scores of the DPM and Fast R-CNN object detection outputs. The presented algorithms were tested on the well-known public ADL dataset.

CHAPTER 6
CONCLUSIONS

In this dissertation we have examined the influence of context on images and videos, and the extraction of the shared context from multiple images. The main contributions of this dissertation can be summarized as follows:

- We have proposed a new image representation that shows the main elements that make up of the image, their relative importance and type of relations. The representation enables the learning of the overall visual content. Given a test image, we have developed a method to obtain the proposed representation of the image. In addition, the method improves the results of object detection.
- We have proposed a method to extract the shared context of a group of diverse images. The shared context corresponds to a set of words, that constitute the topic.
- We have proposed a new image search paradigm where the retrieved images are not only similar to the query images, but also conceptually related.
- We have proposed algorithms to exploit the semantic and temporal context to improve scene identification and object detection in egocentric camera videos.

6.1 Future Directions

In this section we describe possible future directions for the research presented in this dissertation. The approaches proposed in this dissertation can be improved in many ways.

In chapter 3, we proposed a new image representation. Our method uses existing annotations from object detection datasets. The results are highly dependent in the quality and quantity of these annotations. A very important research direction is how to obtain image representations reducing the dependence on annotations. Moving away from fully supervised training of the image representation is a very important topic to explore. In additional issue is that the proposed representation was evaluated by user studies, and indirectly through the improvement in object detection. However, better metrics for image representation are still missing. Additionally, a further step needed is how to use the obtained representation to solve other computer vision problems such as visual questioning-answering and captioning.

In chapter 4, we proposed a method that finds the topic of a set of images and a new image retrieval paradigm from multiple images. The proposed approaches relies on having a dataset of image annotation with captions. Our methods are highly dependent on parameters like the number of retrieved images, the assigned weights, among others. Hence, more research is necessary to make the proposed approach less dependent on parameter tuning, and annotations that are difficult to obtain.

In chapter 5, we have studied some techniques to improve frame level scene identification and object detection for videos. Two of the drawbacks of our current approach is that it needs a predetermined set of scene categories for training, and that it lacks integration between object detectors and scene identities for understanding. Future work could be conducted on extending the image level representation of chapter 3 to model the video content.

LIST OF REFERENCES

[1] R. Fixot, *American Journal of Ophthalmology*, 1957.

[2] O. Vinyals, A. Toshev, S. Bengio, and D. Erha, “Show and tell: A neural image caption generator,” in *CVPR*, 2015.

[3] Q. Wu, C. Shen, L. Liu, A. Dick, A. Dick, and A. van den Hengel, “What value do explicit high level concepts have in vision to language problems?” in *arXiv:1506.01144*, 2015.

[4] H. Fang, L. Deng, M. Mitchell, S. Gupta, P. Dollar, J. C. Platt, F. Iandola, J. Gao, C. L. Zitnick, R. K. Srivastava, X. He, and G. Zweig, “From captions to visual concepts and back,” in *CVPR*, 2015.

[5] L. A. Hendricks, J. Donahue, J. Donahue, Z. Akata, M. Rohrbach, B. Schiele, and T. Darrell, “Generating visual explanations,” in *arXiv:1603.08507:*, 2016.

[6] X. Jia, E. Gavves, B. Fernando, and T. Tuytelaars, “Guiding long-short term memory for image caption generation,” in *ICCV*, 2015.

[7] J. Mao, J. Huang, A. Toshev, ana Camburu, A. Yuille, and K. Murphy, “Generation and comprehension of unambiguous object descriptions,” in *CVPR*, 2016.

[8] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo, “Image captioning with semantic attention,” in *CVPR*, 2016.

[9] B. Zhou, Y. Tian, S. Sukhbaatar, A. Szlam, and R. Fergus, “Simple baseline for visual question answering,” in *1512.02167v2*, 2016.

[10] Q. Wu, P. Wang, C. Shen, A. Dick, and A. van den Henge, “Ask me anything: Free-form visual question answering based on knowledge from external sources,” in *CVPR*, 2016.

[11] A. Oliva and A. Torralba, “The role of context in object recognition,” *TRENDS in Cognitive Sciences*, vol. 11, no. 12, pp. 520–527, 2007.

[12] S. E. Palmer, “The effects of contextual scenes on the identification of objects,” *Memory & cognition*, vol. 3, no. 5, pp. 519–526, September 1975.

[13] J. L. Davenport and M. C. Potter, “Scene consistency in object and background perception,” *phychological science*, vol. 15, pp. 559–566, 2004.

[14] C. Green and J. E. Hummel, “Familiar interacting object pairs are perceptually grouped,” *Journal of Experimental Psychology*, vol. 32, pp. 1107–1119, 2006.

[15] C. Galleguillos and S. Belongie, “Context based object categorization: A critical survey,” in *ECCV*, 2008.

[16] S. K. Divvala, D. Hoiem, J. H. Hays, A. A. Efros, and M. Hebert, “An empirical study of context in object detection,” in *IEEE Conference on Computer VIsion and Pattern Recognition (CVPR)*, 2009.

[17] M. J. Choi, J. J. Lim, A. Torralba, and A. S. Willsky, “Exploiting hierarchical context on a large database of object categories,” in *IEEE Conference on Computer VIsion and Pattern Recognition (CVPR)*, 2010.

[18] Z. Song, Q. Chen, Z. Huang, Y. Hua, and S. Yan, “Contextualizing object detection and classification,” in *CVPR*, 2010.

[19] A. Torralba, K. P. Murphy, W. T. Freeman, and M. A. Rubin, “Context-based vision system for place and object recognition,” in *ICCV*, 2003.

[20] G. Heitz and D. Koller, “Learning spatial context: Using stuff to find things,” in *ECCV*, 2008.

[21] P. Carbonetto, N. de Freitas, and K. Barnard., “A statistical model for general contextual object recognition,” in *ECCV*, 2004.

[22] A. Torralba, K. Murphy, and W. T. Freeman, “Using the forest to see the trees: object recognition in contex,” in *Comm. of the ACM*, 2010.

[23] D. Park, D. Ramanan, and C. Fowlkes, “Multiresolution models for object detection,” in *ECCV*, 2010.

[24] D. Forsyth, J. Malik, M. Fleck, Greenspan, L. T. H., S. Belongie, C. Carson, and C. Bregler, “Finding pictures of objects in large collections of images.” in *Object Representation in Computer Vision*, 1996.

[25] S. Bell, C. L. Zitnick, K. Bala, and R. Girshick, “Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks,” in *arXiv:1512.04143*, 2015.

[26] S. Gidaris and N. Komodakis, “Object detection via a multi-region & semantic segmentation-aware cnn model,” in *arXiv:1505.01749*, 2015. [Online]. Available: http://arxiv.org/pdf/1505.01749v3

[27] J. Li, Y. Wei, X. Liang, J. Dong, T. Xu, J. Feng, and S. Yan, “Attentive contexts for object detection,” in *arXiv:1603.07415*, 2016.

[28] G. Vaca-Castano, S. Das, J. P. Sousa, N. D. Lobo, and M. Shah, “Improved scene identification and object detection on egocentric vision of daily activities,” *Computer Vision and Image Understanding (CVIU)*, 2016.

[29] N. Dalal and B. Triggs, “Histogram of oriented gradients for human detection,” in *CVPR*, 2005.

[30] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (voc) challenge,” *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

[31] M. Everingham, S. M. A. Eslami, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes challenge a retrospective,” *International Journal of Computer Vision*, 2014.

[32] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, “Object detection with discriminatively trained part based models,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, 2010.

[33] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, “Selective search for object recognition,” *International Journal of Computer Vision*, vol. 104, pp. 154 –171, 2013.

[34] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database.” in *CVPR*, 2009.

[35] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick, “Microsoft coco: Common objects in context,” in *ECCV*, 2014.

[36] R. Girshick, “Fast r-cnn,” in *IEEE Conference on Computer VIsion and Pattern Recognition (CVPR)*, 2015.

[37] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” in *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[38] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *CVPR*, 2014.

[39] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *NIPS*, 2012.

[40] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei., “Imagenet large scale visual recognition challenge,” 2014, arXiv:1409.0575.

[41] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," in *ICLR*, 2014.

[42] K. E. A. van de Sande, C. G. M. Snoek, and A. W. M. Smeulders, "Fisher and vlad with flair," in *CVPR*, 2014.

[43] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek, "Evaluating color descriptors for object and scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, pp. 1582–1596, 2010.

[44] M. Lin, Q. Chen, and S. Yan, "Network in network," in *ICLR*, 2014.

[45] C. L. Zitnick and P. Dollr, "Edge boxes: Locating object proposals from edges," in *European Conference in Computer Vision (ECCV)*, 2014.

[46] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2015.

[47] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *arXiv:1512.03385*, 2015.

[48] L. Fei-Fei and P. Perona, "A bayesian hierarchical model for learning natural scene categories," in *CVPR*, 2005.

[49] G. Csurka and F. Perronnin, "Fisher vectors: Beyond bag-of-visual-words image representations," *Computer Vision, Imaging and Computer Graphics. Theory and Applications*, vol. 229, pp. 28–42, 2011.

[50] A. Oliva and A. Torralba., "Modeling the shape of the scene: A holistic representation of the spatial envelope." *IJCV*, vol. 42, pp. 145–175, 2001. [Online]. Available: http://cvcl.mit.edu/Papers/IJCV01-Oliva-Torralba.pdf

[51] C. Siagian and L. Itti, "Rapid biologically-inspired scene classification using features shared with visual attention," *IEEE Trans . Pattern Anal. Mach. Intell.*, vol. 29, pp. 300–312, 2007.

[52] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *CVPR*, 2005.

[53] S. Wang, Y. Wang, and S. Zhu, "Learning hierarchical space tiling for scene modeling, parsing and attribute tagging," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 12, pp. 2478–2491, 2015.

[54] S. Wang, Y. Wang, and S. C. Zhu, "Hierarchical space tiling for scene modelling," in *Asian Conference on Computer Vision (ACCV)*, 2012.

[55] O. Sacks, *Seeing Voices. A jouney into the world of deaf.* Vintage Books, 1989.

[56] R. Arandjelovic and A. Zisserman, “Multiple queries for large scale specific object retrieval.” in *BMVC*, 2012.

[57] F. Basura and T. Tuytelaars, “Mining multiple queries for image retrieval: On-the-fly learning of an object-specific mid-level representation,” in *ICCV*, 2013.

[58] Y. Chen, X. Li, A. Dick, and A. van den Hengel, “Boosting object retrieval with group queries,” *IEEE Signal Processing Letters*, vol. 19, pp. 765–768, 2012.

[59] K.-J. Hsiao, J. Calder, and A. O. H. III, “Pareto-depth for multiple-query image retrieval,” *IEEE Transactions on Image Processing*, vol. 24, no. 2, pp. 583–594, Feb 2015.

[60] M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid, “Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation,” in *Computer Vision, 2009 IEEE 12th International Conference on*, 2009.

[61] M. Chen, A. Zheng, and K. Weinberger, “Fast image tagging,” in *Proceedings of The 30th International Conference on Machine Learning*, 2013, pp. 1274–1282.

[62] T. Deselaers, D. Keysers, and H. Ney, “Features for image retrieval: an experimental comparison,” *Information Retrieval*, vol. 11, pp. 77–107, 2008.

[63] D. Nister and H. Stewenius, “Scalable recognition with a vocabulary tree,” in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, 2006.

[64] H. Jégou, M. Douze, and C. Schmid, “Improving bag-of-features for large scale image search,” *International Journal of Computer Vision (IJCV)*, vol. 87, pp. 316–336, 2010.

[65] M. Jain, H. Jégou, and P. Gros, “Asymmetric hamming embedding: taking the best of our bits for large scale image search,” in *ACM Multimedia*, 2011.

[66] G. Tolias, Y. Avrithis, and H. Jégou, “To aggregate or not to aggregate: Selective match kernels for image search,” in *International Conference on Computer Vision (ICCV)*, 2013.

[67] A. Torralba, R. Fergus, and Y. Weiss, “Small codes and large image databases for recognition,” in *CVPR*, 2008.

[68] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth., “Describing objects by their attributes,” in *CVPR*, 2009. [Online]. Available: https://www.cs.cmu.edu/~afarhadi/papers/Attributes.pdf

[69] F. Yu, R. Ji, M. Tsai, G. Ye, and S.-F. Chang., “Weak attributes for large-scale image retrieval.” in *CVPR*, 2012.

[70] L. Torresni, M. Szummer, and A. Fitzgibbon, “Efficient object category recognition using classemes,” in *ECCV*, 2010.

[71] M. Douze, A. Ramisa, and C. Schmid, “Combining attributes and fisher vectors for efficient image retrieval,” in *IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[72] X. Wang, M. Yang, T. Cour, S. Zhu, K. Yu, and T. X. Han, “Contextual weighting for vocabulary tree based image retrieval,” in *Computer Vision (ICCV), 2011 IEEE International Conference on*, 2011.

[73] S. Zhang, M. Yang, X. Wang, Y. Lin, and Q. Tian, “Semantic-aware co-indexing for image retrieval,” in *IEEE International Conference on Computer Vision (ICCV)*, 2013.

[74] X. Ren and M. Philipose, “Egocentric recognition of handled objects: Benchmark and analysis,” in *CVPR Workshop*, 2009.

[75] A. Fathi, X. Ren, and J. M. Rehg, “Learning to recognize objects in egocentric activities,” in *CVPR*, 2011.

[76] H. Pirsiavash and D. Ramanan, “Detecting activities of daily living in first-person camera views,” in *CVPR*, 2012.

[77] L. Zhang, Y. Li, and R. Nevatia, “Global data association for multi-object tracking using network flows,” in *CVPR*, 2008.

[78] A. R. Zamir, A. Dehghan, and M. Shah, “Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs,” in *ECCV*, 2012.

[79] C. Stauffer, “Estimating tracking sources and sinks,” in *CVPR Workshop*, vol. 4, 2003.

[80] A. Andriyenko and K. Schindler, “Multi-target tracking by continuous energy minimization,” in *CVPR*, 2011.

[81] W. Han, P. Khorrami, T. L. Paine, P. Ramachandran, M. Babaeizadeh, H. Shi, J. Li, S. Yan, and T. S. Huang, “Seq-nms for video object detection,” Technical Report for Imagenet VID Competition 2015, Tech. Rep., 2016.

[82] J. Sivic and A. Zisserman, “Video google: A text retrieval approach to object matching in videos,” in *ICCV*, 2003.

[83] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, “Visual categorization with bags of keypoints.” in *ECCV Workshop on Statistical Learning in Computer Vision.*, 2004.

[84] K. Grauman and T. Darrell, “The pyramid match kernel: Discriminative classificationcation with sets of image features,” in *ICCV*, 2005.

[85] F. Perronnin and C. Dance, “Fisher kernels on visual vocabularies for image categorization,” in *CVPR*, 2007.

[86] F. Perronnin, J. Snchez, and T. Mensink, “Improving the fisher kernel for large-scale image classification,” in *ECCV*, 2010.

[87] H. Jegou, M. Douze, C. Schmid, and P. Pérez, “Aggregating local descriptors into a compact image representation,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[88] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, “Locality-constrained linear coding for image classification,” in *CVPR*, 2010.

[89] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, “Learning and transferring mid-level image representations using convolutional neural networks,” in *CVPR*, 2014.

[90] A. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, “Cnn features off-the-shelf: An astounding baseline for recognition,” in *CVPR DeepVision Workshop*, 2014.

[91] Y. Gong, L. Wang, R. Guo, and S. Lazebnik, “Multi-scale orderless pooling of deep convolutional activation features,” in *ECCV*, 2014.

[92] S. Frintrop, E. Rome, and H. I. Christensen, “Computational visual attention systems and their cognitive foundations: A survey,” *ACM Transactions on Applied Perception*, vol. 7, pp. 1–46, 2010.

[93] G. A. Miller, “Wordnet: A lexical database for english,” *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[94] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. Bernstein, and L. Fei-Fei, “Visual genome: Connecting language and vision using crowdsourced dense image annotation,” in *Arxiv: 1602.07332*, 2016.

[95] X. Zhang, L. Lu, and M. Lapata, “Top-down tree long short-term memory networks,” in *NAACL*, 2016. [Online]. Available: http://arxiv.org/abs/1511.00060

[96] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. Torr, “Bing: Binarized normed gradients for objectness estimation at 300fps,” in *CVPR*, 2014.

[97] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li, “Yfcc100m: The new data in multimedia research.” in *ACM Multimedia*, 2016.

[98] L. Duan, C. Wu, J. Miao, L. Qing, and Y. Fu, “Visual saliency detection by spatially weighted dissimilarity,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[99] J. Xiao, J. Hays, K. Ehinger, A. Oliva, and A. Torralba, “Sun database: Large-scale scene recognition from abbey to zoo,” in *CVPR*, 2010.

[100] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky, “Neural codes for image retrieval,” in *ECCV*, 2014.

[101] T. Mikolov, W.-t. Yih, and G. Zweig, “Linguistic regularities in continuous space word representations,” in *Proceedings of NAACL-HLT*, 2013.

[102] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” in *arXiv preprint arXiv:1301.3781*, 2013.

[103] Y. Boykov and V. Kolmogorov, “An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision.” *IEEE transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1124–1137, September 2004.

[104] Y. Boykov, O. Veksler, and R. Zabih, “Efficient approximate energy minimization via graph cuts,” *IEEE transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 12, pp. 1222–1239, November 2001.

[105] V. Kolmogorov and R. Zabih, “What energy functions can be minimized via graph cuts?” *IEEE transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 2, pp. 147–159, February 2004.

[106] C. D. Manning, P. Raghavan, and H. Schtze, *Introduction to Information Retrieval.* Cambridge University Press New York, 2008.

[107] MSR-Bing, “Image retrieval challenge dataset,” in *http://web-ngram.research.microsoft.com/GrandChallenge/Datasets.aspx*, 2013.

[108] Y. Jia, “Caffe: An open source convolutional architecture for fast feature embedding,” in *http://caffe.berkeleyvision.org/*, 2013.

[109] N. Aletras and M. Stevenson, “Representing topics using images,” in *NAACL-HLT*, 2013.

[110] G. G. Duffy, *Explaining Reading, Second Edition: A Resource for Teaching Concepts, Skills, and Strategies*. Guilford Press, 2009.

[111] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural*, 1997.