UvA-DARE (Digital Academic Repository)

Learning to search for images without annotations

Kordumova, S.

Publication date

2016

Document Version

Final published version

[Link to publication](#)

Citation for published version (APA):

Kordumova, S. (2016). *Learning to search for images without annotations*. [Thesis, fully internal, Universiteit van Amsterdam].

General rights

It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations

If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: <https://uba.uva.nl/en/contact>, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.

As humans we are adjusted to the environment around us. We easily recognize what we see. When the environment is captured in an image, with just a glimpse at the image we can recognize its contents. Machines on the other hand are not aware of the visually depicted environment around us or in images unless trained to do so. To train a machine, a usual approach is to annotate images with the concepts they capture and feed them into a machine learning algorithm.

The focus in this thesis is not to use directly annotated images with the concepts we aim the machine to recognize. We investigate various other sources of information for training. We avoid annotated text on images created by well-instructed human annotators (chapters two and three), or directly annotated examples all together (chapters four and five). We investigate these approaches for recognizing concepts and scenes in images with the goal of image search.

ISBN 978-608-4784-15-9

Learning to Search for Images without Annotations

Svetlana Kordumova

Learning to Search for Images without Annotations

Svetlana Kordumova

Learning to Search for Images without Annotations

Svetlana Kordumova

This book was typeset by the author using L^AT_EX 2_ε.

Printing: Evropa92

Copyright © 2016 by S. Kordumova.

All rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopy, recording, or any information storage and retrieval system, without permission from the author.

ISBN 978-608-4784-15-9

Learning to Search for Images without Annotations

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Universiteit van Amsterdam
op gezag van de Rector Magnificus
prof. dr. ir. K.I.J. Maex
ten overstaan van een door het College voor Promoties
ingestelde commissie,
in het openbaar te verdedigen in de Agnietenkapel
op woensdag 2 november 2016, te 12:00 uur

door

Svetlana Kordumova

geboren te Shtip, Macedonië

Promotiecommissie:

Promotor:	Prof. dr. ir. A.W.M. Smeulders	University of Amsterdam
Co-promotor:	dr. C.G.M. Snoek	University of Amsterdam
Overige leden:	Prof. dr. A. Del Bimbo	University of Firenze
	Prof. dr. ing. Z.J.M.H. Geradts	University of Amsterdam
	Prof. dr. M. Worring	University of Amsterdam
	dr. L. Aroyo	VU University Amsterdam
	dr. E. Gavves	University of Amsterdam

Faculteit der Natuurwetenschappen, Wiskunde en Informatica

UNIVERSITEIT VAN AMSTERDAM

This research is supported by the Dutch national program COMMIT, and the STW STORY project. This work was carried out in the ASCI graduate school, at the Intelligent Systems Lab Amsterdam of the University of Amsterdam.
ASCI dissertation series number 360.

COMMIT/

“It was the best of times, it was the worst of times, it was the age of wisdom, it was the age of foolishness, it was the epoch of belief, it was the epoch of incredulity, it was the season of Light, it was the season of Darkness, it was the spring of hope, it was the winter of despair...”

- Charles Dickens (1812-1870)

Contents

1. Introduction	1
2. Best Practices for Learning Video Concept Detectors from Social Media	
Examples	9
2.1 Introduction	9
2.2 Related Work	12
2.2.1 Data Sources	12
2.2.2 Positive Examples	12
2.2.3 Negative Examples	14
2.2.4 Learning Strategies	15
2.3 Experimental video search engine	16
2.3.1 Harvesting social media sources	16
2.3.2 Positive example selection strategies	17
2.3.3 Negative example selection strategies	20
2.3.4 Learning strategies	21
2.4 Experiments	22
2.4.1 Experiment 1: What source?	22
2.4.2 Experiment 2: What positive examples?	23
2.4.3 Experiment 3: What negative examples?	23
2.4.4 Experiment 4: What learning strategy?	24
2.4.5 Test Set	24
2.4.6 Implementation details	24
2.5 Results	25
2.5.1 Experiment 1: What source?	25
2.5.2 Experiment 2: What positive examples?	25
2.5.3 Experiment 3: What negative examples?	28
2.5.4 Experiment 4: What learning strategy?	31
2.5.5 Benchmark Comparison	34

2.6	Conclusion	34
3.	Exploring the Long Tail of Social Media Tags	37
3.1	Introduction	37
3.2	Related Work	38
3.3	What tags constitute the long tail?	41
3.4	Utilizing the long tail	43
3.4.1	Augmenting rare tags	43
3.4.2	Tag Relevance	45
3.4.3	Learning Detectors	47
3.5	Conclusions	49
4.	Pooling Objects for Recognizing Scenes without Examples	51
4.1	Introduction	51
4.2	Related work	53
4.3	Pooling objects for scenes	55
4.4	Experiments	59
4.4.1	Datasets	59
4.4.2	Implementation details	61
4.4.3	Semantic pooling and appearance pooling	61
4.4.4	Hierarchy pooling	62
4.4.5	Position pooling	64
4.4.6	Comparison with attributes	66
4.5	Conclusions	66
5.	Searching Scenes by Abstracting Things	69
5.1	Introduction	69
5.2	Related Work	71
5.3	Things syntax	74
5.3.1	Query by abstract linguistic statements	75
5.3.2	Query by abstract block illustrations	77
5.4	Datasets	78
5.5	Experiments	83
5.5.1	Comparing automatically proposed and manually annotated things	83
5.5.2	Query by abstract statements	86
5.5.3	Query by block illustrations	88
5.5.4	Abstract vs Semantic	90
5.6	Conclusions	93
6.	Summary and Conclusions	95
6.1	Summary	95

6.2	Conclusions	97
A.	Appendix	99
A.1	SocialZap	99
A.1.1	Introduction	100
A.1.2	System overview	101
A.1.3	Conclusion	104
	Bibliography	105
	Samenvatting	113
	Acknowledgments	117

1

Introduction

As humans we are adjusted to the environment around us. When walking into a park, one would not be surprised to see trees and benches and kids playing. If looking for a shadow, one would most naturally move towards a tall tree, and will easily recognize a fountain for drinking water. If tired, there are benches to sit on and rest. We are aware of the environment due to a lifelong process of seeing and learning.

We rationalize what we see from the many different alternative explanations, which may apply as the reasons of being for that part of the world as it is. Objects are always positioned on the ground, thanks to the gravitational laws of physics. The sky and the sun are above because of other physics laws. Trees grow tall to collect sun energy. Man-made objects are designed to serve a purpose. Fountains will have limited heights compared to the heights of humans, for the purpose of drinking. A bench will be long, since it is designed for several people to sit on. Kids go to parks, kinder gardens, schools, but not so much in bars. We have learned what objects belong in what environment (semantics), and we also have an expectation on how the objects will be arranged (layout).

Humans reconstruct the structure of the environment in images, by means of paintings, printed photographs or digital images. In the famous “Cafe Terrace at Night” painting of Vincent van Gogh, Figure 1.1 (a), the artist nicely portrays a popular cafe, a building structure, a lighted shop, and tree branches surrounding the place. The famous “Abbey Road” photograph of The Beatles, Figure 1.1 (b), shows the band members crossing a street, with cars and trees on the sides. With just a glance at an image, we can recognize the environment re-created in it.

Machines are not aware of the visually depicted environment unless trained to do so. Taking into account that a child goes through a long process of learning to visually understand the environment, it is quite challenging to train a machine with the same

Fig. 1.1: Examples of visually depicted environment re-created in images. Textual descriptions of concepts and scene categories next to the images describe the visual content.

ability of visual recognition. For machines to learn to understand images, they need to be told what is in an image. A child is told that a “bed” is for sleeping, and a “cup” is for drinking. One approach is to use *training data* of digital images that capture the environment, *annotate* them with text of what are they capturing, and feed it into a machine-learning algorithm. The goal of the algorithm is to connect the information of digital images and the annotated text. This is a very challenging problem, known as the semantic gap.

To bridge the semantic gap and to describe images with language, considerable effort has been made [57, 59, 107]. We categorize related work into directly learning machines to find image structures [56, 111], or alternatively into precomputing structural information in the form of image features, and using these to learn machines recognize images [59, 96, 120]. In this thesis we follow the later learning approach. Inspired by the human reasoning for recognition, we investigate two types of features from images, *semantic* and *layout*.

Interestingly, humans can also recognize things without seeing it, or have seen it before. We do so by associating it with other information. If someone is holding an object close to the ear while talking, we easily guess it is a phone. The conclusion holds even when we do not directly see the object as we are used to filling in occlusions. When you go into the house of a friend for the first time, and be asked to sit on a sofa opposite a coffee table and a television, you assume you are in the living room. In a theatrical show called pantomime, performers express meaning through gestures, without showing any objects or talking, and still, we understand. This shows that various other forms of information can help us recognize objects and the environment.

Our main approach in this thesis is to investigate various other sources of information to train machines when recognizing what is in an image. We avoid annotated text on images created by well-instructed human annotators (chapters two and three), or directly annotated examples all together (chapters four and five).

We focus our effort on image search. Image search looks for images related to a query, with a query being a textual description in the form of separate words or sentences, or an image, in order to find similar images to the query image. With the growth of large image collections, image search with a specific content becomes important for many applications. It helps someone look for images of a “mountain”, “temple” or a “party” on the web. It will help a photographer search images through her large personal collection. It will help an architect search for similar buildings to one in the image query. In this thesis, we investigate and evaluate image search on standard image datasets up to date, by either searching for *concepts* present in images, or for *scenes* that describe the environment re-created in the image. See examples of concepts and scenes in Figure 1.1.

A concept can be any depicted view from the environment e.g., “snow”, “dancing”, “people”, “fountain”, “running” are all concepts that can be re-created in images. For an example, in Figure 1.1 (b), the concepts “people”, “walking”, “cross-walk”, “cars” are “present”. Our goal is to train machines to automatically assign concept names to images. To do so, images annotated with concept names are needed as training data. Providing precise human annotations for all concepts is an expensive process. In an attempt to approximately measure how many diverse concept classes there are, Biederman counted 30,000 entries from an English dictionary [10]. To employ human annotators for all concepts is almost infeasible, and blocks the ideal that all concepts can be learned with manual annotations.

On the lookout for an alternative, we exploit the human need for social interaction. We as humans like to be surrounded by other people and share personal experiences and information. The appearance of various social networking platforms and their growth at an explosive rate, nicely confirms to the fundamental human need for social belonging and interpersonal communication. Flickr, Instagram, YouTube, *etc.*, allow users to upload and share their images and videos online with other users. Astonishing numbers have been reported for Flickr, with 2 million images being uploaded every day on average, with total of 5.26 billion images up to date [1]. On these platforms, users provide concept tags to the images they upload. With all this in mind, we pose our first question:

Can we learn machines to search images with socially tagged data?

We address this question in the second chapter of this thesis. The potential of social media tags as image annotations has been recognized by many, to name a few

[66, 117, 142]. Inspired from the success of previous works, in this chapter we focus on investigating best practices for learning effective concept detectors from tagged data. We pose and investigate four research questions. We start with investigating what tagging source is better suited for learning concept detectors, tagged videos (YouTube) or tagged images (Flickr). We continue with analyzing tagged images as positive training examples and tagged images as negative training examples. At last, we pose the question what learning algorithm is most suited for learning from tagged data. The study is based on a complete system that learns visual concept detectors from social media only, and evaluated on images extracted from the TRECVID benchmark video collection [108]. Importantly for this thesis, all questions we investigate provide a basis for how to use social media content as an alternative source for training machines to recognize concepts. The practices we recommend have led to a winning entry on the TRECVID 2014 benchmark for the semantic indexing task with no annotations.

While some concepts appear quite often as tags in social media, other concepts are rarely used as tags, following a long tail distribution. For frequent concepts, learning machines to recognize visual content shows promising performance [52, 68, 143]. However, it is not clear how will learning concept detectors perform for rare tags. Thus, we come to the second question of this thesis:

What happens when socially tagged images are rare for some concepts?

This research question is addressed in Chapter 3. Here we first analyse the correlation of tags frequency and their semantic category. We analyse tags from three different categories: *objects*, *scenes* and *fine-grained animals*. Interestingly, we also find a correlation between the tags occurrence and their n-gram composition. We also measure the performance of tags uniformly sampled from the long tail distribution on two multimedia applications. To improve the performance for rare tags, we propose to augment them with semantic knowledge from an ontology. Notably, this chapter shows that there is a limitation to the general optimism that machines can learn visual detectors for all concepts from social media. There are concepts that rarely appear as image tags, and should be treated differently. In this chapter we present one approach on dealing with rare tags.

One use case employing the potential of learning concept detectors directly from social media is presented in Appendix A. In the appendix we describe a demo system that fully-automatically learns the interest of television viewers from textual microblog posts, and transfers it to the visual channel without any manual effort in the process. Once the interesting concept detectors are applied on the visual channel, users can easily browse through a television broadcast, and directly watch the popular fragments.

While in the second and third chapter we discussed image search for all concepts, in the next two chapters the focus is exclusively on scene categories. The goal is to recognize scenes in images without using any scene images as training data. Since a scene is a composition of many objects, we rely on semantic object categories and pose the question:

Can we recognize scenes with object categories and without examples?

This question is addressed in Chapter 3. We investigate using over 15,000 object classifiers of a large image collection [94], and train a convolutional neural network [111]. Based on the reasoning that not all object classifiers will contribute equally in recognizing scenes, for example, to recognize a beach scene objects like “sand”, “ocean”, “sky” would be beneficial, whereas “bed”, “apple”, “church” would have a negligible or no influence at all, we propose and we analyse four ways of pooling objects for scene recognition. To pool object classifiers we rely on: scene-to-object semantic relationship; the probability of object appearance; the ontological hierarchy of objects; and the probability of objects appearance in image positions. We additionally demonstrate a new scene retrieval scenario, where instead of only searching for images of a specific scene, we also specify what objects should be present, and approximately on what position they should appear. Distinctly from previous works that used attributes to recognize scenes, [42, 59, 87], in this chapter we recognize scenes by pooling object categories only.

In the fifth chapter of the thesis we go to the extreme of using no annotations and no learning at all to search scenes. Where recent works rely on object identification, [23, 63, 92, 94], here we represent scenes independent of object identities. We rely on the layout of objects only. Since we do not use object identities we refer to all objects as *things*. We pose our last research question:

Can we search scenes with things layout only?

The layout of things reveals a preference for a scene-type-specific composition. A “theatre” and a “marching band”, a “soccer crowd” and a “forest” will all demonstrate spatial regularity of the things they are composed of, albeit a spatial regularity of a different kind each. The interior of a sleeping room will usually demonstrate a limited number of spatial layouts, which contributes to the recognition of the scene. For the spatial layout of things, we rely on modern object proposals, [4, 75, 115] and their immediately observable properties: position, size, aspect ratio and colour, and these only. We name

the ensemble of all things properties *things syntax*. We translate the things syntax into abstract linguistic statements and abstract block illustrations. We do so by using simple rules and no learning at all. Although the descriptions we generate are poor and abstract, they are still understandable by humans, and they can be exploited for searching scenes. The analysis and insight we provide in this chapter, shows the potential to combine the things layout information, and the recognition of scenes without the need to keep semantics of the sort of object types or attributes. We show that next to object types, there is another source of information defining what makes a scene.

List of publications

- S.Kordumova, X. Li, and C. G. M. Snoek. Best Practices for Learning Video Concept Detectors from Social Media Examples. In *Multimedia Tools and Applications*, 2015. S. Kordumova took care of all aspects of the paper, X. Li and C.G.M. Snoek provided supervision and insight. [Presented in Chapter 2].
- S. Kordumova, J. van Gemert, and C. G. M. Snoek. Exploring the Long Tail of Social Media Tags. In *Proc. Int’l. Conf. Multimedia Modelling*, 2016. S. Kordumova took care of all aspects of the paper, J. van Gemert and C.G.M. Snoek provided supervision and insight. [Presented in Chapter 3].
- S.Kordumova, T. Mensink, and C. G. M. Snoek. Pooling Objects for Recognizing Scenes without Examples. In *Proc. ACM Int’l. Conf. Multimedia Retrieval*, 2016. S. Kordumova took care of all aspects of the paper, T. Mensink and C.G.M. Snoek provided supervision and insight. (*Best paper award*). [Presented in Chapter 4].
- S. Kordumova, J. van Gemert, and C. G. M. Snoek, and A. W. M. Smeulders. Searching Scenes by Abstracting Things. S. Kordumova took care of all aspects of the paper, J. van Gemert, C.G.M. Snoek and A.W.M. Smeulders provided supervision and insight. [Presented in Chapter 5].
- S. Kordumova, C. Kofler, D. C. Koelma, B. Huurnink, B. Freiburg, J. Kleinveld, M. van Rijn, M. van Deursen, M. Larson, C. G. M. Snoek. SocialZap: Catch-up on Interesting Television Fragments Discovered from Social Media. In *Proc. ACM Int’l. Conf. Multimedia Retrieval*, 2014. S. Kordumova took care of all aspects of the paper, C. Kofler helped with the textual analysis, D. C. Koelma, B. Freiburg, J. Kleinveld, M. van Rijn and M. van Deursen assisted in the programming, B. Huurnink, M. Larson and C. G. M. Snoek provided supervision and insight. [Presented in Appendix A].

-
- C. G. M. Snoek, K. E. A. van de Sande, D. Fontijne, A. Habibian, M. Jain, S. Kordumova, Z. Li, M. Mazloom, S. Pintea, R. Tao, D. C. Koelma, and A. W. M. Smeulders. MediaMill at TRECVID 2013: Searching Concepts, Objects, Instances and Events in Video. In *the 11th TRECVID Workshop*, 2013. [Not included in the thesis].
 - S.Kordumova, X. Li, and C. G. M. Snoek. Evaluating Sources and Strategies for Learning Video Concepts from Social Media. In *Int’l. Workshop on Content-Based Multimedia Indexing*, 2013. [Not included in the thesis].
 - C. G. M. Snoek, K. E. A. van de Sande, A. Habibian, S. Kordumova, Z. Li, M. Mazloom, S. Pintea, R. Tao, D. C. Koelma, and A. W. M. Smeulders. The MediaMill TRECVID 2012 Semantic Video Search Engine. In *the 10th TRECVID Workshop*, 2012. [Not included in the thesis].
 - S.Kordumova, X. Li, and C. G. M. Snoek. Learning Concepts from the Web: Some Frames are More Important than Others. In *Proc. Annual Conf. of the Advanced School for Comput. and Imag.*, 2012. [Not included in the thesis].

Best Practices for Learning Video Concept Detectors from Social Media Examples

2.1 Introduction

Many videos are produced every day. Consider for example the amount of frames that are contained in the 100 hours of Internet videos that are being uploaded to YouTube every minute. In order to pinpoint a specific query to arbitrary fragments of a video, semantic labels at a shot or even frame level are required. This leaves us no other choice but to devise machine tagging mechanisms that can detect visual concepts such as *animal*, *building*, and *snow* at the frame level [15, 106]. The state-of-the-art in video concept detection is to learn SVM classifiers from manually labeled frames represented by visual code features [44, 120]. However, the expense of manual labeling results in training examples with limited availability. As a consequence, the performance of concept detection is bounded to a narrow application domain where a limited array of concepts can be reliably detected [133]¹.

The potential of automatically obtaining training data from the web for all possible video concepts one can think of is recognized by many [24, 61, 68, 99, 102, 106, 118, 127, 142]. To promote progress in this field of research, the well known benchmark for video retrieval TRECVID [106], even started a dedicated task for learning video concepts without manually provided annotations in 2012. To reach this goal, one promising

¹Published in *Multimedia Tools and Applications*, 2015 [52].

line of research is to automatically acquire training examples from social media, where many socially tagged videos and images exist. Ulges *et al.* were among the first to learn video concepts from YouTube [118]. In their system, if social tags of a video match a given concept, all frames of the video are considered as positive examples of that concept. Setz and Snoek [102] conducted a pilot study on learning video concepts from Flickr images, directly treating images labeled with the concept as positives. However, social tags are known to be unreliable and often irrelevant with respect to the visual content they are describing [66, 69, 71, 117, 127, 142]. Hence, for learning meaningful concept detectors from social media, selecting appropriate examples from a proper data source is crucial.

As noted by Yang *et al.* [133], the performance of a concept detector could degenerate severely if the training and the test videos are from different genres, *e.g.*, broadcast news and documentaries. However whether this will also hold true in a cross-source scenario, say applying image classifiers on video data, has not been investigated yet in the literature. We observe prevailing usage of socially tagged videos as the training source of choice [61, 116–118, 127, 139], even though selecting positive training examples from videos is more difficult than selecting positive examples from images. This is not only because video tags are noisy, but also the question of how to propagate tags to the frame level is still open [6, 61]. In fact, the question whether using tagged videos are more beneficial than simply using tagged images has not been addressed in the literature.

The vast collections of images and videos in social media offer many possibilities for learning visual detectors. Some researchers have focused on selecting relevant positive examples [69, 110, 117, 142], where others focus on selecting negative examples [66, 132]. The authors of [39, 61] incorporated the relevance in the learning methods. Among all these possibilities, one may wonder *How to learn effective video concept detectors from social media examples?*

To acquire accurate positive examples from social media, a common approach is through a retrieval process, where socially tagged examples are first ranked in terms of their estimated relevance scores with respect to a given concept [69, 110, 117, 142]. Then, the top ranked proportion of the examples is preserved. In [69, 142], for instance, a fixed number of examples are selected for every concept, while [117] just tried a varying amount of frame fractions. Some even ignored the unreliability of the social labels, and directly used tagged videos [118] or tagged images [102] as positive examples. However, what strategy is the best for selecting positive examples is unsettled.

For selecting negative training examples, a common strategy is to randomly sample images not tagged with the concept name [47, 61, 102, 117, 118, 142]. The classifiers usually misclassify negative examples which are visually similar to the positives. Therefore, inclusion of such misclassified and thus relevant negatives could lead to bet-

Fig. 2.1: In this chapter we perform an experimental study with a video search engine to arrive at best practices for learning video concept detectors from social media examples.

ter concept detectors. Li *et al.* [68] select relevant negatives with an iterative bootstrap approach, where the top misclassified images are selected as relevant negatives. Yan *et al.* [132] select the bottom ranked examples based on some similarity metric to the positives as reliable negatives. Notice that both methods use expert-labeled positive training examples. To the best of our knowledge, learning video concept detectors with relevant negatives, relying fully on social tagged data, has not been investigated in the literature.

Instead of focusing on selecting relevant positive or negative examples [61, 68, 117, 132], some cope with the noisy social data by machine learning strategies that address the relevance during the detector training phase [38, 39, 61]. For example, the authors of [61] use multiple instance learning, where tagged videos are treated as bags, and the frames as instances. Hu *et al.* [38] learn an image ranking model from tagged image pairs with preference relations. Grauman *et al.* [39] learn relative tags on images with RankSVM [45], where ordered pairs are created from a ranked list of examples for the learning procedure. However, whether selecting relevant positive and negative examples a priori is better than relying on a learning strategy that optimizes the selection during training is unknown.

In this chapter, we aim to identify best practices for learning video concept detectors from social media examples. We structure this chapter as an experimental study with a complete system that learns visual concepts from socially labeled media examples, see Figure 2.1. In an initial version of this work [51], we investigated positive example selection strategies only. We present a more comprehensive study by evaluating three negative example selection strategies and three learning strategies on a larger set of Internet videos. These new elements allow us to identify a set of best practices, and consequently lead to the winning system in the TRECVID 2013 benchmark for concept detection without annotations. Before describing our experimental study we first review related work.

2.2 Related Work

We structure our literature review in terms of the choices one has to make for learning video concept detectors from social media examples. In Section 2.2.1 we briefly consider the data sources used. In Section 2.2.2 we review strategies for the selection of positive training examples. We consider the strategies for selecting negative training examples in Section 2.2.3. Learning strategies are summarized in Section 2.2.4.

2.2.1 Data Sources

When harvesting visual training examples from social media, two sources have been used in the literature, namely tagged videos [61, 116–118, 139] and tagged images [24, 47, 68, 102, 142]. Ulges *et al.* [118] learn video concept detectors using YouTube videos by directly treating social tags as relevant labels for all video frames. Wang *et al.* [127] follow a similar approach, but require a set of manually labeled videos to bootstrap the learning process. Multiple instance learning was investigated in [61] treating socially tagged videos as bags, and frames as instances. Setz and Snoek [102] investigate whether Flickr images can be exploited as a direct training source for learning video concepts. Li *et al.* [66, 68] harvest relevant tagged images as positive and as negative training examples. Notice that none of the above works considers which *source* is a better choice. Therefore we raise the first research question:

Research Question 1:

What visual tagging source is most suited for selecting training examples for learning video concept detectors?

This study covers two representative approaches as baselines *i.e.*, Ulges *et al.* [118] and Setz and Snoek [102], naming them as the ALL-FRAME and the ALL-IMAGE respectively.

2.2.2 Positive Examples

For video it is easy to observe that even when tags are relevant at the video level, it does not imply that the tags are relevant for all shots and frames as well. Ulges *et al.* [117] model the relevance of video (key)frames with density estimation in the feature space. Since the density estimation is tag independent, for a video labeled with multiple tags, the method [117] cannot tell which tag is relevant w.r.t. which frame. Also, the

prior information is unknown, so multiple fractions of the top ranked frames are used and the best fraction is selected on the expense of a manually labeled validation set. Since manual annotation is required for validating the prior values, this approach is unscalable for many concepts. Zhao *et al.* [139] suggest new tags on a video level by propagating tags from similar videos, ignoring the fact that not all frames of the video are relevant to the tag [61, 106, 117]. In order to select the relevant frames, we need to estimate tag relevance on a frame level. Given that a video is a composition of continuous frames, some have used the temporal dimension either as a feature to describe the video [97, 104], or to refine the end results of the positive examples with the scores of the temporal neighboring frames [61]. Since temporal features such as motion [125] and audio [43] are not applicable on images, we do not consider temporal information to ensure a fair comparison between the two sources at the image/frame level.

Positive example selection of tagged images has been addressed by many. For instance, the semantic field method [142] determines tag relevance in terms of tag-wise similarity. However if applied on videos, this approach may identify relevant tags for a video, but not for individual frames, since the semantic field works only on a video level. The authors of [24] detect spam, polysemous and synonym tags to filter out noisy tagged images. Their method relies on the assumption that users provide multiple tags per image. However the statistics presented in [103] show that in a large sampled Flickr collection 2/3 of the images have one to three tags at most. Thus relying on methods with tag co-occurrence may result in poor conclusions. Li *et al.* [66] learn the relevance of tags accompanying an image using a neighbor voting algorithm, by finding neighbors in the visual feature space. They exploit the observation that similar tags issued for similar images are reliable, by accumulating tag votes over visually similar images. Including a diverse set of visual features in the neighbor voting algorithm further improves the effectiveness of tag relevance [67]. An alternative to tag relevance is proposed by Liu *et al.* [71]. Their method is also founded on neighbor voting, but the neighbors are weighted with a Gaussian function. Notice that [71] restricts the neighbors to be images labeled with the tag, while [66] exploits the entire collection.

Since there are many possible choices one can make for selecting positive examples from tagged sources when learning video concept detectors, we raise the second question:

Research Question 2:

What strategy should be used for selecting positive examples from tagged sources when learning video concept detectors?

In this chapter we choose the neighbor voting algorithm [66] as the tag relevance measure because of its reported leading performance [119]. We evaluate three strategies for positive example selection, two based on tag relevance and one with random sampling.

2.2.3 Negative Examples

While the problem of selecting positive examples from social media has been well investigated, the importance of negative examples has mostly been overlooked in the literature. Most concept detectors rely on simple random selection of images [47, 72, 102, 142] or videos [61, 117, 118] not tagged with the concept name.

The authors of [132] propose a *pseudo negative* selection of negative examples for video retrieval. The pseudo negative performs an iterative learning procedure by selecting reliable negatives from the worst matching examples identified by the classifier. Although sampling at the lower ranks probably yields reliable negatives, the classifier already ranked them correctly at the bottom, thus adding them to the training process can only yield minor improvements. Therefore, using other ranking measures for the examples, like tag relevance, and then selecting the bottom ranked examples is interesting to investigate.

Li *et al.* [68] propose a *negative bootstrap* approach, where relevant negatives are harvested from a large amount of social-tagged images. Since classifiers tend to misclassify negative examples which are visually similar to positive examples, including these relevant negatives boosts the performance. To that end, the negative bootstrap approach samples relevant negatives in an iterative manner. The misclassified negative examples are considered as relevant and thus leveraged in the consequent learning iteration. Using relevant negative sets result in visual classifiers with higher accuracy. While negative bootstrap is intended and shown to be effective for building image classifiers, its effectiveness for video concept detection is unknown. Moreover, the algorithm uses expert-labeled positive examples. To the best of our knowledge, a joint adventure of selecting relevant positive and relevant negative examples for video concept detection is non-existing. Therefore we raise the third research question:

Research Question 3:

What strategy should be used for selecting negative examples from tagged sources when learning video concept detectors?

In this chapter we evaluate three strategies for negative example selection, negative bootstrap, pseudo negative and random sampling from tagged image and tagged videos.

2.2.4 Learning Strategies

Since social tags can be ambiguous and subjective, tagged images and videos should be treated carefully when used as training data for learning video concept detectors. Many have focused on selecting relevant positive and negative examples before learning. Li *et al.* [61] bypass the selection problem by using Multiple Instance Learning (MIL) [101] to estimate the relevance of a video tag at the shot level. In a MIL setting, instances are organized into bags and it is the bags, instead of individual instances that are labeled for training. In [61] the tagged videos are considered as labeled bags, and the frames are the instances. If a video is labeled with a concept, all its frames are considered as positive, and if not, all its frames are considered as negative. However, the base MIL model [101] will hurt if videos are miss-labelled, and it is known that the social tags are imperfect and ambiguous. To overcome this obstacle, the authors of [61] calculate a tag correctness score and use it as a weight in the optimization function of the MI logistic regression. The tag correctness score is calculated by averaging the cosine similarity between the video tags and text snippets retrieved from a search engine with the target tag as a query. For a test frame, its predicted score is further smoothed using scores of its temporal neighbors. It is worth noting that for content based image retrieval, standard supervised learning is superior to MIL [91].

When learning a concept detector with multiple instance learning or Support Vector Machines [122], every positive and negative example used for training contributes equally in the optimization process. Dealing with noisy tagged data, it is potentially more robust to rely on a relative degree of relevance for each example. The initial idea of using relevant information of examples while learning by RankSVM, was originally proposed by Joachims [45] for web page ranking. Later the idea has been used together with multiple instance learning for finding regions of interest in the image [38]. The authors of [39, 49] use RankSVM to add relative information to tags, like *more then* or *less then*. Click data of image search engine is used with RankSVM to re-rank images [41].

Although different learning strategies have been investigated in many tasks, we have not seen a comparative evaluation on what strategy is most suited for learning video concept detectors from social media examples. Therefore the forth question arises:

Research Question 4:

What learning strategy should be used for learning video concept detectors from socially tagged data?

This study evaluates three learning strategies. We name the multiple instance learning approach of [61] as *Packed examples* learning, since frames are packed into bags. As

Paired examples learning we refer to models learned from example pairs with preference relations [39]. We generate the ordered pairs based on the tag relevance scores of the examples. Moreover, we investigate whether bypassing the selection problem of relevant examples, as done in *Packed examples* and *Paired examples* learning is a better strategy than *Refined examples* learning, where relevant training examples are selected a priori and then used to train a one-versus-rest Support Vector Machines [122].

To answer the four research questions and obtain best practices, we structure this chapter as an experimental study with a complete video search engine that learns concept detectors from social media examples. We identify four key components of such a system, 1) harvesting social media sources, 2) selection of positive training examples, 3) selection of negative training examples and 4) learning strategies to obtain concept detectors. The key components of our video search engine are highlighted in Figure 2.1 and detailed next. Conceptual diagrams of the individual components are given in Figure 2.2, Figure 2.3, and Figure 2.4, in sequence.

2.3 Experimental video search engine

2.3.1 Harvesting social media sources

We harvest two type of media sources, tagged videos and tagged images. The tagged videos are collected from YouTube, which is one of the most popular service for video sharing. The tagged images are selected from Flickr as one of the most popular sharing service for images. In this study, we consider 20 video concepts, covering objects like *Car*, *Plant* and scenes like *Outdoor*, *Landscape*. The set of 20 concepts was chosen to consist of single-word concepts only, for more precise tag relevance estimation, and restricted to those concepts with the most frequent number of positive examples in the data sets considered [66, 106].

Tagged Videos. In order to construct a diverse set, we collect videos retrieved by four distinct ranking criteria provided by YouTube, i.e., view count, relevance, date published and user rating. Consequently, we obtain for each of the 20 concepts four lists of retrieved videos and their metadata, containing videos id, tags, author, video duration etc. The date published ensures that new instances of concepts like *Car* and *Building* models are covered. We include in our dataset the top 50 retrieved videos from each of the four ordering criteria. Hence, for each concept we download the most viewed, most relevant, most recently uploaded and best rated videos. We shot segment each video and define the middle frame of each shot as its keyframe. Since we want to show the influence of frame selection, we maintain only those videos that have at least two shots. This process resulted in 200 hours of web video and 130K keyframes.

Tab. 2.1: Number of positive examples for 20 concepts in both the Tagged Videos and Tagged Images datasets.

Concept	YouTube Videos	Video Frames	Flickr Images
Animal	191	7506	7506
Beach	185	6554	6554
Building	197	5207	5207
Car	192	8508	8508
Child	174	6938	6938
City	169	8326	8326
Face	169	7210	7210
Hand	179	5835	5835
Landscape	181	4005	4005
Mountain	181	6445	6445
Oceans	150	5755	5755
Outdoor	184	6103	6103
Plant	151	5139	5139
Road	170	8047	8047
Sky	154	6261	6261
Snow	177	7467	7467
Sports	196	6685	6685
Streets	141	7493	7493
Trees	158	5805	5805
Vehicle	184	4621	4621

Tagged Images. We adopt the Flickr image collection from [66]. The images are of medium size with width or height fixed to 500 pixels. A subset is generated such that the number of images for each concept is maintained the same as the number of frames from the *Tagged Videos* dataset. In total, the *Tagged Images* collection has 130K images.

Table 2.1 shows the number of videos and images labeled with a given concept in the two sources.

2.3.2 Positive example selection strategies

We investigate three selection strategies, see Figure 2.2. Given a specific concept ω , we select positive training examples from the two sources described in Section 2.3.1. Let x be such an example in consideration. Depending on the source of training data, x is either an image labeled with ω or a frame extracted from a video labeled with ω . Let L_ω be the set of n examples labeled with ω .

Fig. 2.2: Three strategies for positive training example selection for learning video concept detectors from social media: random, relevant ad-hoc and relevant cut-off.

1. Random selection. One straightforward selection strategy is to rely on the social tags and to randomly select k labeled examples with the concept ω . The set of randomly selected positive examples P_ω is

$$P_\omega = \{\forall x \in C \mid C = \{x_1, x_2, \dots, x_k\} \subset L_\omega\}. \quad (2.1)$$

2. Relevant ad-hoc selection. As aforementioned, simply treating x as a positive example is problematic. Therefore another strategy is to calculate tag relevance per example, and then select the top ranked examples. For every example x labeled with concept ω , we use the multi-feature variant of the neighbor voting algorithm [67] to compute tag relevance scores. The tag relevance scores allow us to rank the examples such that the most relevant examples are deemed to be placed at the top. Therefore we can select the k top ranked examples as positives:

$$P_\omega = \{\forall x \in L_\omega \mid \text{rank}(x) \in 0, 1, \dots, k-1\}. \quad (2.2)$$

3. Relevant cut-off selection. Strategy 2. does not answer how many of the k examples should be selected from the ranked list. Validating the selection size comes with the expense of manual annotation for the validation set, making this approach unscalable for many concepts. Instead of setting an ad hoc threshold k [69, 142], or validating it with multiple runs [117] with the expense of manual annotation, in this work we consider a simple strategy to determine a cut-off automatically. To this end, we calculate which example x should be selected based on a Bayesian decision rule. For each example, we use s to denote its tag relevance score and r to denote the corresponding rank, where $r = 0, \dots, n-1$. We observe that the first ranked image after tag relevance is very often a positive response. Hence, we assume this first-hit example to be a correct reference point for estimating the probability of the other examples being positive. In that regard, we introduce a binary random variable y , where $y = 1$ means x is positive, and 0 oth-

erwise. The problem of positive example selection can then be simplified to estimating the conditional probability $p(y = 1|x)$. With the Bayesian decision theorem, we define the selection rule simply as:

$$\begin{cases} x \text{ is selected, if } \frac{p(y = 1|x)}{p(y = 0|x)} > 1, \\ \text{unselected, otherwise.} \end{cases} \quad (2.3)$$

For computing $p(y = 1|x)$, we have access to two observations with respect to x , *i.e.*, the relevance score s , and the corresponding rank r . Using a single observation is limited, since the scores are discriminative but less robust, while the quantized ranks tend to be more robust but less discriminative. Hence, we consider their combination, which should result in a better estimation of $p(y = 1|x)$. Using probability algebra, we have $p(y|x) = p(y|s, r) = p(s, r|y)p(y)/p(s, r)$. For $p(s, r|y)$, we make a practical simplification by estimating it through $p(s|y) \cdot p(r|y)$. We also expand $p(s|y)$ and $p(r|y)$ using Bayes' theorem. Accordingly, we rewrite

$$p(y|x) \approx \frac{p(s|y)p(r|y)p(y)}{p(s, r)} = \frac{p(y|s)p(y|r)p(s)p(r)}{p(s, r)p(y)}, \quad (2.4)$$

For the unknown prior $p(y)$, we assume an uniform prior, and the decision function becomes

$$\frac{p(y = 1|x)}{p(y = 0|x)} = \frac{p(y = 1|s)p(y = 1|r)}{p(y = 0|s)p(y = 0|r)}. \quad (2.5)$$

With the intuition that examples with larger tag relevance scores and higher ranks are more likely to be positive, we approximate $p(y = 1|s)$ as

$$p(y = 1|s) \approx \frac{s}{s_{max}}, \quad (2.6)$$

where s_{max} is the score of the top ranked example, and we compute $p(y = 1|r)$ as

$$p(y = 1|r) \approx 1 - \frac{r}{n}. \quad (2.7)$$

Accordingly, for concept ω we create a positive examples set P_ω as

$$P_\omega = \{\forall x \in L_\omega \mid \frac{p(y = 1|x)}{p(y = 0|x)} > 1\}. \quad (2.8)$$

In contrast to [69, 117, 142], the selection rule defined in Equation (2.8) sets the cut-off automatically, without the need to tune k .

We visualize the three positive example selection strategies in Figure 2.2.

Fig. 2.3: Three strategies for negative training example selection for learning video concept detectors from social media: random, negative bootstrap and pseudo negative.

2.3.3 Negative example selection strategies

We investigate three strategies for selecting negative examples for a specific concept ω , see Figure 2.3. We notate the set of examples x not tagged with the concept ω as $\overline{L_\omega}$.

1.Random selection. The negative example selection strategy most commonly used in the literature is random sampling. We define the set of randomly selected q negative examples as

$$N_\omega = \{\forall x \in C | C = \{x_1, x_2, \dots, x_q\} \subset \overline{L_\omega}\}. \quad (2.9)$$

2.Pseudo negative. In order to acquire negatives which are more informative than the random negatives, we consider examples labeled with the concept name, but having low tag relevance scores. The basic idea is that images/videos tagged with the concept have low tag relevance because they differ from the other images tagged with the same concept, *i.e.* most of their visual neighbors do not have the same tag associated. However, the examples with low tag relevance score can still be informative since they are tagged with the concept name. Therefore it is interesting to investigate whether examples with low tag relevance if used as relevant negative examples will improve the performance of the concept detectors. The pseudo negative approach is inspired by to [132], in terms of selecting the bottom ranked examples, but with a difference in the ranking procedure. In [132] the bottom ranked images from a classification score are selected as negatives, in an iterative procedure. We select the negatives only in one step, using the tag relevance score to rank the examples tagged with the concept name, thus avoiding the expensive iterative learning. Using the decision rule calculated as in Equation. 2.5, we create the pseudo negative set as

$$N_\omega = \{\forall x \in L_\omega \mid \frac{p(y=1|x)}{p(y=0|x)} < 1\}. \quad (2.10)$$

Fig. 2.4: Three learning strategies for video concept detectors with training data collected from socially tagged media: packed examples, paired examples and refined examples learning.

3.Negative bootstrap. As mentioned in Section 2.2.3, using relevant negatives can improve the retrieval performance. Therefore as another selection strategy we evaluate the negative bootstrap [68] into our video search engine, for both tagged images and tagged videos. We keep the positive examples P fixed. The negatives are selected in an iterative learning procedure. In iteration t we denote the learned model as M_t , and a randomly sampled collection from $\overline{L_\omega}$ with γ examples as C_t . In each step t , the set of negative examples is

$$N_{\omega,t} = \{\forall x \in \text{top-misclassified}(C_t, M_{t-1}, q) | C_t = \{x_1, x_2, \dots, x_\gamma\} \subset \overline{L_\omega}\}. \quad (2.11)$$

Where $\text{top-misclassified}(C_t, M_{t-1}, q)$ are the top q elements from a random subset C_t with γ elements of $\overline{L_\omega}$, classified by the previous learned model M_{t-1} as positives. Moreover, in each step of the negative bootstrap process, relevant negatives are selected.

We visualize the three negative example selection strategies in Figure 2.3.

2.3.4 Learning strategies

In order to answer the research question *what learning strategy should be used for learning video concept detectors from socially tagged data?* we consider three learning strategies, see Figure 2.4. We investigate whether selection of relevant positive and negative examples before learning concept detectors is more beneficial than using learning strategies that incorporate the relevance in the optimization function of the classifier.

1.Packed examples. Multiple instance learning bypasses the selection problem of positive and negative examples when learning video concept detectors from tagged videos.

The tagged videos are considered as labeled bags, and the frames are the instances. If a video is labeled with a concept, all its frames are considered as positive, and packed into a positive bag. If a video is not labeled with a concept, it is considered as a negative bag, and its frames as negative instances. We employ the implementation of [101] into our video search engine, with the additional weight for video tag relevance as in [61].

2. Paired examples. RankSVM is a pairwise learning to rank approach that learns the ranking function from ordered pairs of examples. To train concept detectors for each concept, we derive a set of ordered visual examples:

$$R_\omega = \{\forall(x_i, x_j) \in (L_\omega, L_\omega) \mid \text{tagrelevance}(x_i) > \text{tagrelevance}(x_j)\}. \quad (2.12)$$

Each pair (x_i, x_j) denotes that x_i is more relevant than x_j since it has a higher tag relevance score. We use the implementation of [45] to learn video concept detectors both from tagged images and tagged videos.

3. Refined examples. We select relevant positive and relevant negative examples to learn a video concept detector. As classifier we employ a one-vs-all Support Vector Machines [122] with the fast histogram intersection kernel [74] for its high efficiency. The SVM models are optimized by 3-fold cross validation.

We visualize the three learning strategies in Figure 2.4.

2.4 Experiments

In order to arrive at best practices for the four key components described in Section 2.3, we conduct the following four experiments.

2.4.1 Experiment 1: What source?

In this experiment we address the research question *what visual tagging source is most suited for selecting positive training examples to learn video concepts?*. We use the *Tagged Videos* and the *Tagged Images* datasets described in Section 2.3.1 as instantiations of two distinct visual tagging sources. We learn concept detectors using the ALL-FRAME [21] and the ALL-IMAGE [15] scenarios separately, and compare their performance.

2.4.2 Experiment 2: What positive examples?

Besides the ALL-FRAME and the ALL-IMAGE baselines, we evaluate the three positive example selection strategies described in Section 2.3.2 to answer the question *what strategy should be used for selecting positive examples from tagged sources when learning video concept detectors?*. We employ the strategies for both the *Tagged Videos* and the *Tagged Images* datasets.

1. Random. We randomly sample positive frames from the socially tagged sources for each concept. The set of positive examples is formalized in Equation (2.1). We vary the number k of sampled positive frames to investigate their influence.

2. Relevant ad-hoc. We rank the candidate positive examples by their tag relevance scores in descending order. For each concept, an ad-hoc fractions of the top ranked examples are selected as positive training examples. The set of positive frames is formalized in Equation (2.2). We vary the fraction $k = 0.1, 0.2, \dots, 0.9$ to investigate the importance of making a proper cut-off.

3. Relevant cut-off We rely on the same ranking of the examples as in strategy 2, but here we automate the selection with the simple Bayes approach given in Section 2.3.2. The set of positive frames is defined in Equation (2.8). Due to the large variance of the tag relevance scores for tagged images, we smooth the scores using the common logarithm.

Since we evaluate the positive selection strategies in this experiment, we choose the most simple and commonly used strategy for negatives, and keep it fixed. The same set of random sampled negatives, described in Section 2.3.3, were used for all strategies and the All-FRAME and ALL-IMAGE baselines, sampled from the *Tagged Images* and *Tagged Videos* sources respectively. Following the common procedure in the literature [68, 132] we set the size of the negative set to be the same as the number of positive examples. We refer to Table 2.1 for the number of positive examples per concept.

2.4.3 Experiment 3: What negative examples?

To answer *what strategy is most suited for selecting negative examples from social media?* we investigate the following three strategies for negative examples selection, detailed in Section 2.3.3. We employ the strategies on both the *Tagged Videos* and the *Tagged Images* datasets.

1. Random. We randomly sample negative frames from the *Tagged Videos* for each concept. We vary the number of sampled negative frames to investigate their influence. The set of negative frames is formalized in Equation (2.9).

2.Pseudo Negative. We calculate a tag relevance score for each keyframe before ranking. A fraction of the *bottom* ranked frames per concept are selected as the negative training examples. The set of negative frames is formalized in Equation (2.10).

3.Negative Bootstrap. We employ an iterative procedure for selecting relevant negatives [68]. The set of negative frames is formalized in Equation (2.11).

The same set of positives was used for all negative strategies, sampled from the corresponding tagged sources. The automatic cut-off strategy described in Section 2.3.2 was used. Thus, the relevant positive examples are selected once in a probabilistic manner, instead of setting an add-hoc threshold.

2.4.4 Experiment 4: What learning strategy?

We compare the performance of three learning strategies described in Section 2.3.4 to answer the question *what learning strategy should be used for learning video concept detectors from socially tagged data?*. We investigate whether *Refined examples* learning of selected relevant positive and negative training examples is better than methods that avoid prior selection of relevant examples. In *Packed examples* learning videos tagged with the concept name are considered as positive bags, and all its frames as positive instances. In *Paired examples* learning, like RankSVM, ordered pairs of examples are generated prior to learning. In both these methods the selection of relevant training data is ignored. In this experiment we compare the performance of *Paired examples* and *Packed examples* with *Refined examples* learning.

2.4.5 Test Set

As test data we adopt the challenging internet video collection from the TRECVID 2012 benchmark [106]. We use the development dataset provided for the Semantic Indexing task, which consists of 600 hours of Internet Archive videos, having 400,682 shots. Each shot is represented with a single keyframe. The dataset comes with ground truth annotations on a keyframe level, including the 20 concepts identified in our experiments. We evaluate and report all experiments on this test set.

2.4.6 Implementation details

Concept detection features. We employ a standard bag of visual codes pipeline to train video concept detectors [120]. We compute SIFT descriptors [73] at dense sampled

points, at every 6 pixels for two scales. As visual features we employ a spatial pyramid of 1x1 and 1x3. The codebook size is 4,096, constructed with k -means clustering.

Tag relevance features. We follow the implementation from [67] and use three visual features: Color64, CSLBP and GIST. Color64 is a 64-d global feature, combining the 44-d color correlogram, the 14-d texture moments, and the 6-d RGB color moments [65]. CSLBP is a 80-d center-symmetric local binary pattern histogram [36], capturing local texture distributions. GIST is a 960-d feature describing dominant spatial structures of a scene by a set of perceptual measures such as naturalness, openness, and roughness [83].

Evaluation criteria. We adopt average precision to report individual concept detection accuracy, a common approach in the video retrieval literature [106]. We report mean average precision to evaluate the overall detection performance for all concepts.

2.5 Results

2.5.1 Experiment 1: What source?

We summarize the results of Experiment 1 in Table 2.2. We observe that better concept detectors are obtained from tagged images than from the tagged videos. When simply using all available tagged images for training, we obtain an mAP of 0.375, while the video alternative obtains an mAP of 0.214. In terms of an absolute difference tagged videos have a significant 16.1% lower mAP over tagged images. We explain the difference by the observation that the user tagging accuracy of images is more accurate when compared to videos. Moreover, when a tag is assigned to a video, often only a small fraction of the video content is relevant with respect to the tag. Consequently, given the same number of positive training examples, the positive set selected with our relevant selection from the image source contains more genuine positives than the positive set from the video source, as exemplified in Figure 2. As our first best practice, we conclude that tagged images are a better source than tagged videos for learning video concept detectors.

2.5.2 Experiment 2: What positive examples?

Figure 2.6 shows the performance curve of concept detectors derived after positive example selection by the three strategies, for tagged videos and tagged images. In line with the results of Experiment 1, using images as training source is better than video for all strategies. Relevance selection is a better strategy than randomly selecting frames

Tab. 2.2: Experiment 1: What Source? As first research question we investigate what visual tagging source, tagged images or tagged videos, is most suited for selecting training examples for learning video concept detectors. It leads to our first best practice: To learn video concept detectors from socially tagged media, tagged images should be used.

Concept	Tagged Videos	Tagged Images
Animal	0.078	0.122
Beach	0.158	0.359
Building	0.334	0.500
Car	0.157	0.230
Child	0.069	0.118
City	0.064	0.131
Face	0.496	0.606
Hand	0.114	0.175
Landscape	0.207	0.567
Mountain	0.048	0.516
Oceans	0.129	0.481
Outdoor	0.816	0.722
Plant	0.188	0.270
Road	0.186	0.427
Sky	0.258	0.621
Snow	0.063	0.273
Sports	0.129	0.149
Streets	0.099	0.183
Trees	0.470	0.693
Vehicle	0.221	0.351
mAP	0.214	0.375

or images for all cut-offs in both sources. For the best possible relevant image selection (fraction = 0.6), we gain a 5.4% relative improvement over the same amount of randomly selected images. For the video alternative the difference is even more convincing. In that case we obtain the best possible result for a fraction of 0.1, with a 58.9% relative improvement over random selection. There is a clear decrease in performance as we select larger fractions of frames, see 2. *Relevant ad-hoc Frames* in Figure 2.6. Since larger fractions contain more noisy data, *i.e.* frames that also have low tag relevance score, the classifier learns less accurate models. In the case of tagged images there is less noise compared to tagged videos, since the tag is directly appointed to the visual content of the image. Consequently, for images larger fractions show better performance, see 2. *Relevant ad-hoc Images* in Figure 2.6. In this case, selection of smaller fractions also ignores relevant images, which results in less accurate classifiers.

Tab. 2.3: Experiment 2: What positive examples? With the second research question we investigate what strategy should be used for selecting positive examples from social media. We show results per concepts for three strategies, 1.Random, 2. Relevant ad-hoc selection with the best and least performance, and 3 Relevant cut-off selection. It leads to our second best practice: for learning video concept detectors, as positive examples use relevant cut-off selection of tagged images.

Concept	Strategies using tagged videos				Strategies using tagged images			
	1. Random	2. Add-hoc least	2. Add-hoc best	3. Cut-off	1. Random	2. Add-hoc least	2. Add-hoc best	3. Cut-off
Animal	0.060	0.083	0.106	0.102	0.116	0.122	0.140	0.140
Beach	0.208	0.146	0.256	0.245	0.324	0.306	0.340	0.324
Building	0.254	0.347	0.406	0.401	0.486	0.495	0.544	0.541
Car	0.128	0.168	0.249	0.237	0.234	0.322	0.293	0.308
Child	0.062	0.075	0.107	0.093	0.117	0.140	0.130	0.131
City	0.046	0.071	0.153	0.106	0.147	0.277	0.195	0.196
Face	0.400	0.503	0.562	0.536	0.594	0.594	0.619	0.602
Hand	0.125	0.112	0.116	0.117	0.177	0.152	0.163	0.159
Landscape	0.208	0.222	0.410	0.341	0.561	0.554	0.580	0.574
Mountain	0.043	0.057	0.255	0.214	0.490	0.468	0.515	0.506
Oceans	0.100	0.220	0.410	0.398	0.479	0.474	0.516	0.483
Outdoor	0.723	0.821	0.825	0.831	0.719	0.769	0.753	0.754
Plant	0.171	0.184	0.286	0.283	0.254	0.230	0.259	0.255
Road	0.242	0.235	0.327	0.358	0.408	0.372	0.418	0.416
Sky	0.258	0.277	0.371	0.391	0.623	0.412	0.601	0.629
Snow	0.050	0.077	0.138	0.127	0.262	0.198	0.259	0.242
Sports	0.122	0.137	0.204	0.184	0.145	0.258	0.215	0.225
Streets	0.111	0.102	0.169	0.124	0.192	0.191	0.184	0.184
Trees	0.458	0.490	0.580	0.578	0.688	0.609	0.660	0.648
Vehicle	0.183	0.227	0.339	0.329	0.348	0.381	0.381	0.386
mAP	0.197	0.228	0.313	0.300	0.368	0.366	0.388	0.385

Fig. 2.5: Relevant positive training examples selected from social media. Left columns show the selected images and frames, while the right columns show images and videos labeled with a given concept but discarded by the relevant positive selection strategy.

Apart from the gain in performance, it is also important to note that by selecting relevant subsets from the collection we reduce the number of training examples, which speeds up concept detector training.

The simple cut-off approach for automatic selection of positive examples, approximates the best selection quite closely (see the solid and dashed lines in Figure 2.6). In case of image selection it outperforms the best possible ad-hoc relevant image selection for 9 concepts even, and the worst performing ad-hoc selection in 14 concepts (see Table 2.3). We conclude that selecting relevant positive images and video frames is needed when learning concepts from the web. The selection cut off can be estimated automatically. We employed a simple Bayes approach, although more extensive sampling methods can be incorporated in the future. For now, as second best practice we suggest to rely on the relevant cut-off selection strategy described in Section 2.3.2 for selecting relevant examples from socially tagged media.

2.5.3 Experiment 3: What negative examples?

In Figure 2.7 we show the performance for the three negative example selection strategies for both tagged sources, images and videos. When using tagged images, our results

Fig. 2.6: Experiment 2: What positive examples? Training concept detectors after tag relevance selection, with either video frames or images is always beneficial. The simple cut-off selection is a good approximation of the best possible selected fraction for both video frames and images.

confirm the finding of [68]. Compared to random sampling the negative bootstrap wins in 14 out of 20 concepts, with a 6% relative improvement in mAP over random sampling. The pseudo negative strategy shows lower performance compared to random sampling, with 12.6% relative decrease. By examining the selected negatives we find that the pseudo negative tends to select some positive examples as negatives, as shown in Figure 2.8. Obviously, such false negative examples hurt the learning process.

In the case of tagged videos the situation is different. Random sampling is better than both negative bootstrap and pseudo negative. Since videos can have very diverse visual content, even if some video is not tagged with a concept name, it may contain some frames relevant to the concept. The negative bootstrap and the pseudo negative tend to select positive frames as negatives, see Figure 2.8, which in terms of mAP result in less accurate performance over random sampling. However, the negative bootstrap wins over random sampling for 4 concepts, and the pseudo negative for 9 out of 20 concepts. Therefore, selecting the most appropriate strategy for the concepts separately is interesting to be investigated for tagged videos.

Since selecting relevant negative examples from tagged images shows much better performance, our third best practice is to use negative bootstrap of tagged images to select negative training examples for learning video concept detectors.

Tab. 2.4: Experiment 3: What negative examples? As third research question we investigate what strategy should be used for selecting negative examples from tagged sources when learning video concept detectors. It leads to our third best practice: bootstrapping relevant negatives from tagged images.

Concept	Strategies using tagged videos			Strategies using tagged images		
	1. Random	2. Pseudo Negative	3. Negative bootstrap	1. Random	2. Pseudo Negative	3. Negative bootstrap
Animal	0.101	0.095	0.070	0.140	0.099	0.157
Beach	0.225	0.226	0.103	0.324	0.260	0.361
Building	0.383	0.308	0.294	0.541	0.447	0.575
Car	0.221	0.191	0.294	0.308	0.291	0.384
Child	0.096	0.111	0.068	0.131	0.121	0.126
City	0.100	0.131	0.128	0.196	0.216	0.238
Face	0.548	0.528	0.422	0.602	0.531	0.672
Hand	0.120	0.110	0.133	0.159	0.106	0.141
Landscape	0.314	0.331	0.122	0.574	0.537	0.611
Mountain	0.196	0.220	0.037	0.506	0.388	0.539
Oceans	0.358	0.390	0.037	0.483	0.448	0.541
Outdoor	0.815	0.801	0.562	0.754	0.817	0.769
Plant	0.274	0.309	0.161	0.255	0.238	0.265
Road	0.336	0.302	0.190	0.416	0.301	0.474
Sky	0.351	0.368	0.176	0.629	0.587	0.619
Snow	0.109	0.105	0.087	0.242	0.139	0.229
Sports	0.170	0.146	0.153	0.225	0.232	0.171
Streets	0.113	0.125	0.129	0.184	0.186	0.214
Trees	0.576	0.570	0.436	0.648	0.582	0.646
Vehicle	0.339	0.282	0.202	0.386	0.380	0.431
mAP	0.287	0.282	0.190	0.385	0.345	0.408

Fig. 2.7: Experiment 3: What Negative Examples? Selecting relevant negative examples from tagged images with the negative bootstrap improves over random sampling. The result converges after only five iterations. For tagged videos random sampling is the best choice because it suffers less from selecting false negatives than the pseudo negative and the negative bootstrap.

2.5.4 Experiment 4: What learning strategy?

In Table 2.5 we compare the three learning strategies described in Section 2.3.4 together with the ALL-FRAME [118] and ALL-IMAGE [102] baselines. When we follow the ALL-FRAME baseline [118], which simply uses all frames of tagged videos, rather than selecting relevant positive or negative frames, we obtain better results than the packed frames learning [61], with 17.6% relative improvement in terms of mAP. The refined examples strategy for tagged videos also shows lower performance than the baseline, mostly because of the selection of false negative examples, as explained in more detail in Section 2.5.3. The paired examples learning has best performance for tagged videos,

Tab. 2.5: Experiment 4: What learning strategies? As forth research question we investigate what learning strategy is most suited for learning video concept detectors from social media examples. We show results for the three learning strategies and baselines for tagged images and tagged videos. We recommend a forth best practice: To use SVM with refined examples learning from tagged images, by selecting relevant negative and positive training examples.

Concept	Strategies using tagged videos				Strategies using tagged images		
	ALL-FRAME	Packed Examples	Paired Examples	Refined Examples	ALL-IMAGE	Paired Examples	Refined Examples
Animal	0.087	0.061	0.078	0.070	0.122	0.108	0.157
Beach	0.243	0.055	0.158	0.103	0.359	0.326	0.361
Building	0.323	0.154	0.334	0.294	0.500	0.496	0.575
Car	0.165	0.080	0.157	0.294	0.230	0.218	0.384
Child	0.083	0.080	0.069	0.068	0.118	0.111	0.126
City	0.091	0.048	0.064	0.128	0.131	0.160	0.238
Face	0.491	0.313	0.496	0.422	0.606	0.564	0.672
Hand	0.118	0.104	0.114	0.133	0.175	0.152	0.141
Landscape	0.276	0.099	0.207	0.122	0.567	0.520	0.611
Mountain	0.104	0.033	0.048	0.037	0.516	0.478	0.539
Oceans	0.288	0.092	0.129	0.037	0.481	0.463	0.541
Outdoor	0.815	0.743	0.816	0.562	0.722	0.750	0.769
Plant	0.244	0.214	0.188	0.161	0.270	0.249	0.265
Road	0.282	0.215	0.186	0.190	0.427	0.379	0.474
Sky	0.349	0.206	0.258	0.176	0.621	0.626	0.619
Snow	0.102	0.038	0.063	0.087	0.273	0.219	0.229
Sports	0.150	0.109	0.129	0.153	0.149	0.187	0.171
Streets	0.127	0.137	0.099	0.129	0.183	0.180	0.214
Trees	0.544	0.459	0.470	0.436	0.693	0.646	0.646
Vehicle	0.296	0.411	0.221	0.202	0.351	0.311	0.431
mAP	0.214	0.182	0.259	0.190	0.375	0.357	0.408

Fig. 2.8: Training examples selected from tagged images and tagged videos. Left column shows selected positive examples, the middle column shows selected examples with the pseudo negative, and the right column shows negative examples selected with the negative bootstrap. For the concept beach with the negative bootstrap we see selected images of snow or a landscape, *i.e.* images which are most likely to be mistaken by the concept detector. Relevant frames are also selected from videos, but also some false negatives. The pseudo negative method tends to select false negative examples of beach for both images and videos. For the sky concept, since many outdoor images or videos contain sky, but are not explicitly tagged with sky, both pseudo negative and negative bootstrap have a tendency to select false negatives.

showing that using relative information of frames by creating ordered pairs is better than preselecting relevant positive and negative examples. When relying on images as training source for concept detectors we observe different behavior. Learning with refined examples reaches an mAP of 0.408, clearly improving over all other learning strategies, and with best result for 13 out of 20 concepts, see Table 2.5. We conclude with the best practice that training concept detectors from a tagged image source using refined examples, *i.e.* relevant positive and negative images, is the best learning strategy.

Fig. 2.9: Results per concept from the TRECVID 2013 Semantic Indexing with No Annotations task. The recommended best practices of this work within our video search engine achieved best result for 28 out of 38 concepts, compared to the other submitted runs in the same task.

2.5.5 Benchmark Comparison

By taking all best practices into account, we participated in the NIST TRECVID 2013 [106] benchmark, using our video search engine. We considered the no annotation version of the task, and summarize results for all evaluated concepts in Figure 2.9. Our detectors obtain the best performance for 28 out of 38 concepts.

2.6 Conclusion

This chapter strives to answer *how to learn effective video concept detectors from social media examples* through a systematic empirical study. Supported by experiments on a present day testbed, our major findings are: 1) Tagged images are the preferred

choice as training source, when compared to tagged videos. Concept detectors trained on tagged images surpass their counterparts trained on tagged videos, with an absolute improvement of approximately 16% in terms of mAP. While images may not be one's first option due to source change, we find that the better annotation quality let them beat videos with ease. 2) For both tagged videos and tagged images, selecting positive examples from those with the largest tag relevance scores is superior to getting positives at random. We show that the cut-off of the top ranked relevant examples can be simply estimated, obtaining a near-optimal result. 3) The selection of negative training examples is also important. With negative bootstrap of tagged images we gain 6% relative improvement over simple random selection. For videos random sampling is the best choice. 4) We show that preselecting relevant positive and negative training examples beforehand and training support vector machine is more beneficial than strategies where the relevance of the tagged training data is considered during learning. Below we summarize our best practices:

Best Practices:

1. *Tagged images are a better source than tagged videos for learning video concept detectors.*
2. *Positive examples with relevant cut-off of tagged images show best performance.*
3. *Relevant negatives are best selected with negative bootstrap of tagged images.*
4. *For learning we recommend SVM with relevant positive and negative examples.*

When the best practices of this chapter are combined into a single video search engine it results in state of the art video concept detection without the need of any predefined manual annotations.

Exploring the Long Tail of Social Media Tags

3.1 Introduction

In this chapter we focus on the long tail frequency distribution of social tags. It is well known that tag frequencies in social media form a long tail distribution [103, 129]. While some tags are frequent, like *snow*, *beach*, *coffee*, there is a large number of tags which are rare, with only few example images per tag, like *mierkat*, *tank suit*, *dyippy*, see Fig. 3.1. Current works note that many tags from the long tail are “misspelled” or “meaningless” words [129], or only useful for “exceptional cases” [103]. It seems this observation has been accepted in the community as such, and no further investigation has been performed so far. We believe that there are also meaningful tags within the long tail which have been overlooked. Since the tags from the long tail make up a significant portion of the data, they deserve more detailed analysis. On that account, we pose the question: *What tags constitute the long tail?*¹.

We believe that the long tail distribution should not be just accepted as such, but looked at as a challenge. The challenge is to augment the frequencies of rare tags, so that the long tail distribution will change its shape. By augmenting the rare tags, they become a richer source for many multimedia algorithms. Motivated from common approaches in the literature of enriching tags with semantic knowledge [27, 142], we investigate the effect of augmenting the examples of rare tags with semantically related tagged images. We question *What happens when rare tags are augmented?*.

¹Published in *the Proceedings of International Conference on Multimedia Modelling*, 2016 [54].

Fig. 3.1: The log-frequency of Flickr images for 730K tags. Some tags like *beach*, *snow*, *coffee* have millions of example images. We investigate the majority of rare tags such as *miercat*, *mountain tent*, and *tanksuit*.

Social tags come for free and are a valuable resource for many multimedia methods that aim to automatically understand visual content. Examples include automatic concept detection [52, 61, 116], user profiling [30, 80], sentiment analysis [12, 126, 134], and assessing tag relevance [66, 114]. These works give excellent results when ample training examples are available per tag. It is not clear, however, how these methods will perform for rare tags with fewer examples. Since there are many tags which fall on the long tail, we argue that it is of great importance for the tags from the long tail to also be successful on these methods. Therefore, we question *What is the effect of rare tags on multimedia retrieval scenarios?*, and evaluate the effect when rare tags are augmented with semantics.

The contributions of this chapter are three fold. First, we analyze the type of tags that occur in the long tail distribution of social tags. We base the analysis on a representative snapshot of Flickr containing 1 million photos with 700K diverse tags, and additional 38K tags from three categories *objects*, *scenes* and *fine-grain animals*. Second, we investigate augmenting the rare tags with semantic knowledge. Third, we exploit two multimedia retrieval scenarios on sampled tags from the long tail distribution, and analyze their performance on both rare and augmented tags.

3.2 Related Work

Many works in social media are focused on frequent tags with many example images per tag. This is quite understandable, since those tags are popular and evidently important for many users. In Figure 3.2 we show the trend of social media works over the frequency distribution, which confirms the tendency towards frequent tags. In this chapter we analyze the non frequent tags.

Fig. 3.2: The trend of related works over tag frequency distribution, confirming the emphasis in the literature towards frequent tags.

Frequent tags. One widely recognized problem is that tags are noisy, ambiguous and often not directly related to the visual content [66, 142]. For that reason, Li *et al.* [66] defined a *tag relevance* metric, calculated by counting neighborhood votes from visually similar images, with the intuition that visually similar images should share the same tags. Their method has shown impressive performance when evaluated on frequent tags. For rare tags we believe that this method could be problematic. For example if a tag is relevant to an image but occurs very rarely or not at all in the corpus, it will not get enough votes from its neighbors and will be falsely considered as irrelevant. Zhu *et al.* [142] use the related tags to handle imprecise and incomplete image tags. They calculate the relevance of an image tag by collectively analyzing statistics from Flickr and the WordNet hierarchy, so that a tag receives examples from its child nodes. It might be troublesome if a rare tag is a leaf node in the WordNet hierarchy, without child nodes to be enriched with. Having this in mind, we investigate how calculating tag relevance on rare tags performs, and if including semantic knowledge would help improving the relevance of a tag.

Many other diverse topics exist when using frequent tags. The authors of [12] have created a visual sentiment ontology by sending queries of adjective noun pairs (ANP) to Flickr. In their work ANP candidates (about 320k) are ranked by occurrence frequency, with the goal to remove extremely rare constructions and to keep only frequent ones. While the frequent ANPs show valuable performance when learning detectors, it is unclear what will be the performance of the rare ANPs. The tag ranking method of [71] automatically ranks tags associated with an image using the neighborhood of images containing a specific tag. Finding tagged neighbors of rare tags is problematic, which might lead to assigning low scores for relevant tags. In other works, like tag ambiguity [105], learning detectors to tag [129] or tag recommendation [103], the authors use co-occurrence statistics. If a tag is rare, its co-occurrence statistics will be unreliable, which might result in erroneous scores. In [52, 61, 116] social tags are used to collect

training data and learn concept detectors. For rare tags there are less images to be used as training data. Therefore, we believe it is interesting to also investigate the learning performance of rare tags. We choose to evaluate detectors learning, since it is a popular topic, and the performance can be evaluated on a standard benchmark dataset [19].

Rare tags. To the best of our knowledge there is no related work investigating rare tags in social media. The problem of rare tags is somewhat similar to few- or zero-shot learning [59, 109], designed for problems where there is a lack of training samples. Most of these works use textual or attribute descriptions of known concepts and compare with the textual description or attribute scores of the zero-shot concept. These textual or attribute detectors are learned with manual annotations, which limits the type of concepts that can be detected. The manual annotations are also quite precise compared to the noisy tags from social media. Therefore, applying these techniques to multimedia methods is difficult, and also requires modifications in the existing algorithms. We investigate a more generic approach that takes the noise of social tags into account, and still deals with the low number of examples.

Closest to our work is a recent study on objects [144], where it is shown that object categories follow a long tail distribution. Most of the object subcategories are rare, which makes it difficult for learning detectors. The authors address the lack of training data by allowing the rare subcategories to share training examples with dominant ones in their learning model. This is possible since the semantic relationships of object categories-subcategories are known, and the images have precise manual annotations. However, social media does not have known semantic relationships between tags. Tagged images are easy to obtain, but they come with noise as additional complication. The method of [144] does not address these challenges, thus, it can not be directly applied on multimedia applications that use social media tags.

Semantic relationships. Relationships between tags have been calculated using co-occurrence statistics [103, 105, 129], or semantic knowledge from external sources [27, 103, 142]. Co-occurrence statistics can be unreliable for calculating tag relationships of rare tags, since they occur rarely or not at all. In the work of Fergus *et al.* [27] an ontology from an external source is adopted to expand query sets for label propagation. In [142] Zhu *et al.* use an ontology to expand the training data. For example, a training set of a non-leaf concept (e.g., building) is enriched by including representative examples from its child nodes (e.g., church). In this chapter we investigate external semantic knowledge for a different purpose, to augment the frequency of rare tags.

Although there are many works investigating the possibilities of socially tagged images, none has so far looked into the tags from the long tail of the frequency distribution. In this chapter we do a first attempt to analyze rare tags.

Fig. 3.3: Occurrence of n-grams, following the long tail frequency distribution.

3.3 What tags constitute the long tail?

Tag vocabulary. We analyze tags from Flickr as one of the most popular social media platform. We consider the *Flickr 1M* dataset from [66] as a representative sample. This dataset has 1 million images, downloaded by randomly generating photo ids as queries, making it unbiased towards any tags. The images come with about 700K diverse tags. Since its impractical to analyze all these tags manually, we consider tags from three categories of existing image recognition datasets: *objects*, *scenes* and *fine-grained animals*. Many object tags are available in ImageNet [19], 22K classes resulting in 40K tags, since some classes come with multiple tags, like *sea cow*, *sirenian mammal*, *sirenian* represent one object class. The SUN Attribute dataset [87] has the highest number of scene tags so far, 717 classes like *amusement park*, *coast*, *squash court*. Tags from fine-grained animal categories are available in the 120 dog tag from Stanford Dogs [48] like *pekinese*, *irish terrier*, *chihuahua*, and 200 bird tags from Caltech Birds [124] like *shiny cowbird*, *bobolink*, *blue jay*. All tags together form a vocabulary of 730K unique tags. We call this tag set the *Tag vocabulary*, and we analyze it in this experiment.

Tag composition. It is interesting to investigate the occurrence of n-grams, following the long tail frequency distribution. We aim to find if there is a correlation between the number of words a tag is composed of, and its occurrence frequency. It has been noted that tags from the long tail are mostly complex phrases [103], we investigate if this is indeed the case. We merge the tags of objects, scenes and fine-grained animal categories, in total 38K. We send queries with the tag names to Flickr with the Flickr api, and count the occurrence of each tag. We order the tags by their frequency and we count the unigrams, bigrams, trigrams and four-grams in 40 steps. We visualize the histograms in Figure 3.3. It can be observed that the frequent tags are mostly unigrams. As the frequency goes down, more and more bigrams appear, and towards the end trigrams and four-grams start occurring. Some frequent bigrams are *christmas tree* with frequency 250K, or *polar bear* with frequency 160K. The bigrams *zebra wood tree* or *kaffir cat* have frequency zero. The histograms shows that most of the rare tags are

bigrams, and less uni-/tri-/four-grams. We believe this is so because users often use general one word tags to describe an image, and do not try to be precise. We also manually analyzed sampled Flickr tags from the Tag vocabulary, not necessarily only coming from the preselected categories. We looked into 50 tags around 10 steps over the long tail distribution. Interestingly, among the frequent tags we found tags like *iphoneography* and *instagramapp*, which are added from popular mobile applications. We expect that with thousands of images being uploaded in Flickr daily, the frequency of the tags changes, new tags appear, and old ones become more frequent. However, at any given time, when a snapshot of Flickr is taken, like at 2008 [103], at 2009 [129] and ours, the distribution stays heavy tailed. Among the sampled tags from the long tail there were attribute and noun pairs like *scratching post*, *showing work*, named entities like *saint petersburg russia*, *arnold aragon*, phrases like *what color is your time*, *i enjoyed all types of outdoor adventures as a child*, number and word compositions like *photo domino 357*, *hp850* which represent models of products, and also tags in which we could not find meaning like *wo0*, *ell:mcc=222*. Moreover, we summarize, the long tail contains attribute noun pairs, entities of less famous people or geographic places, as well as phrases. The bigrams occur most often in the long tail, and there are less unigrams, trigrams and fourgrams.

Tag categories. We visualize in heat bars the tag occurrence of object, scenes, and fine-grained animal categories in Figure 3.4. Many **object** tags are colored in blue and appear rarely in Flickr. In numbers, 67% of the object tags appear in less than 1K images. The rare objects are mostly specific tags like *lingberry* which so far occurs in only 3 Flickr images, *marsh tea* in 28 images or *space vehicle* in 67 images. From the **scene** category, 46% of the tags occur in less than 1K images. The strong blue response in the heat bar shows that a great portion of scene tags have low frequency. These rare tags mostly represent fine-grained scenes like *artists loft*, *canal urban*, *bakery kitchen*. From the **fine-grained** animal categories, green is the dominant color in the heat bar, and red and blue take small proportions at the ends of the bar. This shows that images of fine grained animal categories occur reasonably often in Flickr. In numbers, 41% of the fine-grained animal tags appear in 1K-10K images, and 36% appear in less than 1K images. This hints that fine-grained classification can be made even more challenging than the classes suggested in the existing fine-grained datasets. Within the fine-grained dogs and birds classes, some unpopular ones are *brabancon griffon* with frequency 0, *brewer blackbird* with frequency 7 and *blenheim spaniel* with frequency 36. Overall, our analysis goes against the prevailing norm in the literature where tags from the long tail have been considered unimportant. We show that there are meaningful tags of objects, scenes and fine-grained categories that occur rarely in Flickr and should not be overlooked.

Fig. 3.4: Heat bars of tag occurrence frequency in Flickr for objects, scenes and fine-grained animal tags. We observe there are many object and scene tags that appear rarely in Flickr, whereas most of the fine-grained animal tags occur reasonably often.

3.4 Utilizing the long tail

3.4.1 Augmenting rare tags

We investigate augmenting rare tags from the long tail with semantically similar tags. We believe that augmenting the rare tags is important for good performance of multimedia retrieval methods. Therefore we analyze the performance of both rare and augmented tags on two multimedia retrieval scenarios in sections 4.2 and 4.3. Motivated from common approaches in the literature of using semantics from external sources like Wikipedia or WordNet [27, 142], we investigate if rare tags can be augmented with semantics. From WordNet we consider synonyms and child nodes of a tag in the hierarchy. From Wikipedia we consider titles of redirect pages, as commonly used for semantic linking in information retrieval [76].

Datasets. Since it is not possible to evaluate all 730K tags from the Tag vocabulary, we obtain a representative sample by uniformly sampling each 2,000th tag from the distribution shown in Figure 3.1. We make sure that the sampled tags have ground truth annotations in some dataset, so that we can evaluate their performance on tag relevance and learning detectors. We consider ground truth classes from ImageNet, as one of the largest available image dataset, with 50 validation images per tag for 1000 classes. If the 2000th tag does not appear in one of the 1000 ImageNet classes, we move to the 2001th tag, and so on. In this manner we select 81 representative tags. These 81 tags contain frequent tags like *light*, *marmot*, *blue jean* and rare tags like *whiskey jug*, *bottle screw*, *rock snake*. For each tag we download up to 2,000 images from Flickr if available, otherwise as much as we can. This resulted in a new *LongTail* dataset with 13K Flickr

Fig. 3.5: Example images from Flickr and ImageNet. Tags 3:blowfish and 7:washbowl have visually very diverse appearance. Tags 2:marmot and 10:mileometer have more consistent visual features, thus show better performance.

images. Additionally, we also downloaded images tagged with semantically similar tags found from WordNet and Wikipedia, forming a new *LongTailAugmented* dataset with 160K images.

Analysis. In Figure 3.6 (a) we show the frequency of the sampled 81 tags, as well as their frequency when augmented with images of semantically related tags. For most of the rare tags we could find images tagged with their synonym tags, magnifying the frequency when joined. Some rare tags have synonyms from WordNet which are quite frequent, like for example tag 18:patrol wagon (id:tag, where id is the position of the tag on the horizontal axis in Figure 3.6) appears in only 41 images, whereas its WordNet synonyms appear in much more: police van in 2310, paddy wagon in 3736, wagon in 200K images. For tag 56:Chlamydosaurus kingi which so far appears in one image, its WordNet synonym frilled lizard has 250 images, and Wikipedia finds more synonyms, Chlamydosaurus in 278, frilled dragon in 150 and frillnecked in 150 images. For some tags like 17:english foxhound and 67:mountain tent, we did not find semantically similar tags, and for tag 39:plumbers helper its synonym plunger has 0 tagged images. We expect that with more sophisticated language processing more semantically related tags can be found, and the rare tags will be even better augmented. Overall we conclude, that rare tags from the long tail can be augmented with simple synonyms.

Fig. 3.6: Frequency and AP of 81 tags, without and with synonym augmentation.

3.4.2 Tag Relevance

In this experiment we investigate the performance of calculating tag relevance for tags which fall on the long tail of the frequency distribution. We investigate the performance of the most popular tag relevance method proposed by Li *et al.*[66], due to its good performance [114].

Augmented Tag Relevance. We simply extend the tag-relevance method of [66] to take semantically related tags into account. Instead of just counting votes of the same tag, we also count votes from its synonyms. The augmented tag relevance of a rare tag is computed as follows.

We denote a rare tag with r and $S_r = \{s_1, s_2, \dots, s_n\}$ is its set of n synonyms. For an image I_r with tags $\{t | t \in \text{tags}(I)\}$, the set of images tagged with semantically similar tags within its k visual nearest neighbors is

$$N(I_r, S_r, k) = \{I | I \in \text{NN}(I_r, k) \wedge \exists t(t \in \text{tags}(I) \wedge t \in S_r)\}. \quad (3.1)$$

We calculate the rare tag relevance R for an image I_r as

$$R(r, I_r, k) = |N(I_r, S_r, k)| - P(S_r, k), \quad (3.2)$$

where $P(S_r, k)$ is the prior tag distribution, which in our case denotes the average prior of the synonym set

$$P(S_r, k) = \frac{1}{|S_r|} \sum_{s \in S_r} k \frac{|L_s|}{|L|}, \quad (3.3)$$

where k is the number of visual neighbors, $|L_s|$ the number of all images labeled with s , and $|L|$ the size of the entire collection. The difference in this formulation from [66], is adding the set of synonyms S_r , on places where only one tag t was used in the voting.

The tag relevance method is developed for tags which are composed of only one word. If a tag is an n-gram, composed of two or more words, we follow the recommendation from [69], and average the tag relevances or the augmented tag relevances for each word.

Features. As visual features for tag relevance and augmented tag relevance we use the same settings as the multi-feature color and texture variant of [69].

Results and analysis. To evaluate, we use the ImageNet validation images of the same 81 classes as the selected tags in the LongTail dataset. For each tag, there are 50 positive images in the ImageNet validation set. To evaluate tag relevance we also need images with noisy tags. Thus, for each tag we additionally sample 100 random images from the other classes of ImageNet, resulting in 150 images for evaluation of each tag. We evaluate with average precision (AP) per tag and mean average precision (MAP) for overall score.

We plot the AP per tag in Figure 3.6 (b), for both regular and augmented tag relevance. A negative effect of the long tail can be clearly seen for tag relevance, where most of the the rare tags obtain low average precision. One reason for this is that when there are few or even zero images in Flickr which have the tag, it is unlikely for the visual neighbors to contain the tag resulting in no votes and failing to learn the relevance of the tag. We also notice some outliers, like for 3:*blowfish*, 4:*bluejean* and id 7:*washbowl* which have lower tag relevance performance compared to their neighbors in the frequency distribution. In Figure 3.5 we show how visually diverse and ambiguous these concepts are, compared to for example 2:*marmot* and 5:*easternfox squirrel*, and 10:*mileometer*.

The augmented tag relevance has a positive effect on the rare tags, since for most of the rare tags the average precision improves. For example, the AP of 73:*lycaenid butterfly*, grows from 37% to 64%, and for 18:*patrol wagon* improves from 49% to 86%. For

some tags, there is none or a small improvement. For example 17:*english foxhound* and 67:*mountain tent* have no improvement since no semantically related tags were found. In some cases although the synonym tags are frequent, the results do not improve. For example tag 24:*woodworking plane* has a synonym *plane* which is not as specific and contains many diverse and visually different images, see Figure 3.7. In cases like this, we expect a more sophisticated semantic method for augmenting the rare tags would help.

Overall, when only few tagged images are present, the tag relevance is determined to fail upfront. When we use augmented tag relevance with synonyms, the mean average precision grows from 43% to 73%. We conclude the tag relevance of the rare tags can be better calculated with augmented semantics.

3.4.3 Learning Detectors

We investigate learning concept detectors from tagged images, and analyze their performance in correlation with the tag frequency occupance. As a training set we use the LongTail and LongTailAugmented datasets. From the LongTail dataset we select the top ranked images based on their tag relevance score, and from the LongTailAugmented dataset we select the top ranked images based on their augmented tag relevance score. We select the top 1,300 ranked images per tag, as the settings of ImageNet, or less if not as many available. We evaluate on validation images from ImageNet, with AP per tag, or MAP overall.

Features. We recognize that deep learning has shown a great improvement in image classification. The features used from the last layer of a Convolutional Neural Network (CNN), or one layer before the last have become popular and widely used [56]. The CNN is mostly trained on images from ImageNet. Thus, these features have already seen all the classes of ImageNet. Since we evaluate on ImageNet classes, and we want to see the performance of using only few images from a rare tag to learn a concept detector, we do not use the CNN features to keep the evaluation fair. Instead as features we employ the once popular Fisher vector encoding [96] with a GMM of size 1,024 and a spatial pyramid of 1x1 and 1x3. We extract SIFT descriptors with dense sampling at every 6 pixels at two scales, PCA reduced to 80D. As a classifier we use a one-vs-all linear SVM.

Results and analysis. We visualize the results in Figure 3.6 (c). As expected, the less frequent the tags are, the lower the average precision is. Similarly as in tag relevance, some tags have low AP, even though they are frequent. For example tag 1:*light* has

Fig. 3.7: Example images of tags with the images of their synonyms. Some synonyms augment the rare tags with good images, while others are more ambiguous.

low score since its meaning is ambiguous, as well as tags 3:*blowfish*, 4:*bluejean* and 7:*washbowl*, see Figure 3.5. For most rare tags, the results improve when we augment the training data with images tagged with their synonyms. We show augmented examples of few tags in Figure 3.7. For tag 26:*chrysanthemum dog* and 53:*galeocerdo cuvieri* the augmented images are quite relevant, improving the tag annotation result from 55% to 74%, and 2% to 56% respectfully. For tags 33:*barracouta* and 34:*sleuthhound* the results do not improve. For example *snoek* is a synonym of *sleuthhound*, and also a name of a car model, which adds noise to the training data of *sleuthhound*.

Overall, learning detectors from tagged images with rare tags gives poor performance since there are not enough images to learn reliable detectors. When simply augmenting the training data with images tagged with synonyms, the MAP improves from 53% to 79%. We conclude, learning detectors for rare tags from the long tail can be improved by augmenting the training data with images tagged with their synonyms.

3.5 Conclusions

We have looked into the long tail of social tags, and analyzed three questions: *What tags constitute the long tail?*, *What happens when rare tags are augmented?* and *What is the effect of rare tags on multimedia retrieval scenarios?*. We uncover that the long tail has valuable tags of objects, scenes and fine-grained animal categories. We show that by augmenting the rare tags with simple semantics, the performance of tag relevance and detector learning improves considerably. Thus, we conclude the rare tags from the long tail are valuable and perform better when augmented with semantic knowledge.

Pooling Objects for Recognizing Scenes without Examples

4.1 Introduction

This chapter strives to recognize the scene of an image, such as *beach*, *supermarket* or *youth hostel*, without using any labeled scene example. Evidence from *supervised* scene recognition suggests that objects emerge in the representations learned from millions of labeled scene images [140], inspiring us to explore whether object classifiers can be leveraged to recognize scenes without examples¹.

The common approach to address classification without examples is to define scene-specific attributes, train attribute detectors and rely on an attribute-to-scene mapping for classification [42, 59, 62, 93]. In [59] for example, Lampert *et al.* use attribute annotations on images from the SUNAttributes [87] datasets, to learn attribute models and create attribute-to-scene mappings. Rather than relying on attributes and attribute-to-class mappings, recent works [14, 40] showed that events, actions and emoji's can be recognized without examples using a representation based on object classifier scores and object-class affinities estimated from a social media corpus [112]. Inspired by these efforts, we explore in this chapter whether object classifier scores, together with a semantic embedding learned from social media, are as well a suitable representation for scene recognition without examples.

Supervised image recognition is currently dominated by deep learning. Convolutional

¹Published in *the Proceedings of ACM International Conference on Multimedia Retrieval*, 2016. Best paper award [53].

Fig. 4.1: We investigate four approaches of object pooling in semantic embeddings for scene recognition without examples and rely on a freely available large-scale object corpus, social media text and an ontology.

neural networks, the deeper the better [35, 111], learn the optimal image representation by tuning its millions of network parameters from huge amounts of labeled data, such as ImageNet [19] (for objects) or Places2 [141] (for scenes). In [140], Zhou *et al.* show evidence that objects emerge in deep nets learned from scene images. Motivating us to rely on a representation of objects to recognize scenes. By doing so we don't need any scene example. Our object classifiers correspond to the output of the last softmax layer of a very deep convolutional neural network trained on ImageNet [111]. These object classifiers have a lingual correspondence derived from nouns in WordNet [78], making them well suited for recognition without examples using a semantic word embedding, *e.g.* [77, 82, 98], as prior knowledge. Rather than simply using the response of all available object classifiers, we pose the question: *What objects to pool when representing scenes?*

In [40] Jain *et al.* evaluate two methods for pooling object classifiers to recognize unseen actions in video; one based on the semantics of the class and the other based on the

appearance of the test imagery. We adopt and adapt their methods for scene recognition in images. In addition, we introduce a hierarchical pooling taking into account the ontological relations of objects as defined in WordNet, see Figure 4.1. Finally, we also introduce a pooling that considers the spatial extent of object classifiers by exploiting a recent box proposal [145]. This spatial pooling allows to answer complex search queries like “finding beach scenes with a human on the right of the image”. We evaluate our proposals for scene recognition without examples on SUNAttributes [87] and Places2 [141]. Before detailing our approach we first provide more context on related work.

4.2 Related work

The goal of recognition without examples, as introduced by Lampert *et al.* [59], is to recognize a set of unseen classes Z using some set of training classes Y which do not overlap with the test classes ($Z \cap Y = \emptyset$). To allow for knowledge transfer in between, usually a set of attributes A is defined which semantically relate to both the training classes Y and the unseen test classes Z . The attributes come with annotated examples X , forming a training dataset $D = \{X, Y, A\}$, from which attribute models are learned, and a human defined class-to-attribute mapping is used to transfer knowledge. In this chapter, instead of manually and carefully choosing training classes Y and attributes A to well fit the characteristics of the unseen scene classes, like scene attributes [87], we investigate pooling off-the-shelf object classifiers derived from ImageNet [94], with 15K diverse object categories, to recognize scenes without examples.

Instead of relying on attributes, Norouzi [82] proposed to use a semantic embedding space to allow for knowledge transfer between the training classes Y and unseen classes Z . The semantic embedding space is trained to minimize the distance between words with similar semantics. We use the word2vec semantic embedding [77], this is a single hidden layer neural network, trained to predict the surrounding words given an input word. The semantic embedding representation is a d -dimensional vector derived from the activation values of the hidden layer. This ensures that, the similarity between two words in the word2vec space will be close only if the words are found in similar semantic context. Cappallo *et al.* [14], show that a word2vec model trained using the textual data accompanied with images, works better than a model trained on Wikipedia, probably because of the more visual nature of these descriptions. We follow their approach and train on a large social media dataset [112], containing the surrounding text (tags, captions, *etc.*) of 100 million Flickr images. Although semantic embeddings have been used before for zero-shot learning [14, 40, 82] it has not yet been explored for objects and scenes.

Fig. 4.2: Flow of data when recognizing scenes without examples using object pooling. For an input test image we calculate its object prediction scores from a very deep convolutional neural network, learned to classify 15K objects from ImageNet. We calculate a knowledge transfer matrix between objects and scenes from a semantic similarity embedding learned using text of Flickr images. The scene with the highest score is assigned to the input test image.

To represent scene and object class names composed of multiple words, we aggregate them using Fisher vectors [16]. The Fisher vector characterizes each word by its deviation in the semantic embedding space *w.r.t.* a Gaussian mixture model. This has shown to perform better than simple word averaging [16], also in the context of recognizing actions and events in a zero-shot setting [40]. We use the Fisher vector for computing similarities between objects and scenes names.

The semantic structure defined by an ontology has been explored for many retrieval and classification problems. Fergus *et al.* [27] leverage the WordNet hierarchy to define semantic distance between any two categories, and use it to share labels for training classifiers. Zhu *et al.* [142] use the WordNet hierarchy to complete imprecise and incomplete social media tags by, transferring tag examples from child to parent nodes. When a tag occurs rarely in social media, Kordumova *et al.* use semantics from the WordNet ontology to enrich training data [54]. In [123] Vreeswijk *et al.*, compare training a classifier of a parent node with an union of their constituting child nodes. Different from these approaches, we investigate the WordNet hierarchy to steer the object representation in a semantic embedding for recognizing scenes without examples.

Object proposals [4, 115, 145] suggest the most likely positions in an image to contain an object. Besides being successful in object detection, they have also proven to obtain better results for image retrieval and classification [131]. Inspired by [131], we use (unsupervised) object proposals with object classifiers trained on the whole image, for the task of recognizing scenes. We prefer object proposals over object detectors, for

two reasons: detection increases the complexity of our approach, and limits the number of object categories to at most 1K, for which bounding box annotations are available. Encoding the spatial layout of objects allow us to answer complex queries, where instead of just querying with the scene name, we include object names and rough estimates of its position, like “finding castle scenes with grass on the bottom of the image”. Different from other complex query approaches [12, 69, 81, 136], which all emphasize on word combinations, we propose a complex query with a scene, an object *and* its position. Next we describe the details of our approach.

4.3 Pooling objects for scenes

Our goal is to recognize unseen scenes Z , using a set of off-the-shelf object classifiers Y (which do not overlap with the scene classes, i.e. $Z \cap Y = \emptyset$) and a semantic embedding space S . Similar to [82] we use a convex combination of object scores and semantic similarities to assign a class z to a test image v :

$$C(v) = \operatorname{argmax}_{z \in Z} \sum_{y \in Y} s(z, y) p(y|v), \quad (4.1)$$

where $p(y|v)$ is the object prediction score for class y given test example v and $s(z, y)$ denotes the semantic similarity between unseen class z and object class y .

The object prediction $p(y|v)$ is the last network layer (after softmax normalization) of a very deep convolutional neural network [111]. We use 15K object categories from the ImageNet dataset, for which at least 200 images are available for training, similar to [14, 40].

The semantic similarity $s(z, y)$, is the cosine similarity between a scene z and an object y in the word2vec space:

$$s(z, y) = \cos(s(y), s(z)) = s(y)^\top s(z), \quad (4.2)$$

where $s(\cdot)$ denotes the d-dimensional Fisher vector encoding of a word2vec vector. We visualize the flow of data for recognizing a scene in an input image in Figure 4.2.

When considering off-the-shelf object classifiers, we expect that not all objects will contribute equally in describing scenes. ImageNet follows a hierarchical structure, derived from WordNet nouns, where objects like *animal*, *vehicle*, *plant*, *food* appear high in the hierarchy, followed by object categories like types of animals *cat*, *dog*, *bird*, *horse*, and fine-grained objects in the lowest level of the hierarchy, for example types of horses *tarpan*, *hackney*, *clydesdale*. However, it is not clear how specific the objects need to

be in order to better retrieve scenes. We also expect that for different scenes, different objects should matter. To recognize a *beach* scene objects like *sand*, *ocean*, *sky* would be beneficial, whereas *bed*, *apple*, *church* would have a negligible or no influence. We describe and investigate four ways of pooling objects for scene recognition.

Semantic Pooling. This method considers the semantically closest objects for each scene class. Similarly as in [40], it does so by first ranking the objects based on their semantic similarity scores to a scene, calculated with Equation 4.2. For each scene $z \in Z$ we then select the closest objects, forming a subset of Y_z from the complete objects set Y :

$$Y_z = \{y \in Y \mid s_{yz} > t_s\} \quad (4.3)$$

where t_s is a hyper parameter of selecting the objects with larger scene similarity, or the similarity value of the m -th closest object, so that we select the top m objects. In this case, for a test image, scene scores will be computed as in Equation 4.1, and only changing the summation over the set Y to the subset Y_z .

Appearance Pooling. A test image v is represented with probability values for each object category $p_{vy}, \forall y \in Y$. Each probability p_{vy} approximates the presence of an object of class y in the image. We follow [40, 82] and we sample only the prominent objects present in an image, resulting in a subset of objects:

$$Y_v = \{y \in Y \mid p_{vy} > t_p\} \quad (4.4)$$

where t_p is a hyper parameter which regulates selecting objects with minimum presence probability in the image. To select the top m present objects, t_p takes the value of the m -th ranked object. In this case, the scene scores of a test image will be computed as in Equation 4.1, with the only difference in the summation over the objects set, from Y to a subset Y_v . Doing so, we avoid summing over many unrelated object classes for an image, with low probability scores, which do not appear in the image and will not contribute to the scene recognition.

Hierarchy Pooling. The hierarchical structure of objects portrays parent-child relationship of object categories, where the children objects are a subcategory of their parent class. The leaf nodes, represent the most fine-grained object classes of their tree branch. We question how important are objects from different hierarchy levels for recognizing a scene. One would most naturally expect to find a *tree* in a *forest*, and a *sofa* in a *living room*, and to discriminate these two scenes the specificity of objects like *tree*, *sky*, *sofa*, *tv* would be sufficient, without going further into knowing the type of a tree. And to discriminate a *bamboo forest* from a *broadleaf forests*, specific types of tree objects might matter, like *bamboo tree*, *oak*, *beech*.

Fig. 4.3: Flattened hierarchy tree of the ImageNet objects, indicating the number of objects per level. The first level contains very general objects, becoming more specific as the levels grow, narrowing down to fine-grained objects. We show word clouds of typical object names in the first, middle and last level.

The hierarchical structure of objects, as defined in the WordNet ontology, forms a tree. In Figure 4.3 we show a flattened tree version of a WordNet subspace, using only the nouns from ImageNet object categories. We denote as $L = \{L_1, \dots, L_k\}$ the set of all $k = 16$ tree levels from the flattened tree. Each level $L_i, i \in \{1, \dots, k\}$ is a subset from all objects Y with depth i :

$$L_i = \{y \in Y \mid \text{depth}(y) = i\} \quad (4.5)$$

For a new test example, we investigate how accurate will the objects from different levels predict its scene class. We use objects from each level separately by changing the summation over the set Y in Equation 4.1, to objects from only one level set L_i at a time.

Position Pooling. We employ [145] to generate bounding box coordinates over the image $[x_{\min}, y_{\min}, x_{\max}, y_{\max}]$, most likely to contain an object. For a test image, we exploit whether being more precise by object scoring the bounding box regions in an image, helps in recognizing the scene. Same as for the whole image, we use the last network layer with softmax normalization of [111], to compute object prediction scores on image regions. For m bounding boxes $\{b_i\}_{i=1}^m$, generated with an object proposal method, the object scores are $p(y|b_i), y \in Y$. There are multiple ways to pool object classifiers when objects scores of image regions are known. We employ two simple approaches, max pooling and average pooling. For max pooling, we consider object classifiers with the largest prediction scores from all generated positions in an image:

$$Y_b = \{y \in Y \mid \max(p(y|b_1), \dots, p(y|b_m)) > t_b\} \quad (4.6)$$

where t_b is a hyper parameter, selecting the top scored objects among all image regions. For average pooling we consider object classifiers with average prediction scores of all bounding box positions in an image, larger then a value parameter t_a :

$$Y_a = \{y \in Y \mid \text{avg}(p(y|b_1), \dots, p(y|b_k)) > t_a\}. \quad (4.7)$$

The position of objects in scene images allows for new scene retrieval scenarios. With a (textual) scene query, one can also specify what object(s) to be present, as well as where in the scene the object(s) should be. We choose three adjectives to approximate the *where* for horizontal position, (*left, center, right*), and three for vertical position (*bottom, middle, top*). Thus, an example query would be: *beach* with *human* on the *right*. We first find all images where the query scene was recognized, and then we rerank them based on the maximum prediction response for that object. If the query has specified also the object position, we consider object responses only from bounding boxes which center falls in that scene area. We compute the horizontal and vertical center of bounding boxes as $b_h = (x_{\max} - x_{\min})/2 + x_{\min}$, and $b_v = (y_{\max} - y_{\min})/2 + y_{\min}$. If I_h, I_w are the image width, height, and $t_w = I_w/3, t_h = I_h/3$, is one third of the image horizontal

and vertical area, for (*left, center, right*) we consider the boxes which center falls in the intervals $(0, t_w)$, $(t_w, 2t_w)$, $(2t_w, I_h)$, and for (*bottom, middle, top*) we consider bounding boxes with center in the intervals $(0, t_h)$, $(t_h, 2t_h)$, $(2t_h, I_h)$ respectively.

We are now ready to evaluate our proposal quantitatively and qualitatively.

4.4 Experiments

4.4.1 Datasets

SUNAttributes. The SUNAttributes dataset has 14,340 images hierarchically grouped in 3 levels [87]. Level 3 has 717 fine-grained scenes like *airplane cabin*, *athletic field outdoor*, *cybercafe*. Level 2 groups the fine-grained scenes in 16 categories like *shopping and dining* or *water, ice, snow*, and level 1 has three general categories of scenes *indoor*, *outdoor natural* and *outdoor man made*. Each of the 717 scenes in level 3 has 20 images per scene, which we use as test data. We scale all images to same dimension of 256x256 pixels.

The dataset also comes with 102 attribute annotations. When using attributes [59], the dataset is split into two parts of 10 random splits, where 90% of the images are used for attribute learning, and the rest for testing. We follow the same approach, and create 10 random splits. Because the data is split into disjoint train (90%), and test classes (10%), 71 classes of the dataset are present at test time for each random split. For our approach we only use the images from the test splits, since we do not need any scene images for learning. Same as in [59], we report results in accuracy. For scene level one and two, a prediction is considered correct if the ground truth class and the predicted class paths run through a common node in that level.

Places2. The Places2 dataset [141] contains more than 10 million images comprising 401 unique scene categories. The scene categories vary from indoor scenes like *bazaar indoor*, *cafeteria*, *toyshop*, and outdoor scenes like *bazaar outdoor*, *creek*, *desert sand*. The dataset features up to 30,000 training images and 50 validation images per class, scaled to 256x256 pixels. Since our goal is to retrieve scenes without any training example, we do not use the training set, and evaluate our approach on the validation set. This dataset was first released in the scene recognition subtask of the ImageNet 2015 challenge [94], and to the best of our knowledge, we are the first to report scene recognition results without examples on this dataset. As evaluation metric we report top-5 accuracy, as set by the challenge.

Fig. 4.4: Pooled objects for two scenes, (a) word clouds of semantically pooled closest objects, (b) and (c) image examples with their top-scoring object predictions. With appearance pooling, the correct scene was assigned to the examples in (b), whereas in (c) the *glacier* image was misclassified into an *ice shelf* scene, and the *movie theater indoor* image into a *steel mill* scene. This happens when the top predicted objects in an image are closer in the semantic embedding space to the misclassified scenes. Overall, all pooled objects look reasonable.

4.4.2 Implementation details

Object classifiers. We train an off-the-shelf deep convolutional network [111], with 22 layers, on all ImageNet categories having more than 200 images available for training. As object classifier scores we use the last network layer with softmax normalization. After removing 304 objects having an exact overlap in name with the scene classes, we end up with 14,989 object classifiers.

Object hierarchy. We use the parent-child noun relationships provided by WordNet [78] to construct a hierarchical tree². The WordNet tree has 20 depth levels, from which we create a flattened subset version, where we take only the nouns that appear as object classes in ImageNet, and keep their depth position as in WordNet. Besides the *sun* object which appears in the second level, the first four levels do not contain objects from ImageNet. Therefore, we ignore them, and start counting from the fifth as a first level, resulting in a 16-level flattened tree, see again Figure 4.3, with ImageNet objects only.

Objects positions. We generate object positions with EdgeBoxes [145]. As we observe a stable recall, we use the top 25% of bounding boxes after ranking by their objectness score. The number of bounding boxes varies per image, between 2 and 1025 for SUNAttributes, and between 1 and 1077 for Places2. We also employ the common non-maximal suppression on the sorted boxes with an overlap threshold of 0.5 [4, 115, 145].

Semantic embedding. We train a 500-dimensional skip-gram word2vec model using the metadata of the YFCC100M dataset [112], which has 100 million Flickr images accompanied with titles, descriptions and tags. We encode the word2vec vectors with Fisher vectors following the setting of [40]. We learn $k=2$ Gaussian Mixture components, apply PCA to reduce the dimensionality by a factor of 2, and use only the partial derivatives *w.r.t* the mean. In this way, the encoded word2vec Fisher vector keeps the dimensionality, $d = 500$, as the original word2vec.

4.4.3 Semantic pooling and appearance pooling

We first investigate semantic pooling and appearance pooling, where the first selects the semantically closest objects to scenes Y_z , and the second the most prominent objects per image Y_v . We show pooled object examples for two scenes in Figure 4.4. We vary the threshold parameters, t_z and t_v , in a way that we select the top m ranked objects, where $m = |Y_z|$ or $m = |Y_v|$. In both cases, we see similar pattern in the results, see Figure 4.5.

²Tree visualisation <http://isis-data.science.uva.nl/svetlana/TreeD3/>.

Fig. 4.5: Impact of the number of semantically closest objects per scene $|Y_z|$, and most prominent objects per image $|Y_v|$, for scene recognition with no examples on the SUNAttributes and Places2 datasets.

When only selecting the single most semantically similar object per scene, $|Y_z| = 1$, the average accuracy for SUNAttributes is 30.91%, and for Places2 is 27.32%. When compared to random accuracy, 1.41% and 0.25% respectively, the results indicate that even one object per scene can help in discriminating scenes. As we sample more objects the accuracy improves. Up to 10 objects it grows rapidly, followed by a slow progress, which saturates after about 1K sampled objects. If we select the single most prominent object in an image, $|Y_v| = 1$, the accuracy for SUNAttributes is 29.82%, and for Places2 is 25.02%. Thus, when one object is selected, the most semantically similar one is more informative than the most prominent one. However, as we sample more prominent objects, the accuracy comes close to the maximum after only 100 objects are selected.

Object classifiers recognize scenes without examples reasonably. We expect that as object classifiers improve further [35], the more accurate the scene recognition without examples will become. From now on we use the 100 most prominent objects in an image to recognize its scene.

4.4.4 Hierarchy pooling

Here we investigate what is the discriminative power of objects from different hierarchy levels when recognizing scenes without examples. In Figure 4.6 (a), we show results

Fig. 4.6: Scene recognition without examples using a representation containing classification scores from ImageNet objects, where we vary the objects in the representation using hierarchy levels. In (a), each hierarchy level forms one object representation, and we show results on Places2, and Level3 SUNAttribute scenes. Object classifiers from the middle levels are most discriminative, whereas the first levels (general objects), and the last levels (fine-grained objects), are not discriminative for recognizing scenes. In (b), in each step objects from the next level of the hierarchy are added, and in (c) in each step objects from the previous level of the hierarchy are added. Again we conclude that the objects from the middle level contribute the most for discriminating scenes.

when we pool objects from each level individually. Interestingly, the top- and low-level objects show lowest accuracy. The top-levels contain mostly general object categories, and the low-levels have fine-grained object categories, see Figure 4.3. From the results we conclude that general and fine-grained object categories from the top- and low-levels of the ImageNet hierarchy, are not discriminative for recognizing scenes. The objects from level 6 and 7 show best performance. If we use objects from only these two levels, in total 5317, or one third of all objects, we already achieve an accuracy of 32.5% for Places2, and 34.5% for SUNAttributes.

The number of objects in each level varies. In our experiments we use the top 100 most prominent objects in an image. Level 1,2,3,15,16 contain less than 100 objects, thus, we believe that the number of objects present might also influence the scene recognition. Therefore, we perform two additional experiments where we investigate what happens if in each step we add objects from the next level, and see how much each level contributes. We first start from level one, and continue adding up objects from the next levels until all objects are used. We show the results in Figure 4.6 (b), for Places2 scene categories, as well as for all three scene levels of SUNAttributes. We see a similar pattern for all scene categories. The average recognition accuracy grows substantially up to level eight. From the ninth level on, there is minor growth in the accuracy. This again confirms that the lower levels do not add extra information which contributes for the recognition of scenes. As we go the other way around, we start from the lowest level and add up objects from the previous levels, see Figure 4.6 (c), the results confirm the previous findings. The fine-grained object classifiers have a small contribution for discriminating scenes, middle level objects help the most, and the general objects from the top levels do not contribute in recognizing scenes.

4.4.5 Position pooling

When we sample the top 100 objects by average pooling over image positions, as generated with EdgeBoxes, we get slightly better results than max pooling, 32.57% vs 32.31% for SUNAttributes, and 31.87% vs 31.35% for Places2. When we merge pooled objects from image positions, for $t_a = 0.3$, with objects pooled from the whole image, the scores do not change much. For SUNAttributes it changes from 41.62% to 41.46%, and for Places2 from 37.23% to 37.63%, for appearance and appearance+position pooling respectively. When we look into individual scene accuracy, in some cases position pooling helps, whereas in others, appearance pooling from the whole image is enough. For example the objects *milldam* and *waterspout* pooled from positions, helped to correctly classify a *dam* scene which was misclassified as *river*. In another example, a *field cultivated* image was misclassified into a *pasture* scene, when the objects *geyser*, *waterspout*, *parhelion*, *cloud*, *azure*,... were pooled from image positions. We did not find a

Fig. 4.7: Top and bottom ranked images when as query we use scenes+object (first two rows), and scenes+object+position (last two rows). Over the top retrieved images we draw the bounding box which contributes the most for finding the object. We do not show bounding boxes on the lowest ranked images, since the objects are not present and the object scores are low, so it does not make sense. Interestingly, the top ranked images are quite relevant to the query, and when the object position is specified, the filtered bounding boxes from only that area in the image indeed help. The bottom ranked images, in all cases are missing the object we query with from the scene.

general rule on when position pooling helps over the whole image, and since it requires more processing, we recommend to use appearance pooling from the whole image only.

A more interesting benefit of knowing the object positions is that they allow retrieval with complex queries. In Figure 4.7 we show qualitative results for four different queries, two with scene+object and two with scene+object+position. We show the top and bottom retrieved images for each query. For the top retrieved images we also draw the bounding box which contributes the most for the object. We do not show bounding boxes on the lowest ranked images, since the object is not present, the object scores are low and do not make sense. Interestingly, the top retrieved scenes contain the object we query with. When an object position is specified, like *right* or *bottom* as in the last two queries in Figure 4.7, the most contributing bounding box of the object in the top retrieved images, comes from the *right* or *bottom* area in the scene. The bottom ranked images, in all cases are missing the object we query with from the scene. Thus, we observe that knowing the object positions in scene images can indeed be used for retrieving images with complex queries of scenes, objects and their spatial extent.

4.4.6 Comparison with attributes

Finally we compare pooling off-the-shelf objects with attribute representations for recognizing scenes without examples. We report all results in Table 4.1.

Our object pooling performs better than the direct attribute prediction (DAP) reported by Lampert *et al.* [59], 41.63% vs 22.20%, on the SUNAttributes dataset. Since their results were calculated using GIST and HOG features, we repeat their experiment with our convolutional neural network features, for fair comparison. We take features from the layer before the last one, of the same network we use throughout all our experiments. We L_2 normalize the features, and for each attribute we learn a linear binary classifier with libsvm and probability outputs. In this setting, the DAP approach improves from 22.20% to 34.78% on SUNAttributes. Still the result does not go over our accuracy with object pooling.

We can not report results with DAP on Places2, since this approach requires attribute-to-scene mapping, which is not available on this dataset. Instead, we investigate how attributes work with a semantic embedding [82], when attribute annotations are missing. For all 102 attribute names of SUNAttributes we find their match among the 15K objects. 22 attributes have an exact name match, and for 80 we use the closest object, like *sailing/boating* with *sailboat*, or *sunbathing* with *sunbather*. As expected, the attributes with semantic embedding perform lower than a DAP setting, with an accuracy of 10.07% on SUNAttributes (71 classes), and 1.27% on the more challenging Places2 dataset (401 classes). The results are comprehensible, since there are only 102 attributes, and in our setting we use 100 objects pooled from a set of 15K object classifiers.

We conclude, scenes can be recognized by pooling relevant objects, better than attributes, when a large corpus of object classifiers is available, and with semantic embedding learned from a large social media corpus to guide the knowledge transfer between objects and scenes.

4.5 Conclusions

In this chapter we recognize scenes without examples by using off-the-shelf object classifiers from a large collection of 15K object classes. We pose the question *what objects to pool when representing scenes?*. Throughout extensive analysis by experiment on two large-scale scene datasets we conclude that object classifiers from a large objects collection are suited for scene recognition without examples. From the four pooling methods considered we recommend object pooling by appearance, as it requires only 100 objects per image. General and fine-grained object categories, from the top and bottom

	SUN Attributes	Places2
Random	1.41	0.25
Attributes DAP [59]	22.20 ± 1.60	n.a.
<i>Using deep learning*:</i>		
Attributes DAP [59]	34.78 ± 2.54	n.a.
Attributes-Embedding [82]	10.07 ± 2.22	1.27
Object pooling	41.63 ± 3.16	37.46

Tab. 4.1: Comparison of object pooling by appearance vs attributes for recognizing scenes without examples. Between the lines marked with deep learning*, we show results which are not available in the cited papers, but computed using their approach with our deep learning setting. Object pooling recognizes scenes better than attribute representations on both datasets.

level of the WordNet hierarchy, do not contribute much for scene recognition and can be avoided in the image representation. When the object classifiers are run on object proposals instead of the full image, our proposal allows for complex queries including position quantifiers, like “finding beach scenes with a human on the right of the image”. Finally, object pooling outperform attribute representation for scene recognition without examples, with the additional benefit of learning the knowledge transfer from social multimedia, rather than manually specifying the scene-to-attribute mappings.

Searching Scenes by Abstracting Things

5.1 Introduction

In general, scenes provide the context by which objects receive their meaning. A picture of the *sea* makes the large pile in the middle a likely candidate to be an *iceberg* or an *oil tanker*. And reversely, the understanding of scenes can be derived from the objects in the scene. The types of objects may be important, such as a *cow* and another *cow* and a *tree* to denominate the scene as a *meadow*. With the current achievements of object recognition [23, 94] some objects may contribute much to the recognition of the scene [64, 83]. Other objects may contribute little or nothing, like a *handkerchief* or a *smartphone* as they may occur in any scene. Where the recent successes and excitement of the field lie in object identification [23, 94], in this chapter we argue that next to object types, there is another source of information contributing to what makes a scene.

We note on the basis of cognitive experiments [100] that object composition provides by itself a clue for the recognition of scenes. The reference shows that humans can recognize scenes even when individual objects are reduced to blobs that only retain their size, aspect ratio and position. As a consequence, when position and size are violated a scene may appear disorganized, see Figure 5.1. By violating these basic properties, recognizing objects in scenes by humans resulted in reaction time and accuracy deficit [11]. There are reasons why the scene has a specific spatial layout. Objects obey the laws of physics. They must be supported by a horizontal surface. Two objects can not occupy the same physical space. Next to physical reasons, objects also have a certain semantic likelihood of appearing in a particular scene [33]. Imagine going to a friend's housewarming party, you have never seen the house before, but you will not be surprised to see a coffee table next to the sofa in the living room, or framed pictures on the walls,

and you will go look under the sink for disposing the waste. Next to position, object size is dependent of cultural habits, human size and purpose [18]. The purpose of a sofa is for sitting so it needs to accommodate a person, whereas the purpose of a cup is to be hold in hands and accordingly it is designed smaller. Object size also relates to scene depth [113], which in turn affects the number of objects [9]. All of this provides ample motivation to study the compositional layout of objects in a scene.

We have found that the removal of all object identities, and calling all objects in the scene “things”, reveals a preference for a scene-type specific composition of things in a scene. A *theater* and a *marching band*, a *soccer crowd* and a *forest* will all demonstrate a certain spatial regularity of the things they are made of, albeit a spatial regularity of a different kind each. The interior of a sleeping room will usually demonstrate a limited number of spatial layouts, which contribute to the recognition of the scene. In this chapter, we start from “things” as generated by modern object proposals, referred to as objectness [4], PRIM [75], or selective search [115]. We note that some works have referred to a thing only if it is an “object that has a specific size and shape”, and to stuff for “material defined by a homogeneous or repetitive pattern of fine-scale properties, but has no specific or distinctive spatial extent or shape” [28, 37]. In our work we name all existing objects, parts and stuff under “things”. We investigate image representations composing the ensemble of these things, on the basis of their immediately observable features: position, size, aspect ratio and color, and those only, see Figure 5.2. We could refer to the representation as sceneness, SRIM, or selective layout, but we prefer to use *things syntax*.

To relate visual information with linguistic meaning is another challenging and open area of research. Linguistic descriptions are usually generated using object identities [25, 57, 64] or attributes [86]. In this chapter we take a different approach and generate descriptions with abstract statements only from the immediately observable features of “things”, for example “Green small squared thing at top middle” or “Blue large wide thing at top right”. Whereas [25, 57, 64, 86, 146] use manual annotations and require learning to generate linguistic descriptions, we generate linguistic statements from the image itself. We investigate the effectiveness of the abstract statements to retrieve scenes without the need for examples.

Furthermore, we investigate querying the things syntax with abstract block illustrations, as presented in Figure 5.5, inspired by Mondrian’s neoplastic style [79]. Mondrian studies the order of the most abstract forms to designate a feeling for the scene as a whole. The painting *Victory Boogie Woogie* is a good example of expressing the crowdedness of a joyful city. In our work, different from [146], we remove object identities and use only abstract block illustrations that preserve the immediate observable properties of things, see Figure 5.2. We study the effectiveness of block illustrations to search scenes.

Fig. 5.1: (a-b): The scene type does not change when particular object are replaced by others in a similar semantic object-class. (c-d): Scene semantics are violated, not because the objects do not belong but because they deviate from their typical shape, size and placement in the scene. Images (c-d) © René Magritte (fair-use).

We make three contributions in this work. First, on the basis of the above motivation, we approximately localize the things in a scene as the output of objectness [4], selective search [115], and PRIM [75], and study the properties of their immediate observables without proceeding to identify these things by type. Second, we study abstraction of the things syntax by translating it into abstract statements and their effectiveness to query scenes. And thirdly, we study abstraction of scenes into block illustrations of things to search among different types of scenes.

5.2 Related Work

Scene representation. To represent and to recognize scenes more approaches have been proposed than we can cover here. We make no attempt to be complete. Instead, we summarize the main research strands which developed from low level statistics, *e.g.*, [31, 83, 96, 128], through mid-level unnamed discriminative regions, *e.g.*, [20, 32, 34, 46] to high-level representations consisting of objects identity scores, *e.g.*, [8, 64] or

Fig. 5.2: Examples of things properties: *position*, *size*, *aspect ratio* and *color*, when things are visualized as cuboids over the image.

attributes [59, 86]. Each of these are successful in their own right and for their intended purpose. In this chapter we follow a different path to search the categories of scenes and while doing so we demonstrate that for scene recognition there is another informative cue besides appearance features and object types.

Scene statistics. We take as inspiration the work of Greene [33], who investigates object statistics like object density, unique object density, mean object size, and variability of objects, to discriminate scenes. These statistics are calculated from manually annotated objects in scene images, with known object types. Interestingly, Greene shows that this ensemble of statistics is sufficient for above-chance scene recognition. Motivated by these results, we investigate immediately observable properties of nameless things. Thanks to the modern object proposals methods [4, 75, 115], we can benefit from automatically suggested locations of things. We investigate how close do the immediate observable properties of automatically generated object proposals come to the properties of manually annotated objects, and compare their discriminative power for retrieving scenes.

Object proposals. We use the recent achievements of the localization of objects to find things in a scene [3, 22, 55, 75, 89, 115, 145]. The goal of these methods is to find locations in an image that have a high likelihood to contain an object. By being efficient in suggesting the most likely locations, these methods allow more computation time to be spent on feature representations and classifying their identify for object detection, which has led to great successes. Very often the proposed locations are around an object part or around some texture shapes. Since we are considering things, there is no longer an objective to identify an object. An object part can often be considered as an object in its own right, for example buildings in a cityscape have windows, apartments and individual bricks. In natural scenes, this recursive fragmentation follows certain rules biasing the statistics of size towards a Weibull distribution [29]. Starting from [3, 75, 115] we consider their output locations in a form of a bounding box over an image to hold a thing, and we refer to them as things proposals in the rest of this chapter. We gladly use their valuable output, right or wrong, without proceeding to classify the content of their boxes into an object type, as these methods are designed for. In this

chapter we are satisfied with the position, size, aspect ratio and color of the proposed boxes to represent the ensemble of just the things.

Semantic scene representation. Predicting written descriptions from visual information is an interesting and challenging problem [7, 25, 26, 57, 84, 85, 135]. All these works use a variety of approaches, like generating semantic sentences relying on object detectors [57, 64], relying on semantic attributes [7, 25, 85, 86], using a large corpus to extract written descriptions [26, 84], language statistics [135] or generating verbs by looking into spatial relationships of objects [57]. Notable efforts have also been done in creating datasets with semantic descriptions and sentences [70, 90]. In this chapter we take a different approach by removing objects, attributes and verbs from written descriptions. We describe a scene by mapping properties of things to adjectives and adverbs, for example, *Green small squared thing at top middle* or *Blue large wide thing at top right*.

Zero-shot recognition. The benefit of having high-level written descriptions of images is that they enable humans and computers to communicate with natural language. One popular application is zero-shot recognition as pioneered by Fahradi *et al.* [25] and Lampert *et al.* [59]. In a zero-shot setting, there are no available category labels on images to learn from, instead human descriptions of zero-shot classes are gathered and matched against detected scores of attributes or object detectors in test images. This setting requires gathering annotations and learning models of attributes [25, 59, 85] or objects detectors [40, 64], from the same scope as the test classes. For example animal attributes can not be used to recognize vehicles. Images from the zero-shot classes are usually shown to humans in the description process, since if a person has never seen an *archeological excavation* scene, it would be difficult to provide attributes for it. In this chapter we remove all scope-dependent object identities and attributes, and represent scenes as an ensemble of “things”. We pose the question, *Can we retrieve a scene with only abstract statements of immediately observable things properties?*. An important benefit of the abstract statements is that they are generated directly from images, without the need to learn any attribute or object detectors beforehand.

Visual abstraction. Zitnik *et al.* [146] have investigated the potential of using abstract images to study high-level semantics, like semantically important features, relations between saliency-memorability of objects, and mapping of sentences to abstract scenes. A dataset of abstract images is collected by asking users to draw images with clip art objects of children, trees, animals, food, toys etc. The authors number four benefits of abstract images: 1) they remove the reliance on noisy low level object and attribute detectors 2) they avoid tedious hand labeling on images, 3) they allow for direct study of high level semantics, and 4) allow to automatically generate sets of semantically similar images. We stand inspired and propose to investigate abstract images from a different perspective, using nameless things. Different from [146] we remove all object

identities and create abstract block illustrations capturing the position, ratio, size and color of things in a scene. Abstract images have also been investigated as a modality for zero-shot recognition by Antol *et al.* [5]. The authors argue that visual modality is needed because some images are not easy to be described in semantic terms, like images of interactions between people, which is also the topic of their work. Since we are interested in scene images, we note that describing scenes in semantic terms is also difficult in a number of situations. For example, when children between the age of 5 to 6 are asked to describe a memory, they show more accurate information by drawing the scene than when they describe it semantically [13]. Also there are scenes where object identities are unknown, but their ensemble of things is well defined, like cosmic and microscopic scenes, architectural inspiration or abstract paintings. We investigate searching scenes by block abstractions.

In this chapter, rather than using sophisticated scene representations, we evaluate simple features from the composition of “things” in the scene, from which we define query representations based on simple linguistic statements and simple block illustrations.

5.3 Things syntax

Starting from three object proposal methods [3, 75, 115], we consider their output locations in an image to hold a thing. The output locations are bounded with a box and we refer to them as windows. We use the bounding box window w to calculate the thing properties. For **horizontal position** and **vertical position** we use the center coordinates w_x and w_y of its window w . Thing **size** is approximated with the window width and height $w_s = w_w w_h$ whereas shape is measured by the **aspect ratio**

$$w_r = \begin{cases} 0.5(w_w/w_h) & \text{if } o_w \geq o_h ; \\ 0.5(w_h/w_w + 1) & \text{if } o_w < o_h, \end{cases} \quad (5.1)$$

which has a value of 0.5 for square objects, a value between $0 \leq w_r < 0.5$ for tall objects and a value $0.5 < w_r < 1$ for broad objects. We also measure the dominant **color** w_c of the thing window with eleven basic colors as defined by [121].

The accidental image sensor resolution influences thing window size and position. If an image is scaled by a factor f , the windows will be scaled accordingly. The window center will move to (fw_x, fw_y) and the window size will become $fw_w fw_y$. Window aspect ratio is invariant to the image resolution since $\frac{fw_w}{fw_h} = \frac{w_w}{w_h}$. We obtain resolution invariance for window position and size by normalizing with the image width I_w and height I_h as $\frac{w_x}{I_w}$ since $\frac{fw_x}{fI_w} = \frac{w_x}{I_w}$. We also investigated translation and scale invariance of thing windows, but we found that adding these invariants does not affect the discriminative potential of things windows.

Each thing window is represented by: horizontal position, vertical position, size, ratio and color, resulting in a 1×5 dimensional vector $\mathbf{w} = [w_x, w_y, w_s, w_r, w_c]$. The ensemble of all windows properties in an image forms the things syntax. If an image has n thing windows, represented as vectors $\{\mathbf{w}_i\}_{i=1}^n$, then the image things syntax is represented by stacking all windows vectors in a matrix

$$W^{n \times 5} = \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \vdots \\ \mathbf{w}_n \end{bmatrix} = \begin{bmatrix} w_{1x}, w_{1y}, w_{1s}, w_{1r}, w_{1c} \\ w_{2x}, w_{2y}, w_{2s}, w_{2r}, w_{2c} \\ \vdots \\ w_{nx}, w_{ny}, w_{ns}, w_{nr}, w_{nc} \end{bmatrix}. \quad (5.2)$$

5.3.1 Query by abstract linguistic statements

To formalize a scene image in linguistic statements, the things syntax needs grounding in natural language. Such a grounding can be obtained by quantizing the window properties of things to human understandable statements. Since we purposely ignore object names, we disregard nouns, and map window properties to adjectives (small, green, tall) and adverbs (left, middle, right). We chose three levels since humans tend to use three denotations for object properties. For example, horizontal position can be quantized to the three words (left, middle, right) and size can be mapped to (small, medium, large). In Figure 5.3 we summarize the set of rules we define to translate the horizontal position, vertical position, size and shape into a things vocabulary composed of adverbs and adjectives. We also map the color of the things windows on one of the eleven dominant color names [121].

Histogram representation. Mapping a continuous property of a thing window, such as its x-position w_x , to a single word such as “left” requires setting binning boundaries for quantization. We quantize each thing property separately and combine them in one histogram representation later. Each bin in the histogram represents an occurrence value for one combination of things properties, and one sentence like “Green large tall thing at bottom left” can be created. A straightforward solution for binning is to split the axis of each window property value in equal bins $[0, \frac{1}{3}, \frac{2}{3}]$. However, things windows may not be uniformly distributed, and thus frequent words may bias the description. On a separate holdout set we calculate simple statistics of things properties to obtain an equal probability of word occurrence. The holdout set can be any scene images, downloaded from internet for example, or by ignoring the annotations on an existing scene dataset, since we calculate statistics directly from the images and we do not need annotations. With a histogram h of three bins over the training images, the word bin boundaries are obtained by splitting the histogram in three parts with equal probability weight, i.e.,

Fig. 5.3: Binning boundaries for quantization of the continuous things properties: horizontal position, vertical position, size and shape, to a single word like *left*, *top*, *small* and *wide*. We quantize each thing property separately and combine them in one histogram representation later. We use eleven colors to represent the most dominant color in a thing window. Thus for each thing an abstract statement like *Green small wide thing at top left* can be created.

$\sum_0^{i_1} h(i) = \sum_{i_1+1}^{i_2} h(i) = \sum_{i_2}^1 h(i)$, where i_1 and i_2 indicate the bin separator. In Figure 5.3 we show word boundaries obtained in this manner on a holdout set. With binning boundaries we can quantize properties to words, and thus convert a set of things windows to a probability distribution over statements. If for horizontal position, vertical position, size and ratio we use 3 words, and for color we use 11 words [121], the histogram representation of the probability distribution over all statements combinations will be $3 \times 3 \times 3 \times 3 \times 11$ dimensional.

Query by abstract statements. Since the abstract statements are in a human understandable form, they allow us to search unseen scene classes. For example, if the unseen class is a bedroom, the formal statements allow, in principle, for a description of this class with “wide big brown thing in the center, small squared things left and right”. When abstract statements like this are provided for a scene, we can easily create a histogram representation. Creating a histogram representation from abstract statements is a reverse process of translating the things properties into abstract statements described above. Each statement stands for the properties of one thing, and is counted in one bin of the histogram representation. The value of each bin is a count on how many times the corresponding statement was used to describe a particular scene. At test time, we compare the histogram representation of scenes with the histogram representation of test images to compute probability score of a test image belonging to a scene class. This procedure is equivalent to the attribute representation of Lampert *et al.* [59]. The difference is that for unknown scenes as representations they use a distribution of an attributes occurrence, whereas we use a histogram of abstract statements. This allows us to use the same Direct Attribute Prediction (DAP) model for ranking test images by probability scores. The DAP-model assigns a test image to a most likely scene $z_1, \dots, z_L$ with

Fig. 5.4: Example of creating two things syntax representations. From a test image, things windows are generated automatically with a things proposals method. The dominant from eleven pre-defined colors is preserved from each window. The things properties are then either quantized into a histogram by binning, or a Fisher vector is created using pre-calculated GMMs. The histogram representation is used when searching scenes by abstract statements, and the Fisher vector representation is used when searching scenes by block illustrations.

$\operatorname{argmax}_l p(z = l|x) = \operatorname{argmax}_l \prod_{m=1}^M \frac{p(a_m^{z_l}|x)}{p(a_m^{z_l})}$, where $p(a_m^{z_l}|x)$ is the probability value of the attribute detector of $a_m^{z_l}$ applied on the test image x , and $p(a_m^{z_l})$ is the value of the m -th attribute occurring in scene z_l . In our case we have statement representations per scene instead of attributes. One important difference in this formulation is that $p(a_m^{z_l}|x)$ in [59] is calculated from attribute detectors, pre-learned on another independent manually annotated set. We calculate the probability distribution over statements of a test image x directly from its things properties. In this way the search process is simplified without the need for any learning.

5.3.2 Query by abstract block illustrations

To search by block illustration of a scene, it is not required for one to remember and draw the exact shape of things. As long as one can mimic the basic size, form, position and dominant color of things in the scene.

Fisher vector representation. To represent the block illustrations and the things syntax of images, we employ the popular Fisher vector encoding [96]. On a holdout set H , for each scene image $\{I|I \in H\}$ we compute its things syntax W_I^{nx5} . We merge

all $\{W_I^{n \times 5}, I \in H\}$, in one matrix W_H holding all things properties of the holdout set, and compute Gaussian Mixture Model (GMM) prototypes of W_H . A GMM with K components models the probability of all things windows properties $w \in W_H$, given the model λ by $P(w|\lambda) = \sum_{i=1}^k t_i g(w|\mu_i, \Sigma_i)$, where g is the Gaussian function and $\sum_{i=1}^k t_i = 1$. For a new block illustration I or image I , it's things properties are encoded with the derivative of the mean and variance to the GMM prototypes [96] as $\nabla \mu_k \log g_k(w) = \gamma_k(w) \left(\frac{w - \mu_k}{\sigma_k^2} \right)$, and $\nabla \sigma_k \log g_k(w) = \gamma_k(w) \left[\frac{(w - \mu_k)^2}{\sigma_k^3} - \frac{1}{\sigma_k} \right]$, where w is one row of the things syntax matrix W_I , μ_k denotes the mean, and σ_k^2 the diagonal of the covariance matrix Σ_k of Gaussian g_k , and $\gamma_k(w)$ are the soft assignment responsibilities of window w to Gaussian k . The final Fisher vector representation of the things syntax is created as a concatenation of $\nabla \mu_k \log g_k(w)$ and $\nabla \sigma_k \log g_k(w)$ for each Gaussian prototype k . Simply said, the procedure of the Fisher vector encoding of the things syntax is equivalent to the Fisher vector encoding of SIFT vectors [96]. The only difference in our case is that instead of using a 128-dimensional SIFT vector sampled from dense or salient points of an image, we use a 5-dimensional vector of individual thing window properties in an image. The rest of the procedure is identical.

Query by block illustrations. When block illustrations I for a scene are available, we can create a Fisher vector representation of the scene. This representation allows us to search images of an unseen scene class. We merge all window properties of block illustrations for a scene S into one things syntax matrix W_S of the scene. This matrix holds the layout of things within the scene through its things properties. Following the procedure described above, we create a Fisher vector representation from W_S . To retrieve test images of an unseen scene class, we first compute a Fisher vector representation of the image, and compare it to the scene Fisher vector representation. Different similarity measures and metric learning can be adopted to measure the similarity between the Fisher vector representation of a scene and the image. For now we simply use an Euclidian Distance to rank the test images. We summarize the process of creating things syntax representations in Figure 5.4.

5.4 Datasets

For the experiments we use both standard scene datasets, and abstract datasets we automatically generate from object annotations. Creating abstract datasets in this manner, examples shown in Figure 5.5, is different from when statements and block illustrations are provided directly by humans, see Figure 5.6 for comparison. However, this is the best approximation we can get for free. We describe the datasets below.

Fig. 5.5: Examples of block illustrations and abstract statements for three scene images, automatically generated from (a) selective search things proposals, and (b) human annotated things. The color is calculated from the dominant color in the bounding box window. Note that the human annotated objects are not complete, however, they are a reasonable and best available approximation.

SUN2012-14Scenes has 1,400 images in 14 classes we select from SUN397 [130], with an objective to have object annotations for at least 100 images per scene. We use this dataset for comparison between properties of human annotated things and properties of automatically generated things by [3, 75, 115].

Indoor67 [88] has 15,620 images from 67 indoor scenes like *bedroom*, *restaurant*, *wine-cellar*. All scenes have at least 100 images per category, with a provided 80-20 training-

Fig. 5.6: Block illustrations and abstract statements manually created. We asked three independent humans to provide statements and block illustrations for three scenes using only their memory of a scene, without looking at any image examples. Interestingly, we observe that sensible descriptions can be provided in this manner, even though this is not the way humans normally communicate or describe scenes in everyday life. Additionally, they are quite diverse for each individual.

test splits. For experiment 1, we use the training-test split as a given. For experiment 2 and 3, since we do not need any examples for training, we use the 80 images as test images for searching without examples. The remaining 20 images we use as a holdout set, where we ignore their class annotations.

SUNAttributes [86] contains 14,340 images hierarchically grouped in 3 levels, starting from fine-grained scenes in level three, growing into more general scene categories

in level two and one. Level one has 3 general classes: *indoor*, *outdoor natural* and *outdoor man made*. Level two has 16 high level scene categories like *shopping and dining* or *water, ice, snow*, and level three has all 717 fine-grained scenes like *airport ticket counter*, *bicycle racks*, *canyon*, with 20 images for each scene, which we use for testing.

Indoor67-AbstractStatements. Many of the Indoor67 images have object annotations provided with the LabelMe toolbox [95] in a shape of a polygon. We ignore the object names, consider them as nameless things and calculate their properties from a bounding box surrounding the polygon. As color we use the most dominant color in the bounding box, computed as in [121]. From the things properties we generate sentences for free as described in Section 5.3.1, and use them to generate 67 scene representations, in a form of a histogram, for each scene to query by.

Indoor67-AbstractBlocks. Similarly as in the AbstractStatements-67 dataset, we reuse the LabelMe annotations from Indoor67, to automatically generate abstract block illustrations. We ignore the objects type and appearance, and we use the position, size, ratio and the dominant color of the things properties to generate block illustrations. From the block illustrations we compute Fisher vector representations for each of the 67 scenes to query by.

SUN717-AbstractStatements. The SUNAttributes dataset also comes with object annotations from the LabelMe toolbox in the form of polygons. Similar as in the creation of Indoor67-AbstractStatements, we ignore the object annotations and generate bounding boxes around the polygons to hold things. For all image things we generate abstract statements. By grouping the abstract statements per image of all three levels from SUNAttributes, we generate 3, 16 and 717 scene representations of abstract statements respectively to query by.

SUN717-AbstractBlocks. We use the thing properties calculated from the object annotations on SUNAttributes to also generate abstract block illustrations. From the block illustrations we create 3, 16 and 717 scene class representations, in a Fisher vector form, for level one, two and three respectively.

Fig. 5.7: Heat maps per property of Kullback-Leibler (KL) divergence between scenes, for (a) selective search things and (b) human annotated things, with pointed maximum (red) and minimum (green) values. The lower the KL divergence score, the more similar the scenes are for that property. Interestingly, we see a similar pattern between the properties of selective search things and human annotated things, indicating the suitability of things proposal methods to approximate true things as annotated by humans.

	Horizontal	Vertical	Size	Ratio	Color
Objectness [4]	0.073	0.029	0.200	0.179	0.286
Selective search [115]	0.043	0.043	0.360	0.066	0.282
PRIMObjects [75]	0.058	0.028	0.230	0.061	0.294

Tab. 5.1: Comparison between things properties distribution of three things proposal methods and human annotated things on the SUN2012-14Scenes dataset, measured with the Kullback-Leibler divergence. The results are computed by averaging the KL divergence over all scene classes. The closest distributions are shown in bold. Each proposal method has a best approximation to the human annotated things for different property, each successful in their own right. When we compare this KL values with the maximum inter-class KL divergences of human things properties as reference numbers, 0.584 for horizontal, 0.248 for vertical, 0.651 for size, 0.292 for ratio and 6.068 for color, as shown in Figure 5.7, we conclude that the properties of things proposal methods come close to the properties of human annotated things.

5.5 Experiments

5.5.1 Comparing automatically proposed and manually annotated things

Do they behave similar? We first consider the question whether properties from things obtained with proposal methods are distributed similarly as things taken from manual things annotations. We generate things proposals with three recent methods: objectness [4], selective search [115] and prime object proposals (PRIM) [75], and we consider manually annotated things from the SUN2012-14Scenes dataset. Although the human annotations in SUN2012-14Scenes are not intended to be a complete ground truth of all things contained in a scene, they are the best approximation available to us. We compute the distributions of things properties per image by binning. We measure the distribution divergence of things proposals property Q to the human annotated thing property P with the Kullback-Leibler divergence $D_{KL}(P||Q) = \sum_i \ln \left(\frac{P(i)}{Q(i)} \right) P(i)$. The Kullback-Leibler divergence measures the expected loss of information when distribution Q is used to approximate the true distribution P . A low score represents a good fit.

Analysis. We present the mean Kullback-Leibler divergence scores over all scenes, between the things properties of proposal methods and human annotated things in Table 5.1. The lower the KL divergence score, the more similar the distributions are for that property. Results show that proposals reasonably approximate human annotation

Fig. 5.8: Comparing properties of human annotations and things proposals. The gray bars represent distributions of properties calculated from human annotated things, and colored lines show the distribution of properties calculated from things proposals. The numbers over the plot lines are the KL-divergence of human vs proposed things. In most cases the thing proposal distribution lines follow the shape of the gray bars of human annotated things. Thus, we conclude that the things proposals are a suitable approximation of real things in a scene.

statistics with an expected average loss of only 5-10%, except for *size* and *color*. The difficulty in estimating the color is due to its computation over the full window, i.e., it is a joint statistic over all four dimensions and thus affected by errors in any of them. If we take as reference numbers the maximum inter-class KL divergences of human things properties, with 0.584 for horizontal, 0.248 for vertical, 0.651 for size, 0.292 for ratio and 6.068 for color, as shown in Figure 5.7, we conclude that the properties of things proposal methods come close to the properties of human annotated things. In Figure 5.7 we also show the minimum values and heat maps of the KL divergence between scene classes per property, for both human annotated things and selective search things. Interestingly, we see a similar pattern between the properties of selective search things and human annotated things. For color, the same scenes have max/min values. For horizontal position of selective search things, *highway* differs most from the rest of the scenes.

We believe this is so because a highway scene usually has things lined horizontally, like the highway, sky, ground, which is not the case within the other 13 scenes. However, we observed that the human annotated things of highway are mostly on the cars on the highway, missing the horizontally aligned things in the scene. Interestingly, some of the most similar scenes are, for horizontal: *hotel room - waiting room*, size: *building facade - skyscraper* and aspect ratio: *waiting room - bathroom*. Most distinctive are, for horizontal: *highway - forest broadleaf*, vertical: *skyscraper - street* and aspect ratio: *highway - skyscraper*. The examples show that the things properties capture scene information, the similar scenes are indeed close, and the dissimilar ones are far for a given things property. Investigating the matter further, we show the distribution curves for the properties of human annotations and all three things proposals for four scenes in Figure 5.8. All proposed things follow the distributions of the horizontal and vertical position properties close to the horizontal and vertical position properties of human annotated things. Object size is overestimated, which may be due to the proposals used in object recognition where larger objects are the norm. Aspect-ratio is coming close reasonably well, except objectness, which tends to generate overly long or broad windows. The things proposals are not perfect, but we conclude that they are a suitable approximation of real objects in a scene. As new and better object proposal methods are being introduced [21, 92], we can expect that the approximation will only improve.

Are they as discriminative? Next we study whether the object proposal window properties are as discriminative as human annotated things. We use the SUN2012-14Scenes dataset with 50 images for training and 50 for testing, and we use the Indoor67 dataset with 67 indoor scene classes using the author suggested splits. As representation we rely on the things syntax with the Fisher vector, calculated using a 1,024 component GMMs. For learning we train one-vs-rest Support Vector Machine with the RBF kernel, and we evaluate with accuracy.

Results. When using as representation the things syntax of human annotations encoded with the Fisher vector, on SUN2012-14Scenes we achieve an accuracy of 57.6%. The Fisher vectors of things syntax from automatically generated things proposals with selective search come close with an accuracy of 54.0%, with PRIM proposals 52.1%, and 32.2% for objectness things. For comparison, On 14 classes the random recognition rate is 7.14%. The results show that the things syntax has a discriminative potential, and the discriminative potential of things proposals, especially the one from selective search, approximates the discriminative potential of human annotated things reasonably. Thus, we use things proposals from selective search in the rest of our experiments.

On Indoor67 the things syntax of selective search encoded with the Fisher Vector achieves an accuracy of 25.8%. The accuracy with GIST [83] is 29.6%, and when GIST is combined with the things syntax the accuracy reaches 38.9%. This shows that the things syntax is orthogonal to GIST, capturing new information. We also recognize the ad-

vances in deep learning and the power of the features of a Convolutional Neural Network (CNN) [56, 137]. For example, when using features from one layer before the last of GoogleNet [111], the accuracy on Indoor67 is 66.8%. When we combine these features with the things syntax, the result improves slightly to 68.5%. This shows that the things syntax also captures some new information not learned by the CNN. Overall, we show that the things syntax has some discriminative power. Yet, the main aim of the things syntax is not to improve the scene classification, the CNN features do a much better job at it. The aim and the added benefit is to generate an example-free abstract description without the need for learning, where all the others, including deep learning, use examples to learn human understandable representations.

Overall, in this experiment we have shown that the things syntax holds discriminative information, that the information from proposed things comes close to human annotated things. We investigate the abstract descriptions from the things syntax in the next experiments.

5.5.2 Query by abstract statements

In this experiment we investigate the use of abstract statements to describe a scene, like “blue wide thing at top center, green medium tall thing at bottom left”, as described in Section 5.3.1, and their descriptive potential in a retrieval setting. We use statements from the SUN717-AbstractStatements and Indoor67-AbstractStatements datasets, and test on the images from SUNAttributes and Indoor67. We investigate three parameters of the abstract statements for scene retrieval, evaluate with mean average precision (MAP), and we summarize the results in Figure 5.9.

Influence of binning. First, we investigate the influence of the number of bins for horizontal position, vertical position, size and aspect ratio. Rather than restricting the binning to three annotations for objects, like (*left, middle, right*) or (*top, center, bottom*), we can easily add more bins to represent the statements, *e.g.* (*most-left, left, center, right, most-right*). The number of bins for color is always fixed to 11, as defined in [121]. We test on images of all three levels from the SUNAttributes dataset and Indoor67. In Figure 5.9 (a), we show retrieval results with bin sizes varying from 3 to 11. We observe that on more general categories, like level one and two of SUNAttributes, using more bins/more precise descriptions of the things properties does not help, whereas, for fine-grained scenes it results in better retrieval MAP. For example on the SUNAttributes 717 scenes the mean average precision grows from 2.25% to 4.06%, and on Indoor67 from 5.24% to 7.04%. We conclude, when more precise statements are available for fine-grained scenes, the better the scene retrieval will be.

Fig. 5.9: Result of investigating abstract statements for retrieving scenes: (a) influence of the number of bins used to quantize the things syntax into histograms and statements, (b) descriptive influence per property on Indoor67, (c) influence of abstract statements quality by adding noise to object annotations on Indoor67.

Influence per property. We investigate how well the thing properties perform independently. In this way, we generate a scene representation for each property, by having statements composed of one word only, as “left” things, or “tall” things, or “small” things. We create scene representations from AbstractStatements-67 and evaluate on Indoor67. The results are shown in Figure 5.9 (b). Ratio and color are the best performing properties. We assume this is because they are both scale invariant, whereas the other properties are not. For example, if things are captured closely or from far away, the aspect ratio and color will be consistent, whereas the position and size of things in the image will change.

Influence of statements quality. The abstract statements we generate in the datasets AbstractStatements-67 and the SUN717-AbstractStatements, are by using human annotated things. To approximate a more realistic scenario where the abstract statements will be provided directly from users, we add noise to the bounding box things annotations. We generate the noise from a Gaussian distribution with mean 0, and standard deviations of [2, 4, 6, 8, 10, 15, 20] pixels, and we scale the images to a maximum dimension of 320px. We investigate adding noise on Indoor67-AbstractStatements and test on Indoor67. We add noise on each property separately, as well as on all properties together and present results in Figure 5.9 (c). As expected, adding noise has some effects the retrieval results. For example when adding noise on all thing properties together up to 10px, the MAP goes down from 5.24% to 4.90%, and for 20px noise it goes down to 4.64%. Interestingly adding noise up to 6px on the things width, the results even improve marginally. This happens when a mistake is made with the binning, the noise acts

Fig. 5.10: Result of investigating block illustrations for retrieving scenes: (a) influence of the number of Gaussian Mixture Model (GMM) prototypes used in encoding the block illustrations with the Fisher vector, (b) descriptive influence per property on Indoor67, (c) influence of Fisher vector representations for block illustrations generated from human annotations, by adding noise to the annotations on Indoor67.

as a correction mechanism. Overall, we conclude that the quality of the object location can tolerate displacement without hurting the scene retrieval substantially.

5.5.3 Query by block illustrations

In the third experiment we investigate to what extent we can retrieve an unseen scene using block illustrations of things. As a representation of block illustrations and things syntax from test images we use Fisher vector encoding, as described in Section 5.3.2. We use Fisher vector representations per scene, created from block illustrations of SUN717-AbstractBlocks and Indoor67-AbstractBlocks datasets, to query scenes in test images from SUNAttributes and Indoor67. We investigate three parameters of the Fisher representation in a scene retrieval setting, we evaluate with mean average precision, and summarize the results in Figure 5.10.

Influence of GMM components. The Fisher vector encoding depends on the number of GMM prototypes, therefore, we investigate their influence. We compute the GMM prototypes on a holdout set of Indoor67. In Figure 5.10 (a) we show results on SUNAttributes, including all three levels, and Indoor67. As expected, since more prototypes capture a better variation of the windows properties, they result in richer Fisher vector representation and higher MAP. On the third level of SUNAttributes the MAP grows from 5.37% for 128 prototypes, to 14.75% for 4,069 prototypes. On Indoor67

	MIT Indoor	SUN Attributes		
	<i>Scenes(67)</i>	<i>Indoor/Outdoor(3)</i>	<i>SceneCategories(16)</i>	<i>Scenes(717)</i>
Query by object attributes	2.06%	34.85%	6.66%	0.33%
Query by object bank	1.96%	45.92%	8.95%	0.37%
Query by classes	2.96%	60.13%	13.27%	0.82%
Query by 1,000 objects CNN-layer	10.15%	57.81%	15.43%	2.09%
Query by abstract statements	5.25%	50.62%	12.40%	2.25%
Query by block illustrations	12.39%	47.66%	12.49%	10.94%
Query by statements and blocks	12.62%	56.67%	17.10%	9.64%

Tab. 5.2: Comparison of scene retrieval MAP results, using query by semantic representations and query by our abstract representations, statements and block illustrations. Surprisingly, by using abstract representations of simple things properties and no learning at all, we manage to retrieve scenes reasonably. The abstract representations in some cases show even better retrieval performance than the semantic representations, which require learning beforehand.

similar improvement fashion of the MAP is followed, from 8.99% to 15.71%. The results grow slowly after 1,024 prototypes. Therefore we use 1,024 component GMMs in the rest of the experiments.

Influence per property. We investigate scene retrieval using query by block illustrations per property. To do so, we generate scene Fisher representations from only one window property at a time from the Indoor67-AbstractBlocks dataset and test on the Indoor67 dataset. We show the results in Figure 5.10 (b). The MAP results follow similar trend as in the scene retrieval setting with abstract statements, with ratio being most descriptive. Color shows a different behavior, having a lower MAP from size and ratio. We believe this is so, because in the histogram representations of statements we use exactly 11 bin values for color, thus being more precise, whereas the Fisher vector encodes the color with 1024 GMMs.

Influence of block illustrations quality. In the Indoor67-AbstractBlocks dataset, we generate the block illustrations from human annotated things. In order to approximate a more realistic drawing scenario, we add Gaussian noise to the human annotations and summarize the results in Figure 5.10 (c). If we add noise only to the position, or only to the width or height of the windows, the MAP is moderately affected up to a deviation of 8 pixels. If noise is added to all the window properties at once the MAP drops more rapidly. We conclude that abstract block drawings can be used to retrieve scenes when the boxes in the block illustrations deviate up to around 8 pixels from the true objects.

Fig. 5.11: Confusion matrices on the Indoor67 dataset of our abstract representations and other semantic representations. In the confusion matrixes of the abstract representations, the diagonal line can be clearly seen, depicting true positives for all scene classes. In the semantic representations most scenes get misclassified into few scenes that have an accidental overlap with a semantic class, which can be seen with the strong vertical red line responses in the confusion matrices. Overall, when there is no direct semantic information available aimed for the scope of the unseen classes, abstract descriptions of things syntax are a good alternative representation.

5.5.4 Abstract vs Semantic

In the last experiment we compare our abstract representations, query by statements and query by block illustrations, to four other representations involving semantic descriptions of object identities [64], object attributes [25], general classemes categories trained from Flickr [8] and 1000 ImageNet objects [94]. Each method has different number of semantic classes, object bank with 177, object attributes with 67, classemes with 2,659, and ImageNet with 1,000 semantic classes. For all methods we use the available software online, provided by the authors, to get semantic scores representation on the test images. For ImageNet we get scores from the output layer of 1,000 classes from an in-house developed Convolutional Neural Network inspired by [137] and trained on ImageNet. We use the same principle for all semantic methods to create a semantic

Fig. 5.12: Top retrieved images for the best performing scene of six methods on Indoor67, and their average precision. Interestingly, since the abstract statements and block illustrations capture the properties of things layout, the best performing scene is *corridor*, which indeed has a specific things layout. The semantic methods use appearance features and their best performing scene depends on the performance of the semantic classes within. We conclude, the things syntax captures the scene things layout, and it can be used to retrieve scenes with abstract descriptions which require no learning at all.

representation per scene to query by. We rely on object annotations from LabelMe [95] on both datasets, Indoor67 and SUNAttributes. For example, classemes has 2,659 semantic classes. We count how often a semantic class from classemes was annotated by humans in images from a scene class. This gives us a distribution histogram of the classemes objects per scene, which we L1 normalize and use as a scene representation to query by. On a test image we run the classemes software as provided by the authors, and get classemes scores. To calculate a ranking score of a test image for a scene, we use the DAP model as proposed by [59] between the classemes scene representation and the classemes scores of the test image. The same procedure we repeat for all semantic methods. We realize we use these methods in a different way than they were originally intended to be used. The reason we choose these works is because the semantic information they provide is not directly aimed to describe scenes, like the case of provided scene attributes in [59]. Our goal is to show that when semantic information is used, it is beneficial to correlate the semantics to the category of the test classes, like annotations of animal attributes for animal classes [58], or scene attributes for scene classes [59]. We believe that when no direct semantic information to the category of the test classes is available, abstract descriptions are a good alternative, since they generalize well to any unseen category. We show this on scene categories.

We present mean average precision results in Table 5.2. Since the object attributes [25] are intended for a different purpose, they do not generalize well to scenes. Object bank [64] is successful when one of its objects has an (accidental) relationship with the scene of interest. Unfortunately, most scenes have a low average precision as an overlap with the objects in the bank is missing. Classemes [8] detectors are trained from weakly supervised images, but their representation is more rich, having over 2K semantic classes, and results in better MAP in general. Similar to object bank, if the objects in the 1,000 ImageNet objects overlap with the scenes, it results in good average precision, else it fails. In a scenario where there is no semantic information for the intended category available, our abstract representation is more general, leading to reasonable accuracy values in almost all settings. This can be clearly seen in Figure 5.11, where we show confusion matrices on the Indoor67 dataset of our abstract representations and other semantic representations. When there is a good overlap of the scene class with the object, like for example *closet* which is an object class in ImageNet and a scene class in Indoor67, then there is a strong response as seen in the red lines in Figure 5.11. We show top retrieved images of the best performing scene for all methods on the Indoor67 dataset in Figure 5.12. Interestingly, since the abstract statements and block illustrations capture the properties of things layout, the best performing scene is *corridor*, which indeed has a specific things layout. The other methods use appearance features, and the best performing scene depends on the performance of their semantic classes. Overall we conclude that when there is no direct semantic information available for unseen classes

in a scene retrieval setting, abstract descriptions of things syntax are a good alternative. An added benefit of our work is that it does not require any labeled examples to build the representation. The abstract things syntax is directly generated from the image itself.

5.6 Conclusions

In this chapter we open up further understanding of the rules composing the visual world around us, the potential to combine the objects layout information, and the recognition of scenes without the need to keep semantics like object types or attributes. We show that next to object types, there is another source of information defining what makes a scene. Object types are unknown in abstract paintings, architecture inspiration, microscopic and cosmic observation, while their “things” composition is well observable. Thus, to describe images of these compositions in an abstract manner is inevitable. We give preference to the composition of “things” as an indicator for the type of scene. We start from “things” as defined by modern object proposals, and investigate their immediately observable features: position, size, aspect ratio and color. We name the ensemble of things properties as things syntax, and we investigate its effectiveness to represent and retrieve a scene. From four experiments we conclude, (1) the distribution of thing-features from proposal methods approximates the distribution of thing-features from human annotated things closely. We investigate and analyze the discriminative potential and properties of the things syntax when translated into (2) abstract language statements and (3) abstract block illustrations for scene retrieval. In both cases we show that things aspect ratio is the most informative property, also that scenes can still be retrieved if their things deviate up to 8 pixels from the true objects, and by providing more precise abstract statements, i.e. more bins in the histogram representations, or by using more GMM prototypes in the Fisher vector representations of block illustrations, the retrieval results on fine-grained scenes improves. At last, (4) we compare the abstract things syntax representations with four other semantic representations which are not directly aimed for scenes. We show that when there is an accidental overlap of the semantic classes with the scene class, using semantics is beneficial. However, when there is no accidental overlap, the abstract descriptions of things syntax are a good alternative, and in some cases show even better retrieval performance than the semantic representations which require learning beforehand. Overall and surprisingly, we show that even though we use the simplest of features from things layout, we can still retrieve scenes reasonably well, and with an additional benefit that we do not require any learning examples.

Summary and Conclusions

6.1 Summary

This thesis contributes to learning machines what is in an image by avoiding direct manual annotation as training data. We either rely on tagged data from social media platforms to recognize concepts (Chapter 2 and 3), or on objects semantics and layout to recognize scenes (Chapter 4 and 5). Before we come to general conclusions we give a short overview of all chapters individually.

In chapter 2 we show that learning video concept detectors from social media sources, such as Flickr images and YouTube videos, has the potential to address a wide variety of concept queries for video search. While the potential has been recognized by many, and progress on the topic has been impressive, we argue that key questions crucial to know *how to learn effective video concept detectors from social media examples?* remain open. As an initial attempt to answer these questions, we conduct an experimental study using a video search engine which is capable of learning concept detectors from social media examples, be it socially tagged videos or socially tagged images. Within the video search engine we investigate three strategies for positive example selection, three negative example selection strategies and three learning strategies. The performance is evaluated on the challenging TRECVID 2012 benchmark consisting of 600 hours of Internet video. From the experiments we derive four best practices: (1) tagged images are a better source for learning video concepts than tagged videos, (2) selecting tag relevant positive training examples is always beneficial, (3) selecting relevant negative examples is advantageous and should be treated differently for video and image sources, and (4) learning concept detectors with selected relevant training data before learning is better than incorporating the relevance during the learning process. The best practices

within our video search engine lead to state-of-the-art performance in the TRECVID 2013 benchmark for concept detection without manually provided annotations.

In chapter 3 we start from the fact that there are millions of users who tag multimedia content, generating a large vocabulary of tags. Some tags are frequent, while other tags are rarely used following a long tail distribution. For frequent tags, most of the multimedia methods that aim to automatically understand audio-visual content, give excellent results. It is not clear, however, how these methods will perform on rare tags. In this chapter we investigate what social tags constitute the long tail and how they perform on two multimedia retrieval scenarios, tag relevance and detector learning. We show common valuable tags within the long tail, and by augmenting them with semantic knowledge, the performance of tag relevance and detector learning improves substantially.

In chapter 4 we aim to recognize scenes in images without using any scene images as training data. Different from attribute based approaches, we do not carefully select the training classes to match the unseen scene classes. Instead, we propose a pooling over ten thousand of off-the-shelf object classifiers. To steer the knowledge transfer between objects and scenes we learn a semantic embedding with the aid of a large social multimedia corpus. Our key contributions are: we are the first to investigate pooling over ten thousand object classifiers to recognize scenes without examples; we explore the ontological hierarchy of objects and analyze the influence of object classifiers from different hierarchy levels; we exploit object positions in scene images and we demonstrate a new scene retrieval scenarios with complex queries. Finally, we outperform attribute representations on scene classes from two challenging datasets, SUNAttributes and Places2.

In chapter 5 we propose to represent a scene as an abstraction of “things”. We start from “things” as generated by modern object proposals, and we investigate their immediately observable properties: position, size, aspect ratio and color, and those only. Where the recent successes and excitement of the field lie in object identification, we represent the scene composition independent of object identities. We make three contributions in this work. First, we study simple observable properties of “things”, and call it things syntax. Second, we propose translating the things syntax in linguistic abstract statements and study their descriptive effect to retrieve scenes. Thirdly, we propose querying of scenes with abstract block illustrations and study their effectiveness to discriminate among different types of scenes. The benefit of abstract statements and block illustrations is that we generate them directly from the images, without any learning beforehand as in the standard attribute learning. Surprisingly, we show that even though we use the simplest of features from “things” layout and no learning at all, we can still retrieve scenes reasonably well.

6.2 Conclusions

The goal of this thesis is to investigate learning to search for images without using annotations. Since manual annotation is an expensive process, we have demonstrated that concepts detectors can be learned using tagged examples from social media platforms. We show that using tagged images and videos directly as ground truth for learning can be problematic because of the noisy nature of tags. To this end, through extensive experimental analysis, we recommend to calculate the relevance of tags, and select only relevant positive and relevant negative examples for learning. Following the findings that some concepts appear rarely as tags in social media platforms, we propose to use semantic knowledge from an ontology to improve calculating tag relevance and to enrich training data for the rare tags.

When searching images of a particular scene, instead of using annotated scene images, we show that with object classifiers we can reasonably well recognize scenes. We exploit 15,000 object classifiers trained with a convolutional neural network. Since not all objects can contribute equally in describing a scene, we show that pooling only the 100 most prominent object classifiers per image is good enough to recognize its scene. Furthermore, we go to the extreme of recognizing scenes by removing all object identities. We refer to the most probable positions in images to contain objects as things. We show that the ensemble of things properties, size, position, aspect ratio and prominent color, and those only, can discriminate scenes. The benefit of removing all object identities is that we also eliminate the learning of object classifiers in the process. We show that the layout of things only can help in recognizing scenes.

Overall, this thesis presents alternative ways to learn what concept is in an image or what scene it belongs to, without using annotated data, for the goal of image search. It investigates new approaches of bridging the semantic gap between images and text, and contributes in learning machines to recognize the visually depicted environment around us one step further.

Appendix

A.1 SocialZap

CATCH-UP ON INTERESTING TELEVISION FRAGMENTS DISCOVERED FROM SOCIAL MEDIA

In the appendix we present a *SocialZap* demo system, which uses the benefits of learning concept detectors directly from social media platforms. SocialZap is a multimedia search engine that finds the most interesting fragments, zap points, in a television broadcast based on microblog posts and socially tagged photos. The main novelty of SocialZap is the fully-automatic transfer of the learned viewers interest from textual posts to the visual channel, without the need for any manual effort in the process. Once SocialZap finds the zap points, users can easily browse through a television broadcast and directly watch the interesting fragments. Thus, SocialZap adds social experience to watching television¹.

¹Published in *the Proceedings of ACM International Conference on Multimedia Retrieval*, 2014 [50].

Fig. A.1: Video zap points, corresponding to the time points within a television broadcast at which the visual appearance of concepts designated by common terms extracted from tweets, have been detected.

A.1.1 Introduction

Existing web services [2] allow viewers to watch missed television broadcast later on the web. However, a system that directly suggests the most interesting fragments to watch, based on social media, is, to the best of our knowledge, non existing. In our SocialZap demo, we analyze data from social media to suggest interesting concepts, zap points, in television broadcast. SocialZap uses textual information from Twitter posts related to a television broadcast of interest, which provide a rich source of information of what viewers find interesting, see Figure A.1. The challenge we face is the temporal mismatch between the moment that the user tweets about a concept and the moment at which it appears in the television broadcast. The tweet-time can radically differ from the appearance-time as viewers either anticipate appearances or continue to tweet about topics that have previously appeared. To transfer the found concepts of interest from the textual to the visual modality, we use state of the art approach [108] which learns concept detectors from social media. We collect Flickr images tagged with the concept names, and we select the most reliable ones as training data for learning. Thus, the proposed system relies totally on social media data, without any manual annotations in the process. The SocialZap pipeline is illustrated in Figure A.2, we describe it in more detail in the following section.

Fig. A.2: SocialZap pipeline. Tweets for a particular television broadcast are gathered and visual detectors for the most common terms in the tweets are then learned to find the visual fragments of those concepts as zap points.

A.1.2 System overview

In order to automatically provide zap points to users, SocialZap obtains interesting (visual) topics about the television broadcast and then learns visual concept models to recognize those concepts in an unseen video. SocialZap exploits social information from (1) Twitter posts related to a television broadcast of interest to find the most common terms as concepts of interest and (2) tagged Flickr images to learn visual concept detectors. SocialZap is composed of different modules, which communicate with each other through a common API in a central server, see Figure A.3. We describe the separate modules in the following sections.

Textual Concepts and Social Media

Assuming that viewers write posts about what they find interesting, we exploit social information about the television broadcast gathered from Twitter with the Tweet Analyzer module of SocialZap. The posts are collected by issuing queries with automatically constructed hashtags from the broadcast show name, as *#dwdd* shown in Figure A.2. The tweets that are posted during the time of the broadcast are considered to be associated with the broadcast. Additionally we exploit closed captions associated with the

Fig. A.3: SocialZap is composed of different modules, which communicate with each other through a common API in a central server. The Tweet and the Caption Analyzer are responsible for finding common terms. The Video Miner learns video concept detectors for the depicted terms, using training data only from Social Media. The Video Indexer finds the zap points by applying the learned concept detectors on television broadcasts.

television broadcast with the Caption Analyzer module. Closed captions are beneficial to collect visually depicted entities if there are not enough found in the tweets, and also to denoise tweet-terms by assuring that they also occur at least once in the captions. After obtaining tweets and closed captions, we translate them to English and perform stopword removal. This preprocessing step is necessary to build the basis for later retrieving more accurate training data for selected visual concepts. Each of the terms contained in tweets and closed captions gets assigned a reranking score in the interval $[0; 1]$, which represents both the importance (i.e., popularity) and (visual) depictedness of these terms with respect to the harvested tweets and the (audiovisual) content of the corresponding broadcast. The importance of a term is obtained by building language models [17, 138] for each television broadcast. Terms that are specific to a television broadcast and at the same time mentioned relatively frequently in Twitter posts receive a higher relevance score. In order to quantify the depictedness of a term, we apply the generic approach from [60]. The overall score of a term is modelled by combining importance- and depictedness-scores in a linear fashion.

Visual Concepts and Social Media

To transfer viewers interest from the textual modality to visual concepts, we need training data to learn visual concept detectors. In order to detect any possible video concept, a promising line of research is to automatically collect training examples from the web, where many socially tagged videos and images exist [67, 117]. In a recent study, we have shown that tagged images are a better training source than tagged videos for learn-

Fig. A.4: User Interface of SocialZap. Once the user selects a show, all found zap points are listed on the right side of the video player, ranked by importance. The user can select any zap point of interest. Once selected the interface starts playing the video at the frame where the zap point occurs.

ing video concept detectors [51]. Therefore we use Flickr as source to collect tagged images. Due to variation between users regarding the use of social tags, simply treating all tagged images as positive examples may be problematic [67]. In order to preserve only good examples, we first calculate tag relevance scores per image. We use the multi-feature variant of the neighbor voting algorithm [67]. The images are then ranked in terms of their estimated tag relevance scores. Then, we preserve a top ranked proportion of the examples [51], as shown in Figure A.2. As negative examples, we employ negative bootstrap [68], which finds relevant negative examples in an iterative manner. We follow a state-of-the-art bag of visual codes pipeline for feature extraction [120], and learn concept detectors using one-vs-all SVM [74]. We call this module of the framework Video Miner. The described approach implemented in SocialZap showed best performance on the TRECVID 2013 semantic indexing with no annotation task [108], with best score for 31 out of 38 concepts.

Once we obtain the learned concept detectors, we apply them on the broadcasted video frames. For each frame we get confidence scores on how likely the concept appears in the frame. The frames with highest scores are suggested as zap points of the television broadcast. This module is the Video Indexer.

User Interface

We created a simple user interface that allows users to catch-up on the interesting fragments of a broadcast, see Figure A.4. The interface provides an option to select a specific television show. Once selected, SocialZap lists the most common terms found in the social media posts with respect to the selected broadcast. When the user chooses one of the terms, the video player directly plays the video segment where the significance score for that concept is highest. If some term occurs more than once in the video, all zap points for that term will be listed, as in the case with the term politics shown in Figure A.4. As it can be seen from the figure, the concept detector of elections sets the zap point correctly at a relevant frame where this concept appears. However since concept detection is a challenging task and does not always give precise results, we found that timing of tweet and caption terms can help for finding correct zap points.

A.1.3 Conclusion

In this appendix we present the SocialZap system, which finds interesting fragments in television broadcasts based on social media. With SocialZap the viewer's interest is depicted from Twitter and transferred into the visual channel as zap points. The system provides an easy way to browse through the zap points of the television broadcast, adding a new social dimension to the television experience.

Bibliography

- [1] Flickr statistics <http://bit.ly/24yClb2>.
- [2] Uitzendinggemist, online video portal of the Dutch public broadcasters <http://www.uitzendinggemist.nl>.
- [3] B. Alexe, T. Deselaers, and V. Ferrari. What is an object? In *CVPR*, 2010.
- [4] B. Alexe, T. Deselaers, and V. Ferrari. Measuring the objectness of image windows. *IEEE Trans. on PAMI*, 34(11):2189–2202, 2012.
- [5] S. Antol, C. L. Zitnick, and D. Parikh. Zero-Shot Learning via Visual Abstraction. In *ECCV*, 2014.
- [6] L. Ballan, M. Bertini, A. Del Bimbo, and G. Serra. Enriching and localizing semantic tags in internet videos. In *MM*, 2011.
- [7] T. L. Berg, A. C. Berg, and J. Shih. Automatic attribute discovery and characterization from noisy web data. In *ECCV*, 2010.
- [8] A. Bergamo and L. Torresani. Classemes and other classifier-based features for efficient object categorization. *IEEE Trans. on PAMI*, 36(10):1988–2001, 2014.
- [9] I. Biederman. On the semantics of a glance at a scene. *Perceptual Organization*, (8):213–263, 1981.
- [10] I. Biederman. Recognition-by-components: A theory of human image understanding. *Psychological Review*, 94:115–147, 1987.
- [11] I. Biederman, R. J. Mezzanotte, and J. C. Rabinowitz. Scene perception: detecting and judging objects undergoing relational violations. *Cognitive psychology*, 14(2):143–177, 1982.
- [12] D. Borth, R. Ji, T. Chen, T. Breuel, and S.-F. Chang. Large-scale visual sentiment ontology and detectors using adjective noun pairs. In *MM*, 2013.
- [13] S. Butler, J. Gross, and H. Hayne. The effect of drawing on memory performance in young children. *Developmental Psychology*, (4):597–608, 1995.
- [14] S. Cappallo, T. Mensink, and C. G. M. Snoek. Image2emoji: Zero-shot emoji prediction for visual media. In *MM*, 2015.
- [15] S.-F. Chang, D. Ellis, W. Jiang, K. Lee, A. Yanagawa, A. C. Loui, and J. Luo. Large-scale multimodal semantic concept detection for consumer video. In *MIR*, 2007.
- [16] S. Clinchant and F. Perronnin. Textual similarity with a bag-of-embedded-words model. In *ICTIR*, 2013.
- [17] S. Cronen-Townsend, Y. Zhou, and W. B. Croft. Predicting query performance. In *SIGIR*, 2002.
- [18] V. Delaitre, D. Fouhey, I. Laptev, J. Sivic, A. Gupta, and A. Efros. Scene semantics from long-term observation of people. In *ECCV*, 2012.
- [19] 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.
- [20] C. Doersch, A. Gupta, and A. A. Efros. Mid-level visual element discovery as discriminative mode seeking. In *NIPS*, 2013.

- [21] P. Dollár and C. L. Zitnick. Fast edge detection using structured forests. *IEEE Trans. on PAMI*, 37(8):1558–1570, 2015.
- [22] I. Endres and D. Hoiem. Category independent object proposals. In *ECCV*, 2010.
- [23] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *Int. J. of Computer Vision*, 111(1):98–136, 2015.
- [24] J. Fan, Y. Shen, N. Zhou, and Y. Gao. Harvesting large-scale weakly-tagged image databases from the web. In *CVPR*, 2010.
- [25] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *CVPR*, 2009.
- [26] A. Farhadi, M. Hejrati, M. A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth. Every picture tells a story: Generating sentences from images. In *ECCV*, 2010.
- [27] R. Fergus, H. Bernal, Y. Weiss, and A. Torralba. Semantic label sharing for learning with many categories. In *ECCV*, 2010.
- [28] D. A. Forsyth, J. Malik, M. M. Fleck, H. Greenspan, T. Leung, S. Belongie, C. Carson, and C. Bregler. Finding pictures of objects in large collections of images. Technical report, University of California, Berkeley, 1996.
- [29] J.-M. Geusebroek and A. W. M. Smeulders. A six-stimulus theory for stochastic texture. *Int. J. of Computer Vision*, 62(1-2):7–16, 2005.
- [30] A. L. Ginsca, A. Popescu, B. Ionescu, A. Armagan, and I. Kanellos. Toward an estimation of user tagging credibility for social image retrieval. In *MM*, 2014.
- [31] Y. Gong, L. Wang, R. Guo, and S. Lazebnik. Multi-scale orderless pooling of deep convolutional activation features. In *ECCV*, 2014.
- [32] S. Gould, R. Fulton, and D. Koller. Decomposing a scene into geometric and semantically consistent regions. In *CVPR*, 2009.
- [33] M. R. Greene. Statistics of high-level scene context. *Frontiers in Psychology*, 4(777), 2013.
- [34] C. Gu, J. Lim, P. Arbeláez, and J. Malik. Recognition using regions. In *CVPR*, 2009.
- [35] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv:1512.03385*, 2015.
- [36] M. Heikkila, M. Pietikainen, and C. Schmid. Description of interest regions with local binary patterns. *Pattern Recognition*, 42(3):425–436, 2009.
- [37] G. Heitz and D. Koller. Learning spatial context: Using stuff to find things. In *ECCV*, 2008.
- [38] Y. Hu, M. Li, and N. Yu. Multiple-instance ranking: Learning to rank images for image retrieval. In *CVPR*, 2008.
- [39] S. J. Hwang and K. Grauman. Learning the relative importance of objects from tagged images for retrieval and cross-modal search. In *Int. J. of Computer Vision*, 2012.
- [40] M. Jain, J. van Gemert, T. Mensink, and C. G. M. Snoek. Objects2action: Clas-

- sifying and localizing actions without any video example. In *ICCV*, 2015.
- [41] V. Jain and M. Varma. Learning to re-rank: query-dependent image re-ranking using click data. In *WWW*, 2011.
- [42] D. Jayaraman and K. Grauman. Zero-shot recognition with unreliable attributes. In *NIPS*, 2014.
- [43] W. Jiang, C. V. Cotton, S.-F. Chang, D. Ellis, and A. C. Loui. Short-term audio-visual atoms for generic video concept classification. In *MM*, 2009.
- [44] Y.-G. Jiang, J. Yang, C.-W. Ngo, and A. Hauptmann. Representations of keypoint-based semantic concept detection: A comprehensive study. In *IEEE Trans. on Multimedia*, 2010.
- [45] T. Joachims. Optimizing search engines using clickthrough data. In *SIGKDD*, 2002.
- [46] M. Juneja, A. Vedaldi, C. Jawahar, and A. Zisserman. Blocks that shout: Distinctive parts for scene classification. In *CVPR*, 2013.
- [47] L. S. Kennedy, S.-F. Chang, and I. V. Kozintsev. To search or to label?: predicting the performance of search-based automatic image classifiers. In *MIR*, 2006.
- [48] A. Khosla, N. Jayadevaprakash, B. Yao, and L. Fei-Fei. Novel dataset for fine-grained image categorization. In *CVPR*, 2011.
- [49] J. Kim and V. Pavlovic. Attribute Rating for Classification of Visual Objects. In *ICPR*, 2012.
- [50] S. Kordumova, C. Kofler, D. C. Koelma, B. Huurnink, B. Freiburg, J. Kleinveld, M. van Rijn, M. van Deursen, M. Larson, and C. G. M. Snoek. Socialzap: Catch-up on interesting television fragments discovered from social media. In *ICMR*, 2014.
- [51] S. Kordumova, X. Li, and C. G. M. Snoek. Evaluating sources and strategies for learning video concepts from social media. In *CBMI*, 2013.
- [52] S. Kordumova, X. Li, and C. G. M. Snoek. Best practices for learning video concept detectors from social media examples. *Multimedia Tools and Applications*, 74(4):1291–1315, 2015.
- [53] S. Kordumova, T. Mensink, and C. G. M. Snoek. Pooling objects for recognizing scenes without examples. In *ICMR*, 2016.
- [54] S. Kordumova, J. van Gemert, and C. G. M. Snoek. Exploring the long tail of social media tags. In *MMM*, 2016.
- [55] P. Krähenbühl and V. Koltun. Geodesic object proposals. In *ECCV*, 2014.
- [56] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*. 2012.
- [57] G. Kulkarni, V. Premraj, S. Dhar, S. Li, Y. Choi, A. C. Berg, and T. L. Berg. Baby talk: Understanding and generating image descriptions. In *CVPR*, 2011.
- [58] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between class attribute transfer. In *CVPR*, 2009.
- [59] C. H. Lampert, H. Nickisch, and S. Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE Trans. on PAMI*, 36(3):453–465,

- 2014.
- [60] M. Larson, C. Kofler, and A. Hanjalic. Reading between the tags to predict real-world size-class for visually depicted objects in images. In *MM*, 2011.
 - [61] G. Li, M. Wang, Y.-T. Zheng, H. Li, Z.-J. Zha, and T.-S. Chua. Shottagger: tag location for internet videos. In *ICMR*, 2011.
 - [62] H. Li, D. Li, and X. Luo. Bap: Bimodal attribute prediction for zero-shot image categorization. In *MM*, 2014.
 - [63] L.-J. Li, H. Su, L. Fei-Fei, and E. Xing. Object bank: A high-level image representation for scene classification & semantic feature sparsification. In *NIPS*, 2010.
 - [64] L.-J. Li, H. Su, Y. Lim, and L. Fei-Fei. Object bank: An object-level image representation for high-level visual recognition. *Int. J. of Computer Vision*, 107(1):20–39, 2014.
 - [65] M. Li. Texture moment for content-based image retrieval. In *ICME*, 2007.
 - [66] X. Li, C. G. M. Snoek, and M. Worring. Learning social tag relevance by neighbor voting. In *IEEE Trans. on Multimedia*, 2009.
 - [67] X. Li, C. G. M. Snoek, and M. Worring. Unsupervised multi-feature tag relevance learning for social image retrieval. In *CIVR*, 2010.
 - [68] X. Li, C. G. M. Snoek, M. Worring, D. C. Koelma, and A. W. M. Smeulders. Bootstrapping visual categorization with relevant negatives. In *IEEE Trans. on Multimedia*, 2013.
 - [69] X. Li, C. G. M. Snoek, M. Worring, and A. W. M. Smeulders. Harvesting social images for bi-concept search. *IEEE Trans. on Multimedia*, 14(4):1091–1104, 2012.
 - [70] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. Zitnick. Microsoft COCO: Common Objects in Context. In *ECCV*, 2014.
 - [71] D. Liu, X.-S. Hua, L. Yang, M. Wang, and H.-J. Zhang. Tag ranking. In *WWW*, 2009.
 - [72] Y. Liu, D. Xu, I. W.-H. Tsang, and J. Luo. Textual query of personal photos facilitated by large-scale web data. In *IEEE Trans. on PAMI*, 2011.
 - [73] D. G. Lowe. Distinctive image features from scale-invariant keypoints. In *Int. J. of Computer Vision*, 2003.
 - [74] S. Maji, A. Berg, and J. Malik. Classification using intersection kernel support vector machines is efficient. In *CVPR*, 2008.
 - [75] S. Manén, M. Guillaumin, and L. Van Gool. Prime object proposals with randomized Prim’s algorithm. In *ICCV*, 2013.
 - [76] E. Meij, W. Weerkamp, and M. de Rijke. Adding semantics to microblog posts. In *WSDM*, 2012.
 - [77] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013.
 - [78] G. A. Miller. Wordnet: A lexical database for english. In *Communications of the ACM*, 1995.

- [79] P. Mondriaan. Neo-plasticism in painting. In *De Stijl*, 1917.
- [80] Y. Ni, M. Zheng, J. Bu, C. Chen, and D. Wang. Personalized automatic image annotation based on reinforcement learning. In *ICME*, 2013.
- [81] L. Nie, S. Yan, M. Wang, R. Hong, and T.-S. Chua. Harvesting visual concepts for image search with complex queries. In *MM*, 2012.
- [82] M. Norouzi, T. Mikolov, S. Bengio, Y. Singer, J. Shlens, A. Frome, G. Corrado, and J. Dean. Zero-shot learning by convex combination of semantic embeddings. In *ICLR*, 2014.
- [83] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *Int. J. of Computer Vision*, 42(3):145–175, 2001.
- [84] V. Ordonez, G. Kulkarni, and T. L. Berg. Im2text: Describing images using 1 million captioned photographs. In *NIPS*, 2011.
- [85] D. Parikh and K. Grauman. Relative attributes. In *ICCV*, 2011.
- [86] G. Patterson and J. Hays. SUN attribute database: Discovering, annotating, and recognizing scene attributes. In *CVPR*, 2012.
- [87] G. Patterson, C. Xu, H. Su, and J. Hays. The SUN attribute database: Beyond categories for deeper scene understanding. *Int. J. of Computer Vision*, 108(1-2):59–81, 2014.
- [88] A. Quattoni and A. Torralba. Recognizing indoor scenes. In *CVPR*, 2009.
- [89] E. Rahtu, J. Kannala, and M. Blaschko. Learning a category independent object detection cascade. In *ICCV*, 2011.
- [90] C. Rashtchian, P. Young, M. Hodosh, and J. Hockenmaier. Collecting image annotations using amazon’s mechanical turk. In *CSLDAMT*, 2010.
- [91] S. Ray and M. Craven. Supervised versus multiple instance learning: An empirical comparison. In *ICML*, 2005.
- [92] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In *NIPS*, 2015.
- [93] B. Romera-Paredes and P. Torr. An embarrassingly simple approach to zero-shot learning. In *ICML*, 2015.
- [94] 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. *Int. J. of Computer Vision*, 115(3):211–252, 2015.
- [95] B. C. Russell, A. Torralba, K. P. M., and W. T. Freeman. Labelme: A database and web-based tool for image annotation. *Int. J. of Computer Vision*, 77(1-3):157–173, 2008.
- [96] J. Sánchez, F. Perronnin, T. Mensink, and J. J. Verbeek. Image classification with the fisher vector: Theory and practice. *Int. J. of Computer Vision*, 105(3):222–245, 2013.
- [97] G. Schindler, L. Zitnick, and M. Brown. Internet video category recognition. *CVPR*, 2008.
- [98] T. Schnabel, I. Labutov, D. Mimno, and T. Joachims. Evaluation methods for

- unsupervised word embeddings. In *EMNLP*, 2015.
- [99] F. Schroff, A. Criminisi, and A. Zisserman. Harvesting image databases from the web. In *ICCV*, 2007.
 - [100] P. G. Schyns and A. Oliva. From blobs to boundary edges: Evidence for time-and spatial-scale-dependent scene recognition. *Psychological Science*, 5(4):195–200, 1994.
 - [101] B. Settles, M. Craven, and S. Ray. Multiple-instance active learning. In *NIPS*. 2008.
 - [102] A. Setz and C. G. M. Snoek. Can social tagged images aid concept-based video search? In *ICME*, 2009.
 - [103] B. Sigurbjörnsson and R. van Zwol. Flickr tag recommendation based on collective knowledge. In *WWW*, 2008.
 - [104] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *ICCV*, 2003.
 - [105] M. Slaney, K. Weinberger, and R. van Zwol. Resolving tag ambiguity. In *MM*, 2008.
 - [106] A. F. Smeaton, P. Over, and W. Kraaij. Evaluation campaigns and TRECVID. In *MIR*, 2006.
 - [107] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Trans. on PAMI*, 22(12):1349–1380, 2000.
 - [108] C. G. M. Snoek, K. E. A. van de Sande, D. Fontijne, A. Habibian, M. Jain, S. Kordumova, Z. Li, M. Mazloom, S.-L. Pintea, R. Tao, D. C. Koelma, and A. W. M. Smeulders. MediaMill at TRECVID 2013: Searching concepts, objects, instances and events in video. In *the 11th TRECVID Workshop*, November 2013.
 - [109] R. Socher, M. Ganjoo, C. D. Manning, and A. Y. Ng. Zero-shot learning through cross-modal transfer. In *NIPS*, 2013.
 - [110] Y. Sun and A. Kojima. A novel method for semantic video concept learning using web images. In *MM*, 2011.
 - [111] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, 2015.
 - [112] B. Thomee, B. Elizalde, D. Shamma, K. Ni, G. Friedland, D. Poland, D. Borth, and L.-J. Li. YFCC100M: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73, 2016.
 - [113] A. Torralba and A. Oliva. Depth estimation from image structure. *IEEE Trans. on PAMI*, 24(9):1226–1238, 2002.
 - [114] B. Truong, A. Sun, and S. Bhowmick. Content is still king: the effect of neighbor voting schemes on tag relevance for social image retrieval. In *ICMR*, 2012.
 - [115] J. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders. Selective search for object recognition. *Int. J. of Computer Vision*, 104(2):154–171, 2013.
 - [116] A. Ulges, M. Koch, and D. Borth. Linking visual concept detection with viewer demographics. In *ICMR*, 2012.

- [117] A. Ulges, C. Schulze, D. Keysers, and T. Breuel. Identifying relevant frames in weakly labeled videos for training concept detectors. In *CIVR*, 2008.
- [118] A. Ulges, C. Schulze, D. Keysers, and T. Breuel. A system that learns to tag videos by watching youtube. In *ICVS*, 2008.
- [119] T. Uricchio, L. Ballan, M. Bertini, and A. Del Bimbo. An evaluation of nearest-neighbor methods for tag refinement. In *ICME*, 2013.
- [120] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek. Evaluating color descriptors for object and scene recognition. In *IEEE Trans. on PAMI*, 2010.
- [121] J. van de Weijer, C. Schmid, J. Verbeek, and D. Larlus. Learning color names for real-world applications. *IEEE Trans. on Image Processing*, 18(7):1512–1523, 2009.
- [122] V. N. Vapnik. *Statistical learning theory*. Wiley, 1998.
- [123] D. Vreeswijk, K. E. A. van de Sande, C. G. M. Snoek, and A. W. M. Smeulders. All vehicles are cars: Subclass preferences in container concepts. In *ICMR*, 2012.
- [124] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical report, 2011.
- [125] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, 2013.
- [126] H. Wang, F. Wu, X. Li, S. Tang, J. Shao, and Y. Zhuang. Jointly discovering fine-grained and coarse-grained sentiments via topic modeling. In *MM*, 2014.
- [127] Z. Wang, M. Zhao, Y. Song, S. Kumar, and B. Li. Youtubecat: Learning to categorize wild web videos. In *CVPR*, 2010.
- [128] J. Wu and J. M. Rehg. CENTRIST: A visual descriptor for scene categorization. *IEEE Trans. on PAMI*, 33(8):1489–1501, 2011.
- [129] L. Wu, L. Yang, N. Yu, and X.-S. Hua. Learning to tag. In *WWW*, 2009.
- [130] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010.
- [131] L. Xie, Q. Tian, R. Hong, and B. Zhang. Image classification and retrieval are one. In *ICMR*, 2015.
- [132] R. Yan, A. G. Hauptmann, and R. Jin. Negative pseudo-relevance feedback in content-based video retrieval. In *MM*, 2003.
- [133] J. Yang and A. Hauptmann. (Un)reliability of video concept detection. In *CIVR*, 2008.
- [134] Y. Yang, P. Cui, W. Zhu, H. V. Zhao, Y. Shi, and S. Yang. Emotionally representative image discovery for social events. In *ICMR*, 2014.
- [135] Y. Yang, C. L. Teo, H. Daumé, III, and Y. Aloimonos. Corpus-guided sentence generation of natural images. In *EMNLP*, 2011.
- [136] J. Yuan, Z.-J. Zha, Y.-T. Zheng, M. Wang, X. Zhou, and T.-S. Chua. Learning concept bundles for video search with complex queries. In *MM*, 2011.
- [137] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014.
- [138] C. Zhai. Statistical language models for information retrieval: A critical review.

- FnTIR*, 2(3):137–213, 2008.
- [139] W.-L. Zhao, X. Wu, and C.-W. Ngo. On the Annotation of Web Videos by Efficient Near-Duplicate Search. In *IEEE Trans. on Multimedia*, 2010.
 - [140] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Object detectors emerge in deep scene cnns. In *ICLR*, 2015.
 - [141] B. Zhou, A. Khosla, A. Lapedriza, A. Torralba, and A. Oliva. Places2: A large-scale database for scene understanding. In *arXiv*, 2015.
 - [142] S. Zhu, C.-W. Ngo, and Y.-G. Jiang. Sampling and ontologically pooling web images for visual concept learning. *IEEE Trans. on Multimedia*, 14(4):1068–1078, 2012.
 - [143] S. Zhu, G. Wang, C.-W. Ngo, and Y.-G. Jiang. On the sampling of web images for learning visual concept classifiers. In *CIVR*, 2010.
 - [144] X. Zhu, D. Anguelov, and D. Ramanan. Capturing long-tail distributions of object subcategories. In *CVPR*, 2014.
 - [145] C. L. Zitnick and P. Dollár. Edge Boxes: Locating Object Proposals from Edges. In *ECCV*, 2014.
 - [146] C. L. Zitnick, R. Vedantam, and D. Parikh. Adopting abstract images for semantic scene understanding. *IEEE Trans. on PAMI*, 38(4):627 – 638, 2014.

Samenvatting

Leren Zoeken naar Afbeeldingen zonder Annotaties

Dit proefschrift draagt bij aan het door een computer begrijpen van wat zich in een plaatje bevindt, zonder direct gebruik te maken van handmatige annotaties. In plaats daarvan maken we of gebruik van data die op sociale media platformen geannoteerd zijn met tekst, zogenaamde tags, om concepten te herkennen (Hoofdstuk 2 en 3), of van de semantiek van objecten en hun positie in de afbeelding om scènes te herkennen (Hoofdstuk 4 en 5). Voordat we de algemene conclusies bespreken, beschrijven we elk hoofdstuk in het kort.

In Hoofdstuk 2 laten we zien dat geleerde videoconcept detectoren op basis van sociale media platformen, zoals plaatjes van Flickr en video's van Youtube, de potentie hebben om een breed scala van zoekopdrachten voor concepten in video's te adresseren. Hoewel dit potentieel al eerder is erkend, en de vooruitgang hierin indrukwekkend is, zijn we van mening dat fundamentele vragen als *hoe leren we effectieve video concept detectoren met behulp van voorbeelden uit sociale media?* open blijven. Als eerste poging om deze vragen te beantwoorden, voeren we een experimentele studie uit met een visuele zoekmachine die concept detectoren kan leren van voorbeelden uit sociale media, zowel video's en afbeeldingen met sociale media tags. Voor de video zoekmachine onderzoeken we drie strategieën voor het selecteren van positieve voorbeelden, drie strategieën voor het selecteren van negatieve voorbeelden, en drie strategieën voor het leren van detectoren. De zoekmachine is geëvalueerd op de uitdagende TRECVID 2012 dataset, die bestaat uit 600 uur van internetvideo's. Op basis van deze experimenten leiden we vier effectieve werkmethodes af, (1) tagged plaatjes zijn een betere bron voor het leren van video concepten dan tagged video's, (2) het selecteren van relevante tags van training voorbeelden is altijd behulpzaam, (3) het selecteren van relevante negatieve training voorbeelden is nuttig maar zou anders toegepast moeten worden voor training video's dan voor training plaatjes, en (4) het leren van concept detectoren op vooraf geselecteerde relevante training data is beter dan het toevoegen van relevantie

tijdens het leren. Het toepassen van deze werkmethodes op onze video zoekmachine resulteert in de best bekende resultaten op de TRECVID 2012 dataset voor het detecteren van concepten zonder gebruik te maken van handmatige annotaties.

In hoofdstuk 3 gaan we uit van het feit dat er miljoenen gebruikers zijn die multimedia data taggen, wat resulteert in een grote verzameling van gebruikte tags. Sommige van deze tags zijn frequent, terwijl andere zeldzaam zijn, voorkomend uit een dikke staart distributie. Voor frequente tags geven de meeste multimediamedethoden om automatisch audiovisuele data te begrijpen uitstekende resultaten. Daarentegen is het niet duidelijk hoe goed deze methoden werken met zeldzame tags. In dit hoofdstuk onderzoeken we welke sociale tags de dikke staart distributie bepalen en hoe goed ze presteren op twee multimedia retrieval scenario's, het bepalen van de relevantie van een tag en het leren van detectoren. We laten gebruikelijk belangrijke tags zien binnen de dikke staart, en door ze uit te bereiden met semantische kennis, wordt het bepalen van de relevantie van tags en het leren van detectoren significant verbeterd.

In hoofdstuk 4 streven wij ernaar om de scene van plaatjes te herkennen zonder gebruik te maken van voorbeelden bestaande uit plaatjes geannoteerd met deze scenes. Anders dan bij methodes die gebruik maken van attributen, selecteren wij niet zorgvuldig de voorbeeldklassen opdat deze in overeenstemming zijn met de klassen van de scenes. In plaats daarvan stellen wij voor om tienduizend kant-en-klare object detectoren te bundelen. Om de kennisoverdracht tussen objecten en scenes te sturen, leren wij een semantische inbedding met behulp van een groot sociale media corpus. Onze belangrijkste bijdragen zijn: we zijn de eersten om de bundeling over tienduizend object detectors te onderzoeken voor het herkennen van scenes zonder voorbeelden; we verkennen de ontologische hiërarchie van objecten en analyseren de invloed van object detectoren uit verschillende lagen in de hiërarchie; we benutten de positie van objecten in plaatjes van scenes en we demonstreren een nieuwe scene zoek scenario met complexe zoekopdrachten. Tenslotte, we overtreffen representaties gebaseerd op attributen op twee uitdagende datasets, SUNattributes en Places2.

In hoofdstuk 5 stellen we voor om scenes te representeren als een abstractie van "dingen". We beginnen met "dingen" zoals die gegenereerd worden door moderne methodes voor het voorspellen van object locaties, en we onderzoeken hun direct zichtbare eigenschappen: positie, grootte, hoogte-breedteverhouding, en kleur, en alleen deze eigenschappen. Waar de recente successen en opwinding in het onderzoeksveld liggen in het identificeren van objecten, representeren wij de compositie van de scene onafhankelijk van de identiteit van de objecten. We maken drie contributies. Ten eerste, bestuderen we simpele zichtbare eigenschappen van "dingen", een noemen dit de syntax der dingen. Ten tweede, stellen we voor om deze syntax der dingen te vertalen naar linguïstische, abstracte uitspraken en bestuderen hun descriptieve uitwerking om scenes te vinden. Ten derde, stellen we voor om scenes te queryën met abstracte blokillustraties en bestud-

eren de doeltreffendheid hiervan om verschillende soorten scenes te onderscheiden. Het voordeel van abstracte verklaringen en blokillustraties is dat we deze direct kunnen genereren van de plaatjes, zonder enige vorm van leren van tevoren, zoals gebruikelijk is bij het leren van attributen. Verrassend genoeg kunnen we laten zien dat ondanks dat we de meeste simpele representaties van "dingen" gebruiken en in zijn geheel niet leren, we nog steeds met redelijke precisie scenes kunnen vinden.

Acknowledgments

Finishing a PhD is a non-trivial job. Besides the hard work, it also involves many personal struggles, which I could not have overcome without the help and support from my colleagues, friends, and family. Therefore, I dedicate these last pages of the PhD thesis to all those who were beside me in these years. As proud and as happy as I am for finishing the PhD, I am even more thankful to have you all in my life.

First, I thank my supervisor Cees Snoek. Ever since my arrival, Cees has been supportive and encouraging. We spent many hours in his office talking about research, making plans, and learning. I could not have found a better supervisor to learn from on how to write a paper, a review, or a rebuttal, on how to present, on how to approach a problem scientifically, and many other skills. We have been through many rejections and great successes like winning TrecVID and the best paper award at ICMR, and Cees was always there to encourage me or to celebrate with me. Of his many good qualities, his ability to easily pivot research to new directions whenever problems arise was quite helpful in times when I was emotionally attached to old work. What I mostly appreciate is his endless optimism, which spreads positive energy through the whole group. The effect of it is great successes and results.

Next I would like to thank my promoter Arnold Smeulders for his guidance and the long talks we shared over research. Arnold is an exceptional scientist. He can always make you think from another perspective and on another level, different from the one you are focused on. His broad knowledge, great ability to read people, and sense of art are other things I appreciate greatly with him. I learned many things from Arnold, and for all of it, I thank him.

At the beginning of my third year, I also started collaborating with Jan van Gemert. His detailed knowledge and discussions were important for my growth as a researcher. We had many talks together on the blue couch in the common room about paper ideas, experiments, methods, detailed plans, and much more. Jan is the type of supervisor you can discuss it all with, from the big picture to small details. Although at some point we had some disagreements, I can only be thankful for all that I learned.

With Thomas Mensink I had the privilege to work on my last paper, which led to a best paper award. His calm approach and discussions on all aspects of the paper helped us improve it. Although he left one month before the deadline to visit one of the best scientific groups in the field at Berkeley, he still made time for us, for which I am very grateful.

Next I would like to thank all my colleagues at the University of Amsterdam that made every day of work enjoyable. Many people left and came during my four and a half years of stay, and I am happy to have known them all. When I started in the office, I met Hamdi, Dung, Gosia, Fares, Sezer, and Silvia. Hamdi was helpful from the very beginning, providing us with many tips. Fares and Sezer were also my neighbors, and we built a great friendship by helping each other, and spending nice time together at dinners, parties, and trips. Everyone else that I met in our group: Ran, Zhenyang, Lou, Yang, Masoud, Amir, Efstratios, Ivo, Agni, Kandan, Jorn, Dennis, Marcel, Mihir, Morris, Honza, Pascal, Spencer, and Stevan, contributed to many nice moments together of barbecues, ISIS dinners or beach volleyball. A big thanks to Virginie, who is always there to land a helping hand for all of us, and Dennis who always managed to make us laugh and bring calmness in times when all of us were stressed for a deadline. Thanks to Honza for being my squash teacher and partner in starting a squash gang. Spencer and Pascal cheered up the office with their arrival and jokes.

Ran, Zhenyang, Masoud, and I spent three months together in New York, on a research visit at Columbia University. During this time we shared many conversations, meals, walks, and we became great friends. I will always be thankful for the support we gave each other, and that we continue it up to this day. Also thanks to Prof. Shih-Fu Chang and Felix Yu, with who I collaborated during my stay at Columbia University, for all the research discussions and time spend with me.

To the many friends in Amsterdam, thank you for all the good times we had together. Special thanks to Sofia, Aziz, Noortje, Marry, Maria, Diego and Andreia, who were always there to share a drink and dinner with me, and ask how am I. To Daniela, RYanne, Ognen, Jose and Bojana, thanks for all the great moments shared. And to Ivana, who despite living in Eindhoven, still managed to meet with me often and share nice moments and talks. To the many friends in Macedonia who are still close and dear to me, Ivana, Stef, Dani, Bibce, Baze, Elena, Elena, Tanja, Ivan, Natasha, Marija and Elena, thanks for your great friendship.

Probably the most valuable person for completing this PhD with my sanity is my twin sister Suzana. Although far apart, she was always only a phone call away to listen to all my complaints, to encourage me and give me strength to carry on. Thanks to all her visits to Amsterdam, for fun or for help when needed. And thanks to her loving husband Goran for always being there for her.

A big thanks also goes to my older sister Marija and her lovely family, Naum, Tea and Stefan, who were always adding joy to my heart whenever we had a chance to meet.

To my parents, who give everything in their power for me and my sisters to have the best of life possible, I am very grateful. You are always there to share advice, to give me support, to be happy for me in times of joy, and help me out when needed. Thanks for your unconditional love, for teaching me all the values in life, and to cherish most the greatest one of all: family.

My final gratitude goes to my best friend, love, and husband Petar. We grew together, and we managed to stay together despite the thousands of kilometers between us. You always succeed to turn all my worries into jokes and to put a smile on my face. Thank you for your support and help throughout the years. I am so excited to start a family and a new life with you.