



UNIVERSITÀ DEGLI STUDI DI FIRENZE  
CORSO DI DOTTORATO IN INGEGNERIA INFORMATICA  
DIPARTIMENTO DELL'INFORMAZIONE (DINFO)  
ING-INF/05

---

IMAGE UNDERSTANDING  
BY SOCIALIZING  
THE SEMANTIC GAP

*Candidate*

Tiberio Uricchio

*Supervisors*

Prof. Alberto Del Bimbo

Prof. Marco Bertini

*PhD Coordinator*

Prof. Luigi Chisci

Università degli Studi di Firenze, Dipartimento dell'Ingegneria dell'Informazione (DINFO).

Thesis submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computer Engineering, Systems and Telecommunications. Copyright © 2015 by Tiberio Uricchio.

*Ai miei genitori.*

## Acknowledgments

This thesis would not have been possible without the help and support of many people. First, I would like to acknowledge the efforts and input of my supervisors, Prof. Alberto Del Bimbo and Prof. Marco Bertini, who were of great help during my research. I thank Telecom Italia and my industrial tutor Carlo Alberto Licciardi for their support of my work.

Many people contributed to the development of this research. The discussions with Dr. Lamberto Ballan were extremely important to me and the success of this work. Thank you for your patience and insightful suggestions. Special thanks to Prof. Cees G.M. Snoek, Dr. Xirong Li, Dr. Lorenzo Seidenari, Claudio Baccchi and Francesco Gelli who collaborated on several parts of my research work. I will be always grateful to you all. Props out to Prof. Paolo Frascioni who inspired me with his visionary look at research. His passion has always encouraged me to take bolder roads.

Many many thanks to all my colleagues of the Media Integration and Communication Center (MICC): Prof. Andrew Bagdanov, Federico Bartoli, Federico Becattini, Giuseppe Becchi, Enrico Bondi, Maxime Devanne, Dario Di Fina, Simone Ercoli, Andrea Ferracani, Claudio Ferrari, Leonardo Galteri, Svebor Karaman, Giuseppe Lisanti, Iacopo Masi, Federico Pernici, Daniele Pezzatini, Francesco Turchini. In these 3 years, their enthusiasm and their support was extremely important to me during the highs and especially in the lows.

Thanks to my long-standing friends Elisa, Francesco, Giulia, Laura, Leonardo, Riccardo, Stefano, Yasamin. You are very important to me. Finally, I don't know how I would have made it through without the support of Valentina and my family. I thank them deeply for all the love and understanding.

# Contents

<b>Contents</b>	<b>v</b>
<b>1 Introduction</b>	<b>1</b>
1.1 The goal . . . . .	1
1.2 Contributions and Organization . . . . .	4
<b>2 Literature review of Assignment, Refinement and Retrieval</b>	<b>9</b>
2.1 Problems and Tasks . . . . .	10
2.2 Scope and Aims . . . . .	11
2.3 Foundations . . . . .	12
2.4 Media for tag relevance . . . . .	14
2.4.1 Tag based . . . . .	15
2.4.2 Tag + Image based . . . . .	15
2.4.3 Tag + Image + User information based . . . . .	16
2.5 Learning for tag relevance . . . . .	17
2.5.1 Instance-based . . . . .	19
2.5.2 Model-based . . . . .	20
2.5.3 Transduction-based . . . . .	21
2.6 Auxiliary components . . . . .	23
2.7 Conclusions . . . . .	24
<b>3 A new Experimental Protocol</b>	<b>27</b>
3.1 Introduction . . . . .	27
3.2 Datasets . . . . .	28
3.3 Implementation and Evaluation . . . . .	30
3.3.1 Evaluating tag assignment . . . . .	31
3.3.2 Evaluating tag refinement . . . . .	32
3.3.3 Evaluating tag retrieval . . . . .	32

---

3.4	Methods under analysis . . . . .	33
3.4.1	SemanticField . . . . .	34
3.4.2	TagRanking . . . . .	34
3.4.3	KNN . . . . .	35
3.4.4	TagVote . . . . .	35
3.4.5	TagProp . . . . .	36
3.4.6	TagCooccur . . . . .	37
3.4.7	TagCooccur+ . . . . .	37
3.4.8	TagFeature . . . . .	37
3.4.9	RelExample . . . . .	38
3.4.10	RobustPCA . . . . .	39
3.4.11	TensorAnalysis . . . . .	40
3.4.12	Considerations . . . . .	42
3.5	Evaluation . . . . .	42
3.5.1	Tag assignment . . . . .	42
3.5.2	Tag refinement . . . . .	45
3.5.3	Tag retrieval . . . . .	48
3.5.4	Flickr versus ImageNet . . . . .	57
3.6	Conclusions . . . . .	60
<b>4</b>	<b>A Cross Modal Approach for Tag Assignment</b>	<b>61</b>
4.1	Introduction . . . . .	61
4.1.1	Contribution . . . . .	64
4.2	Related Work . . . . .	64
4.3	Approach . . . . .	65
4.3.1	Visual and Tags Views . . . . .	66
4.3.2	Kernel Canonical Correlation Analysis . . . . .	68
4.3.3	Tag Assignment Using Nearest Neighbor Models in the Semantic Space . . . . .	70
4.4	Experiments . . . . .	72
4.4.1	Datasets . . . . .	74
4.4.2	Evaluation Measures . . . . .	76
4.4.3	Results . . . . .	76
4.5	Conclusions . . . . .	78
<b>5</b>	<b>Fisher Encoded Bag-of-Windows Representation</b>	<b>81</b>
5.1	Introduction . . . . .	81
5.2	Previous work . . . . .	83

---

5.3	Proposed method . . . . .	85
5.3.1	Image representation . . . . .	85
5.3.2	Image retrieval . . . . .	86
5.3.3	Tag Assignment . . . . .	87
5.4	Experiments . . . . .	88
5.5	Conclusion . . . . .	93
<b>6</b>	<b>Evaluating Temporal Information in Social Images</b>	<b>95</b>
6.1	Introduction . . . . .	95
6.2	Data Analysis Method . . . . .	98
6.2.1	Datasets . . . . .	98
6.2.2	Temporal features . . . . .	99
6.2.3	Flickr Popularity Model . . . . .	100
6.2.4	Processing . . . . .	100
6.2.5	Correlation analysis . . . . .	102
6.3	Experiments and Discussion . . . . .	103
6.3.1	Temporal Evaluation . . . . .	103
6.3.2	Correlation Analysis . . . . .	105
6.4	Conclusions . . . . .	108
<b>7</b>	<b>Multimodal Feature Learning for Sentiment Analysis</b>	<b>111</b>
7.1	Introduction . . . . .	111
7.2	Previous Work . . . . .	112
7.3	The Proposed Method . . . . .	116
7.3.1	Textual information . . . . .	116
7.3.2	Textual and Visual Information . . . . .	120
7.4	Experiments . . . . .	123
7.5	Conclusions . . . . .	130
<b>8</b>	<b>Popularity Prediction with Sentiment and Context Features</b>	<b>133</b>
8.1	Introduction . . . . .	133
8.2	Related work . . . . .	135
8.3	The Proposed Method . . . . .	136
8.3.1	Measuring Popularity . . . . .	136
8.3.2	Visual Sentiment Features . . . . .	137
8.3.3	Object Features . . . . .	137
8.3.4	Context Features . . . . .	137
8.3.5	User Features . . . . .	138

8.3.6	Popularity prediction . . . . .	139
8.4	Experiments . . . . .	139
8.4.1	Results . . . . .	140
8.4.2	Qualitative Analysis . . . . .	141
8.5	Conclusions . . . . .	142
<b>9</b>	<b>Conclusion</b>	<b>145</b>
9.1	Summary of Contribution . . . . .	145
9.2	Direction of future work . . . . .	147
<b>A</b>	<b>Publications</b>	<b>149</b>
	<b>Bibliography</b>	<b>151</b>



# Chapter 1

## Introduction

Sharing images is an essential experience. Be it a drawing carved in rock, a painting exposed in a museum, or a photo capturing a special moment, it is the sharing that relives the experience stored in the image. Several technological developments have spurred the sharing of images in unprecedented volumes. The first is the ease with which images can be captured in a digital format by cameras, cellphones and other wearable sensory devices. The second is the Internet that allows transfer of digital image content to anyone, anywhere in the world. Finally, and most recently, the sharing of digital imagery has reached new heights by the massive adoption of social network platforms. All of a sudden images came with tags, and tagging, commenting, and rating of any digital image has become a common habit. The sharing paradigm is lead by users interactions with each other, like forming groups of shared interests, sharing messages that convey sentiments, and by commenting the photos that have been shared. And consequently, in the huge quantity of available media, some of these images are going to become very popular, while others are going to be totally unnoticed and end up in oblivion.

### 1.1 The goal

Our ultimate goal is to extract *information* from image collections in social networks. In particular, we aim at obtaining tags, i.e. human interpretable labels associated to the content at a global level. These can be related to objective aspects such as the presence of things, properties and activities, or

subjective ones such as the sentiments aroused in a viewer or the attractiveness of an image.

Being able to extract this information can have a great impact in several applications. First, the retrieval of images from collections can be improved. Current image search engines (such as Google or Yahoo), that traditionally rely on the associated text data, have recently exploited the visual content to improve performance. Similarly, in social networks, they mostly rely on user provided metadata in form of tags or textual description. Second, it can ease the browsing of large collections. For instance, through selection or summarization of the most attractive and significative photos. In particular, sentiments aroused in the viewer can play a role in producing significative output. Third, the distribution and enjoyment of contents can be improved. Advertising and distribution of content can be more efficient when matching content to user preferences. Moreover, to the aim of minimizing storage costs, images may be replicated according to popularity and still maintaining a low latency for unpopular content. For these reasons, image retrieval and understanding receive a lot of attention from both the scientific community and industry.

Machine understanding of media is still very poor. While their data processing capabilities are continuously improving (e.g. Moore's law [148]), stemming information from unannotated multimedia is a challenging task. The main hindrance is that machines are able to compute only low level features of the data, hardly correlated to the semantics. Tasks such as recognizing things, understanding the sentiment induced in the viewer or predicting the expected attractiveness of an image, require high level features. This is a well-known problem in the literature, formalized as the **semantic gap** [177]: "The semantic gap is the lack of coincidence between the information that machines can extract from the visual data and the interpretations the user may give to the data.". Hence the ensuing question is:

*How can we fill the semantic gap for multimedia understanding?*

**We believe that Social Networks are promising frameworks that can fill the gap.** Comparing to the classic multimedia databases, social networks provide a dilated context where the user is king. Users can contribute by providing photos with attached metadata (such as tags, description, location) or by expressing interest in others content (e.g. likes, comments). In Figures 1.1 and 1.2 we show two examples of such contributions in two different social networks.



Figure 1.1: Example of a user generated content on social network Instagram. An image of a cat is associated with a little description and several tags. Several users have commented the content.

Social network contributions are provided by common users. They often cannot meet high quality standards related to content association, in particular for accurately describing objective aspects of the visual content according to some expert's opinion [42]. Moreover, when subjective components are considered (e.g. sentiments), different users may read images differently.

The most historically exploited pieces of metadata are the social tags associated to the images. These tags tend to follow context, trends and events in the real world. They are often used to describe both the situation and the entity represented in the visual content. So tagging deviations due to spatial and temporal correlation to external factors, including user influence, semantics of activity and relationships between tags, are common phenomena. Social tags tend to be imprecise, ambiguous, incomplete and biased towards personal perspectives [61, 91, 172, 174].

Quite a few researchers have proposed solutions for image annotation and retrieval in social frameworks [117], although the peculiarities of this domain have been only partially addressed.

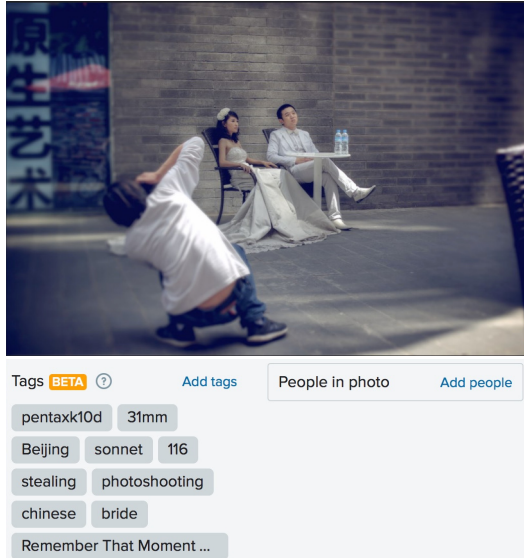


Figure 1.2: Example of a user generated content on social network Flickr. Tags are associated to an image of a newly married couple.

## 1.2 Contributions and Organization

In this thesis we show that the tagged images shared in social media platforms are promising to resolve the semantic gap. In particular, we focus on image annotation and provide a structured survey of methods in social networks with a thorough empirical evaluation of several key methods. Then we describe four novel state-of-the-art methods for extracting information, that explicitly take into account the social context.

Two themes can be highlighted. The first one is related to the task of objective analysis of images (i.e. recognize things), while the second one relates to the tasks of subjective analysis (i.e. recognize the sentiment induced in viewers, predict the expected popularity of images). In spite of the two themes, the underlying idea of our work is the exploitation of social images through the design of features that comprises both the visual observation *and* their tags. Learned or handcrafted, these features provide a robust global representation of the content and context.



Figure 1.3: A machine processed image where an algorithm of tag refinement has been applied. Not relevant tags are removed and additional relevant tags are added.

The thesis is organized as follows<sup>1</sup>. Considering the absence of a comprehensive review of annotation and retrieval in social networks, we start in Chapter 2 with a structured survey of related work. Although image annotation and retrieval in social networks are a relatively recent direction of research, several tasks have been addressed by the multimedia community. We survey three linked semantic tasks (i.e. tag assignment, tag refinement and tag-based image retrieval) that have seen the most contributions to date. Figure 1.3 shows an example of tag refinement of an image and its associated user tags. Recognizing a lack of a structured survey in the literature, we aimed at giving a reference contribution for future researchers in this field. We organize the rich literature of tagging and retrieval in a taxonomy to highlight the ingredients of the main works and recognize their advantages and limitations. In particular, we structure our survey along the line of understanding how a specific method constructs the underlying tag relevance function.

Witnessing the absence of a thorough empirical comparison in the literature for the three semantic tasks, in Chapter 3 we establish a common

<sup>1</sup>Note that each chapter is written in a self-contained fashion and can be read on its own.

experimental protocol and successively exert it in the evaluation of key methods. Our proposed protocol contains training data of varied scales extracted from social frameworks. This permits to evaluate the methods under analysis with data that reflect the specificity of the social domain. We made the data and source code public so that new proposals for tag assignment, tag refinement, and tag retrieval can be evaluated rigorously and easily. Taken together with Chapter 2, these efforts should provide an overview of the field's past and foster progress for the near future.

Chapters 4 and 5 builds on ideas from the previous Chapters to propose two novel approaches for tag assignment.

In Chapter 4, by considering visual content and the tags associated with an image, novel features are automatically learned. A cross-model method is proposed to capture the intricate dependencies between image content and annotations. We propose a learning procedure based on Kernel Canonical Correlation Analysis which finds a mapping between visual and textual words by projecting them into a latent meaning space. The learned mapping is then used to annotate new images using advanced nearest neighbor voting methods. We evaluate our approach on three popular datasets, and show clear improvements over several approaches relying on more standard representations.

In Chapter 5 we present an efficient and powerful method to aggregate a set of Deep Convolutional Neural Network responses, extracted from a set of image windows. We show how to use Fisher Vectors and PCA to obtain a short and highly descriptive signature that can be used for effective image retrieval. We show also how the very good performance in retrieval can be exploited for tag assignment. State-of-the art results are reported for both tasks of image retrieval and tag assignment on standard datasets.

Chapter 6 gives an evaluation of the temporal information in web images. The idea is to use the temporal gist of annotations to improve tasks such as annotation, indexing and retrieval. While visual content, text and metadata, are typically used to improve these tasks, here we look at the temporal aspect of social media production and tagging. The correlation of the time series of the tags with Google searches shows that, for certain concepts, web information sources may be beneficial to the annotation of social media.

Chapters 7 and 8 deal with the non semantic problems of image sentiment analysis and popularity prediction. In particular, Chapter 7 investigate the use of a multimodal feature learning approach using neural network based

models such as Skip-gram and Denoising Autoencoders. The task is to perform sentiment analysis of micro-blogging content, such as Twitter short messages, that are composed by a short text and, possibly, an image. A novel architecture that incorporates these models is proposed and tested on several standard Twitter datasets. We show that the approach is efficient and obtains good classification results.

By considering that attractiveness of images is related to popularity, in Chapter 8 we propose to use visual sentiment features together with three novel context features to predict a concise popularity score of social images. Experiments on large scale datasets show the benefits of proposed features on the performance of image popularity prediction. Moreover, exploiting state-of-the-art sentiment features, we report a qualitative analysis of which sentiments seem to be related to good or poor popularity.

Finally, Chapter 9 summarizes the contribution of the thesis and discusses avenues for future research. Notice also that the full-list of published papers from this thesis is provided in Appendix A.





## Chapter 2

# Literature review of Assignment, Refinement and Retrieval

*This chapter gives an unified survey of related work on the three closely linked problems of Tag Assignment, Tag Refinement and Tag-based Image Retrieval. While existing works vary in terms of their targeted tasks and methodology, they rely on the key functionality of tag relevance, i.e., estimating the relevance of a specific tag with respect to the visual content of a given image and its social context. A taxonomy is introduced to structure the growing literature, understand the ingredients of the main works, clarify their connections and difference, and recognize their merits and limitations.*<sup>1</sup>

Excellent surveys on content-based image retrieval have been published in the past. In their seminal work, Smeulders *et al.* review the early years up to the year 2000 by focusing on what can be seen in an image and introducing the main scientific problem of the field: the semantic gap as “the lack of coincidence between the information that one can extract from the visual data and the interpretation that the same data have for a user in a given situation” [177]. Datta *et al.* continue along this line and describe

---

<sup>1</sup>Part of this chapter was submitted as “Socializing the Semantic Gap: A Comparative Survey on Image Tag Assignment, Refinement and Retrieval” to *ACM Computing Surveys*.

the coming-of-age of the field, highlighting the key theoretical and empirical contributions of recent years [37]. These reviews completely ignore social platforms and socially generated images, which is not surprising as the phenomenon only became apparent after these reviews were published.

In this chapter, we survey the state-of-the-art of content-based image retrieval in the context of social image platforms and tagging, with a comprehensive treatise of the closely linked problems of image tag assignment, image tag refinement and tag-based image retrieval. Similar to [177] and [37], the focus of this survey is on visual information, but we explicitly take into account *and* quantify the value of social tagging.

## 2.1 Problems and Tasks

Social tags are provided by common users. They often cannot meet high quality standards related to content association, in particular for accurately describing objective aspects of the visual content according to some expert's opinion [42]. Social tags tend to follow context, trends and events in the real world. They are often used to describe both the situation and the entity represented in the visual content. So tagging deviations due to spatial and temporal correlation to external factors, including user influence, semantics of activity and relationships between tags, are common phenomena. Social tags tend to be imprecise, ambiguous, incomplete and biased towards personal perspectives [61, 91, 172, 174]. Quite a few researchers have proposed solutions for image annotation and retrieval in social frameworks, although the peculiarities of this domain have been only partially addressed. We categorize existing works into three different main tasks and structure our survey along these tasks:

- **Tag Assignment.** Given an unlabeled image, tag assignment strives to assign a (fixed) number of tags related to the image content [68, 134, 181, 194].
- **Tag Refinement.** Given an image associated with some initial tags, tag refinement aims to remove irrelevant tags from the initial tag list and enrich it with novel, yet relevant, tags [50, 121, 122, 211, 233].
- **Tag Retrieval.** Given a tag and a collection of images labeled with the tag (and possibly other tags), the goal of tag retrieval is to retrieve

images relevant with respect to the tag of interest [44,55,113,179,211].

Other related tasks such as tag filtering [125,228,229] and tag suggestion [113,174,212] have also been studied. As these tasks focus on either cleaning existing tags or expanding them, we view them as variants of tag refinement.

## 2.2 Scope and Aims

Existing works in tag assignment, refinement, and retrieval vary in terms of their targeted tasks and methodology, making it non-trivial to interpret them within a unified framework. Nonetheless, we reckon that all works rely on the key functionality of *tag relevance*, i.e., estimating the relevance of a specific tag with respect to the visual content of a given image and its social context. In general terms, relevance should be evaluated considering the complementarity of tags. They may be of low interest alone but become interesting if in conjunction with others. However in the literature, only few methods consider multi-tag relevance evaluation and only for the task of multi-tag retrieval [19,116,150]. Hence, we focus on methods that implement the unique-tag relevance model.

We survey papers that learn from images tagged in social contexts. We do not cover traditional image classification that is grounded on carefully labeled data. For a state-of-the-art overview in that direction, we refer the interested reader to [46,164]. Nonetheless, one may question the necessity of using socially tagged examples as training data, given that a number of labeled resources are already publicly accessible. An exemplar of such resources is ImageNet [40], providing crowd-sourced positive examples for over 20k classes. Since ImageNet employs several web image search engines to obtain candidate images, its positive examples tend to be biased by the search results. As observed by [199], the positive set of vehicles mainly consists of car and buses, although vehicles can be tracks, watercraft and aircraft. Moreover, controversial images are discarded upon vote disagreement during the crowd sourcing. All this reduces diversity in visual appearance. We empirically show in Chapter 3 the advantage of socially tagged examples against ImageNet for tag relevance learning.

Reviews on social tagging exist. The work by Gupta *et al.* discusses papers on why people tag, what influences the choice of tags, and how to

model the tagging process, but its discussion on content-based image tagging is limited [69]. The focus of [77] is on papers about adding semantics to tags by exploiting varied knowledge sources such as Wikipedia, DBpedia, and WordNet. Again, it leaves the visual information untouched.

Several reviews that consider socially tagged images have appeared recently. In [124], technical achievements in content-based tag processing for social images are briefly surveyed. Sawant *et al.* [171], Wang *et al.* [205] and Mei *et al.* [140] present extended reviews of particular aspects, i.e., collaborative media annotation, assistive tagging, and visual search re-ranking, respectively. In [171], papers that propose collaborative image labeling games and tagging in social media networks are reviewed. In [205] the authors survey papers where computers assist humans in tagging either by organizing data for manual labelling, improving quality of human-provided tags or recommending tags for manual selection, instead of applying purely automatic tagging. In [140] the authors review techniques that aim for improving initial search results, typically returned by a text based visual search engine, by visual search re-ranking. These reviews offer resumes of the methods and interesting insights on particular aspects of the domain, without giving an experimental comparison between the varied methods.

We notice efforts in empirical evaluations of social media annotation and retrieval [9, 179, 192]. In [179], the authors analyze different dimensions to compute the relevance score between a tagged image and a tag. They evaluate varied combinations of these dimensions for tag-based image retrieval on NUS-WIDE, a leading benchmark set for social image retrieval [32]. However, their evaluation focuses only on tag-based image ranking features, without comparing content-based methods. Moreover, tag assignment and refinement are not covered. In [9, 192], the authors compared three algorithms for tag refinement on the NUS-WIDE and MIRFlickr, a popular benchmark set for tag assignment and refinement [75]. However, the two reviews lack a thorough comparison between different methods under the umbrella of a common experimental protocol. Moreover, they fail to assess the high-level connection between image tag assignment, refinement, and retrieval.

## 2.3 Foundations

Our key observation is that the essential component, which measures the relevance between a given image and a specific tag, stands at the heart of

the three tasks. In order to describe this component in a more formal way, we first introduce some notation.

We use  $x$ ,  $t$ , and  $u$  to represent the three basic elements in social images, namely image, tag, and user. An image  $x$  is shared on social media by its user  $u$ . A user  $u$  can choose a specific tag  $t$  to label  $x$ . By sharing and tagging images, a set of users  $\mathcal{U}$  contribute a set of  $n$  socially tagged images  $\mathcal{X}$ , wherein  $\mathcal{X}_t$  denotes the set of images tagged with  $t$ . Tags used to describe the image set form a vocabulary of  $m$  tags  $\mathcal{V}$ . The relationship between images and tags can be represented by an image-tag association matrix  $D \in \{0, 1\}^{n \times m}$ , where  $D_{ij} = 1$  means the  $i$ -th image is labeled with the  $j$ -th tag, and 0 otherwise.

Given an image and a tag, we introduce a real-valued function that computes the relevance between  $x$  and  $t$  based on the visual content and an optional set of user information  $\Theta$  associated with the image:

$$f_{\Phi}(x, t; \Theta)$$

We use  $\Theta$  in a broad sense, making it refer to any type of social context provided by or referring to the user like associated tags, where and when the image was taken, personal profile, and contacts. The subscript  $\Phi$  specifies how the tag relevance function is constructed. We can easily interpret each of the three tasks: assignment and refinement can be done by sorting  $\mathcal{V}$  in descending order by  $f_{\Phi}(x, t; \Theta)$ , while retrieval can be achieved by sorting the labeled image set  $\mathcal{X}_t$  in descending order in terms of  $f_{\Phi}(x, t; \Theta)$ . Note that this formalization does not necessarily imply that the same implementation of tag relevance is applied for all the three tasks. For example, for retrieval relevance is intended to obtain image ranking [109] while tag ranking for each single image is the goal of assignment [212] and refinement [157].

Fig. 2.1 presents a unified framework, illustrating the main data flow of varied approaches to tag relevance learning. Compared to traditional methods that rely on expert-labeled examples, a novel characteristic of a social media based method is its capability to learn from socially tagged examples with unreliable annotations. Such a training media is marked as  $\mathcal{S}$  in the framework. Optionally, in order to obtain a refined training media  $\hat{\mathcal{S}}$ , one might consider designing a filter to remove unwanted tags and images. In addition, prior information such as tag statistics, tag correlations, and image affinities in the training media are independent of a specific image-tag pair. They can be precomputed for the sake of efficiency. As the filter

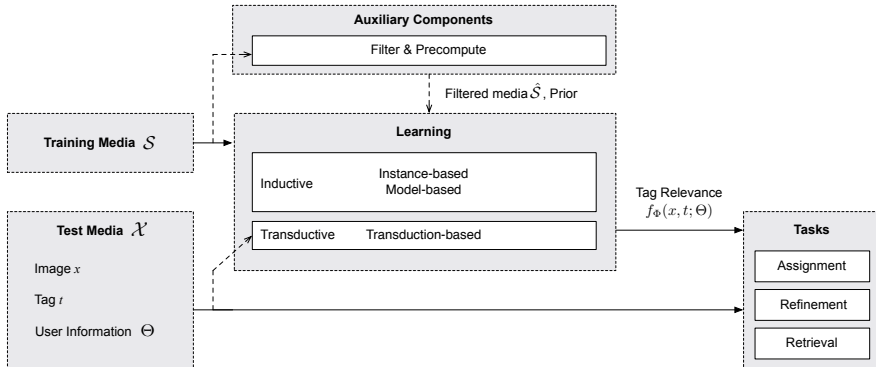


Figure 2.1: **Dataflow to structure the literature on tag relevance learning for image tag assignment, refinement and retrieval.** We follow the input data as it flows through the process of the tag relevance function  $f_{\Phi}(x, t; \Theta)$  to higher level tasks, complete with common internal activities and surrounding auxiliary components. Dashed lines indicate optional processes such as the auxiliary components and transduction-based algorithms.

and the precomputation appear to be a choice of implementation, they are positioned as auxiliary components in Fig. 2.1.

A number of implementations of the relevance function are described and compared in Chapter 3, with regard to their use for tag assignment, refinement and retrieval. Depending on how  $f_{\Phi}(x, t; \Theta)$  is composed internally, we propose a taxonomy which organizes existing works along two dimensions, namely *media* and *learning*. As shown in Table 2.1, the media dimension characterizes *what* essential information  $f_{\Phi}(x, t; \Theta)$  exploits, while the learning dimension depicts *how* such information is exploited. We explore the taxonomy along the media dimension in Section 2.4 and the learning dimension in Section 2.5, followed by a discussion on the two auxiliary components in Section 2.6.

## 2.4 Media for tag relevance

Different sources of information may play a role in determining the relevance between an image and a social tag. For instance, the position of a tag

appearing in the tag list might reflect a user’s tagging priority to some extent [179]. Knowing what other tags are assigned to the image [229] or what other users label about similar images [92, 113] can also be helpful for judging whether the tag under examination is appropriate or not. Depending on what modalities in  $\mathcal{S}$  are utilized, we divide existing works into the following three groups: 1) tag based, 2) tag + image based and 3) tag + image + user information based, ordered in light of the amount of information they utilize. Table 2.1 shows this classification for several papers that appeared in the literature on the subject.

### 2.4.1 Tag based

These methods build  $f_{\Phi}(x, t; \Theta)$  purely based on tag information. Tag position is considered in [179], where a tag appearing top in the tag list is regarded as more relevant. To find tags that are semantically close to the majority of the tags assigned to the test image, tag co-occurrence is considered in [174, 229], while topic modeling is employed in [215]. As the tag based methods presume that the test image has been labeled with some initial tags, i.e. the initial tags are taken as the user information  $\Theta$ , they are inapplicable for tag assignment.

### 2.4.2 Tag + Image based

Works in this group develop  $f_{\Phi}(x, t; \Theta)$  on the base of visual information and associated tags. The main rationale behind them is visual consistency, i.e. visually similar images shall be labeled with similar tags. Implementations of this intuition can be grouped in three conducts. One, leverage images visually close to the test image [48, 113, 114, 131, 194, 213]. Two, exploit relationships between images labeled with the same tag [55, 98, 123, 125, 163]. Three, learn visual classifiers from socially tagged examples [26, 111, 201, 219]. By propagating tags based on the visual evidence, the above works exploit the image modality and the tag modality in a sequential way. By contrast, there are works that concurrently exploit the two modalities. This can be approached by generating a common latent space upon the image-tag association [43, 151, 178], so that a cross media similarity can be computed between images and tags [128, 156, 231]. In [152], the latent space is constructed by Canonical Correlation Analysis, finding two matrices which separately project feature vectors of image and tag into the same subspace.

In [131], a random walk model is used on a unified graph composed from the fusion of an image similarity graph with an image-tag connection graph. In [211, 216, 228], predefined image similarity and tag similarity are used as two constraint terms to enforce that similarities induced from the recovered image-tag association matrix will be consistent with the two predefined similarities.

Although late fusion has been actively studied for multimedia data analysis [4], improving tag relevance estimation by late fusion is not much explored. There are some efforts in that direction, among which interesting performance has been reported in [157] and more recently in [109].

### **2.4.3 Tag + Image + User information based**

In addition to tags and images, this group of works exploit user information, motivated from varied perspectives. With the hypothesis that a specific tag chosen by many users to label visually similar images is more likely to be relevant with respect to the visual content, [113] utilizes user identities to ensure that learning examples come from distinct users. A similar idea is reported in [92], finding visually similar image pairs with matching tags from different users. [58] improves image retrieval by favoring images uploaded by users with good credibility estimates. In [110, 170], personal tagging preference is considered in the form of tag statistics computed from images a user has uploaded in the past. These past images are used in [127] to learn a user-specific embedding space. In [168], user affinities, measured in terms of the number of common groups users are sharing, is considered in a tensor analysis framework. Similarly, tensor based low-rank data reconstruction is employed in [158] to discover latent associations between users, images, and tags. Photo timestamps are exploited for time-sensitive image retrieval [94], where the connection between image occurrence and various temporal factors is modeled. In [136], time-constrained tag co-occurrence statistics are considered to refine the output of visual classifiers for tag assignment. In their follow-up work [137], location-constrained tag co-occurrence computed from images taken in a specific continent is further included. User interactions in social networks are exploited in [170], computing local interaction networks from the comments left by other users. Social-network metadata such as group memberships of images and contacts of users is employed in [87, 135, 202] for image classification.

Comparing the three groups, tag + image appears to be the mainstream,



as evidenced by the imbalanced distribution in Table 2.1. Intuitively, using more media from  $\mathcal{S}$  would typically increase the reliability of tag relevance estimation. We attribute the imbalance among the groups, in particular the relatively few works in the third group, to the following two reasons. First, no publicly available dataset with expert annotations was built to gather representative and adequate user information, e.g. MIRFlickr has nearly 10k users for 25k images, while in NUS-WIDE only 6% of the users have at least 15 images. As a consequence, current works that leverage user information are forced to use a minimal subset to alleviate sample insufficiency [168, 169] or homemade collections with social tags as ground truth instead of benchmark sets [110, 170]. Second, adding more media often results in a substantial increase in terms of both computation and memory, e.g. the cubic complexity for tensor factorization in [168]. As a trade-off, one has to use  $\mathcal{S}$  of a much smaller scale. The dilemma is whether one should use large data with less media or more media but less data.

It is worth noting that the above groups are not exclusive. The output of some methods can be used as a refined input of some other methods. In particular, we observe a frequent usage of tag-based methods by others for their computational efficiency. For instance, tag relevance measured in terms of tag similarity is used in [55, 111, 231] before applying more advanced analysis, while nearest neighbor tag propagation is a pre-process used in [228]. The number of tags per image is embedded into image retrieval functions in [26, 123, 215, 231].

Given the varied sources of information one could leverage, the subsequent question is how the information is exactly utilized, which will be made clear next.

## 2.5 Learning for tag relevance

This section presents the second dimension of the taxonomy, elaborating on various algorithms for tag relevance learning. Depending on whether the tag relevance learning process is transductive, i.e., producing tag relevance scores without distinction as training and testing, we divide existing works into transduction-based and induction-based. Since the latter produces rules or models that are directly applicable to a novel instance [142], it has a better scalability for large-scale data compared to its transductive counterpart. Depending on whether an explicit model, let it be discriminative or generative,

Table 2.1: The taxonomy of methods for tag relevance learning, organized along the *Media* and *Learning* dimensions of Fig. 2.1. Methods for which we provide an experimental evaluation in the next chapter are indicated in **bold font**.

		<b>Learning</b>		
<b>Media</b>		<i>Instance-based</i>	<i>Model-based</i>	<i>Transduction-based</i>
		<b>Sigurbjörnsson et al. [174]</b>		
<i>tag</i>		Sun et al. [179] <b>Zhu et al. [229]</b>	Xu et al. [215]	–
			Wu et al. [212]	
		<b>Liu et al. [123]</b>	<b>Guillaumin et al. [68]</b>	<b>Zhu et al. [228]</b>
		<b>Makadia et al. [134]</b>	Verbeek et al. [194]	Wang et al. [204]
		Tang et al. [181]	Liu et al. [122]	Li et al. [119]
		Wu et al. [213]	Ma et al. [131]	Zhuang et al. [231]
		Yang et al. [218]	Liu et al. [125]	Richter et al. [163]
		Truong et al. [187]	Duan et al. [44]	Kuo et al. [98]
<i>tag + image</i>		Qi et al. [156]	Feng et al. [48]	Liu et al. [128]
		Lin et al. [121]	Srivastava et al. [178]	Gao et al. [55]
		Lee et al. [104]	<b>Chen et al. [26]</b>	Wu et al. [211]
		Uricchio et al. [192]	Lan et al. [99]	Yang et al. [219]
		Zhu et al. [230]	<b>Li et al. [111]</b>	Feng et al. [50]
		Ballan et al. [8]	Li et al. [118]	Xu et al. [216]
		Pereira et al. [152]	Wang et al. [203]	
			Niu et al. [151]	
			Sawant et al. [170]	
		<b>Li et al. [113]</b>	Li et al. [110]	
		Kennedy et al. [92]	McAuley et al. [135]	<b>Sang et al. [168]</b>
<i>tag + image + user</i>		Li et al. [114]	Kim et al. [94]	Sang et al. [169]
		Znaidia et al. [233]	McParlane et al. [137]	Qian et al. [158]
		Liu et al. [127]	Ginsca et al. [58]	
			Ballan et al. [87]	

is built, a further division for the induction-based methods can be made: instance-based algorithms and model-based algorithms. Consequently, we divide existing works into the following three exclusive groups: 1) instance-based, 2) model-based, and 3) transduction-based.

### 2.5.1 Instance-based

This class of methods does not perform explicit generalization but, instead, compares new test images with training instances. It is called instance-based because it constructs hypotheses directly from the training instances themselves. These methods are non parametric and the complexity of the learned hypotheses grows as the amount of training data increases. The neighbor voting algorithm [113] and its variants [92, 104, 114, 187, 230] estimate the relevance of a tag  $t$  with respect to an image  $x$  by counting the occurrence of  $t$  in annotations of the visual neighbors of  $x$ . The visual neighborhood is created using features obtained from early-fusion of global features [113], distance metric learning to combine local and global features [194, 213], cross modal learning of tags and image features [8, 152, 156], and fusion of multiple single-feature learners [114]. While the standard neighbor voting algorithm [113] simply let the neighbors vote equally, efforts have been made to (heuristically) weight neighbors in terms of their importance. For instance, in [104, 187] the visual similarity is used as the weights. As an alternative to such a heuristic strategy, [230] models the relationships among the neighbors by constructing a directed voting graph, wherein there is a directed edge from image  $x_i$  to image  $x_j$  if  $x_i$  is in the  $k$  nearest neighbors of  $x_j$ . Subsequently an adaptive random walk is conducted over the voting graph to estimate the tag relevance. However, the performance gain obtained by these weighting strategies appears to be limited [230]. The kernel density estimation technique used in [123] can be viewed as another form of weighted voting, but the votes come from images labeled with  $t$  instead of the visual neighbors. [218] further considers the distance of the test image to images not labeled with  $t$ . In order to eliminate semantically unrelated samples in the neighborhood, sparse reconstruction from a  $k$ -nearest neighborhood is used in [181, 182]. In [121], with intention of recovering missing tags by matrix reconstruction, the image and tag modalities are separately exploited in parallel to produce a new candidate image-tag association matrix each. Then, the two resultant tag relevance scores are linearly combined to produce the final tag relevance scores. To address the incompleteness of tags associated

with the visual neighbors, [233] proposes to enrich these tags by exploiting tag co-occurrence in advance to neighbor voting.

### 2.5.2 Model-based

This class of tag relevance learning algorithms puts their foundations on parameterized models learned from the training media. Notice that the models can be tag-specific or holistic for all tags. As an example of holistic modeling, a topic model approach is presented in [203] for tag refinement, where a hidden topic layer is introduced between images and tags. Consequently, the tag relevance function is implemented as the dot product between the topic vector of the image and the topic vector of the tag. In particular, the authors extend the Latent Dirichlet Allocation model [18] to force images with similar visual content to have similar topic distribution. According to their experiments [203], however, the gain of such a regularization appears to be marginal compared to the standard Latent Dirichlet Allocation model. [118] first finds embedding vectors of training images and tags using the image-tag association matrix of  $\mathcal{S}$ . The embedding vector of a test image is obtained by a convex combination of the embedding vectors of its neighbors retrieved in the original visual feature space. Consequently, the relevance score is computed in terms of the Euclidean distance between the embedding vectors of the test image and the tag.

For tag-specific modeling, linear SVM classifiers trained on features augmented by pre-trained classifiers of popular tags are used in [26] for tag retrieval. Fast intersection kernel SVMs trained on selected relevant positive and negative examples are used in [111]. A bag-based image reranking framework is introduced in [44], where pseudo relevant images retrieved by tag matching are partitioned into clusters by using visual and textual features. Then, by treating each cluster as a bag and images within the cluster as its instances, multiple instance learning [2] is employed to learn multiple-instance SVMs per tag. Viewing the social tags of a test image as ground truth, a multi-modal tag suggestion method based on both tags and visual correlation is introduced in [212]. Each modality is used to generate a ranking feature, and the tag relevance function is a combination of these ranking features, with the combination weights learned online by the RankBoost algorithm [53]. In [68, 194], logistic regression models are built per tag to promote rare tags. In a similar spirit to [111], [226] learns an ensemble of SVMs by treating tagged images as positive training examples and untagged

images as candidate negative training examples. Using the ensemble to classify image regions generated by automated image segmentation, the authors assign tags at the image level and the region level simultaneously.

### 2.5.3 Transduction-based

This class of methods consists in procedures that evaluate tag relevance for a given image-tag pair of a set of images by minimizing some specific cost function. Given an initial image-tag association matrix  $D$ , the output of the procedure is a new matrix  $\hat{D}$  the elements of which are taken as tag relevance scores. Due to this formulation, no explicit form of the tag relevance function exists nor any distinction between training and test sets [86]. If novel images are added to the initial set, minimization of the cost function needs to be re-computed.

The majority of transduction-based approaches are founded on matrix factorization [50, 89, 128, 168, 211, 216, 228]. In [231] the objective function is a linear combination of the difference between  $\hat{D}$  and the matrix of image similarity, the distortion between  $\hat{D}$  and the matrix of tag similarity, and the difference between  $\hat{D}$  and  $D$ . A stochastic coordinate descent optimization is applied to a randomly chosen row of  $\hat{D}$  per iteration. In [228], considering the fact that  $D$  is corrupted with noise derived by missing or over-personalized tags, robust principal component analysis with laplacian regularization is applied to recover  $\hat{D}$  as a low-rank matrix. In [211],  $\hat{D}$  is regularized such that the image similarity induced from  $\hat{D}$  is consistent with the image similarity computed in terms of low-level visual features, and the tag similarity induced from  $\hat{D}$  is consistent with the tag correlation score computed in terms of tag co-occurrence. In [216], it is proposed to re-weight the penalty term of each image-tag pair by their relevance score, which is estimated by a linear fusion of tag-based and content-based relevance scores. To incorporate the user element, [168] extends  $D$  to a three-way tensor with tag, image, and user as each of the ways. A core tensor and three matrices representing the three media, obtained by Tucker decomposition [188], are multiplied to construct  $\hat{D}$ .

As an alternative approach, in [50] it is assumed that the tags of an image are drawn independently from a fixed but unknown multinomial distribution. Estimation of this distribution is implemented by maximum likelihood with low-rank matrix recovery and laplacian regularization like [228].

Graph-based label propagation is another type of transduction-based

methods. In [98, 163, 204], the image-tag pairs are represented as a graph in which each node corresponds to a specific image and the edges are weighted according to a multi-modal similarity measure. Viewing the top ranked examples in the initial search results as positive instances, tag refinement is implemented as a semi-supervised labeling process by propagating labels from the positive instances to the remaining examples using random walk. While the edge weights are fixed in the above works, [55] argues that fixing the weights could be problematic, because tags found to be discriminative in the learning process should adaptively contribute more to the edge weights. In that regard, the hypergraph learning algorithm [227] is exploited and weights are optimized by minimizing a joint loss function which considers both the graph structure and the divergence between the initial labels and the learned labels. In [130], the hypergraph is embedded into a lower-dimension space by hypergraph Laplacian.

Comparing the three groups of methods for learning tag relevance, an advantage of instance-based methods against the other two groups is their flexibility to adapt to previously unseen images and tags. They may simply add new training images into  $\mathcal{S}$  or remove outdated ones. The advantage however comes with a price that  $\mathcal{S}$  has to be maintained, a non-trivial task given the increasing amount of training data available. Also, the computational complexity and memory footprint grow linearly with respect to the size of  $\mathcal{S}$ . In contrast, model-based methods could be more swift, especially when linear classifiers are used, as the training data is compactly represented by a fixed number of models. As the imagery of a given tag may evolve, re-training is required to keep the models up-to-date.

Different from instance-based and model-based learning where individual tags are considered independently, transduction-based learning methods via matrix factorization can favorably exploit inter-tag and inter-image relationships. However, their ability to deal with the extremely large number of social images is a concern. For instance, the use of Laplacian graphs results in a memory complexity of  $O(|\mathcal{S}|^2)$ . The accelerated proximal gradient algorithm used in [228] requires Singular Value Decomposition, which is known to be an expensive operation. The Tucker decomposition used in [168] has a cubic computational complexity with respect to the number of training samples. We notice that some engineering tricks have been considered in these works, which alleviate the scalability issue to some extent. In [231], for instance, clustering is conducted in advance to divide  $\mathcal{S}$  into much smaller subsets,

and the algorithm is applied to these subsets, separately. By making the Laplacian more sparse by retaining only the  $k$  nearest neighbors [168, 228], the memory footprint can be reduced to  $O(k \cdot |\mathcal{S}|)$ , with the cost of performance degeneration. Perhaps due to the scalability concern, works resorting to matrix factorization tend to experiment with a dataset of relatively small scale.

## 2.6 Auxiliary components

The *Filter* and the *Precompute* component are auxiliary components that may sustain and improve tag relevance learning.

*Filter.* As social tags are known to be subjective and overly personalized, removing personalized tags appears to be a natural and simple way to improve the tagging quality. This is usually the first step performed in the framework for tag relevance learning. Although there is a lack of golden criteria to determine which tags are personalized, a popular strategy is to exclude tags which cannot be found in the WordNet ontology [26, 110, 228, 229] or a Wikipedia thesaurus [123]. Tags with rare occurrence, say appearing less than 50 times, are discarded in [194, 228]. For methods that directly work on the image-tag association matrix [121, 168, 211, 228], reducing the size of the vocabulary in terms of tag occurrence is an important prerequisite to keep the matrix in a manageable scale. Observing that images tagged in a batch manner are often nearly duplicate and of low tagging quality, batch-tagged images are excluded in [116]. Since relevant tags may be missing from user annotations, the negative tags that are semantically similar or co-occurring with positive ones are discarded in [168]. As the above strategies do not take the visual content into account, they cannot handle situations where an image is incorrectly labeled with a valid and frequently used tag, say ‘dog’. In [112], tag relevance scores are assigned to each image in  $\mathcal{S}$  by running the neighbor voting algorithm [113], while in [111], the semantic field algorithm [229] is further added to select relevant training examples. In [158], the annotation of the training media is enriched by a random walk.

*Precompute.* The precompute component is responsible for the generation of the prior information that is jointly used with the refined training media  $\hat{\mathcal{S}}$  in learning. For instance, global statistics and external resources can be used to synthesize new prior knowledge useful in learning. The prior information commonly used is tag statistics in  $\mathcal{S}$ , including tag occurrence

and tag co-occurrence. Tag occurrence is used in [113] as a penalty to suppress overly frequent tags. Measuring the semantic similarity between two tags is important for tag relevance learning algorithms that exploit tag correlations. While linguistic metrics as those derived from WordNet were used before the proliferation of social media [85,200], they do not directly reflect how people tag images. For instance, tag ‘sunset’ and tag ‘sea’ are weakly related according to the WordNet ontology, but they often appear together in social tagging as many of the sunset photos are shot around seashores. Therefore, similarity measures that are based on tag statistics computed from many socially tagged images are in dominant use. Sigurbjörnsson and van Zwol utilized the Jaccard coefficient and a conditional tag probability in their tag suggestion system [174], while Liu *et al.* used normalized tag co-occurrence [128]. To better capture the visual relationship between two tags, Wu *et al.* proposed the Flickr distance [210]. The authors represent each tag by a visual language model, trained on bag of visual words features of images labeled with this tag. The Flickr distance between two tags is computed as the Jensen-Shannon divergence between the corresponding models. Later, Jiang *et al.* introduced the Flickr context similarity, which also captures the visual relationship between two tags, but without the need of the expensive visual modeling [83]. The trick is to compute the Normalized Google Distance [33] between two tags, but with tag statistics acquired from Flickr image collections instead of Google indexed web pages. For its simplicity and effectiveness, we observe a prevalent use of the Flickr context similarity in the literature [55, 111, 123, 157, 204, 228, 229, 231].

## 2.7 Conclusions

We presented a survey on image tag assignment, refinement and retrieval, with the hope of illustrating connections and difference between the many methods and their applicabilities, and consequently helping the interested audience to either pick up an existing method or devise a method of their own given the data at hand. As the topics are being actively studied, inevitably this survey will miss some papers. Nevertheless, it provides a unified view of many existing works, and consequently eases the effort of placing future works in a proper context, both theoretically and experimentally. Based on the key observation that all works rely on tag relevance learning as the common ingredient, exiting works, which vary in terms of their methodolo-



gies and target tasks, are interpreted in a unified framework. Consequently, a two-dimensional taxonomy has been developed, allowing us to structure the growing literature in light of what information a specific method exploits and how the information is leveraged in order to produce their tag relevance scores.



# Chapter 3

## A new Experimental Protocol

*In this chapter we propose an evaluation test-bed for the three linked tasks of Assignment, Refinement and Retrieval. Training sets of varying sizes and three test datasets are considered to evaluate methods of varied learning complexity. A selected set of eleven representative works have been implemented and evaluated. Several overall patterns are recognized. To highlight the advantages of socially tagged training sets, an empirical evaluation between ImageNet and the proposed Flickr-based training sets is reported.*<sup>1</sup>

### 3.1 Introduction

In spite of the expanding literature, there is a lack of consensus on the performance of the individual methods. This is largely due to the fact that existing works either use homemade data, see [26, 55, 123, 204], which are not publicly accessible, or use selected subsets of benchmark data, e.g. as in [50, 168, 228]. As a consequence, the performance scores reported in the literature are not comparable across the papers.

Benchmark data with manually verified labels is crucial for an objective evaluation. As Flickr has been well recognized as a profound manifestation of social image tagging, Flickr images act as a main source for benchmark construction. MIRFlickr from the Leiden University [75] and NUS-WIDE from

---

<sup>1</sup>Part of this chapter is submitted as “Socializing the Semantic Gap: A Comparative Survey on Image Tag Assignment, Refinement and Retrieval” to *ACM Computing Surveys*.

the National University of Singapore [32] are the two most popular Flickr-based benchmark sets for social image tagging and retrieval, as demonstrated by the number of citations. On the use of the benchmarks, one typically follows a single-set protocol, that is, learning the underlying tag relevance function from the training part of a chosen benchmark set, and evaluating it on the test part. Such a protocol is inadequate given the dynamic nature of social media, which could easily make an existing benchmark set outdated. For any method targeting at social images, a cross-set evaluation is necessary to test its generalization ability, which is however overlooked in the literature.

Another desirable property is the capability to learn from the increasing amounts of socially tagged images. While existing works mostly use training data of a fixed scale, this property has not been well evaluated.

Following these considerations, we present a new experimental protocol, wherein training and test data from distinct research groups are chosen for evaluating a number of representative works in the cross-set scenario. Training sets with their size ranging from 10k to one million images are constructed to evaluate methods of varied complexity. To the best of our knowledge, such a comparison between many methods on varied scale datasets with a common experimental setup has not been conducted before. For the sake of experimental reproducibility, all data and code is made available online at [www.micc.unifi.it/tagsurvey/](http://www.micc.unifi.it/tagsurvey/).

## 3.2 Datasets

We describe the training media  $\mathcal{S}$  and the test media  $\mathcal{X}$  as follows, with basic data characteristics and their usage summarized in Table 3.1.

*Training media  $\mathcal{S}$ .* We use a set of 1.2 million Flickr images collected by the University of Amsterdam [116], by using over 25,000 nouns in WordNet as queries to uniformly sample images uploaded between 2006 and 2010. Based on our observation that batch-tagged images, namely those labeled with the same tags by the same user, tend to be near duplicate, we have excluded these images beforehand. Other than this, we do not perform near-duplicate image removal. To meet with methods that cannot handle large data, we created two random subsets from the entire training sets, resulting in three training sets of varied sizes, termed as Train10k, Train100k, and Train1m, respectively.

Table 3.1: Our proposed experimental protocol instantiates the *Media* and *Tasks* dimensions of Fig. 2.1 with three training sets and three test sets for tag assignment, refinement and retrieval. Note that the training sets are socially tagged, they have no ground truth available for any tag.

Media	Media characteristics				Tasks		
	# images	# tags	# users	# test tags	assignment	refinement	retrieval
<i>Training media S:</i>							
Train10k	10,000	41,253	9,249	–	✓	✓	✓
Train100k	100,000	214,666	68,215	–	✓	✓	✓
Train1m [116]	1,198,818	1,127,139	347,369	–	✓	✓	✓
<i>Test media X:</i>							
MIRFlickr [75]	25,000	67,389	9,862	14	✓	✓	–
Flickr51 [204]	81,541	66,900	20,886	51	–	–	✓
NUS-WIDE [32]	259,233	355,913	51,645	81	✓	✓	✓

*Test media X.* We use MIRFlickr [75] and NUS-WIDE [32] for tag assignment and refinement, as in [192, 194, 228] and [135, 181, 192, 228] respectively. We use NUS-WIDE for evaluating tag retrieval as in [108, 179]. In addition, for retrieval we collected another test set namely Flickr51 contributed by Microsoft Research Asia [55, 204]. The MIRFlickr set contains 25,000 images with ground truth available for 14 tags. The NUS-WIDE set contains 259,233 images, with ground truth available for 81 tags. The Flickr51 set consists of 81,541 Flickr images with partial ground truth provided for 55 test tags. Among the 55 tags, there are 4 tags which either have zero occurrence in our training data or have no correspondence in WordNet, so we ignore them. Differently from the binary judgments in NUS-WIDE, Flickr51 provides graded relevance, with 0, 1, and 2 to indicate irrelevant, relevant, and very relevant, respectively. Moreover, the set contains several ambiguous tags such as ‘apple’ and ‘jaguar’, where relevant instances could exhibit completely different imagery, e.g., Apple computers versus fruit apples. Following the original intention of the datasets, we use MIRFlickr and NUS-WIDE for evaluating tag assignment and tag refinement, and Flickr51 and NUS-WIDE for tag retrieval. For all the three test sets, we use the full dataset for testing.

Although the training and test media are all from Flickr, they were collected independently, and consequently they have a relatively small amount of images overlapped with each other, as shown in Table 3.2.

Table 3.2: Data overlap between Train1M and the three test sets, measured in terms of the number of shared images, tags, and users, respectively. Tag overlap is counted on the top 1,000 most frequent tags. As the original photo ids of MIRFlickr have been anonymized, we cannot check image overlap between this dataset and Train1M.

Test media	Overlap with Train1M		
	# images	# tags	# users
MIRFlickr	–	693	6,515
Flickr51	730	538	14,211
NUS-WIDE	7,975	718	38,481

### 3.3 Implementation and Evaluation

This section describes common implementations applicable to all the three tasks, including the choice of visual features and tag preprocessing. Implementations that are applied uniquely to single tasks will be described in the coming sections.

*Visual features.* Two types of features are extracted to provide insights of the performance improvement achievable by appropriate feature selection: the classical bag of visual words (BoVW) and the current state of the art deep learning based features extracted from Convolutional Neural Networks (CNN). The BoVW feature is extracted by the color descriptor software [193]. SIFT descriptors are computed at dense sampled points, at every 6 pixels for two scales. A codebook of size 1,024 is created by K-means clustering. The SIFTs are quantized by the codebook using hard assignment, and aggregated by sum pooling. In addition, we extract a compact 64-d global feature [106], combining a 44-d color correlogram, a 14-d texture moment, and a 6-d RGB color moment, to compensate the BoVW feature. The CNN feature is extracted by the pre-trained VGGNet [175]. In particular, we adopt the 16-layer VGGNet, and take as feature vectors the last fully connected layer of ReLU activation, resulting in a feature vector of 4,096 dimensions per image. The BoVW feature is used with the  $l_1$  distance and the CNN feature is used with the cosine distance for their good performance.

*Vocabulary  $\mathcal{V}$ .* As what tags a person may use is meant to be open, the need of specifying a tag vocabulary is merely an engineering convenience.

For a tag to be meaningfully modeled, there has to be a reasonable amount of training images with respect to that tag. For methods where tags are processed independently from the others, the size of the vocabulary has no impact on the performance. In the other cases, in particular for transductive methods that rely on the image-tag association matrix, the tag dimension has to be constrained to make the methods runnable. In our case, for these methods a three-step automatic cleaning procedure is performed on the training datasets. First, all the tags are lemmatized to their base forms by the NLTK software [17]. Second, tags not defined in WordNet are removed. Finally, in order to avoid insufficient sampling, we remove tags that cannot meet a threshold on tag occurrence. The thresholds are empirically set as 50, 250, and 750 for Train10k, Train100k, and Train1m, respectively, in order to have a linear increase in vocabulary size versus a logarithmic increase in the number of labeled images. This results in a final vocabulary of 237, 419, and 1,549 tags, respectively, with all the test tags included. Note that these numbers of tags are larger than the number of tags that can be actually evaluated. This allows to build a unified learning method that is more handy for cross-dataset evaluation and exploit inter-tag relationships.

### 3.3.1 Evaluating tag assignment

*Evaluation criteria.* A good method for tag assignment shall rank relevant tags before irrelevant tags for a given test image. Moreover, with the assigned tags, relevant images shall be ranked before irrelevant images for a given test tag. We therefore use the image-centric Mean image Average Precision (MiAP) to measure the quality of tag ranking, and the tag-centric Mean Average Precision (MAP) to measure the quality of image ranking. Let  $m_{gt}$  be the number of ground-truthed test tags, which is 14 for MIRFlickr and 81 for NUS-WIDE. The image-centric Average Precision of a given test image  $x$  is computed as

$$iAP(x) := \frac{1}{R} \sum_{j=1}^{m_{gt}} \frac{r_j}{j} \delta(x, t_j), \quad (3.1)$$

where  $R$  is the number of relevant tags of the given image,  $r_j$  is the number of relevant tags in the top  $j$  ranked tags, and  $\delta(x_i, t_j) = 1$  if tag  $t_j$  is relevant and 0 otherwise. MiAP is obtained by averaging  $iAP(x)$  over the test images.

The tag-centric Average Precision of a given test tag  $t$  is computed as

$$AP(t) := \frac{1}{R} \sum_{i=1}^n \frac{r_i}{i} \delta(x_i, t), \quad (3.2)$$

where  $R$  is the number of relevant images for the given tag, and  $r_i$  is the number of relevant images in the top  $i$  ranked images. MAP is obtained by averaging  $AP(t)$  over the test tags.

The two metrics are complementary to some extent. Since MiAP is averaged over images, each test image contributes equally to MiAP, as opposed to MAP where each tag contributes equally. Consequently, MiAP is biased towards frequent tags, while MAP can be easily affected by the performance of rare tags, especially when  $m_{gt}$  is relatively small.

*Baseline.* Any method targeting at tag assignment shall be better than a random guess, which simply returns a random set of tags. The RandomGuess baseline is obtained by computing MiAP and MAP given the random prediction, which is run 100 times with the resulting scores averaged.

### 3.3.2 Evaluating tag refinement

*Evaluation criteria.* As tag refinement is also meant for improving tag ranking and image ranking, it is evaluated by the same criteria, i.e., MiAP and MAP, as used for tag assignment.

*Baseline.* A natural baseline for tag refinement is the original user tags assigned to an image, which we term as UserTags.

### 3.3.3 Evaluating tag retrieval

*Evaluation criteria.* To compare methods for tag retrieval, for each test tag we first conduct tag-based image search to retrieve images labeled with that tag, and then sort the images by the tag relevance scores. We use MAP to measure the quality of the entire image ranking. As users often look at the top ranked results and hardly go through the entire list, we also report Normalized Discounted Cumulative Gain (NDCG), commonly used to evaluate the top few ranked results of an information retrieval system [78]. Given a test tag  $t$ , its NDCG at a particular rank position  $h$  is defined as:

$$NDCG_h(t) := \frac{DCG_h(t)}{IDCG_h(t)}, \quad (3.3)$$



$$DCG_h(t) = \sum_{i=1}^h \frac{2^{rel_i} - 1}{\log_2(i + 1)}, \quad (3.4)$$

where  $rel_i$  is the graded relevance of the result at position  $i$ , and  $IDCG_h$  is the maximum possible  $DCG$  till position  $h$ . We set  $h$  to be 20, which corresponds to a typical number of search results presented on the first two pages of a web search engine. Similar to MAP,  $NDCG_{20}$  of a specific method on a specific test set is averaged over the test tags of that test set.

*Baselines.* When searching for relevant images for a given tag, it is natural to ask how much a specific method gains compared to a baseline system which simply returns a random subset of images labeled with that tag. Similar to the refinement baseline, we also denote this baseline as UserTags, as both of them purely use the original user tags. For each test tag, the test images labeled with this tag are sorted at random, and MAP and  $NDCG_{20}$  are computed accordingly. The process is executed 100 times, and the average score over the 100 runs is reported.

The number of tags per image is often included for image ranking in previous works [123, 215]. Hence, we build another baseline system, denoted as TagNum, which sort images in ascending order by the number of tags per image. The third baseline, denoted as TagPosition, is from [179], where the relevance score of a tag is determined by its position in the original tag list uploaded by the user. More precisely, the score is computed as  $1 - position(t)/l$ , where  $l$  is the tag number.

## 3.4 Methods under analysis

Despite the rich literature, most works do not provide code. An exhaustive evaluation covering all published methods is impractical. We have to leave out methods that do not show significant improvements or novelties w.r.t. the seminal papers in the field, and methods that are difficult to replicate with the same mathematical preciseness as intended by their developers. We drive our choice by the intention to cover methods that aim for each of the three tasks, exploiting varied modalities by distinct learning mechanisms. Eventually we evaluate 11 representative methods. For each method we analyze its scalability in terms of both computation and memory. Our analysis leaves out operations that are independent of specific tags and thus only need to be executed once in an offline manner, such as visual feature

extraction, tag preprocessing, prior information precomputing, and filtering. Main properties of the methods are summarized in table 3.3. Concerning the choices of parameters, we adopt what the original papers recommend. When no recommendation is given for a specific method, we try a range of values to our best understanding, and choose the parameters that yield the best overall performance.

### 3.4.1 SemanticField

SemanticField [229] measures tag relevance in terms of an averaged semantic similarity between the tag and the other tags assigned to the image:

$$f_{SemField}(x, t) := \frac{1}{l_x} \sum_{i=1}^{l_x} sim(t, t_i), \quad (3.5)$$

where  $\{t_1, \dots, t_{l_x}\}$  is a list of  $l_x$  social tags assigned to the image  $x$ , and  $sim(t, t_i)$  denotes a semantic similarity between two tags. SemanticField explicitly assumes that several tags are associated to visual data and their coexistence is accounted in the evaluation of tag relevance. Following [229], the similarity is computed by combining the Flickr context similarity and the WordNet Wu-Palmer similarity [214]. The WordNet based similarity exploits path length in the WordNet hierarchy to infer tag relatedness. We make a small revision of [229], i.e. combining the two similarities by averaging instead of multiplication, because the former strategy produces slightly better results. SemanticField requires no training except for computing tag-wise similarity, which can be computed offline and is thus omitted. Having all tag-wise similarities in memory, applying Eq. (3.5) requires  $l_x$  table lookups per tag. Hence, the computational complexity is  $O(m \cdot l_x)$ , and  $O(m^2)$  for memory.

### 3.4.2 TagRanking

The tag ranking algorithm [123] consists of two steps. Given an image  $x$  and its tags, the first step produces an initial tag relevance score for each of the tags, obtained by (Gaussian) kernel density estimation on a set of  $\bar{n} = 1,000$  images labeled with each tag, separately. Secondly, a random walk is performed on a tag graph where the edges are weighted by a tag-wise similarity. We use the same similarity as in SemanticField. Notice that when

applied for tag retrieval, the algorithm uses the rank of  $t$  instead of its score, i.e.,

$$f_{\text{TagRanking}}(x, t) = -\text{rank}(t) + \frac{1}{l_x}, \quad (3.6)$$

where  $\text{rank}(t)$  returns the rank of  $t$  produced by the tag ranking algorithm. The term  $\frac{1}{l_x}$  is a tie-breaker when two images have the same tag rank. Hence, for a given tag  $t$ , TagRanking cannot distinguish relevant images from irrelevant images if  $t$  is the sole tag assigned to them. It explicitly exploits the coexistence of several tags per image. TagRanking has no learning stage. To derive tag ranks for Eq. 3.6, the main computation is the kernel density estimation on  $\bar{n}$  socially-tagged examples for each tag, followed by an  $L$  iteration random walk on the tag graph of  $m$  nodes. All this results in a computation cost of  $O(m \cdot d \cdot \bar{n} + L \cdot m^2)$  per test image. Because the two steps are executed sequentially, the corresponding memory cost is  $O(\max(d\bar{n}, m^2))$ .

### 3.4.3 KNN

This algorithm [134] estimates the relevance of a given tag with respect to an image by first retrieving  $k$  nearest neighbors from  $\mathcal{S}$  based on a visual distance  $d$ , and then counting the tag occurrence in associated tags of the neighborhood. In particular, KNN builds  $f_{\Phi}(x, t; \Theta)$  as:

$$f_{KNN}(x, t) := k_t, \quad (3.7)$$

where  $k_t$  is the number of images with  $t$  in the visual neighborhood of  $x$ . The instance-based KNN requires no training. The main computation of  $f_{KNN}$  is to find  $k$  nearest neighbors from  $\mathcal{S}$ , which has a complexity of  $O(d \cdot |\mathcal{S}| + k \cdot \log |\mathcal{S}|)$  per test image, and a memory footprint of  $O(d \cdot |\mathcal{S}|)$  to store all the  $d$ -dimensional feature vectors. It is worth noting that these complexities are drawn from a straightforward implementation of  $k$ -nn search, and can be substantially reduced by employing more efficient search techniques, c.f. [79]. Accelerating KNN by the product quantization technique [79] imposes an extra training step, where one has to construct multiple vector quantizers by K-means clustering, and further use the quantizers to compress the original feature vector into a few codes.

### 3.4.4 TagVote

The TagVote [113] algorithm estimates the relevance of a tag  $t$  w.r.t. an image  $x$  by counting the occurrence frequency of  $t$  in social annotations of

the visual neighbors of  $x$ . Differently from KNN, TagVote exploits the user element in the social framework and introduces a unique-user constraint on the neighbor set to make the voting result more objective. Each user has at most one image in the neighbor set. Moreover, TagVote also takes into account tag prior frequency to suppress over frequent tags. In particular, the TagVote algorithm builds  $f_{\Phi}(x, t; \Theta)$  as

$$f_{TagVote}(x, t) := k_t - k \frac{n_t}{|\mathcal{S}|}, \quad (3.8)$$

where  $n_t$  is the number of images labeled with  $t$  in  $\mathcal{S}$ . Following [113], we set  $k$  to be 1,000 for both KNN and TagVote. TagVote has the same order of complexity as KNN.

### 3.4.5 TagProp

TagProp [68, 194] employs neighbor voting plus distance metric learning. A probabilistic framework is proposed where the probability of using images in the neighborhood is defined based on rank or distance-based weights. TagProp builds  $f_{\Phi}(x, t; \Theta)$  as:

$$f_{TagProp}(x, t) := \sum_j^k \pi_j \cdot \mathbf{I}(x_j, t), \quad (3.9)$$

where  $\pi_j$  is a non-negative weight indicating the importance of the  $j$ -th neighbor  $x_j$ , and  $\mathbf{I}(x_j, t)$  returns 1 if  $x_j$  is labeled with  $t$ , and 0 otherwise. Following [194], we use  $k = 1,000$  and the rank-based weights, which showed similar performance to the distance-based weights. Differently from TagVote that uses tag prior to penalize frequent tags, TagProp promotes rare tags and penalizes frequent ones by training a logistic model per tag upon  $f_{TagProp}(x, t)$ . The use of the logistic model makes TagProp a model-based method. In contrast to KNN and TagVote wherein visual neighbors are treated equally, TagProp employs distance metric learning to re-weight the neighbors, yielding a learning complexity of  $O(l \cdot m \cdot k)$  where  $l$  is the number of gradient descent iterations it needs (typically less than 10). TagProp maintains  $2m$  extra parameters for the logistic models, though their storage cost is ignorable compared to the visual features. Therefore, running Eq. (3.9) has the same order of complexity as KNN and TagVote.

### 3.4.6 TagCooccur

While both SemanticField and TagCooccur are tag-based, the main difference lies in how they compute the contribution of a specific tag to the test tag’s relevance score. Different from SemanticField which uses tag similarities, TagCooccur [174] uses the test tag’s rank in the tag ranking list created by sorting all tags in terms of their co-occurrence frequency with the tag in a social framework. In addition, TagCooccur takes into account the stability of the tag, measured by its frequency. The method is implemented as

$$f_{tagcooccur}(x, t) = \text{descript}(t) \sum_{i=1}^{l_x} \text{vote}(t_i, t) \cdot \text{rank-promo}(t_i, t) \cdot \text{stability}(t_i), \quad (3.10)$$

where  $\text{descript}(t)$  is to damp the contribution of tags with a very high-frequency,  $\text{rank-promo}(t_i, t)$  measures the rank-based contribution of  $t_i$  to  $t$ ,  $\text{stability}(t_i)$  for promoting tags for which the statistics are more stable, and  $\text{vote}(t_i, t)$  is 1 if  $t$  is among the top 25 ranked tags of  $t_i$ , and 0 otherwise. TagCooccur has the same order of complexity as SemanticField.

### 3.4.7 TagCooccur+

TagCooccur+ [113] is proposed to improve TagCooccur by adding the visual content. This is achieved by multiplying  $f_{tagcooccur}(x, t)$  with a content-based term, i.e.,

$$f_{tagcooccur+}(x, t) = f_{tagcooccur}(x, t) \cdot \frac{k_c}{k_c + r_c(t) - 1}, \quad (3.11)$$

where  $r_c(t)$  is the rank of  $t$  when sorting the vocabulary by  $f_{TagVote}(x, t)$  in descending order, and  $k_c$  is a positive weighting parameter, which is empirically set to 1. While TagCooccur+ is grounded on TagCooccur and TagVote, the complexity of the former is ignorable compared to the latter, so the complexity of TagCooccur+ is the same as KNN.

### 3.4.8 TagFeature

The basic idea of TagFeature [26] is to enrich image features by adding an extra tag feature. It thus relies on the possible presence of several tags per image in the training set. In particular, a tag vocabulary that consists of

$d'$  most frequent tags in  $\mathcal{S}$  is constructed first. Then, for each tag a two-class linear SVM classifier is trained using LIBLINEAR [47]. The positive training set consists of  $p$  images labeled with the tag in  $\mathcal{S}$ , and the same amount of negative training examples are randomly sampled from images not labeled with the tag. The probabilistic output of the classifier, obtained by the Platt’s scaling [120], corresponds to a specific dimension in the tag feature. By concatenating the tag and visual features, an augmented feature of  $d + d'$  dimension is obtained. For a test tag  $t$ , its tag relevance function  $f_{\text{TagFeature}}(x, t)$  is obtained by re-training an SVM classifier using the augmented feature. The linear property of the classifier allows us to first sum up all the support vectors into a single vector and consequently to classify a test image by the inner product with this vector. That is,

$$f_{\text{TagFeature}}(x, t) := b + \langle x_t, x \rangle, \quad (3.12)$$

where  $x_t$  is the weighted sum of all support vectors and  $b$  the intercept. To build meaningful classifiers, we use tags that have at least 100 positive examples. While  $d'$  is chosen to be 400 in [26], the two smaller training sets, namely Train10k and Train100k, have 76 and 396 tags satisfying the above requirement. We empirically set  $p$  to 500, and do a random down-sampling if the amount of images for a tag exceeds this number. For TagFeature, learning a linear classifier for each tag from  $p$  positive and  $p$  negative examples requires  $O((d + d')p)$  in computation and  $O((d + d')p)$  in memory [47]. Running Eq. (3.12) for all the  $m$  tags and  $n$  images needs  $O(nm(d + d'))$  in computation and  $O(m(d + d'))$  in memory.

### 3.4.9 RelExample

Different from TagFeature [26] that learns from tagged images, RelExample [111] exploits positive and negative training examples which are deemed to be more relevant with respect to the test tag  $t$ . In particular, relevant positive examples are selected from  $\mathcal{S}$  by combining SemanticField and TagVote in a late fusion manner. For negative training example acquisition, they leverage Negative Bootstrap [115], a negative sampling algorithm which iteratively selects negative examples deemed most relevant for improving classification. A  $T$ -iteration Negative Bootstrap will produce  $T$  meta classifiers.

The corresponding tag relevance function is written as

$$f_{RelExample}(x, t) := \frac{1}{T} \sum_{l=1}^T (b_l + \sum_{j=1}^{n_l} \alpha_{l,j} \cdot y_{l,j} \cdot \mathcal{K}(x, x_{l,j})), \quad (3.13)$$

where  $\alpha_{l,j}$  is a positive coefficient of support vector  $x_{l,j}$ ,  $y_{l,j} \in \{-1, 1\}$  is class label, and  $n_l$  the number of support vectors in the  $l$ -th classifier. For the sake of efficiency, the kernel function  $\mathcal{K}$  is instantiated with the fast intersection kernel [132]. RelExample uses the same amount of positive training examples as TagFeature. The number of iterations  $T$  is empirically set to 10. For the SVM classifiers used in TagFeature and RelExample, the Platt’s scaling [120] is employed to convert prediction scores into probabilistic output. In RelExample, for each tag learning a histogram intersection kernel SVM has a computation cost of  $O(dp^2)$  per iteration, and  $O(Tdp^2)$  for  $T$  iterations. By jointly using the fast intersection kernel with a quantization factor of  $q$  [132] and model compression [115], an order of  $O(dq)$  is needed to keep all learned meta classifiers in memory. Since learning a new classifier needs a memory of  $O(dp)$ , the overall memory cost for training RelExample is  $O(dp + dq)$ . For each tag, model compression is applied to its learned ensemble in advance to running Eq. (3.13). As a consequence, the compressed classifier can be cached in an order of  $O(dq)$  and executed in an order of  $O(d)$ .

### 3.4.10 RobustPCA

RobustPCA [228] has been explicitly modeled to deal with a social framework, including noisy tags and several tags per image. On the base of robust principal component analysis [21], it factorizes the image-tag matrix  $D$  by a low rank decomposition with error sparsity. That is,

$$D = \hat{D} + E, \quad (3.14)$$

where the reconstructed  $\hat{D}$  has a low rank constraint based on the nuclear norm, and  $E$  is an error matrix with a  $\ell_1$ -norm sparsity constraint. Notice that the decomposition is not unique. So for a better solution, the decomposition process takes into account image affinities and tag affinities, by adding two extra penalties with respect to a Laplacian matrix  $L_i$  from the image affinity graph and another Laplacian matrix  $L_t$  from the tag affinity graph. Consequently, two hyper-parameters  $\lambda_1$  and  $\lambda_2$  are introduced to balance the error sparsity and the two Laplacian strengths. We follow the original paper

and set the two parameters by performing a grid search on the very same proposed range. As user tags are usually missing, the authors proposed a pre-processing step where  $D$  is reinitialized by a weighted KNN propagation based on the visual similarity. RobustPCA requires an iterative procedure based on the accelerated proximal gradient method with a quadratic convergence rate [228]. Each iteration spends the majority of the required time performing Singular Value Decomposition that, according to [62], has a well known complexity of  $O(cm^2n + c'n^3)$  where  $c, c'$  are constants. Regarding memory, it has a requirement of  $O(cn \cdot m + c' \cdot (n^2 + m^2))$  as it needs to process a full copy of  $D$  and Laplacians of images and labels.

### 3.4.11 TensorAnalysis

This method [168] has been explicitly designed for social frameworks. It explicitly considers ternary relationships between images, tags and user. User relationships are exploited by extending the image-tag association matrix to a binary user-image-tag tensor  $F \in \{0, 1\}^{|\mathcal{X}| \times |\mathcal{V}| \times |\mathcal{U}|}$ . The tensor is factorized by Tucker decomposition into a dense core  $C$  and three low rank matrices  $U, I, T$ , which correspond to the user, image, and tag modalities, respectively:

$$F = C \times_u U \times_i I \times_t T, \quad (3.15)$$

Here  $\times_k$  is the tensor product between a tensor and a matrix along dimension  $k$ . The idea is that  $C$  contains the interactions between modalities, while each low rank matrix represent the main components of each modality. Every modality has to be sized manually or by energy retention, adding three needed parameters  $R = (r_I, r_T, r_U)$ . The eventual tag relevance function is obtained after the optimization process by computing  $\hat{D} = C \times_i I \times_t T \times_u \mathbf{1}_{r_u}$ . Similar to RobustPCA, the decomposition in Eq. (3.15) is not unique and a better solution may be found regularizing the problem with a Laplacian built on a similarity graph for each modality, i.e.,  $L_i, L_t$ , and  $L_u$ , and a  $\ell_2$  regularizer on each factor i.e.  $C, U, I$  and  $T$ . For TensorAnalysis, the complexity is  $O(|P_1| \cdot (r_T \cdot m^2 + r_U \cdot r_I \cdot r_T))$ , proportional to the number  $P_1$  of tags asserted in  $D$  and the dimension of low rank  $r_U, r_I, r_T$  factors. The memory required is  $O(n^2 + m^2 + u^2)$  because of Laplacians of images, tags and users.



Table 3.3: Main properties of the eleven methods evaluated in this survey following the dimensions of Fig. 2.1. The computational and memory complexity of each method is based on processing  $n$  test images and  $m$  test tags by exploiting the training set  $\mathcal{S}$ .

Methods	Test Media	Task	Learning			
			Train Computation	Test Computation	Train Memory	Test Memory
<b>Instance-based:</b>						
SemanticField	tag	Retrieval	–	$O(nml_x)$	–	$O(m^2)$
TagCooccur	tag	Refinement	–	$O(nml_x)$	–	$O(m^2)$
		Retrieval				
TagRanking	tag + image	Retrieval	–	$O(n(md\bar{n} + Lm^2))$	–	$O(\max(d\bar{n}, m^2))$
KNN	tag + image	Assignment	–	$O(n(d \mathcal{S}  + k \log  \mathcal{S} ))$	–	$O(d \mathcal{S} )$
		Retrieval				
TagVote	tag + image	Assignment	–	$O(n(d \mathcal{S}  + k \log  \mathcal{S} ))$	–	$O(d \mathcal{S} )$
		Retrieval				
TagCooccur+	tag + image	Refinement	–	$O(n(d \mathcal{S}  + k \log  \mathcal{S} ))$	–	$O(d \mathcal{S} )$
		Retrieval				
<b>Model-based:</b>						
TagProp	tag + image	Assignment	$O(l \cdot m \cdot k)$	$O(n(d \mathcal{S}  + k \log  \mathcal{S} ))$	$O(d \mathcal{S}  + 2m)$	$O(d \mathcal{S}  + 2m)$
		Retrieval				
TagFeature	tag + image	Assignment	$O(m(d + d')p)$	$O(nm(d + d'))$	$O((d + d')p)$	$O(m(d + d'))$
		Retrieval				
RelExample	tag + image	Assignment	$O(mTdp^2)$	$O(dp + dq)$	$O(nmd)$	$O(mdq)$
		Retrieval				
<b>Transduction-based:</b>						
RobustPCA	tag + image	Refinement	$O(cm^2n + c'n^3)$		$O(cnm + c' \cdot (n^2 + m^2))$	
		Retrieval				
TensorAnalysis	tag + image + user	Refinement	$O( P_1  \cdot (r_T \cdot m^2 + r_U \cdot r_I \cdot r_T))$		$O(n^2 + m^2 + u^2)$	

### 3.4.12 Considerations

An overview of the methods analyzed is given Table 3.3. Among them, SemanticField, counting solely on the tag modality, has the best scalability with respect to both computation and memory. Among the instance-based methods, TagRanking, which works on selected subsets of  $\mathcal{S}$  rather than the entire collection, has the lowest memory request. When the number of tags to be modeled  $m$  is substantially smaller than the size of  $\mathcal{S}$ , the model-based methods require less memory and run faster in the test stage, but at the expense of SVM model learning in the training stage. The two transduction-based methods have limited scalability, and can operate only on small sized  $\mathcal{S}$ .

## 3.5 Evaluation

This section presents our evaluation of the 11 methods according to their applicability to the three tasks using the proposed experimental protocol, that is, KNN, TagVote, TagProp, TagFeature and RelExample for tag assignment (Section 3.5.1), TagCooccur, TagCooccur+, RobustPCA, and TensorAnalysis for tag refinement (Section 3.5.2), and all for tag retrieval (Section 3.5.3). For TensorAnalysis we were able to evaluate only tag refinement with BovW features on MIRFlickr with Train10k and Train100k. The reason for this exception is that our implementation of TensorAnalysis performs worse than the baseline. Consequently, the results of TensorAnalysis were kindly provided by the authors in the form of tag ranks. Since the provided tag ranks cannot be converted to image ranks, we could not compute MAP scores. Finally a comparison between our Flickr based training data and ImageNet is given in Section 3.5.4.

### 3.5.1 Tag assignment

Table 3.4 shows the tag assignment performance of KNN, TagVote, TagProp, TagFeature and RelExample. Their superior performance against the RandomGuess baseline shows that learning purely from social media is meaningful. TagVote and TagProp are the two best performing methods on both test sets. Substituting CNN for BovW consistently brings improvements for all methods.

Table 3.4: Evaluating methods for tag assignment. Given the same feature, bold values indicate top performers on individual test sets.

Method	MIRFlickr			NUS-WIDE		
	Train10k	Train100k	Train1m	Train10k	Train100k	Train1m
<i>MiAP scores:</i>						
RandomGuess	0.147	0.147	0.147	0.061	0.061	0.061
BovW + KNN	0.232	0.286	0.312	0.171	0.217	0.248
BovW + TagVote	0.276	0.310	<b>0.328</b>	0.183	0.231	0.259
BovW + TagProp	0.276	0.299	0.314	0.230	0.249	<b>0.268</b>
BovW + TagFeature	0.278	0.294	0.298	0.244	0.221	0.214
BovW + RelExample	0.284	0.309	0.303	0.257	0.233	0.245
CNN + KNN	0.326	0.366	0.379	0.315	0.343	0.376
CNN + TagVote	0.355	0.378	0.389	0.340	0.370	<b>0.396</b>
CNN + TagProp	0.373	0.384	<b>0.392</b>	0.366	0.376	0.380
CNN + TagFeature	0.359	0.378	0.383	0.367	0.338	0.373
CNN + RelExample	0.309	0.385	0.373	0.365	0.354	0.388
<i>MAP scores:</i>						
RandomGuess	0.072	0.072	0.072	0.023	0.023	0.023
BovW + KNN	0.231	0.282	0.336	0.094	0.139	0.185
BovW + TagVote	0.228	0.280	0.334	0.093	0.137	0.184
BovW + TagProp	0.245	0.293	<b>0.342</b>	0.102	0.149	<b>0.193</b>
BovW + TagFeature	0.200	0.199	0.201	0.090	0.096	0.098
BovW + RelExample	0.284	0.303	0.310	0.119	0.155	0.172
CNN + KNN	0.564	0.613	0.639	0.271	0.356	0.400
CNN + TagVote	0.561	0.613	0.638	0.257	0.358	<b>0.402</b>
CNN + TagProp	0.586	0.619	<b>0.641</b>	0.305	0.376	0.397
CNN + TagFeature	0.444	0.554	0.563	0.262	0.310	0.326
CNN + RelExample	0.538	0.603	0.584	0.300	0.346	0.373

In more detail, the following considerations hold. TagProp has higher MAP performance than KNN and TagVote in almost all the cases under analysis. As discussed in Section 3.4.5, TagProp is built upon KNN, but it weights the neighbor images by rank and applies a logistic model per tag. Since the logistic model does not affect the image ranking, the superior performance of TagProp should be ascribed to rank-based neighbor weighting. A per-tag comparison on MIRFlickr is given in Fig. 3.1. TagProp is almost always ahead of KNN and TagVote. Concerning TagVote and KNN, recall that their main difference is that TagVote applies the unique-user constraint on the neighborhood and it employs tag prior as a penalty term. The fact that the training data contains no batch-tagged images minimizes the influence of the unique-user constraint. While the penalty term does not affect image ranking for a given tag, it affects tag ranking for a given image. This explains why KNN and TagVote have mostly the same MAP. Also, the result suggests that the tag prior based penalty is helpful for doing tag assignment by neighbor voting.

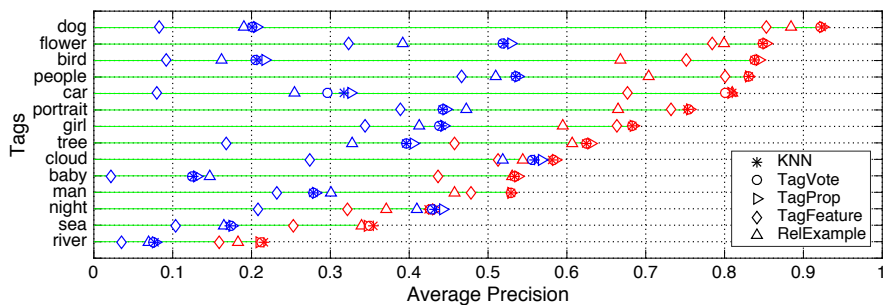


Figure 3.1: **Per-tag comparison of methods for tag assignment on MIRFlickr**, trained on Train1m. The colors identify the features used: **blue** for BovW, **red** for CNN. The test tags have been sorted in descending order by the performance of CNN + TagProp.

We observe that RelExample has a better MAP than TagFeature in every case. The absence of a filtering component makes TagFeature more likely to overfit to training examples irrelevant to the test tags. For the other two model-based methods, the overfit issue is alleviated by different strategies: RelExample employs a filtering component to select more relevant training examples, while TagProp has less parameters to tune.

A per-image comparison on NUS-WIDE is given in Fig. 3.2. The test images are put into disjoint groups so that images within the same group have the same number of ground truth tags. For each group, the area of the colored bars is proportional to the number of images on which the corresponding methods score best. The first group, i.e., images containing only one ground-truth tag, has the most noticeable change as the training set grows. There are 75,378 images in this group, and for 39% of the images, their single label is ‘person’. When Train1m is used, RelExample beats KNN, TagVote, and TagProp for this frequent label. This explains the leading position of RelExample in the first group. The result also confirms our earlier discussion in Section 3.3.1 that MiAP is likely to be biased by frequent tags.

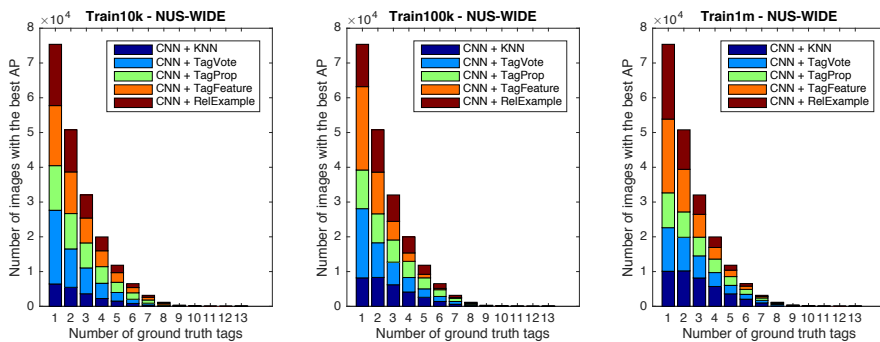


Figure 3.2: **Per-image comparison of methods for tag assignment on NUS-WIDE.** Test images are grouped in terms of their number of ground truth tags. The area of a colored bar is proportional to the number of images that the corresponding method scores best.

In summary, as long as enough training examples are provided, instance-based methods are on par with model-based methods for tag assignment. Model-based methods are more suited when the training data is of limited availability. However, they are less resilient to noise, and consequently a proper filtering strategy for refining the training data becomes essential.

### 3.5.2 Tag refinement

Table 3.5 shows the performance of different methods for tag refinement. We were unable to complete the table. In particular, RobustPCA could not go over 350k images due to its high demand in both CPU time and memory

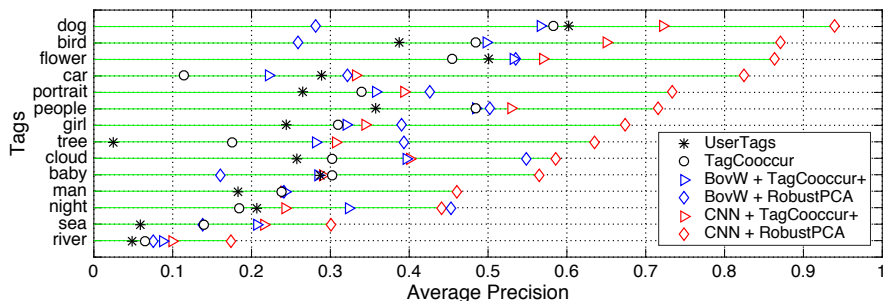


Figure 3.3: **Per-tag comparison of methods for tag refinement on MIRFlickr**, trained on Train100k. The colors identify the features used: **blue** for BovW, **red** for CNN. The test tags have been sorted in descending order by the performance of CNN + RobustPCA.

(see Table 3.3), while TensorAnalysis was provided by the authors only on MIRFlickr with Train10k, Train100k, and the BovW feature.

RobustPCA outperforms the competitors on both test sets, when provided with the CNN feature. Fig. 3.3 presents a per-tag comparison on MIRFlickr. RobustPCA has the best scores for 9 out of the 14 tags with BovW, and wins all the tags when CNN is used.

Concerning the influence of the media dimension, the tag + image based methods (RobustPCA and TagCooccur+) are in general better than the tag based method (TagCooccur). As shown in Fig. 3.3, except for 3 out of 14 MIRFlickr test tags with BovW, using the image media is beneficial. As in the tag assignment task, the use of the CNN feature strongly improves the performance.

Concerning the learning methods, TensorAnalysis has the potential to leverage tag, image, and user simultaneously. However, due to its relatively poor scalability, we were able to run this method only with Train10k and Train100k on MIRFlickr. For Train10k, TensorAnalysis yielded higher MiAP than RobustPCA, probably thanks to its capability of modeling user correlations. It is outperformed by RobustPCA when more training data is used.

As more training data is used, the performance of TagCooccur, TagCooccur+, and RobustPCA on MIRFlickr consistently improves. Since these three methods rely on data-driven tag affinity, image affinity, or tag and image affinity, a small set of 10k images is generally inadequate to compute

Table 3.5: Evaluating methods for tag refinement. The asterisk (\*) indicates results provided by the authors of the corresponding methods, while the dash (-) means we were unable to produce results. Given the same feature, bold values indicate top performers on individual test sets per performance metric.

Method	MIRFlickr			NUS-WIDE		
	Train10k	Train100k	Train1m	Train10k	Train100k	Train1m
<i>MiAP scores:</i>						
UserTags	0.204	0.204	0.204	0.255	0.255	0.255
TagCooccur	0.213	0.242	0.253	0.269	0.305	0.317
BovW + TagCooccur+	0.217	0.262	0.286	0.245	0.297	0.324
BovW + RobustPCA	0.271	<b>0.310</b>	-	<b>0.332</b>	0.323	-
BovW + TensorAnalysis	*0.298	*0.297	-	-	-	-
CNN + TagCooccur+	0.234	0.277	0.310	0.305	0.359	0.387
CNN + RobustPCA	0.368	<b>0.376</b>	-	<b>0.424</b>	0.419	-
CNN + TensorAnalysis	-	-	-	-	-	-
<i>MAP scores:</i>						
UserTags	0.263	0.263	0.263	0.338	0.338	0.338
TagCooccur	0.266	0.298	0.313	0.223	0.321	0.308
BovW + TagCooccur+	0.294	0.343	<b>0.377</b>	0.231	0.345	<b>0.353</b>
BovW + RobustPCA	0.225	0.337	-	0.229	0.234	-
BovW + TensorAnalysis	-	-	-	-	-	-
CNN + TagCooccur+	0.330	0.381	0.420	0.264	0.391	0.406
CNN + RobustPCA	0.566	<b>0.627</b>	-	0.439	<b>0.440</b>	-
CNN + TensorAnalysis	-	-	-	-	-	-

these affinities. The effect of increasing the training set size is clearly visible if we compare scores corresponding to Train10k and Train100k. The results on NUS-WIDE show some inconsistency. For TagCooccur, MiAP improves from Train100k to Train1m, while MAP drops. This is presumably due to the fact that in the experiments we used the parameters recommended in the original paper, appropriately selected to optimize tag ranking. Hence, they might be suboptimal for image ranking. BovW + RobustPCA scores a lower MAP than BovW + TagCooccur+. This is probably due to the fact that the low-rank matrix factorization technique, while being able to jointly exploit tag and image information, is more sensitive to the content-based representation.

A per-image comparison is given in Fig. 3.4. As for tag assignment, the test images have been grouped according to the number of ground truth tags associated. The size of the colored areas is proportional to the number of images where the corresponding method scores best. For the majority of test image, the three tag refinement methods have higher average precision than UserTags. This means more relevant tags are added, so the tags are refined. It should be noted that the success of tag refinement depends much on the quality of the original tags assigned to the test images. Examples are shown in Table 3.7: in row 6, although the tag ‘earthquake’ is irrelevant to the image content, it is ranked at the top by RobustPCA. To what extent a tag refinement method shall count on the existing tags is tricky.

To summarize, the tag + image based methods outperform the tag based method for tag refinement. RobustPCA is the best, and improves as more training data is employed. Nonetheless, implementing RobustPCA is challenging for both computation and memory footprint. In contrast, TagCooccur+ is more scalable and it can learn from large-scale data.

### 3.5.3 Tag retrieval

Tables 3.8 and 3.9 show the performance of different methods for tag retrieval. Recall that when retrieving images for a specific test tag, we consider only images that are labeled with this tag. Hence, MAP scores here are higher than their counterpart in Table 3.5.

We start our analysis by comparing the three baselines, namely UserTags, TagNum, and TagPosition, which retrieve images simply by the original tags. As it can be noticed, TagNum and TagPosition are more effective than UserTags, TagNum outperforms TagPosition on Flickr51, and the latter



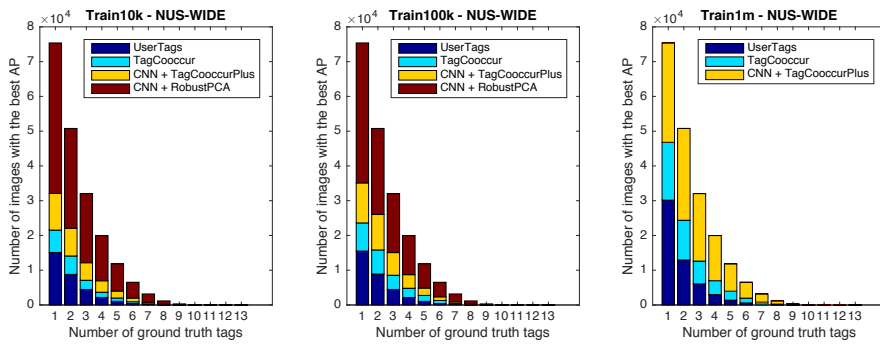


Figure 3.4: **Per-image comparison of methods for tag refinement on NUS-WIDE.** Test images are grouped in terms of their number of ground truth tags. The area of a colored bar is proportional to the number of images that the corresponding method scores best.

Table 3.6: Selected tag assignment results on NUS-WIDE. Visual feature: BovW. The top five ranked tags are shown, with correct prediction marked by the *bold italic* font.



Test image	Ground truth	User tags	Tag assignment			
			KNN	TagVote	TagProp	RelExample
	sign	<i>sign</i> reptile zoo red white	animal flower car horse street	dog house bird <i>sign</i> bear	<i>sign</i> street flower dog bird	soccer whale book toy moon
	animal dog person	colour color <i>dog</i> hound	flower garden horse tree <i>dog</i>	garden flower food cat <i>dog</i>	flower <i>dog</i> garden car tree	garden <i>dog</i> fish fox <i>animal</i>
	cloud grass sky	<i>cloud</i> <i>grass</i>	<i>cloud</i> <i>sky</i> beach water snow	<i>cloud</i> <i>sky</i> water beach mountain	<i>cloud</i> <i>sky</i> beach water lake	<i>cloud</i> ocean surf <i>sky</i> beach
	animal bear water	brown <i>bear</i> salmon national park	snow beach <i>animal</i> <i>water</i> tree	snow <i>animal</i> waterfall tree <i>water</i>	snow beach sand <i>bear</i> <i>water</i>	<i>water</i> sand rock surf ocean
	airplane cloud military sky	flag great	<i>sky</i> <i>cloud</i> snow bird <i>airplane</i>	snow <i>cloud</i> <i>sky</i> mountain bird	<i>airplane</i> <i>sky</i> snow bird airport	snow frost bird <i>airplane</i> tattoo
	cloud garden sky water	china earthquake people hangzhou summer westlake	car beach <i>water</i> street tree	grass tree <i>water</i> <i>water</i> bridge	car road street <i>sky</i> bird	house road grass bird sand
	police road vehicle window	farmer dog motorcycle <i>police</i> train	car street <i>police</i> <i>vehicle</i> <i>road</i>	car street <i>police</i> <i>vehicle</i> sport	<i>police</i> car street <i>road</i> sport	<i>police</i> <i>vehicle</i> street car sport

Table 3.7: Selected tag refinement results on NUS-WIDE. Visual feature: BovW. The top five ranked tags are shown, with correct prediction marked by the *bold italic* font.

Test image	Ground truth	User tags	Tag refinement		
			TagCooccur	TagCooccur+	RobustPCA
	sign	<i>sign</i> reptile zoo red white	animal street <i>sign</i> water car	<i>sign</i> bird dog animal toy	<i>sign</i> bird flower animal street
	animal dog person	colour color <i>dog</i> hound	<i>dog</i> <i>animal</i> car beach flower	<i>dog</i> flower <i>animal</i> cat food	<i>dog</i> flower <i>animal</i> water garden
	cloud grass sky	<i>cloud</i> <i>grass</i>	<i>grass</i> <i>sky</i> tree flower water	<i>cloud</i> <i>sky</i> water beach tree	<i>cloud</i> <i>grass</i> <i>sky</i> water mountain
	animal bear water	brown <i>bear</i> salmon national park	waterfall <i>water</i> tree <i>bear</i> <i>animal</i>	waterfall <i>water</i> <i>animal</i> snow tree	<i>water</i> waterfall <i>bear</i> <i>animal</i> snow
	airplane cloud military sky	flag great	car street snow water <i>sky</i>	snow <i>sky</i> <i>cloud</i> mountain bird	flag <i>sky</i> snow <i>cloud</i> bird
	cloud garden sky water	china earthquake people hangzhou summer westlake	<i>water</i> flower street temple tree	tree <i>water</i> street <i>garden</i> car	earthquake <i>water</i> tree <i>cloud</i> <i>sky</i>
	police road vehicle window	farmer dog motorcycle <i>police</i> train	street car animal train bird	car street <i>police</i> food horse	<i>police</i> train dog bird car

has better scores on NUS-WIDE. The effectiveness of such metadata based features depend much on datasets, and are unreliable for tag retrieval.

All the methods considered have higher MAP than the three baselines. All the methods have better performance than the baselines on Flickr51 and performance increases with the size of the training set. On NUS-WIDE, SemanticField, TagCooccur, and TagRanking, are less effective than TagPosition. We attribute this result to the fact that, for these methods, the tag relevance functions favor images with fewer tags. So they closely follow similar performance and dataset dependency.

Concerning the influence of the media dimension, the tag + image based methods (KNN, TagVote, TagProp, TagCooccur+, TagFeature, RobustPCA, RelExample) are in general better than the tag based method (SemanticField and TagCooccur). Fig. 3.5 shows the per-tag retrieval performance on Flickr51. For 33 out of the 51 test tags, RelExample exhibits average precision higher than 0.9. By examining the top retrieved images, we observe that the results produced by tag + image based methods and tag based methods are complementary to some extent. For example, consider ‘military’, one of the test tags of NUS-WIDE. RelExample retrieves images with strong visual patterns such as military vehicles, while SemanticField returns images of military personnel. Since the visual content is ignored, the results of SemanticField tend to be visually different, so making it possible to handle tags with visual ambiguity. This fact can be observed in Fig. 3.6, which shows the top 10 ranked images of ‘jaguar’ by TagPosition, SemanticField, BovW + RelExample, and CNN + RelExample. Although their results are all correct, RelExample finds jaguar-brand cars only, while SemanticField covers both cars and animals. However, for a complete evaluation of the capability of managing ambiguous tags, fine-grained ground truth beyond what we currently have is required.

Concerning the learning methods, TagVote consistently performs well as in the tag assignment experiment. KNN is comparable to TagVote, due to the reason we have discussed in Section 3.5.1. Given the CNN feature, the two methods even outperform their model-based variant TagProp. Similar to the tag refinement experiment, the effectiveness of RobustPCA for tag retrieval is sensitive to the choice of visual features. While BovW + RobustPCA is worse than the majority on Flickr51, the performance of CNN + RobustPCA is more stable, and performs well. For TagFeature, its gain from using larger training data is relatively limited due to the absence of denoising. In contrast,

Table 3.8: Evaluating methods for tag retrieval, MAP scores. Given the same feature, bold values indicate top performers on individual test sets per performance metric.

Method	Flickr51			NUS-WIDE		
	Train10k	Train100k	Train1m	Train10k	Train100k	Train1m
<i>MAP scores:</i>						
UserTags	0.595	0.595	0.595	0.489	0.489	0.489
TagNum	0.664	0.664	0.664	0.520	0.520	0.520
TagPosition	0.640	0.640	0.640	0.557	0.557	0.557
SemanticField	0.687	0.707	0.713	0.565	0.584	0.584
TagCooccur	0.625	0.679	0.704	0.534	0.576	0.588
BovW + TagCooccur+	0.640	0.732	0.764	0.560	0.622	0.643
BovW + TagRanking	0.685	0.686	0.708	0.557	0.574	0.578
BovW + KNN	0.678	0.742	0.770	0.587	0.632	0.658
BovW + TagVote	0.678	0.741	0.769	0.587	0.632	0.659
BovW + TagProp	0.671	0.748	0.772	0.585	0.636	0.657
BovW + TagFeature	0.689	0.726	0.737	0.589	0.602	0.606
BovW + RelExample	0.706	0.756	<b>0.783</b>	0.609	0.645	<b>0.663</b>
BovW + RobustPCA	0.697	0.701	–	0.650	0.650	–
BovW + TensorAnalysis	–	–	–	–	–	–
CNN + TagCooccur+	0.654	0.781	0.821	0.572	0.653	0.674
CNN + TagRanking	0.744	0.735	0.747	0.589	0.590	0.590
CNN + KNN	0.811	0.859	0.880	0.683	0.722	0.734
CNN + TagVote	0.808	0.859	<b>0.881</b>	0.675	0.724	<b>0.738</b>
CNN + TagProp	0.824	0.867	0.879	0.689	0.727	0.731
CNN + TagFeature	0.827	0.853	0.859	0.675	0.700	0.703
CNN + RelExample	0.838	0.863	0.878	0.689	0.717	0.734
CNN + RobustPCA	0.811	0.839	–	0.725	0.726	–
CNN + TensorAnalysis	–	–	–	–	–	–

Table 3.9: Evaluating methods for tag retrieval,  $NDCG_{20}$  scores. Given the same feature, bold values indicate top performers on individual test sets per performance metric.

Method	Flickr51			NUS-WIDE		
	Train10k	Train100k	Train1m	Train10k	Train100k	Train1m
<i>NDCG<sub>20</sub> scores:</i>						
UserTags	0.432	0.432	0.432	0.487	0.487	0.487
TagNum	0.522	0.522	0.522	0.541	0.541	0.541
TagPosition	0.511	0.511	0.511	0.623	0.623	0.623
SemanticField	0.591	0.623	0.645	0.596	0.622	0.624
TagCooccur	0.482	0.527	0.631	0.529	0.602	0.614
BovW + TagCooccur+	0.503	0.625	0.686	0.590	0.681	0.734
BovW + TagRanking	0.530	0.568	0.571	0.557	0.572	0.572
BovW + KNN	0.577	0.699	0.756	0.638	0.734	0.799
BovW + TagVote	0.573	0.701	0.754	0.629	0.734	0.804
BovW + TagProp	0.570	0.715	<b>0.759</b>	0.666	0.750	<b>0.809</b>
BovW + TagFeature	0.547	0.626	0.646	0.622	0.615	0.618
BovW + RelExample	0.614	0.722	0.748	0.692	0.736	0.776
BovW + RobustPCA	0.549	0.548	–	0.768	0.781	–
BovW + TensorAnalysis	–	–	–	–	–	–
CNN + TagCooccur+	0.504	0.615	0.724	0.571	0.705	0.738
CNN + TagRanking	0.577	0.607	0.597	0.578	0.594	0.583
CNN + KNN	0.709	0.830	0.897	0.773	0.832	0.863
CNN + TagVote	0.722	0.826	<b>0.899</b>	0.740	0.837	<b>0.879</b>
CNN + TagProp	0.768	0.857	0.865	0.764	0.839	0.845
CNN + TagFeature	0.755	0.813	0.818	0.704	0.807	0.787
CNN + RelExample	0.764	0.843	0.879	0.773	0.814	0.866
CNN + RobustPCA	0.733	0.821	–	0.865	0.862	–
CNN + TensorAnalysis	–	–	–	–	–	–

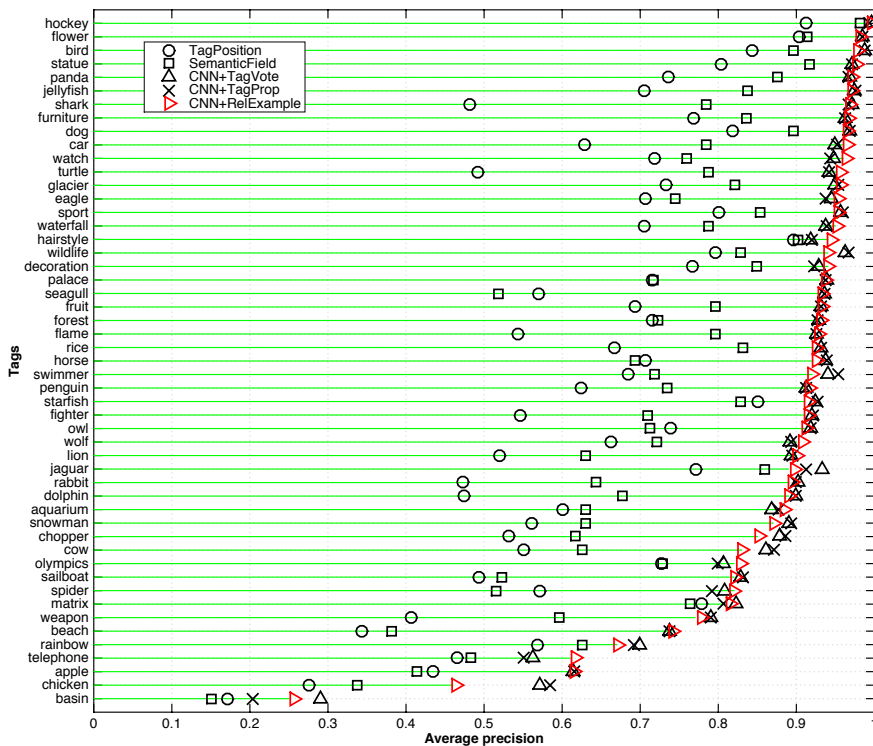


Figure 3.5: **Per-tag comparison between TagPosition, SemanticField, TagVote, TagProp, and RelExample on Flickr51, with Train1m as the training set. The 51 test tags have been sorted in descending order by the performance of RelExample.**

RelExample, by jointly using SemanticField and TagVote in its denoising component, is consistently better than TagFeature.

The performance of individual methods consistently improves as more training data is used. As the size of the training set increases, the performance gap between the best model-based method (RelExample) and the best instance-based method (TagVote) reduces. This suggests that large-scale training data diminishes the advantage of model-based methods against the relatively simple instance-based methods.

In summary, even though the performance of the methods evaluated varies over datasets, common patterns have been observed. First, the more

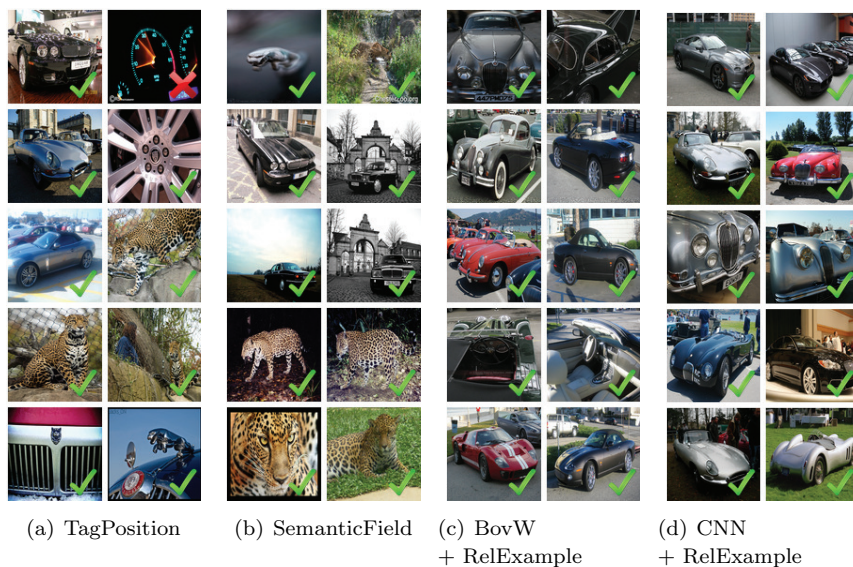


Figure 3.6: **Top 10 ranked images of ‘jaguar’, by (a) TagPosition, (b) SemanticField, (c) BovW + RelExample, and (d) CNN + RelExample.** Checkmarks (✓) indicate relevant results. While both RelExample and SemanticField outperform the TagPosition baseline, the results of SemanticField show more diversity for this ambiguous tag. The difference between (c) and (d) suggests that the results of RelExample can be diversified by varying the visual feature in use.

social data for training are used the better performance is obtained. Since the tag relevance functions are learned purely from social data without any extra manual labeling, and social data are increasingly growing, this result promises that better tag relevance functions can be learned. Second, given small-scale training data, tag + image based methods that conducts model-based learning with denoised training examples turn out to be the most effective solution, This however comes with a price of reducing the visual diversity in the retrieval results. Moreover, the advantage of model-based learning vanishes as more training data and the CNN feature are used, and TagVote performs the best.



### 3.5.4 Flickr versus ImageNet

To address the question of whether one shall resort to an existing resource such as ImageNet for tag relevance learning, this section presents an empirical comparison between our Flickr based training data and ImageNet. A number of methods do not work with ImageNet or require modifications. For instance, tag + image + user information based methods must be able to remove their dependency on user information, as such information is unavailable in ImageNet. Tag co-occurrences are also strongly limited, because an ImageNet example is annotated with a single label. Because of these limitations, we evaluate only the two best performing methods, TagVote and TagProp. TagProp can be directly used since it comes from classic image annotation, while TagVote is slightly modified by removing the unique user constraint. The CNN feature is used for its superior performance against the BovW feature.

To construct a customized subset of ImageNet that fits the three test sets, we take ImageNet examples whose labels precisely match with the test tags. Notice that some test tags, e.g., ‘portrait’ and ‘night’, have no match, while some other tags, e.g., ‘car’ and ‘dog’, have more than one matches. In particular, MIRFlickr has 2 missing tags, while the number of missing tags on Flickr51 and NUS-WIDE is 9 and 15. For a fair comparison these missing tags are excluded from the evaluation. Putting the remaining test tags together, we obtain a subset of ImageNet, containing 166 labels and over 200k images, termed ImageNet200k. For a fair comparison, we considered only Train100k and Train1m training sets of socially tagged images.

The left half of Table 3.10 shows the performance of tag assignment. TagVote/TagProp trained on the ImageNet data are less effective than their counterparts trained on the Flickr data. For a better understanding of the result, we employ the same visualization technique as used in Section 3.5.1, i.e., grouping the test images in terms of the number of their ground truth tags, and subsequently checking the performance per group. As shown in Fig. 3.7, while ImageNet200k performs better on the first group, i.e., images with a single relevant tag, it is outperformed by Train100k and Train1M on the other groups. For its single-label nature, ImageNet is less effective for assigning multiple labels to an image.

For tag retrieval, as shown in the right half of Table 3.10, TagVote/TagProp learned from ImageNet200k in general have higher MAP and NDCG scores than their counterparts learned from the Flickr data. By compar-

Table 3.10: Flickr versus ImageNet. Notice that the numbers on Train100k and Train1M are different from Tables 3.4, 3.8 and 3.9 due to the use of a reduced set of test tags. Bold values indicate top performers on a specific test set per performance metric.

<b>Tag Assignment</b>				
<b>Training Set</b>	<b>MIRFlickr</b>		<b>NUS-WIDE</b>	
	TagVote	TagProp	TagVote	TagProp
<i>MiAP scores:</i>				
Train100k	0.377	0.383	0.392	0.389
Train1M	0.389	<b>0.392</b>	<b>0.414</b>	0.393
ImageNet200k	0.345	0.304	0.325	0.368
<i>MAP scores:</i>				
Train100k	0.641	0.647	0.386	0.405
Train1M	0.664	<b>0.668</b>	<b>0.429</b>	0.420
ImageNet200k	0.532	0.532	0.363	0.362
<b>Tag Retrieval</b>				
<b>Training Set</b>	<b>Flickr51</b>		<b>NUS-WIDE</b>	
	TagVote	TagProp	TagVote	TagProp
<i>MAP scores:</i>				
Train100k	0.854	0.860	0.742	0.745
Train1M	<b>0.874</b>	0.871	0.753	0.745
ImageNet200k	0.873	0.873	<b>0.762</b>	<b>0.762</b>
<i>NDCG<sub>20</sub> scores:</i>				
Train100k	0.838	0.863	0.849	0.856
Train1M	0.894	0.851	<b>0.891</b>	0.853
ImageNet200k	<b>0.920</b>	0.898	0.843	0.847

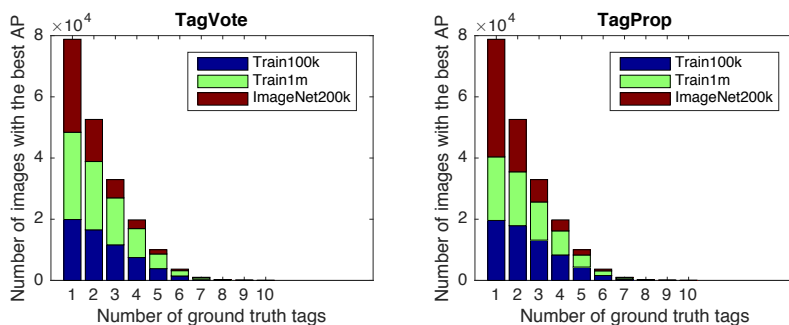


Figure 3.7: **Per-image comparison of TagVote/TagProp learned from different training datasets**, tested on NUS-WIDE. Test images are grouped in terms of the number of ground truth tags. Within each group, the area of a colored bar is proportional to the number of images that (the method derived from) the corresponding training dataset scores the best. ImageNet200k is less effective for assigning multiple labels to an image.

ing the performance difference per concept, we find that the gain is largely contributed by a relatively small amount of concepts. Consider for instance TagVote + ImageNet200k and TagVote + Train1M on NUS-WIDE. The former outperforms the latter for 25 out of the 66 tested concepts. By sorting the concepts according to their absolute performance gain, the top three winning concepts of TagVote + ImageNet200k are ‘sand’, ‘garden’, and ‘rainbow’, with AP gain of 0.391, 0.284, and 0.176, respectively. Here, the lower performance of TagVote + Train1M is largely due to the subjectiveness of social tagging. For instance, Flickr images labeled with ‘sand’ tend to be much more diverse, showing a wide range of things visually irrelevant to sand. Interestingly, the top three losing concepts of TagVote + ImageNet200k are ‘running’, ‘valley’, and ‘building’, with AP loss of 0.150, 0.107, and 0.090, respectively. For these concepts, we observe that their ImageNet examples lack diversity. E.g., ‘running’ in ImageNet200k mostly shows a person running on a track. In contrast, the subjectiveness of social tagging now has a positive effect on generating diverse training examples.

In summary, for tag assignment social media examples are a preferred resource of training data. For tag retrieval ImageNet yields better performance, yet the performance gain is largely due to a few tags where social tagging is very noisy. In such a case, controlled manual labeling seems indis-

pensable. In contrast, with clever tag relevance learning algorithms, social training data demonstrate competitive or even better performance for many of the tested tags. Nevertheless, where the boundary between the two cases is precisely located remains unexplored.

### 3.6 Conclusions

Having established the common ground between methods, a new experimental protocol was introduced for a head-to-head comparison between the state-of-the-art. A selected set of eleven representative works were implemented and evaluated for tag assignment, refinement, and/or retrieval. The evaluation justifies the state-of-the-art on the three tasks. For tag assignment, TagProp and TagVote perform best. For tag refinement, RobustPCA is the choice. For tag retrieval, TagVote achieves the best overall performance. Concerning what media is essential for tag relevance learning, tag + image is consistently found to be better than tag alone. While the joint use of tag, image, and user information (via TensorAnalysis) demonstrates its potential on small-scale datasets, it becomes computationally prohibitive as the dataset size increases to 100k and beyond. Comparing the three learning strategies, instance-based and model-based methods are found to be more reliable and scalable than their transduction-based counterparts. As model-based methods are more sensitive to the quality of social image tagging, a proper filtering strategy for refining the training media is crucial for their success. Despite their leading performance on the small training dataset, we find that the performance gain over the instance-based alternatives diminishes as more training data is used. Finally, the CNN feature used as a substitute for the BovW feature brings considerable improvements for all the tasks.

## Chapter 4

# A Cross Modal Approach for Tag Assignment

*Tag assignment is still an important open problem in multimedia and computer vision. Many approaches previously proposed in the literature do not accurately capture the intricate dependencies between image content and annotations. We propose a learning procedure based on Kernel Canonical Correlation Analysis which finds a mapping between visual and textual words by projecting them into a latent meaning space. The learned mapping is then used to annotate new images using advanced nearest neighbor methods. We evaluate our approach on three popular datasets, and show clear improvements over several approaches relying on more standard representations.*<sup>1</sup>

### 4.1 Introduction

The exponential growth of media sharing websites, such as Flickr or Picasa, and social networks such as Facebook, has led to the availability of large collections of images tagged with human-provided labels. These tags reflect the image content and can thus be exploited as a loose form of labels and context. Several researchers have explored ways to use images with

---

<sup>1</sup>A preliminary version of the work presented in this chapter has been published as “A Cross-modal Approach for Automatic Image Annotation” in *Proc. of International Conference of Multimedia Retrieval (ICMR)*, Glasgow, 2014.

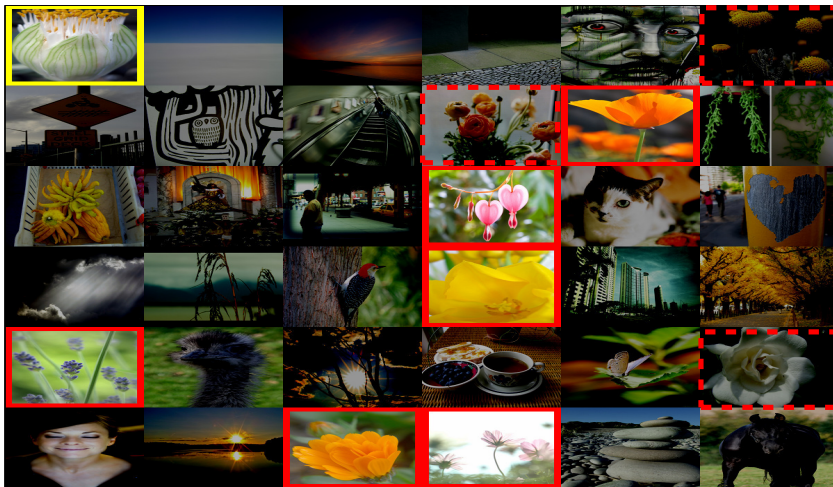
associated labels as a source to build classifiers or to transfer their tags to similar images [45, 68, 105, 113, 133, 233]. Image annotation is therefore a very active subject of research [23, 126, 141, 195, 221, 223] since we can clearly increase performance of search and indexing over image collections that are machine enriched with a set of meaningful labels. In this chapter we tackle the problem of assigning a finite number of relevant tags to an image, given the image appearance and some prior knowledge on the joint distribution of visual features and tags based on some weakly and noisy annotated data.

The main shortcomings of previous works in the field are twofold. The first is the aforementioned *semantic gap* problem, which points to the fact that it is hard to extract semantically meaningful entities using just low level visual features. The second shortcoming arises from the fact that many parametric models, previously presented in the literature, are not rich enough to accurately capture the intricate dependencies between image content and annotations. Recently, nearest neighbor based methods have attracted much attention since they have been found to be quite successful for tag prediction [68, 113, 133, 192, 233] (see also Chapter 2 and 3). This is mainly due to their flexibility and capacity to adapt to the patterns in the data as more training data is available. The base ingredient for a vote based tagging algorithm is of course the source of votes: the set of  $K$  nearest neighbors. In challenging real world data it is often the case that the vote casting neighbors do not contain enough statistics to obtain reliable predictions. This is mainly due to the fact that certain tags are much more frequent than others and can cancel out less frequent but relevant tags [68, 113]. It is obvious that all voting schemes can benefit from a better set of neighbors. We believe that the main bottleneck in obtaining such ideal neighbors set is the semantic gap. We address this problem using a cross-modal approach to learn a representation that maximizes the correlation between visual features and tags in a common semantic subspace.

In Figure 4.1 we show our intuition with an example provided by real data. We compare for the same query, a flower close-up, the first thirty-five most similar examples provided by the visual features and by our representation. The first thing to notice is the large visual and semantic difference between the sets of retrieved neighbors by the two approaches. Note also that some flower pictures, which we highlight with a dashed red rectangle, were not tagged as such. Second, note how the result presented in Figure 4.1(b) have more and better ranked *flower* images than the one in Figure



(a) Baseline



(b) Our Method

Figure 4.1: Nearest neighbors found with baseline representation (a) and with our proposed method (b) for a flower image (first highlighted in yellow in both figures) from the MIRFlickr-25K dataset. Training images with ground truth tag *flower* are highlighted with a red border. Nearest neighbors are sorted by decreasing similarity and arranged in a matrix using a row-major convention. Dashed red lines indicate flower pictures not tagged as such.

4.1(a). Indeed with the result set in Figure 4.1(a) it is not possible to obtain a sufficient amount of meaningful neighbors and the correct tag *flower* is canceled by others such as *dog* or *people*.

In this chapter we present a cross-media approach that relies on Kernel Canonical Correlation Analysis (KCCA) [71,72] to connect visual and textual modalities through a common latent meaning space (called *semantic space*). Visual features and labels are mapped to this space using feature similarities that are observable inside the respective domains. If mappings are close in this semantic space, the images are likely to be instances of the same underlying semantic concept. The learned mapping is then used to annotate new images using a nearest-neighbor voting approach. We present several experiments using different voting schemes. First, the simple KNN voting of Makadia *et al.* [133], and second three advanced NN models such as TagVote [113], TagProp [68] and 2PKNN [195].

### 4.1.1 Contribution

Other existing approaches learn from both words and images, including previous uses of CCA [63, 71, 76, 159]. In contrast, we are the first to propose an approach that combines an effective cross-modal representation with advanced nearest-neighbor models for the specific task of tag assignment.

In the following we show that, if combined with advanced NN schemes able to deal with the class-imbalance (i.e. large variations in the frequency of different labels), our cross-media model achieves high performance without requiring heavy computation such as in the case of metric learning frameworks with many parameters (as in [68, 195]).

We present experimental results for two standard datasets, Corel5K [45] and IAPR-TC12 [67], obtaining highly competitive results. We report also experiments on a challenging dataset collected from Flickr, i.e. the MIRFlickr-25K dataset [74], and our results show that the performance of the proposed method is boosted even further in a realistic and more interesting scenario such as the one provided by weakly-labeled images.

## 4.2 Related Work

In the multimedia and computer vision communities, jointly modeling images and text has been an active research area in the recent years. A first group



of methods uses mixture models to define a joint distribution over image features and labels. The training images are used by these models as components to define a mixture model over visual features and tags [23, 49, 101]. They can be interpreted as non-parametric density estimators over the co-occurrence of images and labels. In another group of methods based on topic models (such as LDA and pLSA), each topic represents a distribution over image features and labels [11, 147]. These kind of generative models may be criticized because they maximize the generative data likelihood, which is not optimal for predictive performance. Another main criticism of these models is their need for simplifying assumptions in order to do tractable learning and inference.

Discriminative models such as support vector machines have also been proposed [65, 196]. These methods learn a classifier for each label, and use them to predict whether a test image belongs to the class of images that are annotated with a particular label. A main criticism of these works resides in the necessity to define in advance the number of labels and to train individual classifiers for each of them. This is not feasible in a realistic scenario like the one of web images. Despite their simplicity, nearest-neighbor based methods for image annotation have been found to give state-of-the-art results [68, 133, 195]. The intuition is that similar images share common labels. The common procedure of the existing nearest-neighbor methods is to search for a set of visually similar images and then to select a set of relevant associated tags based on a tag transfer procedure [68, 113, 133]. In all these previous approaches, this similarity is determined only using image visual features.

## 4.3 Approach

The proposed method is based on KCCA which provides a common representation for the visual and tag features. We refer to this common representation as *semantic space*. Similarly to [71, 76] we use KCCA to connect visual and textual modalities, but our method is designed to effectively tackle the particular problem of image auto-annotation. In Section 4.3.1 we present our visual and text features with their respective kernels; next we briefly describe KCCA (Section 4.3.2) and the different NN schemes (Section 4.3.3). In Figure 4.2 we show an embedding computed with ISOMAP [184] of the visual data and its semantic projection. We randomly pick three tags to show how the semantic projection that we learn with KCCA better suits the actual

distribution of tags with respect to the visual representation. The semantic projection improves the separation of the classes, allowing a better manifold reconstruction and, as our experiments will confirm, an improvement on precision and recall on different datasets.

### 4.3.1 Visual and Tags Views

#### Visual Feature Representation and Kernels

We directly use the 15 features provided by the authors of [68, 194]<sup>2</sup>. These are different types of global and local features commonly used for image retrieval and categorization. In particular we use two types of global descriptors: Gist and color histograms with 16 bins in each channel for RGB, LAB, HSV color spaces. Local features include SIFT and robust hue descriptors, both extracted densely on a multi-scale grid or for Harris-Laplacian interest points. The local feature descriptors are quantized using k-means and then all the images are represented as bag-of-(visual)words histograms. The histograms are also computed in a spatial arrangement over three horizontal regions of the image, and then concatenated to form a new global descriptor that encodes some information of the global spatial layout.

In this work we use  $\chi^2$  exponential kernels for all visual features  $f \in \mathcal{F}$ :

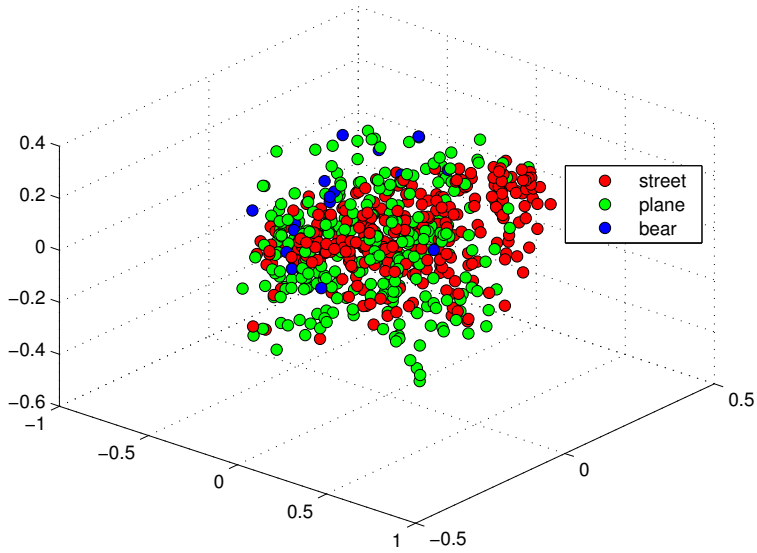
$$K_{\chi^2}(h_i, h_j) = \exp\left(-\frac{1}{2A} \sum_{k=1}^d \frac{(h_i(k) - h_j(k))^2}{(h_i(k) + h_j(k))}\right), \quad (4.1)$$

where  $A$  is the mean of the  $\chi^2$  distances among all the training examples,  $d$  is the dimensionality of a particular feature descriptor and  $h_i$  is its respective histogram representation. It has to be noticed that all the feature descriptors are L1-normalized. Finally, all the different visual kernels are averaged to obtain the final visual representation. We obtain the kernel between two images  $I_i, I_j$  via kernel averaging:

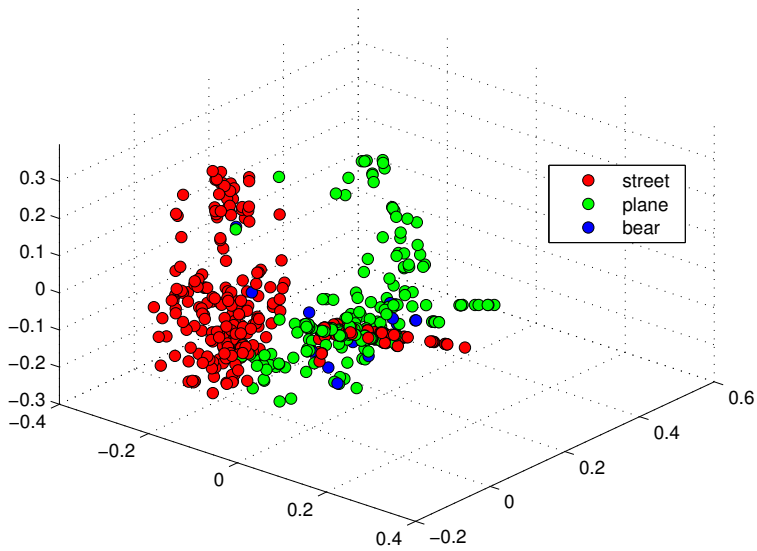
$$K_v(I_i, I_j) = \frac{1}{|\mathcal{F}|} \sum_{f \in \mathcal{F}} K_{\chi^2}(h_i^f, h_j^f). \quad (4.2)$$

---

<sup>2</sup>These features are available at: <http://lear.inrialpes.fr/people/guillaumin/data.php>.



(a) Visual Space



(b) Semantic Space

Figure 4.2: Visualization of three labels (Corel5K): (a) distribution of image features in the visual space (b) distribution of the same images after projecting into the semantic space learned using KCCA. Note the clearer distinction of the clusters in the semantic space.

### Tag Feature Representation and Kernel

We use as tag features the traditional bag-of-words which records which labels are named in the image, and how many times. Supposing  $V$  is our vocabulary size, i.e. the total possible words used for annotation, each tag-list is mapped to an  $V$ -dimensional feature vector  $h = [w_1, \dots, w_V]$ , where  $w_i$  counts the number of times the  $i$ -th word is mentioned in the tag list. In our case this representation is highly sparse and often counts are simply 0 or 1 values. We use these features to compute a linear kernel that corresponds to counting the number of tags in common between two images:

$$K_t(h_i, h_j) = \langle h_i, h_j \rangle = \sum_k^V h_i(k)h_j(k). \quad (4.3)$$

### 4.3.2 Kernel Canonical Correlation Analysis

Given two views of the data, such as the ones provided by visual and textual modalities, we can construct a common representation. Canonical Correlation Analysis (CCA) seeks to utilize data consisting of paired views to simultaneously find projections from each feature space such that the correlation between the projected representations is maximized. In the literature, the CCA method has often been used in cross-language information retrieval, where one queries a document in a particular language to retrieve relevant documents in another language. In our case, the algorithm learns two semantic projection bases, one per each modality (i.e. the  $v$  view is the visual cue while the  $t$  view is the tag-list cue).

More formally, given  $N$  samples from a paired dataset  $\{(v_1, t_1), \dots, (v_N, t_N)\}$ , where  $v_i \in \mathbb{R}^n$  and  $t_i \in \mathbb{R}^m$  are the two views of the data, the goal is to simultaneously find directions  $w_v^*$  and  $w_t^*$  that maximize the correlation of the projections of  $v$  onto  $w_v$  and  $t$  onto  $w_t$ . This is expressed as:

$$w_v^*, w_t^* = \arg \max_{w_v, w_t} \frac{\hat{E}[\langle v, w_v \rangle \langle t, w_t \rangle]}{\sqrt{\hat{E}[\langle v, w_v \rangle^2] \hat{E}[\langle t, w_t \rangle^2]}} = \arg \max_{w_v, w_t} \frac{w_v^T C_{vt} w_t}{\sqrt{w_v^T C_{vv} w_v w_t^T C_{tt} w_t}}, \quad (4.4)$$

where  $\hat{E}$  denotes the empirical expectation,  $C_{vv}$  and  $C_{tt}$  respectively denote the auto-covariance matrices for  $v$  and  $t$  data, and  $C_{vt}$  denotes the

between-sets covariance matrix. The solution can be found via a generalized eigenvalue problem [72].

The common CCA algorithm can only recover linear relationships, it is therefore useful to kernelize it by projecting the data into a higher-dimensional feature space by using the kernel trick. Kernel Canonical Correlation Analysis (KCCA) is the kernelized version of CCA. To this end, we define kernel functions over  $v$  and  $t$  as  $K_v(v_i, v_j) = \phi_v(v_i)^T \phi_v(v_j)$  and  $K_t(t_i, t_j) = \phi_t(t_i)^T \phi_t(t_j)$ . Here, the idea is to search for solutions of  $w_v, w_t$  that lie in the span of the  $N$  training instances  $\phi_v(v_i)$  and  $\phi_t(t_i)$ :

$$\begin{aligned} w_v &= \sum_i \alpha_i \phi_v(v_i), \\ w_t &= \sum_i \beta_i \phi_t(t_i), \end{aligned} \quad (4.5)$$

where  $i \in \{1, \dots, N\}$ . The objective of KCCA is thus to identify the weights  $\alpha, \beta \in \mathbb{R}^N$  that maximize:

$$\alpha^*, \beta^* = \arg \max_{\alpha, \beta} \frac{\alpha^T K_v K_t \beta}{\sqrt{\alpha^T K_v^2 \alpha \beta^T K_t^2 \beta}}, \quad (4.6)$$

where  $K_v$  and  $K_t$  denote the  $N \times N$  kernel matrices over a sample of  $N$  pairs. As shown by Hardoon [72], learning may need to be regularized in order to avoid trivial solutions. Hence, we penalize the norms of the projection vectors and obtain the standard eigenvalue problem:

$$(K_v + \kappa I)^{-1} K_t (K_t + \kappa I)^{-1} K_v \alpha = \lambda^2 \alpha. \quad (4.7)$$

The top  $D$  eigenvectors of this problem yield basis  $A = [\alpha^{(1)} \dots \alpha^{(D)}]$  and  $B = [\beta^{(1)} \dots \beta^{(D)}]$  that we use to compute the semantic projections of any vector  $v_i, t_i$ .

### Implementation Details

In order to avoid degeneracy with non-invertible Gram matrices and to increase computational efficiency we approximate the Gram matrices using the Partial Gram-Schmidt Orthogonalization (PGSO) algorithm provided by Hardoon *et al.* [72]. As suggested in [72] the regularization parameter  $\kappa$  is found by maximizing the difference between projections obtained by correctly and randomly paired views of the data on the training set. In the

experiments we have optimized both the parameters of the PGSO algorithm (i.e.  $\kappa$  and  $T$ ); however, we found as a good starting configuration the setting  $T = 30$  and  $\kappa = 0.1$ . We also found important swapping the use of visual and textual spaces as Haroon [72] fixes  $A$  to be unit vectors while computing  $B$  on the basis of the two kernels.

### 4.3.3 Tag Assignment Using Nearest Neighbor Models in the Semantic Space

The intuition underlying the use of nearest-neighbor methods for tag assignment is that similar images share common labels. Following this key idea, we have investigated and applied several NN schemes to our semantic space in order to automatically annotate images. We briefly describe these models below and refer the interested reader to the Chapter 3.

For all baseline methods the  $K$  neighbors of a test image  $I_i$  are selected as the training images  $I_j$  for which our averaged test kernel value  $K_v(I_i, I_j)$ , defined in Eq. 4.2, scores higher. In case the semantic space projection is used, the  $K$  neighbors are computed using:

$$d(\psi(I_i), \psi(I_j)) = 1 - \frac{\psi(I_i)^T \cdot \psi(I_j)}{\|\psi(I_i)\|_2 \cdot \|\psi(I_j)\|_2} \quad (4.8)$$

where  $\psi(I_i)$  is the semantic projection of a test image  $I_i$ . The projection of  $I_i$  is defined as  $\psi(I_i) = K_v(I_i, \cdot)^T A$ , where  $K_v(I_i, \cdot)$  is the vector of kernel values of a sample  $I_i$  and all the training samples. Note that we only use the *visual* view of our data both for training and test samples.

#### KNN

Given a test image, we project onto the semantic space and identify its  $K$  Nearest-Neighbors. Then we merge their labels to create a tag-list by counting all tag occurrences on the  $K$  retrieved images, and finally we reorder the tags by their frequency. If we fix  $K$  to a very small number (e.g.  $K = 2$ ) this approach is similar to the ad-hoc nearest neighbor tag transfer mechanism proposed by Makadia *et al.* [133].

#### TagVote

Li *et al.* [113] proposed a tag relevance measure based on the consideration that if different persons label visually similar images using the same tags,

then these tags are more likely to reflect objective aspects of the visual content. Following this idea it can be assumed that, given a query image, the more frequently the tag occurs in the neighbor set, the more relevant it might be. However, some frequently occurring tags are unlikely to be relevant to the majority of images. To account for this fact the proposed tag relevance measurement takes into account both the distribution of a tag  $t$  in the neighbor set for an image  $I$  and in the entire collection:

$$\text{tagVote}(l, I, K) := n_t[N(I, K)] - \text{Prior}(t), \quad (4.9)$$

where  $n_t$  is an operator counting the occurrences of  $t$  in the neighborhood  $N(I, K)$  of  $K$  similar images, and  $\text{Prior}(t)$  is the occurrence frequency of  $t$  in the entire collection.

### TagProp

Guillaumin *et al.* [68] proposed an image annotation algorithm in which the main idea is to learn a weighted nearest neighbor model, to automatically find the optimal combination of multiple feature distances. Using  $y_{it} \in \{-1, +1\}$  to represent if tag  $t$  is relevant or not for the test image  $I_i$ , the probability of being relevant given a neighborhood of  $K$  images  $I_j \in N(I_i, K) = \{I_1, I_2, \dots, I_K\}$  is:

$$p(y_{it} = +1) = \sum_{I_j \in N(I_i, K)} \pi_{ij} p(y_{it} = +1 | N(I_i, K)), \quad (4.10)$$

$$p(y_{it} = +1 | N(I_i, K)) = \begin{cases} 1 - \epsilon & \text{for } y_{it} = +1, \\ \epsilon & \text{otherwise} \end{cases} \quad (4.11)$$

$$\pi_{ij} \geq 0, \quad \sum_{I_j \in N(I_i, K)} \pi_{ij} = 1, \quad (4.12)$$

where  $\pi_{ij}$  is the weight of a training image  $I_j$  of the neighborhood  $N(I, K)$  and  $p(y_{it} = +1 | N(I_i, K))$  is the prediction of tag  $t$  according to each neighbor in the weighted sum.

The model can be used with rank-based (RK) or distance-based weighting; the latter can be learnt by using a single distance (referred to as the SD variant) or using metric learning (ML) over multiple distances. Furthermore, to compensate for varying frequencies of tags, a tag-specific sigmoid is used to scale the predictions, to boost the probability for rare tags and decrease

that of frequent ones. Sigmoids and metric parameters can be learned by maximizing the log-likelihood  $\sum_{I_i,t} \ln p(y_{it})$ .

## 2PKNN

Verma and Jawahar [195] proposed a two phase method: a first pass is employed to address the class-imbalance by constructing a balanced neighborhood for each test image and then a second pass, where the actual tag importance is assigned based on image similarity.

The problem of image annotation is formulated similarly as Guillaumin *et al.* [68], by finding the posterior probabilities:

$$P(y_{it}|I_i) = \frac{P(I_i|y_{it})P(y_{it})}{P(I_i)} \quad (4.13)$$

Given a test image  $I_i$ , and a vocabulary  $Y = \{t_1, t_2, \dots, t_M\}$ , the first phase collects a set neighborhoods  $T_{it}$  for each tag  $t \in Y$  by selecting at least the nearest  $M$  training images annotated with  $t$ . The neighborhood of image  $I_i$  is then given by  $N(I_i) = \bigcup_{t \in Y} T_{it}$ . It should be noticed that a tag can have less than  $M$  training image and therefore  $N(I_i)$ , may still be a lightly unbalanced set of tags.

On the second phase of 2PKNN, given a tag  $t \in Y$ , the probability  $P(I_i|t)$  is estimated by the neighborhood defined in phase one for image  $I$ :

$$P(I_i|t) = \sum_{I_j \in N(I_i)} \exp(-D(I_i, I_j))p(y_{it} = +1|N(I_i)) \quad (4.14)$$

where  $p(y_{it} = +1|N(I_i))$  is the presence of tag  $t$  for image  $I_i$  as in Guillaumin *et al.* [68] and  $D(I_i, I_j)$  is the distance between image  $I_i$  and  $I_j$ .

In the simplest version of this algorithm  $D(I_i, I_j)$  is just a scaled version of the distance  $wD(I_i, I_j)$ , where  $w$  is a scalar. Authors in [195] also propose a more complex version where  $D(I_i, I_j)$  can be parameterized as a Mahalanobis distance where the weight matrix can be learned in a way that the resulting metric will pull the neighbors from the  $T_t$  belonging to ground-truth tags closer and push far the remaining ones.

## 4.4 Experiments

We evaluate the performance of our cross-media model for tag assignment on three popular datasets and we compare it to closely related work.



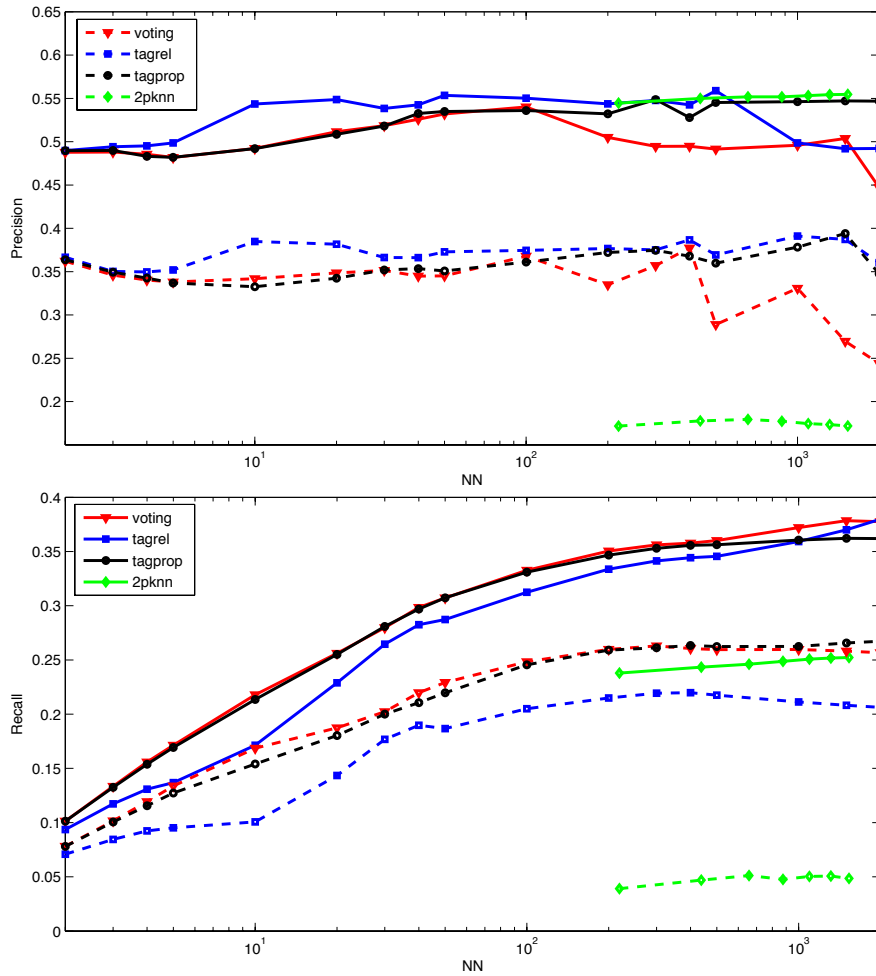


Figure 4.3: Precision and recall of all the methods on MIRFlickr-25k varying the number of nearest neighbors. Dashed lines represent baseline methods. Note that 2PKNN implicitly define the size of the neighborhood based only on the number of images per labels.

(a) Corel5K								
	NN-voting		TagVote [113]		TagProp [68]		2PKNN [195]	
	Baseline	KCCA	Baseline	KCCA	Baseline	KCCA	Baseline	KCCA
<b>P</b>	26	<b>37</b>	25	<b>36</b>	29	<b>35</b>	36	<b>42</b>
<b>R</b>	30	<b>36</b>	35	<b>37</b>	35	<b>40</b>	38	<b>46</b>
<b>N+</b>	135	<b>139</b>	<b>151</b>	144	144	<b>149</b>	169	<b>179</b>

(b) IAPR-TC12								
	NN-voting		TagVote [113]		TagProp [68]		2PKNN [195]	
	Baseline	KCCA	Baseline	KCCA	Baseline	KCCA	Baseline	KCCA
<b>P</b>	32	<b>56</b>	27	<b>57</b>	37	<b>58</b>	46	<b>59</b>
<b>R</b>	21	<b>25</b>	26	<b>28</b>	22	<b>26</b>	29	<b>30</b>
<b>N+</b>	<b>235</b>	213	<b>258</b>	246	225	<b>235</b>	<b>272</b>	259

(c) MIRFlickr-25K								
	NN-voting		TagVote [113]		TagProp [68]		2PKNN [195]	
	Baseline	KCCA	Baseline	KCCA	Baseline	KCCA	Baseline	KCCA
<b>P</b>	34	<b>51</b>	38	<b>50</b>	37	<b>55</b>	16	<b>56</b>
<b>R</b>	26	<b>35</b>	22	<b>37</b>	26	<b>36</b>	6	<b>25</b>
<b>N+</b>	17	<b>18</b>	18	18	18	18	16	<b>18</b>

Table 4.1: This table shows the results of several configurations of our method based on KCCA and baselines on the Corel5K , IAPR-TC12 and MIRFlickr-25K datasets.

#### 4.4.1 Datasets

**Corel5K.** The Corel5K dataset [45] has been the standard evaluation benchmark in the image annotation community for around a decade. It contains 5,000 images which are annotated with 260 labels and each image has up to

	Previously reported results														ML		
	CRM [101]	InfNet [141]	NPDE [221]	MBRM [49]	SML [23]	TGLM [126]	GS [223]	JEC-15 [68]	TagProp $\sigma$ RK [68]	TagProp $\sigma$ SD [68]	RF-opt [54]	KSVML-VT [196]	2PKNN [195]	TagProp $\sigma$ ML [68]	2PKNN ML [195]	Our best result	
<b>P</b>	16	17	18	24	23	25	30	28	26	28	29	32	<b>39</b>	33	<b>44</b>	<b>42</b>	
<b>R</b>	19	24	21	25	29	29	33	33	34	35	40	<b>42</b>	40	42	<b>46</b>	<b>46</b>	
N+	107	112	114	122	137	131	146	140	143	145	157	<b>179</b>	177	160	<b>191</b>	<b>179</b>	

Table 4.2: This table shows the results of our method and related work on the Corel5K dataset (as reported in the literature). JEC-15 refers to the JEC [133] implementation of [68] that uses our 15 visual features.

5 different labels (3.4 on average). This dataset is divided into 4,500 images for training and 500 images for testing.

**IAPR-TC12.** This dataset was introduced in [67] for cross-language information retrieval and it consists of 17,665 training images and 1,962 testing images. Each image is annotated with an average of 5.7 labels out of 291 candidate.

**MIRFlickr-25K.** The MIRFlickr-25K dataset has been recently introduced to evaluate keyword-based image retrieval methods. The set contains 25,000 images that were downloaded from Flickr and for each one of these images the tags originally assigned by the users are available (as well as EXIF information fields and other metadata such as GPS). It is a very challenging dataset since the tags are weak labels and not all of them are actually relevant to the image content. There are also many meaningless words. Therefore a pre-processing step was performed to filter out these tags. To this end we matched each tag with entries in Wordnet and only those tags with a corresponding item in Wordnet were retained. Moreover, we removed the less frequent tags, whose occurrence numbers are below 50. The result of this process is a vocabulary of 219 tags. The images are also manually annotated for 18 concepts (i.e. labels) that are used to evaluate the automatic annotation performances. As in [194], the dataset is divided into 12,500 images for training and 12,500 images for testing.

### 4.4.2 Evaluation Measures

We evaluate our models with standard performance measures, used in previous work on image annotation. The standard protocol in the field is to report Precision and Recall for fixed annotation length [45]. Thus each image is annotated with the  $n$  most relevant labels (usually, as in this chapter, the results are obtained using  $n = 5$ ). Then, the results are reported as mean precision  $\mathbf{P}$  and mean recall  $\mathbf{R}$  over the ground-truth labels;  $\mathbf{N}+$  is often used to denote the number of labels with non-zero recall value. Note that each image is forced to be annotated with  $n$  labels, even if the image has fewer or more labels in the ground truth. Therefore we will not measure perfect precision and recall figures.

### 4.4.3 Results

As a first experiment we compare our method with the corresponding nearest neighbor voting schemes. It can be seen from Table 4.1 that our approach improves over baseline methods in every setting on all datasets. Precision is boosted notably, confirming the better separation of the classes in the semantic space (as previously discussed in Section 4.3). Also recall is improved by a large margin on Corel5K and MIRFlickr-25k. On IAPR-TC12 recall improvement is less pronounced. We believe this is due the different amount of textual annotation: IAPR-TC12 has an average of 5.7 tags per image (TPI) and up to 23 TPI while on Corel5K and MIRFlickr-25k the average TPI is respectively 3.4 and 4.7 with a maximum of 5 and 17 TPI respectively. Recalling that we are predicting  $n = 5$  tags per image, recall is harder to improve on this dataset.

We conduct an evaluation of how the amount of neighbours affect the performance for both our method and the baseline on the challenging MIRFlickr-25k dataset. As can be seen from Figure 4.3 the KCCA variants (solid lines) of the four considered voting schemes systematically improve both precision and recall for any amount of nearest neighbors used. Note that in both cases, a similar pattern emerges due the natural instability of NN methods.

It is interesting to note that while recall gets better as the neighborhood gets bigger, saturating at near 2,000 neighbours, precision depends on the algorithm chosen. Basic voting and TagVote show an improvement until 200 neighbors and then begin decreasing; TagProp improves until saturates at around 900.

	Previously reported results							ML		Our best result
	MBRM [49]	GS [223]	JEC-15 [68]	TagProp $\sigma$ SD [68]	RF-opt [54]	K SVM-VT [196]	2PKNN [195]	TagProp $\sigma$ ML [68]	2PKNN ML [195]	
<b>P</b>	24	32	29	41	44	47	<b>49</b>	46	<b>54</b>	<b>59</b>
<b>R</b>	23	29	19	30	31	29	<b>32</b>	35	<b>37</b>	<b>30</b>
<b>N+</b>	223	252	211	259	253	268	<b>274</b>	266	<b>278</b>	<b>259</b>

Table 4.3: This table shows the results of our method and related work on the IAPR-TC12 dataset (as reported in the literature).

2PKNN misses a direct parameter to choose the dimension of the neighborhood, but it implicitly defines it by choosing at most  $M$  images per label. However, while it has a clear advantage on Corel5K and IAPR-TC12, both as a baseline and after the projection, it fails to achieve comparable performance on MIRFlickr-25K. We believe that this is due to the noisy and missing tags of MIRFlickr-25K, a notable difference on this more realistic and challenging dataset.

Comparing with the state of the art, on Tables 4.2 and 4.3, our method achieves better performance than all previous works while it is comparable with the state of the art method 2PKNN [195] on Corel5K. Our method performs slightly worse than 2PKNN in metric learning configuration. However, metric learning involves a learning procedure with many parameters that rise the complexity of optimization and undermines scalability.

Our method, once learned the semantic space, continues to work in what we call an open world setting. In this setting that is indeed more realistic, the amount of tags per image evolves over time. That is the case of big data from social media and, more in general, from the web.

We also report in Table 4.4 a comparison with the methods presented in [68, 194] using per-image average precision (iAP). This measure indicates how well a method identifies relevant concepts for a given image. Our method combining the 2PKNN voting scheme, without metric learning, with the semantic projection outperforms all the other methods.

	Previously reported results					ML	<b>Our best result</b>
	random	SVM v	SVM t	SVM v+t	TagProp RK	TagProp ML	
<b>iAP</b>	5.6	44.2	32	45	46.3	47.3	<b>50.8</b>

Table 4.4: This table shows the results of our method and related work [194] on the MIRFlickr-25k dataset.

### Qualitative Analysis

In Figure 4.4 we present some anecdotal evidence for our method (from the MIRFlickr-25k dataset). It can be seen that TagProp and TagVote perform better in general for the baseline representation and our proposed KCCA variant. It has to be noted that for challenging images where visual features can be deceiving our cross-modal approach allows to retrieve more tags. As an example see the first two rows: a close-up of a flower and a cloudy sunset with a road. For the first one it is not surprising that visual features do not provide enough good neighbors to retrieve the *flower* tag. For the second one none of the baseline method can retrieve the *sunset* and *cloud* tags; we believe that this is due to the lack of color features. In this two cases it is clear that semantically induced neighbors in the common space can boost the accuracy.

Another challenging example is shown at row five: a *girl* is depicted behind an object that hides a part of the face. This image component do not have enough visual neighbors to retrieve its tags. With our representation we are able to retrieve *girl* and *portrait* in the first three voting schemes and also *people* in the TagProp voting scheme, though *face* and *woman* may be considered correct even if not present in the ground truth tags.

## 4.5 Conclusions

We presented a cross-media model based on KCCA to perform tag assign-








	Baselines				KCCA models			
	NN-voting	TagVote	TagProp	2PKNN	NN-voting	TagVote	TagProp	2PKNN
	dog graffiti people black art	dog graffiti animal people house	graffiti dog people face art	graffiti dog people face art	<b>flower</b> flowers pink green spring	<b>flower</b> flowers pink green red	<b>flower</b> flowers green pink white	graffiti dog people face art
	sky clouds water landscape trees	clouds <b>sky</b> landscape water trees	clouds <b>sky</b> water landscape trees	clouds <b>sky</b> water landscape trees	clouds <b>sky</b> landscape <b>sunset</b> blue	clouds <b>sky</b> <b>sunset</b> landscape <b>cloud</b>	clouds <b>sky</b> landscape <b>sunset</b> beach	clouds <b>sky</b> water landscape trees
	japan art water dog trees trees	japan zoo dog trees art	japan water dog park art	japan water dog park art	<b>portrait</b> <b>girl</b> <b>tree</b> street green	<b>portrait</b> <b>girl</b> woman <b>tree</b> trees	<b>portrait</b> <b>girl</b> green <b>tree</b> trees	japan water dog park art
	pink flower japan baby portrait	pink baby japan cake crochet	pink japan flower japanese vintage	pink japan flower japanese vintage	<b>food</b> chocolate cake fruit red	<b>food</b> cake chocolate dog crochet	<b>food</b> chocolate cake red fruit	pink japan flower japanese vintage
	japan <b>people</b> man street bicycle	japan man <b>people</b> bicycle animal	japan <b>people</b> animal kid eye	japan <b>people</b> animal kid eye	<b>portrait</b> <b>girl</b> girls hair face	<b>portrait</b> <b>girl</b> face woman hair	<b>portrait</b> <b>girl</b> face <b>people</b> woman	japan <b>people</b> animal kid eye
	street architecture beach white snow	street snow architecture beach home	beach street people portrait landscape	beach street people portrait landscape	beach <b>sea</b> clouds <b>sky</b> <b>water</b>	beach <b>sea</b> sunset ocean clouds	beach <b>sea</b> clouds ocean <b>water</b>	beach street people portrait landscape
	green garden people flower spring	green man waterfall garden bird colours	green grass garden feet water	green grass garden feet water	dog <b>animal</b> zoo green dogs	dog <b>animal</b> animals puppy dogs	dog <b>animal</b> zoo dogs green	green grass garden feet water

Figure 4.4: Anecdotal results of the baseline methods and our proposed representation for a set of challenging images (MIRFlickr-25K dataset). The tags are ordered by their relevance scores.

ment. We learn semantic projections for both textual and visual data. This representation is able to provide better neighbors for voting algorithms. The experimental results show that our method makes consistent improvements over standard approaches based on a single-view visual representation as well as other previous work that also exploited tags. We report also experiments on a challenging dataset collected from Flickr and our results show that the performance of the proposed method is boosted even further in a realistic scenario such as the one provided by weakly-labelled images. Possible extensions of this work include the exploration of how richer textual and semantic cues from natural language annotations might improve our model.





# Chapter 5

## Fisher Encoded Bag-of-Windows Representation

*This chapter presents an efficient and powerful method to aggregate a set of Deep Convolutional Neural Network responses, extracted from a set of image windows. We show how to use Fisher Vectors and PCA to obtain a short and highly descriptive signature that can be used for effective image retrieval. We show also how the very good performance in retrieval can be exploited for social image tagging. State-of-the art results is reported for both tasks of image retrieval and tag assignment on standard datasets.<sup>1</sup>*

### 5.1 Introduction

In this chapter we address the problem of image retrieval and tag assignment in the context of social media. In the first task we aim at obtaining a very compact and discriminative signature, that allows the creation of scalable image retrieval systems. The goal of the second task is to predict, for a given image, a finite set of tags from a given vocabulary, serving as a compact description of the image. A popular group of recent image

---

<sup>1</sup>This chapter previously appeared as “Fisher Encoded Convolutional Bag-of-Windows for Efficient Image Retrieval and Social Image Tagging” in *Proc. of International Conference on Computer Vision 2015, 3rd Workshop on Web-scale Vision and Social Media (VSM)*.

annotation methods apply tag propagation using diversely defined neighborhoods [8, 68, 113, 117, 133, 195] (see also Chapter 2). These approaches have been successfully applied to the context of social and user generated media, that are typically annotated with tags that are likely to correlate with image content. However, this rich source of metadata is often hard to exploit both for the noise in labels and for the difficulty to find semantically meaningful visual features. Clearly a good image representation boosts the precision and recall of these techniques by providing a visually consistent neighborhood. In fact, many of these techniques apply a form of metric learning to make up for low quality image features. We point out that an essential requirement of these techniques is the ability to retrieve similar images to compose good image neighborhoods. Hence, excelling in image retrieval is likely to improve image tagging. A recent breakthrough in image representation has been achieved using Convolutional Neural Networks (CNN) with deep architectures. It has been shown that using a large corpus of images CNNs can learn compact and powerful image features. CNNs are typically applied to classification tasks and activations from the latest layers are used as features. These have been used by several approaches to extract generic features for image retrieval [64, 222]. While they show promising results, they leave several questions unaddressed. First, CNNs features are more semantically related to the global image and they hardly preserve local characteristics of objects. Second, while previous approaches address CNNs limited invariant to scale with multi-scale extraction, their approach is onerous due to the requirement of extracting dense patches at multiple scales.

Recently Wei *et al.* [209] have applied a multi-label variation of CNN extracting features from few hundred object proposals. We agree with their intuition and we believe that multiple windows of an image can be carefully selected in order to obtain a more comprehensive representation of image content. This is particularly relevant in the case of image tagging where more than one tag is sought. User tags may refer to the image as a whole but they are also likely to be associated with specific scene elements. Specifically, tags often refer to *things* (e.g. person, car, horse, etc.) and *stuff* (e.g. sky, sand, cloud, water, etc.) present in a scene.

In this chapter we show a technique to combine CNN features from multiple windows into a more discriminative representation for image retrieval and image tagging. This representation improves upon the single global representation approach, obtaining state-of-the-art results with compact image

signatures on three popular public dataset.

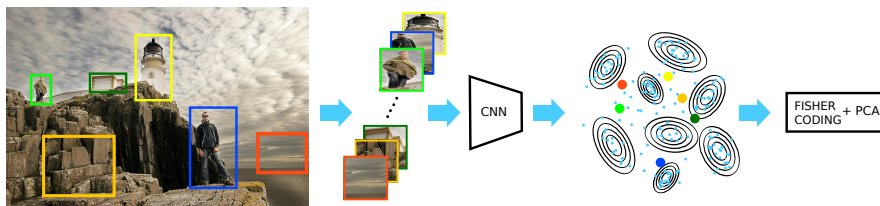


Figure 5.1: Full pipeline of the proposed method. Each image window is represented by the FC7 CNN activations. The final signature is obtained encoding activations (same color dots) with a Fisher Vector computed on a GMM dictionary (blue dots). PCA is further applied to image signature.

## 5.2 Previous work

So far, the best performance in image retrieval has been obtained aggregating SIFT descriptors using Fisher Vectors [80,167], VLAD [3,80], or variations of these approaches e.g. pooling oriented local features [224]. A breakthrough in performance for computer vision algorithms has recently been obtained thanks to supervised image feature learning. Krizhevsky *et al.* revived supervised deep learning for computer vision proposing to solve large scale image classification problem using deep CNN [97]. Following that, several architectures have been proposed in the last 3 years, all sharing a common principle: networks are usually built with a sequence of convolutional/max-pooling layers, followed by low-resolution fully-connected (FC) layers whose activations are fed to a soft-max classifier.

The most interesting fact about CNNs is the ability to perform transfer learning. Indeed a very powerful image representation can be obtained by removing the soft-max classifier and keeping the activations of the last FC layer. This approach has been applied to many computer vision and multimedia retrieval tasks, with dramatic improvements over previously proposed techniques such as Fisher Vectors over local SIFT descriptors. Razavian *et al.* [161] made a comprehensive contribution on this matter testing CNN features for: object and scene classification, attribute prediction and image retrieval. However, their spatial search approach in image retrieval has an unbearable computational cost: their method requires the extraction of CNN

features for a large amount of image sub-windows and the computation of all pairwise distances between them. The approach has scalability issues, since it is quadratic in the number of windows. Approaches close to ours have been proposed in [64, 149, 222]; Gong *et al.* [64], propose to CNN responses from multiple scales using VLAD, thus requires a dense computation of multi-scale CNN responses. In contrast, we show how we can rely on the computation of CNN responses on a few hundreds of proposal windows. Ng *et al.* [149] have speeded up the approach of [64] applying the network only once to the input image and extracting features at each location of the convolutional feature map of each layer. Yoo *et al.* [222] propose to apply Fisher Vector encoding to dense multi-scale CNN activations. Compared to both these methods our approach computes CNN activations on large parts of the image, that are likely to contain objects, rather than considering CNN activations of dense and small patches, that are more similar in spirit to SIFT descriptors. Another difference is that we introduce a simpler and effective multi-scale representation by concatenating the Fisher Vector with a global representation of image content, and reducing the overall descriptor size with PCA.

The identification of relevant patches in an image has been recently addressed in the object detection community, with the introduction of window proposal methods [59, 191]. Object proposals are cheap to compute and cover more than 90% of objects with few thousands boxes of different scales and aspect ratios. This allows the application of expensive classifiers like [59] or kernelized bag-of-words classifiers [191] to perform object detection.

Regarding the task of social image tagging, our work is related to instance based tag assignment methods [117]. Makadia *et al.* [133], in their seminal work, showed that simple tag voting on nearest neighbor outperformed previous complex approaches. Li *et al.* [113] improved upon by adding a penalty on frequent tag votes. As low-level features are hardly semantically related, Guillaumin *et al.* [68] and Verma *et al.* [195] proposed to learn a weighted metric to improve on precision. Ballan *et al.* [8] proposed using KCCA to learn mid-level features to be used with previous nearest neighbors approaches.

## 5.3 Proposed method

Our idea is to represent an image as a bag of windows, each one represented as CNN output activations. The final image signature is obtained using Fisher Encoding and reducing the final descriptor dimensionality using PCA, as shown in Figure 5.1.

This powerful novel image signature is used to boost performance in image retrieval and social image tagging.

### 5.3.1 Image representation

**Patch Sampling** We start by sampling a set of few hundred windows from each image to construct a bag-of-boxes  $\mathcal{X}$  as image representation. We use the object proposal approach from Zitnick *et al.* [232] due to its computational efficiency and performance in terms of detection, recall and repeatability [73]. Nonetheless, this step may be integrated using a set of random windows. In fact, we found in some experiments that employing a set of randomly sampled windows in addition to the Edgeboxes may be beneficial. This is motivated by the fact that some discriminative portions of images, often useful for retrieval are not part of objects or *things* but rather are referred as *stuff*, i.e. part of larger textured regions like trees or mountains.

CNN usually require, as it is in our case, a fixed size input patch. To this end we resize each window to  $224 \times 224$  pixels disregarding the aspect ratio, as it is common practice in object detection [59]. We use the CNN-S-128 CNN architecture from [24] in order to have a low dimensional representation (128D), comparable to that of SIFT.

**Activation Aggregation** To obtain a short signature for each image we perform an aggregation step. Given a set of patches  $x \in \mathcal{X}$ , we encode it using Fisher Encoding.

We first learn a Mixture of Gaussians codebook with diagonal covariances on a subset of the windows extracted at the previous step. Differently from [167] we do not apply PCA on the local window features. This is not needed, and actually slightly worsen the performance in our case, since our window representation has highly decorrelated features. In Fig. 5.2 we show a comparison of the absolute values of correlation coefficients  $\rho$  among dimensions of CNN codes and SIFT descriptors extracted from the INRIA

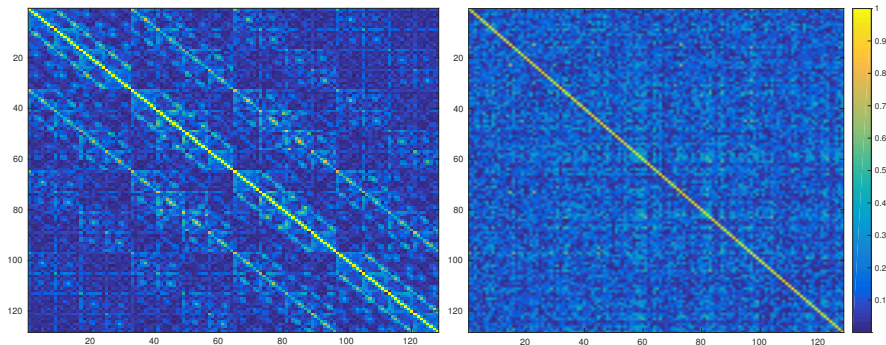


Figure 5.2: Correlation coefficients computed on a set of SIFT descriptors (*left*) and on a set of CNN features on image windows (*right*).

Holidays dataset. The  $\rho$  coefficients of the CNN codes are 1 only on the diagonal, while as a counter-example on SIFT descriptors extracted from the same dataset there are many directions with  $|\rho| > .8$ .

For each bag-of-boxes we compute an Improved Fisher Vector (IFV) applying L2 and Power Normalization as in [167]. Finally to compress the representation we reduce the dimensionality of the IFVs using PCA.

**Global-Local signature** The classical approach for image representation with CNN, is to resize the image to a fixed square size and compute the activations from the first fully connected layer (FC7). This approach, although discards some information about the image, has been proved to be very powerful [161] as we can also observe from our baseline experiments in Tab. 5.1, Tab. 5.2 and Tab. 5.3. Similarly as in [64] we show how, especially for tagging, this single window signature can help. We propose a Global-Local signature concatenating the full image encoding to the IFV encoding by applying PCA. With this technique we try to leverage two different ways of aggregating image responses in a single compact signature.

### 5.3.2 Image retrieval

The first task we address with our novel image representation is image retrieval. To retrieve images means that given an image as query we want to rank a dataset of images in order to assign high ranks to images with the same content of the query. We perform this task in a very straightforward

manner. Given a query image  $I$  and a dataset of images  $Y_i$ , we consider their respective sets of image window features  $\mathcal{I}$  and  $\mathcal{Y}_i$  and signatures  $\phi(\mathcal{I})$  and  $\phi(\mathcal{Y}_i)$ . For each query  $I$  we rank images by cosine distances:

$$d(I, Y_i) = 1 - \frac{\sum \phi(\mathcal{I}) \cdot \phi(\mathcal{Y}_i)}{\|\phi(\mathcal{I})\|_2 \|\phi(\mathcal{Y}_i)\|_2}$$

### 5.3.3 Tag Assignment

A collection of social images, e.g. obtained from Flickr, can be modeled as a set of tuples  $\mathcal{T}_i = \langle Y, \mathcal{W} \rangle$  where  $Y$  is an image and  $\mathcal{W}$  is a set of tags; when performing image annotation we would like to predict tags for an untagged image  $I$ . This problem is usually solved with voting algorithms based on nearest neighbor search [9, 68, 113], because of their scalability and relatively good performance [117]. We use the ranking described in Sect. 5.3.2 to obtain the first  $K$  neighbors, and use the following three different algorithms (refer also to Chapter 3 for an extended description).

**NN voting** The simplest voting algorithm is nearest neighbor tag voting, which is close to the method first proposed by Makadia *et al.* [133]. We count the tag occurrences of images in the neighborhood and rank tags per image using their frequencies.

**Tag Relevance** With NN voting we assume that the more frequently the tag occurs in the neighbor set, the more relevant it might be for the image. However tags occurring frequently in the whole training set are not necessary relevant for all the images. So to moderate this effect, Li *et al.* [113] proposed a tag relevance measure that takes into account both the tags distributions of the neighbor set and of the entire training set.

**TagProp** Guillaumin *et al.* [68] have proposed TagProp, a method that learns a weighted nearest neighbor model.

Weights can be learned based on distance or rank. Moreover, to compensate for varying frequencies of tags, a tag-specific sigmoid is used to boost the probability for rare tags and decrease that of frequent ones. Sigmoids and metric parameters can be learned by maximizing the log-likelihood of tag predictions.

## 5.4 Experiments

**Datasets** For the image retrieval task we use the popular INRIA Holidays dataset [81]. The dataset is composed by 1,491 images in total. We measure average precision (AP) for 500 queries and 991 corresponding relevant images.

We test image tagging on the challenging MIRFLICKR-25K and NUS-WIDE datasets. The MIRFLICKR-25K dataset [75] is composed of 25,000 images with 1,386 tags that is split in 12,500 for training and 12,500 for testing, with exactly the same partition as [8,68]. Images are weakly labeled with tags from Flickr. As in [8], we keep the 219 tags that have an entry in WordNet and whose frequency is at least 50. Manual annotations for 18 tags are provided on the whole set. In the following experiments we propagate the whole set of tags and measure precision and recall on the 18 manual annotations for each image. The NUS-WIDE dataset [32] is composed of 259,233 images with 355,913 tags. Also in this case images are weakly labelled with Flickr tags, and ground truth is available for 81 tags.

**Baselines** The natural baseline for our method is the extraction of a single CNN code per image. We refer to this baseline as CNN-Image. We warp the whole image to  $224 \times 224$  and use the FC7 output as image signature. We develop another baseline by averaging the output of all the CNN features of the bag-of-boxes, and refer to it as AVG-Pooling. We test this variation in order to see if the use of an aggregated signature is relevant to keep the expressiveness of the many windows extracted or if sampling multiple CNN responses is enough to boost retrieval and annotation performance.

**Experimental results: retrieval** We first evaluate the parameters affecting retrieval performance on INRIA Holidays, evaluated in terms of mean average precision (MAP). In a set of preliminary experiments we found that the final PCA step slightly improves results but not significantly. This step is indeed mostly relevant to compress the image signature. The size of the GMM codebook is instead extremely relevant for performance.

Increasing the number of Gaussians allows to model the distribution of CNN activations more precisely, as it has been observed also for SIFT features [167], where increasing the number of Gaussians improves the performance. To see how the codebook size affects retrieval performance we fixed



the final PCA dimension to 512 which we found improving performance across codebook sizes.

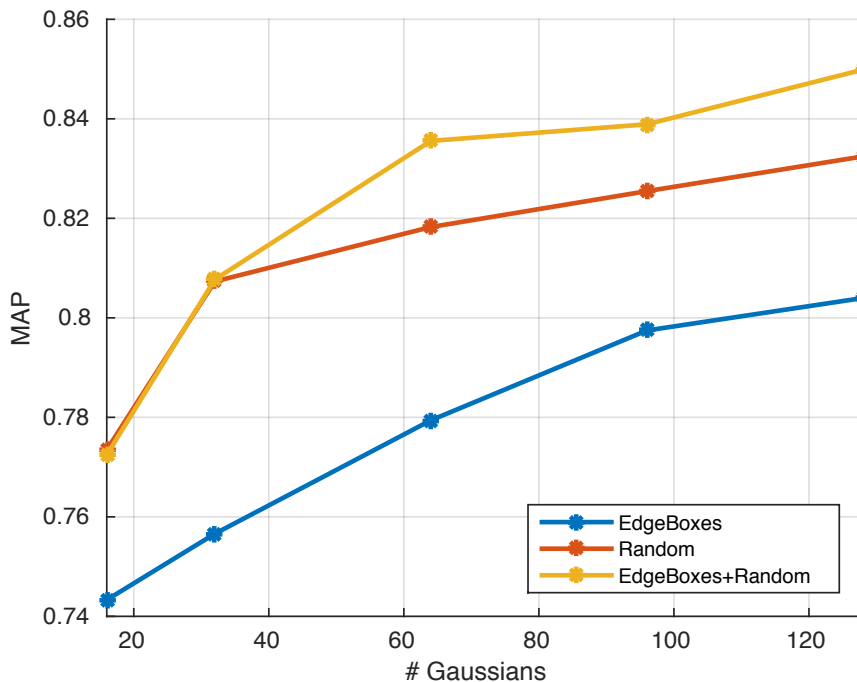


Figure 5.3: Mean average precision of our proposed approaches varying the number of Gaussians on Holidays dataset.

In Figure 5.3 we evaluate the performance of the proposed approach with a varying number of Gaussians and with different types of windows. We can see how using EdgeBoxes alone for retrieval is not sufficient. Adding random boxes increases the performance also for a small amount of Gaussians (32). In Figure 5.4 we report MAP values obtained using different numbers of Edgeboxes and random windows, with different encoding. The combination with the global signature does not improve the MAP for large codebooks but instead allows to get very high results even for small codebooks. Fisher vectors always outperform max and average pooling. In Figure 5.5 we evaluate the performance of Fisher Vector + PCA coding with varying number of windows, either from Edgeboxes, random, or Edgeboxes + random sampling. As for Fig.5.4 it can be observed that FV + PCA outperforms the sue

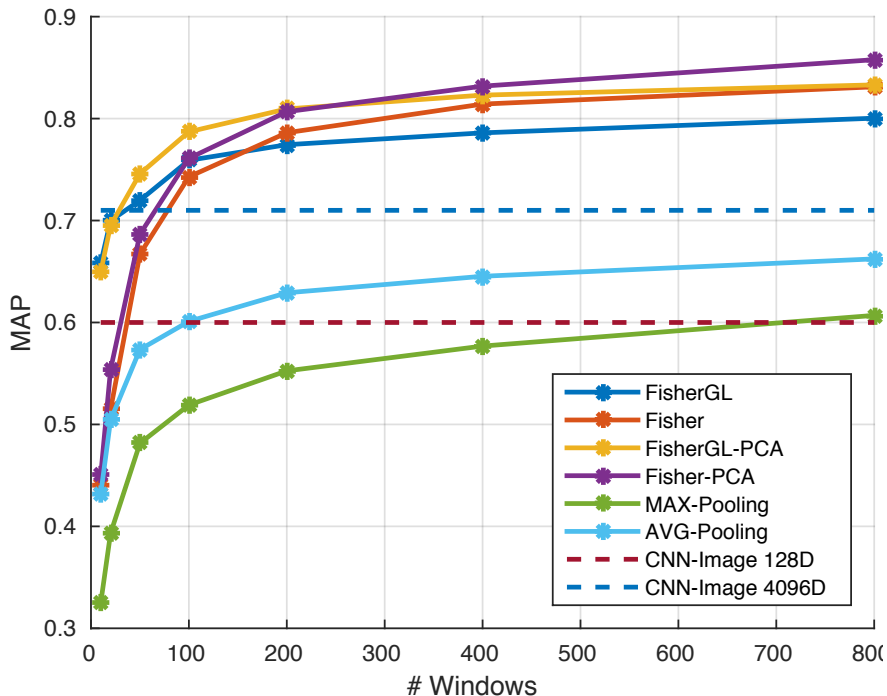


Figure 5.4: Mean average precision of our proposed approaches varying the number of Edgeboxes + Random windows.

of single global CNN descriptors, when using  $> 100$  windows. Considering the random boxes step we report the average of five runs.

Finally, we compare our method with other global methods aggregating local features in Table 5.1 and some recent methods that use either convolutional or fully connected layers of CNNs [5, 64, 149, 161, 162]. We can clearly see that although the 128D CNN is competitive with some smaller size representations based on SIFT features [80] the 4096D outperforms all the approaches based on engineered features. Average pooling of 128D activations outperforms the single image 128D representation indicating that more information is contained in multiple windows. Adoption of Improved Fisher Vector coding improves over the majority of the other methods based on CNN features except [149, 161]. Finally we can see how applying the Fisher encoding and PCA outperforms all other methods, including [149, 161], with

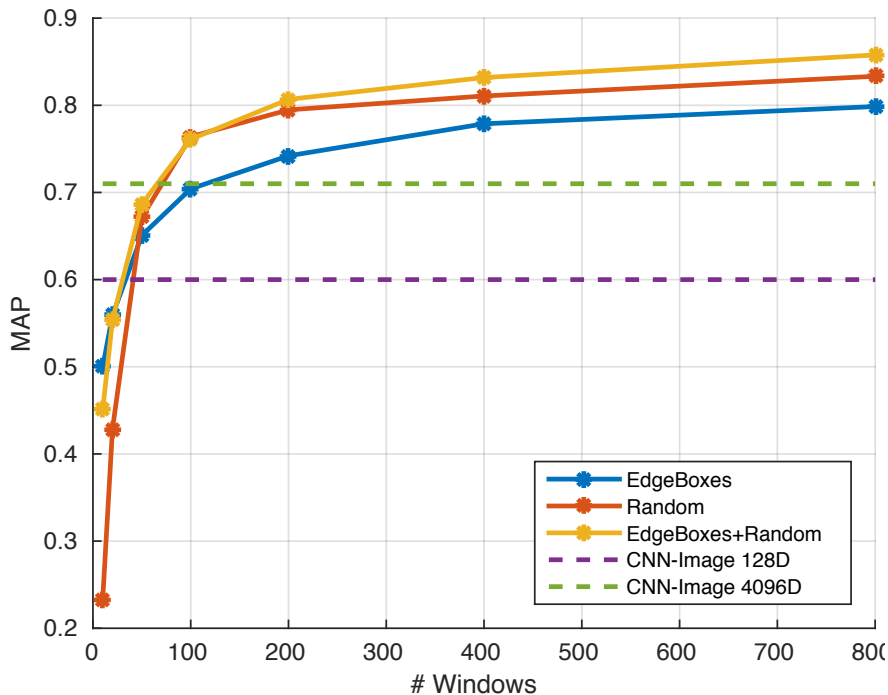


Figure 5.5: Mean average precision of our proposed approaches varying the number of windows, using Fisher-PCA coding.

a very small signature (512D).

**Experimental results: tagging** In this set of experiments we show how our novel representation improves performance on image tagging. We report results as Mean Average Precision (MAP) and Mean image Average Precision (MiAP) in Tab. 5.2 and Tab. 5.3. MAP measures the quality of image ranking and can be affected by the performance on rare tags, while MiAP measures the quality of tag ranking and is biased toward frequent tags [117]. To speedup computations on these larger datasets we have used a GMM codebook of only 32 elements and half the number of windows w.r.t. the experimental setup used for retrieval, but reducing the dimension of the final IVF to 512 dimensions as in the previous case.

Because of the high variability of the images of these datasets the use

Method	Features	Codebook	Dim.	MAP
<b>Fisher-PCA</b>	FC7-CNN	128	512	<b>85.8</b>
Fisher-GL-PCA	FC7-CNN	128	635	83.3
Fisher-GL	FC7-CNN	128	32,889	81.2
Fisher	FC7-CNN	128	32,768	80.3
AVG-Pooling	FC7-CNN	–	128	66.2
CNN-Image 128D	FC7-CNN	–	128	60.0
CNN-Image 4096D	FC7-CNN	–	4,096	71.0
Spatial Pooling [162]	CONV-CNN	–	256	74.2
CNNaug-ss [161]	FC7-CNN	–	4,096	84.3
VLAD+PCA [149]	CONV-CNN	100	128	83.6
Neural codes [5]	FC7-CNN	–	128	78.9
VLAD+PCA [64]	FC7-CNN	100	2,048	80.8
VLAD+PCA [64]	FC7-CNN	100	512	74.2
Fisher [167]	SIFT	4,096	524,288	70.0
Zhao [224]	SIFT	32	32,768	68.8
Delhumeau [39]	SIFT	64	8,192	65.8
Arandjelovic [3]	SIFT	256	32,536	65.3
Fisher [80]	SIFT	256	16,384	62.5
Fisher [80]	SIFT	64	4,096	59.5
VLAD [80]	SIFT	256	16,384	58.7
VLAD [80]	SIFT	64	4,096	55.6

Table 5.1: Image retrieval results on INRIA Holidays compared with state-of-the-art approaches.

of the Global-Local component of the descriptor, that accounts for scales variations, improves the results. In this case the single image approach outperforms [194]. This means that CNN features are indeed a strong representation for image annotation. In this case average pooling is not improving over the single image approach. Finally we can see how adding the Global Local part of the descriptor boosts MAP and MiAP for all voting methods; compressing the descriptor with PCA further improves the results except for a few cases on MIRFLICKR-25K, however in these cases the differences are minimal. It has to be noted that TagProp always outperforms the simpler

NN Voting and TagRel methods, exploiting better the improved visual neighborhood obtained with the proposed method. This is visible when comparing the performance obtained with the single CNN-Image descriptor w.r.t. that of Fisher-GL-PCA.

Features	NN Voting		TagRel		TagProp	
	MAP	MiAP	MAP	MiAP	MAP	MiAP
<b>Fisher-GL-PCA</b>	<b>51.4</b>	<b>48.6</b>	47.6	51.4	<b>58.0</b>	54.8
Fisher-GL	50.9	48.0	<b>48.4</b>	<b>51.5</b>	57.9	<b>54.9</b>
Fisher-PCA	46.1	44.9	43.7	48.2	51.6	50.9
Fisher	46.2	45.2	44.0	48.2	51.6	50.8
MAX-Pooling	40.7	45.6	41.5	47.1	47.6	49.2
AVG-Pooling	40.2	45.0	40.5	46.6	45.9	48.6
CNN-Image	48.3	46.6	46.0	50.1	55.7	53.7
LEAR [194]	–	–	–	–	38.4	47.3

Table 5.2: Image annotation results on MIRFLICKR-25K compared with the state-of-the-art (200 Edgeboxes + 200 random windows).

Features	NN Voting		TagRel		TagProp	
	MAP	MiAP	MAP	MiAP	MAP	MiAP
<b>Fisher-GL-PCA</b>	<b>26.7</b>	<b>43.4</b>	<b>27.7</b>	<b>40.1</b>	<b>39.7</b>	<b>50.9</b>
Fisher-GL	26.8	43.4	27.6	40.1	39.7	50.8
Fisher-PCA	21.7	40.4	24.1	37.0	35.9	48.0
Fisher	21.3	40.3	23.6	36.6	35.5	47.4
MAX-Pooling	18.8	37.8	22.1	34.9	29.1	45.0
AVG-Pooling	19.9	40.2	22.4	37.1	29.8	45.9
CNN-Image	24.4	42.0	25.3	38.7	31.9	48.2

Table 5.3: Image annotation results on NUS-WIDE compared with the state-of-the-art (200 Edgeboxes + 200 random windows).

## 5.5 Conclusion

In this chapter we have shown the importance of extracting CNN activations from multiple windows. We found out that using proposal methods and

randomly sampled features improves object proposal windows alone. We show that encoding multiple CNN activations from the same image using Fisher vectors boosts image retrieval and image annotation performance. We tested our approach on three datasets reporting state-of-the-art results with short 512D signatures.

## Chapter 6

# Evaluating Temporal Information in Social Images

*Can we use the temporal gist of annotations in Web images to improve tasks such as annotation, indexing and retrieval? Typically visual content and text, are used to improve these tasks. A characteristic that has received less attention, so far, is the temporal aspect of social media production and tagging. This chapter gives a thorough analysis of the temporal aspects of two popular datasets commonly used for tasks such as tag ranking, tag suggestion and tag refinement, namely NUS-WIDE and MIR-Flickr-1M. The correlation of the time series of the tags with Google searches shows that for certain concepts web information sources may be beneficial to annotate social media.<sup>1</sup>*

### 6.1 Introduction

Typically visual content, text and metadata, such as geo-tags, are used to improve tasks such as annotation, indexing and retrieval of the huge quantities of media produced every day by the users of such systems. For instance, visual content similarity is used in [113] to perform tag suggestion and image retrieval, tag co-occurrence has been proposed in [174] for tag suggestion,

---

<sup>1</sup>This chapter has been published as “Evaluating Temporal Information for Social Image Annotation and Retrieval” in *Proc. of International Conference on Image Analysis and Processing (ICIAP), 2013, pp. 722-732.*

geo-tags have been used in [176] for tag recommendation, content classification and clustering. A recent review of the state-of-the-art in areas related to web-based social communities and social media has been presented in [180], considering in particular the contribution of contextual and social aspects of media semantics to multimedia applications.

A characteristic that has received less attention, so far, is the temporal aspect of social media production. As noted in [1], extracting time information from documents may improve several applications such as hit-list clustering and exploratory search. More recently, several researchers have shown that the temporal information associated to search engine queries (e.g. frequency of query keywords over time) can be used to predict trends and behaviors related to economics and medicine, such as claims for unemployment benefits [31], and detection of flu epidemics [57].

In [160] “burst” analysis techniques derived from signal processing are compared against a novel method to identify social events in the associated social media, using the tags and geo-localization information of Flickr images. In [96], the temporal evolution of topics in social image collections is proposed to perform subtopic outbreak detection and to classify noisy social images. The authors used a non-parametric approach in which images are represented using a similarity network, created using Sequential Monte Carlo, where images are the vertices and the edges connect the temporally related and visually similar images. Temporal dynamics of social image collections has been studied in [94] to improve search relevance at query time, addressing both a general case and personalized interest searches. The authors propose a unified statistical model based on regularized multi-task regression on multivariate point process, in which an image stream is considered an instance of a process and a regression problem is formulated to learn the relations between image occurrence probabilities and temporal factors that influence them (e.g. seasons).

Analysis of the temporal evolution of social media collections have been proposed in [84] to predict political success and product sales; regression-based and diffusion-based models have been adapted to account for a Flickr-based index, combining images’ metadata and visual similarity, that models the popularity of politicians and products. The work presented in [95] recasts the problem of image retrieval re-ranking as a prediction of which images will be more likely to appear on the web at a future time point. Both collective group level and individual user level cases are considered, using a



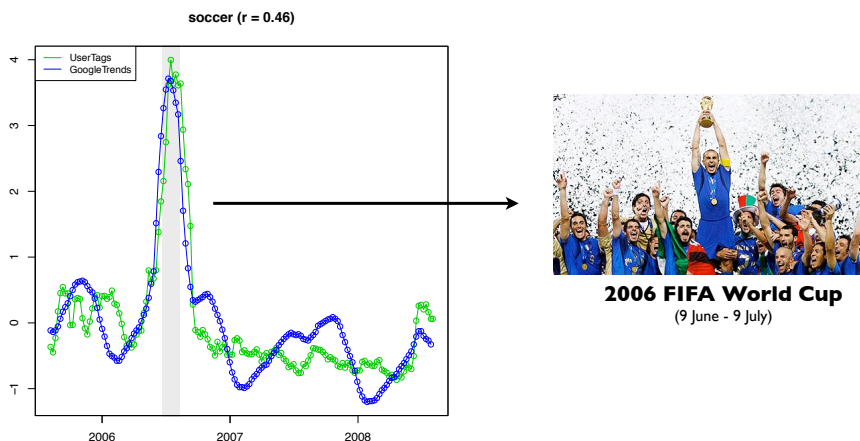


Figure 6.1: Time series of user tags and Google searches for “soccer” in NUS-WIDE dataset.

multivariate point process to model a stream of input images, and using a stochastic parametric model to solve the relations between the occurrences of the images and factors such as visual clusters, user descriptors and month of the image.

All the datasets used in these works are based on custom selections of user-generated images selected from Flickr, and are not publicly available. The main contribution of this chapter is a thorough analysis of the temporal aspects of two “standard” datasets commonly used for tasks such as tag ranking, tag suggestion and tag refinement [123] [113] [228] [125] [192]: NUS-WIDE [32] and MIR-Flickr-1M [75]. These datasets provide images and associated metadata, along with a ground-truth annotation of 81 and 18 tags, respectively. Analysis of the temporal evolution of both user tags and ground-truth tags allows to evaluate the social context (e.g. use of tags related to the semantics associated to social interaction, and not necessarily associated with image content) and visual content (e.g. use of tags that are more strictly related to image content). The correlation of the time series of the tags with Google searches (see Fig. 6.1) shows that for certain concepts web information sources may be beneficial to annotate social media.

## 6.2 Data Analysis Method

### 6.2.1 Datasets

To measure the impact of temporal information for image annotation purposes, we performed a quantitative analysis over two image datasets: NUS-WIDE [32] and MIR-Flickr-1M [75].

NUS-WIDE is a large scale dataset collected from Flickr. It contains 269,648 images, provided as multiple visual features and source URLs, with 5,018 tags of which 81 have been manually checked and can be considered ground-truth tags. Tab. 6.2.1 reports the classification of these tags according to their main WordNet category. In order to obtain all temporal metadata not contained in the set, we had to download again all the original images from Flickr. Unfortunately, some images are not available anymore, therefore we had to use a subset of 238,251 images that are still present on Flickr. We refer to this subset as NUS-WIDE-240K. Images are unbalanced with respect to time, having very different number of images per date. The time interval goes from year 1900 (old photo scans) to 2009, concentrating most of the images between 2005-2008.

MIR-Flickr-1M is also a large dataset crawled from Flickr which contains 1 million images, selected by their Flickr interestingness score [198] [74]. Every image provided has full *Flickr metadata* which includes *taken* and *posted* timestamps, indicating when a photo was taken and when it was shared on Flickr. However, only about half of the images provide a valid “taken” timestamp, in particular only 584,892 are valid, as 330,454 have no timestamps and 84,654 have an invalid timestamp. Like NUS-WIDE-240K, images are unbalanced with respect to time. Images are concentrated around years 2007-2009. A ground-truth comprised of 18 tags is provided for the first 25,000 images only, that compose a subset called MIR-Flickr25K [74].

Object	12	Animal	13	Location	2	Substance	2
Action	5	Plant	4	Top	4	Time	2
Artifact	26	Event	4	Phenomenon	4	Person + Groups	3

Table 6.1: WordNet categories of NUS-WIDE ground-truth tags.

### 6.2.2 Temporal features

Given a set of images  $I$ , all taken in a set of dates  $D$  (as a daily interval), we denote as  $T$  the set of all tags used and  $U$  the set of all users. For every image  $i \in I$  we denote  $\text{tag}(i) \subseteq T$  the set of tags associated,  $\text{day}(i) \in D$  the timestamp associated and  $\text{user}(i) \in U$  the user who owns the image. We also consider two other time spans, a set of weeks  $W$  and a set of months  $M$ , easily computed by integrating over the interval of days considered. These can be thought as time series over the selected index set. For every set considered, we computed a set of features, as proposed in [95]:

- **Images per day:** the number of relevant images which are *taken* in a day. More specifically, given a day  $d \in D$ , the number of images per day (IMD) is defined as

$$\text{IMD}(d) := |\{i \in I | \text{day}(i) = d\}| \quad (6.1)$$

Similarly we also define a feature for the number of images per week (IMW) and per month (IMM).

- **Images per day for a tag:** the number of relevant images associated with a tag which are *taken* in a day. More specifically, given a tag  $t \in T$  and a day  $d \in D$ , the number of images with  $t$  per day (ITD) is defined as

$$\text{ITD}(t, d) := |\{i \in I | \text{day}(i) = d \wedge t \in \text{tag}(i)\}| \quad (6.2)$$

Similarly we also define a feature per week (ITW) and per month (ITM).

However, a phenomenon associated with a social source is that of *batch tagging*: a user may decide to upload an entire album of photos and, instead of carefully tagging each photo, he could simply opt to tag each photo with the same tags (e.g. tag the album instead of every single photo). This may result in a kind of noise with respect to the normal use of tags in time. In addition, the features defined above are sensitive to this kind of noise, producing noisy peaks over single days. To produce a more meaningful analysis we decide to collapse all images that are batch tagged into a single entry. A set of images are considered *batch tagged* if they are all uploaded by the same user on the same day and have the same set of tags. More specifically, given a user  $\hat{u} \in U$ , a day  $\hat{d} \in D$  and a set of tags  $\hat{t} \subseteq T$ , a

set of images  $I_B = \{i_1, i_2, \dots, i_k\}$  are considered *batch tagged* if  $\text{tag}(i) = \hat{t}$ ,  $\text{user}(i) = \hat{u}$ ,  $\text{day}(i) = \hat{d} \forall i \in I_B$ .

### 6.2.3 Flickr Popularity Model

As described in [84], available images from the two datasets are only a sample of all images in Flickr. In addition, the number of images over time in Flickr are mostly variable, based on the popularity of the site itself. This slow change over time can be modeled as a trend over all tags, independent from any particular query. Unfortunately, no statistics are released publicly and other sources such as Alexa<sup>2</sup> or Google Trends<sup>3</sup> are affected by the impact of news. Based on this preliminary analysis and supposing an uniform sampling in Flickr searches, we use the feature IMD to remove this background deviation by normalizing the ITD feature.

Given a tag  $t \in T$  and a date  $d \in D$  we compute:

$$\overline{ITD}(t, d) = \frac{ITD(t, d)}{IMD(d)} \quad (6.3)$$

This may also be considered as a frequentist probability distribution of tag  $t$  in day  $d$  with respect to all other tags considered, which is  $p(t; d)$ . Similarly we also compute  $\overline{ITW}$  and  $\overline{ITM}$  by considering a week and a month granularity, respectively. After collapsing all batch tagged images, the two datasets retain 179,128 images for NUS-WIDE-240K and 531,670 images for MIRFLICKR-1M respectively.

### 6.2.4 Processing

First of all we present a qualitative analysis by measuring the occurrence of tags in time. Given that NUS-WIDE-240K has the biggest ground truth of all datasets considered and that we are looking to discover the relations between tags and image content with respect to time, we choose to use it as the main reference. We use all the 81 manually checked tags as  $T$  set and consider four different information sources which are different in the kind of underlining latent process :

---

<sup>2</sup>Alexa Internet, Inc. <http://www.alexa.com>

<sup>3</sup>Google Trends. <http://www.google.com/trends>

- From NUS-WIDE-240K, for all images, we consider the  $T$  set of tags using the **manually validated** tags which constitute the entire ground truth; we refer to this source as **NUS-GT**.
- From NUS-WIDE-240K, for all images, we consider the  $T$  set of tags using the **user tags** (e.g. the tags provided by the respective Flickr users); we refer to this source as **NUS-TAGS**.
- From MIRFLICKR-1M, for all images, we consider the  $T$  set of tags using the **user tags**; we refer to this source as **MIR-TAGS**.
- Beside image datasets, we also consider a source of temporal query information given by Google Trends. From Google Trends, we have downloaded all available query data for the  $T$  set of tags considered; we refer to this source as **GOO-TAGS**.

All sources are to be considered subject to different kinds of noise, in particular all images are highly unbalanced over time, resulting in days with hundreds of images and others with at most ten images. To reduce this effect, we choose to consider only the largest time span with at least 350 images per week. In addition the two image datasets differ in the time interval which has the most images. This forced us to use a reduced time interval that we choose as starting from 2005-06-01 and ending in 2008-08-01 for NUS-WIDE-240K (retaining 161,176 images from 179,128) and from 2007-01-01 to 2008-08-01 for MIR-Flickr-1M (retaining 110,064 images from 531,670). Those filters were processed with a combination of Python scripts and Google Refine<sup>4</sup>. After this we used the R package [183] to plot and execute any successive analysis. A plotting of features of this data revealed an insufficient reduction in noise to be able to clearly visualize most characteristics pattern. To make the time series patterns more clear, we computed a simple moving average over all time series, varying the windows size  $n$  from 2 to 10 weeks. For a day time series defined over a time span  $\Psi$  for a tag  $t \in T$  is defined as:

$$ITD_n(t, d) = \frac{1}{n} \sum_{i=-n}^n \overline{ITD}(t, d+i) \quad \forall d \in \Psi \quad (6.4)$$

This has the effect to smooth the series, letting to visualize more clearly the trend. On the other hand, tags which have very sparse frequency tends to be

<sup>4</sup>Google Refine. <http://code.google.com/p/google-refine>

worsened, so we adjusted the window size empirically, based on visualization clearness. The final time series are composed of 1,158 and 579 week samples respectively for NUS-WIDE-240K and MIR-Flickr-1M.

### 6.2.5 Correlation analysis

To exploit the underlying time process and to be able to improve image annotation using temporal information, we need a way to evaluate quantitatively the possible correlation between sources. This allows us to analyze if a series can be estimated by another one and how a generalized model may describe the original time series. To this end we compute a correlation measure over two series. First of all we standardize all time series: given a time series  $X = \{x_i : i \in D\}$ , we compute  $x_i = \frac{x_i - \bar{X}}{s}$ , where  $\bar{X}$  is the sample mean and  $s$  is the sample standard deviation. Even if sample mean and sample standard deviation are sensible to outliers, those are removed thanks to the filtering and smoothing procedure described above. To evaluate the correlation between two time series, we choose to use the *sample Pearson correlation coefficient*, often denoted as  $r$ . Given two time series  $X$  and  $Y$  of  $n$  samples,  $r$  is defined as the ratio between covariance and the product of  $X$  variance and  $Y$  variance:

$$r = \frac{\sum_{i=1}^n (x_i - \bar{X})(y_i - \bar{Y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{X})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{Y})^2}} \quad (6.5)$$

which is defined in  $[-1, 1]$ . Values towards the positive or negative end reveal a strong correlation between the two time series, changing only in the sign. We can reformulate it as the mean of the products of the standard scores, which permits us to use standardized time series  $\hat{x}_i = \frac{x_i - \bar{X}}{s_X}$  and  $\hat{y}_i = \frac{y_i - \bar{Y}}{s_Y}$ :

$$r = \frac{1}{n-1} \sum_{i=1}^n \left( \frac{x_i - \bar{X}}{s_X} \right) \left( \frac{y_i - \bar{Y}}{s_Y} \right) = \frac{1}{n-1} \sum_{i=1}^n \hat{x}_i \hat{y}_i \quad (6.6)$$

Given that the strength of correlation is not dependent on the direction or the sign, we also computed r-square. Unfortunately the interpretation of a correlation coefficient depends heavily on the context and purposes that can't be easily defined at this stage of work. However several works like [34] offered some guidelines which can be used to interpret our analysis, that are reported in Tab. 6.2.

Correlation	None	Small	Medium	Strong
Positive	0.0 to 0.09	0.1 to 0.3	0.3 to 0.5	0.5 to 1.0
Negative	-0.09 to 0.0	-0.3 to -0.1	-0.5 to -0.3	-1.0 to -0.5

Table 6.2: Guidelines for sample Pearson correlation coefficient.

## 6.3 Experiments and Discussion

In the following we will consider both the presence of the tags that have been added by the users that uploaded the images to Flickr (referring to them as “user tags”) and the tags that have been manually checked by the creators of NUS-WIDE as referring to visual content of images (referring to them as “ground-truth” tags). In fact, several studies have shown that tags are often ambiguous and personalized [91] [174], and do not necessarily reflect the visual content of the image. As an example consider Fig. 6.2 and 6.3, showing the temporal usage of the tags “snow” and “soccer” in NUS-WIDE, along with the respective Google searches, as obtained from Google Trends. It can be observed that the peak in usage of the “soccer” tag - associated with the 2006 FIFA World Cup - reflects that in Google Trends, but the peak is much less pronounced in the ground truth tags; this indicates that for this tag the relationship between tags and image may exist because of how people react to social events, rather than uploading photos depicting that event on Flickr. On the other hand the peaks of both user and ground truth “snow” tag are corresponding to that of Google Trends: in this case the relationship may exist because it is more likely that people take pictures of snow scenes during winter, and this concept is less related to social aspects than to visual content of these images.

### 6.3.1 Temporal Evaluation

Considering time series composed of the frequencies of image tags (either user or ground-truth) and Google searches obtained from Google Trends, it is possible to observe that they exhibit the presence of different components, that may appear mixed together:

**trend** long term variation, that can be increasing, decreasing or also stable (see Fig. 6.4). Terms such as “computer” or “military” have this pattern;

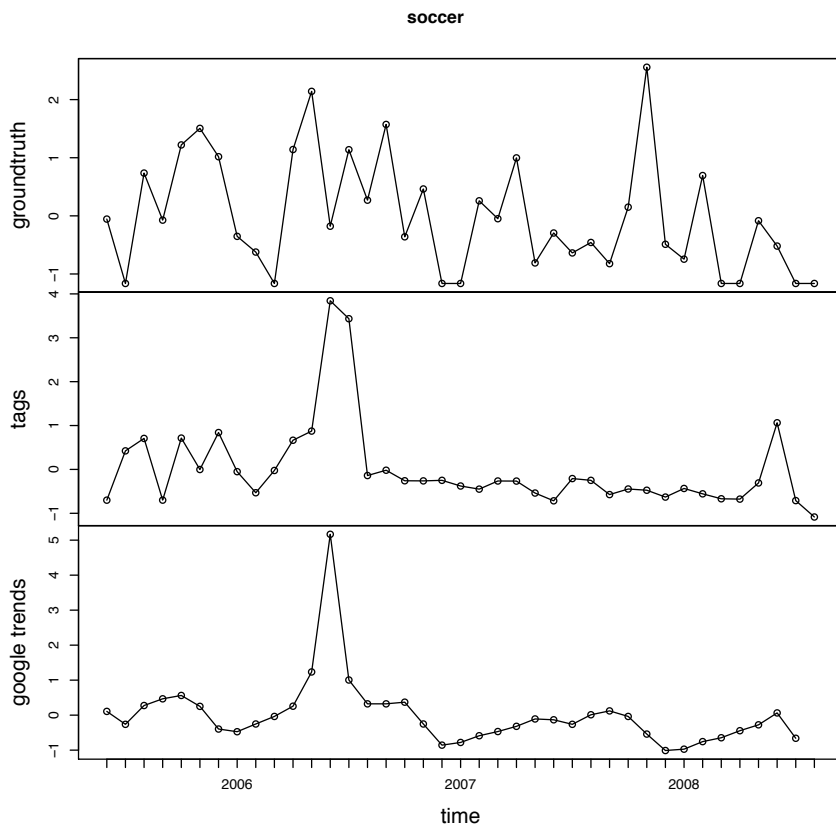


Figure 6.2: Frequency of “soccer” in NUS-GT, NUS-TAGS and GOO-TAGS: the peak of Google Trends and user tags in the summer of 2006 are related to the World Soccer Championship.

**cyclical variation** repeated but not periodic variations. Tags like “sports” or “flags” have this pattern;

**seasonal variation** periodic variations, e.g. due to concepts associated with some regular event (see Fig. 6.4). Concepts related to seasons show this behavior, like “garden”, “snow”, “beach” or “frost”;

**irregular variation** random irregular variations, e.g. due to the sudden emergence of a topic (see Fig. 6.5), that appears as a burst of activ-



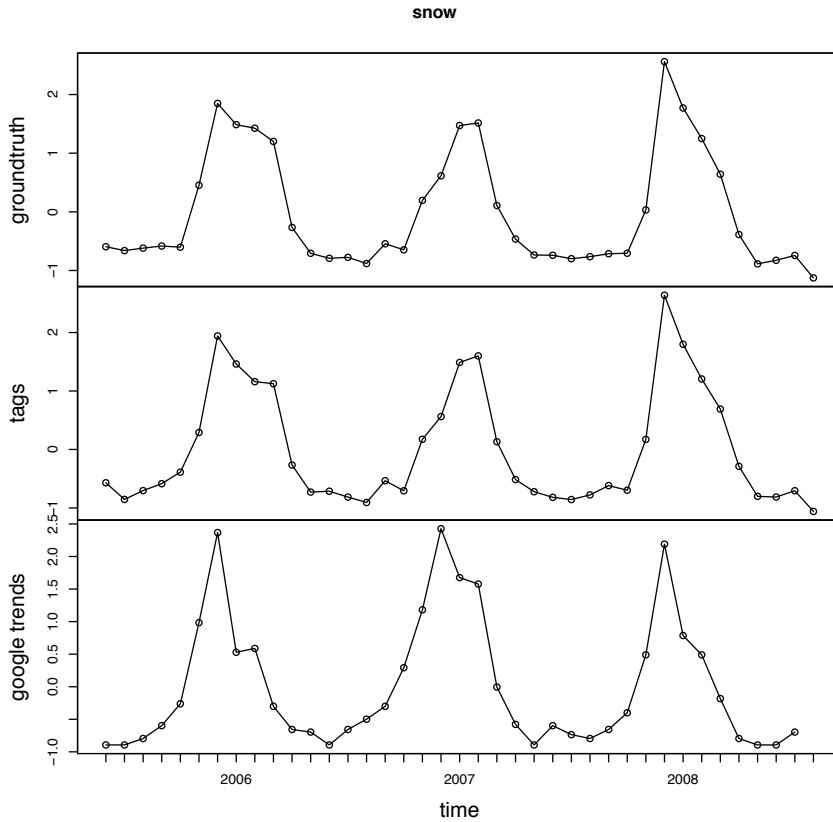


Figure 6.3: Frequency of “snow” in NUS-GT, NUS-TAGS and GOO-TAGS: the peaks are associated with winter seasons. Tag frequencies have been normalized by the number of images of the same day.

ity. Concepts that exhibit this pattern are related to social or natural events like “soccer”, “earthquake” and “protest”.

### 6.3.2 Correlation Analysis

Fig. 6.6 reports the outcome of correlation analysis of NUS-TAGS with NUS-GT, NUS-TAGS with GOO-TAGS and NUS-GT with MIR-TAGS. In par-

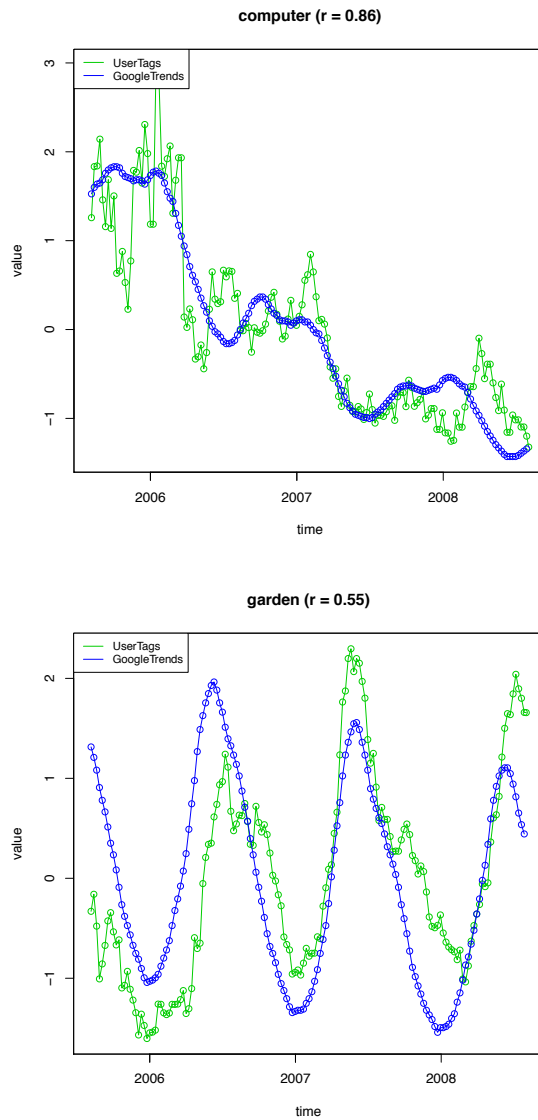


Figure 6.4: Time series patterns of NUS-TAGS and GOO-TAGS, averaged over 10 weeks. *i*) trend (computer); *ii*) seasonal (garden).

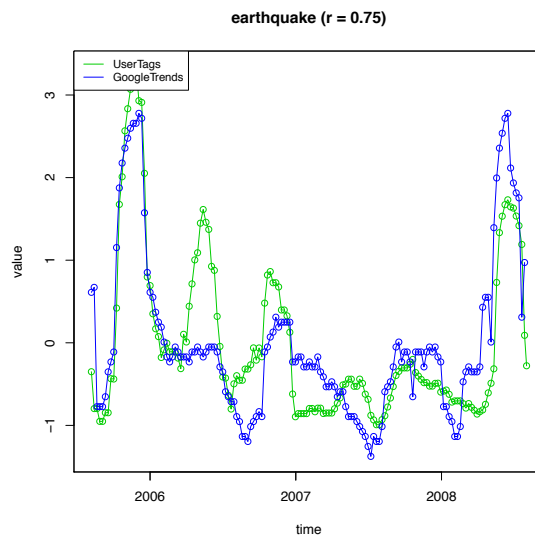


Figure 6.5: Time series patterns of NUS-TAGS and GOO-TAGS, averaged over 10 weeks. Episodic behavior (earthquake: peaks correspond to earthquakes in China and Pakistan).

ticular it can be observed that the correlation of NUS-TAGS and NUS-GT has a vast majority of “Medium” and “Strong” values, while the correlation between user tags and Google searches is overall weaker and can be useful for a selected number of tags. The correlation between NUS-GT and MIR-TAGS has a large number of “Medium” and “Strong” values, suggesting that the temporal information of NUS-WIDE can be used in MIR-Flickr-1M.

Correlation analysis of NUS-TAGS with GOO-TAGS, followed by averaging of r-square values over tags classes (Fig. 6.7 left) shows that Plant, Event, Phenomenon and Action obtain the higher values. A second group of categories comprises Artifact, Person+Group, Animal, Object and Time. In general, the categories that obtain the best performances are benefitting from tags whose time series show seasonal behaviors (e.g. “snow”, “frost”, “grass”, “leaf”) or have a “burst” behavior associated with specific social events (e.g. “soccer”, “protest”, “earthquake”).

Correlation analysis of NUS-GT with GOO-TAGS (Fig. 6.7 right) shows that Plant and Phenomenon categories maintain their position among the best performing classes, because of the tags that exhibit a seasonal pattern. Instead the correlation of Event and Action categories is lower because the ground-truth tags that have an episodic pattern like “soccer”, “protest” and “earthquake” have a lower correlation. This is due to the fact that these tags are employed by users also when the content of the image is not visually related to the described event.

## 6.4 Conclusions

This chapter presented a thorough analysis of the temporal aspects of user annotations in two popular large-scale datasets. The correlation of the time series of the tags with Google searches showed that for certain concepts web information sources may be beneficial to annotate social media.

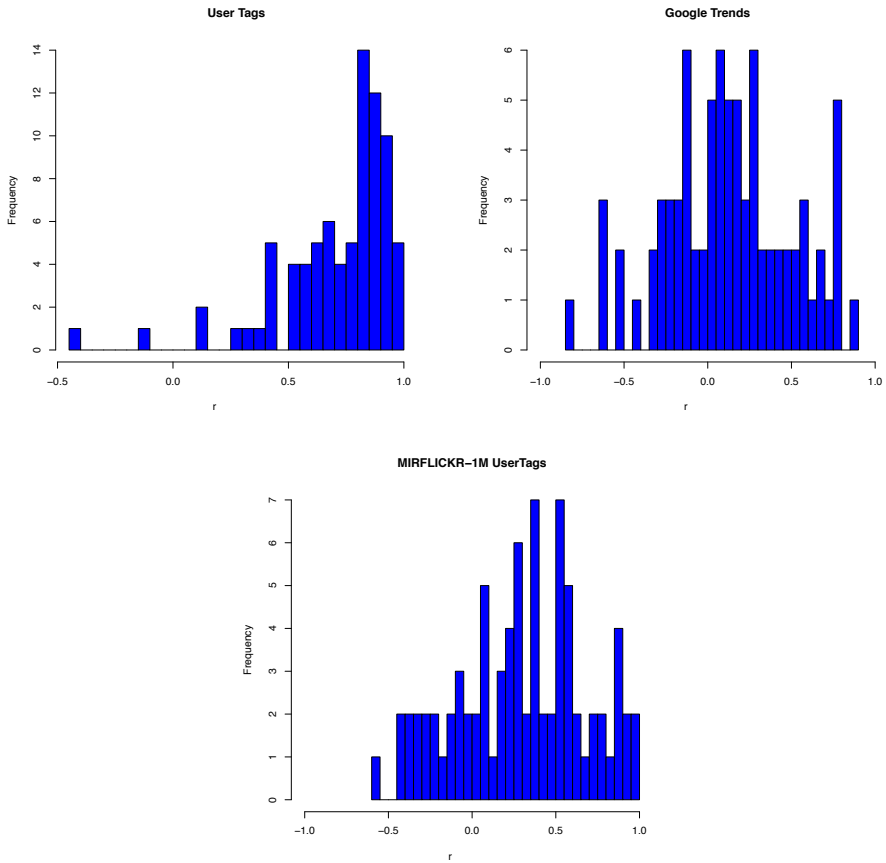


Figure 6.6: *i*)  $r$  values computed between NUS-TAGS and NUS-GT; *ii*)  $r$  values computed between NUS-TAGS and GOO-TAGS; *iii*)  $r$  values computed between NUS-GT and MIR-TAGS.

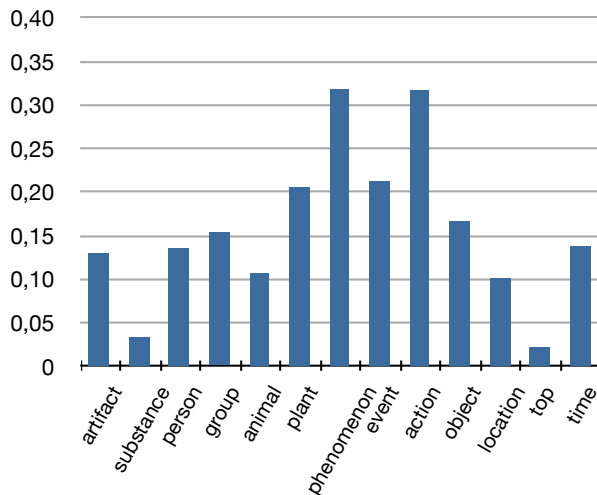
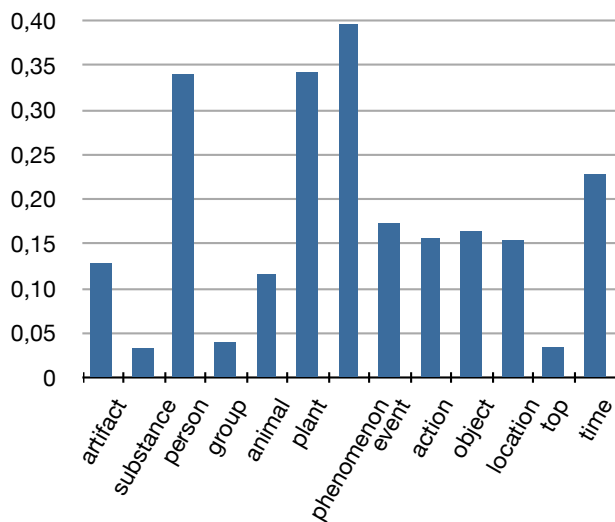
**Avg. R-square per category (NUS-TAGS / GOO-TAGS)****Avg. R-square per category (NUS-GT / GOO-TAGS)**

Figure 6.7: NUS-WIDE dataset: r-square averages for tags classes. *i*) NUS-TAGS correlation with GOO-TAGS; *ii*) NUS-GT correlation with GOO-TAGS.

## Chapter 7

# Multimodal Feature Learning for Sentiment Analysis

*In this chapter we investigate the use of a multimodal feature learning approach, using neural network based models such as Skip-gram and Denoising Autoencoders, to address sentiment analysis of micro-blogging content, such as Twitter short messages, that are composed by a short text and, possibly, an image. Motivated by the recent advances of unsupervised learning of language models and visual features based on neural networks models, we propose a novel architecture that incorporates these models and test it on several standard Twitter datasets. We show that the approach is efficient and obtains good classification results.*<sup>1</sup>

### 7.1 Introduction

In the last few years micro-blogging services, in which users describe their current status by means of short messages, obtained a large success among users. Unarguably, one of the most successful services is Twitter<sup>2</sup>, that is used worldwide to discuss about daily activities, to report or comment news,

---

<sup>1</sup>This chapter previously appeared as “A Multimodal Feature Learning Approach for Sentiment Analysis of Social Network Multimedia” in *Multimedia Tools and Applications*, DOI: 10.1007/s11042-015-2646-x

<sup>2</sup>Twitter reports to have 271 million monthly active users that send 500 million status updates per day - <https://about.twitter.com/company>

and to share information using messages (called ‘tweets’) composed by at most 140 characters. Since 2011 Twitter natively supports adding images to tweets, easing the creation of richer content. A study performed by Twitter<sup>3</sup> has shown that adding images to tweets increases user engagement more than adding videos or hashtags.

Despite their brevity these messages often convey also the feeling and the point of view of the people writing them. The addition of images reinforces and clarifies these feelings (see Fig.7.1). Automatic analysis of the sentiment of these tweets, i.e. retrieving the opinion they express, has received a large attention from the scientific community. This is due to its usefulness in analyzing a large range of domains such as politics [189] and business [56]. Sentiment analysis may encompass different scopes [20]: *i*) polarity, i.e. categorize a sentiment as positive, negative or neutral; *ii*) emotion, i.e. assign a sentiment to an emotional category such as joy or sadness; *iii*) strength, i.e. determine the intensity of the sentiment.

So far, the vast majority of works have addressed only the textual data. In this chapter we address the classification of tweets, according to their polarity, considering both textual and visual information. We propose a novel schema that, by incorporating a language model based on neural networks, can efficiently exploit web-scale sources corpus and robust visual features obtained from unsupervised learning. The proposed method has been tested on several standard datasets, showing promising results.

The chapter is organized as follows: Sect. 7.2 provides an overview of previous works; the proposed method is presented in Sect. 7.3, while experiments on four standard datasets and comparison with state-of-the-art approaches and baselines are reported in Sect. 7.4. Conclusions are drawn in Sect. 7.5.

## 7.2 Previous Work

**Sentiment analysis in texts.** Brevity, sentence composition and variety of topics are among the main challenges in sentiment analysis of tweets (and micro-blogs in general). In fact these texts are short, often they are not composed carefully as news or product reviews, and cover almost any conceivable topic. Several specific approaches for Twitter sentiment analysis have been proposed, typically using sentence-level classification with  $n$ -gram

---

<sup>3</sup><https://blog.twitter.com/2014/what-fuels-a-tweets-engagement>



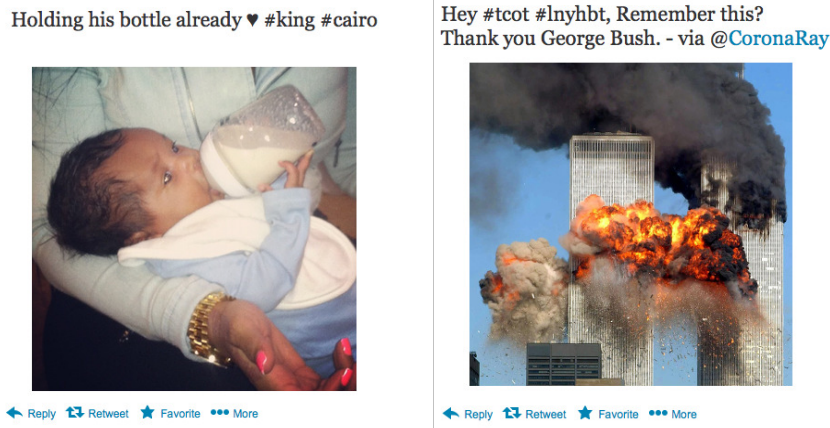


Figure 7.1: Examples of tweets with images from the SentiBank Twitter dataset [19]. *left*) positive sentiment tweet; *right*) negative sentiment tweet.

word models. Liu *et al.* [129] concatenate tweets of the same class (polarity) in large documents, from which a language model is derived and then classify tweets through maximum likelihood estimation, using both supervised and unsupervised data for training; the role of unsupervised data is to deal with words that do not appear in the vocabulary that can be built from a small supervised dataset. In [16] three approaches to sentiment classification are compared: Multinomial Naïve Bayes (MNB), Hinge Loss with Stochastic Gradient Descent and Hoeffding Tree; the authors report that MNB outperforms the other approaches. In [38] unigram and bigram features have been used to train Naïve Bayes classifiers, where bigrams help to account for negation of words. Saif *et al.* [165] have evaluated the use of a Max Entropy classifier on several Twitter sentiment analysis datasets. Since using  $n$ -grams on tweet data may reduce classification performance due to the large number of infrequent terms in tweets, some authors have proposed to enrich the representation using micro-blogging features such as hashtags and emoticons as in [10], or using semantic features as in [166].

**Neural networks language models.** Recently, the scientific community has addressed the problem of learning vector representations of words that can represent information like similarity or other semantic and syntactic re-

lations, obtaining better results than using the best  $n$ -gram models. The use of neural networks to perform this task is motivated by recent works addressing the scalability of training. In this formulation every word is represented in a distributional space where operations like concatenation and averaging are used to predict other words in context, trained by the use of stochastic gradient descent and backpropagation. In the work of [13], a model is trained based on the concatenation of several words to predict the next word: every word is mapped into a vector space where similar words have similar vector representations. A successive work uses multitask techniques [35] to jointly train several tasks showing improvements in generalization. A fast hierarchical language model was proposed in [146], attacking the main drawback of needing long training and testing times. The use of unsupervised additional words was proposed by [190] showing further improvements using word features learned in advance to a supervised NLP task. Recently Mikolov *et al.* [143] have proposed several improvements on Hierarchical Softmax [146] and Negative Sampling [70] and introduced the Skip-gram model [145], reducing further the computational cost, and showing fast training on corpora of billions of words [143]. More recently, researchers also extended these models, trying to achieve paragraph and document level representations [103].

**Micro-blog multimedia analysis.** Most of the works dealing with analysis of the multimedia content of micro-blogs have dealt with content summarization and mining, image classification and annotation. Geo-tagged tweet photos are used in [90, 217] to visually mine events using both textual and visual information. The system presented in [173] provides tools for content curation, creation of personalized web sites and magazines through topic detection of tweets and selection of representative associated multimedia. A system for exploration of events based on facets related to who, when, what, why and how of an event, has been presented in [208], using a Bilateral Correspondence model (BC-LDA) for image and words. A multi-modal extension of LDA has been proposed in [15] to discover sub-topics in microblogs, in order to create a comprehensive summarization.

An algorithm for photo tag suggestion using Twitter and Wikipedia are used in [139] to annotate social media related to events, exploiting the fact that tweets about an event are typically tweeted during its development. Classification of tweets' images in visually-relevant and visually-irrelevant, i.e. images that are correlated or not to the text of the tweet, has been

studied in [28], using a combination of text, context and visual features.

Zhao *et al.* [225] have studied the effects of adding multimedia to tweets within Sina Weibo, a Chinese equivalent of Twitter, finding that adding images boosts the popularity of tweets and authors, and extends the lifespan of tweets.

**Sentiment analysis in social images.** Sentiment analysis of visual data has received so far less attention than that of text data and, in fact, only a few small datasets exist, such as the International Affective Picture System (IAPS) [100] and the Geneva Affective Picture Database (GAPED) [36]. The former provides ratings of emotion (in terms of pleasure, arousal and dominance) for 369 images, while the latter provides 520 images associated to negative sentiment, 89 neutral and 121 positive images. Another related direction is given by works on aesthetics: surveys are provided in [88, 207]. However, none of these datasets deal with social media.

A few works have addressed the problem of multimedia sentiment analysis of social network data. Borth *et al.* [19] have recently presented a large-scale visual sentiment ontology and associated set of detectors, consisting of 3,244 pairs of nouns and adjectives (ANP), based on Plutchik's Wheel of Emotions [155]. Detectors are trained using Flickr images, represented using a combination of global (e.g. color histogram and GIST) and local (e.g. LBP and BoW) features. The paper provides also two publicly available image datasets obtained from Flickr and from Twitter. The system proposed in [22] for the classification of Sina Weibo statuses exploits the ANP detectors proposed in [19], fusing them with text sentiment analysis based on 3 features: *i*) sentiment words from HowNet (Chinese equivalent to WordNet), *ii*) semantic tags and *iii*) rules of sentence construction, to cope with rhetorical questions, negations and exclamatory sentences.

Cross-media bag-of-words, combining bag of text words with bag of image words obtained from the SentiBank detectors of [19], has been proposed in [206] for sentiment analysis of microblog messages obtained from Sina Weibo. Yang *et al.* [220] have proposed a hybrid link graph for images of social events, weighting links based on textual emotion information, visual similarity and social similarity. A ranking algorithm to discover emotionally representative images in microblog statuses is then presented. The work of Chen *et al.* [30], distinguishes between the intended publisher effect and the sentiment that is induced in the viewer ('viewer affect concept') and aims at

predicting the latter. The goals are to recommend appropriate images and suggest image comments.

## 7.3 The Proposed Method

Recent works have shown [144] that neural network based language models significantly outperform N-gram models; similarly, the use of neural networks to learn visual features and classify images has shown that they can achieve state-of-the-art results on several standard datasets and international competitions [97]. The proposed method builds on these advances.

We start by describing the well-known text based approach *Continuous Bag-Of-Words* (CBOW) model [145] that is the base of our scheme, then we present our model for polarity classification problem. Finally, we show a further extension of the model to incorporate visual information, based on a Denoising Autoencoder [197], that allows the same unsupervised capabilities on images as CBOW-based methods on text.

### 7.3.1 Textual information

Mikolov *et al.* [145] showed that in the CBOW model, words with similar meaning are mapped to similar positions in a vector space. Thus, distances may carry a meaning, allowing to formulate questions in the vector space using simple algebra (e.g. the result of  $\text{vector}(\text{'king'}) - \text{vector}(\text{'man'}) + \text{vector}(\text{'woman'})$  is near  $\text{vector}(\text{'queen'})$ ). Another property is the very fast training, that allows to exploit large-scale unsupervised corpora such as web sources (e.g . Wikipedia).

**Continuous Bag-Of-Words model.** In this framework, each word is mapped to a unique vector represented by a column in a word matrix  $W$  of  $Q$  length. Every column is indexed by a correspondent index from a dictionary  $V_T$ . Given a sequence of words  $w_1, w_2, \dots, w_K$ , CBOW model with hierarchical softmax aims at maximizing the average log probability of predicting the central word  $w_t$  given the context represented by its  $M$ -window of words, i.e. the  $M$  words before and after  $w_t$ :

$$\frac{1}{K} \sum_{t=M}^{K-M} \log p(w_t | w_{t-M}, \dots, w_{t-1}, w_{t+1}, \dots, w_{t+M}) \quad (7.1)$$

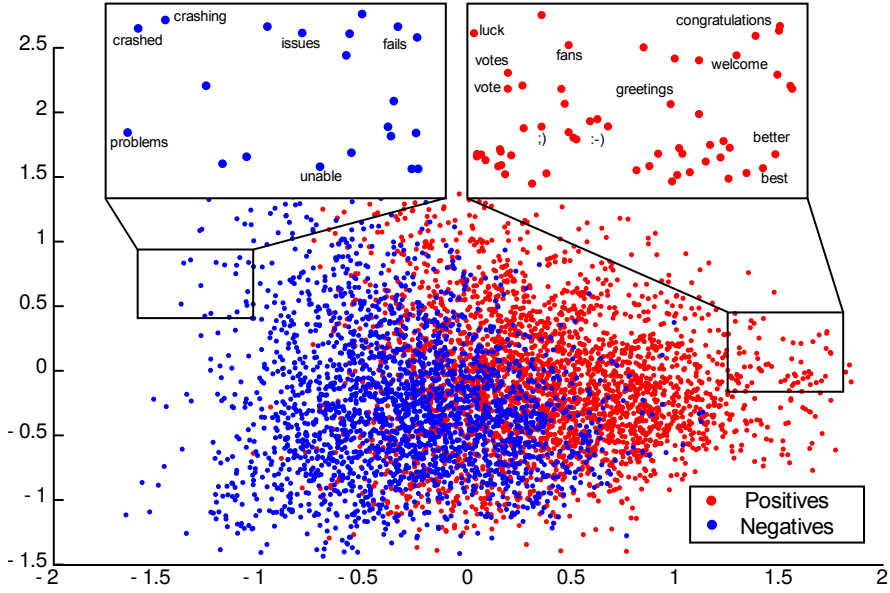


Figure 7.2: Visualization of CBOW word vectors trained on tweets of the SemEval-2013 dataset. Blue points are single words classified as negative, while red ones are positive. Semantically similar words are near (e.g. ‘crashing’ and ‘crashed’, ‘better’ and ‘best’) and share the same polarity.

The output  $f \in \mathbb{R}^{|V_T|}$  for the model is defined as:

$$f_{w_t} = [W_{t-M}, \dots, W_{t-1}, W_{t+1}, \dots, W_{t+M}]^T G \quad (7.2)$$

where  $W_i$  is the column of  $W$  corresponding to the word  $w_i$  and  $G \in \mathbb{R}^{P \times |V_T|}$ . Both  $W$  and  $G$  are considered as weights and have to be trained, resulting in a dual representation of words. Typically the columns of  $W$  are taken as final word features. An output probability is then obtained by using the softmax function on the output of the model:

$$p(w_t | w_{\text{context}}) = \frac{e^{f_{w_t}}}{\sum_i e^{f_{w_i}}} \quad (7.3)$$

where  $w_{\text{context}} = (w_{t-M}, \dots, w_{t-1}, w_{t+1}, \dots, w_{t+M})$ . When considering a high number of labels, it can be computed more efficiently by employing a hierarchical variation [146], requiring to evaluate  $\log_2(|V_T|)$  words instead of  $|V_T|$ .

In [145], an additional task named *Negative Sampling* is considered, where a word  $w_l$  is to be classified as related to the given context or not, i.e.  $p(w_l|w_{\text{context}})$ :

$$u_{w_l} = \sigma([W_{t-M}, \dots, W_l, \dots, W_{t+M}]^T N_s) \quad (7.4)$$

where  $N_s \in \mathbb{R}^Q$  and  $\sigma$  is the logistic function. Depending on  $w_l$  as the actual  $w_t$  word or a randomly sampled one,  $u_{w_l}$  has a target value of respectively 1 or 0.

**The CBOW-LR method.** Our model, denoted as CBOW-LR, is an extension of CBOW with negative sampling, specialized on the task of sentiment classification. An important difference from approaches that directly use a CBOW representation, or from [190], is that our model learns representation and classification concurrently. Considering that multi-task learning can improve neural networks performance [190], the idea is to use two different contributions accounting for semantic and sentiment polarity, respectively.

Given a corpus of tweets  $\mathbf{X}$  where each tweet is a sequence of words  $w_1, w_2, \dots, w_K$ , we aim at classifying tweets as positive or negative, and learn word vectors  $W \in \mathbb{R}^{Q \times |V_T|}$  with properties related to the sentiment carried by words, while retaining semantic representation. Semantic representation can be well-represented by a CBOW model, while sentiment polarity has limited presence or is lacking. Note that polarity supervision is limited and possibly weak, thus a robust semi-supervised setting is preferred: on the one hand, a model of sentiment polarity can use the limited supervision available, on the other hand the ability to exploit a large corpus of unsupervised text, like CBOW, can help the model to classify previously unseen text. This is explicitly accounted in our model by considering two different components:

*i)* inspired by [145], we consider a feature learning task on words by classifying sentiment polarity of a tweet. A tweet is represented as a set of  $M$ -window of words that we denote as  $\mathcal{G}$ . Each window  $\mathcal{G}$  is represented as a sum of their associated word vectors  $W_i$ , and a polarity classifier based on logistic regression is applied accordingly:

$$y(\mathcal{G}) = \sigma(C^T (\sum_{W_i \leftarrow w_i \in \mathcal{G}} W_i) + b_s) \quad (7.5)$$

Here the notation  $W_i \leftarrow w_i \in \mathcal{G}$  refers to selecting the  $i$ -th column of  $W$  by matching the  $w_i$  word from  $\mathcal{G}$ . The matrix  $C \in \mathbb{R}^Q$  and the vector

$b_s \in \mathbb{R}$  are parameters of a logistic regression, while a binary cross entropy is applied as loss function for every window  $\mathcal{G}$ . This is applied for every tweet  $T$  labeled with  $\bar{y}_T$  in the training set and results in the following cost:

$$C_{\text{sent}} = \sum_{(T, \bar{y}_T)} \sum_{\mathcal{G} \in T} -\bar{y}_T \log(y(\mathcal{G})) - (1 - \bar{y}_T) \log(1 - y(\mathcal{G})) \quad (7.6)$$

However, differently from a standard logistic regression, the representation matrix  $W$  is also a parameter to be learned. A labeled sentiment dataset is required to learn this task.

ii) we explicitly represent semantics by adding a task similar to negative sampling, without considering the hierarchical variation. The idea is that a CBOW model may also act as a regularizer and provide an additional semantic knowledge of word context. Given a window  $\mathcal{G}$ , a classifier has to predict if a word  $w_l$  fits in it. To this end, an additional cost is added:

$$C_{\text{sem}} = \sum_T \sum_{\mathcal{G} \in T} \sum_{(r_l, w_l) \in \mathcal{F}} (r_l - u_{w_l})^2 \quad (7.7)$$

where  $\mathcal{F}$  is a set of words  $w_l$  with their associated target  $r_l$ , derived from a training text sequence. This is the core of negative sampling:  $\mathcal{F}$  always contains the correct word  $w_t$  for the considered context  $\mathcal{G}$  ( $r_l = 1$ ) and  $K - 1$  random sampled words from  $V_T$  ( $r_l = 0$ ). It is indeed a sampling as  $K < |V_T| - 1$  of the remain wrong words. Note that differently from the previous task, this is unsupervised, not requiring labeled data; moreover tweets can belong to a different corpus than that used in the previous component. This allows to perform learning on additional unlabeled corpora, to enhance word knowledge beyond that of labeled training words.

Finally, concurrent learning is obtained by forging a total cost, defined by the sum of the two parts, opportunely weighted by a  $\lambda \in [0, 1]$ , and minimized with SGD:

$$C_{\text{CBOW-LR}} = \lambda \cdot C_{\text{sent}} + (1 - \lambda) \cdot C_{\text{sem}} \quad (7.8)$$

Fig. 7.2 visualizes the word vectors learned by our model. Note the tendency of separating the opposite polarities and the fact that similar words are close to each other.

At prediction time, for each word in a tweet  $T$  we consider its  $M$ -window  $\mathcal{G}$  and we compute (7.5) for each window, summing the results:

$$\text{Pred}(T) = \sum_{\mathcal{G} \in T} \left( y(\mathcal{G}) - 0.5 \right) \quad (7.9)$$

If  $Pred(T) < 0$  the tweet is labeled as negative, otherwise it is considered positive. It is worth noticing that at prediction time the method does not consider a word as positive or negative in its own, but it uses also its context to classify its sentiment and how strong it is. Thus the same word can be classified differently if used in different contexts.

### 7.3.2 Textual and Visual Information

The CBOW-LR model presented in Sect. 7.3.1 can be extended to account for visual information, such as that of images associated to tweets or status messages. Popular image representations are the Visual Bag-Of-Words Model [66, 102, 107], Fisher Vector [153] and its improved version [7, 154]. However, as shown recently in [24, 97], neural network based models have been shown to widely outperform these previous models. So, to fit with the CBOW representation discussed in the previous section, we choose to exploit the images by using a representation similar to the one used for the textual information, i.e. a representation obtained from the whole training set by means of a neural network. Moreover, likewise for the text, unsupervised learning can be performed. For these reasons, inspired also by works such as [197], we choose to extend our network with a single-layer Denoising Autoencoder, to take its middle level representation as our image descriptor. As for the textual version, the inclusion of this additional task allows our method to concurrently learn a textual representation and a classifier on text polarity and its associated image.

**Denoising Autoencoder.** In general, an Autoencoder (also called Autoassociator [14]) is a kind of neural network trained to encode the input into some representation (usually of lower dimension) so that the input can be reconstructed from that representation. For this type of network the output is thus the input itself. Specifically, an Autoencoder is a network that takes as input a  $K$ -dimensional vector  $x$  and maps it to a hidden representation  $h$  through the mapping:

$$h = \sigma(P_e x + b_e) \quad (7.10)$$

where  $\sigma$  is the sigmoid function (but any other non-linear activation function can be used),  $P_e$  and  $b_e$  are respectively a matrix of encoding weights and a vector of encoding biases. At this point,  $h$  is the coded representation of the input, and has to be mapped back to  $x$ . This second part is called the



reconstruction  $z$  of  $x$  (being  $z$  of the same dimension and domain of  $x$ ). In this step a similar transformation as in Eq. 7.10 is used:

$$z = \sigma(P_d h + b_d) \quad (7.11)$$

where  $P_d$  and  $b_d$  are respectively a matrix of decoding weights and a vector of decoding biases. One common choice is to constrain  $P_d = P_e^T$ ; in this configuration the Autoencoder is said to have ‘tied weights’. The motivation for this is that tied weights are used as a regularizer, to prevent the Autoencoder to learn the identity matrix when the dimension of the hidden layer is big enough to memorize the whole input; another important advantage is that the network has to learn fewer parameters. With this configuration, Eq. (7.11) becomes:

$$\hat{z} = \sigma(P_e^T h + b_d) \quad (7.12)$$

Learning is performed by minimizing the cross-entropy between the input  $x$  and the reconstructed input  $z$ :

$$L(x, z) = - \sum_{k=1}^K \left( x_k \log z_k + (1 - x_k) \log (1 - z_k) \right) \quad (7.13)$$

using stochastic gradient descent and backpropagation.

In this scenario  $h$  is similar to a lossy compression of  $x$ , that should capture the coordinates along the main directions of variation of  $x$ . To further improve the network, the input  $x$  can be ‘perturbed’ to another slightly different image,  $\tilde{x}$ , so that the network will not adapt too much to the given inputs but will be able to better generalize over new samples. This forms the Denoising variant of the Autoencoder. To do this, the input is corrupted by randomly setting some of the values to zero [14]. This way the Denoising Autoencoder will try to reconstruct the image including the missing parts. Another benefit of the stochastic corruption is that, when using a hidden layer bigger than the input layer, the network does not learn the identity function (which is the simplest mapping between the input and the output) but instead it learns a more useful mapping, since it is trying to also reconstruct the missing part of the image.

**The CBOW-DA-LR method.** The model used to deal with textual and visual information, denoted as CBOW-DA-LR, is an extension of CBOW-LR with the addition of a new task based on a Denoising Autoencoder (DA)

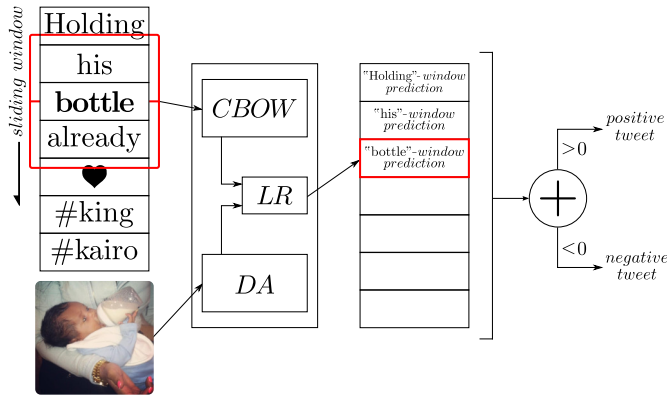


Figure 7.3: The process of polarity prediction of a tweet with its associated image performed by our model. On the left, one tweet text window (in red) at a time is fed into the CBOW model to get a textual representation. Likewise, the associated image is fed into the denoising autoencoder (DA). The two representations are concatenated and a polarity score for the window is obtained from the logistic regression (LR). Finally, each window polarity is summed into a final tweet polarity score.

applied to images, aiming at obtaining a mid-level representation. In this final form, the descriptor obtained from the DA, together with the continuous word representation, represents the new descriptor for a window of words in a tweet and is concurrently used to learn a logistic regressor. Given a tweet, for each window, we compute the continuous word representation and the image descriptor associated with the tweet. Each window in a tweet will be associated with the same image descriptor as the image for the tweet is always the same.

Fig. 7.3 shows an exemplification of the prediction process for a tweet with its accompanying image. While the image gets a fixed representation for the entire process, the text is represented one window at a time through a sliding window process. Each window is processed independently to get a local polarity score. To get the overall tweet polarity, each window polarity is summed into a final score and classified according to its sign.

This can be formalized as follows: if we define  $h_G$  as the encoding of the

image associated to the window  $\mathcal{G}$  of the tweet  $T$ , then Eq. (7.5) becomes:

$$y(\mathcal{G}) = \sigma \left( C^T \left( \left( \sum_{W_i \leftarrow w_i \in \mathcal{G}} W_i \right) \parallel (h_{\mathcal{G}}) \right) + b_s \right) \quad (7.14)$$

where  $\parallel$  is the concatenation operator, i.e. the encoded representation of the image is concatenated to the continuous word representation of the window, forming a new vector whose size is the sum of the size of the continuous word space and the size of the encoding representation of the image.

As stated before, the Autoencoder can be pre-trained in the same fashion as the continuous word representation. Any set of unlabeled images can be used to train the network before the actual training on the tweets.

The DA will be a component of our model and, like the two previous components CBOW and LR, it has its own cost function. Similar to Eq. (7.13), it is:

$$C_{\text{image}} = - \sum_{k=1}^K \left( \tilde{x}_k \log \hat{z}_k + (1 - \tilde{x}_k) \log (1 - \hat{z}_k) \right) \quad (7.15)$$

Since we are aiming at concurrent learning the textual and image representations, the three components are combined together in a single final cost of CBOW-DA-LR. Starting from the previously defined Eq. (7.8) for CBOW and Eq. (7.7) for LR, the cost becomes:

$$C_{\text{CBOW-DA-LR}} = \lambda_1 \cdot C_{\text{sent}} + \lambda_2 \cdot C_{\text{sem}} + \lambda_3 \cdot C_{\text{image}} \quad (7.16)$$

where  $\lambda_1, \lambda_2, \lambda_3$  weight the contribution of each task. The model can be trained by minimizing  $C_{\text{CBOW-DA-LR}}$  with stochastic gradient descend. Symbolic derivatives can be easily obtained by using an automatic differentiation algorithm (e.g. Theano [12]). After training, Eq. (7.9) can be used to predict the label of the tweet in the same manner as it is used when we do not consider the image descriptor.

## 7.4 Experiments

**The datasets.** To evaluate the proposed approach we have used four datasets obtained from Twitter:

*i)* Sanders Corpus<sup>4</sup>, consists of 5,513 manually labelled tweets on 4 topics (Apple, Google, Microsoft and Twitter). Of these, after removing missing

<sup>4</sup><http://sananalytics.com/lab/twitter-sentiment/>

tweets, retweets and duplicates, only 3,625 remain. The dataset does not specify a train and a test subset, so to evaluate the performance the whole set is randomly divided multiple times into subsets each time each one with the same size and the mean performance is reported;

*ii)* Sentiment140<sup>5</sup> [60] consists of a 1.6 million tweet training set collected and weakly annotated by querying positive and negative emoticons, considering a tweet positive if it contains a positive emoticon like “ :) ” and negative if, likewise, it contains a negative emoticon like “ :( ”; the dataset also comprises a manually annotated test set of 498 tweets obtained querying names of products, companies and people;

*iii)* SemEval-2013<sup>6</sup> provides a training set of 9,684 tweets of which only 8,208 are not missing at the time of writing and a test set of 3,813 tweets, selected querying a mixture of entities, products and events; the dataset is part of the SemEval-2013 challenge for sentiment analysis and also comprises of a development set of 1,654 (of which only 1,413 available at the time of writing) that can be used as an addendum to the training set or as a validation set;

*iv)* SentiBank Twitter Dataset<sup>7</sup>, consists of 470 positive and 133 negative tweets with images, related to 21 topics, annotated using Mechanical Turk; the dataset has been partitioned by the authors into 5 subsets, each of around 120 tweets with the respective images, to be used for a 5-fold cross-validation.

In this work we consider the binary positive/negative classification, thus we have removed neutral/objective tweets from the corpora when necessary. This approach follows that of [60] and [129], and is motivated by the difficulty to obtain training data for this class; it has to be noted that even human annotators tend to disagree whether a tweet has a negative/positive polarity or it is neutral [82]. Performance is reported in terms of Accuracy. The evaluation for SemEval is performed using  $F_1$ , since this is the metric originally used in this dataset.

For the Sanders dataset, as described earlier, there is no definition of an actual test set nor of a training set. For these reasons we choose to follow the experimental setup of [129], where experiments on Sanders dataset have been performed varying the number of training tweets between 32 to 768. For each test, first the number of training tweets is selected, then half of them are randomly chosen from all the positive tweets and the other half

<sup>5</sup><http://help.sentiment140.com/for-students>

<sup>6</sup><http://www.cs.york.ac.uk/semeval-2013/task2/>

<sup>7</sup><http://www.ee.columbia.edu/ln/dvmm/vso/download/sentibank.html>

are chosen from the negative ones. Finally, the remaining tweets are used as test set. Since there could be some variation from a random set to another, for each test 10 different runs are evaluated and the mean is taken as the result of the selected test. Results with this dataset are reported with the notation “Sanders@ $n$ ”, where  $n$  is the number of training tweets selected.

The evaluation of the SentiBank dataset has been performed preserving the structure given by the authors so that the results could be comparable. The dataset is divided into 5 subsets for 5-fold cross-validation. Each at a time a subset is considered as test set while the other 4 are considered as training set; 5 runs are performed and in the end the mean of the 5 results is computed and considered the resulting value given by the method for the dataset. Considering the high imbalance between positive and negative tweets of this dataset we report also the  $F_1$  score in addition to Accuracy.

We have evaluated the proposed method through a set of 5 experiments: in the first one we evaluate the performance of the proposed CBOW-LR text model comparing it against the standard CBOW model. Then we assess the performance of these models after pre-training them with large scale Twitter corpora. In a third experiment we compare the proposed approach against a baseline and two state-of-the-art methods. In the final experiment we compare the proposed CBOW-DA-LR text+image model against a state-of-the-art method on a publicly available dataset composed by tweets with images. In all these experiments we empirically fixed  $K = 5$  and  $Q = 100$ . In the last experiment we evaluate the effects of  $K$  and  $Q$  parameters w.r.t. the classification performance on all the datasets. Regarding  $\lambda$  in the first three experiments and  $\lambda_1, \lambda_2, \lambda_3$  in the last one, we tested several combinations and found a good setting by fixing  $\lambda = 0.5$  and  $\lambda_1 = \lambda_2 = \lambda_3 = 0.33$ , respectively. Also the image DA was implemented with ‘tied weights’ to reduce overfitting. Its dimensionality was tested in the range [200, 1000] and found it better performing by fixing it to 500. To perform the optimization using stochastic gradient descent, we employed Theano [12] to automatically compute the derivatives.

**Exp. 1: Comparison with baselines.** Tab. 7.1 compares our proposed method (CBOW-LR) with two baselines: RAND-LR and CBOW+SVM. The purpose is twofold: *i*) since we are learning features crafted for the specific task, we compare our method with randomly generated features. RAND-LR learns a logistic regression classifier on random word features

Dataset	(proposed)		
	CBOW-LR	RAND-LR	CBOW+SVM
Sentiment140	83.01	61.56	79.39
SemEval-2013 ( $F_1$ )	72.57	53.01	71.32
Sanders @ 32	62.55	58.38	59.89
Sanders @ 256	74.91	63.69	67.91
Sanders @ 768	82.69	65.53	73.03

Table 7.1: Comparison between our method and two baselines. Performance is reported in terms of accuracy except for SemEval-2013, where is used the  $F_1$  measure. Sanders@n indicates the number of training tweets used for the experiments on that dataset.

(i.e. we set  $\lambda = 1$  in eq. 7.8); *ii*) we verify the superiority of CBOW-LR learned features against a standard unsupervised CBOW representation. The CBOW+SVM baseline employs SVM with standard pre-trained CBOW representation on the specific dataset.

Performance figures show that the proposed method consistently outperforms both baselines, thus our method learns useful representations with some improvement over CBOW.

**Exp. 2: Exploiting CBOW training on large scale data.** Tab. 7.2 compares our proposed method with two baselines when exploiting large scale training data for the CBOW representation. We pre-trained a CBOW model using the 1.6 million tweets of Sentiment140 and used the learned features (termed  $\text{CBOW}_S$ ) with two standard learning algorithms.  $\text{CBOW}_S$ +LR employs the logistic regression while  $\text{CBOW}_S$ +SVM uses the SVM classifier. In contrast to the baselines, our model  $\text{CBOW}_S$ -LR employs the pre-trained  $\text{CBOW}_S$  features as initialization for the  $W$  matrix. Comparing Tab. 7.2 with Tab. 7.1 shows that  $\text{CBOW}_S$ +SVM baseline benefit from the use of pre-learned  $\text{CBOW}_S$ . This is visible especially on the Sanders dataset, as more rich representation is built. Note that when  $\text{CBOW}_S$ +SVM is applied to Sentiment140 dataset it corresponds to CBOW+SVM, since  $\text{CBOW}_S$  description is trained on Sentiment140; therefore the result is the same.

While both  $\text{CBOW}_S$ +SVM and  $\text{CBOW}_S$ +LR are unable to modify the

Dataset	(proposed)		
	$\text{CBOW}_S\text{-LR}$	$\text{CBOW}_S\text{+LR}$	$\text{CBOW}_S\text{+SVM}$
Sentiment140	83.84	76.32	79.39
Semeval-2013 ( $F_1$ )	72.23	73.73	71.48
Sanders @ 32	66.28	66.90	66.65
Sanders @ 256	76.33	71.14	73.69
Sanders @ 768	82.98	75.43	76.44

Table 7.2: Comparison between our method and two baselines, using an initialization based on CBOW pre-trained aside with 1.6 million tweets of Sentiment140. Performance is reported in terms of accuracy except for SemEval-2013, where is used the  $F_1$  measure. Sanders@n indicates the number of training tweets used for the experiments on that dataset.

word vector representation, our model  $\text{CBOW}_S\text{-LR}$  is able to retain the full richness of the initial representation and improve it on two datasets.

**Exp. 3: Comparison with FSLM and ESLAM.** In this experiment we have compared both textual variants of our approach, one with CBOW trained using the dataset on which the method is applied and one using  $\text{CBOW}_S$ , with two state-of-the-art methods: FSLM and ESLAM, proposed in [129]. FSLM uses a fully supervised probabilistic language model, learned concatenating all the tweets of the same class to form synthetic documents. ESLAM extends FSLM exploiting noisy tweets, based on the presence of ‘positive’ and ‘negative’ emoticons, to smooth the language model. Inclusion of manually labelled data with the unsupervised noisy data gives the power to deal with unforeseen text that is not easily handled by fully supervised methods. Fig. 7.4 shows the Accuracy while varying the number of training tweets of the Sanders dataset. The proposed approach has a much lower performance when using only 32 or 64 tweets for training. However, it can be observed that as the number of training data increases so does the performance of the proposed method, that outperforms that of ESLAM when using 768 tweets for training. In general the proposed method outperforms FSLM. The fact that ESLAM outperforms the proposed method when using smaller training data can be explained by the fact that CBOW

models, as Skip-Gram and feature learning methods, require large training datasets.

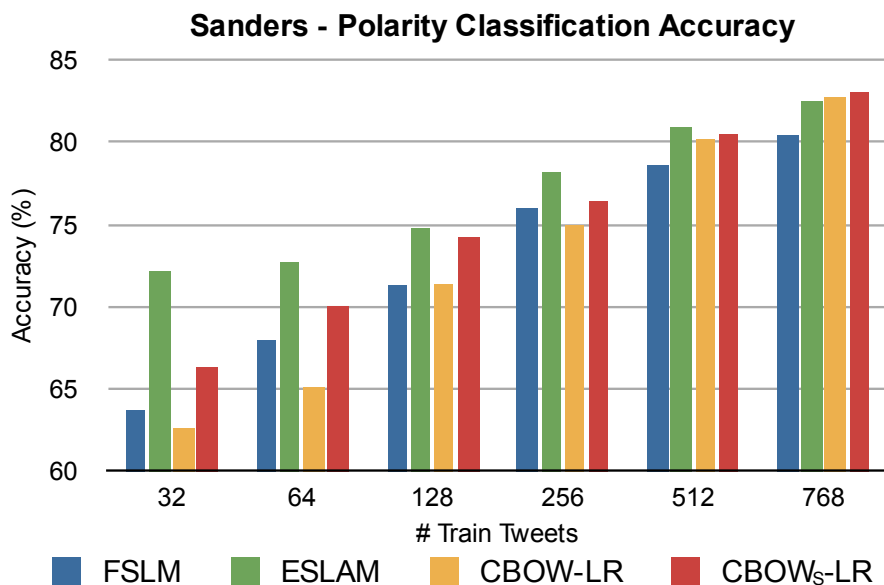


Figure 7.4: Comparison between our method with FSLM and ESLAM [129] on Sanders dataset, while varying the number of training tweets.

**Exp. 4: Exploiting textual and visual data.** In this experiment we have evaluated the performance of three versions of our proposed approach – CBOW-LR for text, DA-LR for visual data, and CBOW-DA-LR for both text and visual information – with different baselines and state-of-the-art approaches.

CBOW-LR has been compared with SentiStrength [185] and the CBOW+SVM baseline used in Exp. 1 and Exp. 2. DA-LR has been compared with SentiBank [19] classifiers. CBOW-DA-LR has been compared with the approach proposed by the authors of the SentiBank Twitter dataset [19], that uses SentiStrength [185] API<sup>8</sup> for text classification and SentiBank classifiers as mid-level visual features, with a logistic regression model. As the dataset is imbalanced, we also compare these approaches with an additional base-

<sup>8</sup><http://sentistrength.wlv.ac.uk/>



Data	Method	SentiBank (AC)	SentiBank ( $F_1$ )
	Random	47	42
Text	SentiStreight [185]	58	51
	CBOw+SVM	72	50
	<sup>(proposed)</sup> CBOw-LR	75	52
Image	SentiBank [19]	71	51
	<sup>(proposed)</sup> DA-LR	69	51
Text+Image	SentiStreight [185] + SentiBank [19]	72	n.a.
	<sup>(proposed)</sup> CBOw-DA-LR	<b>79</b>	<b>57</b>

Table 7.3: Comparison between our method (on single and combined modalities) with baselines and state-of-the-art approaches on SentiBank Twitter Dataset.

line based on random classification, i.e. we assign a random polarity to each test tweet. We used the code provided by the authors of the methods, except for the SentiStreight+SentiBank case, for which we report the result published in [19]. Results reported in Tab. 7.3 show that not only CBOw-LR outperforms both the baseline and SentiStreight, but also the multimodal SentiStreight+SentiBank approach. When using only visual information SentiBank obtains a better performance than DA-LR. Considering the text+image case it can be observed that the proposed multimodal CBOw-DA-LR method improves upon single modalities (CBOw-LR and DA-LR) and outperforms SentiStreight+SentiBank by a larger margin, proving that images hold meaningful informations regarding the polarity of text, and thus can be exploited to improve overall Accuracy and  $F_1$ .

**Exp. 5: Parameters analysis.** Fig. 7.5 shows accuracy and  $F_1$  of our model when varying  $K$  and  $Q$  parameters on Sanders, SemEval-2013 and Sentiment140 datasets. The performance on SentiBank is practically not affected by these parameters. The same set of parameters results in the best performance on all the datasets. The values of  $K$  and  $Q$  are in line with those obtained to train CBOw models on Wikipedia by Mikolov *et al.* .

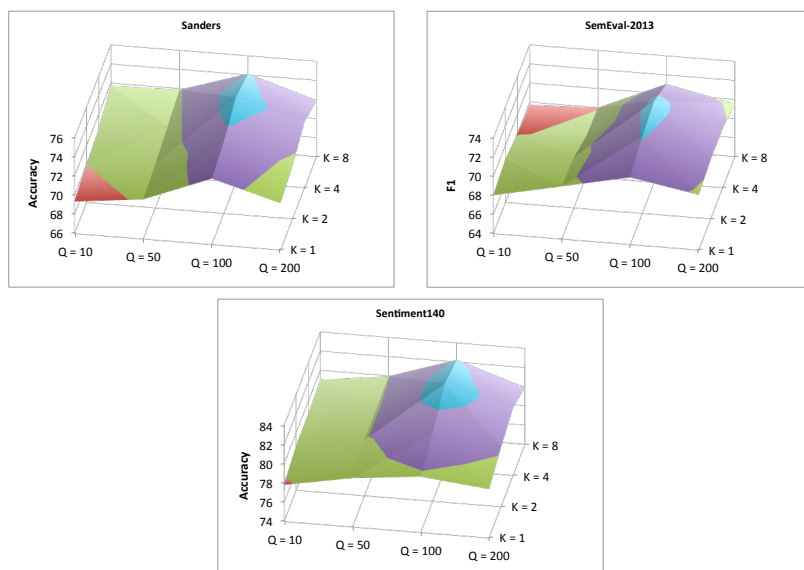


Figure 7.5: Performance of the proposed method when varying  $K$  and  $Q$  parameters on Sanders, SemEval-2013 and Sentiment140 datasets.

## 7.5 Conclusions

In this chapter we have presented a method for sentiment analysis of social network multimedia, presenting an unified model that considers both textual and visual information.

Regarding textual analysis we described a novel semi-supervised model CBOW-LR, extending the CBOW model, that learns concurrently vector representation and a sentiment polarity classifier on short texts such as that of tweets. Our experiments show that CBOW-LR can obtain improved accuracy on polarity classification over CBOW representation on the same quantity of text. When considering a large unsupervised corpus of tweets as additional training data for CBOW, a further improvement is shown, with our model being able to improve the overall accuracy. Comparison with the state-of-the-art methods FSLM and ESLAM shows promising results.

The CBOW-LR model has been expanded to account for visual information using a Denoising Autoencoder. The unified model (CBOW-DA-LR) works in an unsupervised and semi-supervised manner, learning text and

image representation, as well as the sentiment polarity classifier for tweets containing images. The unified CBOV-DA-LR model has been compared with SentiBank, a state-of-the-art approach on a publicly available Twitter dataset, obtaining a higher classification accuracy.



## Chapter 8

# Popularity Prediction with Sentiment and Context Features

*Images in social networks share different destinies: some are going to become popular while others are going to be completely unnoticed. In this chapter we propose to use visual sentiment features together with three novel context features to predict a concise popularity score of social images. Experiments on large scale datasets show the benefits of proposed features on the performance of image popularity prediction. Exploiting state-of-the-art sentiment features, we report a qualitative analysis of which sentiments seem to be related to good or poor popularity.<sup>1</sup>*

### 8.1 Introduction

In the last decade users of social networks such as Flickr and Facebook have uploaded tens of billions of photos, often adding accompanying metadata by tagging and by providing a short description. Users interact with each other by forming groups of shared interests, following the status streams of each other, and by commenting the photos that have been shared. Inevitably, in the huge quantity of available media, some of these images are going to become very popular, while others are going to be totally unnoticed and end

---

<sup>1</sup>This chapter appeared as “Image Popularity Prediction in Social Media Using Sentiment and Context Features” in *Proc. of ACM International Conference on Multimedia*, 2015, pp. 907-910.

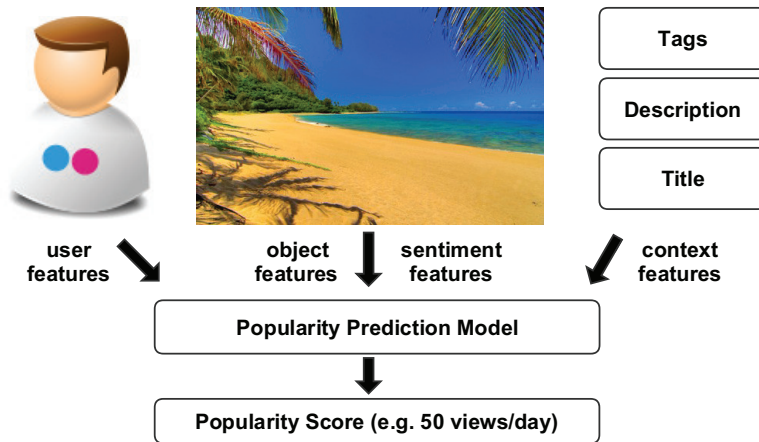


Figure 8.1: A schema of our approach to popularity prediction of images.

up in oblivion. Often, media may be popular because it conveys sentiments or it has a rich meaning in the social context it is put. In fact, sentiments have been known to affect popularity of visual media since the widespread watch of television programs [41]. Also, it was recently found to be related to popularity in tweets [6]. Being able to predict the popularity of a media may have a profound impact on several essential applications such as content retrieval and annotation, but also in other fields such as advertising and content distribution [51].

In this chapter, we address the problem of predicting the popularity of an image posted in a social network, considering different scenarios that are typical of different situations. Despite the recent crop of literature that studies the question of what makes an image popular [93, 138, 186], none of these works addresses the question of how much the visual sentiment is influencing the popularity of media. As social context has been widely found important to predict media popularity [93], we show how to further improve popularity estimation by using a knowledge base to supplement the understanding of semantics in textual metadata.

The main contributions of this chapter are:

- we propose to employ state-of-the-art visual sentiment features [19, 27] to perform image popularity prediction;
- we propose three new textual features based on a knowledge base, to

better model the semantic description of an image, in addition to the social context features proposed in [93, 138];

- we show qualitative results of which sentiments seem to be related to a good or poor popularity.

To the best of our knowledge, this is the first work understanding specific visual concepts that positively or negatively influence the eventual popularity of images, beyond just numerical prediction of photo popularity.

Experiments performed on large scale datasets illustrate several benefits of the two types of proposed features, and show how their combination impacts effectively on the performance of popularity prediction.

## 8.2 Related work

**Popularity Prediction** Recently, a significant effort has been spent on investigating popularity of social media content. Regarding image popularity, the majority of works agree that social features have the greatest predictive power [93, 138, 186]. Visual content features are less powerful than social ones in terms of predictive power, but they are useful when no user meta-data is present (e.g. no tags or description) or to address scenarios such as the case in which no social interactions have been recorded before posting the image (e.g. because the user has just joined the social network). Previous works vary in terms of popularity score definition (e.g. image views, reshares, mean views over a period) but they all share the same basic pipeline: they extract several content and context related features and successively employ a regressor to compute the popularity score.

In [93], Khosla *et al.* investigate both low-level features such as color, GIST, LBP, and content features such as the object predictions and network activations of a state-of-the-art CNN image classifier [97]. Together with user and image context features, they show promising results. McParlane *et al.* [138] propose to use image content, context features and user context to predict popularity. Their analysis is limited to a cold start scenario, i.e. where there exist no or little textual or interaction data. Totti *et al.* [186] investigate the use of aesthetics features such as blur, aspect ratio and color channel statistics together with the output of 85 object classifiers as content features.

**Visual Sentiment** A few works have addressed the problem of multimedia sentiment analysis of social network images. Starting from the 24 basic emotions of Plutchik’s Wheel of Emotions [155], Borth *et al.* [19] have recently presented a large-scale visual sentiment ontology termed SentiBank. They train 3,244 detectors on pairs of nouns and adjectives (ANPs) based on a combination of global and local features. Based on the recent breakthrough of convolutional networks for classification [97], Chen *et al.* [27] used a CNN to replace SVM in the approach of Borth *et al.* [19], obtaining an improved accuracy on ANPs.

The authors in [29] proposed an hierarchical system able to handle sentiment concept classification and localization on objects. They found individual concept detector of SentiBank [19] less reliable for object-based concepts.

Chen *et al.* [30] studied the correlation between the intended publisher sentiment and the actual induced in the viewer (‘viewer affect concept’). They aim to recommend appropriate images for the publisher by predicting in advance the induced sentiment in the viewer.

## 8.3 The Proposed Method

Our proposed method is based on two hypotheses: *i)* the popularity of an image can be fueled by the inherent visual sentiments conveyed; *ii)* semantic descriptions of an image is also important for its popularity, since it makes it easier to be found or looked at.

### 8.3.1 Measuring Popularity

It is difficult to precisely define a single score as measure of popularity, and several ways have been proposed to measure it. Khoshla *et al.* [93] used the number of views on Flickr as the principal metric. McParlane *et al.* [138] consider both the number of views and the number of comments for each image as they have been found correlated in video popularity [25]. However they only aim to predict two classes of popularity: high or low.

In this work we follow Khoshla *et al.* [93] and consider the number of views on Flickr as popularity metric. To cope with the large variation of views, we divide the popularity metric by the difference of time between the user upload and our retrieval, then we apply the log function.



### 8.3.2 Visual Sentiment Features

To discover which visual emotions are roused from the visualization of an image, a visual sentiment concept classification is performed based on the Visual Sentiment Ontology (VSO). The ontology, consisting in a collection of 3,244 Adjective-Noun-Pairs (ANPs), has been defined by Borth *et al.* [19]. In particular we used DeepSentiBank [27]: a convolutional neural network pre-trained from [97] has been fine-tuned to classify images in one of a subset of 2,096 ANPs. Similarly to its previous version [19], this tool provides a mid-level representation of an image.

For each image we extract two descriptors that we term respectively SentANPs and FeatANPs: the ANPs prediction layer of 2,096d and the rectified activations of the 7<sup>th</sup> fully connected layer of 4,096d.

### 8.3.3 Object Features

Since image popularity is related also to the visual content of the image, we extract the convolutional neural networks features, initially proposed in [93]. A very deep CNN with 16 layers [24] was used to extract for each image the final output containing 1,000 objects from ILSVRC 2014 challenge (termed ObjOut) and the 4,096d representation of the 7<sup>th</sup> rectified fully connected layer (termed ObjFC7).

### 8.3.4 Context Features

Image context information such as tags and description contains important cues that may reflect on the number of views that an image obtains. Entities like people, locations or tourist attractions can affect popularity as *i*) people may be more interested in photographs referring some particular subject; *ii*) the presence of tags and description, the submission of a photo to some groups, etc. make it easier to be found by other users. The extraction of entities from image context strongly depends on the nature of the text, i.e. tags and textual description; due to the different nature of these channels, two different approaches are proposed.

**Entity Extraction from Tags** Starting from image tags, we define two new context features that we term TagType and TagDomain. They both rely on Freebase, a large collaborative ontology containing millions of interconnected topics. Given a tag, a search for a *Freebase topic* is performed: if

the tag is related to some topics, the most popular one is picked, according to Freebase popularity ranking. Meaningless tags that do not have a match in Freebase topics are ignored, thus they do not act as a nuisance. When a Freebase topic is retrieved, another query is performed to extract its *Freebase types* with the “notable” property and its *Freebase domain*. While *types* are mostly specific (e.g. Person, Author) *domains* cover broader areas (e.g. Film, Music).

Due to the vast number of types in the ontology, a smaller specific type knowledge base is introduced. We first randomly sampled 10k tags from MIR-Flickr dataset vocabulary [75] and used them to extract Freebase types. We select the 100 most frequent types as our specific knowledge base.

The extraction of TagType feature for an image is then straightforward: each tag is used to query Freebase for a notable type. We count the matches to the 100 selected types and obtain a 100d histogram as final feature.

Regarding the TagDomain feature, we take the full list of 78 domains pre-defined by Freebase curators and count the tag matches, similarly as TagType. Thus, the eventual TagDomain feature result in a 78d histogram.

**Entity Extraction from description** Differently from the concise tags, image descriptions allow users to comprehensively detail their images in natural language. We seek to recognize subjects and objects of this text to detail context. Hence, we adopt a well known CRF-based language model to perform Named Entity Recognition (NER) [52]. We used the pre-trained 7-class model for MUC that is able to recognize Time, Location, Organization, Person, Money, Percent, Date. We count the occurrences for each class and build a 7d feature that we term NER<sub>7</sub>.

### 8.3.5 User Features

Previous works have found that the number of views that a photograph is going to obtain depends not only on the image itself and its context information, but also on the author data. In this work we used the same user features proposed by Khosla *et al.* [93]: among these features the most related one to popularity is the mean views of the images of the user, as it represents the popularity of the user himself.

### 8.3.6 Popularity prediction

In order to predict popularity as a concise score, we used an off-the-shelf Support Vector Machine. As we are working with large-scale dataset, we used a L2 regularized L2 loss Support Vector Regression (SVR) from LIBLINEAR package due to its scalability with large sparse data and huge number of instances compared to a kernelized version.

## 8.4 Experiments

As different scenarios show different aspects of popularity, we structure our experimental setups similarly to those of Khosla *et al.* [93], using Flickr social network. Two datasets were used to represent two different scenarios:

- *One-Per-User (OPU)*: we randomly selected 250k images from the VSO Flickr Dataset [19]. This dataset represents the scenario of a Flickr search, where images belong to different users.
- *User Specific (US)*: 25 users from the VSO Flickr Dataset are selected at random to constitute 25 different trials. For each one, 10k images are randomly selected. This dataset represent the scenario of a user that wants to select which of his pictures should be uploaded to attract the attention of other users.

In each experiment, we extract and concatenate the selected features. We freely provide the extracted features on our website. Multidimensional features are L2 normalized, while scalar attributes are scaled in the  $[0, 1]$  range. We split every dataset in training and evaluation: half was randomly chosen as training set, while the remaining images were equally split in validation and testing set. The  $C$  of SVM was set in the range  $[0.001 - 100]$ .

After the prediction, testing images are ranked in descending popularity scores and compared to the correct ranking obtained by the ground truth scores. The correlation between these two lists  $r$  and  $s$  is computed using *Spearman's rank correlation* that ranges in  $[-1, 1]$ :

$$\rho = \frac{\sum_i (r_i - \bar{r})(s_i - \bar{s})}{\sqrt{\sum_i (r_i - \bar{r})^2} \sqrt{\sum_i (s_i - \bar{s})^2}} \quad (8.1)$$

a score of 1 (or -1) corresponds to perfect (inverse) correlation, while 0 corresponds to random ranks.

### 8.4.1 Results

Experiments have been carried out for visual features, context ones and visual + context + user combination. We train a model with each single feature to show its predictive power. Then, we combine the features and compare a model with all of them against baselines implemented following the method of Khosla *et al.* [93] i.e. without our novel features. Results are reported in terms of *Spearman’s rank correlation* and, for the User Specific dataset, the average scores between the 25 users are reported.

**Visual Features** Visual content features include visual sentiment and object detections (Sec. 8.3.2, 8.3.3). The latter ones are used in this case as a baseline, including ObjOut and ObjFC7.

Dataset	SentANPs	FeatANPs	ObjOut	ObjFC7	Baseline	All
OPU	0.28	0.32	0.13	0.30	0.30	<b>0.36</b>
US	0.31	0.40	0.27	0.40	0.40	<b>0.43</b>

Table 8.1: Visual Features Results

Results are reported in Table 8.1: sentiment features are comparable with object features. As ANPs are learned starting from a similar network for classification, this suggests the existence of some correlation between them. Nevertheless, SentANPs is higher than ObjOut, suggesting that ANPs are better for popularity prediction than purely object classification. Our features are able to improve overall prediction in both scenarios.

**Context Features** The performance of the proposed context features (Sec. 8.3.4) is compared with a baseline composed by the number of tags, the length of title and description (Table 8.2).

Dataset	TagType	TagDomain	NER <sub>7</sub>	TagNum	TitleLen	DescLen	Baseline	All
OPU	0.42	0.36	0.50	0.55	0.22	0.48	0.61	<b>0.63</b>
US	0.44	0.37	0.13	0.23	0.17	0.20	0.33	<b>0.54</b>

Table 8.2: Context Features Results

Our features are comparable with other context-based ones in the OPU scenario. In the US scenario, all the features except TagType and TagDomain lose predictive power due to the limited context of a single user. This is because our features are able to better model semantically the single photos,

regardless of the single user. When combined, our feature boost correlation to 0.54 from 0.33 of the baseline.

**Visual + Context + User** In this experiment we combined visual, context and user features along with the total combination with and without our novel features. User features are added to resemble a state of the art pipeline. Each modality is singularly tested and finally combined together. Results are reported in Table 8.3. Note that User Features can't be used for the User Specific scenario as each model is trained for a single user.

Dataset	Method	Visual Content	Image Context	User Features	All
OPU	proposed	0.36	0.63	0.72	<b>0.76</b>
	baseline	0.30	0.61	0.72	0.74
US	proposed	0.43	0.54	n/a	<b>0.61</b>
	baseline	0.40	0.33	n/a	0.50

Table 8.3: Visual + Context + User Features Results

User Features produce the highest correlation in the OPU scenario, confirming that popularity is highly related to the popularity of the author [93]. Despite this, the combination of the three modalities is helpful, boosting correlation from 0.72 to 0.74. Our features further improve upon this, bringing the value to 0.76. In the User Specific dataset, the improvement from the baseline is more pronounced, where a correlation of 0.61 vs 0.50 is obtained.

### 8.4.2 Qualitative Analysis

We investigate which specific ANP and semantic metadata correlated the most with the number of views of images. This analysis is performed for the One-Per-User scenario, as it aims to be as generic as possible. Fig. 8.2(a) shows the trained SVR weights for each of the 2089 ANPs, in descending order. According to the figure we split the visual sentiments in three categories.

A first group include those ANPs that have a positive impact on image popularity (e.g. *sexy legs*, *beautiful eyes*, *heavy rain*). The rapid drop evinces that a very short number of ANPs corresponds to strongly popular images in the training dataset. Then, we observe that some visual sentiments obtain very low weights, near zero: that ANPs are almost irrelevant to the number of views (e.g. *sunny trees*, *dry forest*). Finally a third group includes ANPs

that are associated to a sufficiently negative score: the detection of those push an image towards unpopularity (e.g. *creepy eyes*, *silly clown*).

Extending our analysis to the 28 basic emotions of the Plutchick wheel, we found out that our model marked as unpopular those images that arouse emotions such as *annoyance* or *serenity*, while high scores are likely to be returned in case of sentiments as *amazement* or *ecstasy*. These last emotions derive from ANPs containing the adjective *sexy*, resulting in 10 occurrences in the top 35 visual emotions. A similar analysis on the 100 semantic entities is shown in Fig. 8.2(b). This plot has a similar trend compared with that of visual sentiment, but for the extreme values: in this case the negatively weighted types (e.g. *religious practice* and *software genre*) have more prominent values than the positively weighted ones (e.g. *garment* and *film character*).

## 8.5 Conclusions

In this chapter we proposed to employ state-of-the-art visual sentiment features and three new context features to address the problem of predicting whether an image posted on a social network may become popular. We are the first to show a qualitative analysis of which sentiments (as ANPs) are correlated to popularity. Our experiments suggest that some sentiments have a correlation with popularity, still smaller than user features. However, together with our novel context features, they have good prediction power, especially when user features are unavailable as in the User Specific scenario.

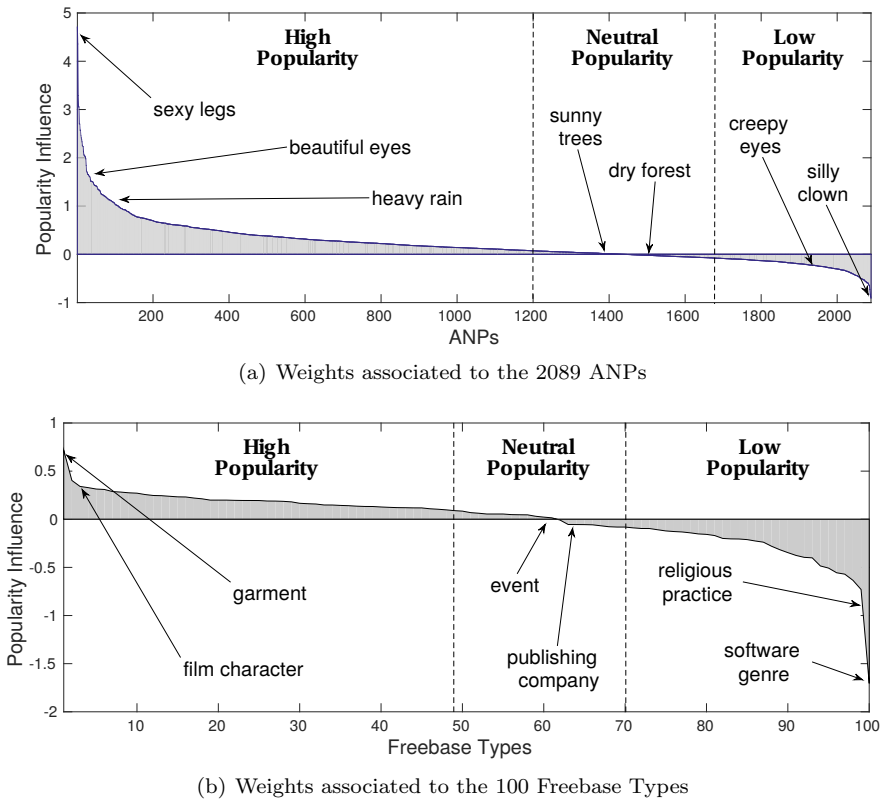


Figure 8.2: Influence of Multimedia Concepts on Popularity: weights of the 2089 ANP visual sentiment concepts (*top*); weights of the 100 Freebase Types extracted from contextual image tags (*bottom*).





# Chapter 9

## Conclusion

*This chapter summarizes the contribution of the thesis and discusses avenues for future research.*

### 9.1 Summary of Contribution

After presenting a structured survey of related work on social tagging and retrieval, we detailed a novel experimental protocol that we used to test and analyze eleven key methods. Established the state of the art, we proposed several models and methods to achieve objective annotation of images. Finally we moved to subjective annotation of sentiments aroused in a viewer and the expected popularity of an image.

In particular, we first presented in Chapter 2 a survey on image tag assignment, refinement and retrieval, with the hope of illustrating connections and difference between the many methods and their applicabilities, and consequently helping the interested audience to either pick up an existing method or devise a method of their own given the data at hand. Based on the key observation that all works rely on tag relevance learning as the common ingredient, exiting works, which vary in terms of their methodologies and target tasks, are interpreted in a unified framework. Consequently, a two-dimensional taxonomy has been developed, allowing us to structure the growing literature in light of what information a specific method exploits and how the information is leveraged in order to produce their tag relevance scores.

Having established the common ground between methods, a new experimental protocol was introduced in Chapter 3 for a head-to-head comparison between the state-of-the-art. A selected set of eleven representative works were implemented and evaluated for tag assignment, refinement, and/or retrieval.

Nearest neighbors methods proved to be the best overall performing method for assignment in Chapter 3. Hence, we proposed two novel techniques in Chapters 4 and 5 that reduce the semantic gap in these class of methods. In Chapter 4, we presented a cross-media model based on KCCA for tag assignment. The key idea was learning a semantic space, where visual and textual data were represented as blended unified features. This representation is able to provide better neighbors for nearest neighbor algorithms. The experimental results showed that our method makes consistent improvements over standard approaches based on a single-view visual representation as well as other previous work that also exploited tags. The properties of tested methods found in Chapter 3 remain still valid in the semantic space, although with an improved capability of retrieving better neighbors. Hence a better performance is obtained.

While a global representation is a requirement for nearest neighbors methods, local cues are important evidence for partially visible objects. Considering that users typically tag both local and global elements of a scene, in Chapter 5 we built a novel global representation that considers both types of features. Nearest neighbor methods are then used to perform the actual assignment or retrieval. Fisher vectors were adopted to produce new signatures that aggregate local descriptors but retain the global feature. The experiments proved the effectiveness of the new signatures compared to the baseline features.

Considering the influence of real world events in tagging behavior, in Chapter 6 we briefly analyzed the correlations between user tags, news and the objective relevance of concepts. The results suggest that analyzing the time series of tags may be beneficial to annotate social media.

Moving on to subjective information extraction, in Chapter 7 and 8 we explored the related tasks of sentiment analysis in tweets and the popularity estimation of images in social networks. In Chapter 7 we have presented a method for sentiment analysis of social network multimedia, capable of learning both textual and visual features in an unified fashion. Our model CBOW-LR, extending the CBOW model, learns concurrently a vector rep-

resentation and a sentiment polarity classifier on short texts. Comparing to previous work, our representation explicitly includes the sentiment of words and maintains good performance. By adding images to the mix, a further extension CBOV-DA-LR was presented. This semi-supervised model concurrently learns text and image representation, as well as the sentiment polarity classifier for tweets containing images. Experiments with large unsupervised corpus of tweets show promising results compared to the state-of-the-art.

Chapter 8 presented a novel approach to predict whether an image posted on a social network may become popular. The approach uses a combination of state-of-the-art visual sentiment features and three novel context features to reduce the semantic gap. The experiments reported suggest that some sentiments have a correlation with popularity. Moreover, our novel context features have good prediction power, especially when user features are unavailable. We also presented the first study that show a qualitative analysis of which sentiments (as ANPs) are correlated to popularity.

## 9.2 Direction of future work

Much remains to be done. Several exciting recent developments open up new opportunities for the future. First, extraction of objective information can profit from recent developments of deep learning. Employing novel deep learning based visual features is likely to boost the performance of annotations method that employ visual features. What is scientifically more interesting is to devise a learning strategy that is capable of jointly exploiting tag, image, and user information in a much more scalable manner than currently feasible. The importance of the filter component, which refines socially tagged training examples in advance to learning, is underestimated. Having a number of collaboratively labeled resources publicly available, research on joint exploration of social data and these resources is important. This connects to the most fundamental aspect of content-based image retrieval in the context of sharing and tagging within social media platforms: to what extent a social tag can be trusted remains open. Image retrieval by multi-tag query is another important yet largely unexplored problem. For a query of two tags, it is suggested to view the two tags as a single bi-gram tag [19, 116, 150], which is found to be superior to late fusion of individual tag scores. Nonetheless, due to the increasing sparseness of n-grams, how to effectively answer generic queries of more than two tag is challenging. Exploiting further modalities remain still a largely unexplored area of research.

In Chapter 6 we investigated the correlation of tags with the ground truth and events gathered from news by considering the time dimension. Although of limited scope, the study found that objective tags have a strong correlation to both content and context, giving a promising direction for improving content understanding. Possible extensions of this work include the exploration of how richer textual and semantic cues from natural language annotations might improve our models. Compared to extracting objective information, subjective information extraction is still young and full of exciting directions. We are still far from getting reliable estimations of sentiments in visual content. Current features are handcrafted on psychological or empirical studies but they are inherently affected by the semantic gap. Automatically learning features alike to approaches used in deep learning could bring considerable improvements in recognizing feelings despite the hard interpretability of filters. We barely scratched the surface in Chapter 7. Similarly, the prediction of popularity is still relying in basic handcrafted features. Although the social network aspects are well known to be related to popularity, visual content and context analysis is still needed when aiming to maximize popularity of a content. An underestimated factor is the peculiarity of different cultures in having different values and thus interest and feelings. Social networks can provide a world playground for study these aspects.

We see contributions of this field as essential to other related fields such as that of computer vision and artificial intelligence. The last two years were marked by a surge of deep convolutional models that showed remarkable improvement on vision tasks such as object recognition and image captioning. However, their limit is related to the strong supervision they need for training. Due to the cost of scaling these approaches, we expect an increased interest in unsupervised and semi-supervised learning, ultimately reaching social networks as an essential source of media.

“One way to resolve the semantic gap comes from sources outside the image ...”, Smeulders *et al.* wrote at the end of their seminal paper [177]. While what such sources would be was mostly unknown by that time, it is now becoming evident that the many images shared and tagged in social media platforms are promising to resolve the semantic gap. By adding new relevant tags, refining the existing ones or directly addressing retrieval, the access to the semantic of the content has been much improved. This is achieved only when appropriate care is taken to attack the unreliability of social tagging.

# Appendix A

## Publications

This research activity has led to several publications in international journals and conferences. These are summarized below.<sup>1</sup>

### International Journals

1. L. Ballan\*, M. Bertini\*, **T. Uricchio\***, A. Del Bimbo\*, “Data-driven approaches for social image and video tagging”. In *Multimedia Tools and Applications*, Feb 2015, Volume 74, Issue 4, pp. 1443-1468. DOI: 10.1007/s11042-014-1976-4 \*equal contribution.
2. C. Baecchi, **T. Uricchio**, M. Bertini, A. Del Bimbo, “A multimodal feature learning approach for sentiment analysis of social network multimedia”. In *Multimedia Tools and Applications*. DOI: 10.1007/s11042-015-2646-x (in press).

### Submitted

1. X. Li\*, **T. Uricchio\***, L. Ballan, M. Bertini, C.G.M. Snoek, A. Del Bimbo, “Socializing the Semantic Gap: A Comparative Survey on Image Tag Assignment, Refinement and Retrieval”. Submitted after major revision to ACM Computing Surveys and available as arXiv preprint arXiv:1503.08248. \*equal contribution.

---

<sup>1</sup>The author’s bibliometric indices are the following: *H*-index = 5, total number of citations = 43, *i10*-index = 1 (source: Google Scholar on November 28, 2015).

## International Conferences and Workshops

### Tutorials

1. X. Li\*, **T. Uricchio\***, L. Ballan, M. Bertini, C.G.M. Snoek, A. Del Bimbo, “Image Tag Assignment, Refinement and Retrieval”. In *Proc. of ACM Conference on Multimedia Conference (ACM MM)*, Brisbane, Australia, 2015. \*equal contribution.

### Conferences and Workshops

1. F. Gelli, **T. Uricchio**, M. Bertini, A. Del Bimbo, S-F. Chang, “Image Popularity Prediction in Social Media Using Sentiment and Context Features”, In *Proc. of ACM Conference on Multimedia Conference (ACM MM)*, Brisbane, Australia, 2015.
2. L. Ballan\*, **T. Uricchio\***, L. Seidenari, A. Del Bimbo, “A Cross-media Model for Automatic Image Annotation”. In *Proc. of ACM International Conference on Multimedia Retrieval (ICMR)*, Glasgow, United Kingdom, 2014, \*equal contribution.
3. **T. Uricchio**, L. Ballan, M. Bertini, and A. Del Bimbo, “Evaluating Temporal Information for Social Image Annotation and Retrieval”. In *Proc. of International Conference on Image Analysis and Processing (ICIAP)*, Napoli, Italy, 2013.
4. **T. Uricchio**, L. Ballan, M. Bertini, and A. Del Bimbo, “An Evaluation of Nearest-Neighbor Methods for Tag Refinement”. In *Proc. of IEEE International Conference on Multimedia & Expo (ICME)*, San Jose, CA, USA, 2013.
5. **T. Uricchio**, L. Ballan, M. Bertini, and A. Del Bimbo, “MICC-UNIFI at ImageCLEF 2013 Scalable Concept Image Annotation”. In *Proc. of Conference and Labs of the Evaluation Forum (CLEF)*, Valencia, Spain, 2013.
6. **T. Uricchio\***, M. Bertini\*, L. Seidenari, A. Del Bimbo, “Fisher Encoded Convolutional Bag-of-Boxes for Efficient Image Annotation and Retrieval”. *Proc. of International Conference on Computer Vision Workshops (ICCVW)*, Santiago, Chile, 2015, \*equal contribution.
7. L. Ballan, M. Bertini, **T. Uricchio**, and A. Del Bimbo, “Social Media Annotation”. In *Proc. of IEEE International Workshop on Content-Based Multimedia Indexing (CBMI)*, Veszprem, Hungary, 2013.

# Bibliography

- [1] O. Alonso, M. Gertz, and R. Baeza-Yates, “On the value of temporal information in information retrieval,” *SIGIR Forum*, vol. 41, no. 2, pp. 35–41, Dec. 2007.
- [2] S. Andrews, I. Tsochantaridis, and T. Hofmann, “Support vector machines for multiple-instance learning,” in *Proc. of NIPS*, 2003, pp. 561–568.
- [3] R. Arandjelovic and A. Zisserman, “All about VLAD,” in *Proc. of CVPR*, 2013.
- [4] P. Atrey, M. Hossain, A. El Saddik, and M. Kankanhalli, “Multimodal fusion for multimedia analysis: a survey,” *Multimedia Systems*, vol. 16, no. 6, pp. 345–379, 2010.
- [5] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky, “Neural codes for image retrieval,” in *Proc. of ECCV*, 2014.
- [6] Y. Bae and H. Lee, “Sentiment analysis of Twitter audiences: Measuring the positive or negative influence of popular twitterers,” *JASIST*, vol. 63, no. 12, pp. 2521–2535, 2012.
- [7] C. Baccchi, F. Turchini, L. Seidenari, A. D. Bagdanov, and A. D. Bimbo, “Fisher vectors over random density forests for object recognition,” in *Proc. of International Conference on Pattern Recognition (ICPR)*, 2014.
- [8] L. Ballan, T. Uricchio, L. Seidenari, and A. D. Bimbo, “A cross-media model for automatic image annotation,” in *Proc. of ACM ICMR*, 2014, pp. 73–80.
- [9] L. Ballan, M. Bertini, T. Uricchio, and A. Del Bimbo, “Data-driven approaches for social image and video tagging,” *Multimedia Tools and Applications*, vol. 74, no. 4, pp. 1443–1468, 2014.
- [10] L. Barbosa and J. Feng, “Robust sentiment detection on Twitter from biased and noisy data,” in *Proc. of International Conference on Computational Linguistics (COLING)*, 2010.
- [11] K. Barnard, P. Duygulu, D. Forsyth, N. De Freitas, D. M. Blei, and M. I. Jordan, “Matching words and pictures,” *JMLR*, vol. 3, pp. 1107–1135, 2003.

- [12] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. Goodfellow, A. Bergeron, N. Bouchard, D. Warde-Farley, and Y. Bengio, “Theano: new features and speed improvements,” *arXiv preprint arXiv:1211.5590*, 2012.
- [13] Y. Bengio, H. Schwenk, J.-S. Senécal, F. Morin, and J.-L. Gauvain, “Neural probabilistic language models,” in *Innovations in Machine Learning*. Springer, 2006, pp. 137–186.
- [14] Y. Bengio, “Learning deep architectures for AI,” *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, Jan. 2009.
- [15] J. Bian, Y. Yang, and T.-S. Chua, “Multimedia summarization for trending topics in microblogs,” in *Proc. of the ACM International Conference on Information & Knowledge Management (CIKM)*, 2013, pp. 1807–1812.
- [16] A. Bifet and E. Frank, “Sentiment knowledge discovery in Twitter streaming data,” in *Proc. of International Conference on Discovery Science (DS)*, 2010.
- [17] S. Bird, E. Loper, and E. Klein, *Natural Language Processing with Python*. O’Reilly Media Inc, 2009.
- [18] D. Blei, A. Ng, and M. Jordan, “Latent Dirichlet Allocation.” *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.
- [19] D. Borth, R. Ji, T. Chen, T. Breuel, and S.-F. Chang, “Large-scale visual sentiment ontology and detectors using adjective noun pairs,” in *Proc. of ACM International Conference on Multimedia (MM)*, 2013, pp. 223–232.
- [20] F. Bravo-Marquez, M. Mendoza, and B. Poblete, “Combining strengths, emotions and polarities for boosting Twitter sentiment analysis,” in *Proc. of ACM International Workshop on Issues of Sentiment Discovery and Opinion Mining (WISDOM)*, 2013.
- [21] E. Candès, X. Li, Y. Ma, and J. Wright, “Robust principal component analysis?” *Journal of the ACM*, vol. 58, no. 3, p. 11, 2011.
- [22] D. Cao, R. Ji, D. Lin, and S. Li, “A cross-media public sentiment analysis system for microblog,” *Multimedia Systems (MS)*, pp. 1–8, 2014.
- [23] G. Carneiro, A. B. Chan, P. J. Moreno, and N. Vasconcelos, “Supervised learning of semantic classes for image annotation and retrieval,” *IEEE TPAMI*, vol. 29, no. 3, pp. 394–410, 2007.
- [24] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, “Return of the devil in the details: Delving deep into convolutional nets,” in *Proc. of BMVC*, 2014.
- [25] G. Chatzopoulou, C. Sheng, and M. Faloutsos, “A first step towards understanding popularity in YouTube,” in *Proc. of INFOCOM*, 2010.



- [26] L. Chen, D. Xu, I. Tsang, and J. Luo, "Tag-based image retrieval improved by augmented features and group-based refinement," *IEEE Transactions on Multimedia*, vol. 14, no. 4, pp. 1057–1067, 2012.
- [27] T. Chen, D. Borth, T. Darrell, and S.-F. Chang, "DeepSentiBank: Visual sentiment concept classification with deep convolutional neural networks," *arXiv:1410.8586*, 2014.
- [28] T. Chen, D. Lu, M.-Y. Kan, and P. Cui, "Understanding and classifying image tweets," in *Proc. of ACM International Conference on Multimedia (MM)*, 2013, pp. 781–784.
- [29] T. Chen, F. X. Yu, J. Chen, Y. Cui, Y.-Y. Chen, and S.-F. Chang, "Object-based visual sentiment concept analysis and application," in *Proc. of ACM MM*, 2014.
- [30] Y.-Y. Chen, T. Chen, W. H. Hsu, H.-Y. M. Liao, and S.-F. Chang, "Predicting viewer affective comments based on image content in social media," in *Proc. of ACM International Conference on Multimedia Retrieval (ICMR)*, 2014, pp. 233:233–233:240.
- [31] H. Choi and H. Varian, "Predicting the present with Google Trends," Google, Tech. Rep., 2011.
- [32] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y.-T. Zheng, "Nus-wide: A real-world web image database from national university of singapore," in *Proc. of ACM Conf. on Image and Video Retrieval (CIVR'09)*, 2009.
- [33] R. Cilibrasi and P. Vitanyi, "The Google similarity distance," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 3, pp. 370–383, 2004.
- [34] J. Cohen, *Statistical power analysis for the behavioral sciences*. Routledge Academic, 1988.
- [35] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proc. of International Conference on Machine Learning (ICML)*, 2008.
- [36] E. Dan-Glauser and K. Scherer, "The geneva affective picture database (gaped): a new 730-picture database focusing on valence and normative significance," *Behavior Research Methods*, vol. 43, no. 2, pp. 468–477, 2011.
- [37] R. Datta, D. Joshi, J. Li, and J. Wang, "Image retrieval: ideas, influences, and trends of the new age," *ACM Computing Surveys*, vol. 40, no. 2, pp. 5:1–5:60, 2008.
- [38] W. Deitrick and W. Hu, "Mutually enhancing community detection and sentiment analysis on Twitter networks," *Journal of Data Analysis and Information Processing*, vol. 1, no. 3, p. 19.29, 2013.

- [39] J. Delhumeau, P.-H. Gosselin, H. Jegou, and P. Perez, “Revisiting the VLAD image representation,” in *Proc. of ACM MM*, 2013.
- [40] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *Proc. of CVPR*, 2009, pp. 248–255.
- [41] E. Diener and D. DeFour, “Does television violence enhance program popularity?” *JPSP*, vol. 36, no. 3, p. 333, 1978.
- [42] J. Dodge, A. Goyal, X. Han, A. Mensch, M. Mitchell, K. Stratos, K. Yamaguchi, Y. Choi, H. Daumé, III, A. Berg, and T. Berg, “Detecting visual text,” in *Proc. of NAACL*, 2012, pp. 762–772.
- [43] K. Duan, D. J. Crandall, and D. Batra, “Multimodal learning in loosely-organized web images,” in *Proc. of CVPR*, 2014, pp. 2465–2472.
- [44] L. Duan, W. Li, I. Tsang, and D. Xu, “Improving web image search by bag-based reranking,” *IEEE Transactions on Image Processing*, vol. 20, no. 11, pp. 3280–3290, 2011.
- [45] P. Duygulu, K. Barnard, J. F. G. de Freitas, and D. A. Forsyth, “Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary,” in *Proc. of ECCV*, 2002.
- [46] M. Everingham, S. Eslami, L. V. Gool, C. Williams, J. Winn, and A. Zisserman, “The PASCAL visual object classes challenge - a retrospective,” *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, 2015.
- [47] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, “LIBLINEAR: A library for large linear classification,” *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.
- [48] S. Feng, C. Lang, and B. Li, “Towards relevance and saliency ranking of image tags,” in *Proc. of ACM MM*, 2012, pp. 917–920.
- [49] S. L. Feng, R. Manmatha, and V. Lavrenko, “Multiple bernoulli relevance models for image and video annotation,” in *Proc. of CVPR*, 2004.
- [50] Z. Feng, S. Feng, R. Jin, and A. Jain, “Image tag completion by noisy matrix recovery,” in *Proc. of ECCV*, 2014, pp. 424–438.
- [51] F. Figueiredo, J. M. Almeida, M. A. Gonçalves, and F. Benevenuto, “On the dynamics of social media popularity: A YouTube case study,” *TOIT*, vol. 14, no. 4, p. 24, 2014.
- [52] J. R. Finkel, T. Grenager, and C. Manning, “Incorporating non-local information into information extraction systems by Gibbs sampling,” in *Proc. of ACL*, 2005.

- [53] Y. Freund, R. Iyer, R. Schapire, and Y. Singer, “An efficient boosting algorithm for combining preferences,” *Journal of Machine Learning Research*, vol. 4, pp. 933–969, 2003.
- [54] H. Fu, Q. Zhang, and G. Qiu, “Random forest for image annotation,” in *Proc. of ECCV*, 2012.
- [55] Y. Gao, M. Wang, Z.-J. Zha, J. Shen, X. Li, and X. Wu, “Visual-textual joint relevance learning for tag-based social image search,” *IEEE Transactions on Image Processing*, vol. 22, no. 1, pp. 363–376, 2013.
- [56] M. Ghiassi, J. Skinner, and D. Zimbra, “Twitter brand sentiment analysis: A hybrid system using n-gram analysis and dynamic artificial neural network,” *Expert Systems with Applications*, vol. 40, no. 16, pp. 6266–6282, 2013.
- [57] J. Ginsberg, M. H. Mohebbi, R. S. Patel, L. Brammer, M. S. Smolinski, and L. Brilliant, “Detecting influenza epidemics using search engine query data,” *Nature*, vol. 457, no. 7232, pp. 1012–1014, 02 2009.
- [58] A. Ginsca, A. Popescu, B. Ionescu, A. Armagan, and I. Kanellos, “Toward an estimation of user tagging credibility for social image retrieval,” in *Proc. of ACM MM*, 2014, pp. 1021–1024.
- [59] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proc. of CVPR*, 2014.
- [60] A. Go, R. Bhayani, and L. Huang, “Twitter sentiment classification using distant supervision,” CS224N Project Report, Stanford, Tech. Rep., 2009.
- [61] S. Golder and B. Huberman, “Usage patterns of collaborative tagging systems,” *Journal of Information Science*, vol. 32, no. 2, pp. 198–208, 2006.
- [62] G. Golub and C. Van Loan, *Matrix computations*. JHU Press, 2012, vol. 3.
- [63] Y. Gong, Q. Ke, M. Isard, and S. Lazebnik, “A multi-view embedding space for internet images, tags, and their semantics,” *IJCV*, vol. in press, 2013.
- [64] Y. Gong, L. Wang, R. Guo, and S. Lazebnik, “Multi-scale orderless pooling of deep convolutional activation features,” in *Proc. of ECCV*, 2014.
- [65] D. Grangier and S. Bengio, “A discriminative kernel-based approach to rank images from text queries,” *IEEE TPAMI*, vol. 30, no. 8, pp. 1371–1384, 2008.
- [66] K. Grauman and T. Darrell, “The pyramid match kernel: Discriminative classification with sets of image features,” in *Proc. of International Conference on Computer Vision (ICCV)*, 2005.
- [67] M. Grubinger, P. Clough, H. Muller, and T. Deselaers, “The IAPR TC-12 benchmark: a new evaluation resource for visual information systems,” in *Proc. of LRECW*, 2006.

- [68] M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid, “TagProp: Discriminative metric learning in nearest neighbor models for image auto-annotation,” in *Proc. of ICCV*, 2009, pp. 309–316.
- [69] M. Gupta, R. Li, Z. Yin, and J. Han, “Survey on social tagging techniques,” *SIGKDD Explorations Newsletter*, vol. 12, no. 1, pp. 58–72, 2010.
- [70] M. U. Gutmann and A. Hyvärinen, “Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics,” *The Journal of Machine Learning Research (JMLR)*, vol. 13, no. 1, pp. 307–361, 2012.
- [71] D. R. Hardoon and J. Shawe-Taylor, “KCCA for different level precision in content-based image retrieval,” in *Proc. of IEEE CBMI*, 2003.
- [72] D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor, “Canonical correlation analysis: An overview with application to learning methods,” *Neural Computation*, vol. 16, no. 12, pp. 2639–2664, 2004.
- [73] J. Hosang, R. Benenson, and B. Schiele, “How good are detection proposals, really?” in *Proc. of BMVC*, 2014.
- [74] M. J. Huiskes and M. S. Lew, “The MIR Flickr retrieval evaluation,” in *Proc. of ACM MIR*, 2008.
- [75] M. J. Huiskes, B. Thomee, and M. S. Lew, “New trends and ideas in visual concept detection: the MIR Flickr retrieval evaluation initiative,” in *Proc. of ACM MIR*, 2010, pp. 527–536.
- [76] S. J. Hwang and K. Grauman, “Learning the relative importance of objects from tagged images for retrieval and cross-modal search,” *IJCV*, vol. 100, no. 2, pp. 134–153, 2012.
- [77] F. Jabeen, S. Khusro, A. Majid, and A. Rauf, “Semantics discovery in social tagging systems: A review,” *Multimedia Tools and Applications*, vol. In press, 2015.
- [78] K. Järvelin and J. Kekäläinen, “Cumulated gain-based evaluation of IR techniques,” *ACM Transactions on Intelligent Systems and Technology*, vol. 20, no. 4, pp. 422–446, 2002.
- [79] H. Jégou, M. Douze, and C. Schmid, “Product quantization for nearest neighbor search,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 1, pp. 117–128, 2011.
- [80] H. Jegou, F. Perronnin, M. Douze, J. Sanchez, P. Perez, and C. Schmid, “Aggregating local descriptors into compact codes,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 9, pp. 1704–1716, 2012.

- [81] H. Jégou, M. Douze, and C. Schmid, “Hamming embedding and weak geometric consistency for large scale image search,” in *Proc. of ECCV*, 2008. [Online]. Available: <http://lear.inrialpes.fr/pubs/2008/JDS08>
- [82] L. Jiang, M. Yu, M. Zhou, X. Liu, and T. Zhao, “Target-dependent Twitter sentiment classification,” in *Proc. of ACL Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (HLT)*, 2011.
- [83] Y.-G. Jiang, C.-W. Ngo, and S.-F. Chang, “Semantic context transfer across heterogeneous sources for domain adaptive video search,” in *Proc. of ACM MM*, 2009, pp. 155–164.
- [84] X. Jin, A. Gallagher, L. Cao, J. Luo, and J. Han, “The wisdom of social multimedia: using Flickr for prediction and forecast,” in *Proc. of ACM MM*, 2010, pp. 1235–1244.
- [85] Y. Jin, L. Khan, L. Wang, and M. Awad, “Image annotations by combining multiple evidence & Wordnet,” in *Proc. of ACM MM*, 2005, pp. 706–715.
- [86] T. Joachims, “Transductive inference for text classification using support vector machines,” in *Proc. of ICML*, 1999, pp. 200–209.
- [87] J. Johnson, L. Ballan, and L. Fei-Fei, “Love thy neighbors: Image annotation by exploiting image metadata,” in *Proc. of ICCV*, 2015.
- [88] D. Joshi, R. Datta, E. Fedorovskaya, Q.-T. Luong, J. Wang, J. Li, and J. Luo, “Aesthetics and emotions in images,” *IEEE Signal Processing Magazine (MSP)*, vol. 28, no. 5, pp. 94–115, Sept 2011.
- [89] M. M. Kalayeh, H. Idrees, and M. Shah, “NMF-KNN: Image annotation using weighted multi-view non-negative matrix factorization,” in *Proc. of CVPR*, 2014, pp. 184–191.
- [90] T. Kaneko, H. Harada, and K. Yanai, “Twitter visual event mining system,” in *Proc. of IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, July 2013, pp. 1–2.
- [91] L. S. Kennedy, S.-F. Chang, and I. Kozintsev, “To search or to label? Predicting the performance of search-based automatic image classifiers,” in *Proc. of ACM MIR*, 2006, pp. 249–258.
- [92] L. S. Kennedy, M. Slaney, and K. Weinberger, “Reliable tags using image similarity: mining specificity and expertise from large-scale multimedia databases,” in *Proc. of ACM MM Workshop on Web-scale Multimedia Corpus*, 2009, pp. 17–24.
- [93] A. Khosla, A. Das Sarma, and R. Hamid, “What makes an image popular?” in *Proc. of WWW*, 2014.
- [94] G. Kim and E. Xing, “Time-sensitive web image ranking and retrieval via dynamic multi-task regression,” in *Proc. of ACM WSDM*, 2013, pp. 163–172.

- [95] G. Kim, L. Fei-Fei, and E. P. Xing, “Web image prediction using multivariate point processes,” in *Proc. of ACM SIGKDD*, 2012, pp. 1068–1076.
- [96] G. Kim, E. P. Xing, and A. Torralba, “Modeling and analysis of dynamic behaviors of web image collections,” in *Proc. of ECCV*, 2010, pp. 85–98.
- [97] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Proc. of Neural Information Processing Systems (NIPS)*, 2012, pp. 1097–1105.
- [98] Y.-S. Kuo, W.-H. Cheng, H.-T. Lin, and W. Hsu, “Unsupervised semantic feature discovery for image object retrieval and tag refinement,” *IEEE Transactions on Multimedia*, vol. 14, no. 4, pp. 1079–1090, 2012.
- [99] T. Lan and G. Mori, “A max-margin riffled independence model for image tag ranking,” in *Proc. of CVPR*, 2013, pp. 3103–3110.
- [100] P. J. Lang, M. M. Bradley, and B. N. Cuthbert, “International affective picture system (iaps): Technical manual and affective ratings,” 1999.
- [101] V. Lavrenko, R. Manmatha, and J. Jeon, “A model for learning the semantics of pictures,” in *Proc. of NIPS*, 2003.
- [102] S. Lazebnik, C. Schmid, and J. Ponce, “Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories,” in *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.
- [103] Q. V. Le and T. Mikolov, “Distributed representations of sentences and documents,” in *Proc. of International Conference on Machine Learning (ICML)*, 2014.
- [104] S. Lee, W. De Neve, and Y. Ro, “Visually weighted neighbor voting for image tag relevance learning,” *Multimedia Tools and Applications*, vol. 72, no. 2, pp. 1363–1386, 2013.
- [105] L.-J. Li and L. Fei-Fei, “OPTIMOL: Automatic online picture collection via incremental model learning,” *IJCV*, vol. 88, no. 2, pp. 147–168, 2010.
- [106] M. Li, “Texture moment for content-based image retrieval,” in *Proc. of ICME*, 2007, pp. 508–511.
- [107] T. Li, T. Mei, I.-S. Kweon, and X.-S. Hua, “Contextual bag-of-words for visual categorization,” *IEEE Transaction on Circuits and Systems for Video Technology (TCSVT)*, vol. 21, no. 4, pp. 381–392, 2011.
- [108] W. Li, L. Duan, D. Xu, and I. Tsang, “Text-based image retrieval using progressive multi-instance learning,” in *Proc. of ICCV*, 2011, pp. 2049–2055.
- [109] X. Li, “Tag relevance fusion for social image retrieval,” *Multimedia Systems*, vol. In press, 2015.

- [110] X. Li, E. Gavves, C. Snoek, M. Worring, and A. Smeulders, “Personalizing automated image annotation using cross-entropy,” in *Proc. of ACM MM*, 2011, pp. 233–242.
- [111] X. Li and C. Snoek, “Classifying tag relevance with relevant positive and negative examples,” in *Proc. of ACM MM*, 2013, pp. 485–488.
- [112] X. Li, C. Snoek, and M. Worring, “Annotating images by harnessing world-wide user-tagged photos,” in *Proc. of ICASSP*, 2009, pp. 3717–3720.
- [113] —, “Learning social tag relevance by neighbor voting,” *IEEE Transactions on Multimedia*, vol. 11, no. 7, pp. 1310–1322, 2009.
- [114] —, “Unsupervised multi-feature tag relevance learning for social image retrieval,” in *Proc. of ACM CIVR*, 2010, pp. 10–17.
- [115] X. Li, C. Snoek, M. Worring, D. Koelma, and A. Smeulders, “Bootstrapping visual categorization with relevant negatives,” *IEEE Transactions on Multimedia*, vol. 15, no. 4, pp. 933–945, 2013.
- [116] X. Li, C. Snoek, M. Worring, and A. Smeulders, “Harvesting social images for bi-concept search,” *IEEE Transactions on Multimedia*, vol. 14, no. 4, pp. 1091–1104, 2012.
- [117] X. Li, T. Uricchio, L. Ballan, M. Bertini, C. G. Snoek, and A. Del Bimbo, “Socializing the semantic gap: A comparative survey on image tag assignment, refinement and retrieval,” *arXiv preprint arXiv:1503.08248*, 2015.
- [118] Z. Li, J. Liu, and H. Lu, “Nonlinear matrix factorization with unified embedding for social tag relevance learning,” *Neurocomputing*, vol. 105, pp. 38–44, 2013.
- [119] Z. Li, J. Liu, X. Zhu, T. Liu, and H. Lu, “Image annotation using multi-correlation probabilistic matrix factorization,” in *Proc. of ACM MM*, 2010, pp. 1187–119.
- [120] H.-T. Lin, C.-J. Lin, and R. Weng, “A note on Platt’s probabilistic outputs for support vector machines,” *Machine Learning*, vol. 68, no. 3, pp. 267–276, 2007.
- [121] Z. Lin, G. Ding, M. Hu, J. Wang, and X. Ye, “Image tag completion via image-specific and tag-specific linear sparse reconstructions,” in *Proc. of CVPR*, 2013, pp. 1618–1625.
- [122] D. Liu, X.-S. Hua, M. Wang, and H.-J. Zhang, “Image retagging,” in *Proc. of ACM MM*, 2010, pp. 491–500.
- [123] D. Liu, X.-S. Hua, L. Yang, M. Wang, and H.-J. Zhang, “Tag ranking,” in *Proc. of WWW*, 2009, pp. 351–360.

- [124] D. Liu, X.-S. Hua, and H.-J. Zhang, "Content-based tag processing for internet social images," *Multimedia Tools and Applications*, vol. 51, no. 2, pp. 723–738, 2011.
- [125] D. Liu, S. Yan, X.-S. Hua, and H.-J. Zhang, "Image retagging using collaborative tag propagation," *IEEE Transactions on Multimedia*, vol. 13, no. 4, pp. 702–712, 2011.
- [126] J. Liu, M. Li, Q. Liu, H. Lu, and S. Ma, "Image annotation via graph learning," *Pattern Recognition*, vol. 42, no. 2, pp. 218–228, 2009.
- [127] J. Liu, Z. Li, J. Tang, Y. Jiang, and H. Lu, "Personalized geo-specific tag recommendation for photos on social websites," *IEEE Transactions on Multimedia*, vol. 16, no. 3, pp. 588–600, 2014.
- [128] J. Liu, Y. Zhang, Z. Li, and H. Lu, "Correlation consistency constrained probabilistic matrix factorization for social tag refinement," *Neurocomputing*, vol. 119, no. 7, pp. 3–9, 2013.
- [129] K.-L. Liu, W.-J. Li, and M. Guo, "Emoticon smoothed language models for Twitter sentiment analysis." in *Proc. of AAAI Conference on Artificial Intelligence (CAI)*, 2012.
- [130] Y. Liu, F. Wu, Y. Zhang, J. Shao, and Y. Zhuang, "Tag clustering and refinement on semantic unity graph," in *Proc. of ICDM*, 2011, pp. 417–426.
- [131] H. Ma, J. Zhu, M.-T. Lyu, and I. King, "Bridging the semantic gap between image contents and tags," *IEEE Transactions on Multimedia*, vol. 12, no. 5, pp. 462–473, 2010.
- [132] S. Maji, A. Berg, and J. Malik, "Classification using intersection kernel support vector machines is efficient," in *Proc. of CVPR*, 2008, pp. 1–8.
- [133] A. Makadia, V. Pavlovic, and S. Kumar, "A new baseline for image annotation," in *Proc. of ECCV*, 2008.
- [134] —, "Baselines for image annotation," *International Journal of Computer Vision*, vol. 90, no. 1, pp. 88–105, 2010.
- [135] J. McAuley and J. Leskovec, "Image labeling on a network: using social-network metadata for image classification," in *Proc. of ECCV*, 2012, pp. 828–841.
- [136] P. McParlane, S. Whiting, and J. Jose, "Improving automatic image tagging using temporal tag co-occurrence," in *Proc. of MMM*, 2013, pp. 251–262.
- [137] —, "On contextual photo tag recommendation," in *Proc. of ACM SIGIR*, 2013, pp. 965–968.
- [138] P. J. McParlane, Y. Moshfeghi, and J. M. Jose, "Nobody comes here anymore, it's too crowded; predicting image popularity on Flickr," in *Proc. of ACM ICMR*, 2014.



- [139] P. J. McParlane and J. Jose, “Exploiting Twitter and Wikipedia for the annotation of event images,” in *Proc. of ACM SIGIR Interantional Conference on Research & Development in Information Retrieval*, 2014, pp. 1175–1178.
- [140] T. Mei, Y. Rui, S. Li, and Q. Tian, “Multimedia search reranking: A literature survey,” *ACM Computing Surveys*, vol. 46, no. 3, p. 38, 2014.
- [141] D. Metzler and R. Manmatha, “An inference network approach to image retrieval,” in *Proc. of ACM CIVR*, 2004.
- [142] R. Michalski, “A theory and methodology of inductive learning,” *Artificial intelligence*, vol. 20, no. 2, pp. 111–161, 1983.
- [143] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *Proc. of Neural Information Processing Systems (NIPS)*, 2013.
- [144] T. Mikolov, A. Deoras, S. Kombrink, L. Burget, and J. H. Cernocky, “Empirical evaluation and combination of advanced language modeling techniques,” in *Proc. of Interspeech*, August 2011.
- [145] T. Mikolov, W.-t. Yih, and G. Zweig, “Linguistic regularities in continuous space word representations,” in *Proc. of NAACL-HLT*, 2013, pp. 746–751.
- [146] A. Mnih and G. E. Hinton, “A scalable hierarchical distributed language model,” in *Proc. of Neural Information Processing Systems (NIPS)*, 2009.
- [147] F. Monay and D. Gatica-Perez, “PLSA-based image auto-annotation: Constraining the latent space,” in *Proc. of ACM Multimedia*, 2004.
- [148] G. E. Moore, “Cramming more components onto integrated circuits,” vol. 38, 1965.
- [149] J. Y.-H. Ng, F. Yang, and L. S. Davis, “Exploiting local features from deep networks for image retrieval,” *arXiv preprint arXiv:1504.05133*, 2015.
- [150] L. Nie, S. Yan, M. Wang, R. Hong, and T.-S. Chua, “Harvesting visual concepts for image search with complex queries,” in *Proc. of ACM MM*, 2012, pp. 59–68.
- [151] Z. Niu, G. Hua, X. Gao, and Q. Tian, “Semi-supervised relational topic model for weakly annotated image recognition in social media,” in *Proc. of CVPR*, 2014, pp. 4233–4240.
- [152] J. C. Pereira, E. Coviello, G. Doyle, N. Rasiwasia, G. Lanckriet, R. Levy, and N. Vasconcelos, “On the role of correlation and abstraction in cross-modal multimedia retrieval,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 3, pp. 521–535, 2014.
- [153] F. Perronnin, Y. Liu, J. Sánchez, and H. Poirier, “Large-scale image retrieval with compressed fisher vectors,” in *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2010.

- [154] F. Perronnin, J. Sánchez, and T. Mensink, “Improving the fisher kernel for large-scale image classification,” in *Proc. of European Conference on Computer Vision (ECCV)*, 2010.
- [155] R. Plutchik, “The nature of emotions,” *American Scientist*, vol. 89, no. 4, pp. 344–350, 2001.
- [156] G.-J. Qi, C. Aggarwal, Q. Tian, H. Ji, and T. Huang, “Exploring context and content links in social media: A latent space method,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 5, pp. 850–862, 2012.
- [157] X. Qian, X.-S. Hua, Y. Tang, and T. Mei, “Social image tagging with diverse semantics,” *IEEE Transactions on Cybernetics*, vol. 44, no. 12, pp. 2493–2508, 2014.
- [158] Z. Qian, P. Zhong, and R. Wang, “Tag refinement for user-contributed images via graph learning and nonnegative tensor factorization,” *IEEE Signal Processing Letters*, vol. 22, no. 9, pp. 1302–1305, 2015.
- [159] N. Rasiwasia, J. Costa Pereira, E. Coviello, G. Doyle, G. R. G. Lanckriet, R. Levy, and N. Vasconcelos, “A new approach to cross-modal multimedia retrieval,” in *Proc. of ACM Multimedia*, 2010.
- [160] T. Rattenbury, N. Good, and M. Naaman, “Towards automatic extraction of event and place semantics from flickr tags,” in *Proc. of ACM SIGIR*, 2007, pp. 103–110.
- [161] A. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, “CNN features off-the-shelf: An astounding baseline for recognition,” in *Proc. of CVPR Workshop on Deep Vision*, 2014, pp. 512–519.
- [162] A. S. Razavian, J. Sullivan, A. Maki, and S. Carlsson, “Visual instance retrieval with deep convolutional networks,” in *Proc. of ICLR Workshops*, 2015.
- [163] F. Richter, S. Romberg, E. Horster, and R. Lienhart, “Leveraging community metadata for multimodal image ranking,” *Multimedia Tools and Applications*, vol. 56, no. 1, pp. 35–62, 2012.
- [164] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. Berg, and L. Fei-Fei, “ImageNet Large Scale Visual Recognition Challenge,” *International Journal of Computer Vision*, 2015, in press.
- [165] H. Saif, M. Fernandez, Y. He, and H. Alani, “Evaluation datasets for Twitter sentiment analysis,” in *Proc. of AI\*IA Emotion and Sentiment in Social and Expressive Media (ESSEM)*, 2013.

- [166] H. Saif, Y. He, and H. Alani, "Semantic sentiment analysis of Twitter," in *Proc. of International Conference on the Semantic Web (ISWC)*, 2012.
- [167] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek, "Image classification with the Fisher vector: Theory and practice," *International Journal of Computer Vision*, vol. 105, no. 3, pp. 222–245, 2013.
- [168] J. Sang, C. Xu, and J. Liu, "User-aware image tag refinement via ternary semantic analysis," *IEEE Transactions on Multimedia*, vol. 14, no. 3, pp. 883–895, 2012.
- [169] J. Sang, C. Xu, and D. Lu, "Learn to personalized image search from the photo sharing websites," *IEEE Transactions on Multimedia*, vol. 14, no. 4, pp. 963–974, 2012.
- [170] N. Sawant, R. Datta, J. Li, and J. Wang, "Quest for relevant tags using local interaction networks and visual content," in *Proc. of ACM MIR*, 2010, pp. 231–240.
- [171] N. Sawant, J. Li, and J. Wang, "Automatic image semantic interpretation using social action and tagging data," *Multimedia Tools and Applications*, vol. 51, no. 1, pp. 213–246, 2011.
- [172] S. Sen, S. Lam, A. Rashid, D. Cosley, D. Frankowski, J. Osterhouse, F. Harper, and J. Riedl, "tagging, communities, vocabulary, evolution," in *Proc. of CSCW*, 2006, pp. 181–190.
- [173] G. Serra, T. Alisi, M. Bertini, L. Ballan, A. Del Bimbo, L. Goix, and C. Licciardi, "STAMAT: A framework for social topics and media analysis," in *Proc. of IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, July 2013, pp. 1–2.
- [174] B. Sigurbjörnsson and R. van Zwol, "Flickr tag recommendation based on collective knowledge," in *Proc. of WWW*, 2008, pp. 327–336.
- [175] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. of ICLR*, 2015.
- [176] S. Sizov, "Geofolk: latent spatial semantics in web 2.0 social media," in *Proc. of ACM WSDM*, 2010, pp. 281–290.
- [177] A. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 12, pp. 1349–1380, 2000.
- [178] N. Srivastava and R. Salakhutdinov, "Multimodal learning with deep boltzmann machines," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 2949–2980, 2014.

- [179] A. Sun, S. Bhowmick, K. Nguyen, and G. Bai, “Tag-based social image retrieval: An empirical evaluation,” *Journal of the American Society for Information Science and Technology*, vol. 62, no. 12, pp. 2364–2381, 2011.
- [180] H. Sundaram, L. Xie, M. De Choudhury, Y.-R. Lin, and A. Natsev, “Multi-media semantics: Interactions between content and community,” *Proceedings of the IEEE*, vol. 100, no. 9, pp. 2737–2758, 2012.
- [181] J. Tang, R. Hong, S. Yan, T.-S. Chua, G.-J. Qi, and R. Jain, “Image annotation by kNN-sparse graph-based label propagation over noisily tagged web images,” *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 2, pp. 14:1–14:15, 2011.
- [182] J. Tang, S. Yan, R. Hong, G.-J. Qi, and T.-S. Chua, “Inferring semantic concepts from community-contributed images and noisy tags,” in *Proc. of ACM MM*, 2009, pp. 223–232.
- [183] R. C. Team, “R: A language and environment for statistical computing. vienna, austria: R foundation for statistical computing; 2008,” 2011.
- [184] J. B. Tenenbaum, V. De Silva, and J. C. Langford, “A global geometric framework for nonlinear dimensionality reduction,” *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.
- [185] M. Thelwall, K. Buckley, G. Paltoglou, D. Cai, and A. Kappas, “Sentiment strength detection in short informal text,” *Journal of the American Society for Information Science and Technology*, vol. 61, no. 12, pp. 2544–2558, 2010.
- [186] L. C. Totti, F. A. Costa, S. Avila, E. Valle, W. Meira Jr, and V. Almeida, “The impact of visual attributes on online image diffusion,” in *Proc. of Web-Sci*, 2014.
- [187] 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 *Proc. of ACM ICMR*, 2012, pp. 9:1–9:8.
- [188] L. Tucker, “Some mathematical notes on three-mode factor analysis,” *Psychometrika*, vol. 31, no. 3, pp. 279–311, 1966.
- [189] A. Tumasjan, T. O. Sprenger, P. G. Sandner, and I. M. Welp, “Predicting elections with Twitter: What 140 characters reveal about political sentiment.” in *Proc. of AAAI International Conference on Weblogs and Social Media (ICWSM)*, 2010.
- [190] J. Turian, L. Ratinov, and Y. Bengio, “Word representations: a simple and general method for semi-supervised learning,” in *Proc. of ACL Annual Meeting of the Association for Computational Linguistics*, 2010.
- [191] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders, “Selective search for object recognition,” *International Journal of Computer Vision*, vol. 104, no. 2, pp. 154–171, 2013.

- [192] T. Uricchio, L. Ballan, M. Bertini, and A. Del Bimbo, “An evaluation of nearest-neighbor methods for tag refinement,” in *Proc. of ICME*, 2013.
- [193] K. van de Sande, T. Gevers, and C. Snoek, “Evaluating color descriptors for object and scene recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1582–1596, 2010.
- [194] J. Verbeek, M. Guillaumin, T. Mensink, and C. Schmid, “Image annotation with TagProp on the MIRFLICKR set,” in *Proc. of ACM MIR*, 2010, pp. 537–546.
- [195] Y. Verma and C. V. Jawahar, “Image annotation using metric learning in semantic neighbourhoods,” in *Proc. of ECCV*, 2012.
- [196] —, “Exploring svm for image annotation in presence of confusing labels,” in *Proc. of BMVC*, 2013.
- [197] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, “Extracting and composing robust features with denoising autoencoders,” in *Proc. of International Conference on Machine Learning (ICML)*, 2008, pp. 1096–1103.
- [198] L. von Ahn and L. Dabbish, “Labeling images with a computer game,” in *Proc. of ACM CHI*, 2004.
- [199] D. Vreeswijk, K. van de Sande, C. Snoek, and A. Smeulders, “All vehicles are cars: Subclass preferences in container concepts,” in *Proc. of ACM ICMR*, 2012, pp. 8:1–8:7.
- [200] C. Wang, L. Zhang, F. Jing, and H.-J. Zhang, “Image annotation refinement using random walk with restarts,” in *Proc. of ACM MM*, 2006, pp. 647–650.
- [201] G. Wang, D. Hoiem, and D. Forsyth, “Building text features for object image classification,” in *Proc. of CVPR*, 2009, pp. 1367–1374.
- [202] —, “Learning image similarity from Flickr groups using stochastic intersection kernel machines,” in *Proc. of ICCV*, 2009, pp. 428–435.
- [203] J. Wang, J. Zhou, H. Xu, T. Mei, X.-S. Hua, and S. Li, “Image tag refinement by regularized latent Dirichlet allocation,” *Computer Vision and Image Understanding*, vol. 124, no. 0, pp. 61–70, 2014.
- [204] M. Wang, X.-S. Hua, and H.-J. Zhang, “Towards a relevant and diverse search of social images,” *IEEE Transactions on Multimedia*, vol. 12, no. 8, pp. 829–842, 2010.
- [205] M. Wang, B. Ni, X.-S. Hua, and T.-S. Chua, “Assistive tagging: A survey of multimedia tagging with human-computer joint exploration,” *ACM Computing Surveys*, vol. 44, no. 4, pp. 25:1–25:24, 2012.
- [206] M. Wang, D. Cao, L. Li, S. Li, and R. Ji, “Microblog sentiment analysis based on cross-media bag-of-words model,” in *Proc. of International Conference*

- on Internet Multimedia Computing and Service (ICIMCS)*, 2014, pp. 76:76–76:80.
- [207] W. Wang and Q. He, “A survey on emotional semantic image retrieval,” in *Proc. of IEEE International Conference on Image Processing (ICIP)*, Oct 2008, pp. 117–120.
- [208] Z. Wang, P. Cui, L. Xie, H. Chen, W. Zhu, and S. Yang, “Analyzing social media via event facets,” in *Proc. of ACM International Conference on Multimedia (MM)*, 2012, pp. 1359–1360.
- [209] Y. Wei, W. Xia, J. Huang, B. Ni, J. Dong, Y. Zhao, and S. Yan, “CNN: single-label to multi-label,” *CoRR*, vol. abs/1406.5726, 2014. [Online]. Available: <http://arxiv.org/abs/1406.5726>
- [210] L. Wu, X.-S. Hua, N. Yu, W.-Y. Ma, and S. Li, “Flickr distance,” in *Proc. of ACM MM*, 2008, pp. 31–40.
- [211] L. Wu, R. Jin, and A. Jain, “Tag completion for image retrieval,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 3, pp. 716–727, 2013.
- [212] L. Wu, L. Yang, N. Yu, and X.-S. Hua, “Learning to tag,” in *Proc. of WWW*, 2009, pp. 361–370.
- [213] P. Wu, S. C.-H. Hoi, P. Zhao, and Y. He, “Mining social images with distance metric learning for automated image tagging,” in *Proc. of ACM WSDM*, 2011, pp. 97–206.
- [214] Z. Wu and M. Palmer, “Verb semantic and lexical selection,” in *Proc. of ACL*, 1994, pp. 133–138.
- [215] H. Xu, J. Wang, X.-S. Hua, and S. Li, “Tag refinement by regularized LDA,” in *Proc. of ACM MM*, 2009, pp. 573–576.
- [216] X. Xu, A. Shimada, and R. Taniguchi, “Tag completion with defective tag assignments via image-tag re-weighting,” in *Proc. of ICME*, 2014, pp. 1–6.
- [217] K. Yanai, “World Seer: A realtime geo-tweet photo mapping system,” in *Proc. of ACM International Conference on Multimedia Retrieval (ICMR)*, 2012, pp. 65:1–65:2.
- [218] K. Yang, X.-S. Hua, M. Wang, and H.-J. Zhang, “Tag tagging: Towards more descriptive keywords of image content,” *IEEE Transactions on Multimedia*, vol. 13, no. 4, pp. 662–673, 2011.
- [219] Y. Yang, Y. Gao, H. Zhang, J. Shao, and T.-S. Chua, “Image tagging with social assistance,” in *Proc. of ACM ICMR*, 2014, pp. 81–88.
- [220] Y. Yang, P. Cui, W. Zhu, H. V. Zhao, Y. Shi, and S. Yang, “Emotionally representative image discovery for social events,” in *Proc. of ACM International Conference on Multimedia Retrieval (ICMR)*, 2014, pp. 177:177–177:184.

- [221] A. Yavlinsky, E. Schofield, and S. Rüger, “Automated image annotation using global features and robust nonparametric density estimation,” in *Proc. of ACM CIVR*, 2005.
- [222] D. Yoo, S. Park, J.-Y. Lee, and I. Kweon, “Multi-scale pyramid pooling for deep convolutional representation,” in *Proc. of CVPR Workshops*, 2015, pp. 71–80.
- [223] S. Zhang, J. Huang, Y. Huang, Y. Yu, H. Li, and D. N. Metaxas, “Automatic image annotation using group sparsity,” in *Proc. of CVPR*, 2010.
- [224] W. Zhao, H. Jegou, and G. Gravier, “Oriented pooling for dense and non-dense rotation-invariant features,” in *Proc. of BMVC*, Sep. 2013.
- [225] X. Zhao, F. Zhu, W. Qian, and A. Zhou, “Impact of multimedia in Sina Weibo: Popularity and life span,” in *Proc. of Chinese Semantic Web Symposium and the First Chinese Web Science Conference (CSWS & CWSC)*, 2012.
- [226] B. Zhou, V. Jagadeesh, and R. Piramuthu, “ConceptLearner: Discovering visual concepts from weakly labeled image collections,” in *Proc. of CVPR*, 2015.
- [227] D. Zhou, J. Huang, and B. Schölkopf, “Learning with hypergraphs: Clustering, classification, and embedding,” in *Proc. of NIPS*, 2006, pp. 1601–1608.
- [228] G. Zhu, S. Yan, and Y. Ma, “Image tag refinement towards low-rank, content-tag prior and error sparsity,” in *Proc. of ACM MM*, 2010, pp. 461–470.
- [229] S. Zhu, C.-W. Ngo, and Y.-G. Jiang, “Sampling and ontologically pooling web images for visual concept learning,” *IEEE Transactions on Multimedia*, vol. 14, no. 4, pp. 1068–1078, 2012.
- [230] X. Zhu, W. Nejdl, and M. Georgescu, “An adaptive teleportation random walk model for learning social tag relevance,” in *Proc. of ACM SIGIR*, 2014, pp. 223–232.
- [231] J. Zhuang and S. Hoi, “A two-view learning approach for image tag ranking,” in *Proc. of ACM WSDM*, 2011, pp. 625–634.
- [232] C. L. Zitnick and P. Dollár, “Edge boxes: Locating object proposals from edges,” in *Proc. of ECCV*, 2014.
- [233] A. Znaidia, H. Le Borgne, and C. Hudelot, “Tag completion based on belief theory and neighbor voting,” in *Proc. of ACM ICMR*, 2013, pp. 49–56.