

Computer Science

Multimodal Concept Detection and Annotation in Image and Video Collections

Satoru Ishikawa



Multimodal Concept Detection and Annotation in Image and Video Collections

Satoru Ishikawa

A doctoral dissertation completed for the degree of Doctor of
Science (Technology) to be defended, with the permission of the
Aalto University School of Science, Remote connection link, on 14
August 2020 at 12 o'clock noon

Aalto University
School of Science
Computer Science

Supervising professor

Samuel Kaski, Aalto University School of Science, Finland

Thesis advisor

Jorma Laaksonen, Aalto University School of Science, Finland

Preliminary examiners

Dorota Glowacka, University of Helsinki, Finland

Vasileios Mezaris, Center for Research and Technology Hellas, Greece

Opponent

Joni Kämäräinen, Tampere University, Finland

Aalto University publication series

DOCTORAL DISSERTATIONS 104/2020

© 2020 Satoru Ishikawa

ISBN 978-952-60-3952-7 (printed)

ISBN 978-952-60-3954-1 (pdf)

ISSN 1799-4934 (printed)

ISSN 1799-4942 (pdf)

<http://urn.fi/URN:ISBN:978-952-60-3954-1>

Unigrafia Oy

Helsinki 2020

Finland



Author

Satoru Ishikawa

Name of the doctoral dissertation

Multimodal Concept Detection and Annotation in Image and Video Collections

Publisher School of Science**Unit** Computer Science**Series** Aalto University publication series DOCTORAL DISSERTATIONS 104/2020**Field of research** Pattern Recognition**Manuscript submitted** 17 June 2020**Date of the defence** 14 August 2020**Permission for public defence granted (date)** 16 June 2020**Language** English **Monograph** **Article dissertation** **Essay dissertation****Abstract**

The World Wide Web has become a common-place for finding for all kinds of purposes. The amount of data which one user can be dealing with has become large and its size is countinuously growing. The relevant data for users have not only become large, but also diverse. Hence, searching relevant information from such large and diverse resources is a critical task. However, users can not always formulate appropriate queries for finding the desired resources. In order to retrieve relevant information, the semantic relationships of the information in different modalities would need to be known and specified.

This thesis approaches the multimodal cross-domain semantic retrieval and fusion problem from the point of view of content-based visual analysis and statistical natural language analysis. It also aims at using cross-domain textual semantics to generate pseudo tags for images to improve the performance of the information retrieval task. The main focus of the thesis is in bridging the semantic gap between textual and visual content domains.

In order to combine and project the unimodal information to multimodal space, two approaches are used: one is the Multimodal Deep Boltzmann Machine (DBM) and the other is the late fusion of unimodal Support Vector Machines (SVM). One problem of the non-linear SVM approach is its high calculation cost. In this dissertation, the homogeneous kernel map method is used to improve the efficiency of SVM. In our experiments, we adopted deep convolutional neural network features, particularly GoogLeNet features, and the retrieval results of the SVM-based approaches improved to be nearly equal to those of the Multimodal DBM approach.

One drawback of the multimodal information retrieval task is the requirement to be able to perform queries in each unimodal domain. In our experiments, if the query for image domain is missing or not appropriate, the approach is just the same as ordinal text search. Additionally, the image contents and its textual description do not always match. In order to improve the multimodal information retrieval, the method of pseudo tag generation is proposed in this thesis. The generation of pseudo tags is based on a text-image semantic map, which is calculated by the cooccurrence of latent topics in text and visual concepts in text-image data. In the experiments, the multimodal information retrieval results were considerably improved by using the pseudo tags.

Keywords**ISBN (printed)** 978-952-60-3952-7**ISBN (pdf)** 978-952-60-3954-1**ISSN (printed)** 1799-4934**ISSN (pdf)** 1799-4942**Location of publisher** Helsinki**Location of printing** Helsinki **Year** 2020**Pages** 227**urn** <http://urn.fi/URN:ISBN:978-952-60-3954-1>

Abstract

The World Wide Web has become a common-place for finding information for all kinds of purposes. The amount of data which one user can be dealing with has become large and its size is continuously growing. The relevant data for users have not only become large, but also diverse. Hence, searching relevant information from such large and diverse resources is a critical task. However, users can not always formulate appropriate queries for finding the desired resources. In order to retrieve relevant information, the semantic relationships of the information in different modalities would need to be known and specified.

This thesis approaches the multimodal cross-domain semantic retrieval and fusion problem from the point of view of content-based visual analysis and statistical natural language analysis. It also aims at using cross-domain textual semantics to generate pseudo tags for images to improve the performance of the information retrieval task. The main focus of the thesis is in bridging the semantic gap between textual and visual content domains.

In order to combine and project the unimodal information to multimodal space, two approaches are used: one is the Multimodal Deep Boltzmann Machine (DBM) and the other is the late fusion of unimodal Support Vector Machines (SVM). One problem of the non-linear SVM approach is its high calculation cost. In this dissertation, the homogeneous kernel map method is used to improve the efficiency of SVM. In our experiments, we adopted deep convolutional neural network features, particularly GoogLeNet features, and the retrieval results of the SVM-based approaches improved to be nearly equal to those of the Multimodal DBM approach.

One drawback of the multimodal information retrieval task is the requirement to be able to perform queries in each unimodal domain. In our experiments, if the query for image domain is missing or not appropriate, the approach is just the same as ordinal text search. Additionally, the image contents and its textual description do not always match. In order to improve the multimodal information retrieval, the method of pseudo tag generation is proposed in this thesis. The generation of pseudo tags is based on a text-image semantic map, which is calculated by the co-occurrence of latent topics in text and visual concepts in text-image data. In the experiments, the multimodal information retrieval results were considerably improved by using the pseudo tags.

Preface

The basis of my multimodal information retrieval research stemmed from my deep interest in the information retrieval field and the passion for learning and developing new methodologies. However, I only had knowledge about text based information retrieval at the beginning. This led to many obstacles in learning the research and I even had to leave Finland in the middle of the study. Hence, without any support, I could not have finalized this research.

First of all, I would like to thank my first supervising professor Erkki Oja. He not only invited me to his research team, but also gave advice and guided towards the right path to the goal. After Erkki was retired, Juha Karhunen became my second supervising professor. He took my research seriously and gave great support for my publications. Unfortunately, he also retired and I had to leave Finland. Samuel Kaski became my last supervisor and without his support I could not have continued my research.

I would like to thank to my advisor, Jorma Laaksonen, who supported me from the beginning to this end, even after I left Finland. Without the cooperation and advices of Ville Viitaniemi, Mats Sjöberg, and Markus Koskela, who were PicSOM group members and the co-authors of my publications, I could not have done most of my research.

I would like to thank Prof. Joni Kämäräinen, who has promised to serve as my opponent, and my pre-examiners, Dorota Glowacka and Vasileios Mezaris, who provided valuable advice and comments. In the end, my special thanks to all the people who supported my long journey.

Fukushima, Japan, June 28, 2020,

Satoru Ishikawa

Contents

Preface	3
Contents	5
List of Publications	9
Author's Contribution	11
List of Figures	13
List of Tables	17
List of Abbreviations	19
List of Symbols	21
1. Introduction	23
1.1 Background of Thesis	23
1.2 Contributions of Thesis	26
1.3 Overview of Thesis	26
2. Text-Based Information Retrieval	27
2.1 Statistical Frequency-Based Approach	28
2.2 Topic Model-Based Approaches	30
2.3 Text Similarity Metrics	32
2.4 Performance Measures	33
2.4.1 Precision and Recall	34
2.4.2 Mean Average Precision	35
2.4.3 Average Area under the ROC curve	36
2.4.4 Subset Accuracy	36
3. Visual Information Retrieval	37

3.1	Feature Extraction	37
3.1.1	Low-level Features	38
3.1.2	Deep Convolutional Neural Networks	42
3.1.3	Semantic Concepts for Visual Information Retrieval	44
3.2	Classification	44
3.2.1	Support Vector Machine	44
3.2.2	Single- and Multi-label Classification	50
4.	Multimodal Information Retrieval	53
4.1	Traditional Fusion-based Approaches	54
4.2	Deep Boltzmann Machine for Multimodal Information Retrieval	56
4.2.1	Restricted Boltzmann Machine	56
4.2.2	Replicated Softmax Model	59
4.2.3	Gaussian-Bernoulli RBM	60
4.2.4	Deep Boltzmann Machine	60
4.2.5	Multimodal DBM	61
4.3	Information Transfer to Other Modality	63
4.3.1	Topic-Concept Similarity Map	63
4.3.2	Pseudo Tag Generation with Similarity Map	64
4.3.3	Unsupervised Pseudo Tag Generation	65
4.4	Semantic Concept Vectors	66
4.5	Discussion	66
4.6	Application Examples	67
4.6.1	PicSOM	67
4.6.2	VisualLabel	68
5.	Experimental Evaluations	69
5.1	TRECVID Semantic Indexing Task	70
5.1.1	Setup	70
5.1.2	TRECVID Dataset	70
5.1.3	Experiments and Results	71
5.1.4	Discussion	74
5.2	Single-label Multimodal Information Retrieval	74
5.2.1	Setup	75
5.2.2	MIR Flickr Dataset	75
5.2.3	Experiments and Results	76
5.2.4	Discussion	80
5.3	Multi-label Information Retrieval	82

5.3.1	Setup	82
5.3.2	NUS-WIDE Dataset	83
5.3.3	Experiments and Results	83
5.3.4	Discussion	86
5.4	Multimodal Experiments with Pseudo Tags	86
5.4.1	Setup	87
5.4.2	ImageCLEF 2010 Wikipedia Collection	88
5.4.3	Experiments and Results	89
5.4.4	Discussion	91
6.	Conclusions	97
	References	101
	Publications	111

List of Publications

This thesis consists of an overview and of the following publications which are referred to in the text by their Roman numerals.

I Satoru Ishikawa and Jorma Laaksonen. Uni- and Multimodal Methods for Single- and Multi-label Recognition. *Multimedia Tools and Applications*, Volume 76, issue 21, pp.22405-22423 , October 2017.

II Mats Sjöberg, Markus Koskela, Satoru Ishikawa and Jorma Laaksonen. Real-Time Large-Scale Visual Concept Detection with Linear Classifiers. In *Proceedings of 21st International Conference on Pattern Recognition*, Tsukuba, Japan, pp.421-424, November 2012.

III Mats Sjöberg, Markus Koskela, Satoru Ishikawa and Jorma Laaksonen. Large-Scale Visual Concept Detection with Explicit Kernel Maps and Power Mean SVM. In *Proceedings of ACM International Conference on Multimedia Retrieval (ICMR2013)*, Dallas, Texas, USA, pp.239-246, April 16-19 2013.

IV Iftikhar Ahmad, Petri Rantanen, Pekka Sillberg, Jorma Laaksonen, Shuhua Liu, Thomas Fross, Aqdas Malik, Marko Nieminen, Rakshith Shetty, Satoru Ishikawa, Jarno Kallio, Jukka P. Saarinen, Moncef Gabbouj and Jari Soini. VisualLabel: An Integrated Multimedia Content Management and Access Framework. *Frontiers in Artificial Intelligence and Applications, Volume 301: Information Modelling and Knowledge Bases XXIX*, pp.321-342, 2018.

V Ville Viitaniemi, Mats Sjöberg, Markus Koskela, Satoru Ishikawa and Jorma Laaksonen. Advances in Visual Concept Detection: Ten Years of TRECVID. *Advances in Independent Component Analysis and Learning Machines, 1st Edition*, pp.249-278 , 2015.

VI Satoru Ishikawa and Jorma Laaksonen. Comparing and Combining Unimodal Methods for Multimodal Recognition. In *Proceedings of the 14th International Workshop on Content-Based Multimedia Indexing*, Bucharest, Romania, pp.1-6, June 2016.

VII Satoru Ishikawa, Jorma Laaksonen and Juha Karhunen. Image Pseudo Tag Generation with Deep Boltzmann Machine and Topic-Concept Similarity Map. In *Proceeding of the 30th International Joint Conference of Neural Network*, Anchorage, Alaska, USA, pp.1305-1312, June 2017.

Author's Contribution

Publication I: “Uni- and Multimodal Methods for Single- and Multi-label Recognition”

The author had the main responsibility of this publication. The author solely planned the experiments, collected and processed the used datasets, prepared the code and run the experiments related to Deep Boltzmann Machines, and draw the conclusions from them. The author carried out the major work of writing the article, where the coauthor participated. This journal article is an extend version of the conference article Publication VI.

Publication II: “Real-Time Large-Scale Visual Concept Detection with Linear Classifiers”

The author participated in preparing the data for experiments, carrying out the experiments, drawing the conclusions, and writing the publication.

Publication III: “Large-Scale Visual Concept Detection with Explicit Kernel Maps and Power Mean SVM”

The author participated in preparing the data for experiments, carrying out the experiments, drawing the conclusions, and writing the publication.

Publication IV: “VisualLabel: An Integrated Multimedia Content Management and Access Framework”

The author was the main responsible for preparing the data and running the experiments for the TRECVID 2014 evaluation, whose results are reported in this publication. The author participated in the writing of those sections of the article that describe the PicSOM back-end and its use for keyword annotation.

Publication V: “Advances in Visual Concept Detection: Ten Years of TRECVID”

The author was the main responsible for preparing the data and running the experiments for the TRECVID 2014 evaluation, whose results are reported in this article. The author participated in the overall planning the article, drawing the conclusions and writing the text.

Publication VI: “Comparing and Combining Unimodal Methods for Multimodal Recognition”

The author had the main responsibility of this publication. The author solely planned the experiments, collected and processed the used datasets, prepared the code and run the experiments related to Deep Boltzmann Machines, and draw the conclusions from them. The author carried out the major work of writing the article, where the coauthor participated.

Publication VII: “Image Pseudo Tag Generation with Deep Boltzmann Machine and Topic–Concept Similarity Map”

The author had the main responsibility of this publication. The author solely planned the experiments, collected and processed the used datasets, prepared the code and run the experiments related to Deep Boltzmann Machines, and draw the conclusions from them. The invention of the novel method for image pseudo tag generation was solely by the author. The author carried out the major work of writing the article, where the coauthor participated.

List of Figures

2.1	A confusion matrix.	34
2.2	An example of the trade-off between precision and recall from [41]. The plot shown the results of retrieval systems tested in a TREC evaluation 1999.	35
3.1	The four phases of SIFT feature extraction. Detector detects most salient regions, called keypoints, which describe image. Descriptor extracts and quantifies the keypoints and their surroundings to a feature vector.	40
3.2	The main steps of the Bag of Visual words (BoV) approach. After extracting the feature vectors, find the centroids of 'words' with semantically meaningful patterns in the images by clustering. Then, build the bins of each 'word' based on its frequency in each image.	41
3.3	Toy example of constructing a three-level pyramid [66]. The image has three feature types, indicated by circles, diamonds, and crosses. At the top, the image in three different levels of resolution to be subdivided. Next, for each level of resolution and each channel, the features that fall in each spatial bin are counted. Then, calculate weights for each histogram. Please refer to [66] for more details.	41
3.4	The architecture of a Feed Forward Network [87]. The left most layer is the input layer which feeds the input data to the hidden layer. The middle layer is the hidden layer which outputs the weighted sum of the data. The right most layer is the output layer which transforms the outputs from the hidden layer to suitable outputs for the task. In a binary classification task, the output will be 0 or 1.	43

3.5 The architecture of a Deep convolutional neural network (DCNN) [102]. DCNN converts the low-level features to the high-level features. This figure shows the dimensionality of outputs are relatively smaller than inputs. 43

3.6 The architecture of semantic concept detection. The architecture is similar to the pipeline of a regular image classifier. First, the semantic concept classifier to trained by using the extracted low-level features of the training dataset. Then, the semantic concepts of the test data are detected by using the trained detector. 45

3.7 **Left:** Two hyperplanes which correctly classify the training vectors. **Right:** Maximizing the margin in order to optimize the decision boundary. 46

4.1 The architecture of early fusion information retrieval. An early fusion approach basically extracts several different features from the datasets and combines them before training the model. 54

4.2 The architecture of late fusion information retrieval. A late fusion approach trains several different machine learning models and fuses their results to produce the final results. . 55

4.3 The architecture of the Boltzmann Machine. The Boltzmann Machine is a fully connected undirected graph. Compared with the other neural network models such as CNN, the parameters of each layer are relatively large. Hence, its computational cost tends to be higher than other models. . 57

4.4 The architecture of the Restricted Boltzmann Machine. Unlike the Boltzmann Machine, the connections of RBM only exist between the layers. Therefore, their computational cost becomes smaller. 57

4.5 Multimodal DBM [106]. The left side layers are an image-specific DBM and the right side layers are a text-specific DBM. 62

4.6 The process of creating a similarity map between the topics of articles and the image concepts. 64

5.1 MXIAP values for all submissions to the TRECVID 2014 semantic indexing task. Our runs highlighted. 72

5.2	Two positive example images for MIRFLICKR-1M concept <i>sea_r1</i> . Left: Ranking improved with multimodal approach. Right: Ranking worsened with multimodal approach. See text for details.	81
5.3	Two false positive example images for MIRFLICKR-1M concept <i>sea_r1</i> . Left: False recognition became less probable with multimodal approach. Right: False recognition became more probable with multimodal approach. See text for details.	81
5.4	Left: Correctly multi-label recognized image, whose true labels are <i>animal</i> and <i>flower</i> . Right: Failed multi-label recognition case, whose true labels are <i>animal</i> , <i>grass</i> , <i>mountain</i> , and <i>sky</i>	86
5.5	Example of pseudo tag generation for a tagged image.	91
5.6	Example of pseudo tag generation for an image with no tags.	92
5.7	Example of an image that contains concept "sea_r1" and its retrieval results without and with the generated tags. Original tags: <i>beach</i> , <i>ocean</i> , <i>coast</i> , <i>pacific</i> , etc. Generated tags: <i>convert</i> , <i>air</i> , island , <i>river</i> , water , <i>airport</i> , <i>aircraft</i> , <i>park</i> , sea , islands , <i>international</i> , <i>lake</i> , etc.	94
5.8	Example of an image that contains concept "animals" and its retrieval results without and with the generated tags. Original tag: <i>bilbao</i> . Generated tags: <i>world</i> , <i>city</i> , <i>people</i> , <i>university</i> , <i>american</i> , <i>island</i> , <i>building</i> , <i>film</i> , <i>history</i> , animal , animals , <i>location</i> , <i>king</i> , <i>author</i> , <i>book</i> , <i>modern</i> , <i>species</i> , dog , dogs	94
5.9	Example of an image that contains concept "indoor" and its retrieval results without and with the generated tags. Original tags: <i>toys</i> , <i>robot</i> . Generated tags: <i>people</i> , <i>news</i> , <i>transport</i> , <i>city</i> , <i>thumb</i> , <i>airport</i> , <i>station</i> , <i>grand</i> , <i>war</i> , <i>engine</i> , <i>cars</i> , <i>aircraft</i> , <i>world</i> , <i>building</i> , <i>convert</i> , <i>car</i> , <i>university</i> , <i>air</i> , <i>american</i> , <i>road</i>	94

List of Tables

2.1	Components of the tf-idf weighting scheme. r_{w_i} is the weight of word w_i . $tf_{w,d}$ is the frequency of the word w in document d and df_w is the number of documents which include the word w . idf_w is the inverse document frequency of word w	30
5.1	An overview of our runs submitted for the TRECVID 2014 evaluation. glob. represents global BoV bag of visual words, Fisher vectors (FV) + VLAD and convolutional neural network (CNN) features. We submitted four runs: Row1 uses global and BoV features. Row2 combines BoV, Fisher vector and CNN features. Row3 uses CNN features with hard negative mining. Row4 combines BoV, Fisher vector and CNN features with hard negative mining.	71
5.2	Processing times (secs) for 1, 50 and 500 concepts and MX-IAP scores for the hand-crafted features of TRECVID2011 dataset. linear represents our baseline model which used linear SVM classifier only. Multi-learn represents the fusion results of multiple learning classifiers. Hkm-INT represents SVM with homogeneous kernel map of intersection. Hkm- χ^2 represents SVM with the homogeneous kernel map of χ^2 kernel.	73

5.3	Processing time and MXIAP scores of TRECVID2012 dataset (SIFT based feature). Linear, hkm-INT, and hkm- χ^2 are the same setting as in Table 5.2. Power mean SVM is used with the homogeneous kernel map of intersection (pm-INT) and χ^2 (pm- χ^2) kernel. As references of non-linear SVM results, we used non-linear kernels of RBF kernel (K_{RBF}), the exponential of χ^2 kernel ($K_{\chi^2}^{exp}$), and the exponential of intersection kernel (K_{INT}^{exp}).	73
5.4	MIRFLICKR-1M 38 concept classification results with different models. LIN = linear SVM, RBF = non-linear RBF kernel SVM, text = 2000-dimensional 0/1 tag features, word2vec = 200-dimensional word2vec features, pre-t = DBM pre-training performed with 975,000 unannotated images and/or tags, PHOW, etc. = hand-crafted features of [106], semantic = semantic concept vectors. Best results in each group are bolded.	78
5.5	MIRFLICKR-1M 94 concept classification results with different models. RBF = non-linear RBF kernel SVM, text = 2000-dimensional 0/1 tag features, pre-t = DBM pre-training performed with 975,000 unannotated images and/or tags, sem = semantic concept vectors. Best results in each group are bolded.	79
5.6	Examples of concept-wise MAP measure differences between unimodal and multimodal results in MIRFLICKR-1M. The row labels refer to the corresponding results in Table 1. . . .	81
5.7	NUS-WIDE 81 classification results with different models. text = 1,000-dimensional 0/1 tag features, sem. = 500-dimensional semantic concept vectors. Results B1 and B2 are from [67]. Best results in each group are labeled.	84
5.8	Classification results with different features and uni- and multimodal models. Results of [106] were obtained using additionally sparsity, fine-tuning and dropout. Best results in each group are bolded.	90

List of Abbreviations

aAUC Average Area the Under the ROC Curve

AUC Area Under the ROC Curve

BoV Bag of Visual keywords

CBIR Content-Based Image Retrieval

CBM Conditional Bernoulli Mixture

CCA Canonical Correlation Analysis

CNN Convolutional Neural Network

DBM Deep Boltzmann Machine

DBN Deep Belief Network

DCNN Deep Convolutional Neural Network

DoG Difference of Gaussians

FN False Negative

FP False Positive

GB Gradient Boosting

GMM Gaussian Mixture Model

KL Kullback–Leibler

LDA Latent Dirichlet Allocation

LR Linear Regression

MAP Mean Average Precision

MCMC Markov Chain Monte Carlo

MXIAP Mean Extended Inferred Average Precision

NIST National Institute of Standard and Technology

PLSA Probabilistic Latent Semantic Analysis

Prec@50 Precision at Rank 50

RBF Radial Basis Function

RBM Restricted Boltzmann Machine

ROC Receiver Operating Characteristic

SIFT Scale Invariant Feature Transform

SOM Self-Organizing Map

ssACC Subset Accuracy

SVM Support Vector Machine

TN True Negative

TP True Positive

VSM Vector Space Model

List of Symbols

$B(\cdot)$ beta distribution

C concept vocabulary

$D(\cdot, \cdot)$ distance

$Dir(\cdot)$ Dirichlet distribution

$E(\cdot)$ energy state

$K(\cdot, \cdot)$ kernel function

$Multinomial(\cdot)$ multinomial distribution

N number of documents

N_w number of documents which contain word w

Q number of queries

$Sim(\cdot, \cdot)$ similarity

$\Gamma(\cdot)$ gamma function

$\mathbb{E}[\cdot]$ Expectation

$\mathbb{I}[\cdot]$ indicator function

θ topic mixture

df document frequency

idf inverse document frequency

k topics

n number of occurrences

d_j j^{th} document

List of Symbols

w_i i^{th} word

tf term frequency

1. Introduction

1.1 Background of Thesis

The internet has become an indispensable part of our daily life, the amount of information contained in it is extremely large and the contents of the information is excessively diverse. For example, the size of Google's index is at least 45 billion web pages, but in 2020, the size is more than 64 billion web pages [117]. Moreover, the amount of information on the web is continuously growing fast. In Twitter, only 5,000 tweets were sent per day in 2007, but the pace has increased to over 350,000 tweets sent per minute, 500 million tweets per day and around 200 billion tweets per year [70]. Because of such a huge amount of data, it is a very hard task for a regular internet user to find the required information. Also the personal data stored in computers and mobile devices, such as photo images and text documents, has become large.

Searching the demanded information from such enormous data on the web, general purpose search engines, such as Google, have had a central role. Because the contents or the metadata on the web are mostly based on text, the search engines utilize text-based keyword search as the main search technique. That is, the user needs to know the words which are related to the current information demand. If the user does not have any related text information to describe his demand, it is difficult to use most of the current search engines efficiently. For instance, if the user has non-textual information, such as pictures, music clips, videos, etc., but he does not know any textual information or metadata about it, and then wants to search and know the details of it, it is hard to find appropriate information by using only text-based search engines.

There exist several solutions to improve the quality of textual search

[78, 119, 108], such as tagging the information resources with appropriate labels. The user tagging is the most popular approach and it is used in many areas and applications [21, 39, 121]. However, it is a very hard task because tagging huge amounts of information by human effort is extremely demanding and time consuming. In addition, there are always semantical differences between the users, modalities, and tags. Each user has a different background, and hence, the users' taggings are not based on common definitions and experiences. For example, one person may upload a picture of his friend to a social network service with a tag "Steve". If the user uploaded to a small database and all "Steve" in it are his friend, there are no problem to have the tag "Steve". However, if user uploaded to a large database such as the internet, there will be many "Steve" tagged pictures. In general, the famous "Steve" pictures such as "Steve Jobs" or "Steve McQueen" are uploaded more. Therefore, his friend's picture would hardly be retrieved and even if it were retrieved, his picture is not appropriate information for the majority of searches. Because words which have multiple meanings can be used as tags, like in this case, it is possible that the tags mislead rather than improve the search results. The tagging is usually done by the owner of the contents or by majority voting which assigns the most voted tags. In either case, the user's demanded information and tagged information are not always matched. This is because of the semantic difference or gap between different modalities.

One possible approach to solve the misunderstanding of the meaning is to gather the information of the targets from many different modalities [116, 105, 9]. Multimodal search methods combine different search methods to find the appropriate information from multiple domain resources. Some applications have been implemented. For example, Google image search [36] can find images with a text and an image as the query inputs. MMRetrieval [125] allows to search multimodal information with multimedia and multilingual queries. In a specific domain, such as in the medical field, multimodal information retrieval approaches have been researched for supporting the domain-specific decisions [86, 24]. In Aalto University, research on visual contents of multimedia resources has been done by developing the PicSOM visual system for statistical media and text analysis [64].

However, if one wants to know the appropriate text query for finding multimedia resources, this task cannot be completely solved due to the *semantic gap* [89] between the different modalities. In the case of textual

and visual information content fusion, the *semantic gap* is most problematic. The *semantic gap* means that a target object has meaning differences between different modality representations.

For example, images maybe indexed by only color information and the metadata by only the objects in the image. Therefore, it is important to have semantical relationships between image features and text features.

The extraction of semantics of the visual content is more difficult than that of textual search, because the visual and textual analyses are dealing with different semantic levels. In the textual domain, a word is usually the lowest level entity for the analysis. The word itself can have one or more meanings. On the other hand, low-level features in visual analysis do not have semantics themselves. In order to associate visual contents with textual information, we need to cluster and define the lowest level of visual information to semantically meaningful high-level features first. Hence, high-level visual features, unlike textual words, are ambiguous. For instance, finding semantics of "car" from text only requires that a word "car" and its synonyms should be found, but finding semantics from images requires to find visual features which are visually similar to "car". In order to achieve such a high-level semantic representation, it is common to use machine learning algorithms to learn and infer the semantics from the visual data. The semantic differences between the modalities make the multimodal information retrieval task very hard. However, there also exists the possibility that the semantic differences can help to find or transfer useful information between the modalities in the information retrieval task because the information in the different modalities can complement each other.

There exists several ambitious research projects for the multimodal information retrieval task. For example, deep learning has recently shown a great success in many multimedia retrieval tasks [33, 127, 8, 123, 109]. In particular, Srivastava et al [105] applied the Deep Boltzmann Machine (DBM) for multimodal search task, and some other works have been based on canonical correlation analysis [57, 4].

In this thesis, I have applied the Support Vector Machine (SVM) [99], which has been experimented with in our research group, and the DBM approach, inspired by [105], for the multimodal information retrieval task. The thesis also demonstrated how generating "pseudo tags" from semantic information in a different modality improves the multimodal search task. The details of the work will be shown in later sections.

1.2 Contributions of Thesis

The main contributions of the thesis are:

- Research development and experiments on several unimodal information retrieval methods, features for them and their combination techniques, in the application setup of concept detection in image–text data.
- Using semantic concept features to successfully improve the multimodal information retrieval on several datasets.
- Comparing the deep neural network approach and the Support Vector Machine approach for uni- and multimodal information retrieval tasks.
- Automatic construction and utilization of pseudo tags with a multimodal architecture.

1.3 Overview of Thesis

This thesis introduces a novel approach for concept fusion between textual and visual information. Basically, our approach is to process the two modalities individually and, then combined with late fusion. Because of the *semantic gap*, it is important to find an efficient way of analyzing and fusing single-modality information sources. In Chapters 2 and 3, the unimodal information retrieval approach will be described such that the text-based information retrieval will be the main focus in Chapter 2 and the visual-based information retrieval will be the main topic in Chapter 3. Through the contribution of PicSOM research, we developed concept annotation and detection in visual content. In Chapter 3, we will also present how the PicSOM system can deal with visual concepts. Based on the unimodal architectures introduced in Chapters 2 and 3, Chapter 4 presents multimodal information retrieval approaches. In Chapter 4, we also develop and apply a topic model for the automatic generation of pseudo tags. Chapter 5 shows the details of the experimental results with the approaches introduced in the previous chapters. The final conclusions of the thesis are drawn in Chapter 6.

2. Text-Based Information Retrieval

In this chapter, we will discuss the text-based information retrieval and performance measures for information retrieval. Information retrieval means searching the user's retrieval demanded knowledge from any information resources. Earlier, the web data used to be very simple, such as text documents and images. However, there now exist more complicated multimedia resources on the web, and because of their increasing amounts and the diversity of their attributes, the users' information retrieval needs have also become complicated. Consequently, information retrieval earlier only needed to focus on single modality searching, such as text search, but multimodal searching approaches are required nowadays.

So, how should we deal with such diverse information? Let's think about finding the car key in your room. If your room is messy, it is obvious that it is harder to find the car key than in an organized room. The same reasoning can be also applied to the diverse information on the web. Before searching any information, the information should be organized or indexed. In this indexing process, machine learning techniques, such as classification will play a central role. Machine learning techniques are also important in the searching phase, which will be shown in later chapters. In the searching process, it is important to know which pieces of information are useful and which are not. Of course, you know what your car key looks like, but if there are several similar keys in the room, you will need to choose which is the right one.

In text-based general purpose search engines, the early approaches, such as Archie search engine [19], were based on indexing of all terms as keywords and simply matching them with the user's query input. However, in this approach, the keyword indexing does not take into account the semantics of the terms. Therefore, those early search engines of-

ten showed inappropriate and incomplete results. For instance, such an approach cannot retrieve documents which have been written with synonyms of the query inputs.

Text documents consist of sentences and phrases, which in turn consist of words. In many languages, a word is the minimum unit which includes semantic meanings. We can therefore use the words as pre-knowledge for analyzing a dataset. For example, if we know that "cat" belongs to "carnivora", we could make a hierarchical relation between "carnivora" and "cat". However, the semantic relations between words are very hard to convert into numeral directly. We could use language and grammatical knowledge to infer the relations between them, but in practical situations, such as dealing with the documents on the web, the sizes of the datasets are huge and their contents are diverse. Hence, we do not always have the suitable knowledge for representing semantic relations between the words. Also, it would be a very time-consuming task to find such relations between every single pair of words.

One efficient method for extracting textual features is the statistical approach. For decades, many statistical text features have been researched for describing the semantics in documents. For example, there exist 1) ontology-based approaches [29] which use the ontology for query processing in order to extract semantically useful information more effectively than the general keyword-based search from massive data, 2) topic model approaches [11, 47] which also take into account the latent semantics in the documents, 3) explicit semantic analysis approaches [25] which automatically extract semantic concept-based features from large human knowledge repositories such as Wikipedia, and 4) word embedding approaches, such as word2vec [74], GloVe [82], and FastText [56], where text tokens (i.e. words and phrases) are embedded to a vector space.

Also deep neural network approaches have become popular topic in the natural language processing area. Some recent studies of transformer models, such as BERT [20], ALBERT [65], XLNet [17], have shown that the novel models are able to extract semantic relationship between word representations.

2.1 Statistical Frequency-Based Approach

One simple statistical approach for extracting textual features is counting the numbers of words. There exist several common basic frequency counts

for a document. One is the *term frequency* count tf which is the number of occurrences of a word in the document. Even though it is a simple approach, it is very useful and can be extended to more powerful features.

If in some document, the word "science" is used a lot, it is an indication of a high probability of the document being written about science. Therefore, when analyzing the document, the term frequency is an important indicator. For counting the frequency of a word w in a document d_j , the term frequency tf_{w,d_j} can be represented as:

$$tf_{w,d_j} = \frac{n_w}{n_{d_j}}, \quad (2.1)$$

where n_w is the number of occurrences of the word w and n_{d_j} is the total number of words in the document d_j .

For finding correlations between documents, it is also useful to know the occurrences of the words among the documents in the corpus, so called *document frequency*. If we know which words are common in the corpus, we can more easily learn the characteristics of the corpus. For example, if the word "science" appears more frequently than the word "cook", the corpus is more likely to have information on science than cooking. The document frequency df_w represents the frequency of the documents which include the particular word w :

$$df_w = \frac{N_w}{N}, \quad (2.2)$$

where N_w is the number of documents which contain the word w and N is the total number of documents in the corpus. The *document frequency* is usually used as the inverse form in a logarithm scale, called the *inverse document frequency*:

$$idf_w = \log \frac{N}{df_w}. \quad (2.3)$$

The term frequency is good at representing the saliency of the words in a document and the inverse document frequency is good at describing the informativeness of the words among the corpus. It is common to combine the term frequency and the inverse document frequency into one single weight value, called the *tf-idf weighting*:

$$tf-idf(w, d_j) = tf_{w,d_j} \cdot idf_w. \quad (2.4)$$

There exist several variations of weighting schemes for the combination of the term frequency weighting, document frequency weighting and their final normalization, some of which are in Table 2.1 [73].

	term frequency	document frequency	normalization
natural	$tf_{w,d}$	df_w	—
logarithm	$1 + \log(tf_{w,d})$	idf_w	$\cos()$
augmented	$0.5 + \frac{0.5tf_{w,d}}{\max_w tf_{w,d}}$	—	$\frac{1}{\sqrt{r_{w_1}^2 + r_{w_2}^2 + \dots + r_{w_n}^2}}$

Table 2.1. Components of the tf-idf weighting scheme. r_{w_i} is the weight of word w_i . $tf_{w,d}$ is the frequency of the word w in document d and df_w is the number of documents which include the word w . idf_w is the inverse document frequency of word w .

In our experiments, we used the scheme with the combination of logarithmic term frequency, logarithmic document frequency and no normalization:

$$weight(w, d_j) = \begin{cases} (1 + \log(tf_{w,d_j})) \cdot idf_w & \text{if } tf_{w,d_j} >= 1, \\ 0 & \text{if } tf_{w,d_j} = 0. \end{cases} \quad (2.5)$$

Multimedia data, such as images and video clips, usually contain short metadata text information, such as tags. Because each tag can appear only once, the frequency of those tags are either one or zero in this case. Therefore, the term frequencies can be represented as a binary feature vector.

2.2 Topic Model-Based Approaches

The counting of the word frequency is a simple and useful approach, but it usually does not capture enough semantic information for being used for natural language processing problems on its own. However, a statistical approach with the Bayesian probability model can present more semantic information [80]. For example, it can discover the word occurrence pattern and the usage pattern of each word. The statistical co-occurrence patterns then predict the semantic relationships between the words among the documents.

Topic models are an example of methods where the documents are assumed to consist of latent topics. The topics are based on the probability distributions over the words [11]. In order to find the latent topics, we need to think about the distributions of topics among the documents, the word distribution for each topic, and to which topics each word belongs. Probabilistic Latent Semantic Analysis (PLSA) [47] and Latent Dirichlet Allocation (LDA) [11] are common topic models and they have been used

in many applications in statistical natural language processing area.

LDA is a generative probabilistic model of a corpus of documents. The general idea is that the documents in the corpus are represented as random mixtures of the latent topics and the latent topics are characterized by a distribution over the words in a document [11]. The generative process for each document in a corpus, assuming each of them is modeled as a mixture of K latent topics, and each topic k is a multinomial distribution ϕ_k over a vocabulary, is then:

$$\phi_k \sim Dir(\beta). \quad (2.6)$$

Assuming that the topic mixture θ_j can be represented as the Dirichlet distribution over the parameter α for each document, then:

$$\theta_j \sim Dir(\alpha), \quad (2.7)$$

where $Dir()$ is the Dirichlet distribution, and α and β are the hyper parameters. The Dirichlet distribution is a multivariate generalization of the *Beta* distribution and it is the conjugate prior of the categorical distribution and the multinomial distribution. The multinomial distribution is the probability distribution of choosing one of a fixed number of K categories for fixed number of n trials. If only one trial ($n = 1$) is attempted for a fixed number of K categories, the distribution becomes the categorical distribution. When $K = 2$ and $n = 1$, the multinomial distribution is the well-known Bernoulli distribution. The Dirichlet distribution of order $K \geq 2$ with parameters $\alpha_1, \dots, \alpha_K > 0$ is

$$Dir(x_1, \dots, x_K; \alpha_1, \dots, \alpha_K) = \frac{1}{B(\alpha)} \prod_{i=1}^K x_i^{\alpha_i - 1}, \quad (2.8)$$

where $B(\alpha)$ is the multivariate beta function,

$$B(\alpha) = \frac{\prod_{i=1}^K \Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^K \alpha_i)}, \quad (2.9)$$

and $\Gamma(\alpha)$ is the gamma function.

$$\Gamma(\alpha) = (\alpha - 1)! \quad (2.10)$$

For each i^{th} word w_{ij} in document j and topic $z_{ij} = k$, the assignments can be modeled as,

$$z_{ij} \sim Multinomial(\theta_j), \quad (2.11)$$

and

$$w_{ij} \sim Multinomial(\phi_{z_{ij}}), \quad (2.12)$$

where θ_j is the distribution of topics in document j and $\phi_{z_{ij}}$ is distribution of topic z_{ij} . Because LDA is based on the Bayesian inference, it requires estimation of the underlying distributions. Therefore, many optimization methods for learning LDA have been proposed. The Markov Chain Monte Carlo (MCMC) method is a common inference method for latent topic methods [11] and Gibbs sampling is the most widely used method for LDA inference approximation [38].

In Gibbs sampling, the mixture θ and the topics ϕ are integrated out by sampling the latent variable z . This is known as collapsing. In this case, a word–topic count matrix C_{word} , document–topic count matrix C_{doc} and topic count vector C_{topic} should be maintained. The conditional probability of the latent variable z_{ij} which is assigned to topic k can be written as

$$P(z_{ij} = k | z^{-ij}, x^{-z_{ij}}, x_{ij} = w, \alpha, \beta) \propto \frac{C_{wk}^{-ij} + \beta}{C_k^{-ij} + W\beta} (C_k^{-ij} + \alpha), \quad (2.13)$$

where C_{wk} is the number of times word w is assigned to topic k and C_k^{-ij} is the number of topics k assigned to the i^{th} word in document d_j and $-ij$ means exclude the parameter of the i^{th} word in the j^{th} document.

Efficient and novel methods for speeding up the inference process have been researched in [69]. However, they are not covered in this dissertation because they are far off the thesis' main topic. We have used the LDA model for pseudo tag generation. The technical details and experimental results with that approach are in Sections 4.3 and 5.4 in Publication VII.

2.3 Text Similarity Metrics

In the previous sections, we have discussed the semantics of text documents. However, how can one measure the relative similarity between words or latent topics? When applying statistical methods to documents, the text features are commonly represented as vectors. Hence, one can associate the measuring of word similarity with a vector distance metric. For example, if in a vector space, the Euclidian distance of two vectors is zero, then those two vectors are identical. On the other hand, if the Euclidian distance of two vectors is large, then the difference between the two vectors and the documents are large. That is, if the distance between two vectors is close, those two vectors are "similar", whereas if the distance is large, those two vectors are "dissimilar". In that way, we could use the distance between two vectors as the similarity metric. This method is well-known as the vector space model (VSM) [93].

Several measures have been proposed for the vector similarity. Let \mathbf{X} and \mathbf{Y} be n -dimensional feature vectors. Some common distance metrics can be presented as follows [110]:

1. City block distance or L_1 measure:

$$D_{CB}(\mathbf{X}, \mathbf{Y}) = \sum_{i=1}^n |X_i - Y_i| \quad (2.14)$$

2. Euclidean distance or L_2 measure:

$$D_{Euc}(\mathbf{X}, \mathbf{Y}) = \sqrt{\sum_{i=1}^n |X_i - Y_i|^2} \quad (2.15)$$

3. Cosine Similarity:

$$D_{cos}(\mathbf{X}, \mathbf{Y}) = \frac{\sum_{i=1}^n X_i Y_i}{\sqrt{\sum_{i=1}^n X_i^2} \sqrt{\sum_{i=1}^n Y_i^2}} \quad (2.16)$$

4. Kullback–Leibler (KL) divergence:

$$D_{KL}(\mathbf{X}, \mathbf{Y}) = \sum_{i=1}^n X_i \ln \frac{X_i}{Y_i} \quad (2.17)$$

The city block distance is the simplest distance calculation, the Euclidean distance is the most common distance metrics, the cosine similarity measure is based on the inner products of the two vectors, and the KL divergence is based on the relative entropy [63].

In VSM, the cosine similarity measure is usually used because the outcome of the cosine similarity between two non-negative vectors is always in the $[0, 1]$ range. Therefore, it is efficient to use it in any high-dimensional positive space, such as text feature vector space. The cosine similarity and KL divergence are also used for comparing probability distributions. Detailed explanations of other distance and similarity measures can be found in [14]. In our experiments, we applied the cosine similarity measure in several semantic similarity measuring tasks in image and multi-modal retrieval. Those methods are introduced in the next chapter and also in Publication I, Publication VI, and Publication VII.

2.4 Performance Measures

Once we have the retrieval results, how do we measure and analyse the performance of the features and the indexing techniques? There exist

	Predicted Positive	Predicted Negative
Ground Truth Positive	True Positive (TP)	False Negative (FN)
Ground Truth Negative	False Positive (FP)	True Negative (TN)

Figure 2.1. A confusion matrix.

several common metrics in modern information retrieval: precision and recall, mean average precision (MAP), and the average area under the ROC curve (aAUC). In this thesis, we also studied the multi-label information retrieval. For the multi-label task, we have applied subset accuracy (ssACC).

2.4.1 Precision and Recall

Precision is the fraction of the retrieved items that are relevant to the user's information needs. On the other hand, recall is the fraction of correctly retrieved relevant items from their total count. Let the number of true positives be denoted as TP , true negatives as TN , false positives as FP , and false negative as FN . These entities are shown in the form of a confusion matrix in Figure 2.1. Precision is then calculated as:

$$precision = \frac{TP}{TP + FP} \quad (2.18)$$

and recall as:

$$recall = \frac{TP}{TP + FN}. \quad (2.19)$$

It is good to have large scores in both precision and recall for constructing working applications. However, it is well known that the precision and recall are usually in a trade-off situation [12]: if one increases, the other is known to decrease. Figure 2.2 shows an example of the precision-recall trade-off.

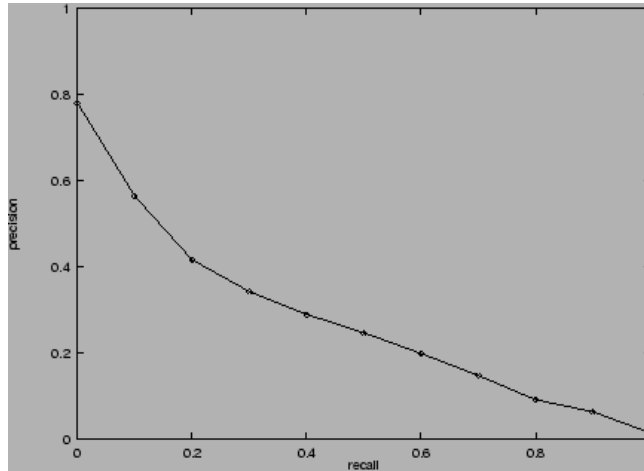


Figure 2.2. An example of the trade-off between precision and recall from [41]. The plot shown the results of retrieval systems tested in a TREC evaluation 1999.

2.4.2 Mean Average Precision

The average precision is a popular measure which takes into account the relationship between precision and recall [129]. Let us denote precision as a function of recall $p(r)$, so that recall r is a continuous variable between $[0, 1]$, then the average precision AP is represented as the definite integral:

$$AP = \int_0^1 p(r)dr. \quad (2.20)$$

In practice, information retrieval usually deals with discrete objects, such as documents, and then:

$$AP = \sum_{i=1}^n p(i)\Delta r(i), \quad (2.21)$$

where i is the rank in the sequence of retrieved documents, n is the number of retrieved documents, $p(i)$ is the precision at cut-off i in the list, and $\Delta r(i)$ is the change in recall from item $i - 1$ to i .

Average precision only takes into account one query at a time. Therefore, we usually use the mean average precision (MAP) which expresses the mean of the average precision of each query in a query set. In practice:

$$MAP = \frac{\sum_{q=1}^Q AP(q)}{Q}, \quad (2.22)$$

where Q is the number of queries.

In a realtime situation, such as a web search, the users are usually interested in the top one or two pages of retrieval results. Therefore, it is

in practice common to prefer precision than recall, Precision can be measured as the precision of top K number of retrieved results among the whole dataset, and be used as the metric individually (Prec@ K). In this thesis, MAP and Prec@ K are mainly used as the evaluation metrics. In the TRECVID workshops, we also used the mean extended inferred average precision (MXIAP) [122], which is an average precision-based method and used as the standard evaluation method in these workshops. MXIAP assumes that the annotation of the test set from a stratified random sampling can be incomplete, and defines the average precision as the outcome of a random experiment.

2.4.3 Average Area under the ROC curve

The receiver operating characteristic curve (ROC) curve is created by plotting the recall against the fall-out. The fall-out is the false positive rate and it represents the probability of false alarm. The area under the ROC curve (AUC) is equal to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one [27]. The AUC varies between 0 and 1, and if the model can classify perfectly, the AUC value becomes 1, whereas if the model classifies at random, the value is 0.5.

2.4.4 Subset Accuracy

The subset accuracy is a common measure for evaluating the performance of multi-label classification [67]. Let $\{(x_n, y_n)\}_{n=1}^N$ be a multi-label dataset with ground truth labels, and $\{\hat{y}_n\}_{n=1}^N$ be the multi-label predictions made by a classifier. The subset accuracy generalizes the conventional multiclass accuracy notion:

$$ssACC = \frac{1}{N} \sum_{n=1}^N \mathbb{I}[y_n = \hat{y}_n], \quad (2.23)$$

where $\mathbb{I}[\cdot]$ is the indicator function. In computing the subset accuracy, a predicted subset is considered correct only when it matches the true subset exactly [67].

3. Visual Information Retrieval

In this chapter, we will discuss information retrieval of visual content. Retrieving visual information with one's own eyes is not at all difficult task for humans. For example, when humans see a photograph of a car, even a 3-year-old child can recognize a car in the picture immediately. However, is it possible that also a newborn baby recognizes a car in a photo? And how many people can recognize and tell the correct names of the species just by looking at pictures of unfamiliar plants? It is obvious that nobody can name or recognize all of them correctly. Of course, we can recognize those objects as visually similar, such as a "tram" can be recognized as a "train", or higher semantic concepts, such as an "American Shorthair" can be recognized as just a "cat". But these recognitions are based on what we have learned, they do not just pop up from nowhere. In order to recognize something new, we have to learn it first. The same can be said about computer perception.

However, the perception of an image is very different between human and computer vision. Human can manage various types of information (e.g. color, texture, shape, etc.) directly from the image, but a computer vision system needs to first convert the information to numerical features. More specifically, in order to retrieve visual information, we have to extract visual features.

3.1 Feature Extraction

Features express the properties of objects as digitized values. For example, we can represent an image as a vector of the intensity values of each pixel. For an information retrieval task in any modality, the feature extraction step is one of the most important processes for getting acceptable results. However, the feature extraction is a very hard task to do fully

automatically. Because the characteristics of the information modalities are different, features for each modality are different even though the information retrieval approaches would otherwise be similar.

Even in a single modality space, feature extraction is often a very hard task. As mentioned in Section 1.1, one of the reasons is the "*semantic gap*" [89]. Seemingly, humans can perceive the semantic contents in the images directly, whereas the computer can only perceive the images through a numerical representation. That is, there is a "gap" between the human and the computer perception of the image. Hence, in order to achieve automatic classification close to that of humans, we need to bridge the semantic gap between the human perception and the numerical representation for the computers step by step.

In image retrieval studies, there are two types of features that are commonly used: low-level features and high-level features. According to [23], the low-level features are usually categorized as a sensory input data which does not represent any semantics but only a statistical description of images. The high-level features represent at least some semantic information, such as class categories. In the following subsections, we will introduce the low- and high-level image features used in the experiments in this thesis.

3.1.1 Low-level Features

Low-level features or descriptors can be directly converted from raw sensory data of images or videos. The low-level features take into account the pixel-wise colors, shape, texture, etc., quantized or digitized to numerical vectors. The pixel-wise information is not semantically useful to humans, and it only represents one or few aspects of the visual scene, such as intensity or color differences, as a scalar variable. It is better to have more information as a vector or matrix, but it is not practical or useful to store all pixel-wise data for a huge image data set. This is because the dimensionality of the feature vectors becomes high. The high dimensionality leads to increase in computational costs and the amount of noisy or otherwise useless data. This is called "*curse of dimensionality*" and it is very problematic for experiments which process large amount of data, such as multimedia dataset. Therefore, descriptors which describe and store the information in the most efficient way are needed.

Scale Invariant Feature Transform

Scale Invariant Feature Transform (SIFT) [71] is one of the most famous feature descriptors invariant to scale, rotation, position, etc. SIFT feature extraction consists of the following four phases as shown in Figure 3.1:

1. Detection of scale-space extrema: Use Difference of Gaussian (DoG) to find keypoint candidates which are invariant to scale and orientation.
2. Calculation of localized keypoints: At each the keypoint candidates location, a detailed model is fit to determine location and scale.
3. Orientation assignment: Because of the local image gradient directions are assigned to all keypoints, one can remove the effect from rotation and scales.
4. Creation of the keypoint descriptors: Use orientation histograms to create the keypoint descriptors.

Speed-Up Robust Features (SURF) [10], GIST [79], and Histogram of Oriented Gradients (HOG) [18] descriptors are also commonly used in computer vision studies. Those descriptors are very powerful tools for matching tasks, such as fingerprint recognition [7, 22, 37]. However, when we focus on object and scene classification in images, the keypoint-wise similarity comparison is not a suitable approach. Each keypoint contains some kind of shape information, but those do not express which parts of the objects they belong to. Of course, increasing the number of keypoints can improve the recognition performance, but it does not solve the semantic gap between the keypoints and the objects and it also leads to a higher computational cost.

Bag of Visual Keywords

One possible solution to the drawback of the keypoint matching approach is the bag of visual keywords (BoV). In text information retrieval, the words themselves represent the semantics like the objects in an image. If we could achieve the same for image features, it would be easier to bridge the semantic gap and more efficient search could be obtained. The BoV approach is adapting the bag of words representation used for text categorization [96] by constructing histograms of the frequency of particular

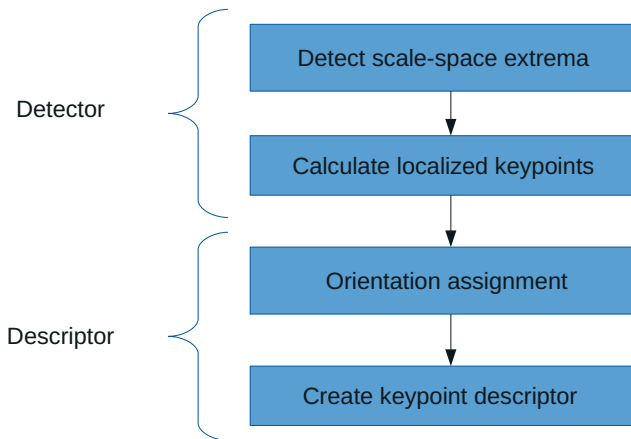


Figure 3.1. The four phases of SIFT feature extraction. Detector detects most salient regions, called keypoints, which describe image. Descriptor extracts and quantifies the keypoints and their surroundings to a feature vector.

patterns in an image. The main pre-processing steps (Figure 3.2) are:

1. Extraction of image descriptors: Extract feature descriptors, such as SIFT for the all images in the dataset.
2. Vector quantization: Cluster the descriptors with a vector quantization algorithm, such as k-means [96].
3. Construction of histograms: For each image, build a histogram of the vector quantized keypoint descriptors with bins based on the cluster.
4. Use those histograms as feature vectors and classify images by applying machine learning methods, such as SVM.

The k-means algorithm is often used as the bin quantization algorithm, but it is a hard assignment clustering and lacks the flexibility of decision boundaries. The k-means algorithm also does not take into account the spatial layout information of the features. In order to overcome the problem, the spatial pyramid matching has been widely used [66]. This approach constructs pyramids of several levels with a grid subdivision of

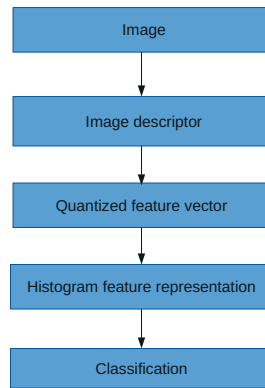


Figure 3.2. The main steps of the Bag of Visual words (BoV) approach. After extracting the feature vectors, find the centroids of ‘words’ with semantically meaningful patterns in the images by clustering. Then, build the bins of each ‘word’ based on its frequency in each image.

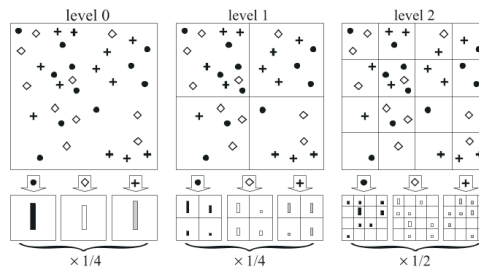


Figure 3.3. Toy example of constructing a three-level pyramid [66]. The image has three feature types, indicated by circles, diamonds, and crosses. At the top, the image in three different levels of resolution to be subdivided. Next, for each level of resolution and each channel, the features that fall in each spatial bin are counted. Then, calculate weights for each histogram. Please refer to [66] for more details.

the image (Figure 3.3) and sparse coding which assigns a local descriptor to several different keywords [81]. In order to obtain a better performance with the bag of visual words representation, we usually need to have a large number of visual words [120]. It means that we have a high computational cost because of the curse of dimensionality.

To improve the accuracy of classification, it would be better to train the visual words dictionary with a specific and precise training dataset for the current purpose, rather than using a generic visual words dictionary. However, dealing with high-dimensional data is time consuming and a new visual word dictionary for a particular scenario cannot always be trained. Several studies have tried to make the BoV approach more efficient, such as Super Vectors [128], Gaussian Mixture Model (GMM) [83],

and Fisher Vectors [84]. These approaches use the statistical values such as means and variance instead of applying the original features. Thus, their dimensionality are relatively small, and their computations are efficient. In most of cases, their results outperform the original BoV results.

3.1.2 Deep Convolutional Neural Networks

Recently, deep learning methods such as Deep Convolutional Neural Network (DCNN) [62] have shown great success on many image recognition tasks and they have taken over the old hand-crafted local feature descriptors. Deep learning is an extension of the feed-forward neural network which is one of the neural network machine learning approaches. The neural network is inspired by and mimics the human brain system. Mimicing of the human brain, the neural network usually consists of nodes, so-called *neurons*. Like the human brain, the neurons are connected to each other and the input signals travel through several nodes. In each neuron, the input signals are weighted and summed and then pass to the function, called *activation function*. The activation function produces the output signal. For example, in a simple binary output case, if the neuron is activated outputs 1, otherwise it outputs 0.

The feed forward neural network is the most fundamental type of neural networks (Figure 3.4). It is directed and the neurons appear in three layers: the input layer, hidden layer, and output layer. The neurons in the input layer take input signals and pass the output to the neurons in the hidden layer. The hidden layer neurons take those output signals from the input neurons as the input signals and then pass the output signal to the output layer. Finally, the output layer converts the outputs from the hidden layer to final output signals. If the network has more than two hidden layers, it is called a deep learning network. Figure 3.5 depicts an example of a Deep Convolutional Neural Network (DCNN).

In recent works on deep neural network architectures for image classification [61, 62, 42], one can notice that the number of layers in the state of the art methods is increasing every year. However, increasing of the layer count leads to a large amount of hidden parameters. This easily leads to overfitting and high computational cost. However, DCNNs partially overcome those drawbacks with their convolutional architecture, which changes from a fully-connected network to shared weights. Basically, DCNN consists of multiple convolutional modules and fully connected layers for a final classification. A basic architecture of a convolutional module

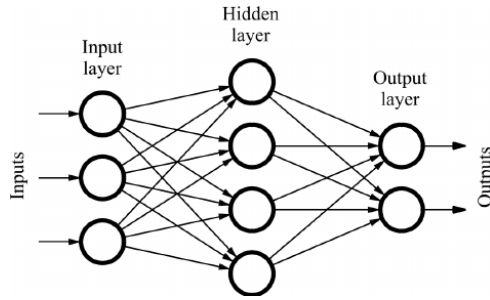


Figure 3.4. The architecture of a Feed Forward Network [87]. The left most layer is the input layer which feeds the input data to the hidden layer. The middle layer is the hidden layer which outputs the weighted sum of the data. The right most layer is the output layer which transforms the outputs from the hidden layer to suitable outputs for the task. In a binary classification task, the output will be 0 or 1.

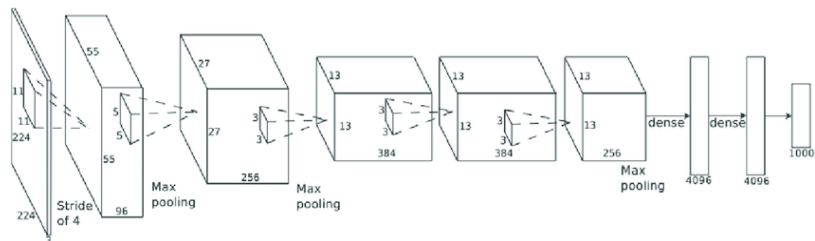


Figure 3.5. The architecture of a Deep convolutional neural network (DCNN) [102]. DCNN converts the low-level features to the high-level features. This figure shows the dimensionality of outputs are relatively smaller than inputs.

is as follows:

1. Convolutional layer: This layer applies multiple trainable filters to input images and performs convolutions.
2. Activation function: Same activation functions which are used in any neural network. Recently, the Rectified Linear Unit (ReLU) [76] is the most commonly used activation function.
3. Pooling layer: Pooling layer performs downsampling to reduce the dimensionality and to extract high-level features from the output of the activation function.

DCNN actually transforms dense low-level local descriptors to sparse high-level semantics approximation. Layer by layer, it calculates the optimal network architecture based on the activations of the previous layer [61]. The activation outputs on the fully connected layers can then be used as

visual features to determine the image contents.

Our research group has used and compared the performance between the GoogLeNet activation features and the BoV features in different classification tasks [61]. We have used the DCNN features for several experiments. Technical details and experiments can be found in Publication III, Publication V, and Publication VI.

3.1.3 Semantic Concepts for Visual Information Retrieval

Semantic concept detectors are based on a set of machine learning classifiers trained with a human annotated dataset, such as images and video clips. In order to train the detectors, the low-level features extracted from the images or the video clips are used as the input data. Then, the semantic concept classifiers can be trained with those training data and labels as explained in Figure 3.6.

In our research group, we have researched the detection of visual concepts from multimedia resources, i.e. images and videos. For example, in our participation in the TRECVID workshops [97, 98, 55, 54, 53, 114], in Publication I, Publication IV, Publication VI, and Publication VII. We have also used the visual concept vectors to improve the performance of a visual content-based information retrieval architecture [115].

3.2 Classification

Unlike low-level features, high-level features, such as semantic visual concepts depend on the knowledge about the semantic content of image, such as what objects are in it. For this purpose we will need a classifier which is trained with supervised learning. The classifier is also useful for retrieving the user's required information because classifying the image with the same concepts means that the classifier can cluster the images which have the same concept as the query image. That is, we can find images which are similar to the query by using the classifier. In this section, we will describe the machine learning based classifiers which we used in experiments.

3.2.1 Support Vector Machine

The Support Vector Machine (SVM) has been a great success in many classification tasks in the last decade [100, 34, 15]. First of all, let we con-

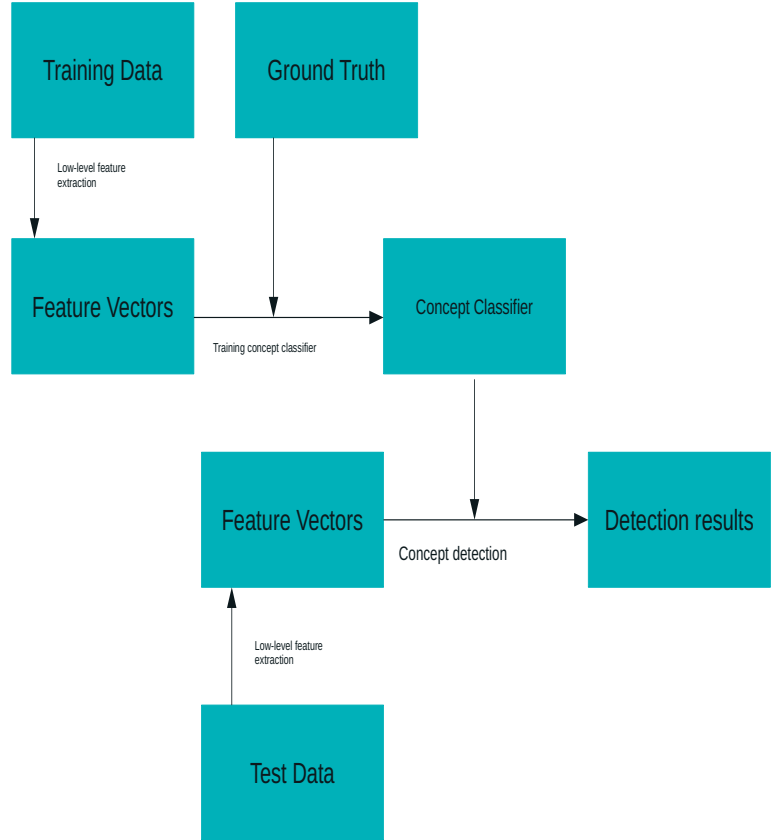


Figure 3.6. The architecture of semantic concept detection. The architecture is similar to the pipeline of a regular image classifier. First, the semantic concept classifier is trained by using the extracted low-level features of the training dataset. Then, the semantic concepts of the test data are detected by using the trained detector.

consider a two-class linear classification task. Let the training set X consist of N number of feature vectors \mathbf{x}_n . These feature vectors belong to either class c_1 or c_2 . That is, each input point has corresponding target values $X = \{(\mathbf{x}_1, t_1), \dots, (\mathbf{x}_n, t_n)\}$, where target $t_n \in \{1, -1\}$, and $t = 1$ means the point belongs to c_1 whereas $t = -1$ means the point belongs to c_2 . The linear model for the decision boundary is then formed as:

$$y(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + w_0. \quad (3.1)$$

This means that there is a function $y(\mathbf{x})$ with parameters \mathbf{w} and w_0 which satisfies $y(\mathbf{x}_n) > 0$ for all training vectors with $t_n = +1$ and $y(\mathbf{x}_n) < 0$ for those with $t_n = -1$. Consequently, $t_n y(\mathbf{x}_n) > 0$ for all training vectors.

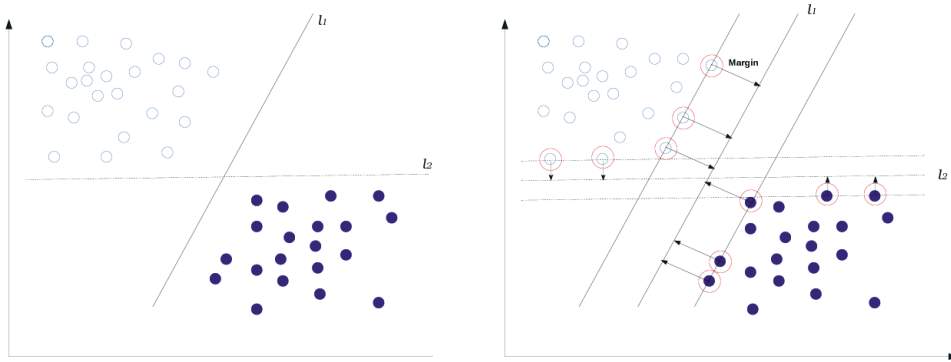


Figure 3.7. **Left:** Two hyperplanes which correctly classify the training vectors. **Right:** Maximizing the margin in order to optimize the decision boundary.

In Figure 3.7 on the left, there are two hyperplanes which correctly classify the linearly separable training vectors. However, those hyperplanes are not guaranteed to separate correctly the test vectors or any practical data set. In order to optimize the hyperplane from the training data, maximizing the perpendicular distance (margin) between the hyperplane and the closest point of the data set is one possible solution as seen in Figure 3.7 on the right.

Solving the SVM

The perpendicular distance of the closest sample points x_n from a hyperplane can be defined as

$$t_n y(x_n) / \|\mathbf{w}\| = t_n (\mathbf{w}^\top \mathbf{x}_n + w_0) / \|\mathbf{w}\|, \tag{3.2}$$

assuming that the hyperplane correctly classifies all the training vectors. The points in the hyperplane satisfy $y(x) = 0$ and the size of the margin is $1/\|\mathbf{w}\|$. In order to maximize the margin, we would like to optimize the parameters \mathbf{w} and w_0 ,

$$\arg \max_{\mathbf{w}, w_0} \{1/\|\mathbf{w}\| \min_n [t_n (\mathbf{w}^\top \mathbf{x}_n + w_0)]\}. \tag{3.3}$$

For the proper form of (\mathbf{w}, w_0) , one needs to have $t_n (y(x_n)) = 1$ for all the closest points to the hyperplane. Those closest points are called the support vectors. Then, for all the data points, it holds:

$$t_n (\mathbf{w}^\top \mathbf{x}_n + w_0) \geq 1. \tag{3.4}$$

While satisfying these constraints, the optimization problem becomes just to find the parameters (\mathbf{w}, w_0) which maximize $1/\|\mathbf{w}\|$. This is equivalent

to minimizing $\|\mathbf{w}\|^2$. The optimization can be solved by using the Lagrange multipliers $\lambda_n \geq 0$,

$$L(\mathbf{w}, w_0, \boldsymbol{\lambda}) = 1/2\|\mathbf{w}\|^2 - \sum_{n=1}^N \lambda_n t_n (\mathbf{w}^\top \mathbf{x}_n + w_0) - 1, \quad (3.5)$$

where $\boldsymbol{\lambda} = \{\lambda_1, \dots, \lambda_N\}$. First solving the derivatives with respect to \mathbf{w} and w_0 and setting them to be equal to zero, we obtain two conditions:

$$\mathbf{w} = \sum_{n=1}^N \lambda_n t_n \mathbf{x}_n \quad (3.6)$$

and

$$\sum_{n=1}^N \lambda_n t_n = 0. \quad (3.7)$$

Then, equation (3.5) can be solved as the dual representation:

$$\max_{\boldsymbol{\lambda}} \left\{ \sum_{n=1}^N \lambda_n - 1/2 \sum_{n=1}^N \sum_{m=1}^N \lambda_n \lambda_m t_n t_m \mathbf{x}_n^\top \mathbf{x}_m \right\}. \quad (3.8)$$

We then end with a quadratic programming problem for solving $\boldsymbol{\lambda}$. Further details of the solution can be found in [81]. After solving the optimal $\boldsymbol{\lambda}$ and \mathbf{w} , we can easily find w_0 for all the data points:

$$w_0 = t_n - \mathbf{w}^\top \mathbf{x}_n. \quad (3.9)$$

In order to get a stable solution, we can average it for all support vectors:

$$W_0 = 1/N_S \sum_{n=1}^{N_S} (t_n - \sum_{n=1}^{N_S} \sum_{m=1}^{N_S} \lambda_m t_m \mathbf{x}_n^\top \mathbf{x}_m), \quad (3.10)$$

where N_S is the number of support vectors and the indexation refers to the support vectors.

The training set is not usually linearly separable in a practical situation, and it is impossible to classify the data correctly with a linear classifier. In many cases, two classes are overlapping with each other. In order to solve this problem, slack variables $\xi_n \geq 0$ have been used [88]. If the sample vector point is on the correct side, ξ_n is set to 0 and otherwise to $|t_n - y_n(x_n)|$. Thus, we should minimize

$$1/2\|\mathbf{w}\|^2 + C \sum_{n=1}^N \xi_n, \quad (3.11)$$

where $C > 0$ is a parameter which optimizes the trade-off between the training error and the model complexity. The Lagrangian dual form is now:

$$\max_{\boldsymbol{\lambda}} \left\{ \sum_{n=1}^N \lambda_n + C \sum_{n=1}^N \xi_n - 1/2 \sum_{n=1}^N \sum_{m=1}^N \lambda_n \lambda_m t_n t_m \mathbf{x}_n^\top \mathbf{x}_m - \sum_{n=1}^N \lambda_n \xi_n \right\}, \quad (3.12)$$

the constraints for equation (3.12) are:

$$\begin{cases} 0 \leq \lambda_n \leq C, \\ C \sum_{n=1}^N t_n \lambda_n = 0 \end{cases} \quad (3.13)$$

From equation (3.1) and equation (3.6), the linear SVM can be represented as:

$$y(\mathbf{x}) = \sum_{n=1}^N \lambda_n t_n \mathbf{x}_n^\top \mathbf{x} + w_0. \quad (3.14)$$

Non-linear SVM kernels

Instead of calculating the inner products of equation (3.14) directly, it is more popular to use a non-linear kernel function that allows separation of data that is linearly unseparable. Thus, the inner product in the discriminative function of linear SVM can be replaced with a kernel function $K(\cdot, \cdot)$:

$$y(\mathbf{x}) = \sum_{n=1}^N \lambda_n t_n K(\mathbf{x}_n, \mathbf{x}) + w_0. \quad (3.15)$$

In order to fulfill the condition of the optimization of equation (3.12), the kernel function must be a positive definite function.

There are two types of kernels which have commonly been used for computer vision research: exponential kernels and additive kernels. The exponential kernels are based on the exponential function:

$$K(\mathbf{x}, \mathbf{y}) = \exp(-\gamma K'(\mathbf{x}, \mathbf{y})), \quad (3.16)$$

where $\gamma \geq 0$. The most popular function is the Gaussian or radial basis function:

$$K(\mathbf{x}, \mathbf{y}) = \exp(-\gamma \|\mathbf{x} - \mathbf{y}\|^2). \quad (3.17)$$

Additive and Power kernels

The non-linear kernel SVM is computationally expensive for real-time application because of its complexity $O(dNs)$, where Ns is the number of support vectors and d is the number of dimensions. One solution for this problem is to approximate the non-linear kernel with linear approximations [72], additive kernel PCA [85], or with homogeneous kernel maps [113]. Additive SVM kernels have the form:

$$K(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^d k(x_i, y_i), \quad (3.18)$$

where $k(x_i, y_i)$ is a scalar function. Such kernels can thus be presented as a sum of one-dimensional functions. Some commonly used one-dimensional functions for $k(x, y)$ are [113]:

- **Intersection:** $k(x, y) = \min(x, y)$
- χ^2 : $k(x, y) = -(x - y)^2 / (x + y)$
- **Bhattacharyya:** $k(x, y) = \sqrt{x, y}$
- **Jensen-Shannon:** $k(x, y) = x/2 \log_2((x + y)/x) + z/2 \log_2((x + y)/z)$
- **Power mean:** $k(x, y) = ((x^p + y^p)/2)^{1/p}$

The intersection and Bhattacharyya kernels can be presented by the power mean kernel with setting $p = -\infty$, $p = 1$, and $p = 0$, respectively.

The homogeneous kernel map is an approach to find a mapping function which projects the non-linear kernel approximately to a linear space and can be computed efficiently [113]. The homogeneous kernel map of order n is a $(2n + 1)$ -dimensional linear approximation of the corresponding non-linear kernel. Hence, the d -dimensional feature vector can be encoded as a $d(2n + 1)$ -dimensional vector and applied to the linear SVM classifier. The complexity of evaluating the classifier is now down to $O(d)$. In Publication III, we also applied the homogeneous kernel map and the power mean SVM [118], which is an alternative method to approximate additive kernels by calculating the gradient, which is the computational bottleneck in the coordinate descent algorithm, by using second-order polynomial regression of scalar functions. Power mean SVM provides efficient training and classification with the power mean kernels for any $p < 0$, as well as with other additive kernels. Now, if we consider the dual SVM problem of equation (3.8) for the power mean kernel, it is presented as:

$$\max_{\lambda} \left\{ \sum_{n=1}^N \lambda_n - 1/2 \sum_{n=1}^N \sum_{m=1}^N \lambda_n \lambda_m t_n t_m K_{pm}(\mathbf{x}_n, \mathbf{x}_m) \right\}. \quad (3.19)$$

The corresponding decision boundary is:

$$\mathbf{w} = \sum_{m=1}^M \lambda_m t_m \Psi(\mathbf{x}_m), \quad (3.20)$$

where $\Psi(\mathbf{x})$ is the feature mapping function. In the coordinate descent al-

gorithm, each variable λ_n is updated separating while keeping the others fixed. Therefore, the gradient G_n with respect to λ_n is:

$$\begin{aligned}
G_n &= t_n \mathbf{w}^\top \Psi(\mathbf{x}_n) - 1 \\
&= t_n \sum_{m=1}^M \lambda_m t_m K_{pm}(\mathbf{x}_n, \mathbf{x}_m) - 1 \\
&= t_n g(\mathbf{x}_n) - 1 \\
&= t_n \sum_{o=1}^O g_o(x_{n,o}) - 1,
\end{aligned} \tag{3.21}$$

where we have a scalar function

$$g_o(x) = \sum_{m=1}^M \lambda_m t_m k_{pm}(x, x_{m,o}) \tag{3.22}$$

and $x_{n,o}$ is the o^{th} component of the vector \mathbf{x}_n . Therefore, it is sufficient to approximate the scalar function $g_o(x)$ for the gradient approximation. In [118], the approximation is done by using a second-order polynomial with parameters $\mathbf{a}_m = \{a_{m,0}, a_{m,1}, a_{m,2}\}$:

$$g_o(x) \approx \sum_{q=0}^2 a_{m,q} (\ln(x + 0.05))^q. \tag{3.23}$$

Instead of using x , $\ln(x + 0.05)$ is used because it gives better approximation results. For classifying a new example \mathbf{x} with power means SVM, it requires to evaluate:

$$y(\mathbf{x}) = \sum_{m=1}^M \sum_{q=0}^2 a_{m,q} (\ln(x + 0.05))^q. \tag{3.24}$$

The classification complexity of this approach is $O(d)$ and it provides the capability of faster training than non-linear SVM.

We have used linear, linear approximation and non-linear SVM in almost all of our experiments and all publications included in this thesis. In our studies the linear approximations of nonlinear kernels usually showed negligible loss of accuracy compared to non-linear kernels in the visual content-based information retrieval task [99].

3.2.2 Single- and Multi-label Classification

In Publication I, we also used the Deep Boltzmann Machine (DBM) approach to be introduced and discussed in details in the next chapter for the multi-label classification task. Some of the single-label classification task is just to find whether a labeled object or other specific content is included

in the target media or not. On the other hand, in the multi-label classification task, each target media can be assigned multiple labels. That is, there is no limitation of the classes each instance can be assigned to.

There exist two main approaches for solving this multi-label classification problem [112]. One approach is transforming the multi-label problem into a set of binary classification problems. The other approach is to adapt the algorithms to directly perform multi-label classification. Our approach is based on binary classification and the details of the experiments will be shown in Section 5.2.

4. Multimodal Information Retrieval

Recently, multimedia information retrieval has been forced to deal with the massive variety of data resources. Multimodal information retrieval which can handle the cross-modal searching is very promising for this area. Besides, in recent multimodal research [35, 106], the multimodal information approaches outperformed the unimodal approaches. However, multimodal information retrieval has a several drawbacks, such as it requires the cross-modal queries and need to handle the relatively large features or train several different models. It is also that there exists the possibility to harm the information retrieval results because of the fusing poor unimodal features or results. In this chapter, we will study the multimodal information retrieval problem. Multimodal information consist of unimodal information sources, hence when we analyse multimodal information, we have to efficiently combine the semantics from the unimodal information sources at some point. The most common approach is the fusion scheme [104]. Because of Jensen's inequality fusion schemes tend to show better results than individual method [13]. There exist two main approaches: early fusion and late fusion. Early fusion is a fusion scheme which integrates the unimodal features before any machine learning process. On the other hand, late fusion is a scheme which combines unimodal outputs to multimodal results.

Multimodal Deep Boltzmann Machine (DBM) proposed by Srivastava et al [106], which combines the unimodal results with an extra hidden layer, has shown great success in multimodal information retrieval. The model can be considered as a late fusion approach. Multimodal DBM will be described in detail in Section 4.2.

We will also consider the approach of transferring the extracted information to other modalities. For example, pseudo tag generation, which we have proposed, generates tags based on the results of image information

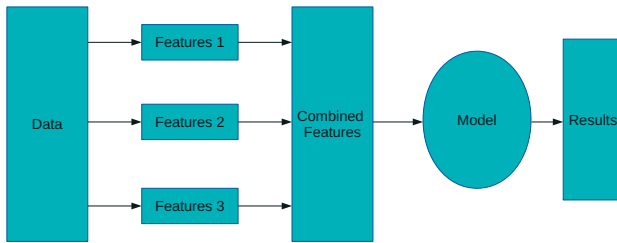


Figure 4.1. The architecture of early fusion information retrieval. An early fusion approach basically extracts several different features from the datasets and combines them before training the model.

retrieval. The use of rich extra new tags can improve the textual search results. The detailed experiments of this approach are in Publication VII. In the semantic concept vector approach, we use the semantic concept detection results on images as an extra feature for the text search task. These approaches can be categorized as early fusion approaches.

4.1 Traditional Fusion-based Approaches

In recent information retrieval approaches, even in the unimodal case, multiple different features are typically used to describe the information contents. In order to retrieve the data objects by using many different features, the information from the features needs to be somehow combined. The fusion approach is a fundamental solution to this task.

As already mentioned in this chapter, the fusion approaches can be divided in two main branches: *early fusion* and *late fusion*, which are described in Figures 4.1 and 4.2, respectively. In early fusion in Figure 4.1, the multiple different features are concatenated to a single long feature vector. A single model is their trained with this combined feature [104]. On the other hand, the late fusion approach in Figure 4.2 trains a separate model for each different feature vector, and then fuses the outputs of all the models to produce the final output. The advantage of early fusion is that it is only required to train a single model, but usually the concatenated feature vector tends to have a high dimensionality. Therefore, an

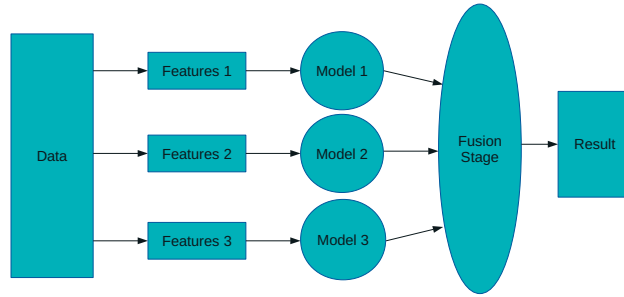


Figure 4.2. The architecture of late fusion information retrieval. A late fusion approach trains several different machine learning models and fuses their results to produce the final results.

early fusion scheme is more likely to be affected by the curse of dimensionality. Because of the features are from different modalities, some extra preprocessing is required. For example, the features may be scaled differently or require different metrics so that we need to somehow normalize or project them to the same space before concatenating them to a single feature vector. The Canonical Correlation Analysis (CCA) [49, 50, 35] is one of the major early fusion approaches.

The late fusion approaches train the model for each feature separately, and therefore it is required to train many different models. However, one can apply suitable models for each feature, and better unimodal outputs can hence be obtained. There are many approaches to perform the late fusion operation [104]. One simple traditional approach is to take the maximum of the outputs from different unimodal models. Another common approach is to assign a weight to each unimodal output and then sum or take the mean the weighted outputs. However, it is necessary to choose an appropriate fusion method and weight values in order to obtain the best results.

In early fusion approaches, because of processing all the different features at once, one could see the cross-feature correlation, which cannot be seen equally well in the late fusion approaches. According to [104], the late fusion approaches tend to outperform the early fusion approaches in information retrieval tasks. Therefore, in this thesis, we are more focusing on late fusion than early fusion approaches.

4.2 Deep Boltzmann Machine for Multimodal Information Retrieval

Deep learning models have greatly impacted and shown magnificent success also in the multimodal information retrieval area, such as visual information retrieval [127, 126, 62] and text information retrieval [45, 95, 107]. There exists many interesting variants of deep learning models that have been used: Deep Belief Network (DBN) [43], Deep Boltzmann Machine (DBM) [90], Convolutional Neural Network (CNN) [62], Generative Adversarial Network (GAN) [91] and Quantum Boltzmann Machine (QBM) [3]. Unfortunately, it is far beyond the scope of this dissertation to describe all these models in details.

In our experiments in Publication I, Publication VI and Publication VII, we applied DBM as one of the main learning models for multimodal classification tasks. We selected to study multimodal DBM because it outperformed the linear and non-linear SVM in multimodal classification task in [106].

Before introducing the architecture of the multimodal DBM model, it is good to understand the general idea of the Boltzmann machine and also the details of unimodal DBM variants for each modality. In this section, we will introduce the details of the DBM and its application to text-based information retrieval.

4.2.1 Restricted Boltzmann Machine

A Boltzmann Machine [26, 46, 2] is a stochastic recurrent neural network model. The network is symmetrically weighted and connected with stochastic binary units which are categorized in two different groups: *visible input units* and *hidden units*. The visible units are fed with the actual input feature vectors and the hidden units form an interactions of the inputs. Let the set of the visible input units be defined as $\mathbf{v} \in \{0, 1\}^d$ and one of hidden units as $\mathbf{h} \in \{0, 1\}^p$, where d and p are the lengths of the vectors \mathbf{v} and \mathbf{h} , respectively. The Boltzmann Machine uses an energy function as the cost function and the energy of the state $\{\mathbf{v}, \mathbf{h}\}$ can be defined as:

$$E(\mathbf{v}, \mathbf{h}|\theta) = -\mathbf{v}^\top \mathbf{L} \mathbf{v} - \mathbf{h}^\top \mathbf{J} \mathbf{h} - \mathbf{v}^\top \mathbf{W} \mathbf{h} - \mathbf{b}^\top \mathbf{v} - \mathbf{c}^\top \mathbf{h}, \quad (4.1)$$

where $\theta = \{\mathbf{W}, \mathbf{L}, \mathbf{J}, \mathbf{b}, \mathbf{c}\}$ are the parameters of the model and $\mathbf{W}, \mathbf{L}, \mathbf{J}$ represent the linear visible-hidden, visible-visible, hidden-hidden interaction

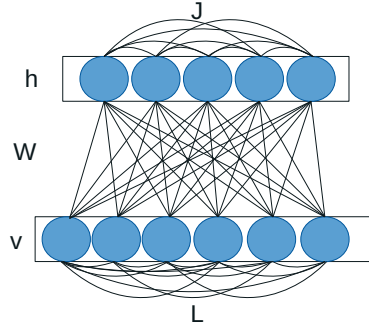


Figure 4.3. The architecture of the Boltzmann Machine. The Boltzmann Machine is a fully connected undirected graph. Compared with the other neural network models such as CNN, the parameters of each layer are relatively large. Hence, its computational cost tends to be higher than other models.

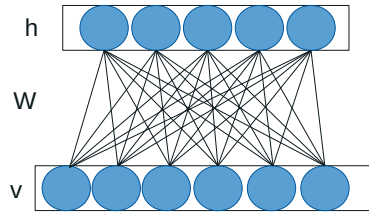


Figure 4.4. The architecture of the Restricted Boltzmann Machine. Unlike the Boltzmann Machine, the connections of RBM only exist between the layers. Therefore, their computational cost becomes smaller.

terms. The probability distribution of the visible input vector \mathbf{v} is:

$$p(\mathbf{v}|\theta) = \frac{1}{Z(\theta)} \sum_{\mathbf{h}} e^{-E(\mathbf{v}, \mathbf{h}|\theta)}, \quad (4.2)$$

where $Z(\theta) = \sum_{\mathbf{v}} \sum_{\mathbf{h}} e^{-E(\mathbf{v}, \mathbf{h}|\theta)}$. The Boltzmann Machine is a fully connected graph and its computational cost tends to be high (Figure 4.3). In order to reduce its complexity, by using only visible-hidden connection, the Restricted Boltzmann Machine (RBM) has been proposed [103]. The energy of the state $\{\mathbf{v}, \mathbf{h}\}$ is now:

$$E(\mathbf{v}, \mathbf{h}|\theta) = -\mathbf{v}^\top \mathbf{W} \mathbf{h} - \mathbf{b}^\top \mathbf{v} - \mathbf{c}^\top \mathbf{h} = -\sum_{i=1}^d \sum_{j=1}^p v_i h_j W_{ij} - \sum_{i=1}^d b_i v_i - \sum_{j=1}^p c_j h_j. \quad (4.3)$$

Because there are no direct connections either between the visible units or between the hidden units (Figure 4.4), it is easy to derive the condi-

tional probability distribution of the visible units:

$$P(\mathbf{v}|\mathbf{h}, \theta) = \prod_{i=1}^d p(v_i|\mathbf{h}) \quad (4.4)$$

with a given hidden vector:

$$p(v_i = 1|\mathbf{h}) = \rho(b_i + \sum_{j=1}^p h_j W_{ij}). \quad (4.5)$$

Similarly for the hidden units:

$$P(\mathbf{h}|\mathbf{v}, \theta) = \prod_{j=1}^p p(h_j|\mathbf{v}) \quad (4.6)$$

with a given visible unit vector:

$$p(h_j = 1|\mathbf{v}) = \rho(c_j + \sum_{i=1}^d v_i W_{ij}). \quad (4.7)$$

In (4.5) and (4.7), $\rho(x)$ is the logistic sigmoid function:

$$\rho(x) = \frac{1}{1 + e^{-x}}. \quad (4.8)$$

For the parameter update, one needs to perform the gradient ascent in the log-likelihood of equation (4.3):

$$\Delta W = \alpha(E_{P_{data}}(\mathbf{v}, \mathbf{h}) - E_{P_{model}}(\mathbf{v}, \mathbf{h})), \quad (4.9)$$

where α is the learning rate. $E_{P_{data}}$ is the expectation with respect to the complete data distribution:

$$P_{data}(\mathbf{h}, \mathbf{v}|\theta) = p(\mathbf{h}|\mathbf{v}, \theta)P_{data}(\mathbf{v}) \quad (4.10)$$

with

$$P_{data}(\mathbf{v}) = \frac{1}{N} \sum_{n=1}^N \delta(\mathbf{v} - \mathbf{v}_n). \quad (4.11)$$

$E_{P_{model}}$ is the expectation of the distribution defined by the model.

In the late fusion approach for multimodal retrieval task, it is preferable to apply an optimal model for each modality. For example, the current image features, such as DCNN activation features are real values rather than binary. Therefore, the RBM which could only handle binary vectors is not suitable to be used directly. Recent studies on text information retrieval usually deal with large corpora. In this case, the vocabulary tends to be large and sparse. Therefore, it is better to have an optimal model for sparse datasets and the RBM is not a suitable model to be used directly in this case.

4.2.2 Replicated Softmax Model

The Replicated Softmax model is one of the variants of the RBM model. In it, the visible binary input variable is replaced with multinomial variables of a number of alternative states [45]. For a binary unit, the probability of activation by the logistic sigmoid function on the input x can be expressed as:

$$\rho(x) = \frac{1}{1 + e^{-x}} = \frac{e^x}{e^x + 1}. \quad (4.12)$$

The energy contribution of an input is $-x$, if it is activated, otherwise it is 0. Therefore, it is easy to generalize the probability of K alternative states:

$$\rho_j = \frac{e^{x_j}}{\sum_i^K e^{x_i}}. \quad (4.13)$$

This unit is often called a *softmax unit*. This can be seen as approximated binary units where exactly one of the values is 1 and the others are 0.

If a softmax unit is considered as the word counter vector for documents with vocabulary size K , it is more suitable for modeling due to its sparsity than general RBM. This is because it can be handled a $M \times K$ matrix, where M is the number of words occurring in a document, as the visible input units. The conditional distribution of the visible binary matrix $\{\mathbf{V}, \mathbf{h}\}$ is then:

$$P(\mathbf{V}, \mathbf{h}|\theta) = \frac{1}{Z(\theta)} e^{-E(\mathbf{V}, \mathbf{h}|\theta)}, \quad (4.14)$$

where $Z(\theta) = \sum_{\mathbf{V}} \sum_{\mathbf{h}} e^{-E(\mathbf{V}, \mathbf{h}|\theta)}$ and:

$$E(\mathbf{V}, \mathbf{h}|\theta) = -\mathbf{V}^\top \mathbf{W} \mathbf{h} - \mathbf{b}^\top \mathbf{V} - \mathbf{c}^\top \mathbf{h} = - \sum_{i,j,k} v_{ik} h_j W_{ijk} - \sum_{i,k} b_{ik} v_{ik} - \sum_j c_j h_j. \quad (4.15)$$

Under the assumption of Replicated Softmax model, the order of the words in the documents can be ignored. Hence, the same weights for the connection between the softmax units and binary hidden units can be shared.

The energy of the state $\{\mathbf{V}, \mathbf{h}\}$ is now represented as:

$$E(\mathbf{V}, \mathbf{h}|\theta) = -\mathbf{V}^\top \mathbf{W} \mathbf{h} - \mathbf{b}^\top \mathbf{V} - \mathbf{c}^\top \mathbf{h} = - \sum_{i,j,k} v_{ik} h_j W_{ijk} - \sum_{i,k} b_{ik} v_{ik} - M \sum_j c_j h_j. \quad (4.16)$$

The conditional distributions are given by:

$$\begin{cases} p(v_i = 1|\mathbf{h}) = \frac{e^{b_i + \sum_j (h_j W_{ij})}}{\sum_i^K e^{b_i + \sum_j h_j W_{ij}}} \\ p(h_j = 1|\mathbf{v}) = \rho(c_j + \sum_k (V_k W_{kj})) \end{cases} \quad (4.17)$$

The log-likelihood for the parameter update is now:

$$\Delta W = \alpha (\mathbb{E}_{P_{data}}(\mathbf{V}, \mathbf{h}) - \mathbb{E}_{P_{model}}(\mathbf{V}, \mathbf{h})). \quad (4.18)$$

In the experiments in [45], the Replicated Softmax model showed better precision than the Latent Dirichlet Allocation (LDA) model.

4.2.3 Gaussian-Bernoulli RBM

RBM is a model for binary input vectors. Hence, it is not appropriate to be used for real-valued input features, such as the SIFT or DCNN features in image representation. Gaussian-Bernoulli RBM [32, 44] is one of the RBM variants designed for handling real-valued feature representation vectors.

Let $\mathbf{v} \in \mathbb{R}^D$ be the real-valued visible input vectors and the energy state $\{\mathbf{v}, \mathbf{h}\}$ representation for the Gaussian-Bernoulli RBM is:

$$E(\mathbf{v}, \mathbf{h}|\theta) = \sum_i \frac{(v_i - b_i)^2}{2\delta_i^2} - \sum_{i,j} \frac{v_i}{\delta_i} h_j W_{ij} - \sum_j c_j h_j. \quad (4.19)$$

The conditional distributions are then given by:

$$v_i|\mathbf{h} \sim \mathcal{N}(b_i + \delta_i \sum_j (h_j W_{ij}), \delta_i^2) \quad (4.20)$$

$$p(h_j = 1|\mathbf{v}) = \rho(c_j + \sum_i \frac{v_i}{\delta_i} W_{ij}), \quad (4.21)$$

where $\mathcal{N}(\mu, \delta^2)$ is the Gaussian distribution with the mean μ and variance δ^2 . The log-likelihood for the parameter update is calculated as:

$$\Delta W = \alpha(\mathbb{E}_{P_{data}}[\frac{\mathbf{v}^\top \mathbf{h}}{\delta}] - \mathbb{E}_{P_{model}}[\frac{\mathbf{v}^\top \mathbf{h}}{\delta}]). \quad (4.22)$$

In [44], the Gaussian-Bernoulli RBM is proposed as a dimensionality reduction method for efficient learning on image features. Recently, the amount of input data tends to be large because of the growing data sizes. Hence, it is favorable to reduce the dimensionality of the inputs. The Gaussian-Bernoulli RBM is applicable to image information retrieval from this point of view.

4.2.4 Deep Boltzmann Machine

The Deep Boltzmann Machine (DBM) [90] is an extended version of the RBM and has additional hidden layers which are fully connected to the previous and following layers. There are no connections between the hidden units in the same layer. According to [90], there are several reasons why DBM is a suitable architecture for information retrieval tasks. First, because of its layered architecture and layer-wise pre-training procedure, DBM has the potential of learning increasingly complex representations

of the input. Second, it is able to build a high-level representation from a large number of unlabeled data and a relatively small number of labeled data. Finally, it is robust to ambiguous inputs because the approximate inference procedure can incorporate top-down feedback.

The multiple hidden layers can be represented as $\mathbf{h}^{(1)}, \dots, \mathbf{h}^{(L)}$, where L is the number of the hidden layers. The energy state of the DBM can be written as:

$$E(\mathbf{v}, \mathbf{h}|\theta) = -\mathbf{v}^\top \mathbf{W}^{(1)} \mathbf{h}^{(1)} - \mathbf{b}^\top \mathbf{v} - \mathbf{c}^{(1)\top} \mathbf{h}^{(1)} + \sum_{l=2}^L (-\mathbf{c}^{(l)\top} \mathbf{h}^{(l)} - \mathbf{h}^{(l-1)\top} \mathbf{W}^{(l-1)} \mathbf{h}^{(l)}). \quad (4.23)$$

The probability that the model assigns to the visible vector \mathbf{v} is:

$$P(\mathbf{v}|\theta) = \frac{1}{Z(\theta)} \sum_{\mathbf{h}} \exp(-E(\mathbf{v}, \mathbf{h}^{(1)}, \dots, \mathbf{h}^{(L)}|\theta)). \quad (4.24)$$

The log-likelihood for the parameter update of each layer is calculated as:

$$\begin{cases} \Delta W_0 = \alpha(E_{P_{data}}(\mathbf{v}, \mathbf{h}^{(1)}) - E_{P_{model}}(\mathbf{v}, \mathbf{h}^{(1)})) \\ \Delta W_l = \alpha(E_{P_{data}}(\mathbf{h}^{(l-1)}, \mathbf{h}^{(l)}) - E_{P_{model}}(\mathbf{h}^{(l-1)}, \mathbf{h}^{(l)})). \end{cases} \quad (4.25)$$

One of the advantages of DBM is the layer-wise architecture. Therefore, one can easily add a new hidden layer and combine the outputs from DBM experiments without harming or modifying the previous architecture. The architecture of DBM is well-suited for the multimodal information retrieval task.

4.2.5 Multimodal DBM

In general, not just any DBM variant can deal with the multimodal inputs such as image-text inputs properly. However, in [106] a multimodal DBM model which joins two unimodal DBMs with an extra hidden layer was proposed. In [106], the Gaussian-Bernoulli RBM based DBM was applied for the image-text bimodal problem, to model the image distribution and the Replicated Softmax model based DBM for modeling the text distribution. Then, both DBM outputs joined with an extra hidden layer, as shown in Figure 4.5. The energy state and the probability that the two layers DBM assigns to the visible vector \mathbf{v}' are given by:

$$\begin{aligned} E(\mathbf{v}', \mathbf{h}'^{(1)}, \mathbf{h}'^{(2)}|\theta') &= \sum_i \frac{(v'_i - b'_i)^2}{\delta_i^2} \\ &\quad - \left(\sum_{i,j} \frac{v'_i}{\delta_i} W'_{ij}{}^{(1)} h'_j{}^{(1)} + \sum_{j,o} W'_{jo}{}^{(2)} h'_j{}^{(1)} h'_o{}^{(2)} \right) \\ &\quad + \sum_j c'_j{}^{(1)} h'_j{}^{(1)} + \sum_o c'_o{}^{(2)} h'_o{}^{(2)}. \end{aligned} \quad (4.26)$$

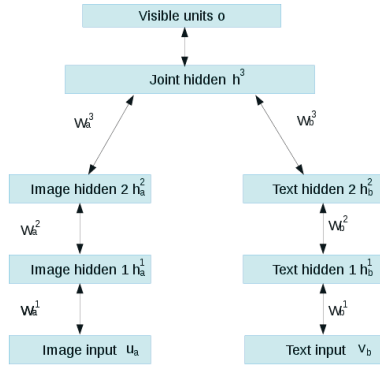


Figure 4.5. Multimodal DBM [106]. The left side layers are an image-specific DBM and the right side layers are a text-specific DBM.

$$P(\mathbf{v}'|\theta') = \frac{1}{Z(\theta')} \sum_{\mathbf{h}'^{(1)}, \mathbf{h}'^{(2)}} \exp(-E(\mathbf{v}', \mathbf{h}'^{(1)}, \mathbf{h}'^{(2)}|\theta')). \quad (4.27)$$

On the other hand, the energy state and the probability that the DBM assigns to the text vector \mathbf{v}'' can be represented as:

$$\begin{aligned} E(\mathbf{v}'', \mathbf{h}''^{(1)}, \mathbf{h}''^{(2)}|\theta'') &= -\left(\sum_{j,k} W_{kj}''^{(1)} h_j''^{(1)} v_k'' + \sum_{j,q} W_{jq}''^{(2)} h_j''^{(1)} h_q''^{(2)}\right) \\ &\quad + \sum_k b_k'' v_k'' + M \sum_j c_j'' h_j''^{(1)} + \sum_q c_q'' h_q''^{(2)} \end{aligned} \quad (4.28)$$

$$P(\mathbf{v}''|\theta'') = \frac{1}{Z}(\theta'') \sum_{\mathbf{h}''^{(1)}, \mathbf{h}''^{(2)}} \exp(-E(\mathbf{v}'', \mathbf{h}''^{(1)}, \mathbf{h}''^{(2)}|\theta'')). \quad (4.29)$$

The energy state and the joint distribution over the multimodal input $\mathbf{h} \in \{\mathbf{h}'^{(1)}, \mathbf{h}'^{(2)}, \mathbf{h}''^{(1)}, \mathbf{h}''^{(2)}, \mathbf{h}^{(3)}\}$ is then expressed as:

$$E(\mathbf{h}''^{(2)}, \mathbf{h}'^{(2)}, \mathbf{h}^{(3)}|\theta) = -\left(\sum_{q,p} W_{qp}'^{(3)} h_q''^{(2)} h_p^{(3)} + \sum_{q,p} W_{qp}'^{(3)} h_q'^{(2)} h_p^{(3)} + \sum_p b_p^{(3)} h_p^{(3)}\right) \quad (4.30)$$

$$\begin{aligned} P(\mathbf{v}', \mathbf{v}''|\theta) &= \frac{1}{Z(\theta)} \sum_{(\mathbf{h})} \exp(-E(\mathbf{v}', \mathbf{h}'^{(1)}, \mathbf{h}'^{(2)}|\theta')) \\ &\quad - E(\mathbf{v}'', \mathbf{h}''^{(1)}, \mathbf{h}''^{(2)}|\theta'') - E(\mathbf{h}'^{(2)}, \mathbf{h}''^{(2)}, \mathbf{h}^{(3)}|\theta). \end{aligned} \quad (4.31)$$

According to [106], the Multimodal DBM outperforms the SVM approach in multimodal information retrieval tasks. It has been interesting to compare multimodal information retrieval with linear and non-linear SVM to the operation of the multimodal DBM. The details of implementation and experimental results are described in Publications I and IV and summarised in Chapter 5.

4.3 Information Transfer to Other Modality

If the fusion approach is improving the information retrieval results, it can also be expected that transferring the semantic information to other modality can improve the unimodal information retrieval. In this section, we will introduce an approach for transferring semantic information to other modality.

4.3.1 Topic-Concept Similarity Map

In order to perform multimodal information retrieval effectively, it is required to have a multimodal query. For example, if one has a text catalog together with an image database, it is better to have both text and image queries for retrieving the required information. However, in a real situation, it is hard to always have both multimodal queries and a multimodal database. For instance, the users often only give the queries and it would be a very hard task to annotate a large image dataset properly. In either case, the solution is to transfer the semantic information to other modality. That is, the automatic tag annotation or query generation by the cross-modality semantics is the key. In order to transfer the semantic information to other modality, a dictionary or map of cross-modality semantics is required. In Publication VII, we proposed topic-concept similarity map. The idea of topic-concept similarity map is to express the similarity between the latent topics of the text and the corresponding visual image concepts. It requires a dataset which has images with corresponding text for training.

After we have modeled the distribution of latent topics in the article texts and the distribution of visual concepts in their corresponding images, we can model the similarity between the topics and concepts. A topic and a concept can be regarded as semantically similar, if they are co-occurring in a multimodal dataset of text–image pairs. If we know for each visual concept its related textual topics, we can then generate *pseudo tags* for the input images.

Assuming that we have N_A text–image pairs, for each of which we have solved the presence of C_t number of textual topics and R_c number of visual concepts, we can do the processing illustrated in Figure 4.6. Taking a topic t we can form an N_A -dimensional vector \mathbf{x}_t representing that topic’s existence in each text article in our dataset. Similarly, we can form another vector \mathbf{y}_c of the same dimensionality for any visual concept c . The

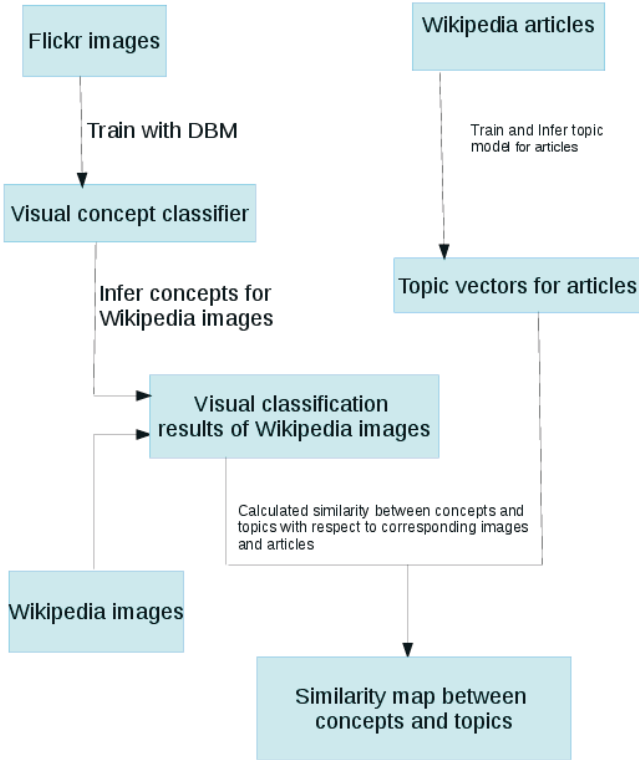


Figure 4.6. The process of creating a similarity map between the topics of articles and the image concepts.

dissimilarity between topic t and visual concept c can then be expressed with the cosine distance between these two N_A -dimensional real-valued vectors:

$$J(t, c) = 1 - \frac{\mathbf{x}_t^\top \mathbf{y}_c}{\|\mathbf{x}_t\| \|\mathbf{y}_c\|} . \quad (4.32)$$

The similarity between t and c can then be defined as the inverse of the distance:

$$Sim(t, c) = \frac{1}{J(t, c)} . \quad (4.33)$$

Then, we can form a $C_t \times R_c$ similarity map M whose components are $Sim(t, c)$.

4.3.2 Pseudo Tag Generation with Similarity Map

Our approach for generating pseudo tags is based on the results of the visual concept detection of each image. Taking into account top n of detected visual concepts for an image I_a and using the similarity map between the visual concepts and the text topics, we pick for each word w in top m of the most similar topics. We denote this set as $T_m(w)$. Then, let B be the vo-

cabulary of all detected topics, and w a word where $w \in B$. The similarity between the text word w and a visual concept c is calculated as:

$$Sim(w, c) = \sum_{t \in T_m(w)} Sim(t, c) \cdot Sim_p(t, w) , \quad (4.34)$$

$Sim_p(t, w)$ is WordNet [28] path similarity which takes into account the semantic similarity of the two argument words and is calculated by using the NLTK toolkit [1].

Then based on the $Sim(w, I_a)$, we choose the top 20 most similar words in this task. $Sim_p(t, w)$ is needed because of the similarity map is only mapping the latent topics and visual concepts, but there are many words which are related to each latent topic. That is, we need to choose a proper tag for the correspond visual concept among those words which are related to each corresponding latent topic. $Sim_p(t, w)$ assigns a score in the range from 0 to 1 based on the shortest path that connects the concepts or topics in the WordNet hierarchy. So, if the relationship between the topic and word is strong according to WordNet, it will give more weight to that word compared to other words. In our experiment, if there is no path or the word of tag does not exists in WordNet, we simply set the similarity equal to zero. The similarity between the word w and an image I_a is then:

$$Sim(w, I_a) = P(c|I_a) \cdot Sim(w, c) , \quad (4.35)$$

where $P(c|I_a)$ is the probability distribution of the class c given image I_a , obtained from the image classification results of DBM.

4.3.3 Unsupervised Pseudo Tag Generation

The above approach for pseudo tag generation is dependent on the supervised concept classification results in the image modality. That is, we need the true label information of the training images for the concept detectors. In a real-world situation, it is very hard to collect large amount of them. Therefore, we use Latent Dirihclet Allocation results as pseudo labels and implement an unsupervised approach for creating pseudo tags.

Let F^A be the training data set and F^B be the testing data set and both data sets have the text and image modalities. By using LDA, C_t^A topics are assigned to the text data in F^A . We can then use the C_t^A topics as labels for F^A , and train an image classifier G . Applying classification with G to the image data in F^B , we can create the vectors \mathbf{v}_t^B consisting of the probabilities of each image belonging to topics C_t^A . Based on their

probabilities, we choose $100 \cdot v_{t_i}^B$ words randomly from the words which have been assigned to topic t for image i .

4.4 Semantic Concept Vectors

Semantic concept vectors or *classemes* [111] can be used to incorporate semantic background information from auxiliary image–label training data. The use of semantic concept vectors is more like an early fusion approach because it generates extra features and extends the existing feature vectors. The labels can be either accurate class information if such exists, or less accurate tag information, if such is available for a large number of images and tags.

In Publications I and IV, the semantic concept vectors are produced in three steps: 1) a large number of semantic concept classifiers are trained from an auxiliary dataset, 2) the semantic concept classifiers are applied to the training and testing images of the primary dataset, and 3) for each image, the semantic concept classifier outputs are collected in a semantic concept vector which is interpreted as novel visual feature describing the image. These vectors can then be used as inputs when training the DBM and SVM models. Let C_1, \dots, C_{K_s} be the semantic concept vocabulary, the semantic concept vector \mathbf{c}_i for each image $m_i, i = 1, \dots, N$, is thus constructed as:

$$\mathbf{c}_i = [p_{i,1}, \dots, p_{i,K_s}]^T, \quad (4.36)$$

where $p_{i,j} \in [0, 1]$ is the SVM classifier output or *concept membership score* for image m_i in concept C_j .

4.5 Discussion

Recently, re-ranking methods have been spotlighted as one way of improving multimodal retrieval. Most of re-ranking methods are based on human labeled data, user relevance feedback and pseudo-relevance feedback to re-rank and improve the search results. The user relevance feedback method is to re-rank search result by using the feedback information of search result from users [92]. The basic idea of pseudo-relevance feedback is to use top-ranked search results as pseudo positive examples and low-ranked search results as pseudo negative examples, and to retrain and re-rank the retrieval results [124]. The active learning approach has

also worked well as an improvement of the ranking algorithm [68]. Since these re-ranking methods have shown improvements in unimodal information retrieval, one can expect them to work well also in multimodal information retrieval.

Because information retrieval is an interactive task with a user in the loop, the user feedback can be used as extra semantic data. Recently, reinforcement learning with deep learning methods has shown great success in user interactive tasks. For example, Volodymyr et al [75] succeeded in reaching the human operation level in Atari games. If one could collect or simulate enough user interaction data for the reinforcement learning process, it might be possible to improve the performance of multimodal information retrieval.

4.6 Application Examples

Because of the rapid growth of multimedia contents in web services, such as social media and cloud services, our research group has studied implementation of integrated multimedia content management systems. The multimodal information search techniques based on deep learning features and SVMs, as described in this and earlier sections, have been applied in two comprehensive multimodal media search systems.

The first one, PicSOM, has been developed by our research team. Its development and application in the TRECVID evaluations from 2005 to 2014 has been described in Publication V and Section 4.6.1 gives a brief overview of it.

The second system, VisualLabel, was developed in a joint project funded by the Finnish Science and Technology Council, together with researchers from two other Finnish universities. The system is presented in Publication IV and briefly introduced in Section 4.6.2.

4.6.1 PicSOM

Our research group has been developing the PicSOM multimodal information retrieval system since late 90's. The original PicSOM was based on the self-organizing map (SOM) algorithm [59]. It was inspired by the WEBSOM text document retrieval system [48] and performed interactive content-based image retrieval (CBIR). SOM visualizes high-dimensional data distributions on dimensional grid. Therefore, it is widely used as

a dimensionality reduction method and as a representation of complex non-linear relationships between data items [60]. SOM was inspired and modeled by the working of the visual cortex in the human brain.

The PicSOM system has gone through substantial evolution in both the statistical features and the detection algorithms employed. Transition from global image features to the bag-of-visual-words features and recently further to convolutional deep neural network-based features has been justified in the light of the performed experiments. Overall, during more than ten years of participation in the TRECVID workshops, the PicSOM system has shown close to state-of-the-art performance in this very rapidly developing field of research. Details of the implementation and the history of PicSOM are described in Publication V.

4.6.2 VisualLabel

VisualLabel is a novel and comprehensive integrated open source multimedia content management and access framework. It enables smart photo services, based on the automated visual content analysis, annotation, search and retrieval state-of-the-art back-ends for services, such as Flickr and Facebook. VisualLabel automatically organizes the user's multimedia data by using state-of-the-art machine learning algorithms to detect, for example, faces and different objects in the user's personal collection of images. The framework can also utilize multiple external web and cloud services (such as Flickr), and offers the end-user a Representational State Transfer (REST) Application Programming Interface (API) for multimedia organization and retrieval.

The full description of the system framework and experiments carried out with it can be found in Publication IV. Our main focus was visual content analysis and we applied our visual content analysis framework PicSOM in this task.

5. Experimental Evaluations

In this chapter, we focus on summarizing the results of the experiments which compare the unimodal and multimodal information retrieval approaches.

First, in Section 5.1, we compare the computational efficiency of non-linear SVM and the homogeneous kernel map SVM approaches in a video retrieval task. The computational cost of the current information retrieval methods is high and therefore the computational efficiency is a key issue. This is especially important for video retrieval because the data size and the dimensionality of the corresponding features tend to be larger than in text-only data.

In Section 5.2, we compare the SVM approach and the DBM approach in uni- and multimodal information retrieval tasks with several different settings for single-labeled data. As mentioned in earlier chapters, the deep learning approach has outperformed all other machine learning approaches in unimodal information retrieval studies. According to Srivastava et al [106], the Multimodal DBM outperforms the SVM approach in a multimodal information retrieval study. Hence, we decided to experiment how and why the DBM approach outperforms general SVM approaches.

Further, in Section 5.3, we experiment with the same machine learning approaches which are used in Section 5.2, but now on multi-labeled data. This is because the user's queries and the required information in a real situation usually contain multiple concepts.

In Section 5.4, we analyse the usefulness of the pseudo tag approach for multimodal information retrieval with experimental results and study auto-annotation of images with tags in order to improve the multimodal information retrieval.

5.1 TRECVID Semantic Indexing Task

Visual content information retrieval techniques have improved by using new machine learning techniques, but most of the algorithms are time-consuming and the processing times have therefore also increased. In Publication III and Publication IV, we focused on speeding up the visual concept detection task on TRECVID data set by using linear approximation kernel SVM (see Section 3.2.1).

5.1.1 Setup

The TRECVID Video Retrieval Evaluation is one of the TREC annual conferences and workshops organized by the National Institute of Standards and Technology (NIST) and other U.S. government agencies [77]. This challenge and competition has been organized since 2003. The main purpose of TRECVID is to advance the content-based retrieval and analysis of video content through challenges and competitions.

The TRECVID tasks are updated every year because of the progress of content-based retrieval research and the demands of real-world application scenarios. Recent interests of TRECVID have been semantic indexing, surveillance event detection, instance search, multimedia event detection, localization, and video hyperlinking. The Semantic Indexing (SIN) task has been organized for many years and our research group has participated for about a decade. The aim of the task is to develop new techniques for automatic assignment of semantic content descriptions representing visual or multimodal concepts found in video shots.

5.1.2 TRECVID Dataset

Data sets for the TRECVID SIN task change annually and the past development and test datasets are used as the new training dataset. In 2015, 60 visual concepts from the LSCOM ontology [58] such as *Airplane*, were extracted and annotated for being used as the test data. The ground truth labels have been provided yearly after each evaluation has ended. The training data of 2015 consists of the training data for 2010 and the testing data for 2010, 2011, and 2012.

The training data contains approximately 3200 Internet Archive videos¹ with duration between 3.6 and 4.1 min with total of 50 GB and 200

¹<https://archive.org/>

id	features			hard neg.	MXIAP
	glob.	BoV	FV	mining	
Row 1	•	•			0.1951
Row 2		•	•	•	0.2722
Row 3				•	0.2843
Row 4		•	•	•	0.2880

Table 5.1. An overview of our runs submitted for the TRECVID 2014 evaluation. glob. represents global BoV bag of visual words, Fisher vectors (FV) + VLAD and convolutional neural network (CNN) features. We submitted four runs: Row1 uses global and BoV features. Row2 combines BoV, Fisher vector and CNN features. Row3 uses CNN features with hard negative mining. Row4 combines BoV, Fisher vector and CNN features with hard negative mining.

hours. Most of the videos have donor annotated metadata information (title, keywords, and description). There are two test datasets. One consists of about 8000 Internet Archive videos with 160 GB and 600 hours in total. The duration of each video is between 10 s and 3.5 min. The other consists of about 7300 Internet Archive videos with donor annotated metadata, and totals 144 GB and 600 hours. The duration of each video is from about 10 s to 6.4 min and the mean duration is almost 5 min.

5.1.3 Experiments and Results

The TRECVID SIN data has been used in two studies of this thesis: 1) development of features and classifiers, and 2) speeding up the classification. The aim of both experiments is to detect a large number of visual concepts from the TRECVID dataset. This is the same setting as the TRECVID SIN evaluation task.

Development of the features and classifiers

The visual concept detectors for the semantic indexing task have been developed in the PicSOM research group for more than a decade. We have tried many approaches for this task, such as the Self-Organizing Map (SOM) and linear and non-linear SVM classifiers. The details of our 10 years of work are described in Publication V. Then, most relevant for the current thesis in this publication are the experiments in TRECVID 2014. During these years, we evaluated several alternative techniques used for implementing visual concept detection. Table 5.1 shows an overview of our submitted runs, where the four columns in the middle refer to the used features: global features, BoV features, Fisher vectors + VLAD, and

CNN features. The next column indicates whether hard negative mining was used, and the rightmost column lists the corresponding mean extended inferred average precision (MXIAP) [122] values.

Run 1 is intended to match our best submission in TRECVID 2013, i.e. to use the same features, classifiers, and method of fusion [55]. In Run 2, the Fisher vector [94] and VLAD [6] features and the set of 24 CNN features were included and the global image features discarded. Run 3 uses only the CNN features, together with hard negative mining, and Run 4 combines the characteristics of Runs 2 and 3, that is, all SIFT-based and CNN features, with hard negative mining [5].

The most striking observation of the results is the notable increase in performance compared to our previous year’s submissions. This is mostly due to the extended set of features, in particular the CNN activation features. By comparing Runs 1 and 2, we observe a 40% increase on MXIAP induced by the different feature sets.

Second, the mining of hard negatives further improved the results, as can be observed by comparing Runs 2 and 4, the latter including the mining step and obtaining the highest MXIAP among our runs, 0.2880 (a 6% increase). The solid performance of the CNN features can furthermore be observed from Run 3, which contains only the CNN features but still almost reaches the MXIAP value of Run 4.

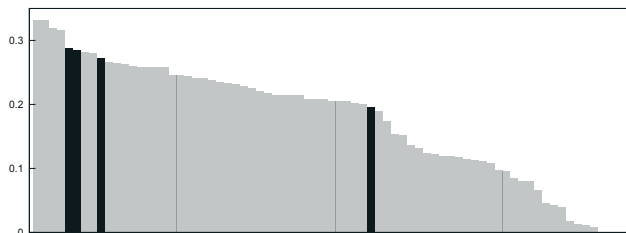


Figure 5.1. MXIAP values for all submissions to the TRECVID 2014 semantic indexing task. Our runs highlighted.

Figure 5.1 shows all runs submitted to the TRECVID 2014 semantic indexing task, our runs highlighted. In total, there were 75 submissions, and only the MediaMill group of the University of Amsterdam submitted runs that were superior to the two best PicSOM runs in their MXIAP results.

Speeding up the classification

In these experiments, we compared four types of linear and non-linear kernels for SVM classification: linear SVM (linear), the fusion results

model	1	50	500	MXIAP
linear	4.7	4.7	4.8	0.107
multi-learn	4.7	4.8	5.4	0.122
hkm-INT	4.7	4.7	5.2	0.126
hkm- χ^2	4.7	4.7	5.2	0.125

Table 5.2. Processing times (secs) for 1, 50 and 500 concepts and MXIAP scores for the hand-crafted features of TRECVID2011 dataset. linear represents our base-line model which used linear SVM classifier only. Multi-learn represents the fusion results of multiple learning classifiers. Hkm-INT represents SVM with homogeneous kernel map of intersection. Hkm- χ^2 represents SVM with the homogeneous kernel map of χ^2 kernel.

model	1 concept (ms)	346 concepts (sec)	MXIAP
linear	0.019	0.0064	0.132
hkm-INT	0.12	0.042	0.191
hkm- χ^2	0.12	0.041	0.182
pm-INT	0.15	0.052	0.200
pm- χ^2	0.15	0.051	0.200
K_{RBF}	180	63	0.198
$K_{\chi^2}^{exp}$	210	74	0.217
K_{INT}^{exp}	100	36	0.214

Table 5.3. Processing time and MXIAP scores of TRECVID2012 dataset (SIFT based feature). Linear, hkm-INT, and hkm- χ^2 are the same setting as in Table 5.2. Power mean SVM is used with the homogeneous kernel map of intersection (pm-INT) and χ^2 (pm- χ^2) kernel. As references of non-linear SVM results, we used non-linear kernels of RBF kernel (K_{RBF}), the exponential of χ^2 kernel ($K_{\chi^2}^{exp}$), and the exponential of intersection kernel (K_{INT}^{exp}).

of multiple learning classifier (multi-learn), homogeneous kernel map of intersection (hkm-INT) and hkm- χ^2 kernels. Details of these SVM approaches are described in Section 3.2.1. and the details of the experiments on the speeding up of the classification task are in Publication II and Publication III.

Table 5.2 shows the feature extraction and detection processing times and MXIAP score results for 1, 50, and 500 visual concepts in the TRECVID2011 semantic indexing task. The performance of the homogeneous kernel classifiers outperform the linear classifier. The linear approximation approach did not increase the processing time substantially either, taken into account the constant time spent in feature extraction.

Table 5.3 further shows a comparison between the linear, linear approximation and non-linear kernel approaches. For non-linear kernels, we ap-

plied the RBF kernel (K_{RBF}), the exponential intersection kernel (K_{INT}^{exp}), and the exponential χ^2 kernel (K_{INT}^{exp}). In this experiment, we also added the power mean SVM (pm-INT, pm- χ^2) which is an alternative approach to additive kernel approximation (see Section 3.2.2). In the overall results of Table 5.3, the linear approximation approach (hkm-INT, hkm- χ^2 , pm-INT, and pm- χ^2) outperforms the linear SVM. The approximation methods also showed their performance to be close to the non-linear SVM without significant increase in processing time. The detailed explanation of this experiments can be found in Publication III.

5.1.4 Discussion

In recent studies it has been observed that despite their limited accuracy, semantic concept detectors can be useful in supporting the indexing of high-level features and querying on multimedia data [40]. This is mainly because such detectors can be trained off-line with computationally more demanding supervised learning algorithms and with considerably well-organized training data compared to what is typically available at query time. In the real world, the amount of multimedia data on the Internet is growing rapidly and the users want to have the information as soon as possible when they use Internet for search purposes. In the experiments in this section, two linear approximation kernel map approaches, homogeneous kernel map and power mean kernel maps, showed great performance in reducing the training time cost without losing the detection performance.

With our experiments in TRECVID 2014, we have shown that the top performance obtained in many image classification tasks with deep convolutional neural networks can be carried over to semantic video indexing tasks. For the reasons of computational complexity, we used linear SVM detectors with homogeneous kernel maps to approximate the intersection kernel. Combined with the hard negative mining technique in detector training, the PicSOM group ranked second among 21 participants in the semantic indexing task.

5.2 Single-label Multimodal Information Retrieval

In this section, we will show a comparison of uni- and multimodal information retrieval approaches for the concept detection task on MIR-

FLICKR Data [51]. The detailed description of the multimodal DBM and SVM can be found in Publication I, and Publication VI.

5.2.1 Setup

The main purpose of the experiments was to find out, whether the Multimodal DBM or the non-linear SVM with late fusion of unimodal recognition results is performing better in single-label recognition tasks. We also evaluated the performance of different visual and textual features we had available. In addition, we also aimed to determine whether the DBM pretraining approach or the use of the semantic concept vectors, both utilizing the same auxiliary data, is superior in performance. Most of the experimental setting is inspired by [106, 105]. The original experiments contain a comparison between the SVM approach and the DBM approach for the multimodal classification task on the MIRFLICKR dataset. In order to compare with their results, we decided to use the MIRFLICKR data set also as our experimental dataset.

5.2.2 MIR Flickr Dataset

MIRFLICKR-25000 is an image collection which consists of 25000 images downloaded from the social photography site Flickr.com [51]. It is offered by the LIACS Medialab at Leiden University, The Netherlands, and the images are freely available for scientific experimenting. The image selection has been done according to "interestingness" as is defined by Flickr.com. The "interestingness" is based on where the click-throughs are coming from, who comments on the image and when, who marks it as a favorite, its tags, and many more things which are constantly changing [30]. Because of adding new users' interests to Flickr over time, the interestingness also changes over time. User-annotated Flickr tag and EXIF metadata files are also contained in the dataset.

The release also provides manual annotation of the whole image dataset. In order to make it easy to extend the annotation with new keywords without the need to go through the whole dataset, the annotation scheme is set with two step levels: *relevance level* and *abstraction level*. In the relevance level, each concept or topic was annotated for all images where it is visible or applicable to at least to some extent. These labels were named potential labels. Then, using the potential labels, the images with relevant labels were annotated by a single annotator. In the abstraction

level, the annotation was based on a semantical hierachy so that, for example, *dog* and *cat* as relevant labels are annotated as the abstract level *animal*. The dataset contains 38 annotated concept categories including scene categories, such as *sky*, *river*, *lake*, and object categories, such as *portrait*, *people*, *car*, and 94 child categories such as *day*, *sun*, *baby*, *male*, under the corresponding super categories.

The developers of MIRFLICKR-25000 have also provided MIRFLICKR-1M which is an extension of MIRFLICKR-25000. MIRFLICKR-1M consists of the core MIRFLICKR-25000 images and additional 975,000 images downloaded from Flickr.com. These new images are also selected based on the "interestingness" score by Flickr. They are unannotated but the user-given Flickr tags, such as "beach", "coast", "ocean", "pacific", "shore", etc, and EXIF metadata files, such as uploading time, have been provided.

5.2.3 Experiments and Results

For our experiments we implemented a setting similar to what Srivastava *et al* used in [106, 105] with the MIRFLICKR-1M dataset. For the multi-modal DBM experiments the method is described in Section 4.2.5 and the SVM methods are described in Section 3.2.1.

For the text feature inputs v_b , we used the same $K = 2000$ sized vocabulary of the most common tags as used in the work [106, 105]. This feature is in the following result Tables, referred to as *text*. Additionally we also used 200-dimensional word2vec features [74], referred to as *word2vec*. In order to compare the image classification results, we used as the image features u_a the PHOW, Gist and MPEG-7 based features (the concatenated dimensionality $L = 3857$) provided in [106] and referred to as *PHOW*, *etc*. Our own DCNN GoogLeNet activation features ($L = 2048$) (see details in Section 3.1.2) are shown as *GoogLeNet* in the results.

The number of hidden units in each DBM layer was the same as in [106]. For the MIRFLICKR-1M dataset, we set the dimensionality of the text hidden layers $h_b^{(1)}$ and $h_b^{(2)}$ (as in Section 4.2.5) to $F = 1024$. For the first image hidden layer $h_a^{(1)}$ is $F = 2048$ and $F = 1024$ for the second hidden layer $h_a^{(2)}$. The multimodal joint layer $h^{(3)}$ has again $F = 2048$ hidden units. Following the original procedure, we used the DBM model with and without pre-training with the 975,000 unannotated images that have tags only. The use of DBM pre-training, which can be seen as an alternative to using the semantic concept vectors (described in Section

3.1.3), is indicated as *pre-t* in the results, whereas the use of the semantic concept vectors is shown as *tags*.

Similarly to the original results, we performed each experiment five times, always using 10,000 objects for training, 5,000 objects for validation and the remaining 10,000 objects for testing. The experiments with the MIRFLICKR-1M dataset were repeated twice, first with the 38 concept categories and then with the 94 categories.

In general, the optimal weights for the SVM classifiers' late fusion were selected with cross validation with the 38 concept categories and the same weights were then applied also to the 94 categories.

For MIRFLICKR-1M dataset, we used the semantic concept vectors ($L = 500$) as additional features. They were used primarily to train separate "semantic" unimodal SVM models. With the Multimodal DBM model, we also experimented by concatenating the semantic concept vectors with the image input features, instead of using the DBM pretraining approach [106].

The margins of error shown in the MIRFLICKR-1M dataset results are based on the estimated standard deviation of the 5-fold cross validation results.

The results of the experiments with the MIRFLICKR-1M dataset are shown in Tables 5.4 and 5.5 for the 38 and 94 concept cases, respectively. The performances are measured as the mean average precision (MAP) (see Section 2.4.2), the precision at rank 50 (Prec@50) (see Section 2.4.2) and the average area under the ROC curve (aAUC) (see Section 2.4.3). The result rows have been labeled so that the corresponding methods in the two tables always have the same label. Based on the results with the 38 MIRFLICKR-1M concepts, we decided not to use the linearly approximated intersection SVMs nor the word2vec features in the experiments with the 94 concepts. Consequently, the corresponding rows are missing in Table 5.5.

Rows A1–A3 show the results with the image only unimodal models trained without using the 975,000 unannotated images for pretraining nor as semantic concept tags. Rows A4–A8 are results with text-only unimodal models by using the binary tag information or word2vec features. Rows A9 and A10 show results with only semantic concept vectors. Rows A11 and A12 are results with the multimodal models using the visual and textual but not semantic features. Rows A13–A17 show models combining image information either with the semantic concept vectors or by using

	model	image	text	pre-training	MAP	Prec@50	aAUC
A1	DBM	GoogLeNet	—	—	0.723±0.004	0.915±0.003	0.9415±0.0005
A2	LIN	GoogLeNet	—	—	0.702±0.007	0.903±0.005	0.9309±0.0039
A3	RBF	GoogLeNet	—	—	0.721±0.004	0.905±0.004	0.9417±0.0010
A4	DBM	—	text	—	0.488±0.004	0.829±0.008	0.8175±0.0020
A5	LIN	—	text	—	0.421±0.010	0.709±0.016	0.7709±0.0019
A6	RBF	—	text	—	0.490±0.006	0.805±0.014	0.8112±0.0014
A7	LIN	—	word2vec	—	0.267±0.004	0.420±0.008	0.7286±0.0041
A8	RBF	—	word2vec	—	0.466±0.003	0.798±0.008	0.8108±0.0024
A9	DBM	—	—	semantic	0.729±0.005	0.916±0.004	0.9446±0.0010
A10	RBF	—	—	semantic	0.720±0.003	0.901±0.005	0.9415±0.0010
A11	DBM	GoogLeNet	text	—	0.745±0.003	0.923±0.003	0.9452±0.0004
A12	RBF	GoogLeNet 70%	text 30%	—	0.741±0.003	0.911±0.005	0.9467±0.0003
A13	DBM [106]	PHOW, etc.	—	pre-t	0.469±0.005	0.803±0.005	
A14	DBM	PHOW, etc.	—	pre-t	0.475±0.002	0.753±0.002	0.8585±0.0010
A15	DBM	GoogLeNet	—	pre-t	0.727±0.003	0.918±0.004	0.9429±0.0006
A16	DBM	GoogLeNet	—	semantic	0.731±0.003	0.919±0.005	0.9460±0.0003
A17	RBF	GoogLeNet 50%	—	semantic 50%	0.735±0.003	0.909±0.004	0.9471±0.0003
A18	DBM	—	text	pre-t	0.511±0.004	0.834±0.005	0.8495±0.0005
A19	DBM [105]	—	text	pre-t	0.531±0.005	0.832±0.004	
A20	DBM	—	text	semantic	0.734±0.002	0.916±0.004	0.9426±0.0010
A21	RBF	—	text 25%	semantic 75%	0.740±0.002	0.909±0.006	0.9462±0.0012
A22	DBM [105]	PHOW, etc.	text	pre-t	0.531±0.005	0.832±0.004	
A23	DBM [105]	PHOW, etc.	text	pre-t	0.609±0.004	0.873±0.004	
A24	DBM [106]	PHOW, etc.	text	pre-t	0.641±0.004	0.888±0.004	
A25	DBM	GoogLeNet	text	pre-t	0.748±0.003	0.919±0.005	0.9467±0.0007
A26	DBM	GoogLeNet	text	semantic	0.753±0.003	0.925±0.005	0.9452±0.0007
A27	RBF	GoogLeNet 37.5%	text 25%	tags 37.5%	0.752±0.002	0.915±0.006	0.9506±0.0007

Table 5.4. MIRFLICKR-1M 38 concept classification results with different models. LIN = linear SVM, RBF = non-linear RBF kernel SVM, text = 2000-dimensional 0/1 tag features, word2vec = 200-dimensional word2vec features, pre-t = DBM pre-training performed with 975,000 unannotated images and/or tags, PHOW, etc. = hand-crafted features of [106], semantic = semantic concept vectors. Best results in each group are bolded.

	model	image	text	additional	MAP	Prec@50	aAUC
A1	DBM	GoogLeNet	—	—	0.405±0.004	0.550±0.006	0.8998±0.0008
A3	RBF	GoogLeNet	—	—	0.439±0.006	0.570±0.003	0.8958±0.0004
A4	DBM	—	text	—	0.270±0.003	0.456±0.007	0.7630±0.0016
A6	RBF	—	text	—	0.262±0.007	0.430±0.007	0.7412±0.0053
A9	DBM	—	—	sem	0.422±0.003	0.565±0.005	0.9009±0.0010
A10	RBF	—	—	sem	0.429±0.005	0.559±0.001	0.8924±0.0005
A11	DBM	GoogLeNet	text	—	0.458±0.004	0.594±0.008	0.9008±0.0012
A12	RBF	GoogLeNet 70%	text 30%	—	0.458±0.003	0.582±0.002	0.8993±0.0038
A15	DBM	GoogLeNet	—	pre-t	0.437±0.004	0.573±0.005	0.9001±0.0005
A16	DBM	GoogLeNet	—	sem	0.441±0.004	0.580±0.008	0.9012±0.0012
A17	RBF	GoogLeNet 50%	—	sem 50%	0.449±0.005	0.577±0.002	0.9039±0.0013
A18	DBM	—	text	pre-t	0.287±0.002	0.463±0.007	0.7921±0.0030
A20	DBM	—	text	sem	0.426±0.004	0.568±0.009	0.8944±0.0030
A21	RBF	—	text 25%	sem 75%	0.449±0.004	0.579±0.005	0.8972±0.0046
A25	DBM	GoogLeNet	text	pre-t	0.459±0.003	0.599±0.008	0.9020±0.001
A26	DBM	GoogLeNet	text	sem	0.464±0.005	0.600±0.009	0.9045±0.001
A27	RBF	GoogLeNet 37.5%	text 25%	sem 37.5%	0.467±0.003	0.591±0.003	0.9069±0.001

Table 5.5. MIRFLICKR-1M 94 concept classification results with different models. RBF = non-linear RBF kernel SVM, text = 2000-dimensional 0/1 tag features, pre-t = DBM pre-training performed with 975,000 unannotated images and/or tags, sem = semantic concept vectors. Best results in each group are bolded.

DBM pretraining. Rows A18–A21 contain results where the unimodal text models were used either with unimodal semantic models by using DBM pretraining. Finally, rows A22–A27 are the recognition results of either genuinely multimodal DBM image–text models or three post-fused unimodal SVM models.

In all cases where the SVM detectors of multiple modalities or features have been combined (i.e. the rows A12, A17, A21 and A27) the weight percentages are shown in the table. The multimodal combination of the DBM results was always performed by using the Multimodal DBM model.

Row A23 shows the best multimodal model in [105]. In this case, the text input was not clamped and the model was allowed to update the text input layer when performing the mean-field update. Similarly, row A24 is the best multimodal result of [106], where various additional techniques were used to improve the MAP result.

Comparing the performance of different features, in rows A13 vs. A15 and A24 vs. A25, it is clear but unsurprising that the GoogLeNet features outperformed the PHOW-based and other hand-crafted features. In the text modality, rows A5 vs. A7 and A6 vs. A8, the 200-dimensional word2vec features gave disappointing result compared to the full 2,000-dimensional binary text features. Also, the semantic concept vector features, when combined with either the visual or textual unimodal model, gave a significant improvement in the MAP results in rows A15 vs. A16,

A6 vs. A20, and A25 vs. A26.

Comparing the classification models, the non-linear RBF kernel SVM outperformed the linear homogeneous kernel map SVM, in rows A2 vs. A3 and A7 vs. A8. Comparing the RBF SVM and the DBM models is not as straightforward. The RBF SVM tends to show slightly better performance in the mean average precision measure, especially in the case of 94 concepts. On the other hand, DBM seems to be better in the rank 50 precision measure. We can also observe that the DBM approach performs slightly better than RBF SVM in the text modality in rows A4 vs. A6. In rows A9 vs. A10 we can see that at least for the 38 concepts, the unimodal DBM model can use the semantic concept vectors more efficiently than the RBF SVM classifier. Rows A18 vs. A20 also show that the semantic concept vectors are more useful inputs to the DBM model than the use of pre-training. Obviously this is due to the strong visual discrimination power of the semantic concept feature.

Finally, based on the results in rows A22–A27, it is evident that the multimodal results are better than either visual or textual unimodal result alone. It also seems in rows A11 vs. A25 that the DBM pretraining with the extra 975,000 images is not necessarily beneficial at all in this dataset. Overall, we can conclude that the DBM and RBF SVM methods are performing equally well within the margins of statistical variation. However, the MAP and Prec@50 measures are more favorable to DBM, while RBF SVM in general shows better performance in the aAUC measure.

5.2.4 Discussion

Table 5.6 shows some examples of concept-wise differences between the unimodal and multimodal results in the MIRFLICKR-1M dataset. The columns titled “row A3” and “row A12” show the MAP values of the corresponding rows in Table 5.4. The following columns show the differences between those two values for the corresponding concepts. On the *baby* and *sea_r1* rows, the differences are positive, which means that the multimodal mean average precision is higher than the visual unimodal. On the other hand, the multimodal approach affects slightly negatively the *clouds* and *tree* concepts. These observations hold for both the RBF SVM method (rows A3 vs. A12) and the Multimodal DBM (rows A15 vs. A25). Actually, we picked in Table 5.6 those two concepts among the set of 38 which displayed the largest absolute positive and negative changes be-

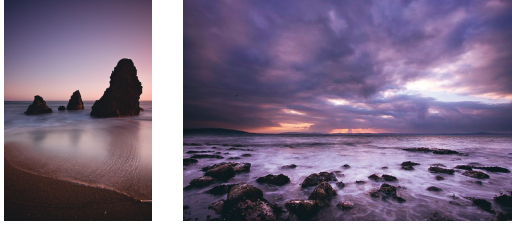


Figure 5.2. Two positive example images for MIRFLICKR-1M concept *sea_r1*. **Left:** Ranking improved with multimodal approach. **Right:** Ranking worsened with multimodal approach. See text for details.

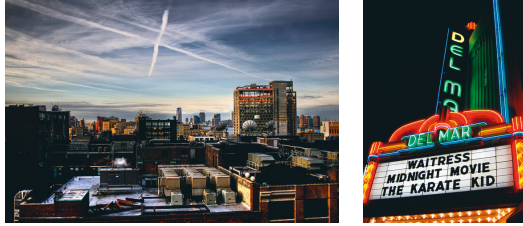


Figure 5.3. Two false positive example images for MIRFLICKR-1M concept *sea_r1*. **Left:** False recognition became less probable with multimodal approach. **Right:** False recognition became more probable with multimodal approach. See text for details.

tween the results in rows A3 vs. A12 and similarly two concepts in rows A15 vs. A25. Of the 38 concepts, 32 benefited from the multimodal approach when the RBF SVM was used, and 30 with the Multimodal DBM. Four concepts, *clouds*, *river_r1*, *sky* and *tree*, suffered from the multimodal approach with both models. We can thus see that even though some concepts suffer in the MAP from the multimodal fusion, this effect is negligible compared to the benefit that some other concepts obtain. Nevertheless, the multimodal approach seems not to be always beneficial for all types of image–text contents.

Figure 5.2 shows two example images of the concept *sea_r1* where the ranking of the image improved (on the left) or worsened (on the right)

concept	row A3	row A12	diff	row A15	row A25	diff
<i>baby</i>	0.451	0.523	0.073	0.449	0.521	0.072
<i>clouds</i>	0.807	0.801	−0.006	0.798	0.796	−0.002
<i>sea_r1</i>	0.452	0.589	0.137	0.488	0.571	0.083
<i>tree</i>	0.773	0.773	−0.000	0.760	0.751	−0.009

Table 5.6. Examples of concept-wise MAP measure differences between unimodal and multimodal results in MIRFLICKR-1M. The row labels refer to the corresponding results in Table 1.

when changing from the visual unimodal method (row A3) to the multi-modal fusion (row A12). The user-given tags for the left image are “beach”, “coast”, “ocean”, “pacific”, “shore”, etc. Most of them really are related to the sea, hence the tags have a positive effect and lead to better ranking of the image. The tags of the right image are “shutter”, “slow”, and “speed”, and they are not related to sea at all. Therefore, the tag information can be regarded as noise and it affects the image’s ranking negatively in this case.

Figure 5.3 shows two example images where false recognition to concept *sea_r1* is becoming either less (on the left) or more (on the right) probable due to the multimodal approach. For the left image, the user-given tags include “buildings”, “city”, “newyork”, “streets”, “urban”, which are clearly not related to sea and make it less probable to classify the image as a “sea view”. On the other hand, for the right image, the tags include “beach”, “cinema”, “coast”, “ocean” and “pacific”, some of which are related to sea. Such tag information thus misleads the multimodal classification and increases the false recognition rate from the visual unimodal case.

To summarize, our examples on the concept and individual image levels show that, inevitably, some concepts and some images benefit and some suffer from the tag-based additional textual input to the multimodal recognition system. On average, however, the gains are larger in magnitude than the losses.

5.3 Multi-label Information Retrieval

In this section, we will study multi-modal information retrieval approaches for the concept detection task in a multi-label dataset. The detailed description of the approach and experimental results can be found in Publication I.

5.3.1 Setup

In Section 5.2, we showed that the multimodal information retrieval approach performs well in single-label recognition task. However, for the multi-label case, does the multimodal approach work as well as in the single-label case? The main goal of the experiments in this section was to find out, how the multimodal information retrieval approach performs in multi-label recognition tasks.

The experimental setting is mostly inspired by [67]. For a reference result, unlike in the original experiments, we also evaluate single-label results in the same setting as in Section 5.2. In order to compare with the results in [67], we decided to use the NUS-WIDE dataset [16].

5.3.2 NUS-WIDE Dataset

The NUS-WIDE dataset has been collected by Chua et al [16] at the National University of Singapore. The dataset consists of 269,648 images with associated tags from Flickr. In total, the images contain 5,018 unique tags, such as "television". In the database the 1,000 most common tags, such as "nature", excluding stop words, have been identified.

Manually annotated ground truth for 81 concept labels is also provided. The 81 concepts are categorized into six super categories of *scene*, *object*, *event*, *program*, *people* and *graphics*. There exist both general concepts, such as *animal*, and specific concepts, such as *dog*. The labels are not mutually exclusive, which means that the database can be used in multi-label recognition experiments. Neither is it required that all images should have any label. Consequently, approximately 22% of the images have no label. The most common single label is *person* followed by *animal*, whereas *clouds+sky* is the most common combination of more than one label. The first 161,789 images are provided as the training data and the remaining 107,859 as the test data.

Six types of low-level visual features (color histogram, color correlogram, edge direction histogram, wavelet texture, block wise color moments, and BoV SIFT) are readily available in the dataset. The Mulan project [31] also provides VLAD features which were used in the experiments of [67]. We have used them too for obtaining comparable results. As the text features in our experiments, we use the 1,000-dimensional binary tag vectors provided in the dataset.

5.3.3 Experiments and Results

When experimenting with the NUS-WIDE dataset, we used as image features the GoogLeNet activation vectors ($L = 2048$) and the VLAD features ($L = 128$) in order to get results comparable with [67]. For the text features, we used the readily available $K = 1000$ tag vocabulary binary vectors. The hidden layer sizes we used with the NUS-WIDE dataset were the same as with the MIRFLICKR-1M dataset except that the first text

	model	image features	text	sem.	MAP	Prec@50	aAUC	ssACC
B1	CBM-LR	VLAD	—	—	—	—	—	0.273
B2	CBM-GB	VLAD	—	—	—	—	—	0.265
B3	DBM	VLAD	—	—	0.126	0.337	0.8290	0.226
B4	DBM	GoogLeNet	—	—	0.426	0.754	0.8638	0.226
B5	RBF	VLAD	—	—	0.135	0.331	0.8410	0.238
B6	RBF	GoogLeNet	—	—	0.384	0.658	0.9425	0.285
B7	DBM	—	text	—	0.508	0.778	0.9380	0.300
B8	RBF	—	text	—	0.432	0.684	0.9292	0.247
B9	DBM	—	—	sem.	0.389	0.713	0.9336	0.325
B10	RBF	—	—	sem.	0.358	0.601	0.9435	0.284
B11	DBM	VLAD	text	—	0.532	0.810	0.9480	0.312
B12	DBM	GoogLeNet	text	—	0.612	0.890	0.9722	0.357
B13	RBF	VLAD 25%	text 75%	—	0.461	0.737	0.9470	0.259
B14	RBF	GoogLeNet 50%	text 50%	—	0.537	0.806	0.9688	0.287
B15	DBM	VLAD	—	sem.	0.393	0.704	0.9423	0.322
B16	DBM	GoogLeNet	—	sem.	0.429	0.753	0.9432	0.328
B17	RBF	VLAD 20%	—	sem. 80%	0.366	0.622	0.9506	0.285
B18	RBF	GoogLeNet 60%	—	sem. 40%	0.399	0.668	0.9466	0.287
B19	DBM	—	text	sem.	0.615	0.881	0.9708	0.369
B20	RBF	—	text 60%	sem. 40%	0.532	0.799	0.9664	0.285
B21	DBM	VLAD	text	sem.	0.618	0.878	0.9718	0.369
B22	DBM	GoogLeNet	text	sem.	0.628	0.893	0.9722	0.368
B23	RBF	VLAD 15%	text 50%	sem. 35%	0.533	0.784	0.9695	0.288
B24	RBF	GoogLeNet 37.5%	text 37.5%	sem. 25%	0.537	0.811	0.9696	0.290

Table 5.7. NUS-WIDE 81 classification results with different models. text = 1,000-dimensional 0/1 tag features, sem. = 500-dimensional semantic concept vectors. Results B1 and B2 are from [67]. Best results in each group are labeled.

hidden layer $\mathbf{h}_b^{(1)}$ was sized $F = 512$ due to the smaller input dimensionality. Also, when using the VLAD or the semantic concept features alone, the image hidden layer sizes for $\mathbf{h}_a^{(1)}$ and $\mathbf{h}_a^{(2)}$ were set to $F = 256$ and $F = 512$, respectively. In these cases, the multimodal joint layer $\mathbf{h}^{(3)}$ had $F = 1024$ hidden units. These configurations were selected because they yielded the best performances in our initial experiments.

Same as the experiments on the MIRFLICKR-1M dataset (Section 5.2), we used the semantic concept vectors ($L = 500$) as additional features. For the DBM training, we randomly split the 161,789 image training set to 151,789 images for the actual model training and kept 10,000 images as validation data. We performed each classification experiment once with the 107,859 test images. With the NUS-WIDE dataset all experiments were performed only once and consequently we do not show error margins.

Table 5.7 shows the experimental results on the NUS-WIDE dataset. The performance is again measured as the mean average precision (MAP)

(see Section 2.4.2), the precision at rank 50 (Prec@50) (see Section 2.4.2), and the average area under the ROC curve (aAUC) (see Section 2.4.3). As the NUS-WIDE dataset is multi-labeled, an additional label subset accuracy (ssACC) [67] measure is used (see Section 2.4.4). In practice, it shows the fraction of recognition where the set of recognized labels matches the ground truth labels exactly. DBM pretraining was not an option with the NUS-WIDE dataset because a large number of auxiliary images with tags were not available as was the case with the MIRFLICKR-1M dataset.

Rows B1 and B2 are the reference results from [67]. B1 used Conditional Bernoulli Mixture (CBM) models trained with linear regression (CBM-LR), while B2 was a CBM model trained with gradient boosting (CBM-GB). The details of these novel CBM approaches can be found in [67]. Rows B1–B6 show the unimodal image classification results with all the used models and features. Rows B7 and B8 show the unimodal text results. Rows B9 and B10 contain results when semantic concept vectors were used as the only feature. Rows B11–B14 show the image–text classification results with no semantic input. Rows B15–B18 are the image results together with the semantic concept vectors. Rows B19 and B20 present the combined text and semantic classification. Finally, rows B21–B24 show the results when all three modalities have been used in recognition.

With this dataset, it seems that both the unimodal DBM models and the Multimodal DBM are generally performing much better than the non-linear RBF SVM – even though the late fusion weights (shown as percentages in the table) for the latter had been optimized in an unorthodox way on the testing data. There is no straightforward explanation to this qualitatively different behavior between the two datasets MIRFLICKR-1M and NUS-WIDE.

In all the experiments with image features, it is obvious that the GoogLeNet features outperform the VLAD features, which of course is no surprise. Consequently, the VLAD features benefit greatly when they are combined together with the text and/or semantic features. Also the GoogLeNet features’ performance is somewhat improved from combination with the semantic features, but not as clearly.

Again, the multimodal methods, with both the Multimodal DBM and RBF SVM with late fusion techniques, are better than any of the unimodal methods. This is especially clear in the Multimodal DBM results with the label subset accuracy which exceeds the reference results in [67] with a



Figure 5.4. **Left:** Correctly multi-label recognized image, whose true labels are *animal* and *flower*. **Right:** Failed multi-label recognition case, whose true labels are *animal*, *grass*, *mountain*, and *sky*.

notable margin.

5.3.4 Discussion

Figure 5.4 shows two examples of multi-label recognition in the NUS-WIDE dataset with the Multimodal DBM. The image on the left is a correctly multi-labeled case. The user-given tags used as the binary text feature for that image are “adult”, “june”, “butterfly”, “interesting”, “quality”, “brown”, “flower”, and “green”. The true and system’s generated labels for it are *animal* and *flower*. Both the visual and textual unimodal DBMs are only capable of recognizing either one of the two labels, but the Multimodal DBM can recognize both labels correctly.

The image on the right in Figure 5.4 is an example of a failed case. It is tagged as “travel”, “california”, “zoo”, “africa”, “family”, “holiday”, “lion”, and “african”. In this case, the classifier could only detect *animal* whereas the ground truth additionally included labels *grass*, *mountain* and *sky*. It seems the classification of background of the image, such as *sky* is a harder task than the recognition of the objects in the image, such as *animal*.

5.4 Multimodal Experiments with Pseudo Tags

Multimodal learning experiments, both unsupervised and supervised, need to have datasets which consist of cross-domain information resources, such as tags and images. It is possible to directly use real-world datasets, such as tagged images uploaded on the world wide web. However, for scientific experiments, we also need to prepare a suitable amount of labeled data. For unsupervised learning tasks, we need the labeled data only for validation and testing. On the other hand, in supervised learning tasks,

the quality and quantity of the training data will affect the results of the experiments. Even when using the same learning model, the result will vary because of differences in the data. For example, if the used dataset contains only images of plants to learn a classifier model for image concepts, the resulting classifier model will be good at classifying concepts related to plants, but not for other concepts. It will also be problematic if only a relatively small amount of training data is available. In this case, the classifier will become restricted. For instance, it could recognize a Volkswagen which is similar to the picture of a Volkswagen in the training data set, but could not classify another otherwise similar Volkswagen which has some different characteristics such as viewing angle, color, type, etc. Therefore, it is always beneficial to have a large amount of well-annotated training data.

Labeled data is usually annotated by humans. It is very time consuming to annotate large datasets with human effort. In multimodal information retrieval studies, it is necessary to have both multimodal queries and corresponding data to assess the performance of the system, because unimodal queries usually cannot be directly applied in the other data domain. As it is very difficult to prepare a perfectly annotated large multimodal dataset, multimodal information retrieval might not work well due to lack of sufficient training data. It is, however, possible to improve the retrieval performance by using the pseudo tag generation approach introduced earlier. In this section, we will show the details of pseudo tag generation and analyse its influence on the results of the multimodal information retrieval task.

5.4.1 Setup

In this section, we study the impact of our new approaches for pseudo tag generation, as described in Sections 4.3.2 and 4.3.3, on multimodal classification results. In this part of the experiments, we applied DBM as the main classifier and used SVM only as a supplemental classifier for the purpose of constructing the similarity map. The purpose of these experiments was to see 1) how the use of GoogLeNet and semantic concept features affects the results of DBM classification, and 2) further impact of our new approach for pseudo tag generation on the classification result. Detailed results of the multimodal DBM search task with pseudo tag generation have been presented in Publication VII.

The multimodal information retrieval task results are comparable to

the single-label results in Table 5.4 of Section 5.2. In the pseudo tag experiments, we only focused on the 38 concept detection task in the MIRFLICKR-1M dataset. We wanted to see the overall performance of the pseudo tag generation process and pick up and analyse some particular examples in which the pseudo tags either improve or make the results worse.

The pseudo tag generation process requires a map or dictionary of cross-domain semantics. In order to produce the semantic similarity map, annotated multimodal data are needed. Wikipedia articles usually contain an example image and a detailed explanation of the topic of the article. That is, the visual concepts of the example image and the latent topic in the article text can be assumed to be semantically related. Therefore, we chose a Wikipedia article dataset as the training data set for the semantic similarity map.

5.4.2 ImageCLEF 2010 Wikipedia Collection

The ImageCLEF 2010 Wikipedia collection consists of 237,434 copyright free images and articles in three language versions: English, German and French, dumped in September 2009 [52]. Basically the dataset is a collection of Wikipedia articles which have versions in all the three languages and are illustrated with at least one image in each version. 70,127 images have only English articles, 50,291 of them have only German articles, 28,461 have only French articles, 26,880 have English and German articles, 20,747 have English and French, 9,646 have German and French and, 22,899 have all the three language versions, whereas 8,144 of them are in other languages, and 239 images have no textual annotation.

The dataset was used in ImageCLEF’s Wikipedia Retrieval task which was one of the evaluation tasks in ImageCLEF 2011. According to ImageCLEF’s website ², this task provided a testbed for the system-oriented evaluation of visual information retrieval from a collection of Wikipedia images and articles. The task was specified with multilingual textual queries and up to five example images describing a user’s multimedia information need. It was assumed that the user wants to find as many relevant images as possible from the Wikipedia image collection. Unfortunately, this task ended in 2011, and we could not thus participate in it. However, we could use the dataset to create the semantic similarity

²<https://www.imageclef.org/>

needed in the experiments presented in this section.

5.4.3 Experiments and Results

In these experiments, which are detailed in Publication VII, we mainly focused on the pseudo tag generation approach which is described in Section 4.3.2. The dimensionality of the textual feature based on the generated supervised pseudo tags was $K = 5631$ and that of the unsupervised pseudo tags was $K = 6193$. The generated pseudo tag vocabulary included the original 2,000 words in [106, 105]. Our semantic concept vectors are 500-dimensional features as in the previous experiments. In these experiments, we concatenated them with the 2048-dimensional GoogLeNet features for the image and multimodal classification tasks and used these vectors ($L = 2548$) as the image inputs \mathbf{u}_a to train the multimodal DBM together with the original binary text feature vectors ($K = 500$). The other settings are the same as in the single-label experiments in Section 5.2.

For the pseudo tag experiments, we used $R = 38$ classes for detection and chose $C_t = 50$ topic clusters for the similarity map. We used the unlabeled MIRFLICKR-1M data for pre-training the DBM, and the ImageCLEF Wikipedia article dataset for training the LDA model described in Section 2.2. For the pseudo tag generation, we extracted for each image the top 20 most matching words by taking into account $n = 8$ top detected image concepts and top $m = 5$ most similar topics for each concept. In order to reduce the running time, we limited to calculate only the top 200 words for each topic.

For the unsupervised pseudo tag generation, we extracted $C_t^A = 50$ LDA topics from the Wikipedia articles. We used these 50 topics as pseudo labels and trained a DBM image classifier on the Wikipedia images. By inferring 50 topics on the MIRFLICKR-1M images we got a vector of the probabilities of the 50 topics in each image. Based on these probabilities, we randomly chose words as the pseudo tags for each topic. For example, if the probability of topic A was 0.02, we randomly picked two words from the top 100 words which had been assigned to topic A. We had also set the maximum of 10 words as a limit for each topic in the experiments.

In Table 5.8, the rows labeled C1–C5 represent the pseudo tag model, whereas the other models whose labels start with A are from the results of Table 5.4 in Section 5.2.

Row C1 is the best text-based unimodal approach in [105]. Row C2 is the result where we have used our proposed supervised method for gen-

	model	MAP	Prec@50
A13	image (PHOW, Gist, MPEG-7) [106]	0.469 ± 0.005	0.803 ± 0.005
A14	image (PHOW, Gist, MPEG-7)	0.475 ± 0.002	0.753 ± 0.002
A15	image (GoogLeNet)	0.727 ± 0.003	0.918 ± 0.004
A16	image (GoogLeNet + semantic concept vectors)	0.731 ± 0.003	0.919 ± 0.005
A18	text	0.511 ± 0.004	0.834 ± 0.005
C1	DBM-GenText [105]	0.531 ± 0.005	0.832 ± 0.004
C2	text + generated pseudo tags	0.684 ± 0.003	0.886 ± 0.005
A20	text + semantic concept vectors	0.734 ± 0.002	0.916 ± 0.004
C3	text + unsupervised pseudo tags	0.528 ± 0.004	0.783 ± 0.006
A23	joint (PHOW, Gist, MPEG-7) [105]	0.609 ± 0.004	0.873 ± 0.004
A24	joint (PHOW, Gist, MPEG-7) [106]	0.641 ± 0.004	0.888 ± 0.004
A25	joint (GoogLeNet)	0.748 ± 0.003	0.919 ± 0.005
C4	joint (GoogLeNet + generated pseudo tags)	0.746 ± 0.003	0.925 ± 0.005
A27	joint (GoogLeNet + semantic concept vectors)	0.753 ± 0.003	0.925 ± 0.005
C5	joint (GoogLeNet + unsupervised pseudo tags)	0.717 ± 0.004	0.917 ± 0.005

Table 5.8. Classification results with different features and uni- and multimodal models. Results of [106] were obtained using additionally sparsity, fine-tuning and dropout. Best results in each group are bolded.

erating additional pseudo tags with the similarity map. Row C3 is the result with the pseudo tags generated by using the unsupervised approach. The additional pseudo tags with the similarity map (row C2) and the semantic concept vectors (row A20) show significant improvements over the baseline rows A18 and C1, but the unsupervised approach for generating additional pseudo tags (row C3) did not show any improvement. In the similarity map approach for pseudo tag generation we use the visual concept classification results for each image. Therefore, it is not fully appropriate to directly compare similarity map result with the original purely unimodal results. However, we can at least conclude that the use of the pseudo tags improves the classification results.

Row C4 shows the result with GoogLeNet features and tags which include the generated pseudo tags as the input. Row C5 is the joint classification result with the GoogLeNet features and composite unsupervised binary vectors as the input. In the unimodal test case, the use of additional information showed better performance than the baseline, but comparing the multimodal results (rows A25, C4, A27 and C5), the additional information from the pseudo tags did not show that much improvement. In the case of unsupervised tags, the multimodal approach actually made the results worse.



Detected concepts	"structure," "indoor," "male," "sky," "people," "female," "night," "people_r1"
Original tags	<i>night, uk, tower, university, Birmingham, clock</i>
Generated tags	<i>web, city, people, world, building, university, station, news, air, national, American, buildings, tower, thumb, road, population, public, rail, bridge, york</i>
Unsupervised tags	<i>tombs, bomb, paris, platform, black, easter, kingdom, railways, arab, michael, commonwealth, chinatown, mosques, surrounding, outside, courtyard, institute, ancient, union, philadelphia, ceiling, construction, project, called, memorial, parish, european, congress, centre, students, coat, ...</i>

Figure 5.5. Example of pseudo tag generation for a tagged image.

Unfortunately we cannot directly compare the exact running times of the different approaches because the Multimodal DBM was implemented for a GPU whereas the SVM experiments were run on conventional CPUs. With a Tesla M2090 GPU, the unimodal processing of 25,000 images or tag vectors took 58 seconds, and that of the multimodal data took 67 seconds.

5.4.4 Discussion

Figures 5.5 and 5.6 show examples of the pseudo tag generation where the similarity map between the visual concepts and the text topics has been used. The tower picture example in Figure 5.5 has the original Flickr tags shown in the figure. As can be seen, the visual DBM concept detection of this image is not performing especially well. However, two out of the six original tags, *university* and *tower*, are seen in the generated tags. Concerning the unsupervised tags, there are several tags which are related to the image, but do not appear in the original tags, such as *courtyard*, *institute*, *surrounding* and *outside*. However, there are many tags which do not correspond to the images, such as human names, and also wrong names for the places.



Detected concepts	"transport," "structure," "car_r1," "plant_life," "car," "tree," "people," "female"
Original tags	(no tags)
Generated tags	<i>car, cars, city, world, web, auto, engine, aircraft, convert, transport, automobile, national, station, grand, air, thumb, university, vehicle, airport, vehicles</i>
Unsupervised tags	<i>control, mini, fuselage, people, developed, energy, modern, motor, bomber, children, market, ground, america, panther, manufacturers, mechanism, production, hat, brown, caliber, nature, ford, flag, novem, ber, francis, tourism, steel, received, vice, industry, mp, military, soviet, grenade, history, ...</i>

Figure 5.6. Example of pseudo tag generation for an image with no tags.

The car picture example in Figure 5.6 does not have any useful original Flickr tags. For this image, the detection of visual concepts performs well, and the detected concepts can be seen in the figure. Most of the generated tags are also related to the object in the picture, but some of them, such as *aircraft*, which is related to a car, but is semantically too far, might cause problems in the classification task. In the unsupervised tags, there is no explicit word "car", but we can see related words, such as *steel*, *fuselage*, *mechanism*, *ford* and *motor*. However, like the tags generated with the similarity map, there are also some words like *military*, which are related to a car, but semantically too far in a common sense. It is possible that this noisy information might have bad influence on the joint classification results.

Figure 5.7 shows a positive example image for the MIRFLICKR-1M concept "sea_r1" and the corresponding retrieval results without and with the generated tags. In the table, the left value is the content-based retrieval rank of the image for the concept and the value in the parentheses is its probability estimated by DBM. Because different models are used, the probability values are not directly comparable, but the ranks are. Nevertheless, we can still see the tendency. In the table, the visual concept classifier's output for "sea_r1" is moderate (around 0.5 on the average)

and most of the original tags are appropriate for "sea_r1". After joining the image and text modalities, the result outperforms the unimodal cases both without and with the generated tags. That is, the semantic information of the image and text modalities complement each other. However, the multimodal model with the generated tags did not improve much in the overall MAP results. As we are already using the visual concept classification results to generate the additional pseudo tags, the multimodal approach has not increased the performance much more.

Figure 5.8 shows a positive example image for the concept "animals" and the corresponding retrieval results. According to the numbers, the visual concept classifier for "animals" has performed well. On the other hand, the original tag, *bilbao*, is not appropriate. The generated tags include some correct tags, but some unrelated words have also been generated. After joining the textual modality to the visual, either without or with the generated tags, the ranking is dropped. In the case without generated tags, it is obvious that the text classification results are negative because of the inappropriate original tag. In the other case, the generated tags were dependent on the visual semantic information and the negative effect of the misleading original tag is quite limited. In addition, most of the semantic information from the generated tags already exists in the visual model.

Figure 5.9 shows a badly-performing example image for the concept "indoor" and the corresponding retrieval results. The image-based results are not as good as the text-only results. In this case, the generated pseudo tags were thus based on low-quality visual results. Therefore, the generated tags are of poorer quality compared to the two previous examples and they actually have had bad influence on the retrieval results.

In these experiments, we have thus proposed and used a similarity map between visual concepts and latent topics in text articles by combining a DBM model with GoogLeNet and semantic concept features and an LDA topic model. The image-only and multimodal classification with the new features performed well and most of the generated pseudo tags were related to the visual contents. When using our pseudo tags for classification, the unimodal text results improved significantly. This means that our model could correctly transfer the similarity and semantic information from the visual space to the textual space.

If the visual classification performs badly, the generated tags are not reliable and not much improvement can be seen in the classification. In



	without tags	with tags
image	30 (0.501)	
text	52 (0.311)	26 (0.454)
joint	25 (0.555)	21 (0.548)

Figure 5.7. Example of an image that contains concept "sea_r1" and its retrieval results without and with the generated tags. Original tags: *beach, ocean, coast, pacific*, etc. Generated tags: *convert, air, island, river, water, airport, aircraft, park, sea, islands, international, lake*, etc.



	without tags	with tags
image	9 (0.998)	
text	8083 (0.018)	243 (0.975)
joint	447 (0.944)	14 (0.998)

Figure 5.8. Example of an image that contains concept "animals" and its retrieval results without and with the generated tags. Original tag: *bilbao*. Generated tags: *world, city, people, university, american, island, building, film, history, animal, animals, location, king, author, book, modern, species, dog, dogs*.

our experiments, we used the same visual concept classifier for creating the similarity map and to predict the visual concepts of the input images. This is one reason why the semantics of our generated pseudo tags are overlapping well with the semantics of the visual concept classification.

Because the latent topic model was trained with Wikipedia articles, many of the words in the topics are not related to any concepts. Hence, some of the generated tags, such as *news*, from the topic concept map are not reliable to be used for the visual classification task. It is also diffi-



	without tags	with tags
image	3370 (0.4473)	
text	828 (0.7875)	2648 (0.613)
joint	1560 (0.8414)	1685 (0.8134)

Figure 5.9. Example of an image that contains concept "indoor" and its retrieval results without and with the generated tags. Original tags: *toys, robot*. Generated tags: *people, news, transport, city, thumb, airport, station, grand, war, engine, cars, aircraft, world, building, convert, car, university, air, american, road*.

cult to distinguish what is related or not. For example, *car* and *aircraft* are both *vehicle*, but if a user wants to get a picture of a *car*, *aircraft* is not a useful keyword. One solution for rejecting inappropriate words and annotating with useful keywords could be the reinforcement learning approach: If an additional tag seems to have a negative effect, it is removed tentatively and the classification is performed again. If the results after removing the tag are not changed or show better performance, the tag is removed permanently. One could also apply a similar method when adding a new tag for an image. A problem with this approach is that it is time consuming and we would need to know the correct classification results for modeling the tag selections.

When making the topic–concept similarity map, we directly calculate the similarity between the distributions of the LDA hidden topics and the image concepts. However, if we would prefer to take into account the correlation between them, applying the Canonical Correlation Analysis (CCA) [49, 50, 35] might be a better choice and improve the quality of the map.

We used the WordNet path similarity when calculating the similarity between tags and visual concepts. This is beneficial for filtering out useless words, such as numbers. However, this technique also filters out useful words which are not in the WordNet dictionary, such as persons’ names. Whether this is good or bad, depends on the user’s demands, but a proper noun is usually an important factor for visual classification. Instead of directly using the WordNet similarity, we could also consider more flexible filtering methods. Furthermore, when utilizing WordNet, we implicitly assumed that all the tags were written in the English language. However, in a real-world situation, the tags might be written in any number of languages. The visual similarity is nevertheless independent of the languages used in the tags, and our previous work [101] has shown that one can even learn word correspondences between languages based on the visual similarity of objects being tagged.

We also experimented with unsupervised tag generation, but compared to the similarity map approach the classification results were worse. However, even with this unsupervised method the classification results with the generated tags were at least slightly better than those with the original tags only.

In the performed experiments, we did not consider the probability distribution of the hidden topics, the word frequencies during the tag choosing

phase, nor the word relatedness to visual objects. If we chose the tag words more precisely by using one or some of these techniques, we could get better results. We might also be able to improve the results by using the LDA hidden topics directly as an additional feature vector, similarly to the semantic concept vectors. Similar to the supervised similarity map approach, we would additionally need to have a better solution for excluding words which are in general unrelated to visual objects.

6. Conclusions

In this thesis, we focused on performance improvement of multimodal information retrieval. The described research has addressed several unimodal information retrieval methods, several features to optimize the retrieval results, and several combination techniques in the application setup of concept detection in image–text data. We have also experimented with how to reduce the computational cost of real-time video information retrieval. Because the deep neural network approach has shown great impact on multimedia information retrieval, we compared the deep Boltzmann machine (DBM) approach with the support vector machine (SVM) approach for uni- and multimodal information retrieval. We also proposed a new method for automatic construction of pseudo tags with a multimodal architecture and studied its performance. In this chapter, we draw the final conclusions of the experiments and their results.

In unimodal information retrieval of multimedia resources, visual concept detection has attracted a lot of research attention in recent years as a method to facilitate semantic indexing and concept-based retrieval of video content. Because of the size and diversity of real-world data, multimedia retrieval requires large ontologies and corresponding content detectors to support a sufficient variety of queries. Also, machine learning approaches for information retrieval, such as non-linear SVMs, typically result in extremely intensive computational cost. Even though the retrieval accuracy would be sufficient, the non-linear SVM approach is not suitable for real-time large-scale information retrieval applications.

For the semantic concept detection task in video data, we demonstrated the feasibility of using linear SVM classifiers. The accuracy of the linear SVM approach was increased with homogeneous kernel maps. In particular, the kernel map approximation method achieved almost the same performance as the non-linear SVM approach with only a fraction of the

detection time. In the experiments, we combined several features, which leads to increase in the computational cost, but was still feasible for real-time performance as it remained orders of magnitude faster than the conventional non-linear approach. The power mean SVM approach showed equal performance compared to the homogeneous approximation kernel maps with fast training and comparable evaluation times.

We also compared uni- and multimodal DBM models and non-linear SVM classifiers in a multimodal information retrieval setup with image-text data of the MIRFLICKR-1M and NUS-WIDE datasets. In these experiments, we also studied the multi-label recognition problem and the performance of different visual and textual features. For the visual features, the GoogLeNet-based DCNN features outperformed the traditional hand-crafted features in the MIRFLICKR-1M datasets and the low-dimensional VLAD features in the NUS-WIDE datasets. The semantic concept vectors, which had been trained by using auxiliary image-tag data, brought significant improvements to the performance. For the textual features, the word2vec feature representation did not show as good performance as we had expected.

Applying feature fusion of semantic concept vectors and binary tag vectors clearly outperformed the use of binary tag vectors only. Comparing the performances between multimodal DBM and the non-linear SVM approaches, multimodal DBM showed slightly better results overall. The multimodal approaches always gave better results than any uni-modal approach alone. In the particular case of the MIRFLICKR-1M dataset, where the user-given image tags are quite unreliable, the visual domain proved to be the more reliable one in the multimodal recognition task, whereas in the NUS-WIDE dataset the textual features, also derived from user-given tags, proved to be as reliable as the visual features.

We also proposed a method for automatic pseudo tag generation by using a similarity map between visual concepts in images and latent topics in text articles. The image-only and multimodal classification with the deep convolutional neuralnet features, such as GoogLeNet features, performed well and most of the generated pseudo tags were correctly related to the visual contents. When using pseudo tags for classification, the unimodal results improved significantly. This means that the model could correctly transfer the concept similarity and semantic information from the visual space to the textual space. However, the generated tags are not always reliable and not much improvement can be seen in some cases. In our ex-

periments, we fine-tuned the visual classifier for the visual classification task to create the similarity map. This is one reason why the semantics of the generated pseudo tags overlap well with the semantics of the visual concept classification.

On the other hand, because the latent topic model was trained with Wikipedia articles, many of the words in the topics are not at all related to visual concepts. Hence, some of the generated pseudo tags from the topic concept map, such as "news", are not reliable when used in a visual classification task.

Overall, the experiments of this thesis showed that the multimodal information retrieval approaches result in better performance than the unimodal approaches. However, dealing with many information domains at the same time definitely increases the computational complexity. Hence, when working with real-time large-scale data, we still need to think more carefully about optimizing the processes.

References

- [1] Natural language toolkit. <http://www.nltk.org/>, (Date accessed: 27.05.2020).
- [2] David H. Ackley, Geoffrey E. Hinton, and Terrence J. Sejnowski. A learning algorithm for Boltzmann machines. *Cognitive science*, 9(1):147–169, 1985.
- [3] Mohammad H. Amin, Evgeny Andriyash, Jason Rolfe, Bohdan Kulchyt-sky, and Roger Melko. Quantum Boltzmann Machine. *Phys. Rev. X*, 8:021050, May 2018.
- [4] Galen Andrew, Raman Arora, Jeff Bilmes, and Karen Livescu. Deep Canonical Correlation Analysis. In *International Conference on Machine Learning*, pages 1247–1255, 2013.
- [5] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. NetVLAD: CNN Architecture for Weakly Supervised Place Recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
- [6] Relja Arandjelović and Andrew Zisserman. All about VLAD. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.
- [7] Ali Ismail Awad. Fingerprint Local Invariant Feature Extraction on GPU with CUDA. *Informatica*, 37:279–284, 2013.
- [8] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2015.
- [9] Soheil Bahrapour, Nasser M. Nasrabadi, Asok Ray, and William Kenneth Jenkins. Multimodal task-driven dictionary learning for image classification. *IEEE Transactions on Image Processing*, 25(1):24–38, 2016.
- [10] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded up robust features. In *Proceedings of the 9th European Conference on Computer Vision (ECCV) 2006*, May 2006.
- [11] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3(Jan):993–1022, 2003.
- [12] Michael Buckland and Fredric Gey. The relationship between recall and precision. *Journal of the American society for information science*, 45(1):12, 1994.

- [13] Nicolò Cesa-Bianchi, Alex Conconi, and Claudio Gentile. On the generalization ability of on-line learning algorithms. *Information Theory, IEEE Transactions on*, 50:2050 – 2057, 10 2004.
- [14] Sung-Hyuk Cha. Comprehensive survey on distance/similarity measures between probability density functions. *City*, 1(2):1, 2007.
- [15] Yunqiang Chen, Xiang Sean Zhou, and Thomas S. Huang. One-class SVM for learning in image retrieval. In *IEEE International Conference on Image Processing (ICIP 2001)*, volume 1, pages 34–37, Thessaloniki, Greece, October 2001.
- [16] Tat-Seng Chua, Jinhui Tang, Richang Hong, Haojie Li, Zhiping Luo, and Yan-Tao. Zheng. NUS-WIDE: A real-world web image database from National University of Singapore. In *ACM Conference on Image and Video Retrieval (CIVR'09)*, Santorini, Greece., July 8-10, 2009.
- [17] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy, July 2019. Association for Computational Linguistics.
- [18] Navneet Dalal, Bill Triggs, and Cordelia Schmid. Human detection using oriented histograms of flow and appearance. In *Proceedings of the 9th European Conference on Computer Vision (ECCV)- Volume Part II, ECCV'06*, pages 428–441, Berlin, Heidelberg, 2006. Springer-Verlag.
- [19] Peter Deutsch. Archie-a darwinian development process. *IEEE Internet Computing*, 4(1):69–71, 2000.
- [20] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.
- [21] Anind K. Dey. Understanding and using context. *Personal and ubiquitous computing*, 5(1):4–7, 2001.
- [22] Matthijs Douze, Hervé Jégou, Sandhawalia Harsimrat, Laurent Amsaleg, and Cordelia Schmid. Evaluation of GIST descriptors for web-scale image search. In *International Conference on Image and Video Retrieval (CVIR)*, pages 19:1–8, Santorini, Greece, July 2009. ACM.
- [23] John P. Eakins. Towards intelligent image retrieval. *Pattern Recognition*, 35(1):3–14, January 2002.
- [24] Pinho Eduardo, Godinho Tiago, Valente Frederico, and Carlos Costa. A multimodal search engine for medical imaging studies. *Journal of Digital Imaging*, 30:39–48, June 2017.
- [25] Ofer Egozi, Shaul Markovitch, and Evgeniy Gabrilovich. Concept-based information retrieval using explicit semantic analysis. *ACM Transactions on Information Systems (TOIS)*, 29(2):8, 2011.

- [26] Scott E. Fahlman, Geoffrey E. Hinton, and Terrence J. Sejnowski. Massively parallel architectures for al: Netl, thistle, and Boltzmann machines. *AAAI-83109*, 113, 1983.
- [27] Tom Fawcett. An introduction to ROC analysis. *Pattern recognition letters*, 27(8):861–874, 2006.
- [28] Ingo Feinerer and Kurt Hornik. WordNet: WordNet interface, (Date accessed: 27.05.2020). R package version 0.1-14.
- [29] Miriam Fernández, Iván Cantador, Vanesa López, David Vallet, Pablo Castells, and Enrico Motta. Semantically enhanced information retrieval: An ontology-based approach. *Web Semantics: Science, Services and Agents on the World Wide Web*, 9(4):434–452, 2011.
- [30] Flickr.com. <http://www.flickr.com/>, (Date accessed: 27.05.2020).
- [31] Mulan: A Java Library for Multi-Label Learning. <http://mulan.sourceforge.net/>, (Date accessed: 27.05.2020).
- [32] Yoav Freund and David Haussler. Unsupervised learning of distributions on binary vectors using two layer networks. In *Advances in Neural Information Processing Systems*, pages 912–919, 1992.
- [33] Andrea Frome, Greg S. Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Tomas Mikolov, et al. Devise: A deep visual-semantic embedding model. In *Advances in Neural Information Processing Systems*, pages 2121–2129, 2013.
- [34] Yuli Gao, Jianping Fan, Hangzai Luo, Xiangyang Xue, and Ramesh Jain. Automatic image annotation by incorporating feature hierarchy and boosting to scale up SVM classifiers. In *Proceedings of ACM Multimedia '06*, Santa Barbara, USA, October 2006.
- [35] Yunchao Gong, Liwei Wang, Micah Hodosh, Julia Hockenmaier, and Svetlana Lazebnik. Improving image-sentence embeddings using large weakly annotated photo collections. In *European Conference on Computer Vision*, pages 529–545, 2014.
- [36] Google.com. Google image. <http://www.google.com>, (Date accessed: 27.05.2020).
- [37] Carsten Gottschlich, Emanuela Marasco, Allen Y. Yang, and Bojan Cukic. Fingerprint liveness detection based on histograms of invariant gradients. In *IEEE International Joint Conference on Biometrics*, pages 1–7, 2014.
- [38] Tom Griffiths. Gibbs sampling in the generative model of latent dirichlet allocation. Technical report, Stanford University, 2002.
- [39] Jochen Guertler and Thomas Chadzelek. Modification free tagging of business application user interfaces, September 30 2009. US Patent App. 12/571,116.
- [40] Alexander G. Hauptmann, Michael G. Christel, and Rong Yan. Video retrieval based on semantic concepts. *Proceedings of the IEEE*, 96(4):602–622, 2008.

- [41] David Hawking, Nick Craswell, Paul Thistlewaite, and Donna Harman. Results and challenges in web search evaluation. *Computer Networks*, 31(11):1321–1330, 1999.
- [42] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
- [43] Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, 2006.
- [44] Geoffrey E. Hinton and Ruslan R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.
- [45] Geoffrey E. Hinton and Ruslan R. Salakhutdinov. Replicated softmax: an undirected topic model. In *Advances in Neural Information Processing Systems*, pages 1607–1614, 2009.
- [46] Geoffrey E. Hinton, Terrence J. Sejnowski, and David H. Ackley. *Boltzmann machines: Constraint satisfaction networks that learn*. Carnegie-Mellon University, Department of Computer Science Pittsburgh, PA, 1984.
- [47] Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 50–57. ACM, 1999.
- [48] Timo Honkela, Samuel Kaski, Krista Lagus, and Teuvo Kohonen. WEBSOM—self-organizing maps of document collections. In *Proceedings of WSOM'97, Workshop on Self-Organizing Maps, Espoo, Finland, June 4-6*, pages 310–315. Helsinki University of Technology, Neural Networks Research Centre, Espoo, Finland, 1997.
- [49] Harold Hotelling. Relations between two sets of variates. *Biometrika*, 28(3/4):321–377, 1936.
- [50] Su-Yun Huang, Mei-Hsien Lee, and Chuhsing Kate Hsiao. Nonlinear measures of association with kernel canonical correlation analysis and applications. *Journal of Statistical Planning and Inference*, 139(7):2162 – 2174, 2009.
- [51] Mark J. Huiskes and Michael S. Lew. The MIR Flickr retrieval evaluation. In *Proceedings of the 1st ACM International Conference on Multimedia Information Retrieval*, pages 39–43. ACM, 2008.
- [52] ImageCLEF2011. Cross-language image retrieval evaluations. wikipedia retrieval task 2011. <http://www.imageclef.org/>, (Date accessed: 27.05.2020).
- [53] Satoru Ishikawa, Rao Muhammad Anwer, Markus Koskela, and Jorma Laaksonen. PicSOM experiments in TRECVID 2015. In *Proceedings of the TRECVID 2015 Workshop*, Gaithersburg, MD, USA, November 2015.
- [54] Satoru Ishikawa, Markus Koskela, Mats Sjöberg, Rao Muhammad Anwer, Jorma Laaksonen, and Erkki Oja. PicSOM experiments in TRECVID 2014. In *Proceedings of the TRECVID 2014 Workshop*, Orlando, FL, USA, November 2014.

- [55] Satoru Ishikawa, Markus Koskela, Mats Sjöberg, Jorma Laaksonen, Erkki Oja, Ehsan Amid, Kalle Palomäki, Annamaria Mesáros, and Mikko Kurimo. PicSOM experiments in TRECVID 2013. In *Proceedings of the TRECVID 2013 Workshop*, Gaithersburg, MD, USA, November 2013.
- [56] Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hervé Jégou, and Tomas Mikolov. Fasttext.zip: Compressing text classification models. *CoRR*, abs/1612.03651, 2016.
- [57] Cuicui Kang, Shiming Xiang, Shengcai Liao, Changsheng Xu, and Chunhong Pan. Learning consistent feature representation for cross-modal multimedia retrieval. *IEEE Transactions on Multimedia*, 17(3):370–381, 2015.
- [58] Lyndon Kennedy and Alex Hauptmann. LSCOM lexicon definitions and annotations version 1.0. Technical Report #217-2006-3, Columbia University, March 2006. DTO Challenge Workshop on Large Scale Concept Ontology for Multimedia.
- [59] Teuvo Kohonen. *Self-Organizing Maps*. Springer-Verlag, Berlin, 1995.
- [60] Teuvo Kohonen, Erkki Oja, Olli Simula, Ari Visa, and Jari Kangas. Engineering applications of the self-organizing map. *Proceedings of the IEEE*, 84(10):1358–84, 1996.
- [61] Markus Koskela and Jorma Laaksonen. Convolutional network features for scene recognition. In *Proceedings of the 22nd ACM International Conference on Multimedia*, Orlando, FL, USA, November 2014.
- [62] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.
- [63] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.
- [64] Jorma Laaksonen, Markus Koskela, Ville Viitaniemi, and Mats Sjöberg. PicSOM: Content-based visual information retrieval system, 2014. Open-source software.
- [65] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. ALBERT: A lite BERT for self-supervised learning of language representations. In *International Conference on Learning Representations*, 2020.
- [66] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2169–2178, 2006.
- [67] Cheng Li, Bingyu Wang, Virgil Pavlu, and Javed Aslam. Conditional Bernoulli mixtures for multi-label classification. In *Proceedings of the 33rd International Conference on Machine Learning*, pages 2482–2491, 2016.
- [68] Tie-Yan Liu et al. Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval*, 3(3):225–331, 2009.

- [69] Zhiyuan Liu, Yuzhou Zhang, Edward Y. Chang, and Maosong Sun. PLDA+: Parallel latent Dirichlet allocation with data placement and pipeline processing. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(26), March 2011.
- [70] Internet live stats. <http://www.internetlivestats.com/twitter-statistics/>, (Date accessed: 27.05.2020).
- [71] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, November 2004.
- [72] Subhransu Maji, Alexander C. Berg, and Jitendra Malik. Classification using intersection kernel support vector machines is efficient. In *Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2008)*, pages 1–8, june 2008.
- [73] Christopher D. Manning, Hinrich Schütze, et al. *Foundations of Statistical Natural Language Processing*, volume 999. MIT Press, 1999.
- [74] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.
- [75] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.
- [76] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve Restricted Boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning, ICML'10*, page 807–814, Madison, WI, USA, 2010. Omnipress.
- [77] NIST, National Institute of Standards and Technology. <http://trecvid.nist.gov/>, (Date accessed: 27.05.2020).
- [78] Michael Noll and Christoph Meinel. Web search personalization via social bookmarking and tagging. *The Semantic Web*, pages 367–380, 2007.
- [79] Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001.
- [80] Rong Pan, Zhongli Ding, Yang Yu, and Yun Peng. A Bayesian network approach to ontology mapping. *The Semantic Web - ISWC 2005. Lecture Notes in Computer Science*, 3729, 2005.
- [81] Sebastien Paris, Xanadu Halkias, and Herve Glotin. Sparse coding for histograms of local binary patterns applied for image categorization: Toward a bag-of-scenes analysis. In *Proceedings of 21th International Conference on Pattern Recognition (ICPR 2012)*, Tsukuba, Japan, November 2012.
- [82] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

- [83] Haim Permuter, Joseph Francos, and Ian Jermyn. A study of Gaussian mixture models of color and texture features for image classification and segmentation. *Pattern Recognition*, 39(4):695–706, 2006.
- [84] Florent Perronnin and Christopher Dance. Fisher kernels on visual vocabularies for image categorization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.
- [85] Florent Perronnin, Jorge Sánchez, and Yan Liu. Large-scale image categorization with explicit data embedding. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2297–2304, June 2010.
- [86] Shamna Poothari, Govindan V. K., and Abdul Nazeer K. A. Content based medical image retrieval using topic and location model. *Journal of Biomedical Informatics*, 91:103112, 2019.
- [87] Ramon Quiza and J. Paulo Davim. Computational methods and optimization. *Machining of Hard Materials*, pages 177–208, 1 2011.
- [88] Alain Rakotomamonjy. Variable selection using svm-based criteria. *Journal of Machine Learning Research*, 3(Mar):1357–1370, 2003.
- [89] Yong Rui, Thomas S. Huang, and Shih-Fu Chang. Image retrieval: Current techniques, promising directions, and open issues. *Journal of Visual Communication and Image Representation*, 10(1):39–62, March 1999.
- [90] Ruslan Salakhutdinov and Geoffrey Hinton. Deep Boltzmann machines. In *Artificial Intelligence and Statistics*, pages 448–455, 2009.
- [91] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 2234–2242. Curran Associates, Inc., 2016.
- [92] Gerard Salton and Chris Buckley. Improving retrieval performance by relevance feedback. *Readings in Information Retrieval*, 24(5):355–363, 1997.
- [93] Gerard Salton, Anita Wong, and Chung-Shu Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.
- [94] Jorge Sánchez, Florent Perronnin, Thomas Mensink, and Jakob Verbeek. Image classification with the Fisher Vector: Theory and practice. *International Journal of Computer Vision*, 105(3):222–245, 2013.
- [95] Aliaksei Severyn and Alessandro Moschitti. Learning to rank short text pairs with convolutional deep neural networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 373–382. ACM, 2015.
- [96] Josef Sivic and Andrew Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proceedings of ICCV03*, volume 2, pages 1470–1477, October 2003.

- [97] Mats Sjöberg, Satoru Ishikawa, Markus Koskela, Jorma Laaksonen, and Erkki Oja. PicSOM experiments in TRECVID 2011. In *Proceedings of the TRECVID 2011 Workshop*, Gaithersburg, MD, USA, December 2011. Available online at <http://www-nlpir.nist.gov/projects/tvpubs/tv.pubs.org.html>.
- [98] Mats Sjöberg, Satoru Ishikawa, Markus Koskela, Jorma Laaksonen, and Erkki Oja. PicSOM experiments in TRECVID 2012. In *Proceedings of the TRECVID 2012 Workshop*, Gaithersburg, MD, USA, November 2012.
- [99] Mats Sjöberg, Markus Koskela, Satoru Ishikawa, and Jorma Laaksonen. Real-time large-scale visual concept detection with linear classifiers. In *Proceedings of 21st International Conference on Pattern Recognition*, Tsukuba, Japan, November 2012.
- [100] Mats Sjöberg, Markus Koskela, Ville Viitaniemi, and Jorma Laaksonen. Indoor location recognition using fusion of SVM-based visual classifiers. In *Proceedings of 2010 IEEE International Workshop on Machine Learning for Signal Processing*, pages 343–348, Kittilä, Finland, August-September 2010.
- [101] Mats Sjöberg, Ville Viitaniemi, Jorma Laaksonen, and Timo Honkela. Analysis of semantic information available in an image collection augmented with auxiliary data. In *Proceedings of 3rd IFIP Conference on Artificial Intelligence Applications and Innovations*, pages 600–608, 2006.
- [102] Evgeny Smirnov, Denis Timoshenko, and Serge Andrianov. Comparison of regularization methods for imagenet classification with deep convolutional neural networks. *AASRI Procedia*, 6:89–94, 12 2014.
- [103] Paul Smolensky. Information processing in dynamical systems: Foundations of harmony theory. Technical report, DTIC Document, 1986.
- [104] Cees G. M. Snoek, Marcel Worring, and Arnold W. M. Smeulders. Early versus late fusion in semantic video analysis. In *Proceedings of the 13th Annual ACM International Conference on Multimedia*, pages 399–402. ACM, 2005.
- [105] Nitish Srivastava and Ruslan Salakhutdinov. Multimodal learning with deep Boltzmann machines. In *Advances in Neural Information Processing Systems*, pages 2222–2230, 2012.
- [106] Nitish Srivastava and Ruslan Salakhutdinov. Multimodal learning with deep Boltzmann machines. *Journal of Machine Learning Research*, 15:2949–2980, 2014.
- [107] Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. Learning sentiment-specific word embedding for twitter sentiment classification. In *ACL (1)*, pages 1555–1565, 2014.
- [108] Qi Tang. Tagger: Enhance database search tools with de novo sequencing tags. *UWSpace*, 2017.
- [109] Marvin Teichmann, Andre Araujo, Menglong Zhu, and Jack Sim. Detect-to-retrieve: Efficient regional aggregation for image search. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

- [110] Sergios Theodoridis and Konstantinos Koutroumbas. *Pattern Recognition, Fourth Edition*. Academic Press, Inc., USA, 4th edition, 2008.
- [111] Lorenzo Torresani, Martin Szummer, and Andrew Fitzgibbon. Efficient object category recognition using classemes. In *European Conference on Computer Vision (ECCV)*, pages 776–789, September 2010.
- [112] Grigorios Tsoumakas and Ioannis Katakis. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining*, 3(3), 2006.
- [113] Andrea Vedaldi and Andrew Zisserman. Efficient additive kernels via explicit feature maps. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2010)*, 2010.
- [114] Ville Viitaniemi, Mats Sjöberg, Markus Koskela, Satoru Ishikawa, and Jorma Laaksonen. Advances in visual concept detection: Ten years of TRECVID. In Ella Bingham, Samuel Kaski, Jorma Laaksonen, and Jouko Lampinen, editors, *Advances in Independent Component Analysis and Learning Machines, 1st Edition*, pages 249–278. Academic Press, 2015.
- [115] Ville Viitaniemi, Mats Sjöberg, Markus Koskela, and Jorma Laaksonen. Automatic video search using semantic concepts. In *Proceedings of 8th European Conference on Interactive TV and Video (EuroITV 2010)*, Tampere, Finland, June 2010.
- [116] Peter Wittenburg, Hennie Brugman, Albert Russel, Alex Klassmann, and Han Sloetjes. Elan: a professional framework for multimodality research. In *Proceedings of LREC 2006, Fifth International Conference on Language Resources and Evaluation*, 2006.
- [117] WorldWideWebSize.com. <http://www.worldwidewebsite.com/>, (Date accessed: 27.05.2020).
- [118] Jianxin Wu. Power mean SVM for large scale visual classification. In *Proceedings of The IEEE Int'l Conference on Computer Vision and Pattern Recognition (CVPR 2012)*, Providence, USA, June 2012.
- [119] Haoran Xie, Xiaodong Li, Tao Wang, Raymond Y. K. Lau, Tak-Lam Wong, Li Chen, Fu Lee Wang, and Qing Li. Incorporating sentiment into tag-based user profiles and resource profiles for personalized search in folksonomy. *Information Processing & Management*, 52(1):61–72, 2016.
- [120] Jun Yang, Yu-Gang Jiang, Alexander G. Hauptmann, and Chong-Wah Ngo. Evaluating bag-of-visual-words representations in scene classification. In *Proceedings of the International Workshop on Multimedia Information Retrieval*, pages 197–206. ACM, 2007.
- [121] Longqi Yang, Chen Fang, Hailin Jin, Matthew D. Hoffman, and Deborah Estrin. Personalizing software and web services by integrating unstructured application usage traces. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 485–493. International World Wide Web Conferences Steering Committee, 2017.
- [122] Emine Yilmaz, Evangelos Kanoulas, and Javed A. Aslam. A simple and efficient sampling method for estimating AP and NDCG. In *Proceedings*

- of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '08)*, pages 603–610, 2008.
- [123] Jun Yu, Xiaokang Yang, Fei Gao, and Dacheng Tao. Deep multimodal distance metric learning using click constraints for image ranking. In *IEEE Transactions on Cybernetics*, volume 47, pages 4014–4024, 2017.
- [124] Shipeng Yu, Deng Cai, Ji-Rong Wen, and Wei-Ying Ma. Improving pseudo-relevance feedback in web information retrieval using web page segmentation. In *Proceedings of the 12th International Conference on World Wide Web*, pages 11–18. ACM, 2003.
- [125] Konstantinos Zagoris, Avi Arampatzis, and Savvas A. Chatzichristofis. www.mmretrieval.net: A multimodal search engine. In *Proceedings of the 3rd International Conference on Similarity Search and Applications*, 2010.
- [126] Fang Zhao, Yongzhen Huang, Liang Wang, and Tieniu Tan. Deep semantic ranking based hashing for multi-label image retrieval. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.
- [127] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 487–495. Curran Associates, Inc., 2014.
- [128] Xi Zhou, Kai Yu, Tong Zhang, and Thomas Huang. Image classification using super-vector coding of local image descriptors. In *Proceedings of European Conference on Computer Vision (ECCV 2010)*, 2010.
- [129] Mu Zhu. Recall, precision and average precision. *Department of Statistics and Actuarial Science, University of Waterloo, Waterloo*, 2:30, 2004.



ISBN 978-952-60-3952-7 (printed)

ISBN 978-952-60-3954-1 (pdf)

ISSN 1799-4934 (printed)

ISSN 1799-4942 (pdf)

Aalto University
School of Science
Computer Science
www.aalto.fi

**BUSINESS +
ECONOMY**

**ART +
DESIGN +
ARCHITECTURE**

**SCIENCE +
TECHNOLOGY**

CROSSOVER

**DOCTORAL
DISSERTATIONS**