Semi-Supervised Learning for Image Classification

A dissertation for the degree of
Doktor-Ingenieur (Dr.-Ing.)

approved by
Saarland University
Computer Science

presented by

Sandra Ebert
Dipl.-Inform.

born in Leipzig, Germany

Saarbrücken, 2012

Supervisor
Prof. Dr. Bernt Schiele
Max Planck Institute for Informatics, Germany

Dean
Prof. Dr. Mark Groves
Saarland University, Germany

Reviewer
Prof. Dr. Bernt Schiele
Max Planck Institute for Informatics, Germany

Prof. Dr. Horst Bischof
TU Graz, Austria

Examination
Chair: Prof. Dr. Matthias Hein
Saarland University, Germany

Examiner: Prof. Dr. Bernt Schiele
Max Planck Institute for Informatics, Germany

Co-Examiner: Prof. Dr. Horst Bischof
TU Graz, Austria

Academic member: Dr. Mario Fritz
Max Planck Institute for Informatics, Germany

Date of Submission: 25. October 2012
Date of Defense: 14.December 2012

ABSTRACT

Object class recognition is an active topic in computer vision still presenting many challenges. In most approaches, this task is addressed by supervised learning algorithms that need a large quantity of labels to perform well. This leads either to small datasets ($< 10,000$ images) that capture only a subset of the real-world class distribution (but with a controlled and verified labeling procedure), or to large datasets that are more representative but also add more label noise. Therefore, semi-supervised learning is a promising direction. It requires only few labels while simultaneously making use of the vast amount of images available today. We address object class recognition with semi-supervised learning. These algorithms depend on the underlying structure given by the data, the image description, and the similarity measure, and the quality of the labels. This insight leads to the main research questions of this thesis: "Is the structure given by labeled and unlabeled data more important than the algorithm itself?", "Can we improve this neighborhood structure by a better similarity metric or with more representative unlabeled data?", and "Is there a connection between the quality of labels and the overall performance and how can we get more representative labels?". We answer all these questions, i.e., we provide an extensive evaluation, we propose several graph improvements, and we introduce a novel active learning framework to get more representative labels.

ZUSAMMENFASSUNG

Objektklassifizierung ist ein aktives Forschungsgebiet in maschineller Bildverarbeitung was bisher nur unzureichend gelöst ist. Die meisten Ansätze versuchen die Aufgabe durch überwachtes Lernen zu lösen. Aber diese Algorithmen benötigen eine hohe Anzahl von Trainingsdaten um gut zu funktionieren. Das führt häufig entweder zu sehr kleinen Datensätzen ($< 10,000$ Bilder) die nicht die reale Datenverteilung einer Klasse wiedergeben oder zu sehr grossen Datensätzen bei denen man die Korrektheit der Labels nicht mehr garantieren kann. Halbüberwachtes Lernen ist eine gute Alternative zu diesen Methoden, da sie nur sehr wenige Labels benötigen und man gleichzeitig Datenressourcen wie das Internet verwenden kann. In dieser Arbeit adressieren wir Objektklassifizierung mit halbüberwachten Lernverfahren. Diese Algorithmen sind sowohl von der zugrundeliegenden Struktur, die sich aus den Daten, der Bildbeschreibung und der Distanzmasse ergibt, als auch von der Qualität der Labels abhängig. Diese Erkenntnis hat folgende Forschungsfragen aufgeworfen: "Ist die Struktur wichtiger als der Algorithmus selbst?", "Können wir diese Struktur gezielt verbessern z.B. durch eine bessere Metrik oder durch mehr Daten?" und "Gibt es einen Zusammenhang zwischen der Qualität der Labels und der Gesamtperformanz der Algorithmen?". In dieser Arbeit beantworten wir diese Fragen indem wir diese Methoden evaluieren. Ausserdem entwickeln wir neue Methoden um die Graphstruktur und die Labels zu verbessern.

ACKNOWLEDGEMENTS

My supervisor Bernt Schiele taught me to always do my best, to never give up, and changed my way of dealing with constructive criticism. Our brainstormings were always full of new ideas, our meetings so far from “cumbersome” that mediating tools such as *LaughingLily* (Antifakos and Schiele, 2003) were never required. Bernt always had a very open ear for our problems and concerns, he was very convincing and very passionate in every discussion, being a role model in his fairness towards all of his students, very quickly understanding our methods and the main issues with these. He also talks, eats and runs very very quickly and, most importantly, never stopped telling me how to use the word "very" very sparingly.

Horst Bischof has influenced my thinking and thus the outcome of this thesis at several conferences by asking questions crucial to finding new points of view. Matthias Hein’s publications provided me with a deeper understanding of the methods used in this work, helping me to strengthen the reasoning behind the present thesis.

Mario Fritz kept me busy on the occasion of several submissions with his seemingly endless number of ideas, while alway helping me to survive all deadlines with a smile on my face. Diane Larlus was faced with the great and sometimes painful challenge to write a paper with me as a first year PhD student. I have learned from her how to structure a paper, how to design experiments, and how to present a work at conferences.

Christian Wojek was a stabilizing element in our group when everybody else was moving around frantically, as before deadlines. Michael Stark advised me during my master thesis, gave me valuable suggestions on how to improve my writing, and always made sure that we never went late for lunch. Paul Schnitzspan acted as a translator from *Berntspeak* to normal speed German until my ears had adapted to the verbal pace of Bernt. Micha (Mykhaylo) Andriluka fascinated me with his way of questioning everything and everyone. Peter Gehler constantly asked me about why we need semi-supervised learning at all. Kristof Van Laerhoven turned all our coffee breaks back at TU Darmstadt into a laughing event. Ulf Blanke, while working on a totally different topic, was my discussion partner for all everyday PhD problems during my thesis. Marcus Rohrbach showed me how to keep one’s self-confidence in the face of harsh criticism and how to shrug off negative comments. Fabio Galasso taught me how to pronounce, prepare, and enjoy really good caffè. Finally, I had the great pleasure to co-supervise Jan Hosang during his master thesis together with Bastian Leibe and Tobias Weyand from RWTH Aachen.

All group members at TU Darmstadt, i.e., Uschi Paeckel, Andreas Zinnen, Ulrich Steinhoff, Stefan Walk, Nikodem Majer, Maja Stikic, and the Max Planck Institute for Informatics, i.e., Anja Weber, Cornelia Balzert, Leonid Pishchulin, Bojan Pepik, Maksim Lapin, Martin Šimonovský, Siyu Tang, Wei-Chen Chiu, Mateusz Malinowski,

Anna Senina, showed me how to combine work and leisure. Additionally, also some group members of the ICG at TU Graz, i.e., Sabine Sternig, Hayko Riemenschneider, and Martin Godec, made sure that I never felt alone at conferences and at the computer vision summer school in Sicily.

During my function as senator and other political activities at the TU Darmstadt, I have learned to never take up a position too early before hearing all parties in the dispute, how to mediate among different people and their opinions, and that communication is the key to a successful negotiation.

Sarah Bolius fascinated me by her self-confident, passionate and direct manner to tackle problems. She turned each coffee break and soup lunch in a relaxing and laughing event that made me forget all my worries for a while. Besides more philosophical discussions about the life in general, Gunter Grieser provided me with insight regarding the logical aspects of machine learning. Finally, my family and my sister gave me constant support by asking questions about my research and its usefulness for everyday life.

Last but not least, my husband Roland Schleser constantly and untiringly supported and motivated me during the entire time, keeping in mind the ups and downs of his time as a PhD student. He encouraged me with his calm yet cheerful nature, especially during less successful phases of my work, and provided me with deep insight from a physicist's point of view.

From Aurelius (180) I learned that we are the result of the thoughts and the behaviors of our fellow human beings and our own interpretations. In that sense, I would like to thank all previously mentioned people as they all contributed more or less directly to the becoming of this thesis.

CONTENTS

1 Introduction 1

1.1 Why do we need semi-supervised learning? 4
1.2 Challenges of semi-supervised learning 6
  1.2.1 Structural problems 6
  1.2.2 Labeling issues 15
1.3 Contributions of this thesis 16
1.4 Outline 17

2 Related Work 19

2.1 Object class recognition 19
  2.1.1 Image description 20
  2.1.2 Object class learning 21
  2.1.3 Relation to own work 23
2.2 Semi-supervised learning 23
  2.2.1 Cluster assumption 24
  2.2.2 Manifold assumption 25
  2.2.3 Graph-based methods 26
  2.2.4 Relation to own work 27
2.3 Graph improvement 27
  2.3.1 Scalable algorithms 27
  2.3.2 Metric learning 29
  2.3.3 Graph construction 31
2.4 Active Learning 34
  2.4.1 Exploitation-driven methods 35
  2.4.2 Exploration-driven methods 36
  2.4.3 Trade-off between exploration and exploitation 37
  2.4.4 Relation to own work 38

3 Graph-based semi-supervised learning 39

3.1 Graph construction 39
  3.1.1 Image representation 40
  3.1.2 Image Similarity 41
  3.1.3 Graph construction 42
3.2 Classification Algorithms 43
3.3 Datasets 44
3.4 Experiments 46
3.5 Conclusion 49

4 Graph Improvement 51
4.1 Introduction . . . 52
4.2 Unsupervised Improvements . . . 53
4.2.1 Dimensionality Reduction with PCA . . . 53
4.2.2 Increasing the dataset . . . 54
4.2.3 Combination . . . 55
4.2.4 Summary . . . 56
4.3 Supervised Improvements . . . 57
4.3.1 Dimensionality Reduction with LDA . . . 58
4.3.2 SVM-based Graph Construction . . . 58
4.3.3 Information theoretic metric learning (ITML) . . . 60
4.3.4 Summary . . . 62
4.4 Semi-Supervised Improvement with IMLP . . . 64
4.5 Conclusion . . . 66

5 Label Improvement by Active Learning 67
5.1 Introduction . . . 68
5.2 Sampling Criteria . . . 70
5.2.1 Previous exploitation criteria . . . 71
5.2.2 Previous exploration criteria . . . 72
5.2.3 Graph density criteria for exploration . . . 73
5.2.4 Experiments . . . 75
5.3 Time-varying Active Learning . . . 76
5.3.1 Method . . . 76
5.3.2 Analyzing the trade-off on synthetic datasets . . . 77
5.3.3 Experiments on real data . . . 79
5.4 RALF - A Reinforced Active Learning Formulation . . . 81
5.4.1 Switching feedback-driven between two criteria . . . 81
5.4.2 Reinforced active learning formulation (RALF) . . . 82
5.4.3 Experiments . . . 84
5.5 Conclusion . . . 86

6 Active Metric Learning 89
6.1 Introduction . . . 90
6.2 Methods . . . 90
6.2.1 Metric learning . . . 91
6.2.2 Active sample selection . . . 91
6.2.3 Classification algorithms . . . 92
6.3 Active metric learning . . . 93
6.3.1 Batch active metric learning (BAML) . . . 93
6.3.2 Interleaved active metric learning (IAML) . . . 94
6.4 Experiments . . . 94
6.4.1 Different sampling criteria for metric learning . . . 94
6.4.2 BAML with LP and SVM . . . 97
6.4.3 Interleaved active metric learning (IAML) . . . 97

6.5 Conclusion 98

7 **Active Dataset Construction** **99**
7.1 Introduction 100
7.2 Related work 101
7.3 General SSL-framework 102
  7.3.1 Label propagation (LP) 102
  7.3.2 Active Learning (AL) 102
7.4 Graph enhancement techniques 103
  7.4.1 Previous techniques 103
  7.4.2 Novel techniques to enrich graph structure 104
  7.4.3 Runtime complexity 105
7.5 Experiments 107
7.6 Conclusion 111

8 **Conclusion and Future Perspectives** **113**
8.1 Discussion of contributions 113
  8.1.1 Contributions towards better structure 114
  8.1.2 Contributions towards better labels 115
8.2 Future perspective 116
  8.2.1 What can we learn from human object recognition? 116
  8.2.2 Beyond human perception, learning, and inference 123

List of Figures **125**

List of Tables **129**

Bibliography **131**

Curriculum Vitae **161**

Publications **163**

INTRODUCTION 1

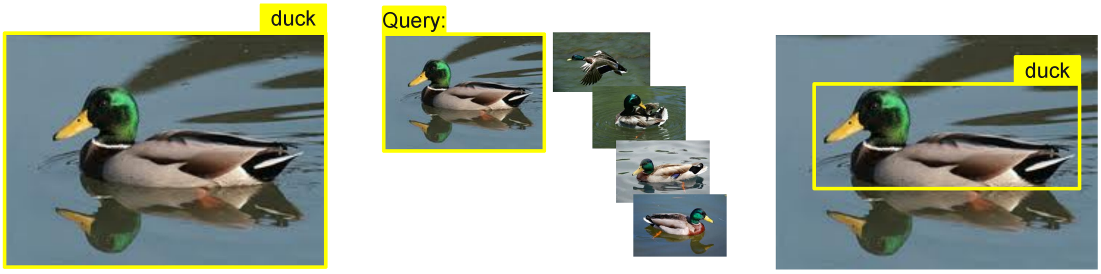
Contents

1.1 Why do we need semi-supervised learning? **4**
1.2 Challenges of semi-supervised learning **6**
  1.2.1 Structural problems 6
  1.2.2 Labeling issues 15
1.3 Contributions of this thesis **16**
1.4 Outline **17**

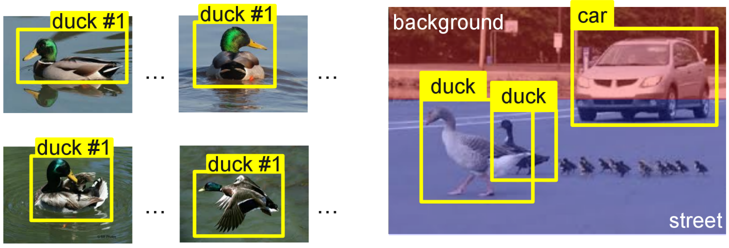
Object recognition is one of the central topics in computer vision and an integral part of many computer vision tasks. To mention only a few, image classification (Figure 1.1 a) is one of the more basic tasks that includes object recognition to classify an image, e.g. *duck*. Content-based Image retrieval (Figure 1.1 b) contains object recognition to search systematically for images that contain these objects. Object detection (Figure 1.1 c) must in addition specify the actual position of the recognized object in the image (marked as a bounding box), thus a clear separation between foreground and background is essential. Tracking (Figure 1.1 d) is based on object detection and tries to track the localized object across a sequence of frames. Finally, scene understanding (Figure 1.1 e) aims to capture the whole scene including all interactions among objects and the environment, e.g., to warn the car driver before an accident happens with the ducks. This list of computer vision tasks could be continued arbitrarily. But although object recognition is a crucial part it is surprising that even image classification, which only aims to provide the image label, does not provide satisfactory results on more challenging datasets.

In contrast, humans can quickly and accurately recognize objects in images or video sequences. They can categorize them into thousands of categories (Biederman, 1987). Beyond that they can track objects in videos and they are able to interpret the entire scene and to infer subsequent events. Of course, human perception, recognition, and inference also have their limits but they might serve as a good starting point to improve upon. Therefore it is not surprising that computer vision, in particular the machine learning part of it, is mainly driven by cognitive science – the science of understanding the learning and thinking of humans. But as controversial theories in cognition are, as diverse are the approaches in computer vision and machine learning.

One of these long-lasting debates in cognition focuses on the question whether a human learns exemplar-based or concept-driven. The exemplar-based model (Medin and Schaffer, 1978; Nosofsky, 1984; Zaki *et al.*, 2003) assumes that humans store a list of exemplars for each category. A category decision will be made based on



a) image classification b) image retrieval c) object detection



d) tracking e) scene understanding

Figure 1.1. Computer vision applications with object recognition as an integral part.

a similarity to existing exemplars. One of the most prominent representative in machine learning is the k nearest neighbor classifier visualized in Figure 1.2 a). This classifier looks for the nearest labeled neighbor in the training set marked as red and blue data point and uses the label of this training sample for classification. In contrast, concept-driven learning assumes that people abstract to a model or a prototype that is used for classifying objects (Posner *et al.*, 1967; Minda and Smith, 2001; Murphy, 2002). This paradigm can be found in many algorithms that learn a model of a category and do any kind of generalization such as SVM that learns a decision boundary (Figure 1.2 b). More recently, there is a tendency towards the theory that humans use either multiple learning systems in parallel (Erickson and Kruschke, 1998; Ashby and Maddox, 2011) or a hybrid version that groups exemplars around a general concept. This approach can be found for example in a combination of metric learning and KNN that first maps the exemplars to a more discriminative description, i.e., examples of the same class are closer together visualized as red and blue area in Figure 1.2 c), and then applies KNN.

Another equally controversial but much older debate revolves around the question how we gain the insight that forms the base of our decisions. This leads to one of the most fundamental questions in machine vision: whether and how much supervision do we need? On one hand, a human learns provable faster with supervised feedback (Ashby, 1992). Therefore, it is not surprising that state-of-the-art performance is achieved by supervised algorithms. However, this success has to be put into perspective as the dataset construction itself contains an enormous amount of supervision. Each method is only as good as the underlying training data. If the learner sees only the side view of a car during training, then the resulting classifier will fail on cars shown in front views or from above. This aspect is often neglected

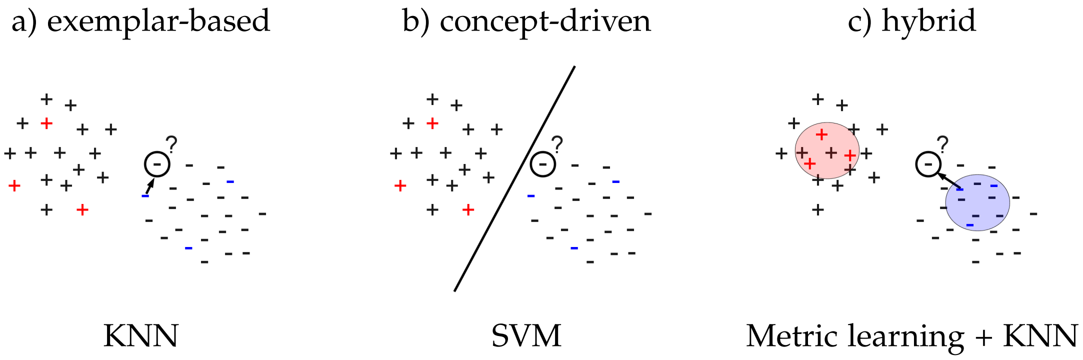


Figure 1.2. Illustration of several learning principles that are used to classify the marked unlabeled data point: a) exemplar-based learning, e.g., KNN classification, b) concept-driven, e.g., SVM, or c) hybrid approach, e.g., that groups exemplars around a general concept by transformation with metric learning and than applying KNN. Blue and red points are the labels of two different classes, and black points are the unlabeled data.

in the subsequent evaluation and leads to datasets that are either small and strongly biased (Ponce *et al.*, 2006; Torralba, 2011) or large and error-prone (Welinder *et al.*, 2010). On the other hand, it is also clear that many human decisions are driven to some extent by intuition, i.e. more or less unsupervised, particularly in unfamiliar or risky situations (Kahneman and Tversky, 1979). But although unsupervised learning is an important research area (Weber *et al.*, 2000; Sivic *et al.*, 2005), e.g. for object discovery or novelty detection, a minimum level of supervision is required at the end to judge the quality and to gain insight. Therefore, semi-supervised learning (SSL) seems to be the paradigm to address these drawbacks by using labels as well as the structure or geometry extracted from the labeled as well as the unlabeled data. In fact, SSL comes indisputable with both a practical relevance and a certain charm. The actual goal is an approach that produces results that is at least comparable to human recognition with as little supervision and interactions as possible from humans.

This thesis ventures into a research area that probably reflects the human thinking and decision making better than other learning paradigm, even under the premise that the gap between human and computer thinking will never be closed (Penrose, 1989). This thesis focuses on the semi-supervised learning with application on image classification. Therefore we are *only* dependent on the image description and the similarity measure. The remaining part of this introduction is structured as follows: We start with a more detailed discussion about semi-supervised learning in Section 1.1 and in particular why we should spend more effort in this highly disputable research area. After that we outline the main challenges in Section 1.2 that come with semi-supervised learning in combination with image classification. Section 1.3 gives a summary of the problems we address. Finally, we provide in Section 1.4 the overall structure of this thesis.

1.1 WHY DO WE NEED SEMI-SUPERVISED LEARNING?

> *"Mind without structure is empty and*
> *perception without labels is blind."*
> translated from Kant (1781)

Semi-supervised learning make use of both labeled and unlabeled data. One of the first steps towards semi-supervised learning was self training (Scudder, 1965; Fralick, 1967; Agrawala, 1970), a method that uses the predictions of unlabeled data to improve the overall classification performance. These approaches were followed by works of transductive learning (Vapnik and Sterin, 1977). They use unlabeled data to regularize the decision boundary. Shortly after, there were many works towards algorithmic contributions ranging from generative models (Nigam *et al.*, 1999; Seeger, 2001) over several kinds of discriminative models (Joachims, 1999; Grandvalet and Bengio, 2004) to graph-based methods (Sindhwani *et al.*, 2005; Zhou *et al.*, 2004a). However, with the first theoretical considerations (Ratsaby and Venkatesh, 1995; Castelli and Cover, 1995; Cozman *et al.*, 2003) the general enthusiasm disappeared.

On one hand, it was almost impossible to provide theoretical guarantees about convergence or success when using additional unlabeled data. And on the other hand, the few theoretical results are rather discouraging or even negative. Although these theoretical studies are important to understand and enhance semi-supervised learning they should be not seen as evidence to question semi-supervised learning altogether. Most of these analyses act on strongly limited assumptions that do not translate into real world applications, for example the presumption that labeled and unlabeled data have the same underlying distribution. Rather, one should see these theoretical considerations and in particular the relatively small number of these publications as a proof that this topic has turned out to be more complex than expected.

In fact, imitation, intuition, and experience play an important role in human decision making (Damasio, 1994) especially under risk (Kahneman and Tversky, 1979). But only a small fraction of this accumulated knowledge is labeled. The discussion around the question on *whether and how much supervision a human need* can be traced back at least to the philosophical theories of the 17th century. Kant (1781) was the first who argued that labels (knowledge) and structure (perception) are closely intertwined, summarized in one of his key phrases (see beginning of this subsection). His theory fundamentally changed and influenced our way of thinking and acting. After 200 years of research, some of his basic assumptions and argumentations might be obsolete. But the underlying theory and the main argumentation itself is still up-to-date. Indeed this theory seems to be obvious because in addition to the things we learn supervised at home or in school, there are many other things that we learn without any teacher. For example, how we move, how we grab a glass, how we use language before we start school, how we make fast decisions and so on. Of course, many of those things are learned feedback-driven in the sense that an action is completed successfully or not. However, there are still many actions or feelings that we cannot explain let alone derive solely based on knowledge. In that sense, SSL seems a natural choice to tackle the learning problem.

But in defense of the more skeptical people, one has to state that this large unlabeled part of semi-supervised learning is almost impossible to grasp. Actually, it is even not clear whether humans will ever be capable to understand it in their completeness. For the simple reason that in the course of evolution, we only had to understand and infer simple causalities, e.g., *I take the glass of water because I am thirsty*. But we were never forced to understand the entire chain of actions and decisions that leads to this final action, e.g., *grasping the glass and drinking*. In fact, every physical movement is a highly individual action that is based on imitation and experience. Although this might lead to sub-optimal movements or actions, the acquired knowledge is quite sufficient to survive in everyday life – even if we notice some limitation and ask for more supervision, e.g., to get rid of pain induced by suboptimal movements or to run faster in a marathon. Most advices only give a direction and do not describe a muscle-induced action in its full complexity.

One might argue that these examples are indeed more physical. But even if we limit these considerations to our decisions that could be purely driven by our knowledge, we still observe many decision based on the so called *gut feeling* or other feelings that we cannot explain. Why do we know that someone is annoyed or sad or impatient although this person uses exactly the same words as he does every day? Why is it impossible to imagine in advance how we will react or feel after a certain event, for example if we loose a competition. Even more complicated is to infer how other people will react on an event. The reason is simple and devastating at the same time: We are overwhelmed by millions of sensations per minute of which only a small fraction of impressions are processed consciously and the rest subconsciously and this is only a tiny fraction of what the entire world perceives. Thus to answer at least the questions with respect to our own, we have to assimilate all sensations. With this insight, it becomes clear why SSL research is still not where it should be. To bring a machine into the closer range of human thinking, we have to tap into the vast amount of unlabeled data meaning a ratio of 1 labeled to 1 million unlabeled data points and not the common ratio of today's task descriptions of 1 labeled to 100 unlabeled observations.

Besides these more philosophical considerations, there are also many practical reasons for SSL. One obvious argument is the reduction of the hypotheses space (Balcan and Blum, 2005) in particular if there are only few labels. Figure 1.3 shows one point per class in the leftmost image. Without any additional information, the space of possible hypotheses is less goal-oriented (Figure 1.3 middle) while unlabeled data reduces this space as shown in the right visualization. This speed up of concept learning through relevant prior knowledge has been also verified in cognition by Pazzani (1991); Murphy and Allopenna (1994). A mechanical engineer will be proceed faster and more goal-oriented when assembling a machine in comparison to a layman because he already knows how to use the tools and where the single items should be approximately placed.

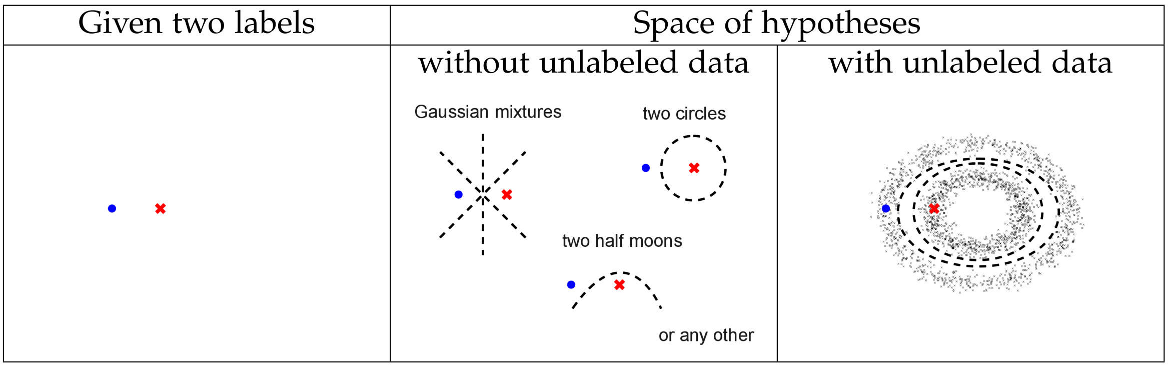


Figure 1.3. Unlabeled data reduce the space of hypotheses if there are only few labels.

Another reason for SSL is the low amount of supervision. In particular for tasks such as semantic image labeling or image understanding, where we need pixel-wise annotations, this advantage becomes increasingly important. But also for tasks such as object detection or recognition, we observe a substantial improvement the more data are used. The most image descriptions are high dimensional resulting in a strong demand for data. But the labeling process is not always reliable – for example when using Mechanical Turk (Welinder *et al.*, 2010). Finally, some applications need a continuous update, e.g. for separating spam emails from valid emails.

1.2 CHALLENGES OF SEMI-SUPERVISED LEARNING

Semi-supervised learning (SSL) makes use of both supervision in terms of labels and structure (geometry) that comes with the unlabeled and labeled data. Therefore it is obvious that both parts strongly influence the performance of SSL. In the following we discuss the challenges of both components separately starting with the structural problems in Section 1.2.1 followed by the difficulties of the supervision in Section 1.2.2.

1.2.1 Structural problems

The apparently most promising but also much more complicated direction for SSL is the improvement of the structure itself. Imagine a dataset like the *two half moons* shown in Figure 1.4 with two labels marked with red numbers. It does not matter which label is used for classification. The used SSL algorithm (Zhou *et al.*, 2004a) achieves always a classification performance of 100%. Although this is an artificial example it still reflects our common sense assumption that there is exactly one concept for each class and each instance of this class is organized around this central prototype (Osherson and Smith, 1981; Cohen and Murphy, 1984). But often, there is a large gap between our base assumption and today's computer vision task descriptions and solutions.

Figure 1.5 visualizes the general workflow of SSL algorithms: Based on our dataset that consists of labeled and unlabeled data, we compute image descriptors

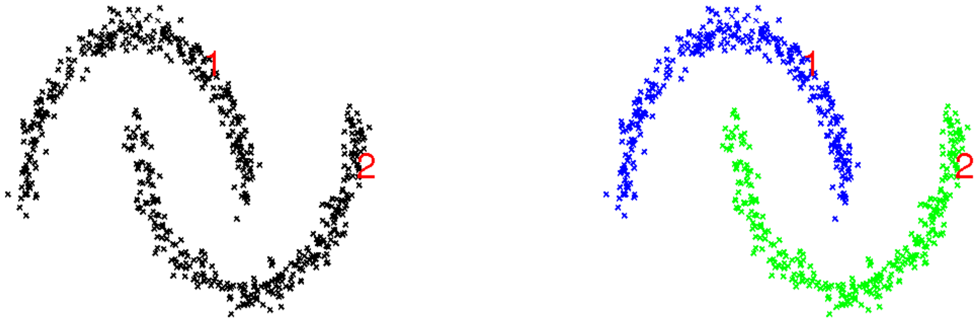


Figure 1.4. Two half moons dataset with exactly one label per class (marked by a red number): before classification (left) and after classification (right) with 100% accuracy.

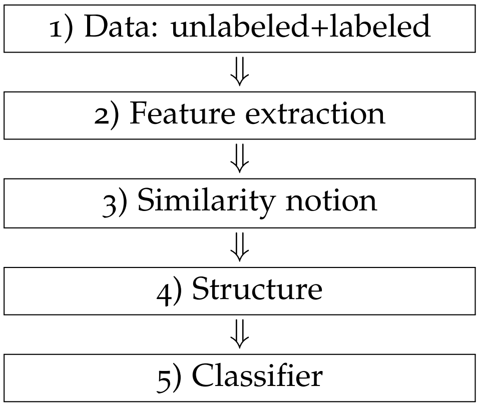
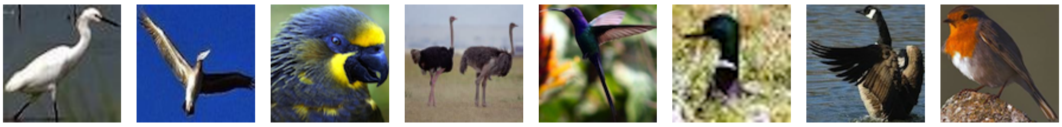


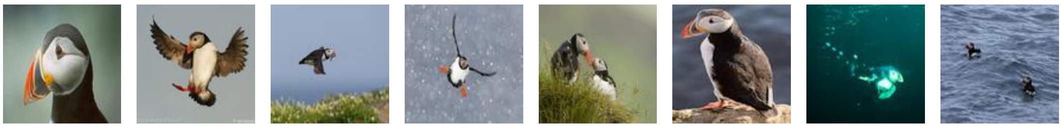
Figure 1.5. Workflow of semi-supervised learning algorithms in vision.

for each image. After that we compute the similarities between each image pair with some measure. The resulting structure is used by SSL algorithms, e.g., for EM clustering (Nigam *et al.*, 1999), as a regularization term to improve SVM (Seeger, 2001; Joachims, 1999), or to build a graph structure and to find a solution with Mincut (Blum and Chawla, 2001) or by label spreading (Zhou *et al.*, 2004a). However, each of these classifiers can be only as good as the extracted geometry of the data and this strongly depends on three main sources: i) data, ii) image description, and iii) similarity notion. Furthermore, the quality of each single source is dependent on both the approaches that are used for these steps but also on the quality of the previous steps. This means, information loss for example through an incomplete dataset will be propagated to the classifier and cannot be compensated by one of the intermediate steps. Similar argument holds for image description: if one aspect is neglected, e.g., color, the best similarity measure will be not able to properly distinguish between *green apples* and *red tomatoes*. In the following we discuss each of these three components separately.

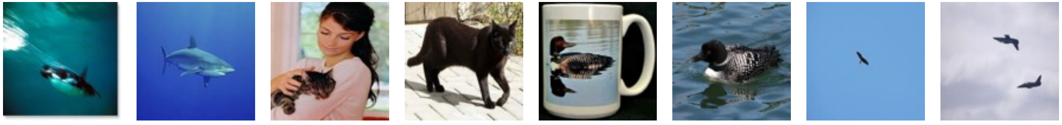
a) intra-class variability of the base category *bird*



b) intra-class variability of the subcategory *puffin*



c) inter-class confusion



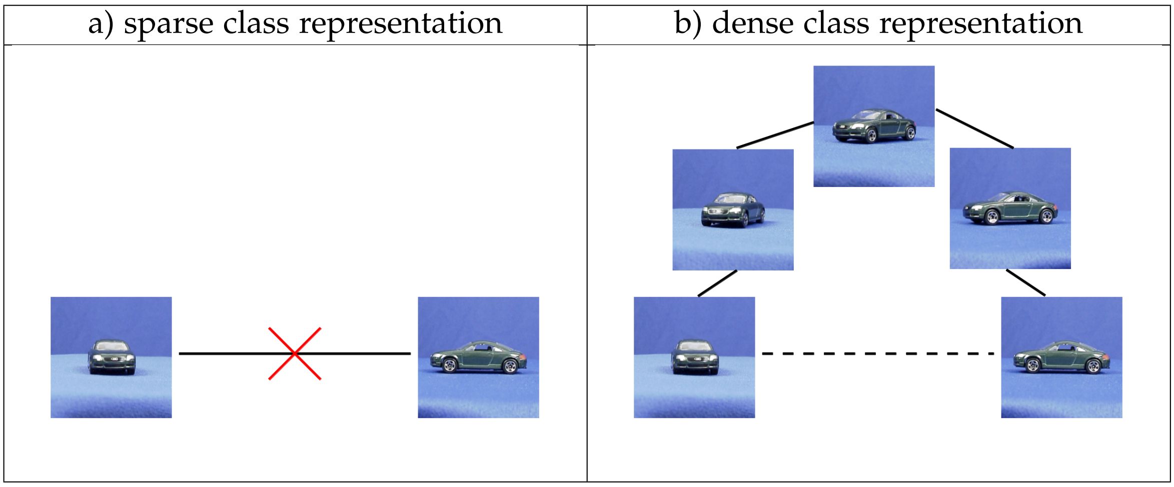
bird ≠ fish person ≠ cat mug ≠ bird bird ≠ airplane

Figure 1.6. Examples for a) large intra-class variability for the base category *bird* (top row) and b) for the subcategory *puffin* of the category*bird*, and c) small inter-class differences (bottom row).

i) Data. Most common datasets for image classification like Caltech 101 (Fei-Fei *et al.*, 2006), PASCAL VOC (Everingham *et al.*, 2008), Animals with Attribute (Lampert *et al.*, 2009), LabelMe (Torralba *et al.*, 2010), animals on the web (Berg and Forsyth, 2006), 80 million tiny images (Torralba *et al.*, 2008) or ImageNet (Deng *et al.*, 2009), are generated for supervised classification. They provide full label information that might be error-prone in particular when crowd-source services like Mechanical Turk are used (Welinder *et al.*, 2010). They contain a large intra-class variety within a base category such as *bird* (Figure 1.6 a) but also within one specific subcategory of this class such as *puffin* (Figure 1.6 b). Without a good description and understanding of the concept, it will be difficult to connect those examples and group them together to one class. Small inter-class variation is the other end of the scale (Figure 1.6 c) leading to many overlapping and confusing areas that are even for humans difficult to learn. Finally and most unfortunately, they contain usually a limited amount of images because labeling is expensive.

An ideal dataset for SSL should be *dense* enough that means each class should be densely sampled that allows to find compact and well separated clusters. In this dataset, we might be able to connect the front view of a car and a side view of car as in Figure 1.7. But on the other hand, this dataset should be also *sparse* enough to avoid overheads due to space and time complexity. Usually, SSL approaches such as graph-based algorithms come with a quadratic time complexity in the number of images. Because of this complexity, the *the-more-data-the-better* strategy can usually not be applied.

A second reason why *the-more-data-the-better* strategy does not work well in practice is that only a small fraction of the data, e.g. added from the internet, is



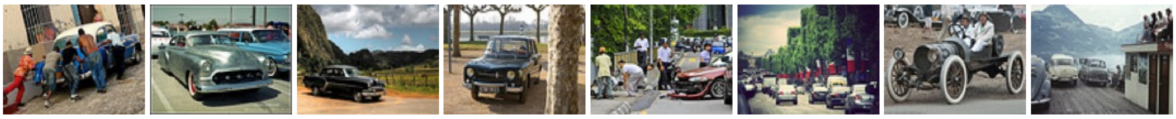
a) sparse class representation

b) dense class representation

Figure 1.7. Example of a) a sparse class description that makes it difficult to find a connection between both images and of b) a dense class representation that makes it possible to find a way from the front view to the side view of a car over several viewpoints.

helpful for our classification task. Furthermore, these data sources often have a certain bias in terms of image type. One example is the data source bias. Flickr, that is the base for PASCAL VOC (Everingham *et al.*, 2008), contains mostly holiday pictures. Therefore *person* is with 31.2% the most common object in this dataset. The second most frequent object is *chair* with 8.5%. Another bias can be also seen in Figure 1.8 (top row) that shows the first results of approx. 5.8 million results for the query *car* in Flickr. In contrast, Google shows more professional images that are often generated for marketing purpose as you can see in Figure 1.8 (bottom row).

Flickr: $\approx$ 5.8 million results



Google: $\approx$ 12.5 million results

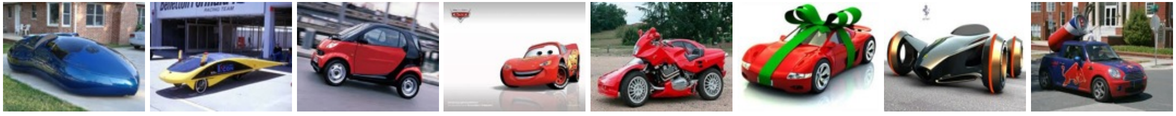


Figure 1.8. Data source bias: First results for the query *car* with a) Flickr that contains more holiday pictures of cars (top row), and b) Google images with focus on racing cars (bottom row).

Another problem comes with the capture bias. This is usually a result from human properties such as body height. Most images are taken from an adult person in a standing position thus from an average height of $1.6 - 1.8$ meters (Figure 1.9 a). A simple change to a child position, i.e. < 1 meter, leads to a different viewpoint and thus perception, e.g., some things appear larger (buildings) or more frightening (animals). Furthermore, most people are right handed resulting, e.g., in many images of mugs with the handle on the right side. Some objects have a quite different appearance (Figure 1.9 b), e.g., salmon considered as an animal vs. food, and other categories might change their appearance over time (Figure 1.9 c). Of course, it seems likely that massive amounts of data also contain relevant images but to find these images we have to process over millions of images for this single class.

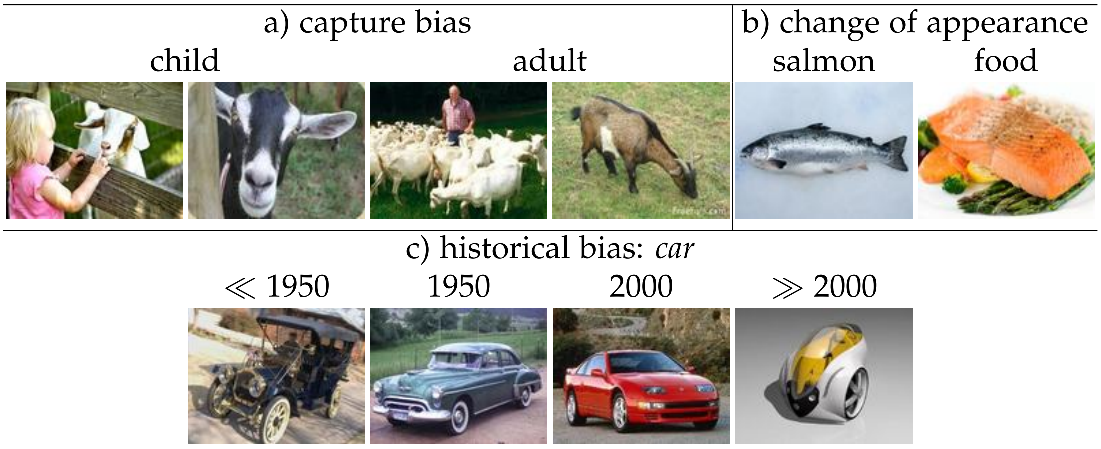


Figure 1.9. Dataset bias due to a) the sighting angle, b) the semantic of an object, or c) because of historical trends.

ii) Image description. Suppose we have a dataset that captures the broad variety of each class, e.g. different viewpoints, several contexts and so on. Thus, there is a chance to build a compact cluster structure similar to Figure 1.10. Then it does not automatically mean that we are also able to exhaust this potential. Today's image descriptors are far away from capturing all these different aspects that humans can easily recognize. In the following we list briefly most of the common problems. See also Freeman (2011) for a short overview of today's problem in computer vision.

Intra-class/Inter-class variability. As it mentioned before, many classes come with a large intra-class variance in their appearance and their surrounding environment. The class *bird* is one of the extreme cases where even the height varies from few centimeters like the hummingbird to almost 2 meters like the flightless ostrich (Figure 1.6 a) not to mention the large variation in shape and in color. Of course, a limitation to one species (Figure 1.6 b) might constrain the general appearance of an object but not the number of different poses or the context around this object. On the other side, there are classes that look similar to each other in particular in some poses, e.g., a bird and an airplane in the sky (Figure 1.6 c), or when two classes jointly appear in an image, e.g., a cat in the arms of a person or a sticker from an animal at a mug.

Background clutter. Some images are dominated by their background as can be seen in Figure 1.11 where it is almonst impossible to see some objects because of the trees. Often, these images are confused because of their similar-looking background.

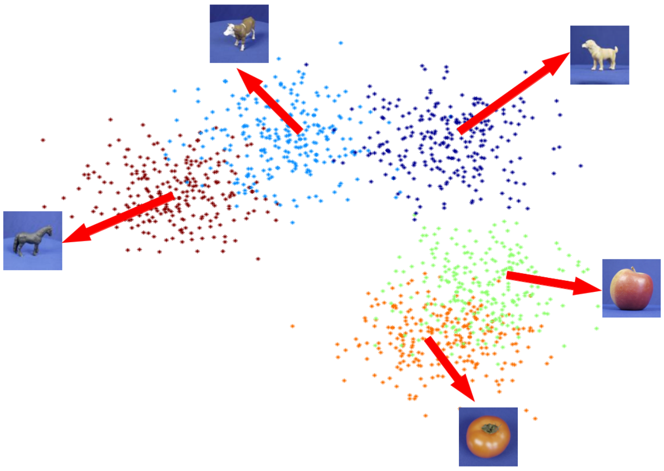


Figure 1.10. Structure with dense representation but still overlapping regions.

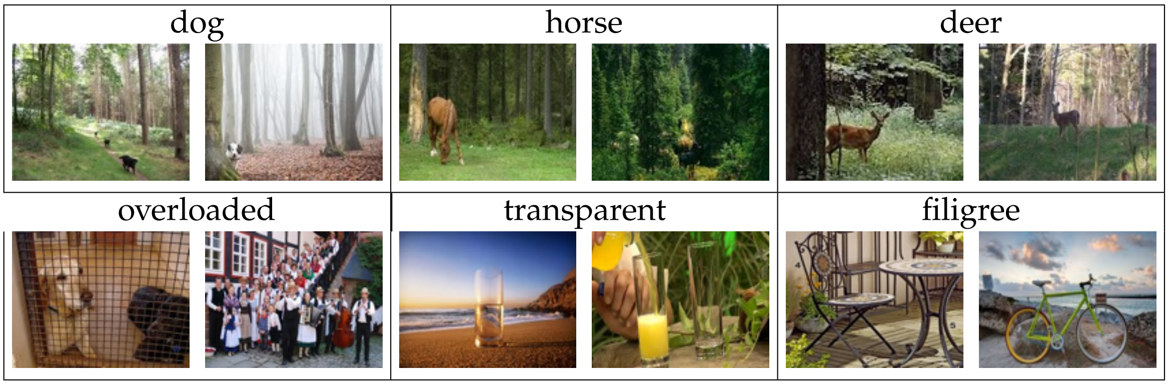


Figure 1.11. Examples with a dominating background that is shared among different classes (top row), and examples with overloaded background and object that are transparent or filigree so that background is always a part of the object (bottom row).

There are images with an overloaded background structure that are similar to many other images in a dataset. Finally, some objects are difficult to distinguish from the background because they are transparent (like glass) (Fritz *et al.*, 2009), or filigree like a bicycle or a chair.

Illumination changes. Another problem is the change in illumination depending on the time of the day and the season (Figure 1.12). E.g., a lake is susceptible to lighting conditions due to its surface and volume properties resulting in a wide color spectrum. But also many other objects look different during the day and at night, e.g., trees are green during the day and dark at night.

Truncation and partial occlusion. Partial occlusions are an omnipresent property. Herd animals like sheep or gazelles occur frequently in groups. As already men-

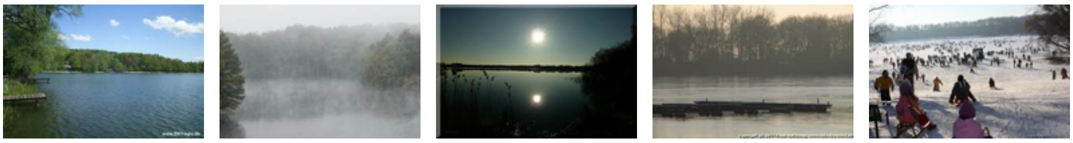


Figure 1.12. Examples of a lake with different illuminations depending on the time of day and the season.

tioned before, some objects can be covered to a large extend by a person using that object like a bicycle or chair. And other object classes are very large so that they are only partly captured or truncated like a cathedral (Figure 1.13).

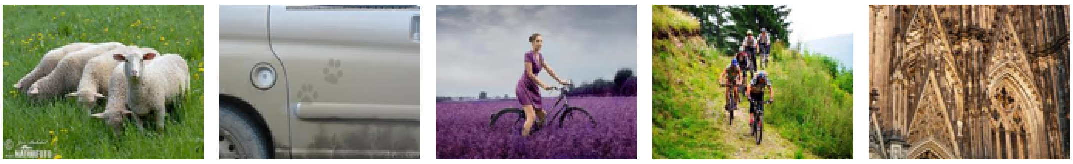


Figure 1.13. Examples of truncations and partial occlusions.

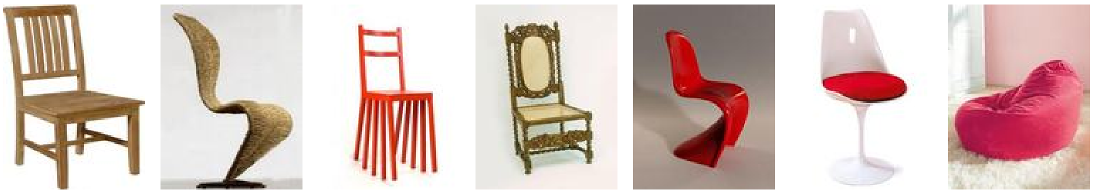


Figure 1.14. Examples of the large variety in shape for the class *chair*.

Shape variation. Some categories have a large variation in shape and appearance, e.g. *chair* (Figure 1.14), *table*, or *lamp*. These categories can be often only described by their function such as *something to sit on*.

Mimesis and other. Another set of problems comes with the evolutionary adaption of some species to their background so that they are difficult to recognize by other animals, e.g. the chameleon or the flounder. Other animals such as zebras are indeed visible but it is difficult to point out an individual animal due to their pattern structure (Figure 1.15).

Basically, the ideal image descriptor should emerge with some prior knowledge about what and where the object is located in the image, how to separate the background from the main object, which color is trustable or rather how to adjust this color, and what are the possible and feasible poses of an object. Furthermore, this descriptor should have a general idea of the shape and texture of an object to infer which part is occluded or truncated. While a human focuses led on the main

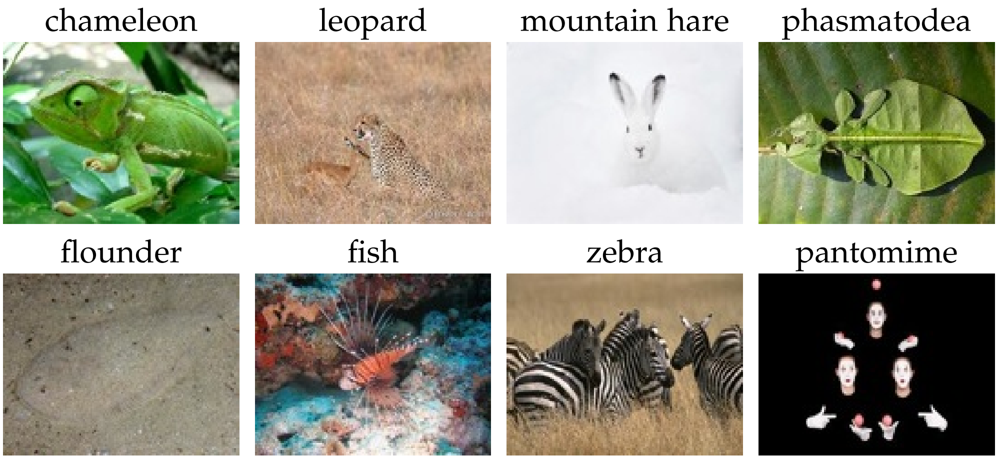


Figure 1.15. Examples of objects that are difficult to distinguish from their background or to identify the object-specific shape.

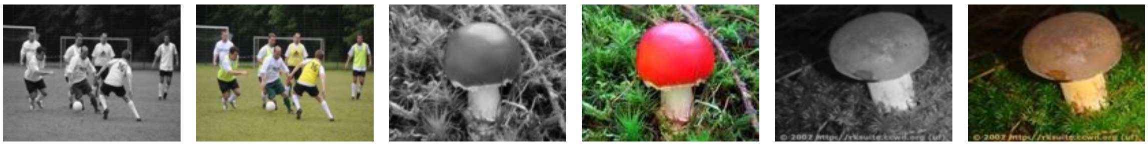


Figure 1.16. Examples of images that are difficult to understand without color information.

object, many of today's image descriptors such as dense SIFT analyze every single blade of grass or every single leaf from a tree leading to an overcrowded image description that often considers only one aspect in the image like color or gradients. Of course a good similarity metric can handle this high dimensionality. But an information loss in this partial extraction propagates to the classifier. Figure 1.16 shows some examples with and without color information. Even for a human it is hard to follow a soccer game or to distinguish between eatable and poisonous mushrooms by just omitting color not to mention information such as texture, or shape.

iii) Similarity notion. The final crucial part of the structure extraction is the similarity measure. Most frequently used is the Euclidean distance with a Gaussian kernel weighting. This is usually a good choice for feature vectors of low dimensionality ($\ll 100$) containing only little noise. But as mentioned before, most image descriptors aggregate many not preprocessed information that leads to a high-dimensional vector ($> 10,000$) from which only a small fraction of dimensions are relevant for a object class. In particular Euclidean distance is known to be sensitive to noise that becomes more prominent the more dimensions are used. One phenomenon that we observe with Euclidean distance is that some images are similar to almost all other images. The resulting structure (such as shown in Figure 1.17) harms almost every classification algorithm because there is no clear separation between different classes (Luxburg *et al.*, 2010).

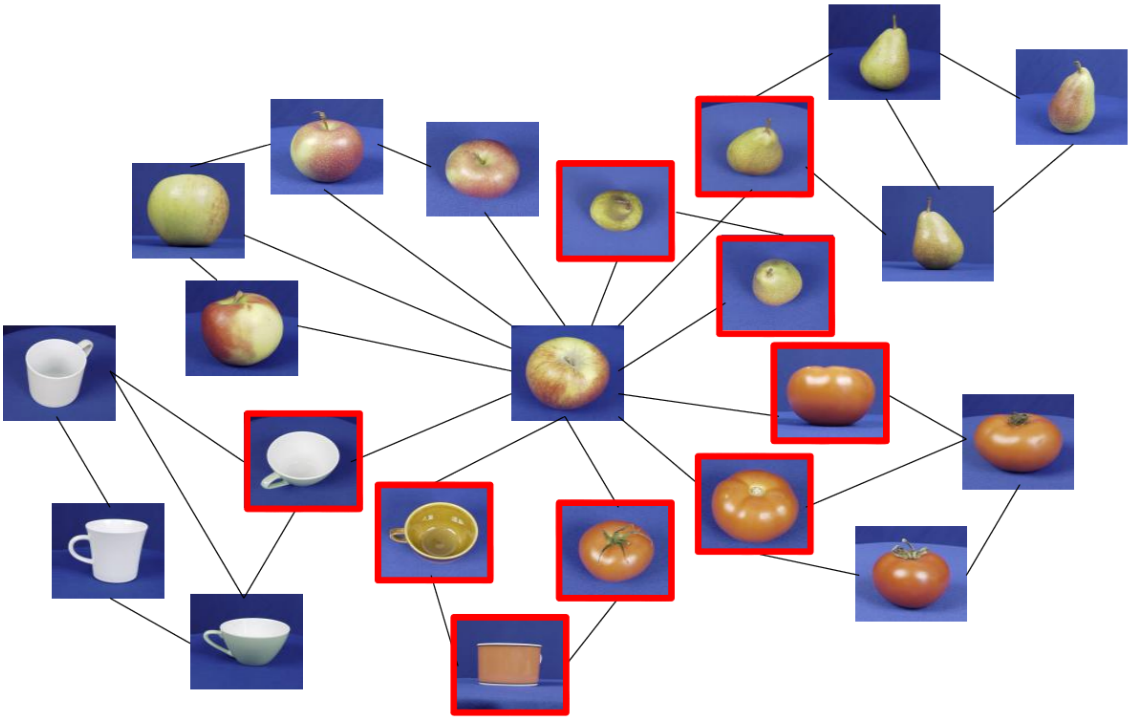


Figure 1.17. Problem of Euclidean distance in a high dimensional space: The image in the middle is similar to many other images. Red boxes indicate false neighbors.

Another problem comes with the missing weighting of the single dimensions in the feature space, i.e., all dimensions are equally considered. But usually only a small fraction of this high dimensional feature vector is relevant for each class. Finally, we often consider image pairs instead of groups of images. This is easy to implement but seems suboptimal for good generalization. A human who has never seen a zebra before and only gets the first image from Figure 1.18 will certainly have problems to build a general concept or model of a zebra because there is no information about the shape, the size, or the environment around this animal. Without these higher order relations extracted from a group of images, it might be difficult to distinguish the first image from the sofa shown in the last image.

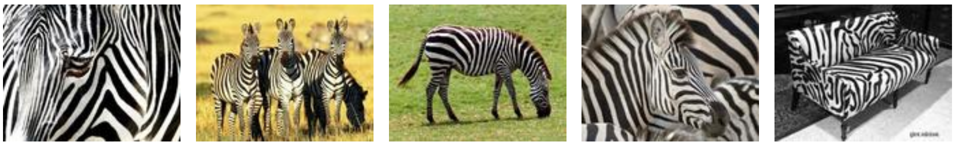
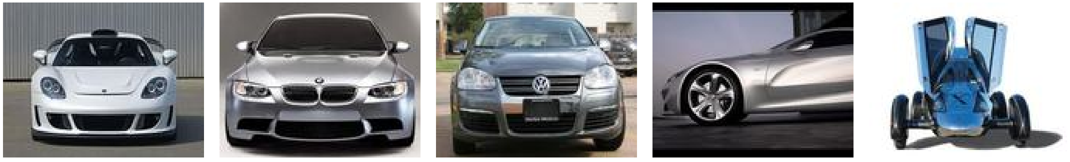


Figure 1.18. Pairwise image similarities might be problematic due to the missing generalization. From the first image it is not clear how to generalize so that this image do not get confused with the sofa in the last image.

1.2.2 Labeling issues

The second import issue besides the structure is the label information. As mentioned before, supervision causes no problems if the structure itself perfectly separates the classes. But usually this is not the case. Therefore, the label information plays an important role in particular for semi-supervised learning where we have only few labels per class. While for supervised learning more data is labeled, in SSL we have to deal with a ratio of $1\% - 5\%$ labeled to $95\% - 99\%$ unlabeled data. Thus, there is a need for high quality labels that are representative for the class and allow a better generalization. Additionally, we have to ensure that there is at least one label for each mixture (e.g., different viewpoints or appearances) of one class otherwise it might be difficult to classify unseen viewpoints. Figure 1.19 show five less representative samples for the class *car* in the first row, assuming that the test set contains also the back or the side view of a car. In contrast, the second row shows more representative samples of this class so that the main properties of this object will be apparent such as the shape and the surface.

a) outlier or redundant viewpoints



b) more representative samples

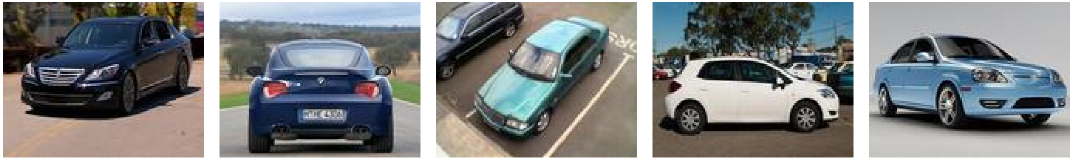


Figure 1.19. SSL is strongly dependent on the representativeness of the small training set: a) less representative samples for the entire class *car* vs. b) more representative samples in terms of viewpoints.

Coming back to Figure 1.17 if the image in the middle with these many false neighbors is labeled then most of the neighboring images will be false classified (marked with a red bounding box) because of the strong impact on the direct neighbors. Another problem occurs when a class is split into separate clusters, e.g., front view of a car and side view of a car, and there are only labels for one of these sub-clusters. The other sub-clusters cannot be classified correctly anymore. Ideally, we have labels that are representative or prototypical for a class that means they lie in a dense region and consider each aspect or viewpoint of a class.

1.3 CONTRIBUTIONS OF THIS THESIS

As we learned from the previous section, there are two sets of challenges with semi-supervised learning approaches, i.e., structural problems and uncharacteristic labels. In this thesis, we address both issues. We start with an exploration of state-of-the-art algorithms for semi-supervised learning with a focus on graph-based algorithms. These methods come with a reasonable runtime, almost no parameters, and a natural interpretation of their neighborhood structure that makes it easy to evaluate the quality of both labels and structure. We confront diverse algorithms with different graph structures given by different image descriptors and distance measures and we show that graph structure is more important than the algorithms itself (Ebert *et al.*, 2010). Additionally, we get similar results when applying these methods to other domains such as activity recognition (Stikic *et al.*, 2011). These observations encourage us to continue research in this direction.

In the following, we concentrate on several improvements for the metric itself. In Ebert *et al.* (2011), we apply a metric learning framework to transform the original data space into a more distinctive one with small intra-class distances and large inter-class distances. We show superior performance of our novel semi-supervised framework on state-of-the-art object recognition datasets such as Caltech 101. Additional, we explore other improvements like dimensionality reduction or combining SVM-based metrics with our semi-supervised learning method.

After showing a strong dependency between labels and classification performance in Ebert *et al.* (2011), we also improve the labels for learning by active learning. This technique comes with different sampling criteria that either query for the least certain examples or for more representative samples. Usually, a combination of both strategies brings the most success but the trade-off of these different criteria is difficult to adjust and strongly dependent on the dataset. This problem is also known as exploitation-exploration-dilemma. As a consequence, the main approach in Ebert *et al.* (2012b) is to find a meta routine that simultaneously combines different sampling criteria (exploration vs. exploitation), finds a good trade-off between these, and adapts this strategy during the learning process to different datasets without any prior knowledge about these datasets. This leads to a reinforced active learning formulation that learns in an on-line fashion a good trade-off between different sampling criteria without manual fine-tuning to one specific dataset and is able to react in a flexible manner to novel requirements.

In the last period of this thesis, we combine improvements of graph structure with more representative labels leading to a stronger performance gain. In Ebert *et al.* (2012a), we show that a potentially large improvement is possible when applying metric learning with more representative labels. To this end, we combine active learning with metric learning and show improvements of more than 10% over our previous publication (Ebert *et al.*, 2011) and more than 20% improvement as compared to our first publication (Ebert *et al.*, 2010). Furthermore, we explore graph improvements by adding more unlabeled data (Ebert *et al.*, 2012c). We achieve a significant performance increase by adding more representative data with our novel framework that is better suited for the task at hand. Our results are going clearly beyond the performance for randomly adding images of the respective classes.

1.4 OUTLINE

This thesis is structured as follows:

Chapter 2: Related work This chapter gives an overview about related work. We start with a review of object class recognition that is frequently addressed by fully supervised approaches. After that, we summarize semi-supervised learning methods with a focus on graph-based algorithm. As mentioned before, these methods are strongly dependent on both quality of supervision and graph structure. Thus, we outline several approaches towards a better graph structure and explores state-of-the-art literature in active learning as a basis to improve the labels.

Chapter 3: Graph-based semi-supervised learning In this chapter, we review state-of-the-art algorithms for graph-based semi-supervised learning and analyze different graph construction methods. Additional, we introduce all datasets and image descriptors that we use in this thesis for image classification. Finally, we empirically show that the graph structure is more important than the different algorithms. This study is part of Ebert *et al.* (2010) and Larlus *et al.* (2010) and is the base for all subsequent publications.

Chapter 4: Graph improvement In this chapter, we extend Ebert *et al.* (2010) by analyzing several unsupervised improvements as well as supervised improvements to get a more discriminative graph structure. Amongst others, we integrate an information-theoretic metric learning framework into the graph construction and propose a new framework *IMLP* that make use of both unsupervised and supervised data with which we achieve state-of-the-art performance on Caltech 101 (Ebert *et al.*, 2011).

Chapter 5: Label Improvement by Active Learning This chapter addresses the second critical part of SSL – the labels mentioned in Section 1.2.2. For this purpose, we integrate active learning into our SSL framework. In the first part, we review a broad range of common active sampling criteria and discuss advantages and disadvantages of those. We propose our new sampling criteria *graph density* that uses the underlying graph structure to find more representative samples from dense regions. Finally, we introduce our new meta active learning framework based on the Markov decision process that allows full flexibility according to the number of combined criteria as well as the combination strategy (Ebert *et al.*, 2012b).

Chapter 6: Active Metric Learning The last two chapters show the large potential when combining techniques for graph improvement with active learning. Chapter

6 improves the previously mentioned metric learning framework by using more representative samples for learning. We propose two novel methods that combines metric learning with active learning and show significant improvement for several classification schemes, different datasets, and descriptors (Ebert *et al.*, 2012a).

Chapter 7: Active Dataset Construction In this chapter, we use active learning techniques to build a richer dataset (Ebert *et al.*, 2012c). We introduce two selection strategies to enhance the neighborhood structure in a fully unsupervised fashion. We compare these criteria to previous methods and show on mid-sized datasets that we improve these approaches in particular when we consider more realistic datasets with occlusions, truncations, and background clutter. After that, we illustrate on a subset of ImageNet (ILSVRC 2010) with 100 classes that we get better performance when using only a representative subset of all images. This emphasizes our claim that there is no need to use all available unlabeled data. We also show that our approach is able to process also the entire ILSVRC 2010 dataset with $1,000$ classes and more than one million images.

Chapter 8: Conclusion In the last part of this thesis, we outline future directions and problems that are not addressed in this work. Finally, we give a brief summary of this thesis.

RELATED WORK 2

Contents

2.1 Object class recognition **19**
  2.1.1 Image description 20
  2.1.2 Object class learning 21
  2.1.3 Relation to own work 23
2.2 Semi-supervised learning **23**
  2.2.1 Cluster assumption 24
  2.2.2 Manifold assumption 25
  2.2.3 Graph-based methods 26
  2.2.4 Relation to own work 27
2.3 Graph improvement **27**
  2.3.1 Scalable algorithms 27
  2.3.2 Metric learning 29
  2.3.3 Graph construction 31
2.4 Active Learning **34**
  2.4.1 Exploitation-driven methods 35
  2.4.2 Exploration-driven methods 36
  2.4.3 Trade-off between exploration and exploitation 37
  2.4.4 Relation to own work 38

In this chapter we give an overview about work related to the topics of this thesis. We start with a review of object class recognition in Section 2.1 by discussing state-of-the-art image description techniques and supervised as well as unsupervised learning methods. Section 2.2 gives a brief summary of semi-supervised learning methods with a focus on graph-based algorithms. Following from Section 2.2 and our introduction, we constitute two main research areas, i.e., structure improvement and better labels. Previous work for both areas are discussed separately. Thus, we describe in Section 2.3 several approaches towards a better graph structure. Finally, Section 2.4 explores state-of-the-art literature in active learning that will be the base to improve the labels for SSL.

2.1 OBJECT CLASS RECOGNITION

Object class recognition is one of the oldest problems in computer vision. To solve this task, we first need a description of the image by extracting the relevant informations from an image. This representation is then used to learn a model of a class. In the following, we examine both parts separately in Section 2.1.1 (image description) and

Section 2.1.2 (object class learning). Finally, we discuss in Section 2.1.3 main issues of state-of-the-art object class recognition systems that serve as a starting point for this thesis.

2.1.1 Image description

The most important part of object recognition is the description of the image that contains the object itself as well as the background or context around this object. The less information is extracted from the image the less information can be used to learn a model. Historically, this part can be split into two main direction: shape-based methods and appearance-based approaches. Approaches based on shape might be closer to human perception and they are invariant to lighting conditions. Moreover, some categories are better defined by their shape than by their appearance, e.g., *bottle* or *bird* (Biederman and Ju, 1988). The spectrum of possible solutions ranges from coordinate theory (Thompson, 1917) that is the base for many state-of-the-art works such as Fischler and Elschlager (1973); Kendall (1984); Belongie *et al.* (2002); Felzenszwalb and Huttenlocher (2005) over grouping of regions or contours (Malik *et al.*, 2001; Leibe *et al.*, 2004) based on the Gestalt theory (von Ehrenfels, 1890; Wertheimer, 1912) to using shade as a primary cue (Barrow and Tenenbaum, 1978; Forsyth and Zisserman, 1990; Barron and Malik, 2012) also known as intrinsic image problem. Although these approaches seem more natural, today's shape-based descriptors are often not competitive to appearance-based descriptors as they usually use some kind of shape matching (Donoser *et al.*, 2009; Stark *et al.*, 2009; Riemenschneider *et al.*, 2010) leading to a viewpoint-dependent description.

In contrast, appearance-based methods are easier to extract as they do not need any basic understanding about the object itself such as physical constraints or other properties. Texture-based methods (Darling and Joseph, 1968; Tamura *et al.*, 1978) were the first attempts towards appearance-based description. Haralick *et al.* (1973) even claims that texture is the most important characteristic to identify objects or regions. Even though they provide useful cues about the object they are often too sensitive to illumination changes, occlusions, varying background, or other types of non-Gaussian noise. Therefore, methods are proposed to handle several variations in parallel and select the best hypothesis based on minimum description length (Bischof and Leonardis, 1998; Leonardis and Bischof, 2000). But most successful and state-of-the-art with respect to image description are the different kinds of gradient-based methods, e.g. local SIFT (Lowe, 2004) and their approximations (Grabner *et al.*, 2006), global HOG (Dalal and Triggs, 2005), spatial pyramid matching (Lazebnik *et al.*, 2003), or color SIFT (van de Sande *et al.*, 2010). An extensive evaluation of several image descriptors can be found either in Mikolajczyk *et al.* (2005) with respect to object recognition or in Pinz *et al.* (2008) in the context of cognitive vision (Vernon, 2005).

However, most of these approaches lack still expressiveness. They consider only one aspect when describing an image, e.g. texture, shape, or color. This leads inherently to an enormous information loss as shown in Figure 2.1. Even for a human it will be impossible to categorize the shown object as a tomato just by looking at the texture (third image) or the shape (first image). Therefore, a combination of several features are essential, for example image-based features with geometry (Burl and Perona, 1996; Wiskott and von der Malsburg, 1993; Pope and Lowe, 1996), shape with texture (Cootes *et al.*, 1998), several local appearances (Schiele and Crowley, 2000), local and global appearances (Leibe *et al.*, 2005), HOG with texture (Wang *et al.*, 2009c), multiple kernels for global as well as local features learned with an SVM (Sonnenburg *et al.*, 2006; Gehler and Nowozin, 2009; Vedaldi *et al.*, 2009), with boosting (Dubout and Fleuret, 2011), or with conditional random fields (Schnitzspan *et al.*, 2009).

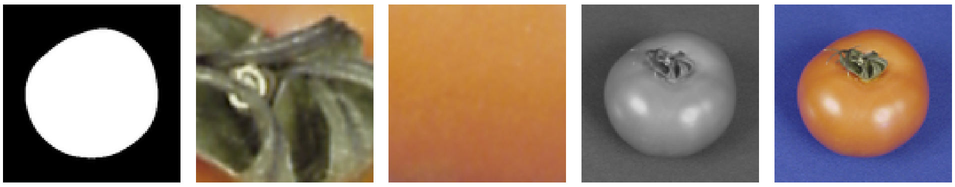


Figure 2.1. Visualization of the information loss when we consider only: a) shape; b and c) texture or image patches; d) gray value images in comparison to e) the information content provided by the entire image.

Additional to the object description, the context around an object also provides an important clue about the object itself. The usage is based on the assumption that objects presented in a familiar context are faster to localize and to recognize (Biederman, 1972; Palmer, 1975; Oliva and Torralba, 2007). This is also attended by a reduction of complexity in terms of scene description (Strat and Fischler, 1991). Some of the previous mentioned descriptors model already indirectly the context, e.g. approaches that densely sample the image with a regular grid (Fei-Fei and Perona, 2005; Tuytelaars and Schmid, 2007; Tola *et al.*, 2008), or by using any other kind of global image description (Murphy *et al.*, 2003; Torralba, 2003; Shotton *et al.*, 2007). Most approaches that directly model context use either contextual constraints (Carbonetto *et al.*, 2004; Rabinovich *et al.*, 2007; Galleguillos *et al.*, 2008) or co-occurrences and relative locations (Bar and Ullman, 1996). Nevertheless, the main problem with most context approaches is often that context is equivalently considered to the description of the object as it is the case for dense SIFT. This leads to an overcrowded and often confusing description (Wolf and Bileschi, 2006). Ideally, the context should only serve as an additional prior similar to Kruppa and Schiele (2003) to guide the recognition of the object and to better focus on the object itself.

2.1.2 Object class learning

Learning a general model from image features is the second important part of object recognition. The way a classifier generalizes depends on both similarity and dissimilarity constraints. The more diverse and representative samples we have from the same class the more general the resulting model can be. Also, the criteria to distinguish a *bird* from a *flower* are quite different from the criteria to distinguish different bird species. Usually, super-classes are often defined by their function while sub-classes are specified by fine-grained properties (Rosch *et al.*, 1976; Hillel and Weinshall, 2007). In general, the learning procedure can be divided into three broad directions when concerning the amount of supervision: supervised learning, unsupervised learning and semi-supervised learning that will be separately discussed in Section 2.2.

Among those directions, supervised learning is most favored because it is easier to control and to understand and there is usually a better theoretical motivation, e.g. VC theory (Vapnik and Chervonenkis, 1971) or PAC learning (Valiant, 1984). Approaches for this paradigm can be divided, e.g. by their learning strategy (i.e. exemplar-based or concept-driven), or by their expressiveness in terms of model complexity, i.e. discriminative (simple causalities) or generative (hidden correlations). Exemplar-based models (Mahamud and Hebert, 2003; Chum and Zisserman, 2007; Boiman *et al.*, 2008; Malisiewicz *et al.*, 2011) assume that all information can be extracted from the nearest neighborhood with almost no learning. Thus, these methods are usually fast in training. But they need a large storage capacity, they do not generalize well, and they strongly depend on the quality of the training data.

In contrast, concept-driven approaches find correlations within the data. These methods can be further divided by their complexity into discriminative methods (Chapelle *et al.*, 1999; Varma and Ray, 2007; Gehler and Nowozin, 2009) and generative method (Fei-Fei *et al.*, 2007; Abbott *et al.*, 2011). The latter one are more typical in unsupervised learning than in supervised learning as they additional try to explore hidden relations. This makes such models indeed more flexible but also less controllable. To overcome those weaknesses, approaches are proposed that combine both discriminative and generative models Bischof *et al.* (1992b); Jaakkola *et al.* (1998); Fritz *et al.* (2005); Grabner *et al.* (2007). However, the main drawback of supervised methods is their need for a large amount of supervision and their dependency on the quality of this supervision. In particular the reliability of labels decreases significantly when using crowdsourcing services such as Mechanical Turk (Welinder *et al.*, 2010) that is often deployed to meet this high demand for supervision.

Unsupervised learning reflects the other end of the scale as they do not use any supervision. All correlations and insights are only based on the underlying structure. These methods discover categories either by clustering (Zelnik-Manor and Perona, 2004; Grauman and Darrell, 2006; Kim *et al.*, 2008; Buehler and Hein, 2009), by exploring a hierarchy (Blei *et al.*, 2003; Bart *et al.*, 2008; Sivic *et al.*, 2008; Torralba *et al.*, 2006; Griffin and Perona, 2008; Gao and Koller, 2011; Marszalek and Schmid, 2008), by looking for re-occurring patterns (Fritz and Schiele, 2006; Liu and Chen, 2007; Tuytelaars *et al.*, 2009), or by using topic models (Weber *et al.*, 2000; Fergus *et al.*, 2003; Sivic *et al.*, 2005; Fei-Fei and Perona, 2005; Russell *et al.*, 2006; Fritz and Schiele, 2008). Although these methods are good in finding relationships latent in the data, the are still highly sensitive to the data and their representation. Furthermore, there is no guarantee that these models converge to a meaningful description of the data due to the missing supervision.

One key component has become apparent for all these paradigm: the data itself, i.e. the inherent geometry (Hein, 2005). They are the heart of each method as they represent the available information apart from methods that include priors in terms of physical or spatial constraints (Crandall *et al.*, 2005; Kapoor *et al.*, 2009; Gao *et al.*, 2012). The more surprising is the little attention dedicated to this topic resulting in strongly biased datasets (Ponce *et al.*, 2006; Dollár *et al.*, 2012; Torralba, 2011). The reasons for these distortions are multifaceted and are discussed more detailed in Section 1.2.1. More recently, there are works that address these dataset biases problem by defining weights for several datasets (Khosla *et al.*, 2012) or by domain adaptation (Bergamo and Torresani, 2010; Saenko *et al.*, 2010; Kulis *et al.*, 2011). But almost all these methods lack a good trade-off between supervision and structure enhancement.

2.1.3 Relation to own work

Semi-supervised learning is the paradigm that address the previously mentioned trade-off between expensive supervision and missing structural information. The few labels used for SSL are easier to get and better to control, i.e. less error-prone. While the performance of supervised learning methods depend strongly on the quality of the training set, semi-supervised learning make use of both labeled and unlabeled data for classification. Furthermore, there is the possibility to extend a given dataset by additional unlabeled data from arbitrary sources to fill and complete the manifold structure without the need to classify them. Therefore, SSL is a promising direction for image classification as supervised datasets are often limited but on the other hand the Internet provide us a large amount of unlabeled images.

2.2 SEMI-SUPERVISED LEARNING

Semi-supervised learning unites supervised learning and unsupervised learning and makes use of both supervision and structure. These methods can be divided either by their purpose, i.e., to improve supervised methods with a regularization term (Blum and Chawla, 2001; Sindhwani *et al.*, 2006; Jiang *et al.*, 2008) or with boosting (Mallapragada *et al.*, 2009; Leistner *et al.*, 2008; Saffari *et al.*, 2008), or to improve unsupervised methods (Wagstaff *et al.*, 2001; Basu and Banerjee, 2004; Bilenko *et al.*, 2004), or by their underlying assumption, i.e., cluster assumption (Section 2.2.1) or manifold assumption (Section 2.2.2). Among these manifold-based methods, graph-based methods (Section 2.2.3) enjoy great popularity due to their lower complexity and their interpretable graph structure. Finally, we will discuss work most related to this thesis in Section 2.2.4.

2.2.1 Cluster assumption

Cluster-based semi-supervised learning methods assume that points in the same cluster or dense region should be also in the same class. This results in two kinds of approaches: generative models (Nigam *et al.*, 1999; Cozman *et al.*, 2003) and low-density separation methods (Joachims, 1999; Lawrence and Jordan, 2005; Grandvalet and Bengio, 2004).

Generative models use the cluster structure to estimate the conditional density $p(x|y)$. Unlabeled data are used as additional information to adjust the prior for the density estimation in an EM-based approach (Nigam *et al.*, 1999; Cozman *et al.*, 2003) or to find a better clustering (Demiriz *et al.*, 1999; Huang *et al.*, 2012). The later one can be improved by external knowledge in form of ranked constraints (Ahmed *et al.*, 2012), or by pairwise constraints (Wagstaff *et al.*, 2001; Basu and Banerjee, 2004; Bilenko *et al.*, 2004), or by optimizing multiple objectives with a Pareto-optimal solution (Ebrahimi and Abadeh, 2012). Moreover, there are hybrid methods that combine generative models with discriminative one (Jaakkola *et al.*, 1998; Fujino *et al.*, 2005; Holub *et al.*, 2005; Druck and McCallum, 2010).

In contrast, low-density separation methods push decision boundaries into low-density regions by considering both labeled and unlabeled data. Density is used to weight the regularizer or to modify the geometry (Bousquet *et al.*, 2003). The most popular representative of these methods is the transductive SVM (TSVM) (Joachims, 1999; Seeger, 2001) that comes with a non-convex problem formulation. As a consequence, there are approaches that tackle this problem either by semi-definite programming (De Bie and Cristianini, 2004), by a approximation of the loss function (Chapelle and Zien, 2004), or by parallelization (Li and Zhou, 2011b). An alternative to TSVM is the Gaussian process approach proposed by Lawrence and Jordan (2005) that uses a null class to explore the space between two class distributions. Finally, there are also information theoretic frameworks that use entropy minimization over unlabeled data to find a prior with minimal class overlap (Grandvalet and Bengio, 2004). These methods are often used and well suited for novelty detection (Schölkopf *et al.*, 2001; Scott *et al.*, 2009).

From the theoretical point of view, cluster-based methods are better provable (Castelli and Cover, 1995; Seeger, 2001; Rigollet, 2007) than manifold-based methods since the manifold assumption turns out to be an insufficient criteria for error bounds (Lafferty and Wasserman, 2007; Nadler *et al.*, 2009). More specifically, Castelli and Cover (1995, 1996) show that the generalization error reduces exponentially in the number of labeled data when the mixtures are identifiable, i.e. almost no overlap among mixtures. Rigollet (2007) goes in the same direction and shows an exponential convergence rate using density level sets. In contrast, Cozman *et al.* (2003) show a degradation of performance when adding noisy or misleading unlabeled data and Ben-David *et al.* (2008) claim that unlabeled data provide not necessarily more insights about the data distribution. The main problem of cluster-based methods is that these methods act globally. Thus, noisy data might distort the entire data space and have a larger impact on the overall performance, e.g., by fitting mixtures to the entire dataset leading to overlapping mixtures and to many falsely classified examples.

2.2.2 Manifold assumption

Methods based on the manifold assumption assume that high-dimensional data lie on a low-dimensional manifold. These approaches can be split into graph-based methods that are discussed in Section 2.2.3 and two-step models. The step-wise models learn first a new data representation based on the unlabeled data. In the second step, they use only the labeled data to learn a classifier within this transformed space. For the first step, these approaches use spectral methods to construct better kernels (Smola and Kondor, 2003; Lafferty *et al.*, 2004; Cristianini *et al.*, 2001) or to reduce nonlinearly the dimensionality of the data such as Isomap (Tenenbaum *et al.*, 2000), locally linear embedding (Roweis and Saul, 2000), and Laplacian eigenmaps (Belkin and Niyogi, 2003).

As mentioned before, manifold-based methods and their functionality are difficult to prove from the theoretical side. The main problem follows from the definition itself: high-dimensional data need an exponentially large number of almost noise-free instances otherwise the parameters to derive a good structure are impossible to tune (Bengio *et al.*, 2004). Therefore, Lafferty and Wasserman (2007) analyze these methods with growing data but a fixed ratio of labeled to unlabeled data. However, they do not observe a faster convergence rate with more data. Similar observations are made by Nadler *et al.* (2009) that studied these methods for infinite numbers of unlabeled data without a fixed ratio. But these studies have to be put into perspective as they act on limiting assumptions such as labeled and unlabeled data come from the same distribution that might unrealistically in computer vision or these methods use a fixed parameter setting, e.g., the bandwidth for the Gaussian filter to compute the similarities is always the same. Similarly, the use of the graph Laplacian is often justified by their connection to the continuous Laplace operator although it is not fully proved (**?**Hein, 2006).

Also in cognitive science, there are several studies with different outcomes. On one side, Vandist *et al.* (2009) show that there is no additional learning effect with more unlabeled data and McDonnell *et al.* (2012) demonstrate that unsupervised data is not used if labeled data is available. On the other side, Zhu *et al.* (2007) provide a proof that humans learn semi-supervised based on a shorter response time and Gibson *et al.* (2010) find out that humans use manifolds for learning if there is no alternative simpler explanation of the data. Although these studies are important to understand human learning, it is still difficult to provide a convincing proof because most cognitive studies come with a weak implementation, e.g., having a separate supervised and unsupervised phase of learning (Lake and McClelland, 2011).

Despite those diverse statements from theory as well as cognition, manifold-based methods enjoy great popularity and they prove beneficial particularly in practice. First, they are quite fast in comparison to most cluster-based methods such as TSVM. But the main advantage comes certainly with the fact that they act mostly locally.

Thus outlier of the data distribution effect only local regions around them and do not impact the performance on the entire dataset.

2.2.3 Graph-based methods

As the name implies, graph-based methods operate on a graph structure that represents the underlying manifold. Nodes are the labeled and unlabeled data and edges reflect the pair-wise similarity among incident data points. The goal of the algorithms is then to find a function f that is smooth with respect to the labels as well as to the entire graph structure. These approaches can be divided into transductive (Blum and Chawla, 2001; Zhou *et al.*, 2004a) and inductive ones (Sindhwani *et al.*, 2005; Belkin and Niyogi, 2005).

Inductive methods. The inductive methods provide a natural extension to kernel methods by introducing an additional regularization term that considers also the global geometry of the data (Sindhwani *et al.*, 2006). The resulting optimization problem is then solved either with a squared loss function (LapRLS) or with a soft margin loss function (LapSVM) (Belkin *et al.*, 2006). There are some practical applications such as urban scene classification (Gomez-Chova *et al.*, 2008), semantic concept learning (Jiang *et al.*, 2008), or dynamic scene understanding in video sequences (Zhang *et al.*, 2008). Despite their benefits to classify also novel and unseen data, these methods are not well established due to their runtime complexity and their many parameters that are difficult to adjust.

Transductive methods. In contrast, transductive methods use the graph structure itself to spread labels from the labeled data to the unlabeled one (Bengio *et al.*, 2006). One of the first approaches was a mincut formulation (Blum and Chawla, 2001) that considers the positive labels as source and negative labels as sinks and solves the resulting st-cut problem. The main problem of this formulation comes with the hard decision values without confidence ranking. This is later addressed by Blum *et al.* (2004) with a bagging approach. But most common approaches to address graph-based learning are the so-called *label propagation* algorithms. These methods propagate labels by random walk with fixed original labels (Zhu *et al.*, 2003) or with a normalized graph Laplacian (Zhou *et al.*, 2004a) that allows a change of the original labels. There are several formulations for multi-label propagation (Kang *et al.*, 2006; Dharmadhikari *et al.*, 2012), for multiple instance learning (Tang *et al.*, 2010), or for multi-modality learning (Tong *et al.*, 2005). Finally, the applications ranging from movie rating (Goldberg and Zhu, 2006) over brain tumor segmentation (Li and Fan, 2012), protein classification (Xu *et al.*, 2012), part-of-speech tagging (Subramanya *et al.*, 2010), image segmentation (Grady and Funka-Lea, 2004), image colorization (Levin *et al.*, 2004), image retrieval (Li *et al.*, 2008) to activity recognition (Stikic *et al.*, 2011).

2.2.4 Relation to own work

In this thesis, we focus on the methods proposed by Zhou *et al.* (2004a) and Zhu *et al.* (2003) as our main goal is to improve the graph structure as well as the labels. For this purpose, these approaches are best suited because of their low complexity. The classification performance depends strongly on the graph structure thus there is a direct connection between graph structure and performance. Additional, there are only few parameters that do not need tedious fine tuning. In the following two sections, we review previous works for graph improvements (Section 2.3) as well as label improvements (Section 2.4).

2.3 GRAPH IMPROVEMENT

From the workflow of Figure 1.5 in the introduction, we identify four sources to improve the structure: data, image description, similarity notion, and the structure construction, i.e., in our case graph construction. Apart from the image description, we address all these issues in this thesis. In the following, we discuss previous literature for each of these parts separately. We start in Section 2.3.1 with an exploration of publications that provide faster algorithms to handle large amounts of unlabeled data we are aiming for. After that, we review in Section 2.3.2 state-of-the-art metric learning literature. Finally, we give in Section 2.3.3 a summary of methods that address the graph construction itself. For each of these subsections, we discuss separately the relations to our own work at the end of each part.

2.3.1 Scalable algorithms

Graph-based algorithms come at least with a runtime of $O(n^2m)$ with n the number of data and m the number of feature dimensions. This runtime is needed to compute all similarities between image pairs and to construct the graph. Thus the applied algorithm depends strongly on the number of data and the dimensionality of the features but also on the approach itself. For example, Zhou *et al.* (2004a) provide both a closed form solution that would need the inversion of a $n \times n$ matrix and an iterative procedure that is faster and often avoids over-fitting. In general, there are three different strategies to reduce the runtime: (i) a reduction of the data space ($\ll n$) to a representative subset of unlabeled data, (ii) an approximation either of the similarity matrix or the eigenvectors, or (iii) a parallelization of the approach.

(i) Data reduction. The most common approach to data reduction is clustering to find representative unlabeled data either by hierarchical clustering (Li and Zhou, 2011a), or by k-means clustering (Simon *et al.*, 2007; Liu *et al.*, 2010). Delalleau *et al.* (2005) propose a Greedy approach that starts with the labels only and successively add unlabeled samples farthest away from the current set of labeled and unlabeled data. Farajtabar *et al.* (2011) find similar nodes by spectral decomposition and merge these together. Another technique is to treat this task as an optimization problem that considers each point in a data set as a convex combination of a set of archetypical or prototypical examples either with a fixed number of archetypes (Cutler and Breiman, 1994) or with an automatically learned number of these prototypes (Prabhakaran *et al.*, 2012). These techniques are used, e.g., to find typical poses (Bauckhage and Thurau, 2009), or to summarize a video sequence (Elhamifar *et al.*, 2012).

(ii) Approximation. In contrast, Nystroem approximation is employed to approximate the entire kernel matrix. This approximation is estimated also on a subset of data that are retrieved either by random sampling (Zhang *et al.*, 2009) or with k-means clustering (Zhang *et al.*, 2008). This approximation can then be used to find a segmentation (Fowlkes *et al.*, 2004), for similarity search (Wang *et al.*, 2012), or face recognition (Talwalkar *et al.*, 2008). To speed up the algorithms, Fergus *et al.* (2009) propose an approximation of the eigenvectors of the normalized graph Laplacian. Tsang and Kwok (2006) solve the dual optimization problem by introducing a sparsity constraint, and Karlen *et al.* (2008) use stochastic gradient descent to solve the TSVM.

(iii) Parallelization. Instead of approximation and reduction, there is also the possibility to parallelize these approaches either by map-reduce (Rao and Yarowsky, 2009), or by organizing unlabeled data into subtrees that are processed in parallel (Wu *et al.*, 2012a). All previously mentioned works mainly focus on processing large datasets without showing the benefit for the graph structure itself when using a richer image collections. There are only few works that go beyond their given datasets by adding new images (Li *et al.*, 2007), by using other datasets to learn a new similarity measure (Wang and Forsyth, 2009), or by adding synthetic data points in the distance space (Yang *et al.*, 2012b) or in the feature space (Chawla *et al.*, 2002).

Relation to own work. In this work, we focus mainly on the question if more unlabeled data have a positive impact on the classification performance. In particular, we challenge the *the-more-data-the-better* strategy that is common sense in the computer vision community but also comes with an increase in runtime and space. But this question is difficult to answer as adding unlabeled data leads to a different field of research due to the dataset bias (Ponce *et al.*, 2006; Torralba, 2011) and the data source bias (Section 1.2.1). Therefore, we focus on ILSVRC 2010 with 1 million images and reduce this large amount of data to a representative subset of unlabeled data showing that this representative subset leads to a better graph structure than using all unlabeled data. We compare our approach to Delalleau *et al.* (2005) and Liu *et al.* (2010) that can be considered state-of-the-art methods to reduce the graph size. But in comparison to previous work, we analyze also the effect of more unlabeled data. Additionally, our work is the first attempt to process more than one million data points that is far more than $30,000$ data points used in previous work (Zhang *et al.*, 2008; Delalleau *et al.*, 2005; Liu *et al.*, 2010).

2.3.2 Metric learning

Usually, a distance measure such as L1 or L2 is used together with a Gaussian filter to express the similarity between image pairs and to build a graph structure. There are several studies (Sebe *et al.*, 2000; Aggarwal *et al.*, 2001) showing that L1 distance is better suited for high dimensional spaces than L2. More recently, Luxburg *et al.* (2010) show that a graph structure built with a L2 distance is meaningless the more data and dimensions are used as the average path length between two nodes is approximately 2. But the main problem of state-of-the-art distance measures is that they consider each dimension of the image descriptor equivalently. This problem becomes more apparent in computer vision as many images contain background clutter or occlusions so that only a small fraction of an image carry valuable information.

Therefore, metric learning is a promising direction to tackle this problem. These methods find a better data representation such that examples within a class are close together and examples from different classes are far away, i.e., small intra-class distances and large inter-class distances. Metric learning approaches can be split into (i) unsupervised, (ii) supervised, and (iii) semi-supervised methods that are further divided into global and local learning methods. See also Yang (2006) who provides a more detailed exploration of metric learning methods.

(i) Unsupervised metric leaning. Unsupervised methods have the advantage that they consider all data, i.e. labeled and unlabeled data. These methods are less prone to over-fitting in particular when only few labels are available. In general, these methods are accompanied by a dimensionality reduction and can be classified as global or local methods.

The most prominent global approaches are PCA (Pearson, 1901) or its extending kernel PCA (Schölkopf *et al.*, 1998), applied to face recognition (Turk and Pentland, 1991; Belhumeur *et al.*, 1997), Multiple Dimension Scaling (MDS), and Isomap (Tenenbaum *et al.*, 2000) that is a combination of PCA and MDS. More recently, there are also approaches that extract a new metric based on a so-called Flickr distance (Wang *et al.*, 2009a; Wu *et al.*, 2012b). They learn SVM classifiers on several Flickr groups and use the decision values of each classifier output to measure the similarity between concepts. The one-shot similarity kernel proposed by Wolf *et al.* (2009) is also similar computing distances of two images based on their similarity or difference to a negative set. Finally, Koestinger *et al.* (2012) optimize a Mahalanobis distance. Usually, these approaches are supervised as they need some constraints that are extracted from the labels. Instead, this work defines equivalence constraint via a likelihood ratio test.

In contrast, local methods such as LLE (Roweis and Saul, 2000), Laplacian Eigenmap (Belkin and Niyogi, 2003), and local tangent space alignment (LTSA) (**?**) exploit local neighborhood structure by building a graph structure and then perform a dimensionality reduction. Another approach is to estimate the intrinsic dimensionality of a submanifold in advance and then reduce the dimensions (Hein and Audibert, 2005). The main problem of most unsupervised methods is that they often cannot handle noisy and high dimensional data (Lee and Chang, 2005) due to the missing label feedback.

(ii) Supervised metric learning. Supervised methods use labeled data to guide the learning procedure and to enforce small distances between data from the same class and vice versa. Similar to unsupervised metric learning, these methods can be split into global and local approaches.

One of the classic global methods that can be also seen as the counterpart to PCA is linear discriminant analysis (LDA) (Fisher, 1936). Often this method is preferred in comparison to PCA because it also considers the class distribution. But this method tends to over-fitting when only few training examples are available (Martinez and Kak, 2001; Liu *et al.*, 2008). Most other global methods learn and optimize a Mahalanobis distance. The proposed methods essentially differ in the parameterization of the learned metric (including regularizers and constraints) and the optimization procedures. Some methods use only similarity constraints (Xing *et al.*, 2003) for optimization. Other approaches enforce both similarity and dissimilarity constraints to be fulfilled either within an information-theoretic framework (ITML) (Davis *et al.*, 2007; Kulis *et al.*, 2009; Saenko *et al.*, 2010), by an eigen decomposition (Globerson and Roweis, 2006; Kamvar *et al.*, 2003; Rangapuram and Hein, 2012), or by minimizing the empirical risk (Bian and Tao, 2007). Another set of algorithms maximize the inter-class distance by a large margin approach (LMNN) (Weinberger and Saul, 2009) that is later extended to handle also noisy side informations (Guillaumin *et al.*, 2009; Huang *et al.*, 2010a) or to incorporate knowledge about the invariance of the data (Kumar *et al.*, 2007; Hirzer *et al.*, 2012). As it mentioned before, these approaches require a large quantity of labeled data particularly in a high-dimensional space to generalize well. Furthermore, the optimization is often time-consuming. Therefore, there are several approaches proposing a faster optimization by using metric ball trees (Weinberger, 2008), a gradient descent procedure (Shen *et al.*, 2010), or treat the entire problem as an online learning problem that successively updates the metric (Shalev-Shwartz *et al.*, 2004; Jain *et al.*, 2010a).

Local methods enforce only local fulfillment of the constraints. In comparison to global methods, these approaches are more flexible and often easier to optimize. But they are also more prone to over-fitting depending on the labels. Most approaches extend global methods by integrating geometrical constraints. Neighborhood component analysis (NCA) (Goldberger *et al.*, 2005) optimizes a Mahalanobis distance by defining a soft neighborhood with probabilities. This framework is later augmented to include also a feature decomposition (Wang *et al.*, 2010) or to enforce sparsity (Hong *et al.*, 2011). Relevant component analysis (RCA) (Shental *et al.*, 2002) extends LDA with an additional weighting scheme. Cai *et al.* (2007b) enhance LDA as well by computing and optimizing two separate k-nearest neighbor graph structures for within-class examples and between-class examples. Stochastic Neighbor Embedding (SNE) (Hinton and Roweis, 2002) transforms LLE (Roweis and Saul, 2000) into a probabilistic formulation. Similarly, Yang *et al.* (2006) integrate probabilities into (Xing *et al.*, 2003). Babenko *et al.* (2009) explore the trade-off between global similarity metric learning and category specific metric learning and propose a joint framework that automatically find the best trade-off. Finally, Frome *et al.* (2007) learn a distance function for each exemplar that is improved by Malisiewicz and Efros (2008) using a SVM-based learning scheme.

(iii) Semi-supervised metric learning. Unsupervised metric learning techniques learn a metric that fits all data. But they do not use any supervision during the learning that might lead to distortion of the new data space in case of noisy data. In contrast, supervised methods use the label information but they learn this new metric only on the labeled data. This becomes problematic if there are only few labels with a high-dimensional feature representation. Semi-supervised metric learning tackles these problems by using both supervision and the geometry of the data.

Most methods extend supervised metric learning approaches with the graph Laplacian as an additional regularization term. Thus, Cai *et al.* (2007a); Song *et al.* (2008); Zhang and Yeung (2008) improve LDA, and Hoi and Lyu (2008) improve the work proposed by Xing *et al.* (2003). In contrast, Chen *et al.* (2005) and Lu *et al.* (2009) use the graph Laplacian itself to optimize pairwise constraints. Niu *et al.* (2012) incorporate an entropy regularization into supervised metric learning that enforces a low-density separation similar to Grandvalet and Bengio (2004). Uray *et al.* (2007) iterate between LDA and PCA. Another approach combines metric learning with clustering where the cluster assignment serves as the low dimensional manifold that is optimized either in an EM fashion (Bilenko *et al.*, 2004) or by Eigen-decomposition (Ye *et al.*, 2007; Okada and Nishida, 2010). Finally, Teramoto (2008) learns a random forest on labeled data, and computes proximities based on these decision trees on both labeled and unlabeled data.

Relation to own work. In this thesis, we analyze several supervised and unsupervised metric learning approaches with respect to a better graph construction. Particularly, we apply PCA (Pearson, 1901) and LDA (Fisher, 1936) to reduce the dimensionality of our feature representation and compare both methods. Furthermore, we also analyze ITML (Davis *et al.*, 2007). Instead of reducing the number of dimensions, it learns a weighting of the feature dimensions. The advantage of ITML is that it can be transformed into a kernelized optimization problem. Thus the runtime depends only on the number of labels that is usually smaller than the number of dimensions ($n \ll d$). Additionally, this approach shows state-of-the-art performance on Caltech 101 (Kulis *et al.*, 2009). Finally, we integrate ITML in a semi-supervised metric learning scheme that leads to an increased performance.

2.3.3 Graph construction

A graph for propagation is built on labeled and unlabeled data with some notion of similarity. Graph construction can be divided into approaches that build (i) a single graph based on pairwise relationships, (ii) a single graph based on higher order relationships with sets of images, i.e., hypergraph, and (iii) multiple graphs and combinations of those into a single graph structure. These are discussed separately in the following.

(i) Pairwise construction. In general, there are several types of graphs. The k-NN graph connects nodes to the k nearest neighbors and enforces that each node is connected at least to k nearest neighbors. Maier *et al.* (2009) study the influence of the parameter k with respect to spectral clustering. They show that k should be large enough to maximize the probability of the cluster identification. The ε-graph use a threshold ε to find the neighboring nodes. This construction method is less robust than k-NN construction as it leads often either to disconnected components or to a fully connected graph. Maier *et al.* (2008) show that clustering based on these two types of graphs converge to different solutions that means the behavior and the performance of both graph structures is different. In practice, k-NN graphs are preferred in comparison to the ε-graph due to it the robustness Maier *et al.* (2008). Finally, the fully connected graph uses all edges. A good edge weighting is crucial for these graphs. Additional, this construction method leads to a full matrix reducing the speed of the classification. Besides the run-time issue, k-NN graphs or ε-graphs are verifiable better than a full graph (Blatt *et al.*, 1997; Felzenszwalb and Huttenlocher, 2004).

In addition to these more general ways of graph construction, there are many other possibilities to improve the graph structure, e.g., by a better edge weighting or a re-arrangement of edges. These methods can be split into unsupervised and supervised approaches. On the unsupervised side, some publications propose a better weighting function for the edges in the graph. Usually, the edge weighting is done with a Gaussian kernel that is sensitive to the hyperparameter σ (band width). In Zhu *et al.* (2003), this parameter is learned automatically with a Laplace approximation. Wang and Zhang (2007a) replaces this Gaussian weighting by a new weighting based on the reconstruction error of the neighboring nodes. Another direction is to balance the graph structure such that dominant nodes are weighted down (Jebara *et al.*, 2009), or global statistical informations are encoded into the node degree through a ranking scheme (Qian *et al.*, 2011). Hein and Maier (2006) uses a diffusion process to remove the noise in the data.

There are several works that go beyond a fixed neighborhood of the k-NN graph and propose a graph construction using an adaptive neighborhood. Yang and Chen (2010) propose a sample-dependent graph by using the mean similarity of neighboring nodes as a threshold to insert or delete edges. Zhang *et al.* (2011b) parameterize LTSA (**?**) so that the number of edges per node are adaptively learned. Carreira-Perpinan and Zemel (2005) introduce local adaptation by combining multiple minimum spanning trees built on different subsets. Finally, there are several approaches combining graph construction with sparse coding (Wright *et al.*, 2009; Cheng *et al.*, 2010) that can be extended such that the Lasso regularization method is used for sparsity (Meinshausen and Bühlmann, 2006; Elhamifar and Vidal, 2009). Li *et al.* (2011) provide an evaluation of different graph structures but only on datasets with a smooth manifold structure.

On the supervised side, there are also works that address the weighting of the edges in the graph either by optimizing the leave-one-out error of the classifier to learn the hyperparameter σ (Zhang and Lee, 2006), within a Bayesian framework (Kapoor *et al.*, 2006), or with active learning (Zhao *et al.*, 2008a). Another approach avoids edges from unlabeled to labeled data to sharpen the influence of labeled nodes (Shin *et al.*, 2006). Liu and Chang (2009) learns a doubly-stochastic adjacency matrix from training examples to balance the graph structure and thus decrease the impact of dense regions. Some works use the decision values of a classifier to confirm or delete edges in a k-NN graph structure (Rohban and Rabiee, 2012) or to build a new graph on this decision values (Alexandrescu and Kirchhoff, 2007). Bertini *et al.* (2012) extend the k-associated optimal graph that connects only nodes if they belong to the same class to unlabeled data. Wang *et al.* (2008) propose an algorithm that jointly learns the graph structure and predicts the labels. Finally, there are also sparse coding formulations similar to the unsupervised approaches that additionally use the training data for optimization (Yan and Wang, 2009; He *et al.*, 2011).

(ii) Set construction. A hypergraph provide an opportunity to express higher order relations and to go beyond pairwise relations that might lead to an information loss and a missing generalization ability as shown in Figure 1.18. Additionally, this kind of construction can be considered as another way to reduce the number of nodes and to speed up the subsequent classification. In a hypergraph, each node represents a set of data points (images) and edges encode the similarity between these sets.

A first work in computer vision that uses a hypergraph was proposed by Agarwal *et al.* (2005) that approximates a hypergraph with a graph and use this graph in a standard clustering algorithm. Instead, Zhou *et al.* (2006) extend the spectral clustering to hypergraphs so that there is no need for a transformation in a simple graph. But this method lacks edge weighting that means each hyperedge is equally weighted. Huang *et al.* (2011) tackle this problem by summing up all pairwise similarities within a hyperedge, Huang *et al.* (2010b) use a fixed neighborhood around an image set to get a weighting that is extended to a adaptive neighborhood by Yu *et al.* (2012), and Wang *et al.* (2009b) estimate edge weights by the reconstruction error similar to LLE (Roweis and Saul, 2000). The applications range from image segmentation (Huang *et al.*, 2009) over multi-label classification (Sun *et al.*, 2008) to image matching (Zass and Shashua, 2008).

Another way to represent higher order relations is a tensor. They can be seen as a generalization of a matrix to a higher dimensional array. Govindu (2005) derives a similarity matrix from a set of points encoding geometric relations and applies this method to motion segmentation and geometric grouping. Shashua and Hazan (2005) use tensors to capture different illumination changes of one face, and Shashua *et al.* (2006) express 3D motion for an individual person as a tensor.

The main problem of today's hypergraph algorithms is that they are still close to pairwise relationship algorithms. This means the similarities are stored in an adjacency matrix that makes them large and sparse. Thus, the runtime and space complexity increases for these methods. The hyperedges often do not encode more information than simple pairwise edges. Therefore, the performance gain is often only minor. Ideally, one would have a concept or a part of a concept in each hypernode and the hyperedges should encode the similarity between these concepts. In this case, we can really benefit from a smaller graph size.

(iii) Multiple graphs. The combination of multiple graphs offers the possibility to capture different aspects in the data, e.g. with different image descriptors or distance measures. For graph-based methods, there are few works that combine graph structures similar to multiple kernel learning (MKL) (Argyriou *et al.*, 2005; Tsuda *et al.*, 2005). In Kato *et al.* (2009), they learn weights for combining graph Laplacian within an EM framework. Daitch *et al.* (2009) propose a method to find one graph from a set of graphs that best fits the data. Tong *et al.* (2005) formulate this combination as a multi-modality learning problem that fuses different modalities either linearly or sequentially. Balcan *et al.* (2005) use domain knowledge to extract three different sources, i.e., time, color and face features, that are combined with different hyperparameters. Finally, Goldberg *et al.* (2007) and Tong and Jin (2007) combine a similarity and a dissimilarity graph and apply label propagation on this mixed graph structure. Most of these previous works are developed for applications in bioinformatics. But more importantly, they combine often only graphs based on different parameters, i.e., a different number of neighbors k or different weight functions.

Relation to own work. In this work, we show the strong influence on the graph quality when combining different image descriptors leading to a completely different and more powerful graph structure. Additionally, we use the SVM output to construct a new graph. But in comparison to Rohban and Rabiee (2012) who use the SVM output to delete and insert existing edges, we build a complete new graph based on these decision values and combine this graph with our original graph. This leads to a richer and better connected graph structure than the graph structure in Rohban and Rabiee (2012).

2.4 ACTIVE LEARNING

Active learning is a well known strategy to reduce the amount of supervision to a small but representative subset and to improve the quality of the learner at the same time. This is also verified by cognitive science as Ashby (1992) shows that a higher accuracy is achieved with feedback during the learning in comparison to the scenario where supervision is only provided at the beginning. In machine learning, active learning leads in most cases to better performance. Angluin and Laird (1988) show that some NP-complete learning problems become polynomial in computation time. On the other side, active learning in combination with some classification algorithms might lead to poor performance, e.g., SVM with few examples at the beginning

(Wang *et al.*, 2003). Model selection is critical for these algorithm (Sugiyama and Rubens, 2008).

In this work, we focus on pool-based active learning. These methods consider all unlabeled data as a pool from which samples are drawn to be labeled. In general, pool-based methods can be divided by their sampling strategy into three different types. Exploitation-driven methods (Section 2.4.1) focus mainly on uncertain regions during the learning process. In contrast, methods based on exploration sampling (Section 2.4.2) estimate the overall distribution of the entire data space and query samples that represent and cover this space. Finally, there are also strategies that combine both exploration and exploitation (Section 2.4.3) to get samples that are uncertain but also diverse. In the following, we review related work for all these strategies and emphasize the difference to our own work in Section 2.4.4. We refer also to Settles (2009) who provides a general overview on different active learning strategies.

2.4.1 Exploitation-driven methods

Uncertainty-based active learning is most popular. This strategy queries for the least certain data point (Settles and Craven, 2008). In general, there are two strategies to calculate these uncertainties. Either a single classifier is used to find the most uncertain regions or multiple classifiers are applied to measure the disagreement of those classifier. The later one is also called Query-by-Committee and is more theoretically motivated (Seung *et al.*, 1992; Freund *et al.*, 1997; McCallum and Nigam, 1998). In practice, this type of sampling is too expensive in terms of training time and the parameter tuning of the several models is tedious. On one side, this committee should be diverse enough to get disagreements and on the other side a member of this committee should not dominate the other committee members.

Instead, active learning with a single classifier is more common in particular the combination with SVM. Schohn and Cohn (2000) and Tong and Koller (2001) are the first to combine active learning with SVM. Campbell *et al.* (2000) propose active learning with a soft margin SVM. But as we mentioned before, the success of SVM-based active learning strongly depends on the model specifically when only few labels are available. Therefore, Wang *et al.* (2003) suggest a bootstrapping method to tackle this problem while Luo *et al.* (2005) extend the approach proposed by Schohn and Cohn (2000) to an automatic model selection approach. Similarly, Wang *et al.* (2011) augment the method by Campbell *et al.* (2000) to a soft margin SVM with model selection by using a pseudo-validation set. More recently, there is also work that integrates the labeling cost into the decision function (Vijayanarasimhan and Grauman, 2009). Kapoor *et al.* (2006) incorporate prior information for face identification into the classic framework that is later augmented by Siddiquie and Gupta (2010) to integrate also contextual information for scene understanding. Joshi *et al.* (2009) extend SVM-based active learning to multi-class learning and analyze two different criteria for uncertainty.

Another direction is the combination of active learning with label propagation.

One of the first approaches in this direction was proposed by Zhu *et al.* (2003) that queries the next label within an expected risk minimization framework. Similar to this work is the approach proposed by Long *et al.* (2008) that maximizes the expected entropy reduction instead. Zhao *et al.* (2008a) publish a joint active sample selection and graph reconstruction method. Zhao *et al.* (2008b) extend the work by Zhu *et al.* (2003) to a large-scale setting that apply active learning on a smaller backbone graph containing only a subset of unlabeled data. Di and Crawford (2010) query labels which violate the consistency assumption. Finally, Lu and Ip (2010) integrate also the context into the learning scheme. Additionally, there are also several other algorithms that combine active learning with boosting (Abramson and Freund, 2005), with mixtures of Gaussians (Cohn *et al.*, 1996), with neural networks (Cohn *et al.*, 1994), with Co-Training (Muslea *et al.*, 2002), or within an Bayesian framework (Roy and McCallum, 2001).

On the application side, there are several works, e.g., video labeling (Yan *et al.*, 2003), image retrieval (Zhou *et al.*, 2004b; Jing *et al.*, 2004; Collins *et al.*, 2008; Jain *et al.*, 2010b), image classification(Holub *et al.*, 2008; Tang *et al.*, 2011), object detection (Vijayanarasimhan and Grauman, 2011), semantic segmentation (Vezhnevets *et al.*, 2012), transfer learning (Yang *et al.*, 2012a), and distance metric learning within a Bayesian framework (Yang *et al.*, 2007). Despite the success of these methods, they can run into problems by not providing enough coverage of the whole domain or focusing on outliers or inherently ambiguous parts of the data due to their discriminative nature.

2.4.2 Exploration-driven methods

Exploration-driven approaches consider the underlying data distribution of the unlabeled data and aim to find more representative samples. These methods can be divided into single mode learning and batch mode learning. While single mode learning asks only for one label per iteration, batch mode active learning queries a batch of labels in each iteration to speed up the active learning procedure.

Single mode active learning. This sampling scheme can be realized with a clustering to find representative samples, e.g., with hierarchical clustering (Buhmann and Zöller, 2000), k-means clustering (Kang *et al.*, 2004), or k-medoid clustering (Nguyen and Smeulders, 2004). Diversity is another approach for exploration that maximizes, for example the angular between two feature vectors (Dagli *et al.*, 2005), the Fisher information in the data (Hoiem *et al.*, 2006), the mutual information among labeled and unlabeled data (Guo, 2010), or the distance between new and closest labeled data point (Baram *et al.*, 2004; Xu *et al.*, 2007). Zhang *et al.* (2011a) look for data points that best reconstruct the entire dataset. Similar to clustering, density-based criteria seek for high density regions usually with an ε radius of the nearest neighborhood around each node (Hu *et al.*, 2010). Qi *et al.* (2008) propose a two-dimensional active learning that considers sample diversity as well as label diversity to ensure that each class is equally represented. Similarly, Hospedales *et al.* (2012) find at least one instance per class with a Dirichlet process mixture model. Finally, Pelleg and Moore (2004) use active learning to find anomalies and rare or novel categories via a mixture model that is later extended by He and Carbonell (2007) to handle also overlapping classes.

Batch mode active learning. Batch mode learning is usually not applied with exploitation-driven criteria (Section 2.4.1) due to the missing diversity of the requested labels. To incorporate this required diversity, Brinker (2003) propose a SVM-based batch mode learning by querying labels that are diverse in terms of their angle to each other in the feature space. Guo and Schuurmans (2007) formulate this task as a complex optimization problem that maximizes the discriminative classification performance. Similarly, Vijayanarasimhan *et al.* (2010) treat this as a continuous optimization problem with an additional cost term that takes also runtime into account. Chakraborty *et al.* (2011) propose a dynamic batch mode learning framework that jointly learns the size of the batch and selected samples within an optimization problem. Azimi *et al.* (2012) estimate the distribution of unlabeled data with a Monte-Carlo simulation and select samples that best match this distribution with a greedy approach. Finally, Fu *et al.* (2012) apply an uncertainty criteria and avoid redundant examples by considering the instance correlation. Most of these batch mode methods are not convenient in terms of runtime because they consider the problem as an optimization problem which makes them slow and they might often miss their actual target of speed-up.

The drawback of using exploration criteria alone is the missing feedback during the labeling process since their main goal is to sample evenly the data space without looking at the classification uncertainty. Consequently, many label requests are required to converge to a good solution.

2.4.3 Trade-off between exploration and exploitation

Through a combination of exploration and exploitation, both strategies take advantage from each other resulting in samples that are informative as well as representative. There are three different ways to combine those criteria: i) switching between both criteria with a certain threshold, ii) consecutively applying exploration and exploitation, and iii) a linear combination with a regularization parameter.

Switching. A simple implementation is proposed by Thrun and Moeller (1992). They switch randomly between uncertainty sampling and random sampling. Donmez and Carbonell (2007) augment the method published by Nguyen and Smeulders (2004) by introducing an additional threshold to switch between density and uncertainty sampling. Baram *et al.* (2004) use a multi-armed bandit (MAB) formulation to switch among three single criteria, i.e., entropy, expected loss, and kernel farthest first that look for samples that are farthest away from current labeled set. Osugi *et al.* (2005) reformulates this work into a simpler but less flexible one. Finally, Maes *et al.* (2012) offer a more theoretical discussion on how multi-armed bandit problems can be used to solve the exploration-exploitation dilemma and how they can be extended to incorporate also prior knowledge about the target class.

Series. Another possibility is to consecutively employ both strategies. Xu *et al.* (2003) find first the uncertain region, i.e., the margin of the SVM, and apply then clustering in this area to get the most representative samples in this margin. Zhu *et al.* (2010) use label propagation to select most uncertain unlabeled data in terms of overall entropy and rank these samples by their density measured with the cosine angle between two samples.

Linear combination. This strategy merges both exploration and exploitation criteria to rank the unlabeled data and request a sample. Cebron and Berthold (2009) propose a weighted combination of two criteria. Additionally, they introduce a new density criteria *node potential* that implements the strategy more exploration at the beginning and more exploitation at the end. The trade-off between both criteria must be set manually or by cross validation. Huang *et al.* (2010a) provide a min-max formulation to balance between prediction uncertainty of labeled data (exploitative) and prediction uncertainty of unlabeled data (explorative). Krause and Guestrin (2007) learn the trade-off within a Gaussian process. Bondu *et al.* (2010) dynamically balance the trade-off by estimating the gain for the next iteration with a Bayesian formulation. Despite the strong progress of more holistic models, these approaches often come with high computational costs, difficult configurable parameters, unbalanced terms, and missing flexibility in terms of more criteria or time-varying trade-offs.

2.4.4 Relation to own work

In this thesis, we address these issues by proposing a reinforced active learning formulation (RALF) that considers the entire active learning sequence as a process. Our approach can deal with multiple criteria, is able to have time-varying trade-offs between exploration and exploitation, and is fast and efficient due to a compact parameterization of the model without dataset-specific tuning. In comparison to Baram *et al.* (2004) who also use a reinforcement procedure, our model comes with fewer parameters, more flexibility in terms of sampling criteria, and provides always a linear combination of exploration and exploitation instead of switching between criteria. For the linear combination, we extend the work proposed by Cebron and Berthold (2009) to a time-varying combination that leads to a better adaptivity to dataset requirements.

GRAPH-BASED SEMI-SUPERVISED LEARNING 3

Contents

3.1 Graph construction **39**
  3.1.1 Image representation 40
  3.1.2 Image Similarity 41
  3.1.3 Graph construction 42
3.2 Classification Algorithms **43**
3.3 Datasets **44**
3.4 Experiments **46**
3.5 Conclusion **49**

This thesis focus mainly on graph-based algorithms as they are well established among semi-supervised algorithms. The way they exploit neighborhood structure is intuitive and the computational demands are usually moderate. One of the key issue of these methods is the construction of the graph. This critical aspect is often neglected (Zhu, 2006) and meaningful neighborhood relations as well as a class structure is assumed to be encoded in the distances of the raw feature space. We have shown in a recent study (Ebert *et al.*, 2010) that for visual categories those assumptions cannot be taken for granted and that the quality of the graph is in fact highly correlated with the final performance. This strongly suggests that learning should start before a neighborhood structure is imposed on the data points in order to surpass the inherent limitations of traditional semi-supervised learning schemes.

In this chapter, we explore different graph structures that are explained in Section 3.1. Then we review state-of-the-art algorithms for label propagation in Section 3.2. In Section 3.3 we introduce four different datasets for image classification with different degrees of difficulty that are used for the analysis and the evaluation of our proposed methods. In section 3.4 we empirically show that the performance depends more on the quality of the neighborhood structure induced by the image representations, the similarity measure, and on the parameters of the graph structure rather than on the particular algorithms employed. Finally, we conclude our comprehension in Section 3.5.

3.1 GRAPH CONSTRUCTION

The foundation and the heart of all graph-based algorithms is the graph itself. Figure 3.1 shows a cutout of a k-nearest neighbor graph structure with images (i.e.

image description) as nodes and edges that reflect the similarity between image pairs. The quality of this structure depends on the construction of the graph as well as the available and used data. The second dependency – the relationship between more data and graph quality – will be explored later in Section 7. In this chapter, we focus on the first dependency – the graph construction. This step can be further divided into three parts: representation of images (Section 3.1.1), similarity between image pairs (Section 3.1.2), and the definition of the graph structure (Section 3.1.3), e.g., symmetric relations or the number of neighbors. In the following we briefly describe all these stages.

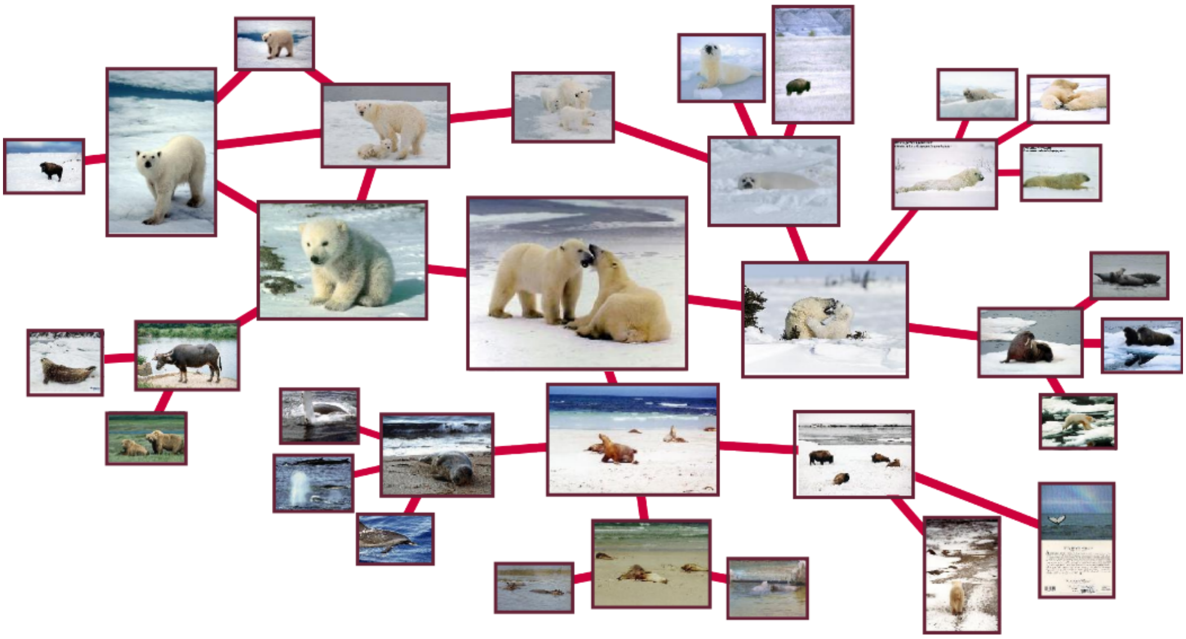


Figure 3.1. A cutout of a symmetric k nearest neighbor graph structure.

3.1.1 Image representation

We explore eight different state-of-the-art image descriptors: four global descriptors, i.e., HOG, Gist, and spatial dense SIFT with and without color, and four local descriptors, i.e., three patch local binary pattern (TPLBP), self similarity (SSim), and dense SIFT (DSIFT) with and without color.

HOG. The normalized histograms of oriented gradient descriptor (HOG) is proposed by Dalal and Triggs (2005). We extract this representation with cells of 8×8 pixels and 9 orientations leading to a 556-dimensional vector. Usually this setting shows a good trade-off between specificity and information loss when computing on the entire image. Other subdivisions, e.g. into cells of 4×4 pixels or 16×16 pixels, show often worse performance.

Gist. This feature, originally named *spatial envelope*, is developed by Oliva and Torralba (2001) to describe the shape of a scene as a set of perceptual properties such as naturalness, openness, or roughness. For our experiments we use their implementation. We get a descriptor with 960 dimensions.

TPLBP. Three patch local binary pattern proposed by Wolf *et al.* (2008) is a texture descriptor. This descriptor introduce a spatial notion by considering three neighboring patches of size 3×3 arranged in a circle around each pixel. We use patches of size 8×8 leading to $2^8 = 256$ binary words. Finally, the occurrences of these words are counted for each subdivision so that we get a $8,960$-dimensional feature vector.

SSim. Local self-similarity (Shechtman and Irani, 2007) captures the internal geometric layout within images and is similar to mutual information (MI) that uses the statistical co-occurrence of pixel intensities across images. We use the implementation VLFeat (Vedaldi and Fulkerson, 2008) and quantize the resulting representation into $1,000$ visual words. Thus our final descriptor is $1,000$-dimensional.

DSIFT. Dense SIFT (DSIFT) is the scale invariant feature transform (SIFT) method applied at a dense grid of locations at a fixed scale and orientation. Similar to SSim, this descriptor is extracted with the implementation VLFeat proposed by Vedaldi and Fulkerson (2008). SIFT features are calculated on a regular grid and quantized into $1,000$ visual words.

DSIFT Color. The color version is computed on the HSV channels. This color representation is more robust in comparison to RGB as illumination changes are encoded in one channel and not distributed into three channels.

Spatial DSIFT (Color). Finally, DSIFT and DSIFT Color are extracted as mentioned above. For the spatial version of these descriptors, we use a subdivision of 4×4 that are concatenated to a final histogram representation similar to Lazebnik *et al.* (2006). This results in a $9,000$-dimensional bag-of-word representation.

3.1.2 Image Similarity

Among all stages of graph construction, the calculation of the distance matrix is the most time-consuming step as it comes with a time complexity of $O(n^2m)$ with n the number of images and m the number of dimensions. The most common distance measure that has been used for graph-based algorithms is the Euclidean distance (L2) although it is common sense that this measure cannot handle large dimensional feature spaces. In addition to this distance we also explore the Manhattan distance (L1) that is more robust with respect to outliers.

Both distance measures, the Euclidean distance

$$d(x_i, x_j) = \|x_i - x_j\|_2 = \sqrt{\sum_{k=1}^{m} (x_{ik} - y_{jk})^2} \tag{3.1}$$

and the Manhattan distance with m the number of dimensions,

$$d(x_i, x_j) = \|x_i - x_j\|_1 = \sum_{k=1}^{m} |x_{ik} - y_{jk}| \tag{3.2}$$

are then transformed into similarities with a Gaussian kernel

$$s(x_i, x_j) = \exp\left(\frac{-d(x_i, x_j)}{2\sigma^2}\right). \tag{3.3}$$

The width σ of this kernel is dataset dependent and can cause that the algorithm does not work at all. But once the range is found in that the bandwidth σ works fine, the differences are marginal, i.e., in the order of 10^{-2} to 10^{-3}. For our experiments, we use a neighborhood heuristic to find a good value for σ. We sort all distances for each image in ascending order and compute the average over the first k neighbors for each image, i.e.,

$$\sigma = \sqrt{\frac{1}{nk} \sum_{i=1}^{n} \sum_{j=1}^{k} d(x_i, x_j)} \tag{3.4}$$

with the k smallest distance $d(x_i, x_j)$ for each image and $i \neq j$. The resulting σ is usually close to the optimal value.

3.1.3 Graph construction

We build a k-nearest neighbor graph based on our previously defined similarities

$$G_{ij} = \begin{cases} s(x_i, x_j) & \text{if } s(x_i, x_j) \text{ is among the } k \text{ largest} \\ & \text{similarities of } x_i \\ 0 & \text{otherwise.} \end{cases} \tag{3.5}$$

Finally, we transform this graph into a symmetric one by summing up edges in both directions, i.e.,

$$W_{ij} = G_{ij} + G_{ji} \tag{3.6}$$

with $0 \leq W_{ij} \leq 2$. Therefore, bidirectional edges get a much higher weight than unidirectional edges. The resulting graph structure has usually a better connectivity than a asymmetric graph structure. Although this leads sometimes to so called *hub* nodes that are connected to a large number of nodes (Luxburg *et al.*, 2010), the overall performance and quality is significantly better in comparison to the asymmetric version.

Other construction methods like ε-graph, that uses a threshold ε to cut down the number of edges, or the full graph are either worse in performance due to isolated nodes or their computational demand. Also a hybrid solution of the ε-graph and k-NN graph did not improve our results.

3.2 CLASSIFICATION ALGORITHMS

Graph-based methods distribute labels from labeled data to unlabeled data. In our experiments, we compare four methods covering a broad range of existing strategies. These methods are designed for binary problems. But they are expandable to multi-class problems with C classes. For this purpose, the original learning problem is split into C one-versus-all binary problems that are solved on the same underlying graph structure.

All algorithms follow the same pattern. Given $n = l + u$ data points with l labeled examples $\{(x_1, y_1), ..., (x_l, y_l)\}$ and u unlabeled ones $x_1, ..., x_u$. $x_i \in \mathbb{R}^m$ are the m-dimensional feature vectors, $y_i \in \mathcal{L} = \{1, ..., C\}$ are the labels, and C is the number of classes. First, labels are initialized for each class $1 \leq c \leq C$, i.e.,

$$Y_c^{(0)} = (y_1^c, ..., y_l^c, 0, ..., 0) \tag{3.7}$$

with

$$y_i^c = \begin{cases} 1 & \text{if } y_i = c \\ -1 & \text{otherwise.} \end{cases} \tag{3.8}$$

Unlabeled data are initialized with 0. Then labels are updated iteratively, i.e.,

$$Y_c^{(t+1)} \leftarrow L Y_c^{(t)} \tag{3.9}$$

with $1 \leq c \leq C$ and Y_c^* the limit of this sequence. Typically a small number of iterations is used to avoid over-fitting. L is the term that varies for each algorithm and sometimes there is also an additional regularization term. The final prediction is obtained by

$$\hat{Y} = \operatorname{argmax}_{1 \leq c \leq C} Y_c^* \tag{3.10}$$

In the following we briefly explain the differences of four different algorithms that we evaluate in our experiments.

Gaussian Fields Harmonic Functions (LPZhu) Zhu *et al.* (2003) uses a transition probability matrix

$$L = D^{-1} W \tag{3.11}$$

with the diagonal matrix

$$D_{ij} = \begin{cases} \sum_{k=1}^{n} W_{ik} & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases} \tag{3.12}$$

to propagate labels. After each iteration labels of l labeled data are set to the original label $Y_c^{(0)}$.

Quadratic Criterion (LPJacobi) Bengio *et al.* (2006) propose a variant of the previous method allowing the original labels to change which can be helpful for ambiguous representations. They also introduce a regularization term for a better numerical stability resulting in the following propagation scheme

$$Y_c^{(t+1)} \leftarrow A^{-1}\left(\mu W Y_c^{(t)} + Y_c^{(0)}\right) \tag{3.13}$$

with

$$A = I_{[l]} + \mu D + \mu \epsilon \tag{3.14}$$

and parameter $\mu = \frac{\alpha}{1-\alpha} \in (0, +\infty)$, $\alpha \in (0,1)$, and a small $\epsilon > 0$.

Local Global Consistency (LPZhou) Zhou *et al.* (2004a) use a normalized graph Laplacian

$$L = D^{-\frac{1}{2}} W D^{-\frac{1}{2}} \tag{3.15}$$

instead of transition probabilities. The initial labels are also allowed to change but with a regularization parameter $\alpha \in (0,1]$. The labels are spread by

$$Y_c^{(t+1)} = \alpha L Y_c^{(t)} + (1-\alpha) Y_c^{(0)}. \tag{3.16}$$

Discrete Regularization (DiscreteReg) Zhou *et al.* (2005) incorporate local graph properties by looking at the degree of two neighboring nodes that is stored in the degree matrix D. An additional cost function

$$F_{ij} = \begin{cases} \frac{1}{1+\mu} \frac{W_{ij}}{\sqrt{D_{ii} D_{jj}}} & \text{if } i \neq j \\ \frac{\mu}{1+\mu} & \text{otherwise.} \end{cases} \tag{3.17}$$

reduces the influence of nodes with many connections, i.e.

$$Y_c^{(t+1)}(x_i) = \sum_{j=1}^{n} F_{ij} Y_c^{(t)}(x_i) + F_{ii} Y_c^{(0)}(x_i). \tag{3.18}$$

3.3 DATASETS

We analyze in this work several state-of-the-art datasets for image classification with an increasing number of object classes and different levels of difficulty.

ETH80. This dataset is introduced by Leibe and Schiele (2003) and contains 3,280 images divided into 8 object classes (*apple*, *car*, *cow*, *cup*, *dog*, *horse*, *pear*, and *tomato*) and 10 instances per class (Figure 3.2 left). Each instance is photographed from 41 viewpoints in front of a uniform background (Figure 3.2 right). This is an almost ideal dataset for SSL as it contains a smooth manifold structure and no background clutter. Each class is mapped to one dense region. Therefore, this dataset serves also as a setting to show that SSL can work better than supervised learning.

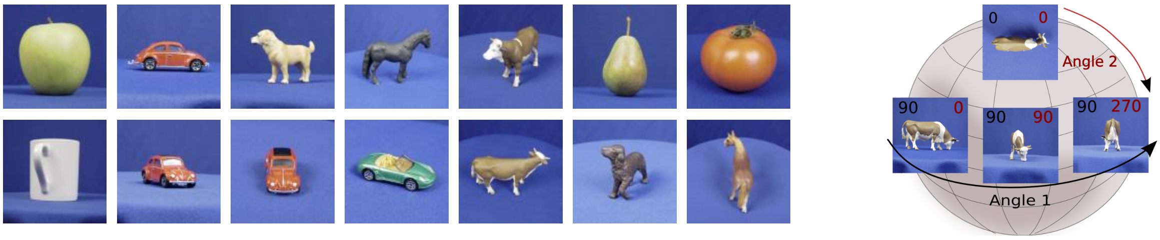


Figure 3.2. Sample images of ETH80 (left) and a visualization of the viewpoint angles (right).

C-PASCAL. We proposed Cropped PASCAL in (Ebert *et al.*, 2010) where we use the bounding box annotations of the PASCAL VOC challenge 2008 training set (Everingham *et al.*, 2008) to extract the objects such that classification can be evaluated in a multi-class setting. The resulting dataset contains 4,450 images of aligned objects from 20 classes but with varying object poses, challenging appearances, background clutter, and truncation (Figure 3.3).

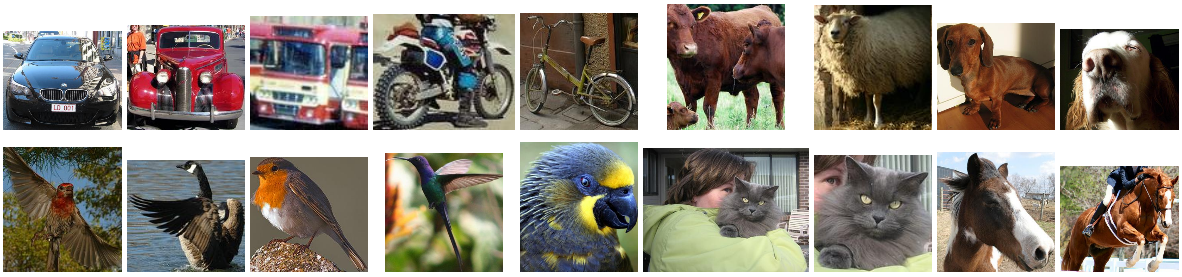


Figure 3.3. Sample images of C-PASCAL.

Caltech 101. This dataset is published by Fei-Fei *et al.* (2006) with 9,144 images and 101 object classes. Objects are located in the middle of the image, but there are still background clutter, a large intra-class variability, drawings, or multiple instances of an object in one image.

ILSVRC 2010. One of the state-of-the-art datasets for large-scale image classification is ILSVRC 2010 with 1,000 categories and approx. 1.26 million images. This is a subset of ImageNet provided by Deng *et al.* (2009). Objects can be anywhere in an image and images contain background clutter, occlusions, or truncations. In some of

Figure 3.4. Sample images of Caltech 101.

our experiments, we use also a subset of this dataset that we call **IM100**. This subset contains 100 classes similar to Caltech 101 with approx. 130,000 images.

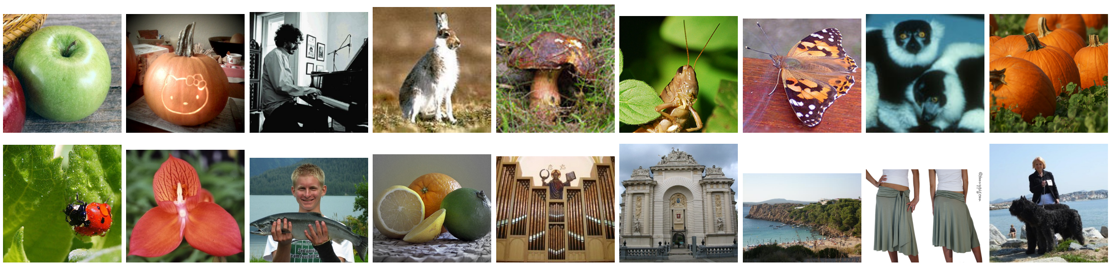


Figure 3.5. Sample images of ILSVRC 2010.

Preprocessing. ETH80 consists of squared images with size 100 × 100. Thus, there is no preprocessing of images necessary. Caltech101 and C-PASCAL vary in their image size such that rescaling is required as most descriptors need for comparability the same image size. In this work, we extend the smaller side of the image to get a squared image and rescale it to 120 × 120. Other methods like directly rescaling the image to squared form distort the objects and lead to worse performance. Also, it is not possible to find a rectangular form that fits most of the images as there are almost the same amount of both landscape and portrait images. As a last preprocessing step, we increase the contrast of each image by adjusting intensity values into the range of $[0, 1]$ such that low and high intensities are saturated to 1%. This improves all image descriptors up to 2% and in particular the color descriptors.

3.4 EXPERIMENTS

The main purpose of this section is to look on different graph structures, their qualities, and their influence on the final result. Therefore, we question two aspects: (i) What are the differences among the previously mentioned four algorithms if we use the optimal parameter setting? and (ii) How large is the impact of the underlying graph structure on the final classification performance?

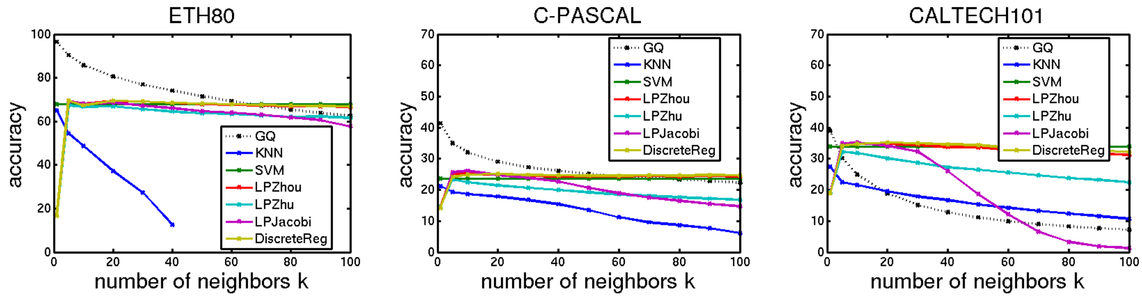


Figure 3.6. Overall accuracy for HOG and 5 labels per class with different algorithms over the number of neighbors k. Graph Quality (GQ) is a theoretical measure calculating the number of correct nearest neighbors. KNN and SVM are the supervised algorithms. Note SVM is a straight line because there are no different neighborhood structures.

(i) Different algorithms. In the first experiment, we fix the number of labeled data to 5 randomly selected training example per class and vary the number of nearest neighbors k. Additional, we compare to the supervised k-nearest neighbor algorithm (KNN) and SVM with an RBF kernel. Linear SVMs show always worse performance (up to $5\% - 10\%$) in comparison to the kernelized versions, so that we skip these results. For SVM, we use LibSVM (Chang and Lin, 2011) and determine best parameters by cross validation.

Figure 3.6 shows results for HOG, L1 distance, and all three datasets. We plot overall accuracy for different numbers of neighbors k for the graph construction. *Graph quality* (GQ, dotted line) indicates the accuracy of the nearest neighbors that means the number of nearest neighbors $\mathcal{N}(x_i)$ with the same label as x_i, i.e.,

$$GQ = \frac{1}{n}\sum_{i=1}^{n} \frac{\sum_{x_j \in \mathcal{N}(x_i)} \mathbf{1}_{[y_i = y_j]}}{|\mathcal{N}(x_i)|}. \tag{3.19}$$

For testing, we assume all labels Y for labeled as well as unlabeled data are known. This measure serves only as a theoretical measure to get a better intuition of the quality of the graph structure as we expect a high correlation between GQ and classification performance. SVM (red line) is plotted as a straight line because there are no different neighborhood structures.

From Figure 3.6 we can easily observe that the difference according to the performance among the SSL algorithms are only minor. The accuracy ranges from 69.3% (LPJacobi) to 69.5% (LPZhou) for ETH80. The only exception is LPZhu with 67.4% because of the missing regularization term and the simple propagation matrix. Another important observation is that all SSL algorithms apart from LPZhu have better performance than SVM with 67.8% and KNN with 64.9% for ETH80. Clearly, unlabeled data convey important informations about the global data distribution in particular when only few labeled data are available.

(ii) Different graph structures. In the second experiment, we fix the number k of nearest neighbors to 10, and vary the graph structure based on different distance measures and image descriptors. In Figure 3.7, these results are plotted for all three

		L1			L2			L1-L2
desc	SVM	GQ	KNN	LP	GQ	KNN	LP	GQ
ETH80								
HOG	67.8	85.7	64.8	68.0	84.3	62.0	**69.2**	1.4
DSIFT	65.0	85.7	69.5	**74.2**	83.6	64.4	72.0	2.1
SpDSIFT	65.6	84.8	67.8	**72.1**	83.3	63.2	70.0	1.5
C-PASCAL								
HOG	23.6	32.0	21.1	25.2	28.5	17.3	22.6	3.6
DSIFT	21.2	31.1	22.9	24.6	25.8	17.3	19.4	5.2
SpDSIFT	21.0	33.5	24.6	27.0	26.3	14.1	19.7	7.3
Caltech101								
HOG	33.9	24.9	27.4	**34.8**	20.8	21.4	29.5	5.4
DSIFT	22.0	22.5	27.3	**31.0**	14.1	16.9	20.0	11.0
SpDSIFT	27.4	27.0	33.5	**37.8**	16.4	18.7	23.2	14.7
mean	38.6	47.5	39.9	42.8	42.6	32.8	38.4	5.8

Table 3.1. Overall accuracy for the three best image descriptors HOG, Dense SIFT (DSIFT), and Spatial DSIFT (SpDSIFT) with L1 and L2 distance. Last column is the difference between L1 and L2 distance for the graph quality (GQ) of the 10-NN graph (2nd and 5th column). Last row is the average over all datasets and descriptors in this table.

datasets. As we can see, there is a large variability among the different descriptors. For ETH80, accuracy ranges from 69.0% (HOG) to 75.7% (DSIFT Color), and for Caltech101 from 22.9% (DSIFT Color) to 37.0% (Spatial DSIFT).

As a next step, we vary also the distance measure, e.g., we use L1 and L2 distances. Results for 5 training samples can be found in Table 3.1 for the best three descriptors in average: HOG, DSIFT, and SpDSIFT. These descriptors are sorted by their dimensionality starting with HOG ($d = 576$) over DSIFT ($d = 1000$) to SpDSIFT ($d = 16000$). The last column shows the difference between L1 and L2 accuracy for LP. From these column, we can easily observe that L1 is always better than L2 distance. In addition, this difference increases the more dimensions are involved. For Caltech101, the difference for HOG is only 5.4% between L1 with 34.8% and L2 with 29.5% while for SpDSIFT this difference increases to 14.7% between L1 with 37.8% and L2 with 23.2%. This emphasizes that the graph structure is highly sensitive not only to the used image descriptor but also to the distance measure. For Caltech101, there is almost a factor of three between the worst performance of SpDSIFT Color and L2 with 13.7% and the best performance of SpDSIFT and L1 with 37.8%. Not even the best SSL algorithm is able to compensate these deficits.

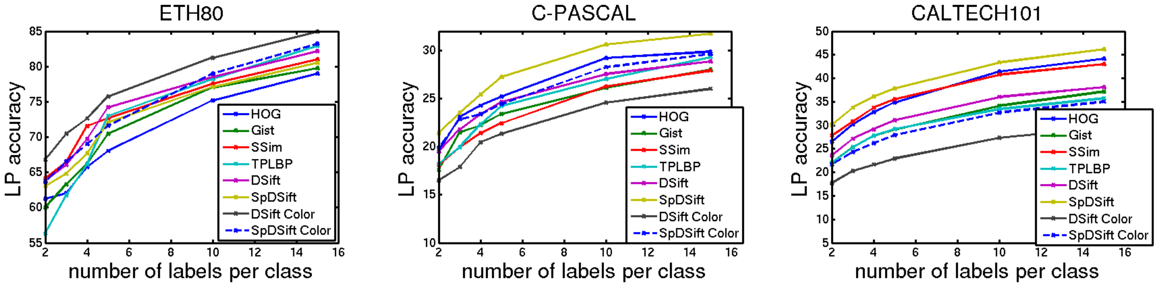


Figure 3.7. Overall accuracy for different descriptors with LP Zhou and different number of labels per class averaged over 5 different runs.

3.5 CONCLUSION

In this chapter, we looked at different graph structures caused by different image representations, and different similarity measures. We review four state-of-the-art graph-based algorithms that cover a broad range of existing methods. We then show empirically that graph structure is more important for graph-based methods than the algorithm itself. A good image descriptor in combination with a good distance measure leads to a much better graph structure and thus a much higher performance than using a different algorithm. Furthermore, we show that the L1 distance is more robust against noise than the L2 distance particularly in high dimensional spaces.

These results confirm the statement of Zhu (2006) that graph structure is more important than the algorithm itself and motivates us to spend more effort on improving the graph quality. In the following chapter, we explore and propose different unsupervised as well as supervised improvements of the graph structure leading to a better classification performance.

GRAPH IMPROVEMENT

4

Contents

4.1 Introduction **52**
4.2 Unsupervised Improvements **53**
  4.2.1 Dimensionality Reduction with PCA 53
  4.2.2 Increasing the dataset 54
  4.2.3 Combination 55
  4.2.4 Summary 56
4.3 Supervised Improvements **57**
  4.3.1 Dimensionality Reduction with LDA 58
  4.3.2 SVM-based Graph Construction 58
  4.3.3 Information theoretic metric learning (ITML) 60
  4.3.4 Summary 62
4.4 Semi-Supervised Improvement with IMLP **64**
4.5 Conclusion **66**

In the previous chapter, we show that the success of graph-based methods critically depends on the neighborhood relations in the data (Ebert *et al.*, 2010). This strongly suggests that learning and improving this neighborhood structure should start before the construction of the graph to surpass the inherent limitations of traditional semi-supervised learning schemes. In this chapter, we systematically explore a broad range of graph improvements from unsupervised to supervised and look at the impact on this pre-existing neighborhood structure. We analyze advantages and disadvantages of each method and show which of these approaches are more promising. Finally, we propose a new semi-supervised structure improvement.

We organize this chapter as follows. After reviewing most relevant existing work in Section 4.1, we present different unsupervised improvements in Section 4.2 that intervene before the actual graph construction. We show the influence of both representation of images as well as the dataset itself. In Section 4.3, we analyze and propose different supervised improvements by learning or extracting a better metric instead of using standard euclidean distance. This enables us to generate a graph structures that represents better the underlying manifold structure. Finally, we suggest in Section 4.4 our new semi-supervised improvement that we call *interleaved metric learning and label propagation (IMLP)* with which we show state-of-the-art performance on Caltech 101.

4.1 INTRODUCTION

As we mentioned before, graph construction is one of the key issue of graph-based methods. Usually, a k-nearest neighbor structure is build based on the Euclidean distance matrix that is finally weighted by a Gaussian kernel. Although a good graph structure is a crucial issue, it is surprising how little attention graph construction has received in comparison to various algorithmic contributions. Most of these methods towards better graph construction (Wang and Zhang, 2007b; Shin *et al.*, 2006; Zhang and Lee, 2006; Zhang *et al.*, 2011b; Jebara *et al.*, 2009) often assume that the pre-exisiting neighborhood structure in the data is already meaningful and propose, e.g., a better weighting function for already existing connections. Of course this leads to an improvement but is often not more than a finetuning step. However the real foundations of a good neighborhood structure – the representation of data, the distance measure, and the dataset itself – are often overlooked. This becomes more prominent in computer vision, where we have to deal with large intra-class, small inter-class variability, background clutter, and truncated or occluded objects (see Section 1.2). Therefore, we have to start earlier in this pipeline with better image representations, better datasets, and better distance measures.

Recent work to graph construction can be divided into unsupervised and supervised approaches. On the unsupervised side, some publications propose a better weighting function for the edges in the graph. Usually, this weighting is done with a Gaussian kernel that is sensitive to the hyperparameter. In Zhu *et al.* (2003), this parameter is learned automatically while Wang and Zhang (2007a) replaces this Gaussian weighting by a new weighting based on the reconstruction error of the neighboring nodes. Another direction is to balance the graph structure such that dominant nodes are weighted down (Jebara *et al.*, 2009), or that the number of edges per node are adaptively learned (Zhang *et al.*, 2011b). Hein and Maier (2006) use a diffusion process to remove the noise in the data. Finally, there is also work that combines similarities and dissimilarities in one graph (Goldberg *et al.*, 2007). Almost all these methods assumes a smooth manifold structure of the data that can be rarely found in computer vision.

On the supervised side, there are also works that address the weighting of graph either by optimizing the leave-one-out error of the classifier to learn the hyperparameter σ (Zhang and Lee, 2006), or with active learning (Zhao *et al.*, 2008a). Another approach avoids edges from unlabeled to labeled data to sharpen the influence of labeled nodes (Shin *et al.*, 2006). Liu and Chang (2009) learn a doubly-stochastic adjacency matrix from training examples to balance the graph structure and thus decrease the impact of dense regions. Some works use SVM decision values to confirm or delete edges in a k-NN graph structure (Rohban and Rabiee, 2012).

In this work, we explore different unsupervised and supervised methods that have an impact on the graph structure and analyze the advantages and disadvantages of these improvements. Finally, we propose a novel semi-supervised framework that tackles the drawbacks of the previously analyzed methods.

4.2 UNSUPERVISED IMPROVEMENTS

Unsupervised methods concentrate mainly on the structure in the data and try to improve the clustering property based on some similarity notion. The advantage lies in the consideration of the overall data structure. Accordingly, changes have a wide influence on the entire data space. One disadvantage comes in when the data itself contains much noise. This can result in distortions of the original data space as there is no supervision involved that controls these transformations.

In the following, we explore three common unsupervised methods. Each of these methods work well in many applications but they are not explored in the context of graph-based algorithms. We address the representation of images as well as the quality of the datasets. For a better image description, we use dimensionality reduction with PCA to make our high dimensional image representation more suitable for Euclidean distance. PCA is a well-known strategy to improve face recognition (Turk and Pentland, 1991) or to cope with large image collections (Fergus *et al.*, 2009). We also show the importance of this preprocessing step in Section 4.2.1.

After that, we increase the quality of datasets in Section 4.2.2 by flipping all images so that in particular global image descriptors can better deal with different viewpoints. Finally we combine different image descriptors represented as single graph structures into one final graph in Section 4.2.3. The resulting graph is characterized by a better connectivity and by a more reliable edge weighting because edges that appear in several graph structures get a much higher weight than edges that appear only for one of these single graph structures. In this work, we show the strong influence on the graph quality and then combine different image descriptors leading to a more powerful graph structure. Finally, we consecutively apply all three strategies to show the overall improvement in Section 4.2.4.

4.2.1 Dimensionality Reduction with PCA

The first improvement addresses the high dimensionality of our descriptors. As we have seen in the previous chapter in Table 3.1, the difference between L1 and L2 distance is larger with increasing dimensionality. Thus, the performance drops significantly for graph structure build with the L2 distance. PCA is a well-known method to reduce the dimensionality of a representation.

Table 4.1 shows results for L1 and L2 with PCA and the difference to our baseline in Table 3.1. In our experiments, we reduce the number of dimensions to 100. As expected, PCA improves the results of L2. For Caltech101, SpDSIFT the descriptor with 16000 dimensions is improved by 12.3% from 23.2% without PCA to 35.5% with PCA that is close to the performance of L1 without PCA with 37.8%. The same can be observed for C-PASCAL. ETH80 is a special case as there is no background clutter and almost identical objects in the middle so that there is less noise in the data representation. A reduction of the dimensions leads to an information loss and to a worse classification performance.

	GQ				LP			
descriptor	L1+PCA	gain	L2+PCA	gain	L1+PCA	gain	L2+PCA	gain
	ETH80							
HOG	84.5	-1.2	84.3	-	70.1	2.1	69.0	-0.2
DSIFT	83.8	-1.9	83.3	-0.4	71.9	-2.4	71.5	-0.4
SpDSIFT	83.3	-1.5	82.9	-0.4	68.2	-3.9	67.1	-2.9
	C-PASCAL							
HOG	28.8	-3.2	30.4	1.9	22.6	-2.6	23.7	1.1
DSIFT	29.1	-1.9	30.2	4.4	23.4	-1.2	24.3	4.9
SpDSIFT	32.9	-0.7	34.1	7.8	25.7	-1.5	26.2	6.4
	Caltech101							
HOG	20.5	-4.4	21.8	1.0	29.3	-5.6	30.5	1.0
DSIFT	21.4	-1.1	21.7	7.5	28.8	-2.3	29.3	9.3
SpDSIFT	26.9	-0.1	27.3	10.9	35.3	-2.5	35.5	12.3
mean	45.7	-1.8	46.2	3.6	41.7	-2.2	41.9	3.5

Table 4.1. Graph quality (GQ) of the 10-NN graph and overall accuracy for L1 and L2 with PCA reduction to 100 dimensions and the gain to our baseline.

When we look at the graph structure based on L2, we observe that few nodes are connected to almost all other nodes. This effect becomes more pronounced the more dimensions are used. For C-PASCAL and SpDSIFT, the node with the maximum number of neighbors in a 10-nearest neighbor structure with L1 distance has 66 neighbors due to the symmetric relations between two nodes. This number increases to 2045 neighbors for L2. Thus these nodes have a large impact on their direct neighbors that becomes more troublesome if these node are selected as labeled node. After PCA, this number of neighbors decreases to 102 for L2 distance.

4.2.2 Increasing the dataset

As a next step, we enrich our graph structure by flipping all images. This leads to a larger and more flexible dataset and should be particularly helpful for global descriptors such as HOG because these descriptors are sensitive to position and orientation of one object. Table 4.2 shows results for L1 and L2 together with the difference to our baseline results. As expected, HOG is improved by 5.3% from 69.2% to 74.5% for ETH80 with L2 and by 1.5% for C-PASCAL with L2.

In contrast, performance of Caltech-101 decreases because of several artifacts in the data. The objects in these classes have a fixed orientation and often they are rotated by the same degree. Figure 4.1 shows three of these biased classes with average precision (AP) before and after flipping. As we can see, AP decreases significantly for these classes since the orientation itself provides an important clue for classification. This information is lost by flipping so that these classes are confused with other classes rotated by the same degree in the other direction. Finally, for the local descriptors no significant decrease or increase is observable.

	GQ				LP			
descriptor	L1+flip	gain	L2+flip	gain	L1+flip	gain	L2+flip	gain
	ETH80							
HOG	89.6	3.9	88.5	4.2	73.5	5.5	74.5	5.3
DSIFT	85.6	-	83.7	-	73.9	-0.3	71.4	-0.6
SpDSIFT	84.8	-	83.1	-	71.1	-1.0	69.3	-0.7
	C-PASCAL							
HOG	35.5	3.5	31.6	3.1	26.8	1.6	24.1	1.5
DSIFT	31.1	-	26.1	0.3	24.4	-0.2	19.7	0.2
SpDSIFT	33.6	-	26.3	-	27.7	0.4	19.4	-0.3
	Caltech101							
HOG	27.8	2.9	22.9	2.2	33.6	-1.2	28.2	-1.3
DSIFT	22.5	-	14.1	-	31.0	-0.1	20.0	-0.1
SpDSIFT	27.2	-	16.3	-	38.1	0.3	22.8	-0.3
mean	48.6	1.1	43.6	1.1	44.5	0.6	38.8	0.4

Table 4.2. Graph quality and overall accuracy for L1 and L2 with flipped images and the gain to our baseline results in Table 3.1.

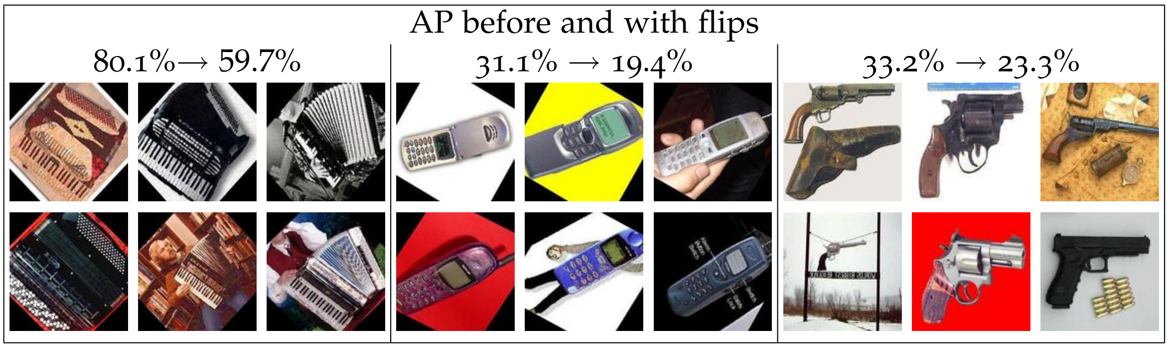


Figure 4.1. Example classes with biased objects in Caltech101 with average precision (AP) before and after flipping for TPLBP.

4.2.3 Combination

In this subsection, we combine multiple features. We average (i) all 8 kernels given by the different image descriptors, and (ii) three best kernels for each dataset. In Table 4.3 we see performance for both strategies. It stands out that there is a consistent improvement for L1 as well as L2 distance. For Caltech101, we improve a simple HOG descriptor with L1 from 34.8% to 42.2% by combining this descriptor with DSIFT and SpDSIFT. Combination of all descriptors leads to less increase of performance in comparison to the combination of the three best descriptors. This can be explained by the simple averaging of kernels. There is no weighting of the kernels so that weak kernels decrease the performance of strong kernels.

	GQ				LP			
descr.	L1+com	gain	L2+com	gain	L1+com	gain	L2+com	gain
	ETH80							
all	88.6	0.3	87.0	0.6	75.8	0.1	72.6	0.6
3 best	89.4	1.1	87.2	0.8	77.5	1.7	74.3	2.3
	C-PASCAL							
all	38.0	4.4	31.1	1.3	29.0	1.8	23.0	-
3 best	36.4	2.8	32.1	2.3	28.9	1.7	24.9	1.9
	Caltech101							
all	28.7	1.7	21.4	0.6	39.3	1.5	30.1	0.6
3 best	30.2	3.1	24.3	3.5	42.2	4.4	34.4	5.0

Table 4.3. Graph quality and overall accuracy for L1 and L2 with combination and the gain to our baseline. We combine all descriptors and three best descriptors for each datasets by averaging the kernel matrices.

	ETH80		C-PASCAL		Caltech101	
strategy	L1	L2	L1	L2	L1	L2
baseline	68.0	69.2	25.2	22.6	34.8	29.5
+PCA	70.1	-	-	23.7	-	30.5
+Flips	75.9	74.5	26.8	24.7	-	-
+Combination	80.3	78.5	30.5	28.5	43.7	43.3
improvement	12.3	9.3	5.3	5.9	8.9	13.9

Table 4.4. Summary of all strategies for HOG

4.2.4 Summary

In this section, we explored several unsupervised methods to improve our graph structure for label propagation. PCA (Section 4.2.1) is used to reduce the dimensionality of our descriptors. This is important for L2 distance because of its sensitivity to noise. Higher dimensionality leads to a larger difference between L1 and L2 and a larger improvement of L2 after PCA. Flipping of images (Section 4.2.2) enriches and improves our graph structure for global descriptors like HOG because these descriptors are strongly dependent on the position and orientation of an object. Finally, combination of multiple descriptors (Section 4.2.3) always helps as long as not too many weak descriptors are involved.

To summarize our insight from this section, we successively apply all three strategies to HOG (see Table 4.4). As we can see in the last row, we improve performance on all datasets and all distance measures from 5.3% to 13.9%. These final results are also better than each improvement alone. For example, ETH80 with L1 is improved by 12.3% from 68.0% to 80.3% while each strategy alone brings only improvements from 1.5% (combination) to 5.5% (flipping). These results illustrate the importance of improving graph structure that can result, as shown in the previous section, to more prominent improvements than using different SSL algorithms.

4.3 SUPERVISED IMPROVEMENTS

We now look at several supervised techniques to improve our neighborhood structure. All these methods make use of information given by the labeled data. In contrast to the unsupervised approaches, this additional information will better guide the transformation process of the data space. But on the other side, this can also lead to a loss of generalization due to the small amount of labels we use. To keep track of this issue, we split the previous introduced graph quality (GQ) from Equation (3.19) into GQ of labeled (GQ_L) and unlabeled data (GQ_U). This allows us to notice a possible tendency to over-fitting on the labeled part of the data.

First, we look in Section 4.3.1 at linear discriminant analysis (LDA) the counterpart to PCA for direct analogy to Section 4.2.1. Often this method is preferred in comparison to PCA because it also considers the class distribution. But Martinez and Kak (2001) show that PCA usually outperforms LDA when only few training examples are available. We will analyze this aspect since SSL is typically close to this setting.

In Section 4.3.2, we propose a metric learning scheme based on the decision values of a SVM classifier. This scheme successfully combines our generative classifier with a discriminative classifier and benefits from both paradigm. Rohban and Rabiee (2012) also enhance the graph construction with SVM. But in comparison to our approach, they use a pre-computed k-NN graph structure as a baseline to confirm or to reject existing edges using SVM results. Thereby, the new edge set is only a subset of the original edge set leading to less connected graph structures. Instead, our approach comes with more flexibility by allowing also new edges.

Another and well established way to learn a representation better suited for the task at hand is metric learning. These methods transform the data into a more discriminative space such that intra-class examples are closer together and inter-class examples are far away. The published methods essentially differ in the parameterization of the learned metric (including regularizers and constraints) and optimization procedures. In Section 4.3.3, we apply (Davis *et al.*, 2007) to our data that learns a Mahalanobis distance based on pairwise constraints because this method shows state-of-the-art performance on Caltech 101 (Kulis *et al.*, 2009). Besides the success, it is scalable to large problems also in high dimensional spaces.

Similar to Section 4.2, we analyze each method separately and show the gain to our baseline results of label propagation (LP) and graph quality (GQ) of the 10-NN graph given in Table 3.1. As mentioned before, we also document GQ_L, GQ_U, and the difference $GQ_U - GQ_L$ between both measures. High negative difference values $GQ_U - GQ_L$ indicate strong over-fitting. In the first two subsections, we show only results for L2 as we do not observe any improvement for L1. Finally, we combine all supervised strategies in Section 4.3.4 to have a direct comparison to the unsupervised improvements.

4.3.1 Dimensionality Reduction with LDA

Linear discriminant analysis (LDA) uses additional information provided by the labels of the C classes. Eigenvalues are computed for $S_w^{-1}S_b$ with a within-class scatter matrix S_w and a between-class scatter matrix S_b. Finally, this new space of eigenvectors $T : \mathbb{R}^{l \times d} \mapsto \mathbb{R}^{d \times C-1}$ is used to transform our original data space $X \in \mathbb{R}^{n \times d}$ such that $\tilde{X} : \mathbb{R}^{n \times d} \mapsto \mathbb{R}^{n \times C-1}$ with

$$\tilde{X} = XT. \tag{4.1}$$

In contrast to PCA, where we have usually $\leq d$ non-zero eigenvalues, we get at most $C - 1$ non-zero eigenvalues because $\text{rank}(S_b) \leq C - 1$. Therefore, our representation is limited by the number of classes.

Table 4.5 shows results of 10-NN quality and LP with LDA. As mentioned before, we show only results for L2 as there is no improvement for L1 observable. From this table we make three observations. (i) L2 benefits when LDA is used on datasets with many classes. For Caltech101, we get an improvement for LP from 23.2% to 26.8% for SpDSIFT. (ii) The strong dependency between dimensions and number of classes hurts for datasets with few classes. When we look again at the LP results for SpDSIFT, we observe a decrease of -17.7% for ETH80 with 8 classes. The decrease for C-PASCAL with 20 classes is only -3.7%. And finally, Caltech101 with 102 classes ($d = 101$) is improved by 3.6%. Therefore, a minimum of classes (dimensions) are needed to get a discriminative data space. Another reason why PCA consistently outperforms LDA is given in Martinez and Kak (2001) where the authors show that few training labels per class may lead to a decreased performance. (iii) Fewer dimensions of the original feature space lead to more overfitting (column 4-6) and thus to worse performance. For Caltech101, the difference between GQ_U and GQ_L is for HOG ($d = 576$) -22.0%, for DSIFT ($d = 1000$) -17.9%, and for SpDSIFT ($d = 16000$) -3.8%. Similar observations hold for the other datasets as well. One explanation might be that the high-dimensional space conveys more discriminative information about the labeled data distribution.

4.3.2 SVM-based Graph Construction

In this subsection, we propose a new approach that combines a discriminative classifier based on SVM with the generative label propagation scheme to benefit from both strategies. First, we learn SVM classifiers for each class to each other class (one-vs-one classification) for the labeled data, i.e.,

$$\max_{\alpha} \sum_{i=1}^{d} \alpha_i - \frac{1}{2} \sum_{j=1}^{d} \alpha_i \alpha_j y_i y_j \phi(x_i, x_j)$$

$$\text{subject to } 0 \leq \alpha_i \leq \xi \text{ and } \sum_{i=1}^{d} \alpha_i y_i = 0$$

	GQ		overfitting			LDA+LP	
desc	L2+LDA	gain	GQ_L	GQ_U	diff	L2+LDA	gain
			ETH80				
HOG	69.8	-14.6	72.0	69.8	-2.2	62.9	-6.3
DSIFT	68.9	-14.7	69.9	68.9	-1.0	56.9	-15.1
SpDSIFT	66.5	-16.8	66.8	66.5	-0.3	52.3	-17.7
			C-PASCAL				
HOG	23.3	-5.2	26.6	23.2	-3.4	20.5	-2.1
DSIFT	23.9	-1.9	24.6	23.9	-0.7	19.1	-0.4
SpDSIFT	23.0	-3.3	22.3	23.0	0.7	16.0	-3.7
			Caltech101				
HOG	15.2	-5.6	33.5	11.5	-22.0	17.7	-11.8
DSIFT	18.2	+4.1	33.1	15.2	-17.9	23.0	+3.0
SpDSIFT	19.7	+3.3	22.8	19.1	-3.8	26.8	+3.6
mean	36.5	-6.1	41.3	35.7	-5.6	32.8	-5.6

Table 4.5. Overall accuracy and graph quality of a 10-NN graph for L1 and L2 with LDA and the gain to our baseline results in Table 3.1, and the difference (diff) $NN_U - NN_L$.

with a RBF kernel $\phi(x,y) = exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right)$, a positive constant $\xi > 0$, and $1 \leq i,j \leq l$. Then, we extract the corresponding decision values for each classifier

$$f_{i,j}(x) = \sum_{k=1}^{d} \alpha_k y_k \phi(x_k, x) + b \tag{4.2}$$

with $1 \leq i,j \leq c$, and build our final data space $\tilde{X} : \mathbb{R}^{n\times d} \mapsto \mathbb{R}^{n\times\binom{c}{2}}$ by concatenating these decision vectors

$$\tilde{X} = \left\{x | x = \{f_{1,2}(x), ..., f_{i,j}(x), ..., f_{c-1,c}(x)\}\right\}. \tag{4.3}$$

Another possibility would be to use one-vs-all classification, i.e., learning SVM classifiers for each class with respect to all other classes. But we discard this option due to the lower performance that can be explained by the low dimensionality of C instead of $\binom{C}{2}$.

In this work, we explore two different metric spaces: (i) *SVM→LP* that uses $\tilde{X}$ to build a new graph structure

$$\tilde{W}_{ij} = \tilde{P}_{ij} \exp\left(\frac{-d(\tilde{x}_i, \tilde{x}_j)}{2\sigma^2}\right), \tag{4.4}$$

and (ii) SVM+LP that combines both graph structures built from X and $\tilde{X}$

$$\hat{W}_{ij} = W_{ij} + \tilde{W}_{ij}. \tag{4.5}$$

(i) *SVM→LP*. Results are shown in Table 4.6. In comparison to LDA, we observe a significantly improvement for nearest neighbor quality of labeled data (GQ_L).

SpDSIFT of C-PASCAL increases from 26.3% (Table 3.1) to 60.4%. On the other side, there is almost no improvement for the unlabeled data (GQ_U). This implies that this strategy again suffers from overfitting.

	GQ		overfitting			$SVM \rightarrow LP$	
desc	L2+SVM	gain	GQ_L	GQ_U	diff	L2	gain
ETH80							
HOG	76.5	-7.9	92.2	76.3	-15.9	68.8	-0.4
DSIFT	74.8	-8.9	94.4	74.6	-19.8	66.1	-5.9
SpDSIFT	74.1	-9.1	95.9	73.9	-22.0	66.3	-3.7
C-PASCAL							
HOG	28.0	-0.5	34.4	27.8	-6.6	23.8	+1.2
DSIFT	24.0	-1.8	49.1	23.4	-25.7	19.5	+0.1
SpDSIFT	24.9	-1.4	60.4	24.1	-36.3	21.4	+1.7
Caltech101							
HOG	22.7	+1.9	27.8	21.6	-6.1	30.9	+1.4
DSIFT	13.2	-0.9	22.5	11.4	-11.1	18.0	-2.0
SpDSIFT	19.2	+2.8	39.3	15.2	-24.1	24.9	+1.8
mean	39.7	-2.9	57.3	37.7	-18.6	37.7	-0.6

Table 4.6. Overall accuracy and graph quality (GQ) for L2 with SVM score and the gain to our baseline results in Table 3.1, and the difference (diff) $GQ_U - GQ_L$.

(ii) *SVM+LP*. Table 4.7 shows the results for the combined version. Here, we can see a consistent improvement of label propagation for all datasets and all descriptors. HOG for C-PASCAL is improved by 2.2% from 22.6% (Table 3.1) to 24.8%. All results of LP are better than using decision values alone ($SVM \rightarrow LP$). Overfitting is decreased due to the averaging with the original graph structure. Finally, this method always enhances our graph structure when using it in the combined version.

4.3.3 Information theoretic metric learning (ITML)

In this subsection, we apply the information theoretic metric learning (ITML) proposed by Davis *et al.* (2007) to our data space to get a more distinctive one. ITML optimizes the Mahalanobis distance between each point pair $x_i, x_j \in \mathbb{R}^m$

$$d_A(x_i, x_j) = (x_i - x_j)^T A (x_i - x_j) \tag{4.6}$$

Eq. (1) reduces to a simple euclidean distance if $A = I$. To learn a Mahalanobis matrix A, the algorithm minimizes the logdet divergence D_{ld} between a matrix A and an initial matrix A_0 with respect to pairwise similarity and dissimilarity constraints that are extracted from the labeled data, i.e.,

	GQ		overfitting			*SVM+LP*	
desc	L2+SVM	gain	GQ_L	GQ_U	diff	L2	gain
ETH80							
HOG	80.9	-3.5	91.7	80.7	-11.0	70.4	+1.2
DSIFT	78.4	-5.3	90.8	78.3	-12.5	72.2	+0.2
SpDSIFT	78.1	-5.2	91.2	77.9	-13.3	70.1	0.1
C-PASCAL[1]							
HOG	29.3	0.8	33.3	29.2	-4.1	24.8	+2.2
DSIFT	25.3	-0.5	42.1	24.9	-17.1	19.9	+0.4
SpDSIFT	26.0	-0.3	55.1	25.4	-29.7	21.7	+1.9
Caltech101							
HOG	23.6	+2.9	27.8	21.6	-6.1	32.7	+3.2
DSIFT	14.3	+0.2	21.9	12.9	-9.0	20.7	+0.7
SpDSIFT	20.3	+3.9	38.1	16.8	-21.3	26.7	+3.6
mean	41.8	-0.8	54.7	40.8	-13.8	39.9	+1.5

Table 4.7. Overall accuracy for L1 and L2 with SVM score combined with original data and the gain to our baseline results in Table 3.1. Graph quality of the unlabeled data GQ_U and the difference to the labeled data $GQ_U - GQ_L$.

$$\begin{aligned} \min \quad & D_{ld}(A, A_0) \\ s.t. \quad & d_A(x_i, x_j) \leq b_u \quad (i,j) \in \mathcal{S} \\ & d_A(x_i, x_j) \geq b_l \quad (i,j) \in \mathcal{D} \end{aligned} \tag{4.7}$$

b_u and b_l are upper and lower bounds of similarity and dissimilarity constraints. $\mathcal{S}$ and $\mathcal{D}$ are sets of similarity and dissimilarity constraints based on the labeled data. To make this optimization feasible, a slack parameter γ is introduced to control the trade-off between satisfying the constraints and minimizing $D_{ld}(A, A_0)$. The larger γ the more constraints are ignored. The optimization is done by repeatedly Bregman projections of a single constraint per iteration.

One benefit of this optimization scheme is the efficient kernelization with $K = X^T AX$. A proof can be found in Davis *et al.* (2007). The kernel version has several advantages. The run time depends only on the number of constraints $\binom{l}{2}$ extracted from l labeled examples and not on the number of dimensions m that is critical particularly in a high dimensional space. We can subsample the number of constraints such that $l \ll m$ which reduces the costs from $O(m^2)$ to $O(l^2)$. Finally, we can easily compute the most violated constraint per iteration since only matrix additions ($K_{ii} + K_{jj} - 2K_{ij}$) are required and no complex multiplications as in Equation (4.6). This leads to faster convergence.

Finally, we use the transformed data space

$$\tilde{X} = X^T A \tag{4.8}$$

for the graph construction. We show results of this method in Ebert *et al.* (2011) but only for one descriptor. In the following, we review and extend this publication by evaluating different image representations as well as different distance measures. Additionally, we give a more detailed analysis of the nearest neighbor quality.

	GQ		overfitting			ITML+LP	
desc	L1+ITML	gain	GQ_L	GQ_U	diff	L1	gain
			ETH80				
HOG	85.3	-0.5	92.4	85.2	-7.2	69.1	+1.0
DSIFT	85.9	+0.2	95.5	85.8	-9.7	**75.2**	+0.9
SpDSIFT	85.3	+0.5	92.6	85.2	-7.4	74.3	+2.2
			C-PASCAL[1]				
HOG	32.0	0.0	26.9	32.2	5.3	25.2	0.0
DSIFT	31.2	+0.2	36.2	31.1	-5.1	26.0	+1.3
SpDSIFT	33.8	+0.3	38.0	33.8	-4.3	**28.2**	+0.9
			Caltech101				
HOG	29.4	+4.5	49.4	25.5	-24.0	35.2	+0.4
DSIFT	28.9	+6.4	46.6	25.5	-21.2	34.3	+3.2
SpDSIFT	34.4	+7.4	52.4	30.9	-21.5	41.5	+3.7
mean	49.6	2.1	58.9	48.3	-10.6	45.4	1.5

Table 4.8. Overall accuracy for L1 with ITML and the gain to our baseline results in Table 3.1. Graph quality (GQ) and the difference $GQ_L - GQ_U$.

We document results for L1 (Table 4.8) as well as for L2 (Table 4.9) as we observe improvements for both measures. Obviously, ITML improves consistently the performance for all datasets and all distance measures. For L1 on Caltech101 with SpDSIFT, we improve LP accuracy by 3.7% from 37.8% to 41.5% with ITML and for L2 by 6.6% from 23.2% to 29.8% on the same dataset. In contrast, overfitting is in average moderate in comparison to the previous proposed method. The difference between GQ_L and GQ_U is for L2 with ITML -6.5% while for L2 with *SVM→LP* (Table 4.6) the difference is -18.6% and with *SVM+LP* (Table 4.7) -13.8%. This can be explained by the slack variable γ. With this parameter, we are able to control how many of the constraints have to be fulfilled. The smaller γ (i.e. $\ll 10^{-3}$) the more constraints are satisfied leading to a rather specific metric that fits only the labeled data. In our experiments, we set λ to 0.1 as it shows empirically the best trade-off between our optimization objective and generalization performance. Finally, this method leads to the highest improvement with respect to the supervised methods. LP with L2 is improved in average by 3.1% that is better than *SVM+LP* with 1.5% improvement and LDA with in average worse performance of -5.6%.

4.3.4 Summary

In this section, we explored different supervised graph improvements. Linear discriminant analysis (LDA) as a counterpart to PCA and two different metric

	GQ		overfitting			ITML+LP	
desc	L2+ITML	gain	GQ_L	GQ_U	diff	L2	gain
			ETH80				
HOG	84.5	+0.2	85.2	84.5	-0.7	71.3	+2.1
DSIFT	84.1	+0.4	84.9	84.1	-0.8	**73.3**	+1.4
SpDSIFT	83.9	+0.7	93.3	83.8	-9.5	72.6	+2.6
			C-PASCAL[1]				
HOG	30.4	+1.9	25.5	30.5	5.0	**24.2**	+1.6
DSIFT	26.4	+0.5	42.4	26.0	-16.4	22.2	+2.8
SpDSIFT	28.2	+1.9	39.0	27.9	-11.0	22.9	+3.1
			Caltech101				
HOG	23.7	+2.9	23.2	23.8	0.6	33.6	+4.1
DSIFT	16.9	+2.8	16.2	17.1	0.9	23.7	+3.6
SpDSIFT	25.9	+9.6	48.3	21.5	-26.8	29.8	+6.6
mean	44.9	2.3	50.9	44.3	-6.5	41.5	3.1

Table 4.9. Overall accuracy for L2 with ITML and the gain to our baseline results in Table 3.1. Graph quality (GQ) and the difference $GQ_L - GQ_U$.

	ETH80		C-PASCAL		Caltech101	
strategy	L1	L2	L1	L2	L1	L2
baseline	68.0	69.2	25.2	22.6	34.8	29.5
+LDA	-	-	-	-	-	-
+SVM	69.8	70.4	-	24.8	-	32.7
+ITML	70.1	71.3	-	-	35.2	33.6
improvement	2.1	2.1	-	2.2	0.4	4.1

Table 4.10. Summary of all strategies

transformations. The first uses the SVM decision values to build a new metric space and the second method optimizes the Mahalanobis distance based on pairwise constraints. Equivalent to Table 4.4 in the previous section, we incrementally add each supervised method for HOG as long as it leads to an improvement. In Table 4.10, we see a consistent improvement for all datasets and almost all distance measures. For Caltech101 with L2, we increase our performance from 29.5% to 33.6% with SVM-based metric extraction and ITML.

However, the benefit from these supervised approaches is not as large as for the unsupervised methods where we increase Caltech101 up to 8.9% with L2 to 43.3%. The main reason is that we have, in a typical semi-supervised setting, only few labels and a large amount of unlabeled data. This leads nearly inevitably to over-fitting the labeled data while unlabeled data benefit only marginally from these changes.

4.4 SEMI-SUPERVISED IMPROVEMENT WITH IMLP

In the previous section, we show that supervised methods often overfit labeled data. Based on this observation, we address the lack of generalization by incorporating few predictions from unlabeled data. We propose an iterative procedure with interleaved metric learning and label propagation (IMLP). This improves incrementally the nearest neighbor precision with the condition that the manifold structure given by the unlabeled data is taken into account. The resulting procedure is as follows:

1. metric learning using the current set of labeled data $L^{(t)}$ to obtain kernel K
2. label propagation with kernel K to obtain predictions $\hat{Y}$ of unlabeled data
3. choose $d = d + n_s$ data points x_i such that $|\tilde{y}_1| \geq ... \geq |\tilde{y}_i| \geq |\tilde{y}_{i+1}| \geq ... \geq |\tilde{y}_d|$ with $\tilde{Y} = \max_{1 \leq j \leq c} Y^*_j$ and $l < i <= u$
4. add these data points to the labeled data to get an extended set of labeled data, i.e., $L^{(t+1)} \leftarrow L^{(t)} \cup \{(x_1, \tilde{y}_1), ...(x_m, \tilde{y}_m)\}$
5. go to step 1.

We fix $n_s = 100$. We apply this approach to both metric learning methods, i.e., SVM-based extraction *SVM+LP* (Section 4.3.2) and ITML (Section 4.3.3). For *SVM+LP*, we document results for HOG with L2 distance in Table 4.11 because this descriptor benefits most in Section 4.3.2. Similar to the previous section, we show NN quality for all data and split into labeled and unlabeled data and LP results. In the first line for each dataset, we show our baseline results from Table 3.1. The improvement from *SVM+LP* is in the second line. The third line contains our results when we apply IMLP. In fact, over-fitting of labeled data is less present when we add additional predictions of unlabeled data. For ETH80, we decrease the difference between GQ_L and GQ_U from 11.0% to 4.4% while we increase the performance from 70.4% to 72.0%. Similar observations consistently hold for all datasets.

Table 4.12 shows results for IMLP in combination with ITML. For this table, we use our best combined kernel from Section 4.2.3 for ETH80 and C-PASCAL. For Caltech101, we use the same kernel as in Jain and Kapoor (2009) (obtained from the authors) which uses an average of four kernels: two kernels based on the geometric blur descriptor, Pyramid Match Kernel (PMK) and the Spatial PMK using SIFT features. This kernel shows similar performance to our best combination kernel from Section 4.2.3 but benefits more from ITML. We improve our results of LP+ITML to 58.7% that goes beyond existing best known numbers of 56.9% by Boiman *et al.* (2008) and 54.2% by Gehler and Nowozin (2009). Finally, we also get a consistent improvement for ETH80 and C-PASCAL.

To summarize this section, we show how we can further increase our quality of graph structure and thus the performance of label propagation by considering unlabeled data for supervised improvements. Although our proposed method is to

	GQ		difference			*SVM+LP*	
desc	10-NN	gain	GQ_L	GQ_U	diff	LP	gain
			ETH80				
baseline	84.4		84.4	84.4	0.0	69.2	
SVM+LP	80.9	-3.5	91.7	80.7	11.0	70.4	+1.2
IMLP	82.4	-2.0	86.7	82.3	4.4	72.0	+2.9
			C-PASCAL				
baseline	28.5		24.6	28.5	-4.0	22.6	
SVM+LP	29.3	+0.8	33.3	29.2	4.1	24.8	+2.2
IMLP	29.5	+1.0	32.6	29.4	3.2	24.9	+2.3
			Caltech101				
baseline	20.8		20.1	20.9	-0.9	29.5	
SVM+LP	23.6	+2.9	26.5	23.1	3.4	32.7	+3.2
IMLP	23.7	+3.0	26.4	23.2	3.2	32.9	+3.4

Table 4.11. Overall accuracy and graph quality of 10-NN graph for L2 with SVM-based metric extraction.

	10-GQ		difference			ITML+LP	
desc	GQ	gain	GQ_L	GQ_U	diff	LP	gain
			ETH80				
baseline	89.4		90.4	89.4	1.0	77.5	
ITML	89.7	+0.2	94.2	89.3	4.9	78.0	+0.6
IMLP	89.8	+0.3	89.6	89.8	-0.2	80.7	+3.3
			C-PASCAL				
baseline	38.0		32.1	38.1	-6.0	29.1	
ITML	38.1	+0.1	32.9	38.2	-5.4	29.2	+0.1
IMLP	38.4	+0.4	34.3	38.5	-4.2	29.5	+0.4
			Caltech101				
baseline	33.3		33.5	33.3	0.3	47.2	
ITML	41.2	+7.9	61.0	37.3	23.8	54.5	+7.3
IMLP	43.4	+10.1	61.7	39.3	22.5	58.7	+11.5

Table 4.12. Overall accuracy and graph quality (GQ) of the 10-NN graph for L2 with ITML.

some extent sensitive to the quality of predictions it still indicates that we are able to decrease the over-fitting effect of supervised methods while increasing the overall performance.

4.5 CONCLUSION

In this chapter, we explored different unsupervised and supervised improvements for local neighborhood structure that intervene before graph construction. We show for three different datasets of increasing difficulty a large enhancement of up to 13.9% for unsupervised methods and up to 3.2% for supervised methods. When we combine unsupervised and supervised improvements we get a final improvement up to 17.7%. Table 4.13 summarize all improvements for HOG and all three datasets. We compare unsupervised methods with supervised methods and finally combine all improvements. As we can see, the impact of unsupervised approaches are much higher than supervised ones. This can be explained by the over-fitting of labeled data in the supervised case so that the benefit for unlabeled data is only minor.

	baseline	unsup		sup		unsup+sup	
desc	HOG	LP	gain	LP	gain	LP	gain
ETH80							
L1	68.0	80.3	+12.3	70.1	+2.1	80.9	+12.9
L2	69.2	78.5	+9.3	71.3	+2.1	78.5	+9.3
C-PASCAL							
L1	25.2	30.5	+5.3	25.2	0.0	31.1	+5.9
L2	22.6	28.5	+5.9	24.8	+2.2	29.6	+7.0
Caltech101							
L1	34.8	43.7	+8.9	35.2	+0.4	49.6	+14.8
L2	29.5	43.3	+13.9	32.7	+3.2	47.7	+17.7

Table 4.13. Overall accuracy for L1 and L2 of HOG and the gain to our baseline in Table 3.1.

Overall, this study shows that we are able to challenge state-of-the-art supervised methods without any graph improvements. When we add all proposed methods we improve supervised SVM for Caltech101 from 33.9% to 49.6% with semi-supervised LP. But still current image descriptors are not powerful enough to capture different aspects of one image so that we are far away from image understanding. Another promising direction is to expand recent available data collections with unlabeled data to get more information about different viewpoints, truncations, and occlusions of one object and to properly distinguish from background clutter and other classes. Finally, we show that the missing generalization in supervised approaches can be addressed by incorporating information about unsupervised data. This should be further improved by more robust methods that are less sensitive to the prediction quality.

LABEL IMPROVEMENT BY ACTIVE LEARNING 5

Contents

5.1 Introduction 68
5.2 Sampling Criteria 70
  5.2.1 Previous exploitation criteria 71
  5.2.2 Previous exploration criteria 72
  5.2.3 Graph density criteria for exploration 73
  5.2.4 Experiments 75
5.3 Time-varying Active Learning 76
  5.3.1 Method 76
  5.3.2 Analyzing the trade-off on synthetic datasets 77
  5.3.3 Experiments on real data 79
5.4 RALF - A Reinforced Active Learning Formulation 81
  5.4.1 Switching feedback-driven between two criteria 81
  5.4.2 Reinforced active learning formulation (RALF) 82
  5.4.3 Experiments 84
5.5 Conclusion 86

ALTHOUGH graph structure is essential to propagate reliably labels from labeled to unlabeled data, the representativeness of labels has also a considerable influence on the final results. This becomes more important in a small sample regime as it is the case in semi-supervised learning. Even if the distribution is well distinguishable as in Figure 5.1 (right), it will be impossible to classify the front view of a car if there are only labels for the side view of a car. But also for distributions such as in Figure 5.1 (left) is the label important. Ideally one would have one label from the middle of each mixture otherwise it can happen that one class is overwritten by another class. See also Section 1.2 for a more detailed discussion and visualization of these problems.

The goal of this chapter is to increase the performance of semi-supervised learning with more representative labels by integrating active learning. First, we review state-of-the-art work in this topic and point out difficulties of these approaches and in particular the exploration-exploitation-dilemma in Section 5.1. Then, we introduce in Section 5.2 most common sampling criteria from pure exploration-driven to pure exploitation-driven. We discuss advantages and disadvantages of those. Furthermore, we propose our new sampling criteria *graph density* that uses the underlying graph structure to find more representative samples from dense regions in the data distribution. After that, we argue in Section 5.3 that a time-varying combination of exploration and exploitation criteria is always useful. But this trade-off is highly dependent on the dataset distribution. Thus, each dataset needs either a specific and time-consuming fine-tuning to find this trade-off or prior knowledge about the data distribution beforehand that is usually not available. Therefore, we propose in Section 5.4 our new meta active learning framework based on the Markov decision process that allows full flexibility according to the number of combined criteria as well as the combination strategy and learns automatically a dataset-specific trade-off (Ebert *et al.*, 2012b). Finally, we close this chapter with a conclusion in Section 5.5.

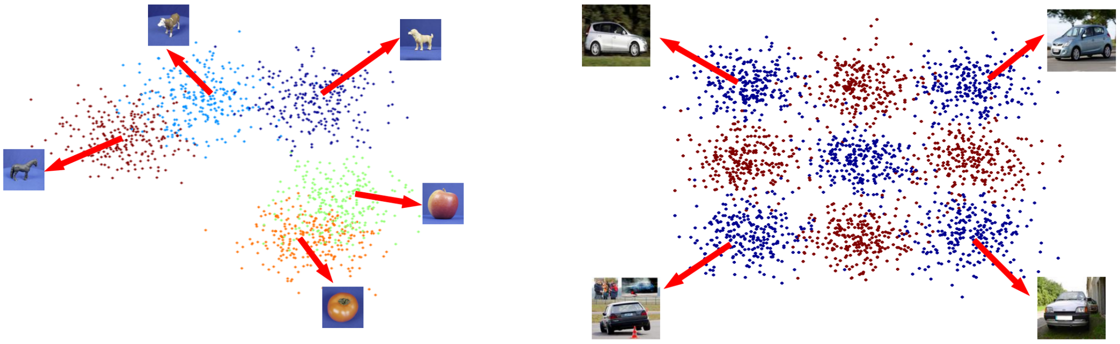
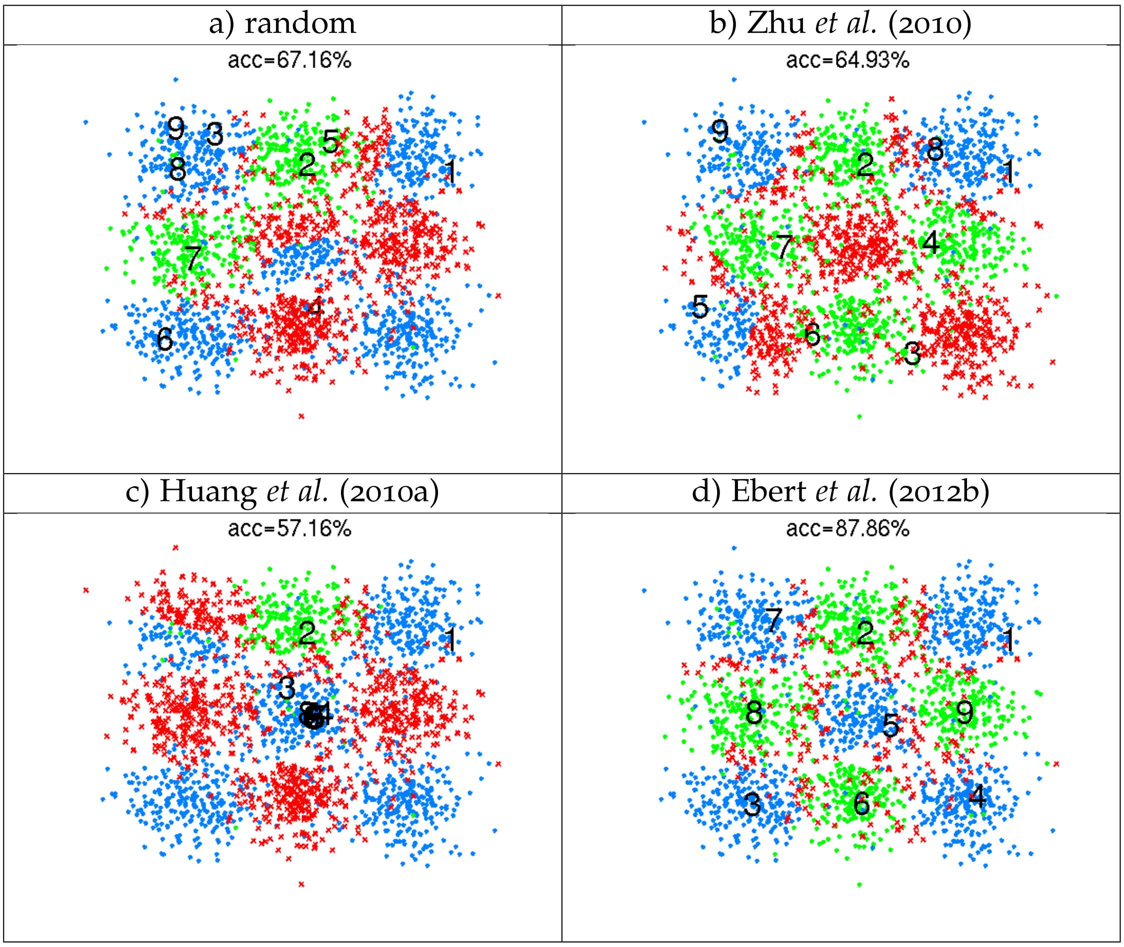


Figure 5.1. Depending on the dataset and their completeness in terms of viewpoints and appearance, the data distribution can result in a more compact distribution where each class build one group (left) or in a more fragmented distribution where a class consists of a separate groups (right). Note that the data points are not produced from real image data.

5.1 INTRODUCTION

Active learning is a promising research direction to reduce the total labeling effort and to get more representative labels. Most common in this area is pool-based active learning (Settles and Craven, 2008) that considers all unlabeled data as a pool from which most informative examples given a certain criteria are selected for labeling. Usually, a single criteria is used. But this limits significantly the performance of active learning that is also known as the exploitation-exploration-dilemma. A pure exploitative (uncertainty-based) criteria leads often to a serious sampling bias as it only focuses on regions that are difficult to learn. This problem is more prominent in a multi-class scenario when some classes are more often requested while other classes are completely overlooked, or on challenging datasets when one class consists of many spatially separated dense regions (e.g., front and side view of a car). In contrast, a pure explorative (density-based) criteria covers the entire data space but needs too many iterations before a good decision boundary is found.

Consequently, methods have been proposed that address this problem by combining different criteria. Methods range from randomly switching between exploration and exploitation (Osugi *et al.*, 2005) over constructing a bound to switch between both strategies (Donmez and Carbonell, 2007; Krause and Guestrin, 2007) to combi-



a) random

b) Zhu *et al.* (2010)

c) Huang *et al.* (2010a)

d) Ebert *et al.* (2012b)

Figure 5.2. Starting with 1 label per class (green and blue dots), we request iterative labels with the different sampling strategies: a) randomly sampled and b-d) three different sampling strategies that combine exploration and exploitation. False classified examples after label propagation are marked with red dots. The numbers beside the selected points indicate the order in which they were selected.

nations with a fixed and time independent parameter (Cebron and Berthold, 2009). More recently, we have seen a trend towards dynamically balancing this trade-off by estimating the gain for the next iteration (Bondu *et al.*, 2010; Wang *et al.*, 2011; Huang *et al.*, 2010a). Finally Zhu *et al.* (2010) select a subset of the least certain samples and among those the most representative one.

The main problem of these hybrid methods is that their combination is often not well balanced resulting in a strong bias either on the exploitative side (Figure 5.2b) or on the explorative side (Figure 5.2c). Figure 5.2 visualize this bias for the artificial dataset *checkerboard*. This dataset consists of two classes (green and blue dots) that are split into several small mixtures arranged as a checkerboard. We start with 1 label per class (marked as 1 and 2) and request 7 additional labels with several methods. Red dots indicate the false classified examples after 9 iterations. For comparison, we show also the random sampling in the first box (Figure 5.2a). While the method of Zhu *et al.* (2010) shown in Figure 5.2b) has a bias to the exploitative side, Huang *et al.* (2010a) focus only on one dense region as there is no down weighting of this region as can be seen in Figure 5.2c). Ideally we want for each of these mixtures one label after 9 iterations similar to the result in Figure 5.2d) that we get with our method. For this artificial dataset, the accuracy for both previous active learning methods is with 64.9% (Figure 5.2b) and 57.2% (Figure 5.2c) lower than for random sampling with 67.2%.

Another problem is that these approaches combine usually only two criteria with a fixed trade-off parameter (Cebron and Berthold, 2009; Bondu *et al.*, 2010) instead of a time-varying trade-off. Also, for many of those methods it is not obvious how to generalize them to multi-class scenarios. Last but not least, we will show empirically in Section 5.3, there is no single, pre-determined combination scheme that does work well across various datasets with differing properties such as varying number of classes, different training set size, and data clustering structure.

Therefore, we propose a framework that addresses these problems by considering active learning as a Markov decision process. This gives us the full flexibility to handle more than two sampling criteria, to model all possible trade-offs between exploration and exploitation, and to adapt this trade-off during the labeling process. This adaptation is learned on-line during the labeling process. We use the feedback from the classifier to guide the next label request. Since we get feedback in each round from the classifier about the label uncertainty it seems a natural choice to use the entire sequence in a reinforcement learning framework. To the best of our knowledge there is little prior work (Baram *et al.*, 2004; Osugi *et al.*, 2005) that has touched upon this idea of incorporating this type of feedback. Both methods switch randomly between two (Osugi *et al.*, 2005) or three (Baram *et al.*, 2004) criteria but the probability is adapted by reinforcement learning. Baram *et al.* (2004) formulate this guiding routine as a multi-armed bandit problem that comes with much more parameters in comparison to our model. Osugi *et al.* (2005) is easier to implement and has less parameters but it allows only a combination of two criteria and we observe a strong bias to exploitation (similar to Figure 5.2b). In contrast, the label sequence in our model is well balanced so that we get one label for each mixture after 7 iterations resulting in the highest accuracy with 87.9% (Figure 5.2d).

5.2 SAMPLING CRITERIA

A central part of active learning is the sampling criteria. These criteria rank unlabeled data and request a label for the sample with highest rank. In this section, we discuss two common exploitation (Section 5.2.1) and two exploration criteria (Section 5.2.2). We introduce our new exploration-driven criteria *graph density* in Section 5.2.3. We explain the intuition behind this criteria and show which problems are tackled. Finally, we analyze advantages and disadvantages of these criteria in our experimental section in Section 5.2.4. This section serves only as a preparation and a prelude to the next section 5.3 where we show the importance of combining both exploration and exploitation criteria.

5.2.1 Previous exploitation criteria

Given $n = l + u$ data points with l labeled examples $L = \{(x_1, y_1), ..., (x_l, y_l)\}$ and u unlabeled examples $U = \{x_{l+1}, ..., x_n\}$ with $x_i \in \mathbb{R}^d$. We assume C classes and denote $y \in \mathcal{L} = \{1, ..., C\}$ the labels. We run a classification algorithm to get prediction values $\hat{y}_{ij} \in \mathbb{R}$ for all unlabeled data $1 \leq i \leq u$ and each class $1 \leq j \leq C$. These prediction values are used to find most uncertain samples.

Entropy (Ent) over the class posterior is the most common criterion for exploitation (Baram *et al.*, 2004; Joshi *et al.*, 2009; Osugi *et al.*, 2005):

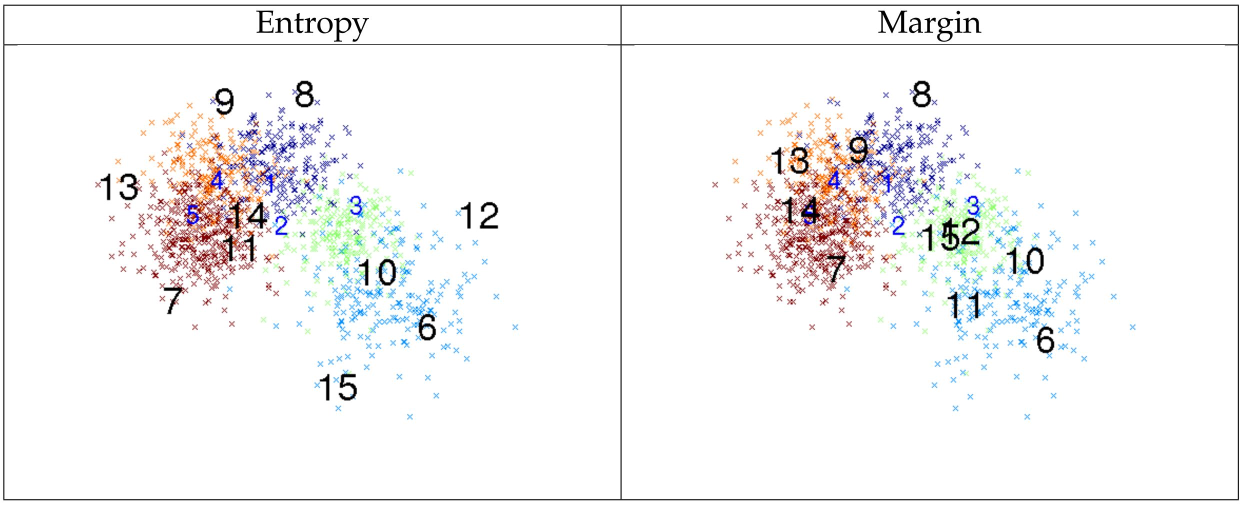
$$Ent(x_i) = -\sum_{j=1}^{c} P(y_{ij}|x_i) \log P(y_{ij}|x_i) \tag{5.1}$$

where $\sum_j P(y_{ij}|x_i) = 1$ are the normalized predictions of a classifier.

Margin (Mar) measures the difference between best versus second best class prediction (Joshi *et al.*, 2009; Settles and Craven, 2008):

$$Mar(x_i) = P(y_{ik_1}|x_i) - P(y_{ik_2}|x_i) \tag{5.2}$$

such that $P(y_{ik_1}|x_i) \geq P(y_{ik_2}|x_i) \geq ... \geq P(y_{ik_c}|x_i)$. In each iteration, label $x^* = \text{argmin}_{x_i \in U} Mar(x_i)$ is requested.


Entropy

Margin

Figure 5.3. Starting with 1 label per class (green and blue dots), we request iterative labels with the different sampling strategies: a) entropy request often samples far away from dense regions; and b) margin focus more on the decision boundaries. The numbers beside the selected points indicate the order in which they were selected.

Entropy captures the overall uncertainty of one example belonging to a class. In practice, it tends to sample uninformative ones (Joshi *et al.*, 2009) that means samples far away from the dense regions as the prediction score y_{ij} for these examples is close to zero for each class j. In contrast, *margin* focus more on the decision boundaries between two overlapping classes so that the samples are more representative. Figure 5.3 visualize these effects for an artificial dataset with 5 mixtures. We start with one label per class (denoted with blue numbers 1-5) and draw successively one label per iteration. The numbers indicate the order in which they are selected. While *Entropy* tends to request samples far away from dense regions, e.g. 7-9, 12, 13, 15, samples from *Margin* are more from highly overlapping regions.

5.2.2 Previous exploration criteria

Criteria for exploration are mostly used in combination with an exploitation criteria. They usually consider only the overall data distribution and do not get feedback from the classifier during the labeling process. These criteria are computed once at the beginning while exploitation criteria are recomputed after each label. Although this feedback for exploitation is limited as it only considers the current label situation and not the entire sequence it leads often to a slightly better performance when using density-based criteria alone.

Node potential (Nod) finds dense regions based on a Gaussian weighting function (Cebron and Berthold, 2009):

$$Nod(x_i) = \sum_{j=1}^{n} e^{-\alpha d^2(x_i,x_j)} \tag{5.3}$$

with $\alpha = \frac{4}{r_a^2}$, r_a the inverse neighborhood radius of a node, and Euclidean distance d. After choosing a sample x_i the neighborhood of this image is downweighted with $Nod(x_j) = Nod(x_j) - Nod(x_i)e^{-\beta d(x_i,x_j)^2}$, $\beta = \frac{4}{r_b^2}$. In principle this measure requests samples from dense regions. However, adjusting the parameters can be difficult. When the neighborhood is too small multiple samples are drawn from the same dense region as can be seen in Figure 5.4 (right), whereas when the neighborhood is too large the approach tends to sample outliers shown in Figure 5.4 (left). We use the suggested setting of Cebron and Berthold (2009), e.g., $r_a = 0.4$ and $r_b = 1.25 r_a$.

Kernel farthest first (Ker) searches for most unexplored regions given the set of already labeled data (Baram *et al.*, 2004; Osugi *et al.*, 2005). First, it computes the minimum distance of an unlabeled sample to all labeled data

$$Ker(x_i) = \min_{x_j \in L} d(x_i, x_j), \tag{5.4}$$

with Euclidean distance d. Then, it requests the label for the farthest sample $x^* = \text{argmax}_i Ker(x_i)$. This criteria works well for datasets with a smooth manifold structure as it samples evenly the entire data space. As soon as there are more complex datasets, this measure selects many outliers (see Figure 5.5 left).

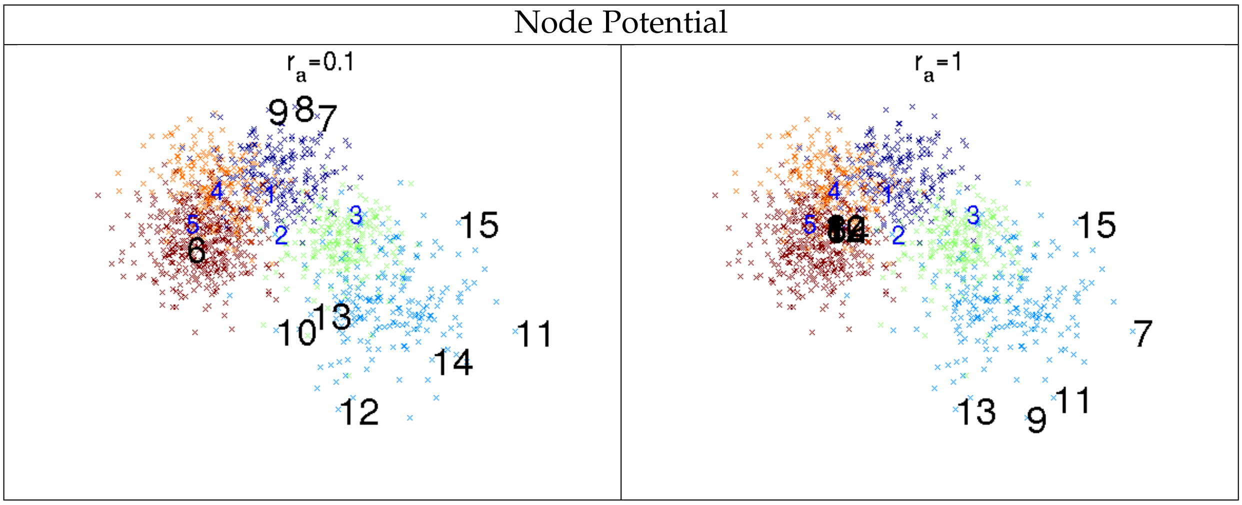


Figure 5.4. Visualization of different parameters for the node potential. Small r_a (left) causes that only outliers are sampled because of the large neighborhood radius and larger r_a results in oversampling of dense regions (right).

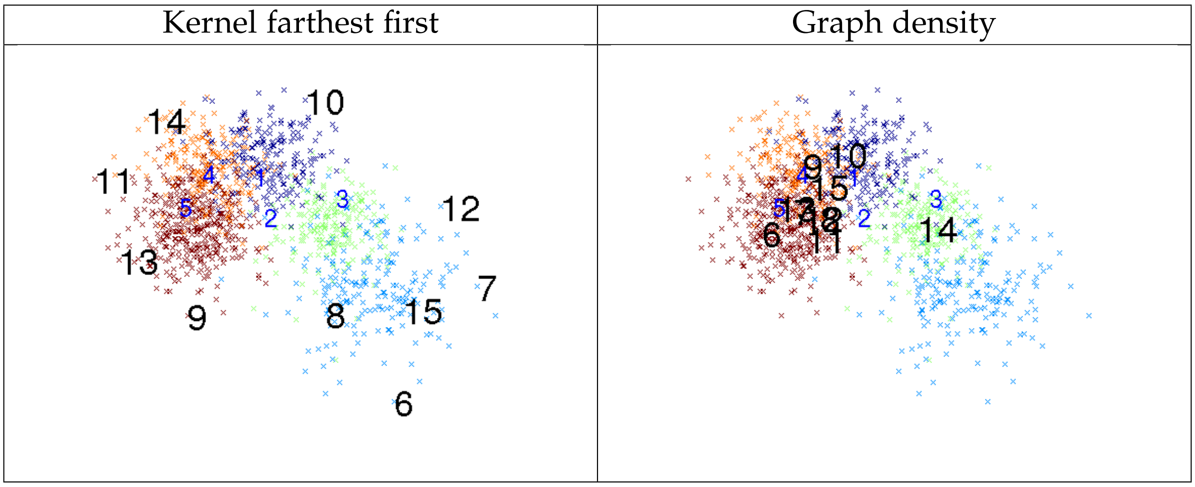


Figure 5.5. Starting with 1 label per class (green and blue dots), we request iterative labels with the different sampling strategies: a) kernel farthest first sample evenly the entire space leading to many outlier; b) graph density focus more on the dense regions of the mixtures. The numbers beside the selected points indicate the order in which they were selected.

5.2.3 Graph density criteria for exploration

Our novel sampling criteria uses the symmetric k nearest neighbor graph structure W from Equation (3.6) to identify highly connected nodes. We assume that these nodes are more representative for a class because they are usually well embedded in this graph structure and have many edges ($\gg k$) with high weights. To distinguish among data points with many small weighted neighbors that we called in Section 1.2

hub nodes, we normalize these weights by the number of edges:

$$Gra(x_i) = \frac{\sum_i W_{ij}}{\sum_i P_{ij}}, \tag{5.5}$$

with adjacency matrix P. Similar to the *node potential*, we reduce the weights of direct neighbors of the currently selected node x_i with

$$Gra(x_j) = Gra(x_j) - Gra(x_i)P_{ij} \tag{5.6}$$

This avoids that the same dense region is selected multiple times. Our criteria focuses on representative regions due to the normalization factor, it avoids sampling of outliers, and is more robust than *node potential* due to the underlying k-NN graph structure (Figure 5.5 (right)).

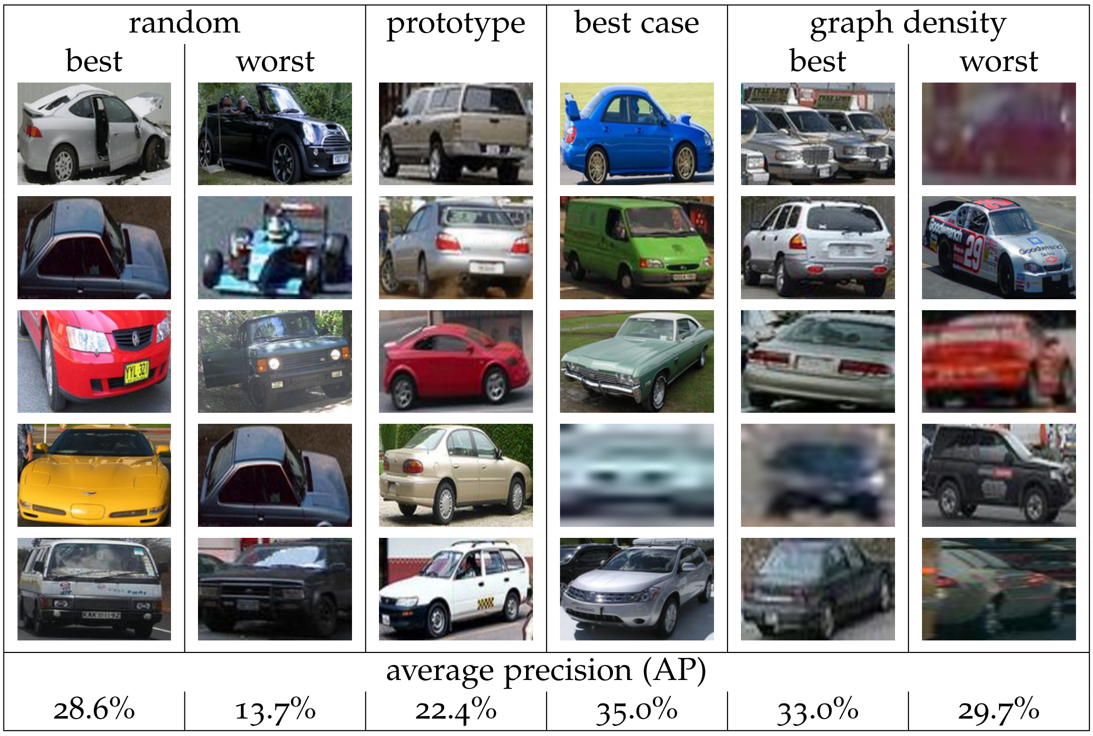


Figure 5.6. Training samples of C-PASCAL: random best and worst seed (column 1-2), prototypical selection, and best case estimation (4th column), and with our graph density criteria as a filter criteria for best and worst seed (column 5-6). AP is the average precision for this class calculated after applying label propagation.

To demonstrate the representativeness of our criteria, we show in Figure 5.6 the labels we get for the class *car* from C-PASCAL in comparison to random sampling. Additional, we compute the average precision (AP) after applying label propagation on these 5 labels. The first column shows the best random draw of 5 random draws w.r.t. average precision (PASCAL VOC criteria) of the retrieved unlabeled examples. We observe a good coverage of intra-class variation and view-points. The next column shows the worst draw. Atypical examples, less viewpoint variation, and truncation have lead to a drop in average precision from 28.6% to 13.7%. Next we selected 5 prototypical examples by hand to convey our domain knowledge of cars, which results in a performance of 22.4%, right in between the best and worst results of a random draw. For the best case (column 4), we cluster the data to find 5 modes and use then our density criteria to find the most representative sample per mode. Please note that this is a best-case type analysis as we are finding the modes for the cars isolated from the other classes. However, this leads to a performance of 35% which is over 6% better than the best random draw we have and over 20% better than the worst one. For the last two columns, we use *graph density* to rank the images according to their density and select the first 500 images with the highest score. From these images we randomly draw 5 images resulting in a best and a worst draw w.r.t. average precision. As we can see from the images, these examples are more representative for this class and capture a broad range of different viewpoints. And also the average precision is with 29.7% (worst) to 33.0% (best) better than the precision for the best random draw.

Note, this figure shows only the theoretic representativeness of labels. It assumes that we know which images belonging to class *car* otherwise there is no guarantee to sample exactly 5 labels per class. In a real active learning scenario, we cannot control the number of labels per class.

5.2.4 Experiments

In the following, we show results of these previously mentioned single criteria for the HOG descriptor. We compare to a random baseline where samples are drawn with a uniform distribution. Overall accuracy after $\max(5C, 100)$ iterations with C number of classes are shown in Table 5.1 for all three datasets. The lower bound of 100 ensures for each dataset a minimum of labels. In all experiments, we start with one randomly drawn label per class. We document results for label propagation (LP) as well as SVM. Finally, we average over all datasets shown in the last two columns.

We make the following observations: (1) LP is always better than SVM due to the small amount of labels. (2) The ordering of criteria according to the performance is the same for both SVM and LP.[1] When comparing the different criteria *Gra* works best for Caltech 101 with 35.5% (SVM) and 38.9% (LP). In average, *Mar* has best performance with 45.3% (SVM) and 48.8% (LP). (3) Differences between SVM and LP are larger the better the underlying criteria. For example, the difference for random sampling between SVM with 43.7% and LP with 45.3% is only 1.6% while for the best criteria *Mar* this difference increase to 3.5% between SVM and LP. This illustrates the potential of SSL if the commonly used random sampling is replaced by a more appropriate choice.

(4) In average, exploitation criteria work better than exploration criteria due to the local feedback after each labeling iteration. But given this drawback, our *Gra*

---

[1] As this holds true for this and all subsequent experiments of this paper we will report results on LP only for the remainder of this paper.

	ETH		C-PASCAL		Caltech 101		mean	
criteria	SVM	LP	SVM	LP	SVM	LP	SVM	LP
random	74.3	74.8	26.5	27.7	30.4	33.4	43.7	45.3
exploitation								
Ent	79.8	81.6	26.7	28.1	28.7	33.5	45.1	47.8
Mar	78.8	**81.7**	27.6	**30.2**	29.5	34.4	45.3	48.8
exploration								
Nod	57.3	66.2	17.1	18.0	17.8	23.8	30.7	36.0
Ker	74.6	71.0	21.2	22.6	27.3	29.3	41.0	41.0
Gra	66.8	71.8	28.7	29.9	35.5	**38.9**	43.6	46.9

Table 5.1. Overall accuracy for all single criteria and random sampling after $\max(5C, 100)$ iterations.

works remarkably well. The more difficult the dataset the larger the benefit from this criteria. For Caltech 101, our criteria is with 38.9% even better than *Mar* with 34.4%.

(5) Finally, there is no criteria that works best across all datasets. *Mar* that has in average best performance over all datasets shows indeed best performance for ETH and C-PASCAL but it loses almost 5% for Caltech 101. To conclude this section, all of the criteria have their strengths and weaknesses and it is hard to choose one criteria that works consistently best for all datasets.

5.3 time-varying active learning

In the following, we analyze different combinations of exploration and exploitation criteria as all criteria lack either on representativeness or on informativeness. In particular the exploration criteria unfold their full potential only in combination with exploitation criteria because they do not get any feedback about the classification uncertainty during the sampling process. We first explain our framework that allows a fixed as well as a time-varying trade-off between exploration and exploitation. After that, we show in the experiments that there is a consistent improvement compared to the previous section.

5.3.1 Method

Our framework is inspired by Cebron and Berthold (2009) combining exploration and exploitation with a parameter β:

$$H(x_i) = \beta U(x_i) + (1 - \beta) D(x_i) \tag{5.7}$$

where $U \in \{Ent, Mar\}$, $D \in \{Nod, Ker, Gra\}$, and $\beta \in [0, 1]$. In addition, we introduce two new improvements. First, we use a ranking function $r : \mathbb{R} \to \{1, ..., u\}$

$$r(F(x_i)) = m_i, \text{ where } F(x_i) \leq F(x_j) \Leftrightarrow m_i \geq m_j \tag{5.8}$$

with $m \in \{1, ..., u\}$ for all $l+1 \leq i, j \leq n$ and $F \in \{-Ent, -Mar, Nod, Ker, Gra\}$. This maps the continuous values of D and U to a natural number and makes both terms comparable among each other and across all datasets. Otherwise the range of values of D and U is strongly dependent on the given dataset and requires a non-trivial adjustment of β. This step is essential in our framework and avoids the previous mentioned bias to either exploration or exploitation (Figure 5.2 b and c).

Our second improvement replaces the fixed β by a sequence over time, i.e.,

$$\beta(t) : \{1, ...T\} \rightarrow [0, 1] \tag{5.9}$$

with T the maximal number of queried labels. This allows us to have a constant trade-off as well as a time-varying trade-off. The main idea is that some datasets might need more exploration at the beginning and more exploitation at the end while other datasets might need a constant trade-off.

The final active learning framework is of the following form:

$$H(x_i) = \beta(t)r(U(x_i)) + (1 - \beta(t))r(D(x_i)) \tag{5.10}$$

In each iteration we request the label for the sample with the minimal score

$$\text{argmin}_{x_i \in U} H(x_i). \tag{5.11}$$

5.3.2 Analyzing the trade-off on synthetic datasets

Before we evaluate our framework on real datasets, we analyze the importance and the diverseness of this trade-off on different artificial datasets. For this purpose, we select four different synthetic dataset that have been used in previous active learning literature (Zhang *et al.*, 2011a; Dasgupta and Hsu, 2008; Osugi *et al.*, 2005): *Checkerboard*, *Mixture*, *SwissRoll*, and *TwoCircle* (Figure 5.7). These cover a broad range of the different types of manifold structures. On one side, *Checkerboard* consists of two classes distributed into disconnected dense regions. And on the other side, *Two Circle* is composed of two circles with the same number of data points. Thus the inner circle has a higher density than the outer circle that leads to a misleading density information. For this dataset, we would expect that density criteria decrease the performance as only points from the inner circle are sampled. For all datasets, we use 250 points per class or dense regions (in the case of *Checkerboard*). The mixtures in Figure 5.7a) and b) and the noise in Figure 5.7c) and d) are generated with a normal distribution.

We fix the combination of criteria to *Gra* for exploration and to *Ent* for exploitation and experiment with different $\beta(t)$ curves which reflects the different requirements of the individual data sets from pure exploitation-driven sampling over different kinds of trade-offs to pure exploration-driven sampling. We investigate the following sequences for $\beta(t)$: $\beta(t) = 0.0$, $\beta(t) = 0.25$, $\beta(t) = 0.5$, $\beta(t) = 0.75$, $\beta(t) = 1.0$, $\beta(t) = t$, $\beta(t) = -t$, $\beta(t) = \log(t)$, $\beta(t) = t^2$ and $\beta(t) = -t^2$. An additional degree of freedom is introduced by varying the top and low point of the time varying curves. For the low point we use 0 and 0.25 and for the high point 0.75 and 1. These scaled versions of our functions are denoted by an additional index, e.g. $\beta(t)_{[0.25,1.0]}$. For all experiments, we start with one label per class that is randomly drawn, and request labels until we have 100 labels. To compare these different strategies, we calculate the area under the accuracy curve (AUC). These numbers can be found in Table 5.2. We sort the β sequences from Figure 5.7 and the datasets by their amount of exploration and mark the best value per line with bold. This makes the shift from density-based sampling to the uncertainty-based sampling obvious as the best values lie on the diagonal.

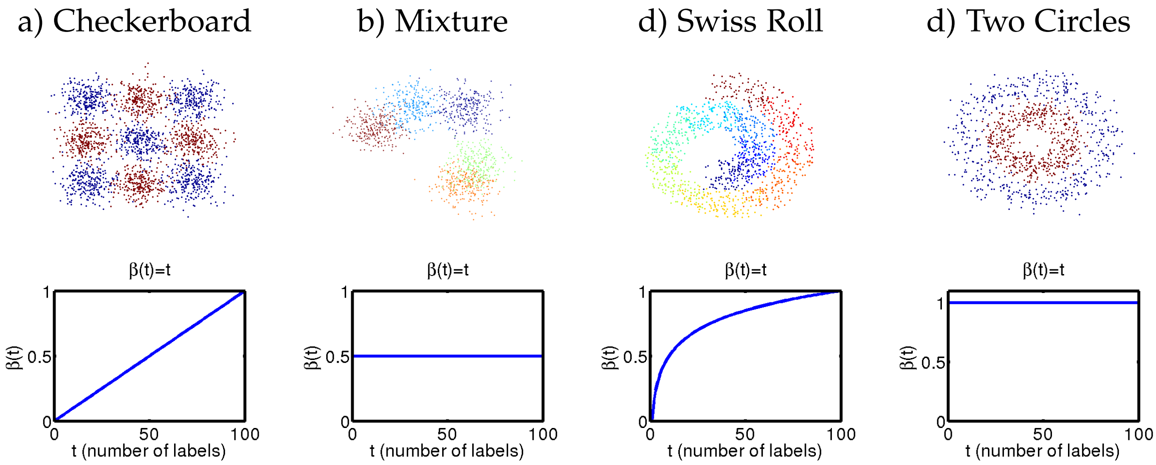

a) Checkerboard b) Mixture d) Swiss Roll d) Two Circles

Figure 5.7. Several synthetic datasets (upper row) and their corresponding time varying curve $\beta(t)$ (lower row).

The second row of Figure 5.7 shows for each dataset the best performing $\beta(t)$ time series. We found these in line with our expectations on the requirements for each set: *Checkerboard* for example needs a high exploration phase at the beginning to find all modes of the data. Thus the $\beta(t)$ sequence starts with values close to zero and increases linearly to one. *Mixture* do not need this strong exploration phase at the beginning as we start with one randomly drawn label per class. The main clusters are found but we still want to avoid outlier sampling, so best function turns out to be $\beta(t) = 0.5$. For *SwissRoll* we expect a mixed strategy but with a higher focus on uncertainty-based sampling. And finally, for *TwoCircles* with the misleading density information, we find the optimal strategy to be $\beta(t) = 1.0$ that is identical to the pure uncertainty-based sampling.

Finally, we have a closer look on the dataset *Two Circle*. This dataset comes with a misleading density information as the inner circle contains the same number of examples as the outer circle but more densely sampled. As we can see in Figure 5.8, the more focus is on the exploitation-driven sampling the more labels from the outer circle are requested leading to the best solution when using only the exploitation criteria. But more important is that this visualization shows the good balancing between exploration and exploitation caused by our previously introduced ranking

	$\beta(t)$								
	best functions				other functions				
dataset	t	0.5	$\log(t)$	1.0	0.0	0.25	$-t$	$\log(t)_{[0.25,1.0]}$	t^2
Checkerboard	**83.9**	83.4	83.3	75.5	72.6	83.6	81.2	83.5	82.1
Mixtures	78.0	**80.0**	79.3	79.6	70.3	78.4	79.1	78.7	76.5
Swiss Roll	67.8	68.0	**69.6**	69.3	53.5	62.4	67.8	68.9	62.7
Two Circles	78.8	89.1	90.9	**91.7**	50.7	55.9	89.0	90.4	65.5
mean	77.1	80.1	**80.7**	79.0	61.8	70.1	79.3	80.4	71.7

Table 5.2. Area under Curve (AUC) for the synthetic datasets for different $\beta(t)$

function. This means that for $\beta(t) = 0.5$ both criteria contribute equally resulting in a semi-correctly labeled outer circle.

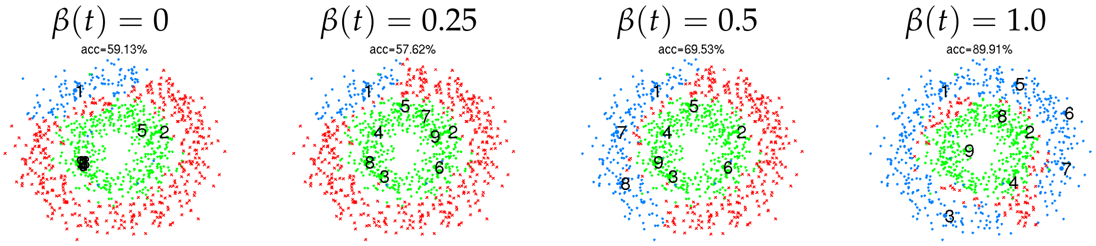


Figure 5.8. First queried labels of *Two Circles* for different strategies from full density-based to full uncertainty-based sampling. Green and blue points are correct classified. The first two labels are given. The numbers beside the data points indicate the order in which they were selected.

To conclude this analysis, we show that a time-varying trade-off is crucial. But there is no single strategy that works best for all datasets. Some datasets need more exploration at the beginning and more exploitation at the end (*Checkerboard*). Other datasets do not require exploration at all (*Two Circles*), and *Mixture* needs a constant trade-off.

5.3.3 Experiments on real data

Similar to the previous subsection, we investigate different forms of $\beta(t)$ ranging from constant to concave and convex shapes. We document overall accuracy for $\max(5C, 100)$ samples per dataset. Figure 5.9 shows all combinations with different constant functions from $\beta(t) = 0$ (pure exploration) to $\beta(t) = 1$ (pure exploitation). For the time-varying function, we use $log(t)$ and t that represents the strategy of exploration at the beginning followed by more exploitation, and $-log(t)$ and $-t$ as the complement. These values are rescaled such that $\beta(t) \in [0, 1]$. Table 5.3 contains all results. As fixed combination, we show results for $\beta(t) = 0.5$ that in average is among the best-performing fixed combinations (see Figure 5.9). Finally, we compute the average over all datasets.

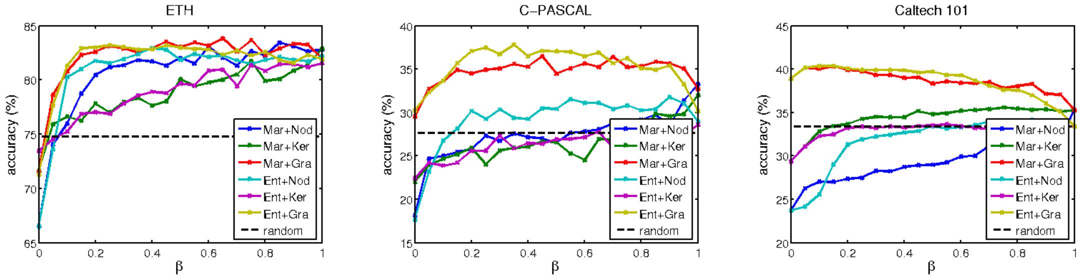


Figure 5.9. Simple mixtures with different constant β for all datasets and the comparison to random sampling.

	a) ETH					**b) C-PASCAL**				
	fixed	time-varying				fixed	time-varying			
combination	0.5	log(t)	t	-log(t)	-t	0.5	log(t)	t	-log(t)	-t
Mar+Nod	81.5	**82.9**	82.5	78.5	81.0	27.0	**29.3**	28.7	25.3	27.9
Mar+KFF	78.9	**81.0**	80.6	74.2	77.3	26.9	**29.8**	26.3	26.5	28.6
Mar+Gra	**83.9**	83.1	83.4	77.6	82.5	30.1	33.6	34.5	34.6	**36.2**
Ent+Nod	**82.5**	81.5	82.0	79.0	81.0	26.3	**31.5**	30.8	26.7	30.3
Ent+Ker	78.7	**82.3**	81.6	75.1	78.0	23.6	**27.3**	25.9	24.8	25.3
Ent+Gra	**83.0**	81.8	82.3	78.2	82.3	31.4	34.9	35.5	34.4	**36.6**

	c) Caltech 101					**d) mean over all datasets**				
	fixed	time-varying				fixed	time-varying			
combination	0.5	log(t)	t	-log(t)	-t	0.5	log(t)	t	-log(t)	-t
Mar+Nod	29.5	**32.0**	28.8	25.8	29.1	46.0	**48.0**	46.7	43.2	46.0
Mar+Ker	34.4	**35.2**	34.9	30.6	33.6	46.7	**48.7**	47.2	43.8	46.5
Mar+Gra	39.5	37.9	**39.7**	39.5	39.1	51.1	51.5	52.5	50.5	**52.6**
Ent+Nod	33.5	34.2	**35.0**	26.7	32.8	47.4	49.1	**49.3**	44.2	48.0
Ent+Ker	33.2	32.4	**33.3**	30.4	32.2	45.2	**47.4**	46.9	43.4	45.2
Ent+Gra	39.8	36.0	**39.9**	39.7	39.1	51.4	50.9	52.6	50.8	**52.7**

Table 5.3. Overall accuracy for $\beta(t) = 0.5$ and four different time-varying combinations.

We observe a consistent improvement over single criteria. This can be seen in Figure 5.9 where best values of solid lines are usually in the range of $0 < \beta(t) < 1$. Almost all combinations are better than random sampling. For C-PASCAL, we improve LP with random sampling from 27.7% to 36.6% with *Ent+Gra*. Our *Gra* criteria works always best for all datasets in combination either with *Ent* (red curve) or with *Mar* (yellow curve). This improvement is more pronounced the more difficult the dataset. A time-varying scheme is across all datasets (Table 5.3d) better than a fixed combination. For *Mar+Gra*, we improve the fixed combination of $\beta(t) = 0.5$ from 51.1% to 52.5% when using $\beta(t) = t$. Different datasets need a different trade-off (see Figure 5.9). While all curves for ETH have a tendency to increase toward $\beta(t) = 1$, the two top-curves of Caltech 101 are decreasing, i.e., need more exploration than exploitation. In average over all datasets *Ent+Gra* with $\beta(t) = -t$ work best with 52.7%. But when we look at each specific dataset this only holds true for C-PASCAL. ETH shows best performance with *Mar+Gra* and $\beta(t) = 0.5$ and Caltech 101 with *Ent+Gra* and $\beta(t) = t$.

Again, there is no single strategy that works best across all datasets. Consequently, we propose a new dataset-specific method in the next section that allows time-varying combinations as well as different combinations of criteria.

5.4 RALF - A REINFORCED ACTIVE LEARNING FORMULATION

A combination of two criteria as well as a time-varying trade-off between exploration and exploitation are key ingredients to improve active learning. But there is no single strategy that works best across all datasets. The question arises how can we find a good trade-off and combination of criteria without prior knowledge on the dataset and its interplay with the criteria? To address this challenge, we consider the entire active learning sequence as a process that is optimized by learning a strategy from feedback "on the fly". This constitutes a challenging meta-learning problem only allowing for indirect and approximative observations. In the following, we investigate different proxies for the true gain in performance in each stage and how they can be used to guide the next label query. A crucial difference to previous methods is that we now aim at modeling the progress of the learned classifier and exploit this information to control the trade-off between different individual criteria. In the second part we propose a model to aggregate feedback over time and learn an effective strategy from experience over multiple active learning rounds. Hence, we phrase the problem as learning to perform active learning. The integration of multiple criteria which are combined in this flexible and adaptive fashion shows excellent performance across three challenging datasets without any need to inform the method about the specifics of the dataset or the available criteria.

5.4.1 Switching feedback-driven between two criteria

Inspired by Osugi *et al.* (2005), we explore feedback after each labeling iteration to update a probability

$$p = \max(\min(p\lambda \exp(r^{(t)}), 1-\epsilon), \epsilon) \tag{5.12}$$

that is used to switch randomly between exploration and exploitation with reward $r^{(t)}$ and $p \in [\epsilon, 1-\epsilon]$. ϵ ensures minimal exploration (resp. exploitation). The higher p the higher the probability that exploration is selected for the next iteration. Parameter λ is the learning rate that controls the influence of the reward. Feedback $r^{(t)}$ is given by the change of the previous hypothesis $Y^{(t-1)}$ given by our classifier to the current hypothesis $Y^{(t)}$

$$r^{(t)} = \frac{\langle Y^{(t-1)}, Y^{(t)} \rangle}{\|Y^{(t-1)}\| \|Y^{(t)}\|}. \tag{5.13}$$

This feedback is rescaled with $r^{(t)} \leftarrow 3 - 4r^{(t)}$ otherwise exploration dominates exploitation.

Our first improvement integrates p in our active learning framework that means

$$\beta(t) = 1 - p \tag{5.14}$$

so that there is always a mixture of two criteria. Second, we propose a more general rescaling function $s : \mathbb{R} \mapsto I = [-1, 1]$ as the previous mentioned rescaling turns out to not generalize well to new datasets. Instead, we consider all observed rewards until iteration t to map these values into an interval I,

$$s(r^t) = \tilde{r}^{(t)} \frac{\max(I) - \min(I)}{\max_i r^{(i)} - \min_i r^{(i)}} - \min(I) \tag{5.15}$$

with $1 \leq i \leq t$ and $\tilde{r}^{(t)} = r^{(t)} - \min_i r^{(i)}$.

Third, we propose a new reward function $r^{(t)}$ that is more closely related to the actual performance of the classifier compared to the mere change in the prediction. While classification accuracy on the whole dataset is obviously not available during learning, we propose to use the difference in the overall entropy of the class posteriors between two time steps as a proxy for measuring the learning progress,

$$r_{Ent}^{(t)} = \sum_{i=1}^{u} Ent^{(t-1)}(x_i) - \sum_{i=1}^{u} Ent^{(t)}(x_i). \tag{5.16}$$

with $Ent^{(t)}(x_i)$ the entropy of unlabeled sample x_i at iteration t. This reward is rescaled with our function s from Equation (5.15) to get positive as well as negative feedback.

5.4.2 Reinforced active learning formulation (RALF)

The previous method proposes a first way to incorporate feedback. But there is no learning involved yet. Therefore we suggest a method that accumulates feedback over time and is also capable to deal with more than one criteria.

Markov decision process formulation. We address this problem by formulating active learning as a Markov decision process (MDP). Figure 5.10 shows on the left side a simple MDP for two criteria. In this MDP denoted by a 4-tuple (S, A, Q, R), there is only one state $S = \{U + D\}$ with $U \in \{Mar, Ent\}$ and $D \in \{Nod, Ker, Gra\}$ a mixture of two sampling criteria. Further, there are n actions that represent n different fixed trade-offs, i.e., $A = \{\beta_1(t) = a_1, \beta_2(t) = a_2, ..., \beta_n(t) = a_n\}$ with $a_i \in [0, 1]$. Note, although actions have a fixed $\beta(t)$ this does not contradict our previous assumption of using a time-varying trade-off because we are always able to switch among different actions. R is the reward for executing action a_i in state s_j. We use the overall entropy from Equation (5.15). Finally, Q are the transition weights that action a_i is selected in state s_j. Even though each state consists of a mixture of

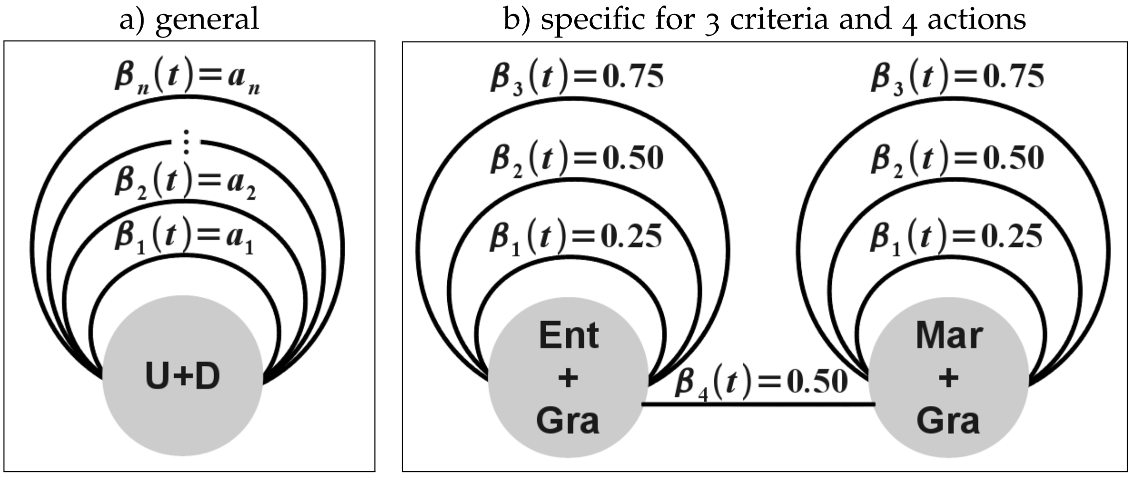


Figure 5.10. Simple Markov decision process with a) 1 state $S = \{U + D\}$ with $U \in \{Mar, Ent\}$ and $D \in \{Nod, Ker, Gra\}$, and n actions $A = \{\beta_i(t) = a_i\}$ with $a_i \in [0, 1]$; b) 2 states $S = \{Ent + Gra, Mar + Gra\}$ and 4 actions $A = \{0.25(stay), 0.5(stay), 0.75(stay), 0.5(switch)\}$

two criteria it is still possible to have single criteria by choosing action $\beta_i(t) = 0$ or $\beta_i(t) = 1$.

This simple MDP can be naturally extended to a larger state space with more sampling criteria. On the right side of Figure 5.10 there is an example for three criteria, i.e., $U \in \{Ent, Mar\}$, $D \in \{Gra\}$, and three different mixtures. Additional to the three actions of changing the trade-off there are now m actions for each new state to switch the state with m different trade-offs.

Learning. To learn this MDP, we use the *model-free* method Q-Learning as we have no prior knowledge about the underlying model. Q-Learning is a fast and adaptive reinforcement learning algorithm that learns our transition table $Q \in \mathbb{R}^{|S| \times |A|}$ online during the active learning process. After each transition $s^{(t-1)} \rightarrow a \rightarrow s^{(t)}$ entry $Q(s^{(t-1)}, a)$ is updated given the current reward $r^{(t)}$, i.e,

$$\begin{aligned} Q(s^{(t-1)}, a) \quad \leftarrow \quad & Q(s^{(t-1)}, a) + \lambda(r^{(t)} + \\ & \gamma \max_{a_i} Q(s^{(t)}, a_i) - Q(s^{(t-1)}, a)). \end{aligned} \tag{5.17}$$

Parameter λ is the learning rate that controls the influence of the current reward $r^{(t)}$, and $0 \leq \gamma \leq 1$ is the discount factor that weights the future reward. When $\gamma = 0$ only the current reward $r^{(t)}$ is considered for updating and any previous experience with this state-action-pair are ignored. During the active learning process, we decide for action $a = \max_{a_i} Q(s^{(t-1)}, a_i)$ and use mixture of state $s^{(t)}$ with trade-off a to request the next label.

In summary, our model has two parameters for Q-Learning that are obtained from previous reinforcement learning papers. In addition, there are the number of states and actions that should be kept as small as possible to speed up the initialization. All parameters are the same across all datasets. There is no tuning to one specific dataset.

Initialization. One challenge we face is initialization of the method as we start with an empty Q table. Ideally, we visit each state-action-pair once or twice but this is undesired for our state and action space. The number of iterations are limited and we would try out many transitions that are not helpful for our learning process.

Therefore, we propose a guided initialization phase inspired by Yan *et al.* (2003). We compute the expected entropy reduction $\hat{r}_i^{(t)}$ for all actions a_i. Each action a_i requests a label for sample x_i. As we do not know the label for this sample, we apply our classifier for each class and calculate the overall entropy. These entropies are weighted by our current prediction probability $p(y_{ij}|x_i)$, i.e.,

$$\hat{r}_i^{(t)} = \sum_{j=1}^{c} p(y_{ij}|x_i) \sum_{k=1}^{n} Ent_j(x_k). \tag{5.18}$$

Ent_j is the entropy after running our classifier with label j for sample x_i. Finally, we select the next action with $a = \text{argmax}_i \hat{r}_i^{(t)}$. Of course, this is a time-consuming step but we use this only for the first few iterations. Also, we can reduce the number of classes for estimation with threshold $p(y_{ij}|x_i) > 0.01$. Usually, there are only 2 to 4 classes left.

We set $\epsilon = 0.05$ and we fix $\gamma = 1$ as we want as much as possible benefit from our previous experience. Finally, we want a time-varying also called *non-stationary* model. So we set $\lambda = 0.5$ as otherwise it converges to a fixed solution and later changes are almost impossible.

5.4.3 Experiments

In Table 5.4, we show results for all feedback-driven methods with two criteria. The first column of each block shows performance of the method by Osugi *et al.* (2005). The second and third columns contain our improvements for this method from Equations (5.15) and (5.16). The last column shows our MDP method with one state $S = \{U + D\}$ for each combination and three different trade-offs $A = \{\beta(t) = 0.25, \beta(t) = 0.5, \beta(t) = 0.75\}$. As before, we document overall accuracy after $T = \max(100, 5c)$ iterations, and start with one label per class.

All our proposed methods outperform Osugi *et al.* (2005). Moreover, for our best combination *Ent+Gra*, there is a consistent increase in performance from the first column to the last column across datasets. For C-PASCAL, e.g., we get a performance of 28.4% with Osugi *et al.* (2005). It is then increased to 31.9% with our general scaling function $s(r^{(t)})$, and to 33.7% with our entropy-based reward function. Finally, we improve up to 37.3% when using our MDP-based method that outperforms the best time-varying combination from Table 5.3 $\beta(t) = -t$ with 36.6%. This observation also holds true for the mean over all datasets where we increase *Ent+Gra* from 47.9% to 53.7%.

	ETH				**C-PASCAL**			
combination	[Osugi]	$s(r^t)$	$r^{(t)}_{Ent}$	RALF	[Osugi]	$s(r^t)$	$r^{(t)}_{Ent}$	RALF
Mar+Nod	81.4	82.9	**83.2**	81.8	31.3	32.5	32.1	31.7
Mar+KFF	81.5	81.2	**82.8**	80.0	31.2	**33.2**	30.5	31.6
Mar+Gra	82.0	83.2	83.6	**83.8**	32.1	32.8	34.2	**36.5**
Ent+Nod	80.9	82.1	81.6	**82.5**	27.6	30.0	29.4	**31.2**
Ent+Ker	81.5	81.9	81.9	82.1	27.8	29.9	**30.1**	29.8
Ent+Gra	81.5	81.8	82.3	**83.6**	28.4	31.9	33.7	**37.3**
	Caltech 101				**mean over all datasets**			
combination	[Osugi]	$s(r^t)$	$r^{(t)}_{Ent}$	RALF	[Osugi]	$s(r^t)$	$r^{(t)}_{Ent}$	RALF
Mar+Nod	35.1	**35.8**	35.4	30.5	49.3	**50.4**	50.2	48.0
Mar+KFF	34.6	**35.8**	35.4	35.1	49.1	**49.9**	49.6	48.9
Mar+Gra	35.0	35.4	35.9	**39.8**	49.7	50.5	51.2	**53.4**
Ent+Nod	33.0	33.1	33.6	**33.9**	47.1	48.4	48.2	**49.2**
Ent+Ker	33.4	33.4	**33.6**	33.1	47.6	48.4	**48.5**	48.3
Ent+Gra	33.6	33.7	33.8	**40.2**	47.9	49.1	49.9	**53.7**

Table 5.4. Accuracy for the original method proposed by Osugi *et al.* (2005), our rescaling function (Equation (5.15)), the entropy-based reward (Equation (5.16)), and our MDP-based method RALF.

In the last part of this section, we demonstrate the flexibility of our MDP-based model. In Table 5.5, we add consecutively states to our model starting with 2 states $S = \{Ent + Gra\}$ and ending with 4 states $S = \{Ent + Gra, Mar + Gra, Ent + Ker, Mar + Ker\}$, i.e., 4 criteria. In addition, we compare our results to the baseline of randomly switch between all state-action-pairs to show that our model goes beyond this baseline.

		ETH			C-PASCAL			Caltech 101		
$\lvert S\rvert$	criteria	rand	QL	diff	rand	QL	diff	rand	QL	diff
2	Ent, Gra	82.2	83.6	+1.5	36.2	37.3	+1.1	36.0	40.2	+3.7
3	+Mar	81.5	83.2	+1.7	35.7	36.7	+1.0	35.2	38.3	+3.1
4	+Ker	81.7	82.9	+1.2	34.1	36.2	+2.1	34.3	36.3	+2.1

Table 5.5. Accuracy for our MDP-based approach with 2 to 4 states compared to randomly switching among those action-state-pairs, and the difference to the later one.

We observe a slight decrease in performance due to the larger number of states. Both initialization and the time-varying trade-off are more difficult to learn. Nevertheless, all results are better than the random-state-transition baseline. This illustrates once more that our model benefits from the accumulated knowledge represented by the Q table. After a short initialization, our algorithm makes use of the collected experience so far, and picks a good state-action pair given the current Q table. For the number of states $|S| = 4$, where we use four different sampling criteria, our algorithm favors after only a few iterations either *Mar+Gra* or *Ent+Gra* that is in agreement with our results from the previous section.

5.5 CONCLUSION

In this work, we model active learning as a feedback-driven Markov decision process that can change over time and find a successful strategy for each individual dataset. The proposed model is based on our findings from the first part of this chapter where we analyze different sampling criteria (Section 5.2) as well as different combinations of exploration and exploitation (Section 5.3). We argue that different datasets need different sampling strategies in a time-varying manner.

In Table 5.6, we summarize the main findings of this chapter. The first row contains results for random sampling when selecting samples with a uniform distribution. In all following lines, we calculate the difference to these numbers (column *diff*). The next two rows show best single criteria for exploitation, i.e., *margin* and exploration, i.e., *graph density* our novel criteria that works best across all datasets. Almost all these numbers are better than random sampling. In average, exploitation works slightly better than exploration due to the local feedback after each labeling iteration.

	ETH		C-PASCAL		CALTECH		mean	
strategy	LP	diff	LP	diff	LP	diff	LP	diff
random	74.8		27.7		33.4		45.3	
single criteria								
margin	81.7	+6.9	30.2	+2.5	34.4	+1.0	48.8	+3.5
graph density	71.8	-3.0	29.9	+2.2	38.9	+5.5	46.9	+1.6
fixed and time-varying trade-off								
$\beta(t) = 0.5$	83.0	+8.2	31.4	+3.7	39.8	+6.4	51.4	+6.1
$\beta(t) = -t$	82.3	+7.5	36.6	+8.9	39.1	+5.7	52.7	+5.7
$\beta(t) = t$	82.3	+7.5	35.5	+7.8	39.9	+6.5	52.6	+7.3
Feedback-driven								
our RALF	**83.6**	+8.8	**37.3**	+9.6	**40.2**	+6.8	**53.7**	+8.4

Table 5.6. Summary: Random sampling, best single exploitation and exploration criteria, best combination with fixed and time-varying trade-off, our RALF approach, and differences to random sampling.

Below, we list three different fixed and time-varying trade-offs that work best across all datasets. As can be seen, time-varying strategies are better than fixed strategies. Surprisingly, not the common sense strategy $\beta(t) = t$ with a short exploration at the beginning and a long exploitation at the end is the best time-varying trade-off but rather the opposite strategy with $\beta(t) = -t$ in particular for C-PASCAL. In the last line, we show results of our MDP-based method that outperforms all previous methods and leads to a final improvement of 9.6% for

C-PASCAL and in average to 8.4% across all datasets. This underlines the capabilities of our model to adapt to different dataset and learn an effective active learning strategy “on the fly”.

For future work, we intend a faster convergence of this model by incorporating domain knowledge or other prior knowledge as we observe even better performance when using a previous learned transition table. Another important issue is the feedback itself. At the moment, we use the entropy over all predictions that reflects the overall uncertainty of the classifier. The feedback is more positive if the selected sample causes a large reduction of this entropy. Although this reflects our goal we are aiming for, this measure still contains a certain bias because in the beginning almost each sample causes a reduction in the classifier uncertainty. Thus, states can get a positive feedback although they are not successful in the long run. Of course, our model can deal with such situations because it will successively update these states and will most likely switch to another state after few iterations. But ideally one would push a good state from the beginning on.

ACTIVE METRIC LEARNING 6

Contents

6.1 Introduction **90**
6.2 Methods **90**
  6.2.1 Metric learning 91
  6.2.2 Active sample selection 91
  6.2.3 Classification algorithms 92
6.3 Active metric learning **93**
  6.3.1 Batch active metric learning (BAML) 93
  6.3.2 Interleaved active metric learning (IAML) 94
6.4 Experiments **94**
  6.4.1 Different sampling criteria for metric learning 94
  6.4.2 BAML with LP and SVM 97
  6.4.3 Interleaved active metric learning (IAML) 97
6.5 Conclusion **98**

In the last two chapters, we show the large impact when combining techniques for graph improvements and label improvements. In this chapter, we use active learning to find more representative labels for metric learning that we employed in Section 4.3.3. The main intuition behind is that these approaches based on Mahalanobis distance usually need a large number of labels to perform reasonably well. Since our setting comes usually with a small amount of labels, they should be more representative, i.e., no redundant or misleading information from outliers. Finally in chapter 7, we improve our datasets by adding more unlabeled data resulting in a smoother manifold structure.

This chapter is structured as follows. We briefly motivate and introduce related work in Section 6.1. After reviewing the used metric learning framework and our active sampling scheme in Section 6.2, we propose two novel methods in Section 6.3 that combine both approaches in a different way. In Section 6.4, we analyze different sampling strategies to answer the question which sampling criteria and strategy is best suited to improve metric learning. This analysis is done for three different datasets, for three different algorithms, and for settings where only a small number of labels is available. We show also a significant improvement when using our proposed methods. Finally, we conclude our insight and give an outlook in Section 6.5.

6.1 INTRODUCTION

Similarity-metrics are a core building block of many computer vision methods e.g. for object detection (Malisiewicz *et al.*, 2011), human pose estimation (Straka and Hauswiesner, 2011), or image retrieval (Frome *et al.*, 2007). Consequently, their performance critically depends on the underlying metric and the resulting neighborhood structure. The ideal metric should produce small intra-class distances and large inter-class distances. But standard metrics often have problems with high dimensional features due to their equal weighting of dimensions. This problem is particularly prominent in computer vision where different feature dimensions are differently affected by noise e.g. due to signal noise, background clutter, or lighting conditions.

We have shown in Section 4.3.3 that metric learning is a promising direction to address this issue and to improve the underlying neighborhood structure for label propagation. These approaches transform the original feature space into a more distinctive one that is better suited for the task at hand. E.g. pairwise constraints from labeled data are used to enforce that examples within a class are closer than examples of different classes (Davis *et al.*, 2007; Kulis *et al.*, 2009; Rangapuram and Hein, 2012). But this strategy can be problematic (Yang *et al.*, 2007; Basu and Banerjee, 2004) if only few labels are available that might be not informative enough to learn a better metric. For example, outliers may completely distort the metric while redundant samples may have little effect on metric learning. To address the above issues of metric learning this chapter aims to combine active sampling of labels with metric learning.

In general, active learning methods such as the framework introduced in Section 5.3 use sample selection strategies to request uncertain as well as representative samples so that a higher classification performance can be achieved with only a small fraction of labeled training data. However, the success of active learning critically depends on the choice of the sample selection strategy. To the best of our knowledge, this is the first work that analyze different sampling criteria in terms of representativeness for metric learning and that combines active learning with metric learning.

6.2 METHODS

Here, we briefly review the employed metric learning algorithm (Davis *et al.*, 2007) as well as the active sampling procedure including two criteria for exploration and two criteria for exploitation. These can be used either separately or in combination within the active selection method. Finally, we briefly introduce three different classification algorithms that are used to show the improvement on applying our active metric learning, i.e., k nearest neighbor classifier (KNN), SVM, and the semi-supervised label propagation (LP) (Zhou *et al.*, 2004a).

6.2.1 Metric learning

As in Section 4.3.3, we use the information-theoretic metric learning (ITML) proposed by Davis *et al.* (2007). ITML learns a global metric by optimizing the Mahalanobis distance,

$$d_A(x_i, x_j) = (x_i - x_j)^T A(x_i - x_j), \tag{6.1}$$

between two labeled points $x_i, x_j \in \mathbb{R}$ with a Mahalanobis matrix A such that intra-class distances are small and inter-class distances are large, i.e.,

$$\begin{array}{rll} \min & D_{ld}(A, A_0) & \\ s.t. & d_A(x_i, x_j) \leq u & (i,j) \in \mathcal{S} \\ & d_A(x_i, x_j) \geq l & (i,j) \in \mathcal{D} \end{array} \tag{6.2}$$

u and l are upper and lower bounds of similarity and dissimilarity constraints. $\mathcal{S}$ and $\mathcal{D}$ are sets of similarity and dissimilarity constraints based on the labeled data. This linear optimization can be easily transformed into a kernelized optimization by $K = X^T AX$. A good solution can be efficiently found by concentrating always on the maximal violated constraint given the current metric.

6.2.2 Active sample selection

We analyze two exploration and two exploitation criteria with respect to the representativeness for metric learning. These and the time-varying active sample selection framework are briefly introduced. Let us assume we have $n = l + u$ data points with l labeled examples $L = \{(x_1, \hat{y}_1), ..., (x_l, \hat{y}_l)\}$ and u unlabeled examples $U = \{x_{l+1}, ..., x_n\}$ with $x_i \in \mathbb{R}^d$. We denote $\hat{y} \in \mathcal{L} = \{1, ..., C\}$ the labels with C the number of classes.

Exploitation. *Entropy* (Ent) is the most common criteria for exploitation (Baram *et al.*, 2004) that uses the class posterior:

$$Ent(x_i) = -\sum_{j=1}^{c} P(y_{ij}|x_i) \log P(y_{ij}|x_i) \tag{6.3}$$

where $\sum_j P(y_{ij}|x_i) = 1$ are predictions of a classifier. This criteria focuses more on examples that have a high overall class confusion. Usually these samples come either from highly overlapping regions or from low-density regions.

Margin (Mar) computes the difference between best versus second best class prediction (Settles and Craven, 2008):

$$Mar(x_i) = P(y_{ik_1}|x_i) - P(y_{ik_2}|x_i) \tag{6.4}$$

such that $P(y_{ik_1}|x_i) \geq P(y_{ik_2}|x_i) \geq ... \geq P(y_{ik_c}|x_i)$. In each iteration, label $x^* = \text{argmin}_{x_i \in U} Mar(x_i)$ is queried. In contrast to *Ent*, this criteria concentrates more on the decision boundaries between two classes.

Exploration. These criteria are often used in combination with exploitation criteria as they do not get any feedback during the active sample selection so that more labels are required to obtain good performance.

Kernel farthest first (Ker) captures the entire data space by looking for the most unexplored regions given the current labels (Baram *et al.*, 2004; Basu and Banerjee, 2004) by computing the minimum distance from each unlabeled sample to all labels

$$Ker(x_i) \quad = \quad \min_{x_j \in L} d(x_i, x_j), \tag{6.5}$$

and then requesting the label for the farthest sample $x^* = \text{argmax}_i Ker(x_i)$. This criteria samples evenly the entire data space but often selects many outliers.

Graph density (Gra) introduced in Section 5.2.2 is a sampling criteria that uses a k-nearest neighbor graph structure to find highly connected nodes, i.e.,

$$Gra(x_i) = \frac{\sum_i W_{ij}}{\sum_i P_{ij}}. \tag{6.6}$$

with the similarity matrix $W_{ij} = P_{ij} \exp\left(\frac{-d(x_i,x_j)}{2\sigma^2}\right)$ and the adjacency matrix P_{ij}. After each sampling step, the weights of direct neighbors of sample x_i are reduced by $Gra(x_j) = Gra(x_j) - Gra(x_i)P_{ij}$ to avoid oversampling of a region.

Active sampling. We use a time-varying combination of exploration and exploitation that we introduced in Section 5.3, i.e.,

$$H(x_i) = \beta(t) r(U(x_i)) + (1 - \beta(t)) r(D(x_i)) \tag{6.7}$$

with $U \in \{Ent, Mar\}$, $D \in \{Ker, Gra\}$, $\beta(t) : \{1, ..., T\} \rightarrow [0, 1]$, and a ranking function $r : \mathbb{R} \rightarrow \{1, ..., u\}$ that uses the ordering of both criteria instead of the values itself. We set $\beta(t) = log(t)$ that means more exploration at the beginning followed by exploitation at the end of the sampling process. Finally, we request the label for the sample with the minimal score $\text{argmin}_{x_i \in U} H(x_i)$.

6.2.3 Classification algorithms

In the following, we explain the use of three different classifier in the active sampling framework because not all classifier provide a class posterior that can be immediately used for *Ent* or *Mar*.

1) KNN. Similar to Kulis *et al.* (2009), we show results for the k nearest neighbor classifier with $k = 1$ because it shows consistently best performance. For the class posterior $p(y_{ij}|x_i)$, we use the confusion of the 10 nearest labels for each unlabeled data point weighted by their similarity and finally normalized by the overall sum.

2) SVM. We apply libSVM (Chang and Lin, 2011) with our own kernels in a one-vs-one classification scheme. The accumulated and normalized decision values are used as the class posterior. This shows better performance than using accumulated probability estimates. Parameter C is empirically determined but is quite robust.

3) Label propagation (LP). For semi-supervised learning, we use (Zhou *et al.*, 2004a) that propagates labels through a k nearest neighbor structure, i.e.,

$$Y_j^{(t+1)} = \alpha S Y_j^{(t)} + (1-\alpha) Y_j^{(0)} \tag{6.8}$$

with $1 \leq j \leq c$, the symmetric graph Laplacian $S = I - D^{-1/2} W D^{-1/2}$ based on the similarity matrix W from above, the diagonal matrix $D_{ii} = \sum_j W_{ij}$, the original label vector $Y_j^{(0)}$ consisting of $1, -1$ for labeled data and 0 for the unlabeled data, and parameter $\alpha \in (0, 1]$ that controls the overwriting of the original labels. The final prediction is obtained by $\hat{Y} = \mathrm{argmax}_{j \leq c} Y_j^{(t+1)}$. For the class posterior, we use the normalized class predictions

$$P(y_{ij}|x_i) = \frac{y_{ij}^{(t+1)}}{\sum_{j=1}^{c} y_{ij}^{(t+1)}}. \tag{6.9}$$

6.3 ACTIVE METRIC LEARNING

As motivated above we combine active sampling with the ITML framework. By requesting more informative and representative training examples, we expect the metric learning method to achieve better performance given the same amount of training data or – respectively – achieve equal performance already with significantly less annotated data. To this end, we explore two different ways to combine active sampling with metric learning.

6.3.1 Batch active metric learning (BAML)

Our first active metric learning approach starts by querying the desired number of labeled data points according to the chosen sample selection strategy and learns a metric based on this labeled data. As the metric is learnt only once across the whole pool of labeled data points, we call this approach *Batch active metric learning (BAML)*. While this method obtains good performance, it does not get any direct feedback involving the learnt metric during sampling. To improve the coupling between the two processes we propose a second version of our method which interleaves active sampling and metric learning.

6.3.2 Interleaved active metric learning (IAML)

The second active metric learning approach alternates between active sampling and metric learning. We start with active sampling in order to have a minimum of similarity constraints for metric learning. In our experiments, we apply metric learning each mc iterations with $2 \leq m \leq |L|$, c the number of classes, and $|L|$ the average number of requested labels per class. After metric learning we use the learned kernel to request the next batch of labels with active sampling. In each iteration we learn the metric based on the original feature space with the current available labels and all pairwise constraints. We found experimentally that using the original feature space is less susceptible to drift than incrementally updating the learnt metric.

6.4 EXPERIMENTS

In our experimental section, we first analyze in Section 6.4.1 different sampling criteria and their combinations in terms of representativeness for metric learning. We focus on the 1-NN classification performance as it reflects the change of the underlying metric. Then, we explore in Section 6.4.2 if these insights transfer also to other algorithms. Finally in Section 6.4.3, we show further improvements by applying our interleaved active metric learning (IAML) framework. All our experiments are done on three different datasets, i.e., ETH-80, C-PASCAL, and IM100 that is a subset of ILSVRC 2010 containing 100 classes, with dense SIFT. See Section 3.3 for more details.

6.4.1 Different sampling criteria for metric learning

In this subsection, we explore several sampling criteria and mixtures of those in comparison to random sampling and their influence on the entire metric. For this purpose, we look at the 1-NN accuracy as this measure gives a good intuition about the learned neighborhood structure. Table 6.1 shows results before and after metric learning for different average number of labels per class $|L|$. As we are interested in particular in a low sample regime we request at most 10% labels, i.e., for ETH we vary $|L|$ from 5 to 25 and for IM100 from 3 to 10. *Rand* is our baseline using random sampling where we draw exactly $|L|$ labels per class with a uniform distribution. Last line in each table is the average performance over the whole column. All results are averaged over 5 runs.

Before metric learning (Table 6.1, top), we notice large differences between several sampling criteria. In average, we observe a performance of 29.7% for random sampling while for single active sampling criteria the accuracy vary from 26.2% for *Ker* to 31.4% for *Mar*. Both *Mar* and *Gra* are better than *Rand*. *Ent* and *Ker* are worse than *Rand* due to their tendency to focus more on low density regions. Then we look at each specific dataset. *Mar* performs best for ETH that contains a smooth

Accuracy before metric learning

		Single criteria				Mixture of two criteria			
$\|L\|$	Rand	Ent	Mar	Gra	Ker	M+G	M+K	E+G	E+K
ETH									
5	50.6	45.9	57.0	51.1	46.0	**59.8**	43.3	55.0	49.1
15	69.1	59.7	69.7	62.6	64.0	**71.0**	65.1	62.0	60.5
25	74.2	62.7	74.4	69.8	72.4	**77.3**	72.1	66.2	66.4
C-PASCAL									
5	12.6	11.3	16.1	17.8	9.8	19.1	11.1	17.1	10.3
15	17.5	19.8	21.0	**24.1**	12.4	23.2	14.9	21.8	17.5
25	19.3	21.8	23.4	**27.5**	13.9	24.8	17.7	24.5	19.7
IM100									
3	6.3	5.1	5.6	8.2	5.1	**8.2**	5.4	7.2	5.2
5	7.6	6.0	6.8	**9.3**	5.6	**9.3**	6.2	8.1	5.9
10	9.8	7.3	8.6	10.5	7.0	**10.6**	7.9	9.0	7.0
Overall average									
	29.7	26.6	31.4	31.2	26.2	**33.7**	27.1	30.1	26.8

Accuracy after metric learning

		Single criteria				Mixture of two criteria			
$\|L\|$	Rand	Ent	Mar	Gra	Ker	M+G	M+K	E+G	E+K
ETH									
5	61.6	59.3	67.7	52.7	67.5	**70.0**	63.3	62.7	65.8
15	79.8	67.9	82.2	69.1	80.0	**83.0**	82.0	70.7	76.3
25	82.8	74.6	84.5	78.1	83.5	**86.3**	86.1	73.3	79.4
C-PASCAL									
5	16.9	19.4	22.4	23.5	17.1	25.7	20.0	**26.2**	18.9
15	25.2	32.5	32.6	34.4	18.5	**34.5**	22.4	33.2	29.1
25	28.8	37.9	**39.0**	36.9	22.5	38.4	29.6	38.4	36.6
IM100									
3	6.7	6.4	7.4	9.3	6.8	**10.6**	7.0	9.6	6.8
5	11.4	8.6	9.6	10.7	8.0	**13.0**	9.2	11.7	8.6
10	15.9	12.6	14.6	12.5	11.1	**16.3**	14.6	15.3	12.4
Overall average									
	36.6	35.5	40.0	36.4	35.0	**42.0**	37.1	37.9	37.1

Table 6.1. 1-NN accuracy before (1st table) and after (2nd table) metric learning for single criteria and the mixtures Ent+Gra (E+G), Ent+Ker (E+K), Mar+Gra (M+G), and Mar+Ker (M+K).

manifold structure. In contrast, *Gra* tends to oversample dense regions, e.g., *pear*, leading to worse performance in comparison to *Ker*. On more complex datasets such as C-PASCAL or IM100, *Gra* clearly outperforms all other single criteria. For C-PASCAL with 25 labels per class we achieve a performance of 27.5% for *Gra* while *Mar* shows a performance of 23.4% and *Ker* achieves only 13.9% accuracy. Finally, the

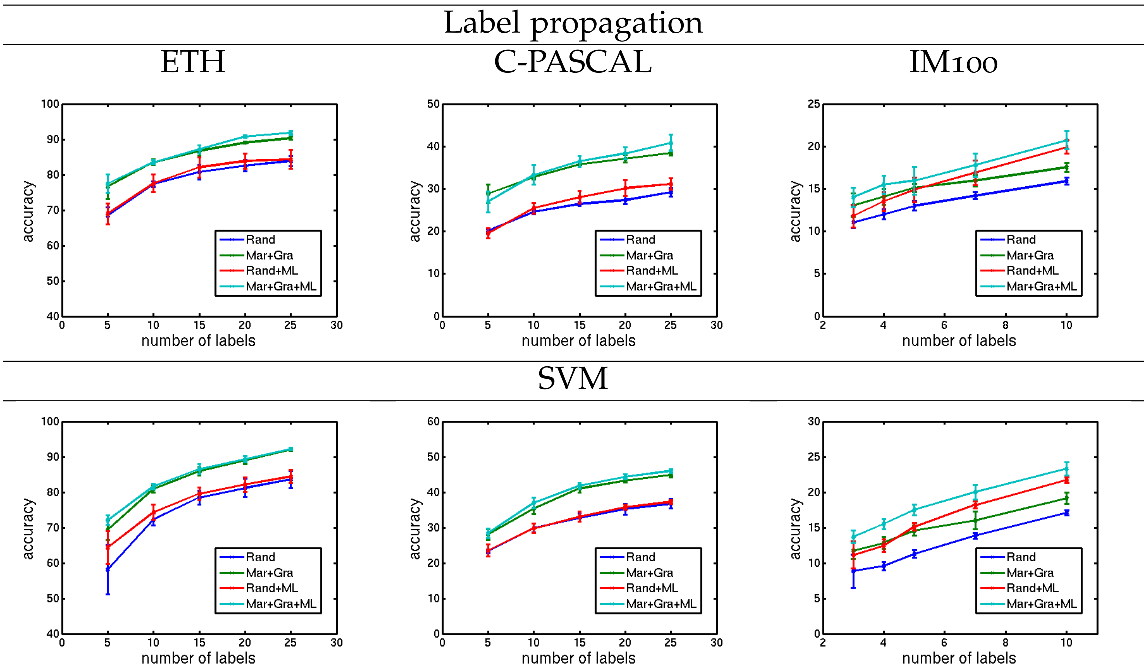


Figure 6.1. LP and SVM accuracy of all three datasets and different number of labels for random sampling and the mixture Mar+Gra with and without metric learning.

combination *Mar+Gra* outperforms with 33.7% in average the best single criteria with 31.2%. All other combinations are strongly limited by the power of the combined criteria that means using *Gra* shows better performance than using *Ker*, and mixtures with *Mar* are in average better than mixtures with *Ent*.

After metric learning (Table 6.1, bottom), we observe a consistent improvement to the previous table that means metric learning always helps. For example, *Rand* is overall improved by 6.9% from 29.7% without metric learning to 36.6% with metric learning and our best combination *Mar+Gra* is increased in average by 8.3% from 33.7% to 42.0%. From these improvements we see also that there is a larger benefit when using our BAML in comparison to *Rand* with metric learning. This observation also holds true for most other active sampling selection methods, e.g., *Ent+Ker* is improved by 10.3% from 26.8% to 37.1% that is better than *Rand* after metric learning. Another important insight results from the comparison of the influence of active sample selection on metric learning. Obviously, metric learning has a larger impact on the overall performance than active sample selection that means *Rand* is improved from 29.7% to 33.7% with *Mar+Gra* and to 36.6% with metric learning alone. But if we combine both strategies we achieve a final performance of 42.0% that corresponds to an overall increase of 12.3% across three datasets.

To conclude this subsection, metric learning benefits significantly from labels that are better suited for the task at hand. In average, *Mar+Gra* is the best sampling strategy for our BAML. Finally, metric learning combined with active sample selection achieves consistent improvements over random sampling of up to 12.3%.

	ETH			C-PASCAL			IM100		
$\|L\|$	BAML	IAML	diff	BAML	IAML	diff	BAML	IAML	diff
5	70.0	68.0	-2.0	25.7	23.3	-2.4	10.6	10.5	-0.1
10	77.4	79.8	+2.4	30.6	32.1	+1.5	11.5	12.0	+0.5
15	83.0	82.6	-0.4	34.5	40.7	+6.2	13.0	13.0	0.0
20	85.1	87.2	+2.1	36.7	41.7	+5.0	14.2	14.9	+0.7
25	86.3	90.3	+4.0	38.4	43.5	+5.1	16.3	17.1	+0.8

Table 6.2. Interleaved active metric learning (IAML) in comparison to the batch active metric learning (BAML) both for Mar+Gra sampling.

6.4.2 BAML with LP and SVM

In this subsection, we explore if our insights from the previous subsection translate to more complex classification schemes such as label propagation (LP) or SVM. Figure 6.1 shows accuracy for random sampling (*Rand*) and *Mar+Gra* – the best sampling strategy from Section 6.4.1 – before and after metric learning. The first row contains results of LP and the second row for SVM. Again, we show the average over 5 runs including standard deviation for different numbers of labels.

We also observe a consistent improvement for LP and SVM when applying BAML. For IM100 with 10 labels per class, we increase our performance with LP from 15.9% (*Rand*) to 17.5% (*Mar+Gra*) to 19.9% (*Rand+ML*) to 20.7% (*Mar+Gra*), and with SVM from 17.1% (*Rand*) to 19.2% (*Mar+Gra*) to 21.7% (*Rand+ML*) to 23.3% (*Mar+Gra+ML*). For datasets with a small number of classes, i.e., ETH and C-PASCAL, active sampling is more important than metric learning that is contrary to the previous subsection. The reason is that these methods benefit from their regularization during the learning while the KNN performance is directly connected to the neighborhood structure. But for datasets with a large number of classes like IM100, metric learning is still more important because there are more constraints to fulfill. Another interesting point becomes apparent when looking at the SVM results. For a small number of labels, SVM benefits more from metric learning although this algorithm learns a metric by itself. This can be seen in particular for ETH and IM100.

6.4.3 Interleaved active metric learning (IAML)

In this subsection, we show 1-NN results in Table 6.2 for the interleaved active metric learning (IAML) when using our best active sampling strategy *Mar+Gra*. In average, we observe an additional improvement that tends to be higher the more labels we use. For example, C-PASCAL with 15 labels is increased by 6.2% from 34.5% (BAML) to 40.7% (IAML). In few cases, we also observe a decrease in performance in particular for a small number of labels that can be explained by a drifting effect. In all experiments we recover from these issues for $|L| > 15$.

6.5 CONCLUSION

We presented an active metric learning approach that combines active sampling strategies with metric learning. While a first version (BAML) of the approach operates in batch mode and already allows to learn better metrics from fewer training examples by combining density and uncertainty-based sampling criteria, our second version (IAML) interleaves active sampling and metric learning even more tightly which leads to further performance improvements by providing better feedback to the active sampling strategy.

Our analysis of different sampling criteria and their influence on the KNN performance shows the importance of choosing an appropriate sampling scheme for metric learning. While we show consistent improvements over a random sample selection baseline, a combination of density and uncertainty-based criterion performs best on average. We show how our scheme can be applied to different supervised as well as semi-supervised classification schemes. All our experiments are carried out on three challenging object class recognition benchmark, where our new approaches consistently outperform random sample selection strategies for metric learning.

Finally, we improve the KNN results by up to 9.5% with metric learning alone and up to 24.2% with our interleaved active metric learning approach. This emphasizes our claim that a combination of both neighborhood structure improvement and label improvement leads to an even better performance.

ACTIVE DATASET CONSTRUCTION 7

Contents

7.1 Introduction **100**
7.2 Related work **101**
7.3 General SSL-framework **102**
  7.3.1 Label propagation (LP) 102
  7.3.2 Active Learning (AL) 102
7.4 Graph enhancement techniques **103**
  7.4.1 Previous techniques 103
  7.4.2 Novel techniques to enrich graph structure 104
  7.4.3 Runtime complexity 105
7.5 Experiments **107**
7.6 Conclusion **111**

This last chapter of this thesis is a first attempt towards richer and more complete datasets. In Section 1.2, we already visualize the problem when classes are splitted into disconnected dense regions, e.g., front and side view of a car. A good classification performance for these distributions can be only achieved if the training data reflect approximately the distribution of the test data, i.e., there is at least one training sample for each of these mixtures. But this assumption cannot be taken for granted in semi-supervised learning setting with only few labels. So far, we have proposed methods that tackle this problem by active learning (Section 5) or by a combination of structure and label improvement (Section 6). But these approaches tend to require many interactions from the user and it is usually not clear how many labels are needed to perform reasonably. Datasets such as C-PASCAL (Figure 3.3) that are diverse and sparse need a large quantity of labels as there is almost no underlying manifold structure that can be used for propagation. Thus, there are too many small dense regions that would require a label.

To get the full potential of semi-supervised learning algorithms, we have to uncouple ourselves from the idea that state-of-the-art datasets provide enough structural information that can be used to perform well with only few labels. Unlike to supervised learning algorithms, we are not tied to the provided and often biased datasets (Torralba, 2011). Instead, we can take these datasets as a starting point to expand and to complement those fully unsupervised under the assumption that there is a underlying manifold structure. Of course, this scenario is more complex and not so easy to control, since we do not know whether and which of the added samples are really helpful. Therefore, we first analyze existing large datasets, e.g. ILSVRC 2010, by expanding small subsets of these datasets with more unlabeled data from the remaining pool of images. Afterwards, we can easily evaluate these expansion steps as we have access to all labels. Finally, we question the "the-more-data-the-better" strategy that comes with a high runtime and space complexity.

This chapter is structured as follows. We will give a short introduction of this topic in Section 7.1 and review related work in Section 7.2. After that, we recap in Section 7.3 the used semi-supervised learning and active learning framework that we use in this work. In Section 7.4 we introduce two selection strategies to enhance our neighborhood structure in an unsupervised fashion. We compare these criteria to previous methods and show on mid-sized datasets that we improve these approaches in particular when we consider more realistic datasets with occlusions, truncations, and background clutter (Section 7.5). After that, we illustrate on a subset of ILSVRC 2010 with 100 classes that we get better performance when using only a representative subset of all images. This emphasizes our claim that there is no need to use all available unlabeled data. Additional, we also show that our approach is able to process the entire ILSVRC 2010 dataset with 1,000 classes and more than one million images. Finally, we conclude our work in Section 7.6 by applying graph propagation in combination with active learning resulting in increased performance.

7.1 INTRODUCTION

Research on semi-supervised learning (SSL) aims to leverage unlabeled data to support learning and classification tasks. A key assumption is that the underlying data distribution carries valuable information about the class distribution. In combination with the limited amount of labeled data one can achieve better performance than with labeled data alone. This idea is also fueled by the availability of vast sources of unlabeled images from the web.

Due to the active research on semi-supervised learning, the understanding of theory and algorithms in this area have greatly improved. One of the most promising frameworks is graph-based label propagation which lead to many insights (Hein and Maier, 2006) as well as high performance algorithms (Zhou *et al.*, 2005; Liu and Chang, 2009). However, those algorithms typical come with a quadratic complexity that is contradictory to the initial goal to scale up to large datasets. The "the-more-data-the-better" strategy that usually increases the performance of SSL (Ebert *et al.*, 2010) can often be not applied due to time and space complexity.

In this work, we question this strategy and show that we can indeed increase the performance with a more careful selection of *unlabeled* data. As a result we get similar or even better performance with only a fraction of all unlabeled data. This advantage becomes particularly evident when using large datasets like ILSVRC 2010 with 1,000 categories and more than a million images. In contrast to previous selection approaches (Delalleau *et al.*, 2005; Liu *et al.*, 2010) that are only applicable to mid-sized data collections with up to several 10,000 data points, we are able to handle also larger datasets. A further advantage of our selection method is that we can efficiently combine label propagation with active learning to further improve performance. In the context of active learning graph size plays a crucial role and thus our effective selection of unlabeled data becomes even more advantegous.

7.2 RELATED WORK

Large-scale computer vision has become more and more prominent in recent research. There is many works utilizing vast amount of images from the internet in order to improve one specific object category (Schroff *et al.*, 2007; Fergus *et al.*, 2010), to generate new datasets within an active learning framework (Collins *et al.*, 2008), or to use it for image retrieval (Chum *et al.*, 2008; Kulis and Grauman, 2009). For image classification, ILSVRC 2010 (Deng *et al.*, 2009) with $1,000$ classes and more than one million images is currently one of the most difficult datasets according to size and number of classes. Although, there are many approaches addressing this dataset most of them focus on faster and better image description (Perronnin and Liu, 2010; Lin *et al.*, 2011), analyze semantic similarities (Deselaers and Ferrari, 2011), or evaluate the scalability of knowledge transfer (Rohrbach *et al.*, 2011). However, there are surprisingly few works that consider more adavanced classification schemes beyond linear classifiers.

In contrast, semi-supervised learning (SSL) and in particular graph-based methods are made to leverage labeled as well as unlabeled data to improve performance of classification. We observe significant progress with respect to algorithmic contributions (Zhou *et al.*, 2004a; Sindhwani *et al.*, 2005). More recently, there is also a focus on improving graph construction – the most critical part of these algorithms. Previous works propose a better weighting function (Zhu *et al.*, 2003; Wang and Zhang, 2007a; Zhang and Lee, 2006), make use of discriminative algorithms like SVM (Zhang *et al.*, 2011b), or remove noise of the data (Hein and Maier, 2006). But although there is a common believe that more unlabeled data helps for learning, there is almost no work that address the scalability issue to take advantage of this large amount of data.

Main problem is that graph-based algorithms come with a quadratic running and space complexity. Previous work propose methods to reduce the dimensionality of the used image descriptors (Torralba *et al.*, 2008), or classify with an approximation (Fergus *et al.*, 2009). Other works reduce the amount of unlabeled data for the graph construction by constructing an anchor graph (Delalleau *et al.*, 2005; Liu *et al.*, 2010). These anchors represent the entire data distribution and label propagation can be done on a much smaller graph. In this work, we build on this idea. But instead of representing the entire data space we focus on the data regions that are most helpful for our image classification task. This allows us to run SSL on ImageNet.

7.3 GENERAL SSL-FRAMEWORK

This section briefly introduces our semi-supervised learning setup consisting of label propagation (Zhou *et al.*, 2004a) that can be combined with the active learning framework from Section 5.3 to further improve performance.

7.3.1 Label propagation (LP)

Given $n = l + u$ data point with l labeled examples $L = \{(x_1, y_1), ..., (x_l, y_l)\}$ and u unlabeled ones $x_{l+1}, ..., x_n$ with $x \in \mathbb{R}^d$ the features and $y \in \mathcal{L} = \{1, ..., C\}$ the labels. C is the number of classes. We build a symmetric k-nearest neighbor graph with the Manhattan distance (L1) and use a Gaussian kernel to get the final weighted graph W. Based on this graph a normalized graph Laplacian is computed

$$\mathcal{S} = D^{-1/2} W D^{-1/2} \quad \text{with} \quad D_{ij} = \begin{cases} \sum_j W_{ij} & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \tag{7.1}$$

We use an iterative procedure to propagate labels through this graph structure

$$Y_m^{(t+1)} = \alpha \mathcal{S} Y_m^{(t)} + (1 - \alpha) Y_m^{(0)} \text{ with } 1 \leq m \leq c, \tag{7.2}$$

with Y_m^* the limit of this sequence. Thus we avoid the time-consuming matrix inversion that would be part of the closed form solution. The initial label vector is set as follows $Y_m^{(0)} = (y_1^m, ..., y_l^m, 0, ..., 0)$ with $y_i^m \in \{1, -1\}$ for the labeled data and zero for the unlabeled data. Parameter $\alpha \in (0, 1]$ controls the overwriting of the original labels. Finally, the prediction of the data $\hat{Y} \in \mathcal{L}$ is obtained by $\hat{Y} = \text{argmax}_{1 \leq m \leq c} Y_m^*$.

7.3.2 Active Learning (AL)

Similar to Section 5.3, we combine uncertainty (exploitation) and density (exploration) criteria. For uncertainty, we use entropy over the class posterior $P(\tilde{y}_{ij}|x)$ by normalizing the prediction values from Equation (7.2):

$$\mathcal{H}(x_i) = -\sum_{j=1}^{c} P(\tilde{y}_{ij}|x_i) \log P(\tilde{y}_{ij}|x_i). \tag{7.3}$$

For the density-based sampling, we employ the graph density criteria introduced in Section 5.2.2. This criteria makes use of the symmetric k-NN graph to find dense regions and is defined by the sum of all neighboring nodes divided by the number of neighbors

$$\mathcal{D}(x_i) = \frac{\sum_j W_{ij}}{\sum_j P_{ij}}, \tag{7.4}$$

with an adjacency matrix P and the weight matrix W. To make both criteria comparable, we compute a ranking for each criteria separately such that high entropies or dense regions are mapped to small ranking values. These numbers are used to combine both criteria $s(x_i) = \beta\mathcal{H}(x_i) + (1-\beta)\mathcal{D}(x_i)$ with parameter $\beta \in [0,1]$. Finally, we query the label with the smallest score s and add this sample to our labeled set.

7.4 GRAPH ENHANCEMENT TECHNIQUES

As motivated above, graph-based SSL-techniques are quadratic in the size of the underlying graph. Therefore, we are interested in techniques that benefit from more unlabeled data while simultaneously keeping down the runtime. After reviewing previous techniques (Section 7.4.1) we propose two novel techniques (Section 7.4.2) that can be scaled to far larger datasets than previous techniques due to their lower computational complexity (Section 7.4.3).

7.4.1 Previous techniques

Several approaches have been proposed to enrich a given dataset. The simplest one is to add unlabeled data randomly with a uniform distribution either from an already existing dataset or from the internet. To have a strong baseline for our experiments, we enrich our data distribution with already existing datasets to exclude wrongly annotated and thus misleading images that are an integral part of web sources. We call this baseline *random*.

There are two other approaches that propose a graph construction with a representative unlabeled subset called anchor graph. In Liu *et al.* (2010), *k-means* cluster centroids are used as anchor points which can be advantageous when the clusters represent one class each. Otherwise they introduce many shortcuts between different classes. We show experimentally that *k-means* works well for datasets with a smooth manifold structure but fails for more difficult data collections.

The second approach (Delalleau *et al.*, 2005) finds representative unlabeled data in a *greedy* fashion by repeatedly selecting the sample that is farthest from the current subset consisting of the training set L and the already selected unlabeled data Z: $\arg\min_{j \in Z \setminus S} \sum_{i \in L \cup S} W_{ij}$, with W a similarity matrix for all images using Manhattan distance and a Gaussian weighting function. This method covers the entire data space without introducing redundant information and works well as long as there are not too many outliers in the data collection.

Both methods aim to represent the entire unlabeled data space independently from the task itself. If the test set distribution correlates with the unlabeled data as it is the case for ETH (Section 3.3), these methods work well. However, when the ratio between test samples and unlabeled data is too small as it is often the case for large datasets, these approaches fail to focus on the relevant part of the distribution thus not achieving optimal performance.

7.4.2 Novel techniques to enrich graph structure

In this work, we propose two novel selection criteria called *dense* and *NN* that focus on the classification task at hand but in a completely unsupervised way. Our goal is to enrich the area around a given set T consisting of training and test data with unlabeled data. The idea behind this is that we want to benefit from unlabeled where it is most needed and helpful. Additionally, for large-scale datasets we cannot apply "the-more-data-the-better" strategy due to the time and space complexity issues.

We consider three scenarios for extension: 1) training set only; 2) test set only; and 3) training+test set. Enriching around the training set increases the likelihood that the neighborhood contains relevant images to propagate label information. If the neighborhood is sparse and not representative the labels might be propagated to samples of different classes even within one iteration. Enhancing the area around the test data improves the results for the same reasons. Experimentally we observed that enriching the neighborhood of both training and test works best so that we report only results for this setting in the following.

(i) *dense.* Our first criteria uses the previously introduced graph structure to find dense and thus representative regions. Of course, these regions can be anywhere in the unlabeled data space. Therefore, we look only in the immediate neighborhood of T for high density nodes. More specifically, we select the k nearest neighbors for each $x_i \in T$ so that we have a pool of at least $|Z_{pool}| + c$ samples with $|Z_{pool}| + c \gg |Z|$, i.e.,

$$Z_{pool} \leftarrow \{x_j\} \text{ with } x_j \text{ the k nearest neighbors of } x_i \in T. \tag{7.5}$$

We order these data points by their graph density $\mathcal{D}$ from Equation (7.4)

$$r(x_i) = m_i, \quad \text{where} \quad m_i \leq m_j \Leftrightarrow \mathcal{D}(x_i) \geq \mathcal{D}(x_j) \tag{7.6}$$

with $x_i, x_j \in Z_{pool}$. Finally, we select the first $|Z|$ data points with the smallest score $r(x_i)$,

$$Z \leftarrow \{x_i\} \text{ where } r(x_i) \text{ belongs to the } |Z| \text{ smallest scores} \tag{7.7}$$

Usually, the chosen data points are more representative for a group of samples so that propagation is more reliable. In the experimental part, we will see this positive behavior in particular for a small set of Z. The larger $|Z|$ becomes, the more redundant nodes are selected.

(ii) *NN.* Beside this positive behavior regarding our set T, this method still does not scale well to large datasets (see Section 7.4.3) as we have to calculate the entire distance matrix. For this reason, we propose a second criteria *NN* that can be seen as an approximation of *dense*. This selection technique needs only the distances between $x_i \in T$ to all unlabeled data $x_i \in U$ with $U = N \setminus T$. Usually, we have $|T| \ll |U|$ so that the runtime is moderate. To enhance T, we select the first k nearest neighbors for each $x_i \in T$, i.e.,

$$Z_{pool} \leftarrow \{x_{i_k}\} \text{ with } x_{i_k} \text{ the k nearest neighbors of } x_i \in T \tag{7.8}$$

This procedure ensures that each point in T is separately enriched. For the case that $|Z_{pool}| > |Z|$ we randomly subsample this set until we achieve our selection size $|Z|$.

7.4.3 Runtime complexity

In the following, we briefly analyze the time complexity of all introduced graph enhancement techniques and then compare their runtime behavior in the context of label propagation (see Figure 7.1). Given $|N| = |T| + |U|$ images with T the original dataset consisting of training and test set and U the pool of unexplored and unlabeled data. The runtime of *k-means* is directly linked to the number of clusters, i.e., $O(|Z||U|m)$ with $|Z|$ the number of anchor points ($\sim$ number of added data) and m the dimensionality of the image descriptor. As the unlabeled data volume increases, memory and and runtime requirements increase disproportionately as can be seen in Figure 7.1 (left).

For *greedy*, we have to compute all distances between the current point set $L \cup Z^{(t)}$ at time $1 \leq t \leq |Z|$ to all remaining unlabeled data $U \setminus Z$, i.e., $O(|Z||T||U|m)$. This iterative procedure is the most time-consuming part. Depending on the dataset size and $|Z|$, it is faster to compute the entire distance matrix once ($O(|N|^2 m)$). But for large pools of unlabeled data with more than one million data, the full matrix does not fit into memory so that we have to deal with approximations instead.

For our *dense* criteria, we require $O(|N|^2 m)$ to compute all distances and to sort these distances we need $O(|N|^2 \log(|N|)$. Graph construction and calculation of graph density is considered a linear operation. An advantage of this method is the small memory requirement because we can split $|N|$ into smaller pieces $N_i \ll |N|$ so that we need at most $N_i \times |N|$ space. Finally, we are only interested in the first k nearest neighbor, i.e., we disregard all other distances. In our case, we set $k = 1,000$. One advantage of this method in comparison to all previous methods is that we have to compute this distance matrix only once because we can reuse it for label propagation itself or for different training and test sets.

As mentioned before, *NN* serves as a good approximation of *dense*. Instead of computing the entire distance matrix over $|N|$, we only need to calculate all distances between T and all unlabeled data U. Additionally, we also have to sort T times the according distances. Finally, we get a runtime complexity of $O(|T||U|m + |T||U|\log(|U|))$.

To run LP, we have to construct the k-NN graph thus requiring $O((|T| + |Z|)^2 m)$ to compute all distances for the set $T \cup Z$, and $O((|T| + |Z|)^2 \log(|T| + |Z|))$ to sort these. LP itself needs $O((|T| + |Z|)^2 C)$ with C the number of classes. The calculation of the graph Laplacian $\mathcal{S} = D^{-1/2} W D^{-1/2}$ is fast because D is a diagonal matrix. During run time, we have an extremely sparse graph. Usually, we build a 10-NN graph structure so that we do not observe any memory problems.

Figure 7.1 visualizes on the left side the runtime of the several graph enhancement methods including the *random* baseline for the dataset IM100 introduced in the next section. This is a subset of ILSVRC 2010 with 100 classes and approx. $130,000$ images. We plot the number of added images against the expected runtime. To approximate

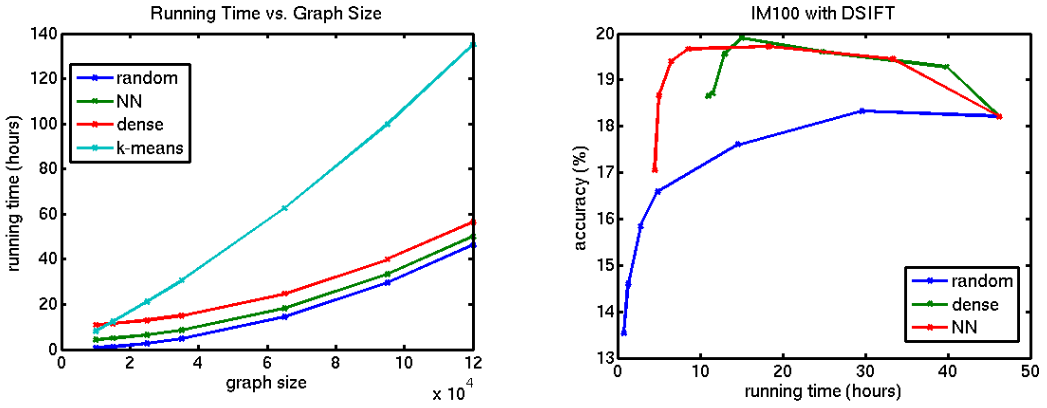


Figure 7.1. Left: Complexity for selecting $|S|$ unlabeled data $x \in U$ with m dimensions of the image descriptor given a fixed training and test set $|T|$ and label propagation. Right: Complexity against performance of IM100 (see Section 3.3) for DSIFT.

the runtime, we run one experiment 5 times under almost ideal conditions, i.e., only one process per time and scale this value to all other points in this plot given our complexity analysis. Note, the values of *k-means* are optimistic because it assumes that the algorithm converges after one iteration which is usually not the case.

Greedy is not shown in this figure because it does not fit on the y-axis: For the first point, i.e., adding 10,000 unlabeled images we need approx. 80 hours. *k-means* needs only 8 hours and is slightly faster than our *dense* criteria with 10.9 hours but slower than *NN* with 4.4 hours. To increase the dataset size by 25,000 unlabeled data points, *k-means* needs 21.1 hours while *NN* requires only 6.4 hours and *dense* needs 12.9 hours. For *random*, we would need 2.7 hours.

On the right side of Figure 7.1, we plot runtime against classification performance for the same dataset. *k-means* and *greedy* cannot be applied on this large unlabeled pool due to their time and space complexity. Most interestingly we see for a given time budget that we achieve better performance than *random*. For example if we look at 20 hours for *random* that corresponds to a graph size of 65,000 images, we get a performance of 17.6%. In contrast, *dense* and *NN* need only a graph size of 25,000 to get a higher performance with 19.9% and 19.6% respectively. This emphasizes our claim that we are not only faster but also obtain better performance with a more representative subset of the unlabeled data. Although "the-more-data-the-better" strategy actually leads to a consistent improvement (blue curve) the final performance is clearly below the results achieved with our methods (red and green curves). This loss of performance is often a consequence of added images that connect many images from different classes bringing them unintentionally close together.

7.5 EXPERIMENTS

In our experiments, we select randomly 5 training samples and 45 test samples per class that serves as the original dataset T. This setting exactly corresponds to the classical semi-supervised setting with 10% labeled data (Zhou *et al.*, 2005; Zhu *et al.*, 2003; Ebert *et al.*, 2010). The remaining images of these datasets are considered as the data pool U from which we select unlabeled data to enrich T. We run all experiments 5 times with 5 different sets T and evaluate the performance on the test set only. Therefore, we are able to compare our results independently from the amount of added data. In the following, we analyze each dataset separately.

ETH80. Figure 7.2 shows for all three image descriptors graph quality (GQ, first row) and accuracy after label propagation (second row) without (solid lines) and with (dashed lines) active learning (AL). Graph quality denotes the average number of correct nearest neighbors in our symmetric k-NN graph structure for the training and test data and serves only as a theoretic measure as we need to know all labels for this evaluation. For AL, we start with one training example per class randomly selected from our fixed training set of 5 samples per class, and request in average 4 labels per class from the remaining training set plus the additional unlabeled set.

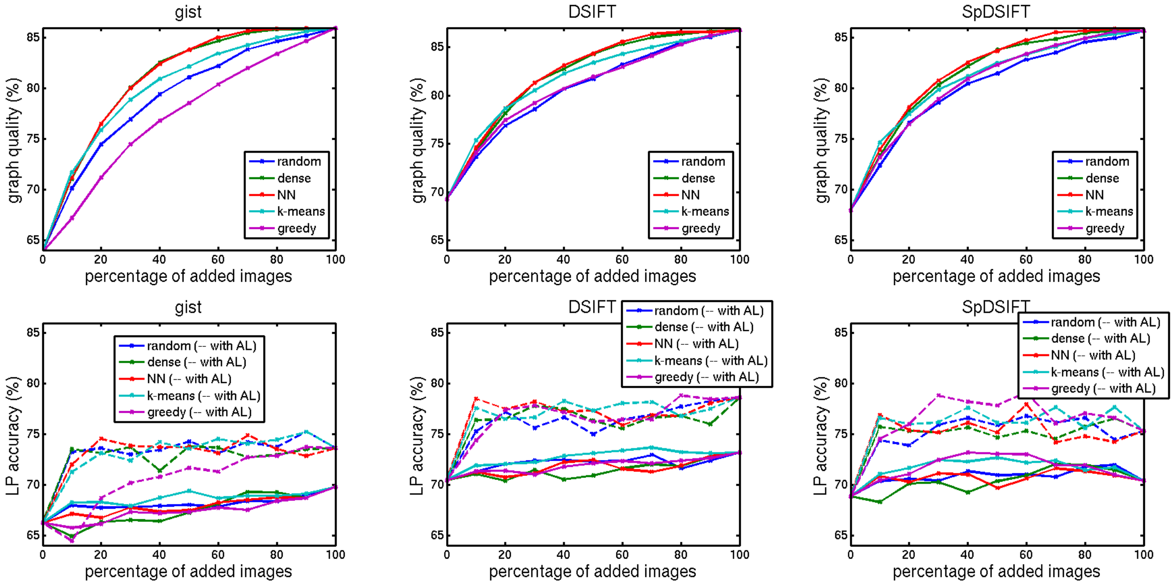


Figure 7.2. Graph quality (first row) and LP accuracy (second row) for ETH80 with (dashed lines) and without (solid lines) active learning for different number of added images: Gist (left), dense SIFT (middle), and spatial dense SIFT (right)

We observe that the graph quality starts saturating after 60% to 70% added data. The performance of all selection methods is similar including the *random* baseline. This can be explained by the smooth manifold structure of the dataset. There are almost no outliers in this dataset so that our test set benefits from almost all images equally. For LP, we see a consistent improvement when active learning is used.

As this is true also for all other datasets we show only the performance for active learning in the following. Table 7.1 shows graph quality (GQ) and accuracies with 50% (≈ 1500) additional unlabeled data. For DSIFT with *NN* selection we improve LP without AL from 72.4% to 77.3% with AL. *k-means* performs slightly better for LP without AL. The cluster centers seem to be good anchor points for the test data. Our density selection criteria shows on average slightly worse performance for LP without AL probably due to the oversampling of dense regions (e.g. apples and tomatoes are high density regions which are preferred by this criteria).

	Gist			DSIFT			SpDSIFT		
method	GQ	LP	+AL	GQ	LP	+AL	GQ	LP	+AL
random	81.5	68.0	**74.3**	82.1	72.33	75.0	82.3	70.9	75.9
dense	83.0	67.2	73.8	84.1	70.9	76.3	83.0	70.3	74.7
NN	**83.3**	67.4	73.7	**84.1**	72.4	**77.3**	**83.5**	69.7	75.2
k-means	82.5	**69.4**	73.6	83.6	**73.1**	**77.3**	82.9	72.7	76.1
greedy	78.1	67.3	71.7	81.7	72.1	76.2	82.2	**73.1**	**77.8**

Table 7.1. Graph quality (GQ) and LP accuracy without and with (+AL) active learning for ETH80 after adding 50% unlabeled images.

C-PASCAL. This dataset corresponds to a more difficult classification problem with many outliers and overlapping classes. We observe for both GQ and LP (Figure 7.3) a large performance gap between our selection methods and previous methods. For SpDSIFT and DSIFT, *k-means* and *greedy* are even worse than the random baseline, e.g., LP+AL decreases for SpDSIFT from 28.3% with *random* to 20.1% with *k-means*, and to 23.5% with *greedy*. For *k-means*, this drop is a direct consequence of the used cluster centroids. Many clusters contain more than one class so that these clusters connect all examples of those classes and bring them closer together. In contrast, *greedy* focus more on outliers.

	Gist			DSIFT			SpDSIFT		
method	GQ	LP	+AL	GQ	LP	+AL	GQ	LP	+AL
random	21.1	**21.1**	21.4	21.7	19.0	21.8	28.9	27.3	28.3
dense	23.8	20.8	22.1	**26.1**	**20.3**	**24.3**	**33.4**	29.0	32.2
NN	**23.9**	20.9	**22.7**	25.9	20.0	24.0	33.1	**29.0**	**32.9**
k-means	20.5	20.8	21.6	21.6	19.1	21.2	24.0	25.0	20.1
greedy	19.4	20.6	21.3	20.1	19.8	19.5	25.4	26.2	23.5

Table 7.2. Graph quality (GQ) and LP accuracy without and with (+AL) active learning for C-PASCAL after adding 50% unlabeled images.

NN and *dense* perform similarly well. Furthermore, we observe a decrease in graph quality as well as LP accuracy when using all unlabeled data. For SpDISFT, we get best performance for 50% (≈ 4,600) added images with 33.4% GQ, and 29.0% LP accuracy. These values drop to 29.8% GQ and 28.4% LP+AL when using all data. This is an important insight because it demonstrates that there is no need to use an arbitrary large number of unlabeled data. As a consequence we are able to reduce the amount of unlabeled data drastically without loss of performance. Note, the decrease of the GQ is a side effect of the symmetric graph structure. The more data the more unrelated samples connect to our training and test data. Although the graph quality of a non-symmetric graph shows better performance, label propagating through this graph structure consistently leads to worse results (up to 5%).

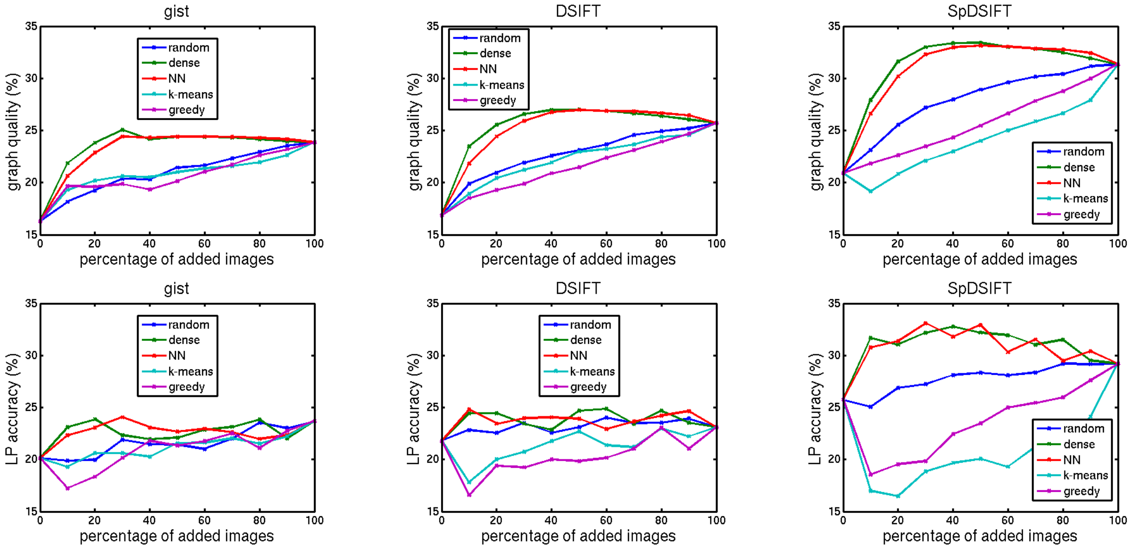


Figure 7.3. Graph quality (first row) and LP accuracy (second row) for C-PASCAL with (dashed lines) and without (solid lines) active learning for different number of added images: Gist (left), dense SIFT (middle), and spatial dense SIFT (right)

IM100. In the following, we analyze a subset of ILSVRC 2010 with approx. 130,000 images. This subset is large enough to increase the amount of unlabeled data by a factor of 25 but also small enough to run SSL on the entire dataset. *k-means* and *greedy* cannot be applied to this dataset due to their time and space complexities (see Section 7.4). Similar to all previous subsections, we show GQ and LP+AL in Figure 7.4 for different numbers of added data (graph size), and Table 7.3 contains results when adding 20% unlabeled data.

Again, we observe a significant improvement of our selection methods over *random*. For SpDSIFT, we increase GQ from 20.4% with *random* to 30.5% with *dense* and to 30.2% with *NN*, and LP+AL from 21.1% to 27.0%. Similar to C-PASCAL, our performance is with 20% to 30% additional data better than using all unlabeled data. For SpDSIFT, we observe a decrease of GQ from 31.2% with *dense* and 30% unlabeled data to 27.6% with all data.

	Gist			DSIFT			SpDSIFT		
method	GQ	LP	+AL	GQ	LP	+AL	GQ	LP	+AL
random	15.7	11.6	14.9	17.0	12.2	16.6	20.4	16.4	21.1
dense	**23.2**	12.6	**17.7**	24.0	**13.0**	**19.9**	**30.5**	17.9	**27.0**
NN	22.0	**12.7**	17.3	**24.1**	**13.0**	19.7	30.2	**18.0**	26.2

Table 7.3. Graph quality (GQ) and LP accuracy without and with (+AL) active learning for IM100 after adding 30,000 unlabeled images (≈ 23%).

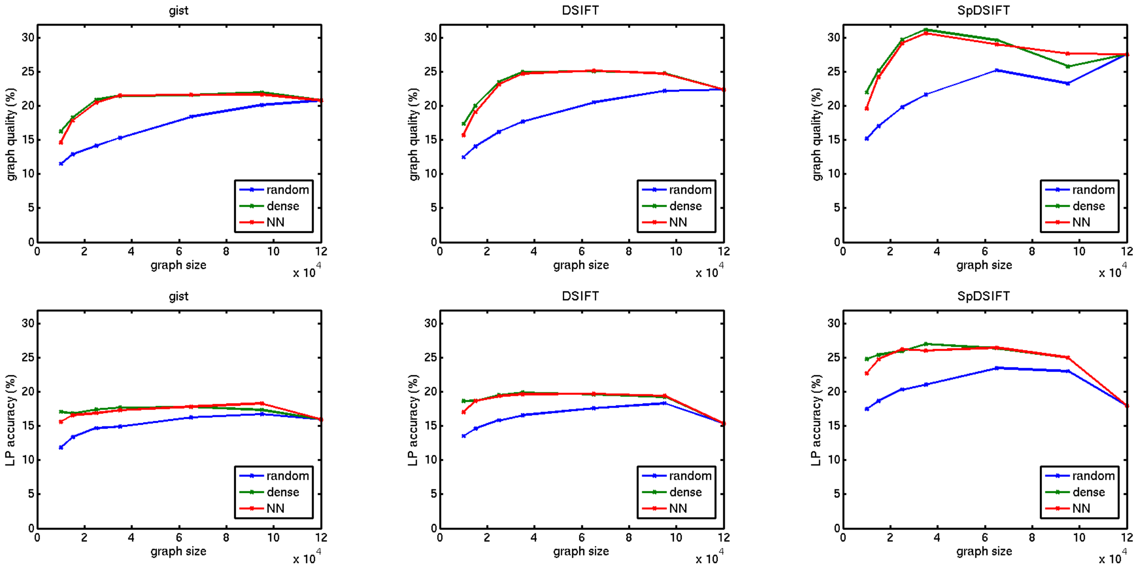


Figure 7.4. Graph quality (first row) and LP accuracy (second row) for IM100 with active learning for different number of added images: Gist (left), dense SIFT (middle), and spatial dense SIFT (right)

ILSVRC 2010. Finally, we run LP on the entire ILSVRC 2010 challenge with 1,000 classes. We start with our set T given by 5 training samples and 45 test sample per class, i.e., 50,000 images (Table 7.4, first line). After that, we continuously add 50,000 unlabeled data from the pool of the remaining 1.2 million images. Table 7.4 shows graph quality (GQ), top 1, and top 5 accuracy for LP+AL and the difference to *random* selection. For computational reason, we apply only *NN*. To further increase the speed of AL, we use batch active learning with a batch size of 100 labels per query. So that we request 400 times a batch of 100 labels to get in average 5 labels per class.

For comparison, we run also a linear SVM on the base setting with 50,000 images and with different parameters. The best performance we observe is 0.22% averaged over 5 different runs. In contrast with LP without enrichment we get 2.8% top 1 accuracy. This large difference can be explained by the additional graph structure we used in SSL. According to the selection criteria, we improve increasingly our graph quality (GQ). For 50,000 additional unlabeled images we note a difference between *random* and *NN* of +1.7% while for 250,000 added images this difference increase to +4.4%. We also observe an improvement for LP. For 150,000 additional images, we

	random			NN selection					
added data	GQ	top 1	top 5	GQ	diff	top 1	diff	top 5	diff
0	2.4	2.8	7.1	2.4		2.8		7.1	
50,000	3.5	3.9	8.4	5.3	+1.7	5.0	+1.2	9.4	+1.0
100,000	4.3	4.1	8.7	7.2	+2.9	5.4	+1.3	9.7	+1.0
150,000	4.8	4.2	8.8	8.5	+3.7	5.5	+1.3	9.9	+1.1
200,000	5.3	4.5	9.0	9.5	+4.1	5.7	+1.2	10.0	+1.0
250,000	5.8	4.5	9.1	10.1	+4.4	5.7	+1.2	10.0	+1.0

Table 7.4. ILSVRC 2010 with *random* and *NN* enrichment for DSIFT: graph quality (GQ), top 1 and top 5 accuracy after LP with AL, and the difference to *random*.

increase LP from 4.2% with *random* to 5.5% with *NN*. However, LP benefits only in a limited way from this improved structure. One explanation might be that we run a batch AL instead of a single AL. Usually these batch AL show worse performance in comparison to single AL.

7.6 CONCLUSION

In this work, we enhance the graph structure for graph-based algorithms with more unlabeled data and address the scalability of these approaches. These algorithms come with a quadratic runtime so that "the-more-data-the-better" strategy does not scale to large datasets like ILSVRC 2010 with 1,000 classes and over a million of images. We propose two selection criteria for enriching a dataset and to improve the graph structure. These criteria drastically reduce the amount of unlabeled data in comparison to the "the-more-data-the-better" strategy while still achieving better performance than using all unlabeled data. Moreover, given a fixed time budget we show significant improvements on four different datasets with less unlabeled data in contrast to previous approaches.

	gist		DSIFT		SpDSIFT	
	acc	gain	acc	gain	acc	gain
LP	11.2		11.4		14.7	
+25,000 random	11.8	+0.6	12.2	+0.8	16.6	+2.0
+AL	14.5	+3.3	16.3	+4.8	21.4	+6.7
+25,000 NN	12.2	+1.0	13.0	+1.6	17.8	+3.1
+AL	**17.9**	**+6.8**	**19.7**	**+8.3**	**26.3**	**+11.6**
using all data	12.4	+1.2	12.8	+1.4	16.7	+2.0
+AL	16.3	+5.1	16.1	+4.7	20.6	+5.9

Table 7.5. IM100: baseline (5 training + 45 test images per class), 25,000 randomly added data without and with AL (row 2-3), with 25,000 *NN* selections without and with AL (row 4-5), and using all unlabeled data without and with AL (row 6-7).

Table 7.5 summarizes our main findings from this work on the dataset IM100. First of all, we see a consistent improvement when adding more unlabeled data. For SpDSIFT, we increase from 14.7% to 16.6% with randomly added $25,000$ unlabeled data points to finally 16.7% when adding all available data. But these results are clearly below the performance of 17.8% that we achieve with our novel criteria *NN*. This fact becomes even more obvious in combination with active learning where we improve SpDSIFT with our new criteria by 11.6% to 26.3% while we increase this performance only by 5.9% when applying "the-more-data-the-better" strategy.

This summary shows once more that a careful selection of unlabeled data leads to better results as well as to a more compact graph that scales also to large datasets such as the complete ILSVRC 2010 dataset containing over a million images.

CONCLUSION AND FUTURE PERSPECTIVES 8

Contents

8.1 Discussion of contributions **113**
8.1.1 Contributions towards better structure 114
8.1.2 Contributions towards better labels 115
8.2 Future perspective **116**
8.2.1 What can we learn from human object recognition? 116
8.2.2 Beyond human perception, learning, and inference 123

Based on the observation that fully supervised datasets are difficult to collect and to control but internet sources provide us an enormous abundance of unlabeled data, we investigate semi-supervised learning on the task of image classification. This enables us new perspectives to tackle computer vision problems. We are no longer dependent on the existing datasets that are either small and too specific for one problem setting (Ponce *et al.*, 2006; Torralba, 2011) or larger and error-prone due to the outsourced annotation process (Welinder *et al.*, 2010). The few labels available during the learning help us to guide the learning procedure and to evaluate the performance of these algorithms.

We identify two main sources of challenges that have an high impact on the final classification performance: structure and supervision. Both issues are addressed in this thesis. In the following, we discuss our contributions towards a better structure in Section 8.1.1 and towards better labels in Section 8.1.2. After that, we conclude this thesis by pointing to future directions in Section 8.2. We divide them into ideas that directly result from the individual chapters of this thesis with looking at human object recognition (Section 8.2.1) and ideas that go beyond human abilities (Section 8.2.2).

8.1 DISCUSSION OF CONTRIBUTIONS

As mentioned before, we recognize a strong dependency between the quality of the structure and the supervision on one side and the classification performance on the other side. Assuming there is only one concept per class, the quality of the structure is more important than the quality of the labels. In this case, all images of one class are mapped to one densely connected region that is separated from all other classes. Representative labels are only needed to speed up the labeling process because the accuracy is always 100%. But these ideal conditions cannot be met by state-of-the-art datasets and methods for image classification. Therefore, we have to deal with both structure (Section 8.1.1) and supervision (Section 8.1.2) at the moment. Our contributions to these parts are briefly summarized in the following two subsections.

8.1.1 Contributions towards better structure

At the beginning of this thesis, there was only a side note in Zhu (2006) stating that structure is more important than the algorithm itself. Even though this was a plausible statement, it was still difficult to imagine the extent of this claim without any proof or extensive evaluation. Therefore, we provided a comprehensive study on different algorithms, distance measures, image descriptors, and graph construction methods in Chapter 3. We focused on graph-based algorithm as there is a strong correlation between structure and performance that makes it easy to verify this statement. We have shown that with a good parameter setting the differences among algorithms are marginal – for some datasets smaller than 0.5%. But the discrepancy caused by the underlying graph structures were remarkable and in some cases up to 24.1%. This encouraged us to continue research in this direction. In the following, we located three main sources that have an impact on the structure: data, image descriptor, and distance measure.

Data are the most important factor since they represent the knowledge base for our classifier. The more gaps are in the dataset in terms of missing viewpoints, appearances, or variations, the lower is the generalization of the learner. We addressed this critical point twice. Firstly, we augmented our datasets by flipping the images (Section 4.2.2) to enlarge the space of possible orientations of an object. Especially global image descriptors like HOG benefited from this step. Secondly and more importantly, we have questioned the common sense "the-more-data-the-better"-strategy. We proposed two selection methods in Chapter 7 that are able to process over one million images within few hours to find a representative subset that improves significantly the graph structure. Furthermore, we achieved with this approach a higher performance than using all data. Two crucial insights emerged from this observation. Large collections of data contain many noisy images that negatively impact the final structure. From that it follows immediately, there is no need to use all images which is advantageous in terms of runtime and space complexity.

Image description is the next important ingredient for a good structure. As we already mentioned in Section 1.2.1, state-of-the-art descriptors have several issues. One of those is, that they usually capture only one aspect in an image such as shape, color, or texture. To overcome this limitation, we investigated several combination strategies in Section 4.2.3 and showed a significant boost in performance.

Finally, distance measures are used to express the similarity between images and to find the nearest neighbors for each image. An ideal distance measure should produce small intra-class distances and large inter-class distances. But measures such as Euclidean distance cannot deal with high-dimensional features as it is often the case in computer vision. In this thesis, we tackled this problem by using metric learning. In Section 4.3.3, we integrated the information-theoretic metric learning (ITML) proposed by Davis *et al.* (2007) into our graph construction procedure. We discovered a tight relation between labels for metric learning and the resulting graph quality that was often attended by an over-fitting of the labeled data. Therefore, we proposed in Section 4.4 an interleaved metric learning and label propagation framework that integrates with each iteration more and more unlabeled data with their predictions leading to state-of-the-art performance on Caltech 101. Additionally, we introduced in Chapter 6 an active metric learning procedure that uses active learning to get more representative labels for metric learning that further improves the quality of our graph structure.

8.1.2 Contributions towards better labels

During our investigations of different graph structures and graph improvements, we also observed an high correlation between labels and classification performance in Chapter 5. For this reason, we looked at active learning strategies to get more representative labels for propagation. We explored several sampling strategies from pure exploitation-driven criteria that query the most uncertain samples to pure exploration-driven criteria that aim to capture the entire data distribution. We soon found out that a combination of both strategies is essential. But there is no single or combined sampling strategy that works best across different datasets. Most previously proposed methods assume that each dataset needs exploration at the beginning and exploitation at the end (Nguyen and Smeulders, 2004; Cebron and Berthold, 2006). For computer vision, this observation does not hold because some datasets need only exploitation, other datasets need a constant trade-off between exploration and exploitation, and some datasets require the opposite strategy, i.e., exploitation at the beginning to get better prediction values and exploration at the end to explore also unobserved areas.

As a consequence, we introduced in Section 5.4 a meta learning framework that adapts the sampling strategy to each dataset individually during the sampling itself. We considered active learning as a Markov decision process (MDP) that gives us the full flexibility to combine more than two criteria with arbitrary trade-offs also allowing for a time-varying trade-off. We solved this MDP with reinforcement learning by using the overall entropy as feedback to get a time-varying trade-off. Additionally, we proposed a novel exploration sampling criteria called *graph density* that considers the k nearest neighbor structure to find densely connected regions. This criteria performs significantly better than other exploration criteria for label propagation but also for SVM and k nearest neighbor classifier Section 6.4.2.

8.2 FUTURE PERSPECTIVE

In this last section, we will mention open issues of this thesis and how we could tackle those in Section 8.2.1. Most of these suggestions arise from cognitive science for the simple reason that particularly object recognition raises automatically the question how humans form class concepts. Thus it is not surprising that this close relationship has been studied earlier, e.g., in the *Roadmap of Cognitive Vision* (Vernon, 2005). Of course, the human object recognition should be only seen as a good starting point to improve on. Therefore, we finally discuss in Section 8.2.2 also the human weaknesses in perception, learning and inference and how we might overcome those with machine learning and computer vision.

8.2.1 What can we learn from human object recognition?

Following the general structure of this thesis, we discuss open issues of each of the components of SSL separately, i.e. i) data, ii) image description, iii) similarity notion, and iv) supervision. Additionally, we also challenge the use of label propagation in the last subsection called v) exemplar-based vs. concept-driven learning.

i) Data. Zhu *et al.* (2009) stated in their position paper about graph-based SSL that one of the limitations of these algorithms results from the common sense assumption that each class can be projected to a single manifold. Thus they suggest to model classes by a mixture of multiple manifolds. This observation is particularly in computer vision not novel. Also Schiele and Crowley (1997) distinguish between visual classes and object classes as there are classes such as *chair* that cannot be modeled with one mixture. Even if this is an important aspect, an at least equally important problem is the incompleteness of today's datasets in terms of viewpoints and variations for a class to fully leverage the power of SSL algorithms. In this thesis, we assume that most classes can be described with one concept and all exemplars of these classes are grouped around this concept (Cohen and Murphy, 1984). But often we have to deal with class descriptions as shown in Figure 8.1 for the class *kingfisher*. Even for a human who have never seen this class before might have problems to merge these images in one compact mixture because color, shape, and appearance are quite different. But an image sequence might provide a path among those images as visualized in Figure 8.2 and helps to extract an object models similar to the work of (Schiele, 2000) shown for car tracking.

In this work, we analyze the problem of incomplete datasets by looking at existing datasets such as ImageNet with more than one million images and their behavior and performance when adding more unlabeled data. But this should be only considered as a first attempt towards a smooth manifold structure. As a next step we have to get away from this controlled setting used in Ebert *et al.* (2012c) because we still depend on the quality of the datasets given by the limited (although larger) amount of images and the quality of labels. Instead, we have to tap into other sources.

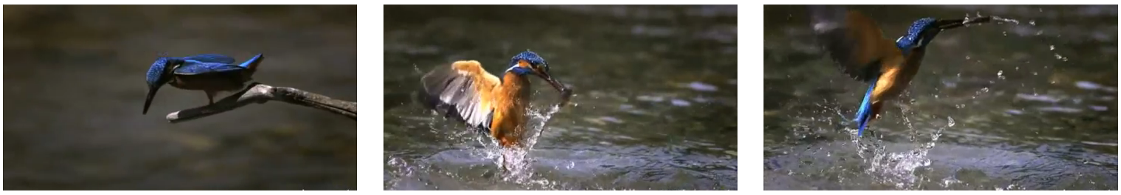


Figure 8.1. Visualization of an incomplete object class description that makes it difficult to find relation between these images.

In general, there are three possibilities: 1) combining several datasets; 2) adding synthetic data, or 3) browsing the internet.

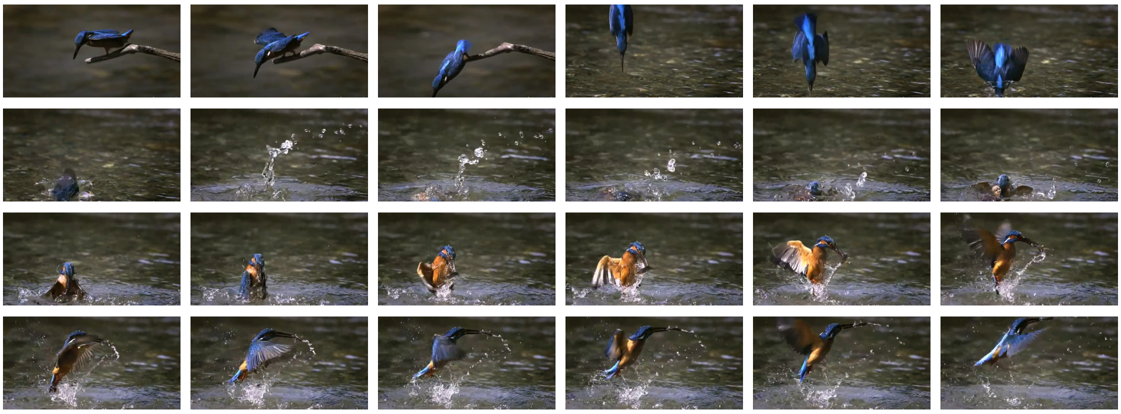


Figure 8.2. Visualization of a more complete object class description extracted from a video sequence.

Merging different datasets is problematic because most of the available datasets have an inherent bias attached to the dataset (Ponce *et al.*, 2006; Torralba, 2011). Although there are works that try to undo the damage of dataset by estimating the bias for each dataset (Khosla *et al.*, 2012), combining itself seems an unsatisfactory strategy because each dataset has different classes and the amount of images is also strongly limited. In contrast, adding synthetic data is a more promising direction but is still in a early stage of development. There are only few works that either generate new training images (Pishchulin *et al.*, 2012; Li and Fritz, 2012), or add synthetic data points (so-called *ghost points*) in the distance space itself (Chawla *et al.*, 2002; Yang *et al.*, 2012b). The former approach is currently bound to certain classes such as *people* for where 3D shape models exist that ensure the generation of feasible poses and shape variations. The latter one is hard to control because the semantic meaning of these *ghost points* is not clear and it can lead to a blending of different classes. To expand approaches suggested by Pishchulin *et al.* (2012) to other classes, we have to integrate more physical constraints to guarantee feasible poses and appearances of the object. Figure 8.3 show some images of the class *anteater*. A human does not necessarily have to know and to observe this animal to decide if a pose is likely or not. Informations such as size of $1.5 - 1.8$ m or weight of ≈ 60 kg might be already a good advice.

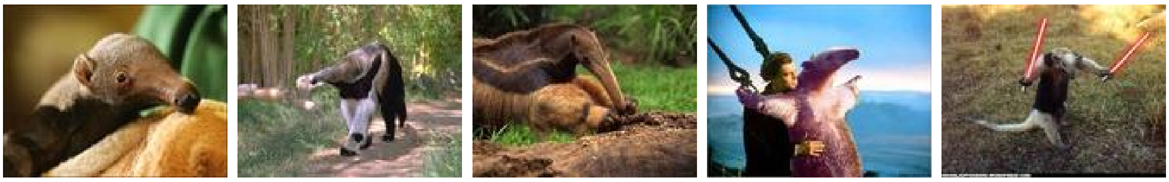


Figure 8.3. Visualization of feasible and unfeasible poses and scenes for the class *anteater* that will be more obvious with informations such as $1.5 - 1.8$m long.

Another limitation comes with the fixed pool of unlabeled data in particular if we use existing datasets. This is similar to use the knowledge of a child for our entire life without any update. But in fact, our knowledge base will be permanently updated. That is why children regard a lost rabbit in a magic show as the reality while an adult knows that this can be only a trick. However, the internet provides us a large amount of images as well as videos and it is steadily updated with new data. A transformation of current SSL algorithms into on-line learning algorithms might be necessary (Grabner *et al.*, 2008; Saffari *et al.*, 2010; Sternig *et al.*, 2012) to benefit from these changes. Nevertheless, tapping into this data source poses many problems and questions: How do we get representative samples for each class out of these large amounts of data? How should we deal with incorrect tags? Do we get enough images for each class, e.g., endangered animals/plants or deep sea fish? But even the first question is of great importance if we look at the first examples of the query *fish* in Google (Figure 8.4) that contains drawings, a robot fish (3rd image), body paintings and other atypical examples.

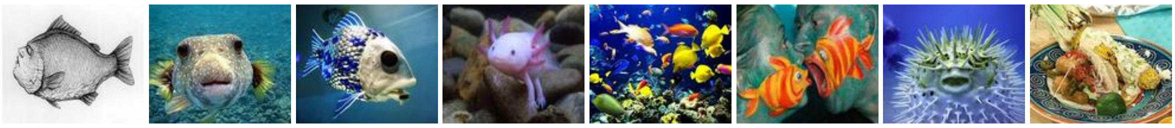


Figure 8.4. First examples for the query *fish* in google images (out of 1.5 billion results) that are not representative for this class.

ii) Image description. Missing data is clearly not the only bottleneck that can be seen in our experiments for ETH-80. This dataset is well suited for semi-supervised learning because each object is photographed from different viewpoints and there is no background clutter, occlusion, or truncation of the object. But in our experiments we achieve at most 80% with 5 randomly drawn labels per class and a combination of three different descriptors. Figure 8.5 shows the most confusing classes for this dataset with the corresponding binary masks, i.e., tomatoes are mixed up with apples and the animals (cow, dog, horse) are confused with each other. By using also a color descriptor, we are able to distinguish green apples and red tomatoes but the final improvement is only minor.

This poses many questions. Which information do we miss? Do we need a better texture description to distinguish the different surfaces of tomato and apple? Why

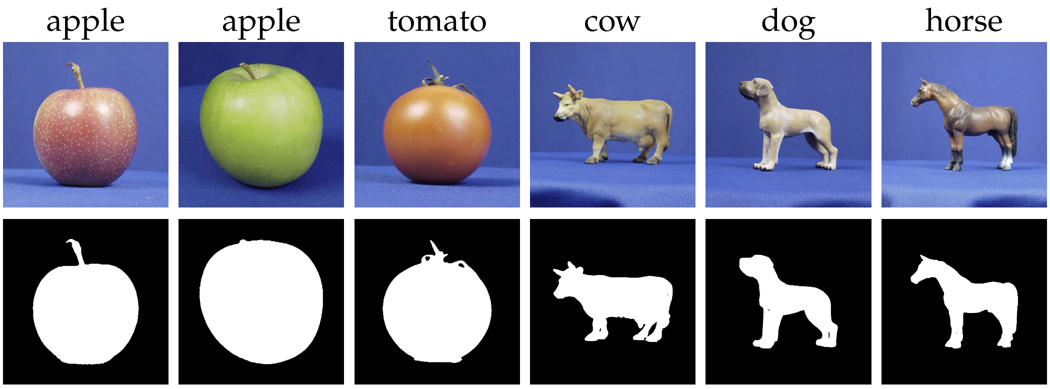


Figure 8.5. Most confusing classes for ETH-80 although the conditions are optimal for SSL, i.e. smooth manifold structure and no background clutter.

can a human easily guess the object class for the animals by only looking at the binary masks in the second row in Figure 8.5? Do we describe a concept of a class in terms of proportions? It seems obvious that a better shape description is needed. But it is still not clear how to extract, to store and to use this structural description. In Schiele and Pentland (1999) they show that only 15% to 30% of an object is required to recognize 100 classes correctly independent of the orientation and the view point of the object. But many of today's shape descriptors lack on this generalization. They extract often too detailed and too specific contours of an object and store this description as a template that will be later used for classification by matching. Although this is a good starting point, this approach needs too many templates to perform reasonably well. One promising direction is to use 3D information (Stark *et al.*, 2010; Pepik *et al.*, 2012) that allows to extract structural informations about the object and to make assumptions about unseen parts in the image. In these mentioned works, they use a CAD model of an object to get this information. In general, this information is difficult to get for the most object classes that we tackle in this work. Furthermore, they consider the detection of the object also as a matching problem. But ideally we would use these rich 3D models to extract structural invariances for a set of viewpoints to get a more general description assuming that we need only a representative set of salient points to maximize the discrimination between objects (Schiele and Crowley, 1996).

Apparently, there is a similar discussion in cognitive science (Hayward, 2003). Supporter of the viewpoint-based model theory believe that we can extract all information from different viewpoints assuming infinite many viewpoints, i.e., templates, that would clearly exceed our brain capacity. In contrast, advocates of the structural description models argue that each object can be explained by viewpoint-independent invariances. For some classes this assumption might be true such as *bottle* or *orange*. But for many other classes it will be difficult to find such invariances over all viewpoints, e.g., the table legs are invisible in the top view. More recently, there is a common agreement that we need both templates for completeness and structural description for generalization. But this trade-off is currently missing in computer vision. Thus, we need a set of representative and most discriminative viewpoints (Schiele and Crowley, 1998) as shown for the class *armadillo* in Figure 8.6 but we also need a more general description, e.g., properties, and proportions that are invariant over a set of different viewpoints.

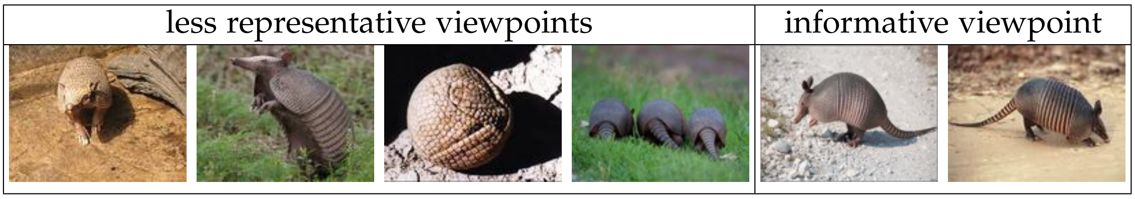


Figure 8.6. Representativeness of viewpoints for the class *armadillo*: images on the left side show less representative viewpoints as they miss important properties of this species while the viewpoints in the right images are most informative for this class.

Another issue in our experimental setting is that we compute the image descriptor on the entire image. In fact, this is a fast and simple way of extraction but it is not clear whether this is an advantage due to the additional context information or a disadvantage because of the background clutter. But this could be analyzed by using for example part-based models (Felzenszwalb *et al.*, 2010), or foreground extraction Lee and Grauman (2009).

Finally, we also miss prior knowledge and context to speed up and enhance image description similar to the work of Fussenegger *et al.* (2006) for image segmentation. Murphy and Allopenna (1994) show that humans learn three times faster and more accurate if the features of an object were related to each other. A human learns even more although this additional knowledge is not strictly necessary for accurate performance (Kaplan and Murphy, 2000). Thus, there is obviously a strong correlation between associations among features and final performance, e.g. animals with feathers are more likely to have also wings in comparison to animals with fur. Equally important is the grouping of features or objects otherwise it would be impossible to follow a soccer match if the player do not wear an uniform. But in many applications including this thesis, the raw image description is fed directly into the classifier without any intermediate steps such as grouping, ranking, or finding associations. This is rather disappointing because we cannot really reconstruct and understand what went wrong during the classification. Apart from that, some categories are almost only defined by their function, e.g. *chair*. Thus, to boost the recognition of those classes, we need associations for example with human poses as shown in Delaitre *et al.* (2012).

iii) Similarity notion. Encouraged from the positive results of previous metric learning literature, we integrate several of those methods in the graph construction procedure. But the outcome did not meet our expectations. The main reason is that previous work almost exclusively compare their methods to the Euclidean distance (L2). In this work, we also observe a larger improvements for the L2 distance but these final numbers are lower than just applying Manhattan (L1) distance. In fact, it is almost impossible to improve L1 distance with any metric learning procedure. PCA decrease the performance of L1 and also the most supervised metric learning approaches decrease the performance or do not have any effect. Only with ITML (Davis *et al.*, 2007), we observe a small improvement of approximately 1.5%. But this benefit seems rather out of proportion if we consider the runtime and the tedious parameter search.

One problem is that the supervised approaches tend to over-fit due to the small amount of labeled data. Lu *et al.* (2009) addresses this problem by including the geometry of the entire dataset as an additional regularization parameter. But this geometry is not updated during the learning that strongly limits the outcome of this algorithm. In principle, any change of the metric space should also cause a change in the geometry of the data. We tackled this issue by using an interleaved procedure that integrates successively unlabeled data with their highest prediction values. This method works fine for datasets with an already high graph quality. Otherwise the predictions are often incorrect so that the algorithm drifts to a worse solution. Additionally, we cannot control the label distribution leading to an unbalanced metric learning as some classes are more often requested than other classes. To further improve this approach, we have to incorporate a balancing factor and we should find a way to adjust and update the predictions.

In the long term, we require also different models and levels of granularity to express the similarity between objects. The properties and the description is completely different between base categories such as *cat* and *dog* and two species of the base category *dog*. Also Rosch *et al.* (1976) argue that basic level categories carry most information of a category and the categorization of objects into sub- or super-categories takes usually longer than the assignment of a base level category because super-classes ask for a generalization and sub-classes need a specification. Therefore, it is not surprising that many learning algorithms do not improve their performance when using also a hierarchy for learning as shown in Rohrbach *et al.* (2011) because they assume always the same level of similarity description. A better approach would be to start with base level categories (mid-level of a hierarchy) and to switch the strategies when learning super- and sub-classes. The general benefit of a hierarchy should be more obvious as it allows to structure our data. Another important issue might be to integrate also relations into the similarity notion such as *larger head*, *more compact body*, *thinner legs* similar to the work of Parikh and Grauman (2011) that use relative attributes.

Finally, we also need a better visualization of the resulting graph structure. Bischof *et al.* (1992a) visualized in their work a neural network to answer the questions what has the network learned and how is the knowledge represented inside this network. For graph structures, similar questions cannot be answered or only insufficiently. In this thesis, we look usually at the next nearest neighbors. But this is only one aspect of structure. It does not reflect the interactions in the entire graph. The shortest path between two nodes might be an interesting information. But usually this does not offer any valuable clue to the graph structure as the average shortest path length is ≈ 2 due to the previously mentioned *hub* nodes (Luxburg *et al.*, 2010). Also information visualization strategies such as multidimensional scaling do not produce revealing results.

iv) Supervision. We improve the quality of labels with active learning. This is a promising direction and should be always considered within semi-supervised learning due to the small amount of labels and the stronger dependency of the quality of those. However, our model, that automatically estimates the trade-off between exploration and exploitation and combines more than two criteria, has still some open issues. The trade-off is modeled with discrete states and not continuously. The feedback given by the overall entropy might be unreliable. The number of parameters is in comparison to previous work (Baram *et al.*, 2004; Osugi *et al.*, 2005) smaller but still to high. Finally, the initialization for this reinforcement learning is difficult and time-consuming as we start with no prior knowledge. Thus, one improvement could be the integration of domain knowledge or by using counterexamples (Cebron *et al.*, 2012).

v) Concept-driven vs. exemplar-based learning. In this work, we focused on graph-based algorithms as we were mainly interested in exploring the correlation between structure and classification performance. These algorithms reflect more or less the exemplar-based theory in cognition (Medin and Schaffer, 1978; Nosofsky, 1984; Kruschke, 1992) assuming that humans store a list of exemplars and use a nearest neighbor approach to categorize objects. But this theory seems inconsistent as Posner *et al.* (1967) and Zaki and Nosofsky (2007) show that people abstract to prototypes sometimes even without seeing those (Minda and Smith, 2001). Thus, we possess a generalization ability from which the used algorithms are far away. In this thesis, we approach this problem by combining label propagation with some prototype-based methods. Metric learning transforms the data space such that classes are more compact and in Ebert *et al.* (2012c) we add prototypical unlabeled examples.

Although these approaches are a step in the right direction, they still miss a notion of the concept that is flexible enough to classify also unseen constellations and appearances of one object. Concepts allows us to go beyond the information given or visible (Smith and Medin, 1981), e.g. if a human knows that an object is an apple then he also knows that there is most likely a core inside. This leads to one of the fundamental questions: "What makes a category seem coherent?" that is not yet satisfactory answered. Murphy and Medin (1985) argue that similarity alone is not sufficient to describe a concept. We need also feature correlations, a structure of the attributes that are internal to a concept, and background knowledge as already discussed in the previous subsections. Beyond that we also require a relation of the concepts to each other. One possibility to get away from this purely similarity-driven approach of label propagation is to consider groups of images instead of pairwise similarities.

From the theoretical point of view, the graph structure itself leads to a discrete normalized optimization problem that can be quite loose in comparison to the continuous one. Therefore, Hein and Bühler (2010) rewrite the original optimization problem into a continuous optimization problem that produces improved graph cuts for spectral clustering. Moreover, they provide in Hein and Setzer (2011) a quality guarantee that their approach always outputs at least as good or better partitions than previous clustering methods. Even though this is an impressive result and might help to improve graph-based SSL, it is still unclear how to incorporate this insight into current algorithms.

8.2.2 Beyond human perception, learning, and inference

In the previous subsection, we discussed some future work strongly based on the insight of cognitive science. This focus on human object recognition might serve as a good starting point. But also human perception and inference has their weaknesses that might be tackled by computers. One of these shortcomings is the selective attention also known as *change blindness*. There are several studies showing that a human does not recognize large differences such as a complete different clothing of a person in a video sequence of the same situation when focusing on the conversation (Levin and Simons, 1997). In Simons and Levin (1998) one person is exchanged by another while the other person explains the direction without noticing the exchange. Most famous is the *invisible gorilla* (Chabris and Simons, 2010) that runs through a video sequences and most people overlook this disguised person. But 78% of the people are sure to recognize unexpected objects (Simons and Chabris, 1999) that is also called *memory illusion* meaning that we have the feeling of continuous attention because we cannot remember the unconscious moments. In this point, computers are trustworthy and this is one reason why most of the assembly line work or other production steps are done by a machine. Also in computer vision we can benefit from this advantage by completely analyzing video sequences (not partially like a human) or by scanning through millions of images to find prototypical examples of one class.

Another problem comes with the limited knowledge base of a human. Even if a person learns day and night, he will never be capable to acquire the entire knowledge and experience existing in our world. Also in the case that we bound this knowledge to a particular area for example a lawyer who read all cases to his topic or a doctor who is specialized to one organ. We cannot be sure that this specialist will remember the appropriate precedent or the disease pattern if it is needed. In contrast, with a computer we are able to get more information at the same time and to remind humans on the existence of some facts, e.g. to assist the diagnosis. This ability is also in computer vision of great importance as we can acquire more and better knowledge from the internet that might be helpful for semi-supervised learning.

Finally, also human inference is highly dependent on the knowledge of a person. Sure we infer quickly the position of a glass and can grasp it within few seconds and we immediately recognize the *Wolpertinger* – a bavarian mythical creature – shown in Figure 8.7 as a fake because no hare has a dear head and bird wings. But on

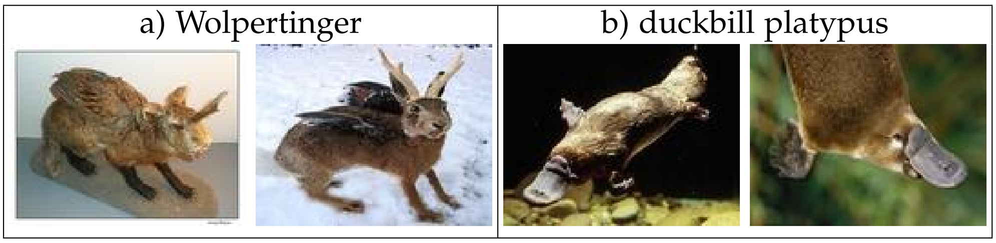


Figure 8.7. Visualization of rare categories and their effect on our inference: a) Wolpertinger a fake object, and b) duckbill platypus a real object that seems like a fake as it mix up properties of different species.

the other side, rare species such as the duckbill platypus (Figure 8.7 right) looks also like an elaborate fraud to us as if someone stick the duckbill on this animal. In fact, this species comes with an unusual appearance and atypical properties for a mammal such as laying eggs like a bird or a reptile, having a tail like a beaver, a bill like a duck, and foots like an otter. Assuming that we can collect more knowledge with a computer then this added information should also improve the inference beyond that of a human. In particular in the shown case from Figure 8.7, a computer should be in a better position to decide which one is a fake. Firstly, each imitation of the *Wolpertinger* looks different in comparison to images of the duckbill platypus. Secondly, we can also take into account the trustability of the source.

LIST OF FIGURES

Fig. 1.1 Computer vision applications with object recognition as an integral part. 2
Fig. 1.2 Illustration of several learning principles that are used to classify the marked unlabeled data point: a) exemplar-based learning, e.g., KNN classification, b) concept-driven, e.g., SVM, or c) hybrid approach, e.g., that groups exemplars around a general concept by transformation with metric learning and than applying KNN. Blue and red points are the labels of two different classes, and black points are the unlabeled data. 3
Fig. 1.3 Unlabeled data reduce the space of hypotheses if there are only few labels. 6
Fig. 1.4 Two half moons dataset with exactly one label per class (marked by a red number): before classification (left) and after classification (right) with 100% accuracy. 7
Fig. 1.5 Workflow of semi-supervised learning algorithms in vision. . . . 7
Fig. 1.6 Examples for a) large intra-class variability for the base category *bird* (top row) and b) for the subcategory *puffin* of the category*bird*, and c) small inter-class differences (bottom row). 8
Fig. 1.7 Example of a) a sparse class description that makes it difficult to find a connection between both images and of b) a dense class representation that makes it possible to find a way from the front view to the side view of a car over several viewpoints. 9
Fig. 1.8 Data source bias: First results for the query *car* with a) Flickr that contains more holiday pictures of cars (top row), and b) Google images with focus on racing cars (bottom row). 9
Fig. 1.9 Dataset bias due to a) the sighting angle, b) the semantic of an object, or c) because of historical trends. 10
Fig. 1.10 Structure with dense representation but still overlapping regions. 11
Fig. 1.11 Examples with a dominating background that is shared among different classes (top row), and examples with overloaded background and object that are transparent or filigree so that background is always a part of the object (bottom row). 11
Fig. 1.12 Examples of a lake with different illuminations depending on the time of day and the season. 12
Fig. 1.13 Examples of truncations and partial occlusions. 12
Fig. 1.14 Examples of the large variety in shape for the class *chair*. 12
Fig. 1.15 Examples of objects that are difficult to distinguish from their background or to identify the object-specific shape. 13

Fig. 1.16 Examples of images that are difficult to understand without color information. 13
Fig. 1.17 Problem of Euclidean distance in a high dimensional space: The image in the middle is similar to many other images. Red boxes indicate false neighbors. 14
Fig. 1.18 Pairwise image similarities might be problematic due to the missing generalization. From the first image it is not clear how to generalize so that this image do not get confused with the sofa in the last image. 14
Fig. 1.19 SSL is strongly dependent on the representativeness of the small training set: a) less representative samples for the entire class *car* vs. b) more representative samples in terms of viewpoints. 15
Fig. 2.1 Visualization of the information loss when we consider only: a) shape; b and c) texture or image patches; d) gray value images in comparison to e) the information content provided by the entire image. 21
Fig. 3.1 A cutout of a symmetric k nearest neighbor graph structure. . . . 40
Fig. 3.2 Sample images of ETH80 (left) and a visualization of the viewpoint angles (right). 45
Fig. 3.3 Sample images of C-PASCAL. 45
Fig. 3.4 Sample images of Caltech 101. 46
Fig. 3.5 Sample images of ILSVRC 2010. 46
Fig. 3.6 Overall accuracy for HOG and 5 labels per class with different algorithms over the number of neighbors k. Graph Quality (GQ) is a theoretical measure calculating the number of correct nearest neighbors. KNN and SVM are the supervised algorithms. Note SVM is a straight line because there are no different neighborhood structures. 47
Fig. 3.7 Overall accuracy for different descriptors with LP Zhou and different number of labels per class averaged over 5 different runs. 49
Fig. 4.1 Example classes with biased objects in Caltech101 with average precision (AP) before and after flipping for TPLBP. 55
Fig. 5.1 Depending on the dataset and their completeness in terms of viewpoints and appearance, the data distribution can result in a more compact distribution where each class build one group (left) or in a more fragmented distribution where a class consists of a separate groups (right). Note that the data points are not produced from real image data. 68
Fig. 5.2 Starting with 1 label per class (green and blue dots), we request iterative labels with the different sampling strategies: a) randomly sampled and b-d) three different sampling strategies that combine exploration and exploitation. False classified examples after label propagation are marked with red dots. The numbers beside the selected points indicate the order in which they were selected. . . 69

Fig. 5.3 Starting with 1 label per class (green and blue dots), we request iterative labels with the different sampling strategies: a) entropy request often samples far away from dense regions; and b) margin focus more on the decision boundaries. The numbers beside the selected points indicate the order in which they were selected. . . 71
Fig. 5.4 Visualization of different parameters for the node potential. Small r_a (left) causes that only outliers are sampled because of the large neighborhood radius and larger r_a results in oversampling of dense regions (right). 73
Fig. 5.5 Starting with 1 label per class (green and blue dots), we request iterative labels with the different sampling strategies: a) kernel farthest first sample evenly the entire space leading to many outlier; b) graph density focus more on the dense regions of the mixtures. The numbers beside the selected points indicate the order in which they were selected. 73
Fig. 5.6 Training samples of C-PASCAL: random best and worst seed (column 1-2), prototypical selection, and best case estimation (4th column), and with our graph density criteria as a filter criteria for best and worst seed (column 5-6). AP is the average precision for this class calculated after applying label propagation. 74
Fig. 5.7 Several synthetic datasets (upper row) and their corresponding time varying curve $\beta(t)$ (lower row). 78
Fig. 5.8 First queried labels of *Two Circles* for different strategies from full density-based to full uncertainty-based sampling. Green and blue points are correct classified. The first two labels are given. The numbers beside the data points indicate the order in which they were selected. 79
Fig. 5.9 Simple mixtures with different constant β for all datasets and the comparison to random sampling. 80
Fig. 5.10 Simple Markov decision process with a) 1 state $S = \{U + D\}$ with $U \in \{Mar, Ent\}$ and $D \in \{Nod, Ker, Gra\}$, and n actions $A = \{\beta_i(t) = a_i\}$ with $a_i \in [0, 1]$; b) 2 states $S = \{Ent + Gra, Mar + Gra\}$ and 4 actions $A = \{0.25(stay), 0.5(stay), 0.75(stay), 0.5(switch)\}$ 83
Fig. 6.1 LP and SVM accuracy of all three datasets and different number of labels for random sampling and the mixture Mar+Gra with and without metric learning. 96
Fig. 7.1 Left: Complexity for selecting $|S|$ unlabeled data $x \in U$ with m dimensions of the image descriptor given a fixed training and test set $|T|$ and label propagation. Right: Complexity against performance of IM100 (see Section 3.3) for DSIFT. 106
Fig. 7.2 Graph quality (first row) and LP accuracy (second row) for ETH80 with (dashed lines) and without (solid lines) active learning for different number of added images: Gist (left), dense SIFT (middle), and spatial dense SIFT (right) 107

Fig. 7.3 Graph quality (first row) and LP accuracy (second row) for C-PASCAL with (dashed lines) and without (solid lines) active learning for different number of added images: Gist (left), dense SIFT (middle), and spatial dense SIFT (right) 109
Fig. 7.4 Graph quality (first row) and LP accuracy (second row) for IM100 with active learning for different number of added images: Gist (left), dense SIFT (middle), and spatial dense SIFT (right) 110
Fig. 8.1 Visualization of an incomplete object class description that makes it difficult to find relation between these images. 117
Fig. 8.2 Visualization of a more complete object class description extracted from a video sequence. 117
Fig. 8.3 Visualization of feasible and unfeasible poses and scenes for the class *anteater* that will be more obvious with informations such as $1.5 - 1.8$m long. 118
Fig. 8.4 First examples for the query *fish* in google images (out of 1.5 billion results) that are not representative for this class. 118
Fig. 8.5 Most confusing classes for ETH-80 although the conditions are optimal for SSL, i.e. smooth manifold structure and no background clutter. 119
Fig. 8.6 Representativeness of viewpoints for the class *armadillo*: images on the left side show less representative viewpoints as they miss important properties of this species while the viewpoints in the right images are most informative for this class. 120
Fig. 8.7 Visualization of rare categories and their effect on our inference: a) Wolpertinger a fake object, and b) duckbill platypus a real object that seems like a fake as it mix up properties of different species. 124

LIST OF TABLES

Tab. 3.1 Overall accuracy for the three best image descriptors HOG, Dense SIFT (DSIFT), and Spatial DSIFT (SpDSIFT) with L1 and L2 distance. Last column is the difference between L1 and L2 distance for the graph quality (GQ) of the 10-NN graph (2nd and 5th column). Last row is the average over all datasets and descriptors in this table. 48

Tab. 4.1 Graph quality (GQ) of the 10-NN graph and overall accuracy for L1 and L2 with PCA reduction to 100 dimensions and the gain to our baseline. 54

Tab. 4.2 Graph quality and overall accuracy for L1 and L2 with flipped images and the gain to our baseline results in Table 3.1. 55

Tab. 4.3 Graph quality and overall accuracy for L1 and L2 with combination and the gain to our baseline. We combine all descriptors and three best descriptors for each datasets by averaging the kernel matrices. 56

Tab. 4.4 Summary of all strategies for HOG 56

Tab. 4.5 Overall accuracy and graph quality of a 10-NN graph for L1 and L2 with LDA and the gain to our baseline results in Table 3.1, and the difference (diff) $NN_U - NN_L$. 59

Tab. 4.6 Overall accuracy and graph quality (GQ) for L2 with SVM score and the gain to our baseline results in Table 3.1, and the difference (diff) $GQ_U - GQ_L$. 60

Tab. 4.7 Overall accuracy for L1 and L2 with SVM score combined with original data and the gain to our baseline results in Table 3.1. Graph quality of the unlabeled data GQ_U and the difference to the labeled data $GQ_U - GQ_L$. 61

Tab. 4.8 Overall accuracy for L1 with ITML and the gain to our baseline results in Table 3.1. Graph quality (GQ) and the difference $GQ_L - GQ_U$. 62

Tab. 4.9 Overall accuracy for L2 with ITML and the gain to our baseline results in Table 3.1. Graph quality (GQ) and the difference $GQ_L - GQ_U$. 63

Tab. 4.10 Summary of all strategies 63

Tab. 4.11 Overall accuracy and graph quality of 10-NN graph for L2 with SVM-based metric extraction. 65

Tab. 4.12 Overall accuracy and graph quality (GQ) of the 10-NN graph for L2 with ITML. 65

Tab. 4.13 Overall accuracy for L1 and L2 of HOG and the gain to our baseline in Table 3.1. 66

Tab. 5.1 Overall accuracy for all single criteria and random sampling after $\max(5C, 100)$ iterations. 76
Tab. 5.2 Area under Curve (AUC) for the synthetic datasets for different $\beta(t)$ 79
Tab. 5.3 Overall accuracy for $\beta(t) = 0.5$ and four different time-varying combinations. 80
Tab. 5.4 Accuracy for the original method proposed by Osugi *et al.* (2005), our rescaling function (Equation (5.15)), the entropy-based reward (Equation (5.16)), and our MDP-based method RALF. . . . 85
Tab. 5.5 Accuracy for our MDP-based approach with 2 to 4 states compared to randomly switching among those action-state-pairs, and the difference to the later one. 85
Tab. 5.6 Summary: Random sampling, best single exploitation and exploration criteria, best combination with fixed and time-varying trade-off, our RALF approach, and differences to random sampling. 86
Tab. 6.1 1-NN accuracy before (1st table) and after (2nd table) metric learning for single criteria and the mixtures Ent+Gra (E+G), Ent+Ker (E+K), Mar+Gra (M+G), and Mar+Ker (M+K). 95
Tab. 6.2 Interleaved active metric learning (IAML) in comparison to the batch active metric learning (BAML) both for Mar+Gra sampling. 97
Tab. 7.1 Graph quality (GQ) and LP accuracy without and with (+AL) active learning for ETH80 after adding 50% unlabeled images. . . 108
Tab. 7.2 Graph quality (GQ) and LP accuracy without and with (+AL) active learning for C-PASCAL after adding 50% unlabeled images.108
Tab. 7.3 Graph quality (GQ) and LP accuracy without and with (+AL) active learning for IM100 after adding 30,000 unlabeled images ($\approx 23\%$). 110
Tab. 7.4 ILSVRC 2010 with *random* and *NN* enrichment for DSIFT: graph quality (GQ), top 1 and top 5 accuracy after LP with AL, and the difference to *random*. 111
Tab. 7.5 IM100: baseline (5 training + 45 test images per class), 25,000 randomly added data without and with AL (row 2-3), with 25,000 *NN* selections without and with AL (row 4-5), and using all unlabeled data without and with AL (row 6-7). 111

BIBLIOGRAPHY

J. Abbott, K. Heller, Z. Ghahramani, and T. L. Griffiths (2011). Testing a Bayesian Measure of Representativeness Using a Large Image Database, in *NIPS 2011*. 22

Y. Abramson and Y. Freund (2005). Active learning for visual object detection, in *CVPR 2005*. 36

S. Agarwal, L. Zelnik-Manor, P. Perona, D. Kriegman, and S. Belongie (2005). Beyond Pairwise Clustering, in *CVPR 2005*. 33

C. Aggarwal, A. Hinneburg, and D. A. Keim (2001). On the surprising behavior of distance metrics in high dimensional space, in *ICDT 2001*. 29

A. K. Agrawala (1970). Learning With a Probabilistic Teacher, *Trans. on Inf. Theory*, vol. 16(4), pp. 373–379. 4

E. B. Ahmed, A. Nabli, and F. Gargouri (2012). SHACUN : Semi-supervised Hierarchical Active Clustering Based on Ranking Constraints, in *ICDM 2012*. 24

A. Alexandrescu and K. Kirchhoff (2007). Data-Driven Graph Construction for Semi-Supervised Graph-Based Learning in NLP, in *NAACL 2007*. 33

D. Angluin and P. Laird (1988). Learning From Noisy Examples, *ML*, vol. 2, pp. 343–370. 34

S. Antifakos and B. Schiele (2003). LaughingLily: Using a Flower as a Real World Information Display, in *Ubicomp 2003*. vii

A. Argyriou, M. Herbster, and M. Pontil (2005). Combining Graph Laplacians for Semi–Supervised Learning, in *NIPS 2005*. 34

F. Ashby (1992). Multidimensional models of categorization, in *Multidimensional models of perception and cognition 1992*, pp. 449–483, lea edn. 2, 34

F. G. Ashby and W. T. Maddox (2011). Human category learning 2.0., *Annals of the New York Academy of Sciences*, vol. 1224, pp. 147–61. 2

M. Aurelius (180). First Book, in *Meditations of the Emperor Marcus Aurelius Antoninus (orig. Ta eis heauton) 180*. viii

J. Azimi, A. Fern, X. Z. Fern, G. Borradaile, and B. Heeringa (2012). Batch Active Learning via Coordinated Matching, in *ICML 2012*. 37

B. Babenko, S. Branson, and S. Belongie (2009). Similarity Metrics for Categorization: from Monolithic to Category Specific, in *ICCV 2009*. 30

M.-F. Balcan and A. Blum (2005). A PAC-style Model for Learning from Labeled and Unlabeled Data, in *COLT 2005*. 5

M.-f. Balcan, A. Blum, P. P. Choi, J. Lafferty, B. Pantano, M. R. Rwebangira, and X. Zhu (2005). Person Identification in Webcam Images : An Application of Semi-Supervised Learning, in *ICML WS 2005*. 34

M. Bar and S. Ullman (1996). Spatial context in recognition., *Perception*, vol. 25(3), pp. 343–52. 21

Y. Baram, R. El-yaniv, and K. Luz (2004). Online Choice of Active Learning Algorithms, *JMLR*, vol. 5, pp. 255–291. 36, 37, 38, 70, 71, 72, 91, 92, 122

J. T. Barron and J. Malik (2012). Shape, Albedo, and Illumination from a Single Image of an Unknown Object, in *CVPR 2012*. 20

H. Barrow and J. Tenenbaum (1978). *Recovering Intrinsic Scene Characteristics from Images*, Academic Press. 20

E. Bart, I. Porteous, P. Perona, and M. Welling (2008). Unsupervised learning of visual taxonomies, in *CVPR 2008*. 22

S. Basu and A. Banerjee (2004). Active Semi-Supervision for Pairwise Constrained Clustering, in *SIAM 2004*. 23, 24, 90, 92

C. Bauckhage and C. Thurau (2009). Making Archetypal Analysis Practical, in *DAGM 2009*. 28

P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman (1997). Eigenfaces vs. Fisherfaces: recognition using class specific linear projection, *TPAMI*, vol. 19(7), pp. 711–720. 29

M. Belkin and P. Niyogi (2003). Laplacian Eigenmaps for Dimensionality Reduction and Data Representation, *Neural Comput*, vol. 15(6), pp. 1373–1396. 25, 29

M. Belkin and P. Niyogi (2005). Towards a theoretical foundation for Laplacian-based manifold methods, in *COLT 2005*. 26

M. Belkin, P. Niyogi, and V. Sindhwani (2006). Manifold Regularization: A Geometric Framework for Learning from Labeled and Unlabeled Examples, *JMLR*, vol. 7, pp. 2399–2434. 26

S. Belongie, J. Malik, and J. Puzicha (2002). Shape matching and object recognition using shape contexts, *TPAMI*, vol. 24(4), pp. 509–522. 20

S. Ben-David, T. Lu, and D. Pal (2008). Does Unlabeled Data Provably Help ? Worst-case Analysis of the Sample Complexity of Semi-Supervised Learning, in *COLT 2008*. 24

Y. Bengio, O. Delalleau, and N. L. Roux (2006). Label Propagation and Quadratic Criterion, in O. Chapelle, B. Sch\"{o}lkopf, and A. Zien (eds.), *Semi-supervised Learning 2006*, chapter 11, pp. 185–207, MIT Press, Cambridge. 26, 44

Y. Bengio, J.-F. Paiement, P. Vincent, O. Delalleau, N. L. Roux, and M. Ouimet (2004). Out-of-sample Extensions for LLE, Isomap, MDS, Eigenmaps, and Spectral Clustering, in *NIPS 2004*. 25

T. Berg and D. Forsyth (2006). Animals on the Web, in *CVPR 2006*. 8

A. Bergamo and L. Torresani (2010). Exploiting weakly-labeled Web images to improve object classification : a domain adaptation approach, in *NIPS 2010*. 23

J. a. R. Bertini, A. D. A. Lopes, and L. Zhao (2012). Partially labeled data stream classification with the semi-supervised K-associated graph, *JBCS*. 33

W. Bian and D. Tao (2007). Learning a Distance Metric by Empirical Loss Minimization, in *IJCAI 2007*. 30

I. Biederman (1972). Perceiving Real-World Scenes, *Science*, vol. 177(4043), pp. 77–80. 21

I. Biederman (1987). Recognition-by-components: a theory of human image understanding., *Psych Rev*, vol. 94(2), pp. 115–47. 1

I. Biederman and G. Ju (1988). Surface versus edge-based determinants of visual recognition., *Cognitive psychology*, vol. 20(1), pp. 38–64. 20

M. Bilenko, S. Basu, and R. J. Mooney (2004). Integrating constraints and metric learning in semi-supervised clustering, in *ICML 2004*. 23, 24, 31

H. Bischof and A. Leonardis (1998). Robust recognition of scaled eigenimages through a hierarchical approach, in *CVPR 1998*. 20

H. Bischof, A. Pinz, and W. G. Kropatsch (1992a). Visualization Methods for Neural Networks, in *IAPR 1992*. 121

H. Bischof, W. Schneider, and A. Pinz (1992b). Multispectral classification of Landsat-images using neural networks, *IEEE Transactions on Geoscience and Remote Sensing*, vol. 30(3), pp. 482–490. 22

M. Blatt, S. Wiseman, and E. Domany (1997). Data Clustering Using a Model Granular Magnet, *Neural Computation*, vol. 9(8), pp. 1805–1842. 32

D. M. Blei, A. Y. Ng, and M. I. Jordan (2003). Latent Dirichlet Allocation, *JMLR*, vol. 3, pp. 993–1022. 22

A. Blum and S. Chawla (2001). Learning from Labeled and Unlabeled Data using Graph Mincuts, in *ICML 2001*. 7, 23, 26

A. Blum, J. Lafferty, M. R. Rwebangira, and R. Reddy (2004). Semi-supervised learning using randomized mincuts, in *ICML 2004*. 26

O. Boiman, E. Shechtman, and M. Irani (2008). In defense of Nearest-Neighbor based image classification, in *CVPR 2008*. 22, 64

A. Bondu, V. Lemaire, and M. Boulle (2010). Exploration vs. exploitation in active learning: a Bayesian approach, in *IJCNN 2010*. 38, 69, 70

O. Bousquet, O. Chapelle, and M. Hein (2003). Measure Based Regularization, in *NIPS 2003*. 24

K. Brinker (2003). Incorporating Diversity in Active Learning with Support Vector Machines, in *ICML 2003*. 37

T. Buehler and M. Hein (2009). Spectral Clustering based on the graph p-Laplacian, in *ICML 2009*. 22

J. M. Buhmann and T. Zöller (2000). Active Learning for Hierarchical Pairwise Data Clustering, in *ICPR 2000*. 36

M. Burl and P. Perona (1996). Recognition of Planar Object Classes, in *CVPR 1996*. 21

D. Cai, X. He, and J. Han (2007a). Semi-supervised Discriminant Analysis, in *ICCV 2007*. 31

D. Cai, X. He, K. Zhou, J. Han, and H. Bao (2007b). Locality Sensitive Discriminant Analysis, in *IJCAI 2007*. 30

C. Campbell, N. Cristianini, and A. Smola (2000). Query Learning with Large Margin Classifier, in *NIPS 2000*. 35

P. Carbonetto, N. D. Freitas, and K. Barnard (2004). A Statistical Model for General Contextual Object Recognition, in *ECCV 2004*. 21

M. A. Carreira-Perpinan and R. S. Zemel (2005). Proximity graphs for clustering and manifold learning, in *NIPS 2005*. 32

V. Castelli and T. M. Cover (1995). On the exponential value of labeled samples, *PRL*, vol. 16, pp. 105–111. 4, 24

V. Castelli and T. M. Cover (1996). The Relative Value of Labeled and Unlabeled Samples in Pattern Recognition with an Unknown Mixing Parameter, *TIT*, vol. 42(6), pp. 2102—-2117. 24

N. Cebron and M. R. Berthold (2006). Adaptive active classification of cell assay images, in *PKDD 2006*. 115

N. Cebron and M. R. Berthold (2009). Active learning for object classification: from exploration to exploitation, *DMKD*, vol. 18(2), pp. 283–299. 38, 69, 70, 72, 76

N. Cebron, F. Richter, and R. Lienhart (2012). "I can tell you what it's not": active learning from counterexamples, *Progress in Artificial Intelligence*. 122

C. Chabris and D. Simons (2010). *The Invisible Gorilla: How Our Intuitions Deceive Us*, Crown Publishing Group. 123

S. Chakraborty, V. Balasubramanian, and S. Panchanathan (2011). Dynamic batch mode active learning, in *CVPR 2011*. 37

C.-C. Chang and C.-J. Lin (2011). LIBSVM: A library for support vector machines, *ACM Transactions on Intelligent Systems and Technology*, vol. 2(3), pp. 1–27. 47, 93

O. Chapelle, P. Haffner, and V. N. Vapnik (1999). Support vector machines for histogram-based image classification., *NN*, vol. 10(5), pp. 1055–64. 22

O. Chapelle and A. Zien (2004). Semi-supervised classification by low density separation, *AISTATS WS*. 24

N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer (2002). SMOTE : Synthetic Minority Over-sampling TEchnique, *JAIR*, vol. 16, pp. 341—-378. 28, 117

H.-t. Chen, H.-w. Chang, and T.-L. Liu (2005). Local Discriminant Embedding and Its Variants, in *CVPR 2005*. 31

B. Cheng, J. Yang, S. Yan, Y. Fu, and T. S. Huang (2010). Learning With L1 -Graph for Image Analysis, *TIP*, vol. 19(4), pp. 858–866. 32

O. Chum, J. Philbin, and A. Zisserman (2008). Near duplicate image detection: min-hash and tf-idf weighting, in *BMVC 2008*. 101

O. Chum and A. Zisserman (2007). An Exemplar Model for Learning Object Classes, in *CVPR 2007*. 22

B. Cohen and G. L. Murphy (1984). Models of concepts, *Cognitive Science*, vol. 8(1), pp. 27–58. 6, 116

D. Cohn, L. Atlas, and R. Ladner (1994). Improving Generalization with Active Learning, *Mach Learn*, vol. 15(2), pp. 201–221. 36

D. A. Cohn, Z. Ghahramani, and M. I. Jordan (1996). Active Learning with Statistical Models, *JAIR*, vol. 4, pp. 129–145. 36

B. Collins, J. Deng, K. Li, and L. Fei-fei (2008). Towards scalable dataset construction : An active learning approach, in *ECCV 2008*. 36, 101

T. Cootes, G. Edwards, and C. Taylor (1998). Active Appearance Models, in *ECCV 1998*. 21

F. G. Cozman, I. Cohen, and M. C. Cirelo (2003). Semi-Supervised Learning of Mixture Models, in *ICML 2003*. 4, 24

D. Crandall, P. Felzenszwalb, and D. Huttenlocher (2005). Spatial Priors for Part-Based Recognition Using Statistical Models, in *CVPR 2005*. 23

N. Cristianini, J. Shawe-Taylor, A. Elisseeff, and J. Kandola (2001). On Kernel-Target Alignment, in *NIPS 2001*. 25

A. Cutler and L. Breiman (1994). Archetypal Analysis, *Technometrics*, vol. 36(4), pp. 338–347. 28

C. Dagli, S. Rajaram, and T. Huang (2005). Combining Diversity-Based Active Learning with Discriminant Analysis in Image Retrieval, in *ICITA 2005*. 36

S. I. Daitch, J. A. Kelner, D. A. Spielman, and N. Haven (2009). Fitting a Graph to Vector Data, in *ICML 2009*. 34

N. Dalal and B. Triggs (2005). Histograms of Oriented Gradients for Human Detection, in *CVPR 2005*. 20, 40

A. Damasio (1994). *Descartes' Error: Emotion, Reason, and the Human Brain*, Penguin Group. 4

E. M. Darling and R. D. Joseph (1968). Pattern Recognition from Satellite Altitudes, *Trans Sys Sci Cyb*, vol. 4(1), pp. 38–47. 20

S. Dasgupta and D. Hsu (2008). Hierarchical sampling for active learning, in *ICML 2008*. 77

J. Davis, B. Kulis, P. Jain, S. Sra, and I. Dhillon (2007). Information-theoretic metric learning, in *ICML 2007*. 30, 31, 57, 60, 61, 90, 91, 115, 121

T. De Bie and N. Cristianini (2004). Convex Methods for Transduction, in *NIPS 2004*. 24

V. Delaitre, D. F. Fouhey, I. Laptev, J. Sivic, A. Gupta, and A. A. Efros (2012). Scene semantics from long-term observation of people, in *ECCV 2012*. 120

O. Delalleau, Y. Bengio, and N. Le Roux (2005). Efficient non-parametric function induction in semi-supervised learning, in *AISTATS 2005*. 27, 28, 100, 101, 103

A. Demiriz, K. P. Bennett, and M. J. Embrechts (1999). Semi-Supervised Clustering Using Genetic Algorithms, in *Proc. Artificial Neural Networks in Engineering 1999*. 24

J. Deng, W. Dong, R. Socher, Li-Jia Li, K. Li, and L. Fei-Fei (2009). ImageNet: A large-scale hierarchical image database, in *CVPR 2009*. 8, 45, 101

T. Deselaers and V. Ferrari (2011). Visual and Semantic Similarity in ImageNet, in *CVPR 2011*. 101

S. C. Dharmadhikari, M. Ingle, and P. Kulkarni (2012). Towards Multi Label Text Classification through Label Propagation, *IJACSA*, vol. 3(6), pp. 31–34. 26

W. Di and M. M. Crawford (2010). Locally consistent graph regularization based active learning for hyperspectral image classification, in *Hyperspectral Image and Signal 2010*. 36

P. Dollár, C. Wojek, B. Schiele, and P. Perona (2012). Pedestrian detection: an evaluation of the state of the art., *TPAMI*, vol. 34(4), pp. 743–61. 23

P. Donmez and J. Carbonell (2007). Dual strategy active learning, in *ECML 2007*. 37, 68

M. Donoser, H. Riemenschneider, and H. Bischof (2009). Efficient Partial Shape Matching of Outer Contours, in *ACCV 2009*. 20

G. Druck and A. McCallum (2010). High-Performance Semi-Supervised Learning using Discriminatively Constrained Generative Models, in *ICML 2010*. 24

C. Dubout and F. Fleuret (2011). Tasting families of features for image classification, in *ICCV 2011*. 21

S. Ebert, M. Fritz, and B. Schiele (2011). Pick your Neighborhood – Improving Labels and Neighborhood Structure for Label Propagation, in *DAGM 2011*. 16, 17, 61

S. Ebert, M. Fritz, and B. Schiele (2012a). Active Metric Learning for Object Recognition, in *DAGM 2012*. 16, 18

S. Ebert, M. Fritz, and B. Schiele (2012b). RALF : A Reinforced Active Learning Formulation for Object Class Recognition, in *CVPR 2012*. 16, 17, 68, 69

S. Ebert, M. Fritz, and B. Schiele (2012c). Semi-Supervised Learning on a Budget: Scaling up to Large Datasets, in *ACCV 2012*. 16, 18, 116, 122

S. Ebert, D. Larlus, and B. Schiele (2010). Extracting Structures in Image Collections for Object Recognition, in *ECCV 2010*. 16, 17, 39, 45, 51, 100, 107

J. Ebrahimi and M. S. Abadeh (2012). Semi Supervised Clustering : A Pareto Approach, in *MLDM 2012*. 24

E. Elhamifar, G. Sapiro, and R. Vidal (2012). See All by Looking at A Few : Sparse Modeling for Finding Representative Objects, in *CVPR 2012*. 28

E. Elhamifar and R. Vidal (2009). Sparse Subspace Clustering, in *CVPR 2009*. 32

M. a. Erickson and J. K. Kruschke (1998). Rules and exemplars in category learning., *Journal of experimental psychology. General*, vol. 127(2), pp. 107–40. 2

M. Everingham, L. Van Gool, and C. K. Williams (2008). *The PASCAL VOC*. 8, 9, 45

M. Farajtabar, A. Shaban, H. R. Rabiee, and M. H. Rohban (2011). Manifold Coarse Graining for Online Semi-supervised Learning, in *ECML 2011*. 27

L. Fei-Fei, R. Fergus, and P. Perona (2006). One-shot learning of object categories, *TPAMI*, vol. 28(4), pp. 594–611. 8, 45

L. Fei-Fei, R. Fergus, and P. Perona (2007). Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories, in *CVIU 2007*. 22

L. Fei-Fei and P. Perona (2005). A Bayesian Hierarchical Model for Learning Natural Scene Categories, in *CVPR 2005*. 21, 22

P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan (2010). Object detection with discriminatively trained part-based models., *TPAMI*, vol. 32(9), pp. 1627–45. 120

P. F. Felzenszwalb and D. P. Huttenlocher (2004). Efficient Graph-Based Image Segmentation, *International Journal of Computer Vision*, vol. 59(2), pp. 167–181. 32

P. F. Felzenszwalb and D. P. Huttenlocher (2005). Pictorial Structures for Object Recognition, *IJCV*, vol. 61(1), pp. 55–79. 20

R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman (2010). Learning object categories from internet image searches, *Proc IEEE*, vol. 98(8), pp. 1453–1466. 101

R. Fergus, P. Perona, and A. Zisserman (2003). Object Class Recognition by Unsupervised Scale-Invariant Learning, in *CVPR 2003*. 22

R. Fergus, Y. Weiss, and A. Torralba (2009). Semi-supervised learning in gigantic image collections, in *NIPS 2009*. 28, 53, 101

M. Fischler and R. Elschlager (1973). The Representation and Matching of Pictorial Structures, *Trans Comp*, vol. 22(1), pp. 67–92. 20

R. A. Fisher (1936). The Use of Multiple Measurements in Taxonomic Problems, *Annals of Eugenics*, vol. 7, pp. 179–188. 30, 31

D. Forsyth and A. Zisserman (1990). Shape from Shading in the Light of Mutual Illumination, *Image and Vision Computing*, vol. 8(1), pp. 42–49. 20

C. Fowlkes, S. Belongie, F. Chung, and J. Malik (2004). Spectral Grouping Using the Nystrom Method, *TPAMI*, vol. 26(2), pp. 214–225. 28

C. Fralick (1967). Learning to Recognize Patterns Without a Teacher, *Trans on Inf Theory*, vol. 13(1), pp. 57–64. 4

W. T. Freeman (2011). Where computer vision needs help from computer science, in *ACM-SIAM Symposium on Discrete Algorithms 2011*. 10

Y. Freund, H. S. Seung, E. Shamir, and N. Tishby (1997). Selective Sampling Using the Query by Committee, *Mach Learn*, vol. 28, pp. 133–168. 35

M. Fritz, M. Black, G. Bradski, and T. Darrell (2009). An additive latent feature model for transparent object recognition, in *NIPS 2009*. 11

M. Fritz, B. Leibe, B. Caputo, and B. Schiele (2005). Integrating representative and discriminant models for object category detection, in *ICCV 2005*. 22

M. Fritz and B. Schiele (2006). Towards Unsupervised Discovery of Visual Categories, in *DAGM 2006*. 22

M. Fritz and B. Schiele (2008). Decomposition, discovery and detection of visual categories using topic models, in *CVPR 2008*. 22

A. Frome, Y. Singer, and J. Malik (2007). Image Retrieval and Classification Using Local Distance Functions, in *NIPS 2007*. 31, 90

Y. Fu, T. M. Hospedales, T. Xiang, and S. Gong (2012). Attribute Learning for Understanding Unstructured Social Activity, in *ECCV 2012*. 37

A. Fujino, N. Ueda, and K. Saito (2005). A Hybrid Generative / Discriminative Approach to Semi-supervised Classifier Design, in *AAAI 2005*. 24

M. Fussenegger, P. M. Roth, H. Bischof, and A. Pinz (2006). On-Line , Incremental Learning of a Robust Active Shape Model, *Pattern Recognition*, vol. 4174, pp. 122–131. 120

C. Galleguillos, A. Rabinovich, and S. Belongie (2008). Object Categorization using Co-Occurrence , Location and Appearance, in *CVPR 2008*. 21

T. Gao and D. Koller (2011). Discriminative Learning of Relaxed Hierarchy for Large-scale Visual Recognition, in *ICCV 2011*. 22

T. Gao, M. Stark, and D. Koller (2012). What Makes a Good Detector? – Structured Priors for Learning From Few Examples, in *ECCV 2012*. 23

P. Gehler and S. Nowozin (2009). On Feature Combination for Multiclass Object Classification, in *ICCV 2009*. 21, 22, 64

B. Gibson, X. Zhu, T. Rogers, and C. Kalish (2010). Humans learn using manifolds, reluctantly, in *NIPS 2010*. 25

A. Globerson and S. Roweis (2006). Metric learning by collapsing classes, in *NIPS 2006*. 30

A. B. Goldberg and X. Zhu (2006). Seeing stars when there are not many stars : Graph-based semi-supervised learning for sentiment categorization, in *Workshop on TextGraphs 2006*. 26

A. B. Goldberg, X. Zhu, and S. Wright (2007). Dissimilarity in Graph-Based Semi-Supervised Classification, *AISTATS*. 34, 52

J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov (2005). Neighbourhood Components Analysis, in *NIPS 2005*. 30

L. Gomez-Chova, G. Camps-valls, J. Munoz-Mari, and J. Calpe (2008). Semisupervised Image Classification With Laplacian Support Vector Machines, *Geoscience and remote sensing letters*, vol. 5(3), pp. 336–340. 26

V. M. Govindu (2005). A Tensor Decomposition for Geometric Grouping and Segmentation, in *CVPR 2005*. 33

H. Grabner, C. Leistner, and H. Bischof (2008). Semi-supervised on-line boosting for robust tracking, *ECCV*. 118

H. Grabner, P. M. Roth, and H. Bischof (2007). Eigenboosting: Combining Discriminative and Generative Information, in *CVPR 2007*. 22

M. Grabner, H. Grabner, and H. Bischof (2006). Fast Approximated SIFT, in *ACCV 2006*. 20

L. Grady and G. Funka-Lea (2004). Multi-Label Image Segmentation for Medical Applications Based on Graph-Theoretic Electrical Potentials, in *ECCV 2004*. 26

Y. Grandvalet and Y. Bengio (2004). Semi-supervised Learning by Entropy Minimization, in *NIPS 2004*. 4, 24, 31

K. Grauman and T. Darrell (2006). Unsupervised learning of categories from sets of partially matching image features, in *CVPR 2006*. 22

G. Griffin and P. Perona (2008). Learning and Using Taxonomies For Fast Visual Categorization, in *CVPR 2008*. 22

M. Guillaumin, J. Verbeek, and C. Schmid (2009). Is that you? Metric learning approaches for face identification, *ICCV*, pp. 498–505. 30

Y. Guo (2010). Active Instance Sampling via Matrix Partition, in *NIPS 2010*. 36

Y. Guo and D. Schuurmans (2007). Discriminative Batch Mode Active Learning, in *NIPS 2007*. 37

R. M. Haralick, K. Shanmugam, and I. Dinstein (1973). Textural Features for Image Classification, *Trans Sys Man Cyb*, vol. 3(6), pp. 610–621. 20

W. G. Hayward (2003). After the viewpoint debate: where next in object recognition?, *Trends in cognitive sciences*, vol. 7(10), pp. 425–7. 119

J. He and J. Carbonell (2007). Nearest-Neighbor-Based Active Learning for Rare Category Detection, in *NIPS 2007*. 37

R. He, W. Zheng, B. Hu, and X. Kong (2011). Nonnegative sparse coding for discriminative semi-supervised learning, in *CVPR 2011*. 33

M. Hein (2005). *Geometrical Aspects of Statistical Learning Theory*, Ph.D. thesis, TU Darmstadt. 23

M. Hein (2006). Uniform Convergence of Adaptive Graph-Based Regularization, in *COLT 2006*. 25

M. Hein and J.-Y. Audibert (2005). Intrinsic dimensionality estimation of submanifolds in R^d, in *ICML 2005*. 29

M. Hein and T. Bühler (2010). An Inverse Power Method for Nonlinear Eigenproblems with Applications in 1-Spectral Clustering and Sparse PCA, in *NIPS 2010*. 123

M. Hein and M. Maier (2006). Manifold Denoising, in *NIPS 2006*. 32, 52, 100, 101

M. Hein and S. Setzer (2011). Beyond Spectral Clustering - Tight Relaxations of Balanced Graph Cuts, in *NIPS 2011*. 123

A. Hillel and D. Weinshall (2007). Subordinate class recognition using relational object models, in *NIPS 2007*. 22

G. Hinton and S. Roweis (2002). Stochastic neighbor embedding, in *NIPS 2002*. 30

M. Hirzer, P. M. Roth, and H. Bischof (2012). Person Re-Identification by Efficient Impostor-based Metric Learning, in *ICAVSS 2012*. 30

S. C. Hoi and M. R. Lyu (2008). Semi-supervised SVM batch mode active learning for image retrieval, in *CVPR 2008*. 31

D. Hoiem, A. A. Efros, and M. Hebert (2006). Putting Objects in Perspective, in *CVPR 2006*. 36

A. Holub, P. Perona, and M. C. Burl (2008). Entropy-based active learning for object recognition, in *CVPR WS 2008*. 36

A. D. Holub, M. Welling, and P. Perona (2005). Combining Generative Models and Fisher Kernels for Object Recognition Kernel Methods, in *ICCV 2005*. 24

Y. Hong, Q. Li, J. Jiang, and Z. Tu (2011). Learning A Mixture of Sparse Distance Metrics for Classification and Dimensionality Reduction, in *ICCV 2011*. 30

T. M. Hospedales, S. Gong, and T. Xiang (2012). A Unifying Theory of Active Discovery and Learning, in *ECCV 2012*. 36

R. Hu, S. J. Delany, and B. M. Namee (2010). EGAL : Exploration Guided Active Learning for TCBR, in *ICCBR 2010*. 36

S. Huang, R. Jin, and Z. Zhou (2010a). Active Learning by Querying Informative and Representative Examples, in *NIPS 2010*. 30, 38, 69

X. Huang, H. Cheng, J. Yang, J. X. Yu, H. Fei, and J. Huan (2012). Semi-supervised Clustering of Graph Objects : A Subgraph Mining Approach, in *DASFAA 2012*. 24

Y. Huang, Q. Liu, F. Lv, Y. Gong, and D. N. Metaxas (2011). Unsupervised Image Categorization by Hypergraph Partition, *TPAMI*, vol. 33(6), pp. 1266–1273. 33

Y. Huang, Q. Liu, and D. Metaxas (2009). Video Object Segmentation by Hypergraph Cut, in *CVPR 2009*. 33

Y. Huang, Q. Liu, S. Zhang, and D. N. Metaxas (2010b). Image Retrieval via Probabilistic Hypergraph Ranking, in *CVPR 2010*. 33

T. Jaakkola, M. Meila, and T. Jebara (1998). Maximum entropy discrimination, in *NIPS 1998*. 22, 24

P. Jain and A. Kapoor (2009). Active learning for large multi-class problems, in *CVPR 2009*. 64

P. Jain, B. Kulis, I. S. Dhillon, and K. Grauman (2010a). Online Metric Learning and Fast Similarity Search, *NIPS*, pp. 1–8. 30

P. Jain, S. Vijayanarasimhan, and K. Grauman (2010b). Hashing Hyperplane Queries to Near Points with Applications to Large-Scale Active Learning, in *NIPS 2010*. 36

T. Jebara, J. Wang, and S.-F. Chang (2009). Graph construction and b -matching for semi-supervised learning, in *ICML 2009*. 32, 52

W. Jiang, S.-f. Chang, T. Jebara, and A. C. Loui (2008). Semantic Concept Classification by Joint Semi-supervised Learning of Feature Subspaces and Support Vector Machines, *ECCV*, pp. 270–283. 23, 26

F. Jing, M. Li, H.-j. Zhang, and B. Zhang (2004). Entropy-based active learning with support vector machines for content-based image retrieval, in *ICME 2004*. 36

T. Joachims (1999). Transductive Inference for Text Classification using Support Vector Machines, in *ICML 1999*. 4, 7, 24

A. J. Joshi, F. Porikli, and N. Papanikolopoulos (2009). Multi-class active learning for image classification, in *CVPR 2009*. 35, 71

D. Kahneman and A. Tversky (1979). Prospect Theory: An Analysis of Decision under Risk, *Econometrica*, vol. 47(2), pp. 263–291. 3, 4

S. D. Kamvar, D. Klein, and C. D. Manning (2003). Spectral Learning, in *IJCAI 2003*. 30

F. Kang, R. Jin, and R. Sukthankar (2006). Correlated Label Propagation with Application to Multi-label Learning, in *CVPR 2006*. 26

J. Kang, K. R. Ryu, and H.-c. Kwon (2004). Using Cluster-Based Sampling to Select Initial Training Set for Active Learning, in *PAKDD 2004*. 36

I. Kant (1781). *Kritik der reinen Vernunft (Critique of Pure Reason)*, Johann Friedrich Hartknoch Verlag (engl. 1838 by Francis Haywood). 4

A. S. Kaplan and G. L. Murphy (2000). Category Learning With Minimal Prior Knowledge, *J Exp Psych*, vol. 26(4), pp. 829–846. 120

A. Kapoor, G. Hua, A. Akbarzadeh, and S. Baker (2009). Which faces to tag: Adding prior constraints into active learning, in *ICCV 2009*. 23

A. Kapoor, Y. A. Qi, H. Ahn, and R. W. Picard (2006). Hyperparameter and Kernel Learning for Graph Based Semi-Supervised Classification, in *NIPS 2006*. 33, 35

M. Karlen, J. Weston, A. Erkan, and R. Collobert (2008). Large scale manifold transduction, in *ICML 2008*. 28

T. Kato, H. Kashima, and M. Sugiyama (2009). Robust label propagation on multiple networks., *Trans. on Neural Networks*, vol. 20(1), pp. 35–44. 34

D. G. Kendall (1984). Shape Manifolds, Procrustean Metrics, and Complex Projective Spaces, *Bull London Math Soc*, vol. 16(2), pp. 81–121. 20

A. Khosla, T. Zhou, T. Malisiewicz, A. A. Efros, and A. Torralba (2012). Undoing the Damage of Dataset Bias, in *ECCV 2012*. 23, 117

G. Kim, C. Faloutsos, and M. Hebert (2008). Unsupervised modeling of object categories using link analysis techniques, in *CVPR 2008*. 22

M. Koestinger, M. Hirzer, P. Wohlhart, P. M. Roth, and H. Bischof (2012). Large Scale Metric Learning from Equivalence Constraints, in *CVPR 2012*. 29

A. Krause and C. Guestrin (2007). Nonmyopic active learning of gaussian processes: an exploration-exploitation approach, in *ICML 2007*. 38, 68

H. Kruppa and B. Schiele (2003). Using Local Context To Improve Face Detection, in *BMVC 2003*. 21

J. K. Kruschke (1992). ALCOVE: An Exemplar-Based Connectionist Model of Category Learning, *Psychological Review*, vol. 99(1), pp. 22—-44. 122

B. Kulis and K. Grauman (2009). Kernelized locality-sensitive hashing for scalable image search, in *ICCV 2009*. 101

B. Kulis, P. Jain, and K. Grauman (2009). Fast Similarity Search for Learned Metrics, *TPAMI*, vol. 31(12), pp. 2143–2157. 30, 31, 57, 90, 92

B. Kulis, K. Saenko, and T. Darrell (2011). What you saw is not what you get: Domain adaptation using asymmetric kernel transforms, in *CVPR 2011*. 23

M. P. Kumar, P. Torr, and A. Zisserman (2007). An Invariant Large Margin Nearest Neighbour Classifier, in *ICCV 2007*. 30

J. Lafferty and L. Wasserman (2007). Statistical Analysis of Semi-Supervised Regression, in *NIPS 2007*. 24, 25

J. Lafferty, X. Zhu, and Y. Liu (2004). Kernel Conditional Random Fields : Representation and Clique Selection, in *ICML 2004*. 25

B. M. Lake and J. L. McClelland (2011). Estimating the strength of unlabeled information during semi-supervised learning, in *Cognitive Science Society 2011*. 25

C. Lampert, H. Nickisch, and S. Harmeling (2009). Learning to detect unseen object classes by between-class attribute transfer, in *CVPR 2009*. 8

D. Larlus, S. Ebert, and B. Schiele (2010). D'une collection d'images a sa structure semantique, vers un processus automatique, in *RFIA 2010*. 17

N. D. Lawrence and M. I. Jordan (2005). Semi-supervised Learning via Gaussian Processes, in *NIPS 2005*. 24

S. Lazebnik, C. Schmid, and J. Ponce (2003). A sparse texture representation using affine-invariant regions, in *CVPR 2003*. 20

S. Lazebnik, C. Schmid, and J. Ponce (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories, in *CVPR 2006*. 41

I.-H. Lee and E. Chang (2005). Manifold Learning: A Promised Land or Work in Progress?, in *ICME 2005*. 30

Y. J. Lee and K. Grauman (2009). Foreground Focus : Unsupervised Learning from Partially Matching Images, *IJCV*, vol. 85, pp. 143–166. 120

B. Leibe, A. Leonardis, and B. Schiele (2004). Combined object categorization and segmentation with an implicit shape model, in *ECCV WS 2004*. 20

B. Leibe and B. Schiele (2003). Analyzing Appearance and Contour Based Methods for Object Categorization, in *CVPR 2003*. 45

B. Leibe, E. Seemann, and B. Schiele (2005). Pedestrian Detection in Crowded Scenes, in *CVPR 2005*. 21

C. Leistner, H. Grabner, and H. Bischof (2008). Semi-supervised boosting using visual similarity learning, in *CVPR 2008*. 23

A. Leonardis and H. Bischof (2000). Robust Recognition Using Eigenimages, in *CVIU 2000*. 20

A. Levin, D. Lischinski, and Y. Weiss (2004). Colorization using optimization, *ACM Transactions on Graphics*, vol. 23(3), p. 689. 26

D. T. Levin and D. J. Simons (1997). Failure to detect changes to attended objects in motion pictures, *Psychonomic Bulletin & Review*, vol. 4(4), pp. 501–506. 123

C.-g. Li, X. Qi, J. Guo, and B. Xiao (2011). An Evaluation on Different Graphs for Semi-supervised Learning, in *IScIDE 2011*. 32

F. Li, Q. Dai, W. Xu, and G. Er (2008). Multilabel Neighborhood Propagation for Region-Based Image Retrieval, *IEEE Transactions on Multimedia*, vol. 10(8), pp. 1592–1604. 26

H. Li and Y. Fan (2012). Label Propagation with Robust Initialization for Brain Tumor Segmentation, in *ISBI 2012*. 26

L.-J. Li, G. Wang, and L. Fei-Fei (2007). OPTIMOL: automatic Online Picture collecTion via Incremental MOdel Learning, in *CVPR 2007*. 28

W. Li and M. Fritz (2012). Recognizing Materials from Virtual Examples, in *ECCV 2012*. 117

Y.-F. Li and Z.-H. Zhou (2011a). Improving Semi-Supervised Support Vector Machines Through Unlabeled Instances Selection, in *AAAI 2011*. 27

Y.-F. Li and Z.-H. Zhou (2011b). Towards Making Unlabeled Data Never Hurt, in *ICML 2011*. 24

Y. Lin, F. Lv, S. Zhu, M. Yang, T. Cour, K. Yu, L. Cao, and T. Huang (2011). Large-scale image classification: fast feature extraction and SVM training, in *CVPR 2011*. 101

D. Liu and T. Chen (2007). Unsupervised Image Categorization and Object Localization using Topic Models and Correspondences between Images, in *ICCV 2007*. 22

J. Liu, S. Chen, and X. Tan (2008). A study on three linear discriminant analysis based methods in small sample size problem, *Pattern Recognition*, vol. 41(1), pp. 102–116. 30

W. Liu and S. Chang (2009). Robust multi-class transductive learning with graphs, in *CVPR 2009*. 33, 52, 100

W. Liu, J. He, and S. Chang (2010). Large graph construction for scalable semi-supervised learning, in *ICML 2010*. 27, 28, 100, 101, 103

J. Long, J. Yin, W. Zhao, and E. Zhu (2008). Graph-Based Active Learning Based on Label Propagation, in *MDAI 2008*. 36

D. G. Lowe (2004). Distinctive Image Features from Scale-Invariant Keypoints, *IJCV*, pp. 1–28. 20

Z. Lu and H. H. S. Ip (2010). Combining Context, Consistency, and Diversity Cues for Interactive Image Categorization, *TM*, vol. 12(3), pp. 194–203. 36

Z. Lu, P. Jain, and I. S. Dhillon (2009). Geometry-aware metric learning, in *ICML 2009*. 31, 121

T. Luo, K. Kramer, D. B. Goldgof, L. O. Hall, S. Samson, A. Remsen, and T. Hopkins (2005). Active Learning to Recognize Multiple Types of Plankton, *JMLR*, vol. 6, pp. 589–613. 35

U. V. Luxburg, A. Radl, and M. Hein (2010). Getting lost in space : Large sample analysis of the commute distance, in *NIPS 2010*. 14, 29, 42, 121

F. Maes, L. Wehenkel, and D. Ernst (2012). Meta-Learning of Exploration/Exploitation Strategies: The Multi-Armed Bandit Case, *arXiv*. 37

S. Mahamud and M. Hebert (2003). The Optimal Distance Measure for Object Detection ∗, in *CVPR 2003*. 22

M. Maier, M. Hein, and U. von Luxburg (2009). Optimal construction of k-nearest-neighbor graphs for identifying noisy clusters, *TCS*, vol. 410(19), pp. 1749–1764. 32

M. Maier, U. V. Luxburg, and M. Hein (2008). Influence of graph construction on graph-based clustering measures, in *NIPS 2008*. 32

J. Malik, S. Belongie, T. Leung, and J. Shi (2001). Contour and Texture Analysis for Image Segmentation, *IJCV*, vol. 43(1), pp. 7–27. 20

T. Malisiewicz and A. Efros (2008). Recognition by association via learning per-exemplar distances, in *CVPR 2008*. 31

T. Malisiewicz, A. Gupta, and A. a. Efros (2011). Ensemble of exemplar-SVMs for object detection and beyond, in *ICCV 2011*. 22, 90

P. K. Mallapragada, R. Jin, A. K. Jain, and Y. Liu (2009). SemiBoost: boosting for semi-supervised learning., *TPAMI*, vol. 31(11), pp. 2000–14. 23

M. Marszalek and C. Schmid (2008). Constructing Category Hierarchies for Visual Recognition, in *ECCV 2008*. 22

A. Martinez and A. Kak (2001). PCA versus LDA, *TPAMI*, vol. 23(2), pp. 228–233. 30, 57, 58

A. K. McCallum and K. Nigam (1998). Employing EM and Pool-Based Active Learning for Text Classification, in *ICML 1998*. 35

J. V. McDonnell, C. A. Jew, and T. M. Gureckis (2012). Sparse category labels obstruct generalization of category membership, in *Cognitive Science Society 2012*. 25

D. L. Medin and M. M. Schaffer (1978). Context Theory of Classification Learning, *Psychological Review*, vol. 85(3), pp. 207–238. 1, 122

N. Meinshausen and P. Bühlmann (2006). High Dimensional Graphs and Variable Selection with the Lasso, *Annals of Statistics*, vol. 34(3), pp. 1436–1462. 32

K. Mikolajczyk, B. Leibe, B. Schiele, and T. U. Darmstadt (2005). Local Features for Object Class Recognition, in *ICCV 2005*. 20

J. P. Minda and J. D. Smith (2001). Prototypes in category learning: The effects of category size, category structure, and stimulus complexity, *Journal of Experimental Psychology: Learning, Memory, and Cognition*, vol. 27(3), pp. 775–799. 2, 122

G. L. Murphy (2002). *The Big Book of Concepts*. 2

G. L. Murphy and P. D. Allopenna (1994). The locus of knowledge effects in concept learning., *Journal of experimental psychology. Learning, memory, and cognition*, vol. 20(4), pp. 904–19. 5, 120

G. L. Murphy and D. L. Medin (1985). The Role of Theories in Conceptual Coherence, *Psychological Review*, vol. 92(3), pp. 289–316. 122

K. Murphy, A. Torralba, and W. T. Freeman (2003). Using the Forest to See the Trees : A Graphical Model Relating Features , Objects , and Scenes, in *NIPS 2003*. 21

I. Muslea, S. Minton, and C. A. Knoblock (2002). Active + Semi-Supervised Learning = Robust Multi-View Learning, in *ICML 2002*. 36

B. Nadler, N. Srebro, and X. Zhou (2009). Semi-Supervised Learning with the Graph Laplacian : The Limit of Infinite Unlabelled Data, in *NIPS 2009*. 24, 25

H. T. Nguyen and A. Smeulders (2004). Active learning using pre-clustering, in *ICML 2004*. 36, 37, 115

K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell (1999). Text Classification from Labeled and Unlabeled Documents using EM, *Mach Learn*. 4, 7, 24

G. Niu, B. Dai, M. Yamada, and M. Sugiyama (2012). Information-theoretic Semi-supervised Metric Learning via Entropy Regularization, in *ICML 2012*. 31

R. M. Nosofsky (1984). Choice, Similarity, and the Context Theory of Classification, *Experimental Psychology*, vol. 10(1), pp. 104–114. 1, 122

S. Okada and T. Nishida (2010). Multi Class Semi-Supervised Classification with Graph Construction Based on Adaptive Metric Learning, in *ICANN 2010*. 31

A. Oliva and A. Torralba (2001). Modeling the shape of the scene: A holistic representation of the spatial envelope, *IJCV*. 41

A. Oliva and A. Torralba (2007). The role of context in object recognition., *Trends in cognitive sciences*, vol. 11(12), pp. 520–7. 21

D. N. Osherson and E. E. Smith (1981). On the adequacy of prototype theory as a theory of concepts, *Cognition*, vol. 9(1), pp. 35–58. 6

T. Osugi, D. Kun, and S. Scott (2005). Balancing exploration and exploitation: A new algorithm for active machine learning, in *ICDM 2005*. 37, 68, 70, 71, 72, 77, 81, 84, 85, 122, 130

T. E. Palmer (1975). The effects of contextual scenes on the identification of objects, *Memory & Cognition*, vol. 3(5), pp. 519–526. 21

D. Parikh and K. Grauman (2011). Relative attributes, in *ICCV 2011*. 121

M. J. Pazzani (1991). Influence of Prior Knowledge on Concept Acquisition : Experimental and Computational Results, *Journal of Experimental Psychology*, vol. 17(3), pp. 416–432. 5

K. Pearson (1901). On lines and planes of closest fit to systems of points in space., *Philosophical Magazine*, vol. 2(6), pp. 559–572. 29, 31

D. Pelleg and A. Moore (2004). Active Learning for Anomaly and Rare-Category Detection, in *NIPS 2004*. 37

R. Penrose (1989). *The Emperor's New Mind: Concerning Computers, Minds and The Law of Physics*, Oxford University Press. 3

B. Pepik, M. Stark, P. Gehler, and B. Schiele (2012). Teaching 3D Geometry to Deformable Part Models, in *CVPR 2012*. 119

F. Perronnin and Y. Liu (2010). Large-scale image retrieval with compressed Fisher vectors, in *CVPR 2010*. 101

A. Pinz, H. Bischof, W. Kropatsch, G. Schweighofer, Y. Haxhimusa, A. Opelt, and A. Ion (2008). Representations for Cognitive Vision: A Review of Appearance-Based, Spatio-Temporal, and Graph-Based Approaches, *Electronic Letters on Computer Vision and Image Analysis*, vol. 7(2), pp. 35–61. 20

L. Pishchulin, A. Jain, M. Andriluka, T. Thormälen, and B. Schiele (2012). Articulated People Detection and Pose Estimation : Reshaping the Future, in *CVPR 2012*. 117

J. Ponce, T. L. Berg, M. Everingham, D. A. Forsyth, M. Hebert, S. Lazebnik, M. Marszalek, C. Schmid, B. Russell, A. Torralba, C. Williams, J. Zhang, and A. Zisserman (2006). Dataset Issues in Object Recognition, in J. Ponce, M. Hebert, C. Schmid, and A. Zisserman (eds.), *Towards Category-Level Object Recognition 2006*, pp. 29–48, Springer LNCS. 3, 23, 28, 113, 117

A. Pope and D. G. Lowe (1996). Learning appearance models for object recognition, in *Object Representation in Computer Vision II 1996*. 21

M. I. Posner, R. Goldsmith, and K. E. Welton (1967). Perceived distance and the classification of distorted patterns., *Journal of experimental psychology*, vol. 73(1), pp. 28–38. 2, 122

S. Prabhakaran, S. Raman, J. E. Vogt, and V. Roth (2012). in Archetype Analysis, in *DAGM 2012*. 28

G.-j. Qi, X.-s. Hua, Y. Rui, J. Tang, and H.-j. Zhang (2008). Two-Dimensional Active Learning for image classification, in *CVPR 2008*. 36

J. Qian, V. Saligrama, M. Zhao, and M. View (2011). Graph Construction for Learning with Unbalanced Data, *arXiv*. 32

A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie (2007). Objects in Context, in *ICCV 2007*. 21

S. S. Rangapuram and M. Hein (2012). Constrained 1-Spectral Clustering, in *AISTATS 2012*. 30, 90

D. Rao and D. Yarowsky (2009). Ranking and Semi-supervised Classification on Large Scale Graphs Using Map-Reduce, in *TextGraphs-4, WS NLP 2009*. 28

J. Ratsaby and S. S. Venkatesh (1995). Learning from a mixture of labeled and unlabeled examples with parametric side information, in *COLT 1995*. 4

H. Riemenschneider, M. Donoser, and H. Bischof (2010). Using Partial Edge Contour Matches for Efficient Object Category Localization, in *ECCV 2010*. 20

P. Rigollet (2007). Generalization Error Bounds in Semi-supervised Classification Under the Cluster Assumption, *JMLR*, vol. 8, pp. 1369–1392. 24

M. H. Rohban and H. R. Rabiee (2012). Supervised neighborhood graph construction for semi-supervised classification, *PR*, vol. 45(4), pp. 1363–1372. 33, 34, 52, 57

M. Rohrbach, M. Stark, and B. Schiele (2011). Evaluating Knowledge Transfer and Zero-Shot Learning in a Large-Scale Setting, in *CVPR 2011*. 101, 121

E. Rosch, C. B. Mervis, W. D. Gray, D. M. Johnson, and P. Boyes-Braem (1976). Basic Objects in Natural Categories, *Cognitive Psychology*, vol. 8, pp. 382—-439. 22, 121

S. T. Roweis and L. K. Saul (2000). Nonlinear dimensionality reduction by locally linear embedding., *Science*, vol. 290(5500), pp. 2323–6. 25, 29, 30, 33

N. Roy and A. McCallum (2001). Toward Optimal Active Learning through Sampling Estimation of Error, in *ICML 2001*. 36

B. Russell, A. Efros, J. Sivic, W. Freeman, and A. Zisserman (2006). Using multiple segmentations to discover objects and their extent in image collections, in *CVPR 2006*. 22

K. Saenko, B. Kulis, M. Fritz, and T. Darrell (2010). Adapting visual category models to new domains, in *ECCV 2010*. 23, 30

A. Saffari, M. Godec, T. Pock, C. Leistner, and H. Bischof (2010). Online multi-class LPBoost, in *CVPR 2010*. 118

A. Saffari, H. Grabner, and H. Bischof (2008). SERBoost : Semi-supervised Boosting with Expectation Regularization, in *ECCV 2008*. 23

B. Schiele (2000). Towards Automatic Extraction and Modeling of Objects from Image Sequences, in *Int Sym on Intelligent Robotic Systems 2000*. 116

B. Schiele and J. Crowley (1996). Where to look next and what to look for, in *IROS 1996*. 119

B. Schiele and J. L. Crowley (1997). The Concept of Visual Classes for Object Classification, in *Scand Conf Image Analysis 1997*. 116

B. Schiele and J. L. Crowley (1998). Transinformation for Active Object Recognition, in *ICCV 1998*. 120

B. Schiele and J. L. Crowley (2000). Recognition without Correspondence using Multidimensional Receptive Field Histograms, *IJCV*, vol. 36(1). 21

B. Schiele and A. Pentland (1999). Probabilistic object recognition and localization, in *ICCV 1999*. 119

P. Schnitzspan, M. Fritz, S. Roth, B. Schiele, and U. C. B. Eecs (2009). Discriminative Structure Learning of Hierarchical Representations for Object Detection, in *CVPR 2009*. 21

G. Schohn and D. Cohn (2000). Less is more: Active learning with support vector machines, *ICML*. 35

B. Schölkopf, J. C. Platt, J. Shawe-Taylor, a. J. Smola, and R. C. Williamson (2001). Estimating the support of a high-dimensional distribution., *Neural computation*, vol. 13(7), pp. 1443–71. 24

B. Schölkopf, A. Smola, and K. R. Müller (1998). Nonlinear Component Analysis as a Kernel Eigenvalue Problem, *Neural Comput*, vol. 10(5), pp. 1299–1319. 29

F. Schroff, A. Criminisi, and A. Zisserman (2007). Harvesting Image Databases from the Web, *ICCV*, pp. 1–8. 101

C. Scott, A. Arbor, G. Blanchard, and F. F. Ida (2009). Novelty detection : Unlabeled data definitely help, in *AISTATS 2009*. 24

H. Scudder (1965). Probability of error of some adaptive pattern-recognition machines, *IEEE Transactions on Information Theory*, vol. 11(3), pp. 363–371. 4

N. Sebe, M. Lew, and D. Huijsmans (2000). Toward improved ranking metrics, *TPAMI*, vol. 22(10), pp. 1132–1143. 29

M. Seeger (2001). Learning with labeled and unlabeled data, Technical report, University of Edinburgh. 4, 7, 24

B. Settles (2009). Active Learning Literature Survey, Technical report, University of Wisconsin–Madison. 35

B. Settles and M. Craven (2008). An analysis of active learning strategies for sequence labeling tasks, in *Emp. Meth. in NLP 2008*. 35, 68, 71, 91

H. S. Seung, M. Opper, and H. Sompolinsky (1992). Query by committee, in *COLT 1992*. 35

S. Shalev-Shwartz, Y. Singer, and A. Y. Ng (2004). Online and Batch Learning of Pseudo-Metrics, in *ICML 2004*. 30

A. Shashua and T. Hazan (2005). Non-negative tensor factorization with applications to statistics and computer vision, in *ICML 2005*. 33

A. Shashua, R. Zass, and T. Hazan (2006). Multi-way Clustering Using Super-Symmetric Non-negative Tensor Factorization, in *ECCV 2006*. 33

E. Shechtman and M. Irani (2007). Matching Local Self-Similarities across Images and Videos, in *CVPR 2007*. 41

C. Shen, J. Kim, and L. Wang (2010). Scalable large-margin Mahalanobis distance metric learning, *TNN*, vol. 21(9), pp. 1524–1530. 30

N. Shental, T. Hertz, D. Weinshall, and M. Pavel (2002). Adjustment learning and relevant component analysis, in *ECCV 2002*. 30

H. H. Shin, N. J. Hill, and G. Ratsch (2006). Graph Based Semi-Supervised Learning with Sharper Edges, in *ECML 2006*. 33, 52

J. Shotton, J. Winn, C. Rother, and A. Criminisi (2007). TextonBoost for Image Understanding : Multi-Class Object Recognition and Segmentation by Jointly Modeling Texture , Layout , and Context, *IJCV*. 21

B. Siddiquie and A. Gupta (2010). Beyond active noun tagging: Modeling contextual interactions for multi-class active learning, in *CVPR 2010*. 35

I. Simon, N. Snavely, and S. M. Seitz (2007). Scene Summarization for Online Image Collections, in *ICCV 2007*. 27

D. J. Simons and C. F. Chabris (1999). Gorillas in our midst: sustained inattentional blindness for dynamic events., *Perception*, vol. 28(9), pp. 1059–74. 123

D. J. Simons and D. T. Levin (1998). Failure to detect changes to people during a real-world interaction, *Psychonomic Bulletin & Review*, vol. 5(4), pp. 644–649. 123

V. Sindhwani, M. Belkin, and P. Niyogi (2006). The geometric basis of semi-supervised learning, in O. Chapelle, B. Schölkopf, and A. Zien (eds.), *Semi-supervised Learning 2006*, chapter 12, pp. 209–227, MIT Press, Cambridge. 23, 26

V. Sindhwani, P. Niyogi, and Belkin (2005). Beyond the point cloud: from transductive to semi-supervised learning, *ICML*. 4, 26, 101

J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman (2005). Discovering Object Categories in Image Collections, in *ICCV 2005*. 3, 22

J. Sivic, B. C. Russell, A. Zisserman, W. T. Freeman, and A. a. Efros (2008). Unsupervised discovery of visual object class hierarchies, in *CVPR 2008*. 22

E. Smith and D. L. Medin (1981). *Categories and concepts*, Harvard University Press, Cambridge. 122

A. J. Smola and R. Kondor (2003). Kernels and Regularization on Graphs, in *COLT 2003*. 25

Y. Song, F. Nie, and C. Zhang (2008). Semi-supervised sub-manifold discriminant analysis, *PRL*, vol. 29(13), pp. 1806–1813. 31

S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf (2006). Large Scale Multiple Kernel Learning, *JMLR*, vol. 7, pp. 1531–1565. 21

M. Stark, M. Goesele, and B. Schiele (2009). A shape-based object class model for knowledge transfer, in *ICCV 2009*. 20

M. Stark, M. Goesele, and B. Schiele (2010). Back to the Future : Learning Shape Models from 3D CAD Data, in *BMVC 2010*. 119

S. Sternig, P. M. Roth, and H. Bischof (2012). On-line inverse multiple instance boosting for classifier grids, *Pattern recognition letters*, vol. 33-178(7), pp. 890–897. 118

M. Stikic, D. Larlus, S. Ebert, and B. Schiele (2011). Weakly Supervised Recognition of Daily Life Activities with Wearable Sensors., *TPAMI*, vol. 33(12), pp. 2521–2537. 16, 26

M. Straka and S. Hauswiesner (2011). Skeletal Graph Based Human Pose Estimation in Real-Time, in *BMVC 2011*. 90

T. M. Strat and M. A. Fischler (1991). Context-Based Vision : Recognizing Objects Using Information from Both 2-D and 3-D Imagery, *TPAMI*, vol. 13(10), pp. 1050–65. 21

A. Subramanya, S. Petrov, and F. Pereira (2010). Efficient Graph-Based Semi-Supervised Learning of Structured Tagging Models, in *EMNLP 2010*. 26

M. Sugiyama and N. Rubens (2008). Active Learning with Model Selection in Linear Regression, in *DMKD 2008*. 35

L. Sun, S. Ji, and J. Ye (2008). Hypergraph spectral learning for multi-label classification, in *KDD 2008*. 33

A. Talwalkar, S. Kumar, and H. Rowley (2008). Large-scale manifold learning, *CVPR*, pp. 1–8. 28

H. Tamura, S. Mori, and T. Yamawaki (1978). Textural Features Corresponding Visual Perception, *Trans on Sys, Man, and Cyb*, vol. 75(6), pp. 460–473. 20

J. Tang, H. Li, G.-j. Qi, and T.-s. Chua (2010). Image Annotation by Graph-Based Inference With Integrated Multiple/Single Instance Representations, *TM*, vol. 12(2), pp. 131–141. 26

J. Tang, Z.-j. Zha, and D. Tao (2011). Semantic-Gap Oriented Active Learning for Multi-Label Image Annotation, *Image Processing*, vol. 21(4), pp. 2354–2360. 36

J. B. Tenenbaum, V. de Silva, and J. C. Langford (2000). A global geometric framework for nonlinear dimensionality reduction., *Science*, vol. 290(5500), pp. 2319–23. 25, 29

R. Teramoto (2008). Prediction of Alzheimer's diagnosis using semi-supervised distance metric learning with label propagation., *Computational biology and chemistry*, vol. 32(6), pp. 438–41. 31

D. W. Thompson (1917). On the Theory of Transformations, or the Comparison of Related Forms, in *On Growth and Form 1917*, chapter XVII, Cambridge University Press, 1st edn. 20

S. B. Thrun and K. Moeller (1992). Active Exploration in Dynamic Environments, in *NIPS 1992*. 37

E. Tola, V. Lepetit, and P. Fua (2008). A Fast Local Descriptor for Dense Matching, in *CVPR 2008*. 21

H. Tong, J. He, M. Li, C. Zhang, and W. Ma (2005). Graph based multi-modality learning, in *ACM Multimedia 2005*. 26, 34

S. Tong and D. Koller (2001). Support Vector Machine Active Learning with Applications to Text Classification, *JMLR*, vol. 2, pp. 45–66. 35

W. Tong and R. Jin (2007). Semi-supervised learning by mixed label propagation, in *AAAI 2007*. 34

A. Torralba (2003). Contextual priming for object detection, *IJCV*, vol. 53(2), pp. 169–191. 21

A. Torralba (2011). Unbiased look at dataset bias, in *CVPR 2011*. 3, 23, 28, 99, 113, 117

A. Torralba, R. Fergus, and Y. Weiss (2008). Small codes and large image databases for recognition, in *CVPR 2008*. 8, 101

A. Torralba, K. P. Murphy, and W. T. Freeman (2006). Shared Features for Multiclass Object Detection, in J. Ponce, M. Hebert, C. Schmid, and A. Zisserman (eds.), *Toward Category-Level Object Recognition 2006*, pp. 345–361. 22

B. A. Torralba, B. C. Russell, and J. Yuen (2010). LabelMe : Online Image Annotation and Applications, *Proc IEEE*. 8

I. W. Tsang and J. T. Kwok (2006). Large-Scale Sparsified Manifold Regularization, in *NIPS 2006*. 28

K. Tsuda, H. Shin, and B. Schoelkopf (2005). Fast protein classification with multiple networks, *Bioinformatics*, vol. 21, pp. 59–65. 34

M. Turk and A. Pentland (1991). Face recognition using eigenfaces, in *CVPR 1991*. 29, 53

T. Tuytelaars, C. H. Lampert, M. B. Blaschko, and W. Buntine (2009). Unsupervised Object Discovery: A Comparison, *IJCV*, vol. 88(2), pp. 284–302. 22

T. Tuytelaars and C. Schmid (2007). Vector Quantizing Feature Space with a Regular Lattice, in *ICCV 2007*. 21

M. Uray, D. Skocaj, P. M. Roth, H. Bischof, and A. Leonardis (2007). Incremental LDA Learning by Combining Reconstructive and Discriminative Approaches $*$, in *BMVC 2007*. 31

L. Valiant (1984). A Theory of the Learnable, *Communications of the ACM*, vol. 27(11), pp. 1134–1142. 22

K. E. a. van de Sande, T. Gevers, and C. G. M. Snoek (2010). Evaluating color descriptors for object and scene recognition., *Trans. on PAMI*, vol. 32(9), pp. 1582–96. 20

K. Vandist, M. D. Schryver, and Y. Rosseel (2009). Semisupervised category learning: The impact of feedback in learning the information-integration task, *Attention, Perception, & Psychophysics*, vol. 71(2), pp. 328–341. 25

V. Vapnik and A. Chervonenkis (1971). On the uniform convergence of relative frequencies of events to their probabilities, *Theory of Probability and its Applications*, vol. 16(2), pp. 264–280. 22

V. Vapnik and A. Sterin (1977). On structural risk minimization or overall risk in a problem of pattern recognition, *Automation and Remote Control*, vol. 10(3), pp. 1495–1503. 4

M. Varma and D. Ray (2007). Learning The Discriminative Power-Invariance Trade-Off, *ICCV*. 22

A. Vedaldi and B. Fulkerson (2008). *VLFEAT: An Open and Portable Library of Computer Vision Algorithms*. 41

A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman (2009). Multiple kernels for object detection, in *ICCV 2009*. 21

D. Vernon (2005). A Research Roadmap of Cognitive Vision, Technical report, ECVision: The European Research Network for Cognitive Computer Vision Systems. 20, 116

A. Vezhnevets, V. Ferrari, and J. M. Buhmann (2012). Weakly Supervised Semantic Segmentation with a Multi-Image Model, in *CVPR 2012*. 36

S. Vijayanarasimhan and K. Grauman (2009). What's it going to cost you?: Predicting effort vs. informativeness for multi-label image annotations, in *CVPR 2009*. 35

S. Vijayanarasimhan and K. Grauman (2011). Large-Scale Live Active Learning : Training Object Detectors with Crawled Data and Crowds, in *CVPR 2011*. 36

S. Vijayanarasimhan, P. Jain, and K. Grauman (2010). Far-sighted active learning on a budget for image and video recognition, in *CVPR 2010*. 37

C. von Ehrenfels (1890). On the Qualities of Form, *Vierteljahrsschrift für wissenschaftliche Philosophie*, vol. 14, pp. 249–292. 20

K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl (2001). Constrained K-means Clustering with Background Knowledge, in *ICML 2001*. 23, 24

F. Wang and C. Zhang (2007a). Label propagation through linear neighborhoods, *TKDE*, vol. 1, pp. 55–67. 32, 52, 101

F. Wang and C. Zhang (2007b). Robust self-tuning semi-supervised learning, *Neurocomputing*, vol. 70(16-18), pp. 2931–2939. 52

G. Wang and D. Forsyth (2009). Joint learning of visual attributes, object classes and visual saliency, in *ICCV 2009*. 28

G. Wang, D. Hoiem, and D. Forsyth (2009a). Learning image similarity from flickr groups using stochastic intersection kernel machines, in *ICCV 2009*. 29

G. Wang, B. Wang, X. Yang, and G. Yu (2012). Efficiently Indexing Large Sparse Graphs for Similarity Search, *TKDE*, vol. 24(3), pp. 440–451. 28

J. Wang, T. Jebara, and S.-F. Chang (2008). Graph transduction via alternating minimization, in *ICML 2008*. 33

J. Wang, F. Wang, C. Zhang, H. C. Shen, and L. Quan (2009b). Linear neighborhood propagation and its applications., *TPAMI*, vol. 31(9), pp. 1600–15. 33

L. Wang, K. L. Chan, and Z. Zhang (2003). Bootstrapping SVM active learning by incorporating unlabelled images for image retrieval, in *CVPR 2003*. 35

X. Wang, T. X. Han, and S. Yan (2009c). An HOG-LBP human detector with partial occlusion handling, in *ICCV 2009*. 21

Z. Wang, Y. Hu, and L.-t. Chia (2010). Image-to-Class Distance Metric Learning for Image Classification, in *ECCV 2010*. 30

Z. Wang, S. Yan, and C. Zhang (2011). Active learning with adaptive regularization, *Pattern Recognition*, pp. 1–9. 35, 69

M. Weber, M. Welling, and P. Perona (2000). Unsupervised Learning of Models for Recognition, in *ECCV 2000*. 3, 22

K. Weinberger (2008). Fast solvers and efficient implementations for distance metric learning, in *ICML 2008*. 30

K. Q. Weinberger and L. K. Saul (2009). Distance Metric Learning for Large Margin Nearest Neighbor Classification, *JMLR*, vol. 10, pp. 207–244. 30

P. Welinder, S. Branson, S. Belongie, and P. Perona (2010). The multidimensional wisdom of crowds, in *NIPS 2010*. 3, 6, 8, 22, 113

M. Wertheimer (1912). On Perceived Motion and Figural Organization, *Zeitschrift für Psychologie*, vol. 60, pp. 321–378. 20

L. Wiskott and C. von der Malsburg (1993). A Neural System for the Recognition of Partially Occluded Objects in Cluttered Scenes, *IJPRAI*, vol. 7(4). 21

L. Wolf and S. Bileschi (2006). A Critical View of Context, *IJCV*, vol. 69(2), pp. 251–261. 21

L. Wolf, T. Hassner, and Y. Taigman (2008). Descriptor Based Methods in the Wild, in *ECCV 2008*. 41

L. Wolf, T. Hassner, and Y. Taigman (2009). The One-Shot similarity kernel, in *ICCV 2009*. 29

J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. Huang, and S. Yan (2009). Sparse Representation For Computer Vision and Pattern Recognition, *Proc IEEE*. 32

G. Wu, Y. Li, J. Xi, X. Yang, and X. Liu (2012a). Local Learning Integrating Global Structure for Large Scale Semi-supervised Classification, in *ICNC 2012*. 28

L. Wu, X.-S. Hua, N. Yu, W.-Y. Ma, and S. Li (2012b). Flickr distance: a relationship measure for visual concepts., *TPAMI*, vol. 34(5), pp. 863–75. 29

E. Xing, A. Ng, M. Jordan, and S. Russell (2003). Distance metric learning with application to clustering with side-information, in *NIPS 2003*. 30, 31

X. Xu, L. Lu, P. He, Z. Pan, and C. Jing (2012). Protein Classification Using Random Walk on Graph, in *ICIC 2012*. 26

Z. Xu, R. Akella, and Y. Zhang (2007). Incorporating Diversity and Density in Active Learning for Relevance Feedback, in *ECIR 2007*. 36

Z. Xu, K. Yu, V. Tresp, X. Xu, and J. Wang (2003). Representative sampling for text classification using support vector machines, in *Advances in Information Retrieval 2003*. 38

R. Yan, J. Yang, and A. Hauptmann (2003). Automatically labeling video data using multi-class active learning, in *ICCV 2003*. 36, 84

S. Yan and H. Wang (2009). Semi-supervised Learning by Sparse Representation, in *ICDM 2009*. 33

B. Yang and S. Chen (2010). Sample-dependent graph construction with application to dimensionality reduction, *Neurocomputing*, vol. 74(1-3), pp. 301–314. 32

L. Yang (2006). Distance Metric Learning : A Comprehensive Survey, Technical report, Michigan State University. 29

L. Yang, S. Hanneke, and J. Carbonell (2012a). A theory of transfer learning with applications to active learning, *ML*. 36

L. Yang, R. Jin, and R. Sukthankar (2007). Bayesian active distance metric learning, in *UAI 2007*. 36, 90

L. Yang, R. Jin, R. Sukthankar, and Y. Liu (2006). An efficient algorithm for local distance metric learning, in *AAAI 2006*. 30

X. Yang, X. Bai, S. Köknar-Tezel, and L. J. Latecki (2012b). Densifying Distance Spaces for Shape and Image Retrieval, *J Math Imaging Vis*. 28, 117

J. Ye, Z. Zhao, and H. Liu (2007). Adaptive Distance Metric Learning for Clustering, in *CVPR 2007*. 31

J. Yu, D. Tao, and M. Wang (2012). Adaptive hypergraph learning and its application in image classification., *Trans on Image Processing*, vol. 21(7), pp. 3262–72. 33

S. R. Zaki and R. M. Nosofsky (2007). A high-distortion enhancement effect in the prototype-learning paradigm: Dramatic effects of category learning during test, *Memory & Cognition*, vol. 35(8), pp. 2088–2096. 122

S. R. Zaki, R. M. Nosofsky, R. D. Stanton, and A. L. Cohen (2003). Prototype and exemplar accounts of category learning and attentional allocation: a reassessment., *Journal of experimental psychology. Learning, memory, and cognition*, vol. 29(6), pp. 1160–73. 1

R. Zass and A. Shashua (2008). Probabilistic graph and hypergraph matching, in *CVPR 2008*. 33

L. Zelnik-Manor and P. Perona (2004). Self-Tuning Spectral Clustering, in *NIPS 2004*. 22

K. Zhang, J. T. Kwok, and B. Parvin (2009). Prototype vector machine for large scale semi-supervised learning, in *ICML 2009*. 28

L. Zhang, C. Chen, J. Bu, D. Cai, X. He, and T. S. Huang (2011a). Active Learning based on Locally Linear Reconstruction, *PAMI*, vol. 33(10), pp. 2026–2038. 36, 77

X. Zhang and W. S. Lee (2006). Hyperparameter Learning for Graph Based Semi-supervised Learning Algorithms, in *NIPS 2006*. 33, 52, 101

Y. Zhang and D.-y. Yeung (2008). Semi-Supervised Discriminant Analysis using robust path-based similarity, in *CVPR 2008*. 31

Z. Zhang, J. Wang, and H. Zha (2011b). Adaptive Manifold Learning., *TPAMI*, pp. 1–14. 32, 52, 101

Z. Zhang, H. Zha, and M. Zhang (2008). Spectral Methods for Semi-supervised Manifold Learning, *CVPR*, (1). 26, 28

B. Zhao, F. Wang, C. Zhang, and Y. Song (2008a). Active model selection for graph-based semi-supervised learning, in *ICASSP 2008*. 33, 36, 52

W. Zhao, J. Long, E. Zhu, and Y. Liu (2008b). A scalable algorithm for graph-based active learning, in *Frontiers in Algorithmics 2008*. 36

D. Zhou, O. Bousquet, T. N. Lal, Jason Weston, and B. Schölkopf (2004a). Learning with Local and Global Consistency, in *NIPS 2004*. 4, 6, 7, 26, 27, 44, 90, 93, 101, 102

D. Zhou, J. Huang, and B. Schölkopf (2005). Learning from Labeled and Unlabeled Data on a Directed Graph, in *ICML 2005*. 44, 100, 107

D. Zhou, J. Huang, and B. Scholkopf (2006). Learning with Hypergraphs : Clustering , Classification , and Embedding, in *NIPS 2006*. 33

Z.-h. Zhou, K.-J. Chen, and Y. Jiang (2004b). Exploiting unlabeled data in content-based image retrieval, in *ECML 2004*. 36

J. Zhu, H. Wang, B. K. Tsou, and M. Ma (2010). Active Learning With Sampling by Uncertainty and Density for Data Annotations, *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18(6), pp. 1323–1331. 38, 69

X. Zhu (2006). Semi-supervised learning literature survey, Technical report, UW. 39, 49, 114

X. Zhu, Z. Ghahramani, and J. Lafferty (2003). Semi-supervised learning using gaussian fields and harmonic functions, in *ICML 2003*. 26, 27, 32, 36, 43, 52, 101, 107

X. Zhu, A. Goldberg, and T. Khot (2009). Some new directions in graph-based semi-supervised learning, in *ICME 2009*. 116

X. Zhu, T. Rogers, R. Qian, and C. Kalish (2007). Humans Perform Semi-Supervised Classification Too, in *AAAI 2007*. 25

CURRICULUM VITAE

Sandra Ebert

Education:	2012	**Max Planck Institute for Informatics, Saarbrücken, Germany**
		PhD student at Computer Vision and Multimodal Computing Group (D2) of Prof. B. Schiele
	2009–2011	**TU Darmstadt, Germany**
		PhD student at the Multimodal Interactive Systems Group of Prof. B. Schiele
	2003–2008	**TU Darmstadt, Germany**
		Diploma in Computer Science
		Diploma thesis: Dirichlet Process Mixture Models for Object Categorization, supervised by Michael Stark, Mario Fritz, and Bernt Schiele.
Experience:	2010–	PhD representative of D2, Max Planck Institute for Informatics, Saarbrücken.
	2010–	Reviewer for Image and Vision Computing, Pattern Recognition, and ECCV
	2009–2010	Senator and member of Universitätsversammlung, TU Darmstadt.
		Member of search committee for Software Engineering, TU Darmstadt.
	2009	Teaching assistent, Machine Learning II, TU Darmstadt.
	2008	Member of search committee for professorship Intelligent Systems, TU Darmstadt.
Honors:	2009	DAGM - Young Researcher's Forum, presentation of outstanding achievements during diploma thesis
	2008	IBM - EMEA Best Student Recognition Event

PUBLICATIONS

Semi-Supervised Learning on a Budget: Scaling up to Large Datasets
Sandra Ebert, Mario M. Fritz, Bernt Schiele
In Asian Conference on Computer Vision (**ACCV**), Daejeon, 2012

Active Metric Learning for Object Recognition
Sandra Ebert, Mario M. Fritz, Bernt Schiele
In Pattern Recognition DAGM'12-Symposium (**DAGM**), Graz, 2012

RALF: A Reinforced Active Learning Formulation for Object Class Recognition
Sandra Ebert, Mario M. Fritz, Bernt Schiele
In IEEE Conference on Computer Vision and Pattern Recognition (**CVPR**), Providence, 2012

Pick your Neighborhood – Improving Labels and Neighborhood Structure for Label Propagation
Sandra Ebert, Mario M. Fritz, Bernt Schiele
In Pattern Recognition DAGM'11-Symposium (**DAGM**), Frankfurt, 2011

Weakly Supervised Recognition of Daily Life Activities with Wearable Sensors
Maja Stikic, Diane Larlus, Sandra Ebert, Bernt Schiele
IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33 (12), pp. 2521-2537, 2011

Extracting Structures in Image Collections for Object Recognition
Sandra Ebert, Diane Larlus, Bernt Schiele
In European Conference on Computer Vision (**ECCV**), Crete, 2010

D'une collection d'images a sa structure semantique, vers un processus automatique
Diane Larlus, Sandra Ebert, Bernt Schiele
In Reconnaissance des Formes et Intelligence Artificielle (**RFIA**), Caen, 2010