PARTIALLY SUPERVISED LEARNING OF MODELS FOR VISUAL SCENE AND OBJECT RECOGNITION

Dissertation
zur Erlangung des Grades eines

DOKTORS DER INGENIEURWISSENSCHAFTEN

der Technischen Universität Dortmund
an der Fakultät für Informatik

von

RENE GRZESZICK

Dortmund

2018

Tag der mündlichen Prüfung:	19.07.2018
Dekan:	Prof. Dr.-Ing. Gernot A. Fink
Gutachter:	Prof. Dr.-Ing. Gernot A. Fink
	Prof. Dr. Simone Frintrop

Maybe I'll start with the acknowledgements ...
- Mike Slackenerny, PhD Comic 149, 05.11.1999

ACKNOWLEDGEMENT

I tend to describe myself as a rather ambitious and at times tense person and I hate nothing more than loosing or giving up. This being said, I have to admit that writing a PhD thesis is probably the toughest thing I have done so far. I thought about giving up several times, but I am happy that I never did. I have been working on this thesis for a time of roughly five to six years and looking back it has been a hell of a ride. I (co-)authored twenty scientific papers, collaborated with industry partners and did research on many topics, including a DFG funded project which resulted in this thesis. None of this would have been possible without the help of all the great people around me.

First and most importantly, I would like to thank Prof. Gernot A. Fink. Not only for giving me the chance to do research under your supervision, but also for your encouragement to work on a DFG proposal. I always had the possibility to follow my own ideas and discuss them with you, which always resulted in valuable input for my research. Also thank you for all the valuable feedback on this thesis. Another big thank you to Prof. Simone Frintrop for agreeing to be my second reviewer and the valuable feedback on the last part of this journey. Many thanks to my project partners, Prof. Christian Wöhler and Thorsten Wilhelm for the collaboration in the DFG project. Great thanks also to Prof. Kristian Kersting for all the helpful discussions about research and life in academia. You have been a great mentor.

I would further like to thank my colleagues Axel Plinge, Leonard Rothacker, Sebastian Sudholt, Julian Kürby, Fernando Moya Rueda and Eugen Rusakov for being great office mates. Most notably I would like to thank Axel and Leonard for making our office a good working and at times non-working environment. Sometimes the office has been our home, but then again it also felt like home.

I thank my parents for always being there for me and supporting me. Many thanks also go to all my friends for their advice and support over the last years. Special thanks to Malte Fittkau, Sharon Duine and Natalie Jäschke for always putting things in perspective and making me not forget about life itself. I also have to thank the *Promotions-Studenten-Kolloquium* (Katharina Behring, Martin Bring, Tim Dellbrügger, Sebastian Homann, Dawid Kopetzki, Fabian Peternek, Nils Vortmeier, Thomas Zeume and many more) for designing the best day of the week and organizing a great colloquium. Last but not least, thanks to Leonard Rothacker, Manuel Allhoff, Julian Kürby, Thorsten Wilhelm for the valuable feedback on this thesis.

CONTENTS

AKRONYMS

AED acoustic event detection.

AP average precision.

BoAW Bag-of-Audio-Words.

BoF Bag-of-Features.

BoW Bag-of-Words.

CBA cluster-based annotation learning.

CNN Convolutional Neural Network.

CRF conditional random field.

DNN Deep Neural Network.

DPM deformable part-based model.

DSP digital signal processor.

EM expectation maximization.

FV Fisher Vector.

GFCC gammatone frequency cepstral coefficients.

GMM Gaussian mixture model.

GPU graphics processing unit.

HMM hidden Markov model.

HoG Histogram of oriented gradients.

LBP Local Binary Pattern.

LLGC learning with local and global consistency.

LSI latent semantic indexing.

mAP mean average precision.

MDS multidimensional scaling.

MFCC Mel frequency cepstral coefficients.

MLP multi layer perceptron.

NMF non-negative matrix factorization.

PBA iterative partition-based annotation learning.

PCA principle component analysis.

R-CNN Regional Convolutional Neural Network.

RBA retrieval-based annotation learning.

RF Random Forest.

RNN Recurrent Neural Network.

RPN Region Proposal Network.

SIFT Scale Invariant Feature Transform.

SVD singular value decomposition.

SVM Support Vector Machine.

VLAD Vector of Locally Aggregated Descriptors.

Acoustic Event Detection the task of determining the type and temporal extend of a sound class in an audio stream.

Computer Vision is the field of research that enables machines to understand the content of images.

Deep Learning describes the field of pattern recognition where multiple representations are learned on top of each other, typically using neural networks.

Image Classification is the task of assigning a (class) label to an image.

Object Classification is the task of recognizing the object that is displayed in an image or an image crop. It is thus a specialization of Image Classification.

Object Detection is the task of recognizing the type and position of an object instance within an image..

Scene Classification is the task of recognizing the scene that is displayed in an image and assigning a scene label to said image. It is thus a specialization of Image Classification.

Semi-supervised Learning describes the part of machine learning where the learning is based on labeled as well as unlabeled samples.

Supervised Learning describes the part of machine learning where the learning is based on a set of labeled samples.

Unsupervised Learning describes the part of machine learning where the learning is solely based on data without any labels.

NOTATION & DEFINITIONS

A scalar is denoted by a small letter, i.e., x.

Given an index, i.e., i, the count of this index will be denoted by the corresponding capital letter, i.e., I.

A vector is denoted by a bold small letter, i.e., $\boldsymbol{x}$.

A matrix is denoted by a bold capital letter, i.e., $\boldsymbol{X}$.

Sets are denoted by bold capital calligraphic letters, i.e., $\boldsymbol{\mathcal{Q}}$.

Given two vectors, they are denoted by $\boldsymbol{x_1}$, $\boldsymbol{x_2}$.

Indices are typically denoted on the lower right and additional indices, e.g., for further states such as time, are denoted on the upper right.

The i^{th} entry of a vector $\boldsymbol{x}$ is therefore denoted by x_i.

The i^{th} entry of a vector $\boldsymbol{x}$ at time t is denoted by x_i^t.

The entries of a matrix are denoted by $\boldsymbol{X}_i$ and $X_{i,j}$ respectively

The set of all indices in a given dimension be denoted by a dot, e.g., $\boldsymbol{X}_{\cdot,j}$

General	
b, B	A bin index and number of bins (e.g. for HoGs or bins in the feature augmentation)
c, C	Class index and the total number of classes
$d(\cdot,\cdot)$	A distance function
$f(\cdot)$	An arbitrary function
$\boldsymbol{h}$	A histogram
i, j, k	Indices
l, L	Index for a layer of level of detail and corresponding number (e.g. in a deep neural network or a spatial pyramid)
m, M	Index for image width and image width
n, N	Index for image height and image height
p, $P(x)$, $P(x\|c)$	Probabilities
$q(\cdot)$	A quantization function
r	A radius
t	A time value
$\boldsymbol{V}$, $\boldsymbol{v}$	A codebook of a Bag-of-Features representation and a single codebook entry
$\|\boldsymbol{V}\|$	The number of centroids in a codebook
w, $\boldsymbol{w}$	A single weight (scalar) and a weight vector
x, $\boldsymbol{x}$, $\boldsymbol{X}$	A single numerical feature, a feature vector and a feature matrix
y, $\boldsymbol{y}$, $\boldsymbol{Y}$	A single label, a label vector and a label matrix (represented as numerical values)
α	An angle (e.g. between two feature vectors)
$\delta(\cdot)$	A helper function
η	A learning rate
ϵ	An error
$\kappa(\cdot,\cdot)$	A kernel function
μ	A mean value (i.e. a cluster centroid)
$\sigma, \boldsymbol{\Sigma}$	Variance and a covariance matrix
π	A prior

φ	An activation function
τ	A threshold
Ω_C	A problem set of C classes
$\mathcal{N}(\mu, \sigma)$	The normal distribution
$\boldsymbol{\rho}$	Pixel coordinates
ϑ	A scaling factor
χ	A neuron without an activation function
$\mathcal{I}$	An image
$\mathcal{Q}$	A set, e.g., representing permutations in a randomization test or a single partition within a dataset
$\boldsymbol{\mathcal{U}}, \boldsymbol{\mathcal{V}}$	Singular vectors

Evaluation

$\mathcal{B}$	A bounding box
$\mathcal{O}(\mathcal{B}_i, \mathcal{B}_j)$	The overlap between two Bounding Boxes
$\mathcal{P}$	Precision
$\mathcal{R}$	Recall
$\mathcal{F}_1$	F1-Score

Feature Augmentation

a, $\boldsymbol{a}$	An appearance feature and a feature vector (i.e. a SIFT descriptor)
s, $\boldsymbol{s}$	Spatial information and a spatial information vector that is appended to a feature vector $\boldsymbol{x}$ (e.g. 2d coordinates)
z, Z	Index for a class-wise codebook entry and number of centroids in the codebook
ω	Length of a classification window in ms
$\mathcal{G}$	Gammatone frequency cepstral coefficients (GFCC)
$\mathcal{M}$	Mel frequency cepstral coefficients (MFCC)
$\mathcal{L}$	Perceptual Loudness

Semi-supervised Scene Learning	
G	A helper matrix for computing an update in learning with local and global consistency
O	Double center matrix for multidimensional scaling
u, U	Index for a view, i.e. a feature representation, and the number of views

Zero-shot Object Prediction	
Q	Scene - Object relation matrix
$\mathfrak{n}$	A noun in a text sequence
$\mathfrak{o}$, $\mathfrak{O}$	An object index and number of objects
$\mathfrak{p}$	A preposition in a text sequence
$\mathfrak{r}$	A relation between a scene and an object
$\mathfrak{s}$, $\mathfrak{S}$	A scene index and number of scenes
$\mathfrak{t}$	A tuple of scene, relation, object
$\mathfrak{v}$	A verb in a text sequence

I also took the liberty of signing this contract for you.
You didn't really want your soul, right?
- Prof. Smith, PhD Comic 91, 26.10.1999

1 INTRODUCTION

The recognition of scenes and objects is a key aspect of many applications in the field of Computer Vision, which is concerned with the automatic understanding of images. Some of the most common examples are automatic tagging of images, for example, for social media sites, stock photo collections or digital audiovisual libraries [AXE]. A more specific example is a Visual Wikipedia as proposed in [Per10]. Visual recognition is also a key requirement for the guidance of robots and autonomous cars. In order to allow cars to drive autonomously they need to be aware of their environment. In the same way robots need to be aware of their surrounding in order to interact with it. There are many more fields of application that directly rely on the recognition of objects or are to some extent related to it, such as smart homes, surveillance, virtual reality or entertainment. Having a look at the development in these areas in recent years, it can be said that Computer Vision and related machine learning techniques have a growing impact on every day life [Par16, Cad17]. The development of algorithms which address the aforementioned tasks will, therefore, also be of growing importance in the next years.

A crucial aspect for many applications is that every machine learning approach requires annotated training samples which are representative for the given task. These samples are then used for learning a model that solves the given task. However, it is often difficult to obtain such samples and even more difficult to annotate them. In case of Computer Vision systems these training samples are typically annotated images. For example, a set of images showing cats and dogs with the respective labels would allow for training a recognizer that learns the visual difference between those two classes. The annotation effort is typically correlated with the granularity of the subsequent recognition. Following the example above, distinguishing images of cats and dogs would only require a label that states whether a cat or a dog is shown for each image. However, a detection that is also concerned with the location of cats or dogs within an image, requires a more detailed annotation of the position within the training images. While the availability of large image corpora, annotations and even models has been greatly improved in the last decade [PBE$^+$06, RDS$^+$15], these are usually limited to scientific applications. For many practical applications, the provided data is either not representative or the license prohibits their use in commercial applications.

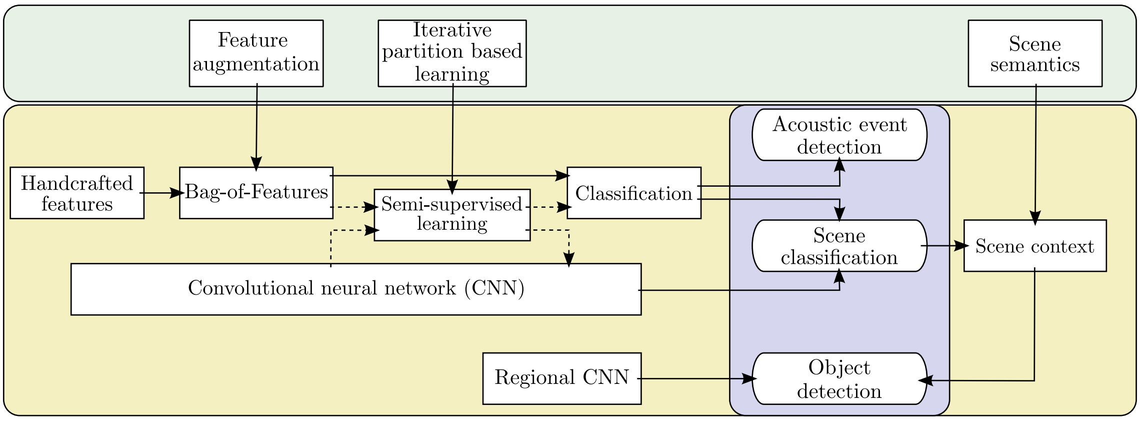


Figure 1.1: Overview of the proposed methods, their context and applications. (Yellow) Existing fields/methods; (Green) Contributions to the respective methods; (Blue) Applications for the presented contributions.

Hence, the requirement of training a new classifier that is also based on new training data and annotations remains for most practical applications.

The goal of this thesis is to provide methods that lower the annotation effort for visual scene and object recognition. It will be shown that learning a scene classifier can be done with only a small set of annotated images and that scene labels can be used for inferring knowledge about the objects in a given scene.

1.1 CONTRIBUTION

This thesis will present three major contributions to the fields of scene classification and object prediction which are illustrated in figure 1.1. Yellow building blocks are existing methods, with contributions to one of these methods symbolized in green. Blue are applications that are evaluated as a part of this thesis.

The contributions roughly follow a traditional pattern recognition pipeline: Given input data, features are extracted which are then aggregated in an intermediate representation such as the Bag-of-Features (BoF) and then classified. Here, the first contribution introduces a feature augmentation approach in order to incorporate more information at feature level. The result is a compact representation which is an important building block for training a classifier with minimal annotation effort. This will be discussed in two application contexts: scene classification and, as a side note, it will be shown that the same principle can also be applied to acoustic event detection. The second contribution is a novel method for learning a scene classifier in a semi-supervised manner. This semi-supervised approach is combined with the proposed BoF approach as well as features extracted from Convolutional Neural Networks (CNNs) which recently showed state-of-the-art performance in Computer Vision applications. Third, it will be shown that scene classification is an important step toward recognizing objects in a scene. In combination with both a supervised and a semi-supervised scene learning approach, a zero-shot approach to object prediction based on semantic information will be introduced.

Preliminary versions of each of these contributions have already been published at conferences or in journals. The respective publications are summarized in the following. All of these works were published under the supervision of Prof. Dr.-Ing. Gernot A. Fink.

Feature Augmentation:

1. R. Grzeszick, L. Rothacker, and G. A. Fink. Bag-of-Features Representations using Spatial Visual Vocabularies for Object Classification. In *Proc. International Conference on Image Processing (ICIP)*, 2013
2. A. Plinge, R. Grzeszick, and G. A. Fink. A Bag-of-Features Approach to Acoustic Event Detection. In *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014
3. R. Grzeszick, A. Plinge, and G. A. Fink. Temporal Acoustic Words for Online Acoustic Event Detection. In *Proc. German Conference on Pattern Recognition (GCPR)*, Aachen, Germany, 2015
4. R. Grzeszick, A. Plinge, and G. A. Fink. Bag-of-Features Methods for Acoustic Event Detection and Classification. *IEEE/ACM Transactions Audio, Speech and Language Processing*, 25(6), 2017

The first publication introduces the proposed feature augmentation approach and its application to image classification. A preliminary version of this idea has already been sketched in the author's master thesis in 2012 [Grz12]. The second work introduces a BoF approach to the task of Acoustic Event Detection, which is then extended by feature augmentation in the third publication. The application to the area of acoustic event detection has been joint work with Axel Plinge. This thesis will solely focus on the author's contribution regarding the feature augmentation. The works in the field of Acoustic Event Detection have been summarized in a journal publication in the IEEE Transactions on Audio, Speech and Language Processing in 2017.

In this thesis both applications will be evaluated. The original publications are extended by a more detailed evaluation, including parameters, design decision and datasets as well as a comparison with approaches to feature augmentation from the literature. It will be shown that by the proposed feature augmentation approach a compact, yet descriptive BoF representation can be computed.

Semi-supervised Learning:

5. R. Grzeszick and G. A. Fink. An Iterative Partitioning-based Method for Semi-supervised Annotation Learning in Image Collections. *International Journal of Pattern Recognition and Artificial Intelligence*, 2015

This work presents a semi-supervised learning approach for natural scene images that is integrated in a traditional pattern recognition pipeline including handcrafted features. It directly incorporates the idea of feature augmentation in order to derive a more compact representation for the semi-supervised learning algorithm. The proposed algorithm is the author's work. It is based on some key ideas of previous works introduced by Jan Richarz and Szilard Vajda. A summary of their works can be found in a joint publication:

> J. Richarz, S. Vajda, R. Grzeszick, and G. A. Fink. Semi-Supervised Learning for Character Recognition in Historical Archive Documents. *Pattern Recognition, Special Issue on Handwriting Recognition*, 47(3), 2014

In this thesis an extension incorporating features derived from CNNs will be presented. For deriving different sets of features from a CNN, ideas presented in

> R. Grzeszick, S. Sudholt, and G. A. Fink. Optimistic and Pessimistic Neural Networks for Object Recognition. In *Proc. IEEE International Conference on Image Processing*, 2017

will be picked up. This publication is also mainly based on the author's ideas, whereas the co-author Sebastian Sudholt contributed further evaluations and ideas for network architectures.

It will be shown that given the same amount of labels, the proposed semi-supervised learning algorithm outperforms the algorithms from the previous works as well as supervised classifier training for classifying natural scene images. Especially when training a deep network, which usually requires large amounts of labeled samples for training or even adapting the networks, the application of the proposed semi-supervised learning algorithm is beneficial.

Zero-shot object recognition:

6. R. Grzeszick and G. A. Fink. Zero-shot Object Prediction Using Semantic Scene Knowledge. In *Proc. International Conference on Computer Vision Theory and Applications (Visapp)*, 2017

This work shows a novel approach for zero-shot object prediction that is based on the relations between scenes and objects. The only requirement is training a recognizer for a scene label. The proposed approach is the author's work.

It will be shown that object presence can be predicted with a minimal human effort, as the only requirement is the availability of large text corpora as well as scene level annotations. While the aforementioned publication is based on a supervised learning approach, the work in this thesis extends the results of the paper by presenting a case where the scene classifier is learned in a semi-supervised manner. This allows for fine-tuning a deep neural network with only a few samples per class, which still obtains a high accuracy for scene classification and in conjunction with semantic relations between scenes and objects is able to predict the presence of object classes on a challenging dataset of natural scene images without having any visual information about any of the object classes. Furthermore, an application to object detection will be shown. It is shown, that the proposed approach can be applied for post-processing the predictions of an object instance detector that has been trained in a supervised manner. Thus, improving the prediction quality at the minimal cost of providing additional scene labels.

1.2 OUTLINE

The remainder of this thesis is organized as follows: chapter 2 will review the developments in scene and object recognition over the past years. The most important fundamentals up to the state-of-the-art will be introduced. This includes approaches for scene recognition which are typically concerned with the classification of a complete image as well as methods for object recognition which are often also concerned with the detection of the specific location of the object within the image. Finally, the influence of additional contextual information that can improve these recognition steps will be discussed. The last section of this chapter will introduce the most common tasks in these areas that the research community has focused on in the last years. This includes a detailed look at datasets and their evaluation protocols.

The following three chapters will each focus on one of the contributions. In chapter 3 the feature augmentation will be presented. It is discussed in the context of image classification and also includes a generalization to the domain of acoustic events. In chapter 4 an iterative partitioning-based method for semi-supervised learning of natural scene images is presented. In chapter 5 the zero-shot approach for predicting object within a scene image is discussed. Each chapter encloses a review of the related work, explains the details of the method and then ends with an evaluation and a short discussion. As the methods do at least partly build on the previous method, the evaluations will also build on the results from the previous methods.

The last chapter will summarize the work presented in this thesis, highlight the important contributions and close with a brief conclusion.

Research should be the sole focus of your life.
I expect you to eat, drink and sleep research.
- Prof. Smith, PhD Comic 784, 08.11.2006

2 SCENE AND OBJECT RECOGNITION

Almost all modern approaches for the recognition of scenes and objects are based on the principles of statistical pattern recognition. Given an annotated set of sample images, a statistical model is learned from the data in order to infer information about previously unknown images. In general, the tasks for scene and object recognition can be split up into two major categories. The first ones are typical classification tasks that process a complete image. The second ones are detection tasks that aim toward localizing and then classifying an entity within an image. Both of these tasks play an important role in the field of Computer Vision which has the goal to enable machines to understand the content of images. An overview of these tasks as well as the required fundamentals of statistical pattern recognition and their application in Computer Vision are explained in section 2.1.

Prominent examples for classification tasks are the recognition of natural scenes or the classification of objects which are already in the focus of the image so that the background is negligible. In both cases, a single label has to be assigned to an image. Similarly, the detection of the presence of a certain object in a scene is also often solved by the classification of a complete image. Traditionally, for each class a separate classifier is trained which outputs a probability that one or more instances of this object class are present in the image. Some of the most influential classification methods will be discussed in section 2.2.

Detection tasks follow the same principle as the object prediction tasks. However, for detection tasks it is not only important whether an object is present in a scene but also where it is located. Again, one detector is trained per object class. Most approaches return a set of bounding boxes indicating the location of different objects in a scene. The field of scene parsing has a rather special role in object detection. These approaches assigned a label to every pixel of its own, creating a very fine grained analysis of the scene. As a result object detection methods are more powerful and provide more detailed information about the image content, but they are also computationally more expensive [Sze10]. Recent developments in the field of object detection will be discussed in section 2.3.

All of the tasks described above can be improved by introducing contextual information. In some cases this is top down knowledge, which results from expert knowledge, but context information can also be learned from data. Typical examples include the co-occurrence of objects, objects occurring more frequently in some locations than in others, but also information from external modalities such as textual sources (cf. [Sze10]). In section 2.4 the role of context in classification and detection tasks will be discussed.

Finally, this chapter will close with an overview of the prominent tasks and benchmarks for scene an object recognition in section 2.5. The most important datasets for the evaluation of the various visual recognition tasks will be introduced. Typical evaluation metrics for assessing the performance of classification and detection systems will be explained.

2.1 FUNDAMENTALS

This section introduces important fundamentals of classification and detection in the context of Computer Vision applications. These are important prerequisites for the recognition of scenes and objects in images. Furthermore, different learning approaches based on labeled, unlabeled or partially labeled sample sets will be discussed.

2.1.1 *Classification*

A requirement for almost every classification task is the availability of an annotated dataset which contains samples for all classes of interest. Each sample needs to be associated with the respective class label. This dataset is called the training data and is used for learning the classification model. The training samples need to be representative for the given task, i.e., cover the variability that can be expected in the classification setting [DHS01]. After training, the classifier is, in principle, able to recognize an unknown sample and assign a class label to it. The classification capabilities are usually evaluated on an additional labeled dataset, the test set. Furthermore, in order to avoid optimizing meta parameters for either the features or the classifier on the test set, some datasets also provide an additional validation set for exactly that purpose [DHS01]. In Computer Vision, the dataset typically contains images which depict visually distinguishable classes. Typical examples are images showing certain objects or natural scenes. They may also include more complex scenes in which the classification is concerned with detecting the presence of a specific object class. Most classification tasks like the recognition of scenes or objects aim for the assignment of one class label per image and further require all classes to be known beforehand. These tasks are, therefore, often referred to as closed world tasks [SdSB13]. This is different in tasks such as presence detection where multiple classes can be associated with one image. In these settings every class is typically considered on its own. Thus, the set of classes can easily be extended making these more open tasks [SdSB13].

There are many approaches in order to solve these classification tasks. Following the traditional pattern recognition pipeline (cf. [Sze10]), three steps are performed: pre-processing, feature computation and finally the training or evaluation of a classifier. Pre-processing is concerned with improving the quality of the input image. The goal is to improve the quality with respect to the classification and remove unwanted variabilities, such as illumination changes or sensor noise. However, these steps are often heuristic operations which subjectively improve the quality of the input image. After pre-processing the images, a set of numerical features is derived from the training images. The set of features can either be designed by experts, often referred to as handcrafted features, or derived from the data in an unsupervised manner, i.e., by clustering. The goal is to derive a feature representation which is able to represent the difference between the classes but is invariant to other variabilities in the dataset. For example, for a car detector it is desirable that the features represent the shape differences between a car and a road, but not necessarily the color of the car. Based on the feature representations of the training samples, a classifier is trained. The classifier learns to distinguish the classes based on the given features. Hence, for a test image the same feature representation is computed and the classifier then assigns a class label to the image. Typical examples for classifiers include Support Vector Machines (SVMs), Random Forests (RFs) or the naive Bayes classifier (cf. [Sze10]). A rather special case are Deep Neural Networks (DNNs) or, in case of image recognition, Convolutional Neural Networks (CNNs) where the feature representation is learned in conjunction with the classifier (cf. [LBH15]). An overview of common feature representations and classification approaches will be given in section 2.2.

2.1.2 *Detection*

When considering detection tasks which not only assign a label, but also consider the location of an object within an image, the recognition task becomes more complex. Typically, a sliding window that is moved over the image is evaluated. At each position, features are computed and evaluated by a classifier with respect to the presence of a given object class in order to locate a single object instance. Some approaches also incorporate object proposals which generate a set of region candidates (cf. [Sze10]). These candidates are either predicted based on a prior learning phase or computed in an unsupervised manner based on heuristics (e.g. [GDDM16], [ADF10]). In order to be able to train and evaluate a detector, the annotations must be more fine-grained than for classification tasks. Hence, a labeled bounding box is required instead of only one class label per image. These bounding boxes need to be provided in the training and test set. The classifier is then trained based on the labeled bounding boxes and typically bounding boxes from the background in order to classify a single region with respect to the presence of an object. This classification is then verified in the test set based on the overlap between the annotated and the detected objects. Similar to the classification approaches, there are traditional pipelines that include handcrafted features which are used as input to a classifier [DT05, FGMR10]. Lately also CNN based approaches emerged into the field of object detection [GDDM16]. An overview of methods for detection will be given in section 2.3.

2.1.3 *Levels of Supervision*

The common setting where a classifier is trained based on a completely labeled training set is referred to as supervised learning. However, there are also tasks where the training samples are only partially labeled [Zhu06]. Hence, there is a large amount of additional unlabeled training samples available, which may be used in order to derive additional knowledge about the target domain. This is often the case in settings where labels are expensive to obtain or where it is easy to obtain large amounts of unlabeled samples. Images, for example, can be taken automatically without any labels, whereas assigning labels to the images requires additional manual effort. Training a classifier on only partially labeled datasets is in general referred to as semi-supervised learning. In some special cases it is also possible to request additional labels from a human annotator, the so-called human in the loop. Settings involving a human annotator are often referred to as active learning approaches. For tasks like object detection, there are also approaches where the detail of the annotations is lower for the training set than for the required output of the system. For example, an object detector can be trained on images that are assigned a label which indicates the presence of the object within the image but not its location. The classifier is then required to extract the important information about these objects from the image during the learning phase. Such settings are referred to as weakly supervised learning (e.g. [DAF12, ZKL$^+$16]).

Note that there are also completely unsupervised approaches. For example, the data can be clustered so that similar samples are grouped. These methods do not require any labels, but in general these cannot assigning a semantically meaningful label to an unknown sample as the pattern recognition system has no knowledge about labels. Unsupervised methods are, however, often used as intermediate steps during training and then combined with supervised or semi-supervised learning (i.e. [OD11]) in order to classify samples with respect to semantically meaningful labels.

2.2 SCENE AND OBJECT CLASSIFICATION

The following section will introduce some of the most influential methods in the field of scene and object classification. All of these methods consider a complete image as the input for the classification system. Hence, either a scene or an object that is the focus of the image can be recognized or the presence of an object in a complex scene can be predicted. However, no information about an object's location is predicted.

Several traditional approaches which are based on handcrafted feature representations for image classification will be introduced. These features include global feature representations which are used as a representation for the complete image and then used as input for a classifier such as the GIST of a scene [OT06]. They also include representations which build on local features which only describe a part of an image like LBPs [OPM00], SIFT [Low99] or HoG [DT05] and which are then aggregated into a global description. The most prominent example that learns a representation based on a set of local features is the so-called Bag-of-Features (BoF) representation [OD11]. In a traditional pattern recognition pipeline, these feature representation are then processed by a classifier like Random Forests, SVMs or the naive Bayes classifier. Besides the traditional approaches, CNNs that learn a set of features from the images and have shown state-of-the-art results [KSH12] will be discussed.

2.2.1 *Feature Representations*

In the history of scene and object classification different feature representations have been employed. Traditionally, they were designed based on expert knowledge and, therefore, also referred to as handcrafted features. Many are biologically inspired by the understanding of the human brain and what humans do in order to understand the content of an image in every day life. In the following some of the most important feature representations will be reviewed.

2.2.1.1 *GIST*

The GIST of a scene is inspired by the human capability to recognize coarse information about an image within a few milliseconds or in a single glance [OT01, OT06]. For example, even when an image is blurred, image properties like *openness*, *expansion*, *naturalness* or *roughness* are easily recognizable without recognizing any objects or details of a scene. Most likely also a rough idea of the scene can be developed, as is illustrated in figure 2.1. In an attempt to model these capabilities of the human brain, global image descriptors, the so-called GIST descriptors were proposed.

The first GIST descriptor has been introduced in [OT01]. It has been shown that general image properties like *openness*, *expansion*, *naturalness* or *roughness* can be estimated based on spectral information that is coarsely localized within the image. Discriminant spectral templates have been introduced which represent different types of structures within the image. It is shown that these templates describe a scene's properties well. A subdivision of the image in a 4 × 4 grid (so-called windowed discriminant scene templates) is applied to

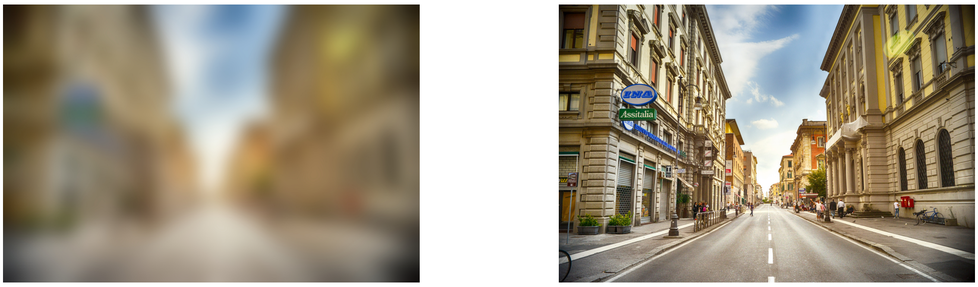


Figure 2.1: Idea of the GIST of a scene. Although the image on the left is heavily blurred and all details are missing, the human brain is able to figure out that a street scene is shown based on this short glance. Public domain *(CC0)* image: Italian street.

create a more detailed description of the scene content. The description of a scene based on these aforementioned scene properties is termed the spatial envelope.

In a later attempt, a color GIST descriptor has been introduced [DJS+09]. Besides different levels of detail, this descriptor also considers different color channels. The idea is to extract information from each channel of an RGB image independently. The image is subdivided in a 4×4 grid and orientation histograms are extracted for each color channel in each of the cells. In order to capture different levels of detail, the image is resized to different scales for which a squared image is expected. At each scale the orientation histograms are binned with a different level of detail. In [DJS+09] three scales ranging from 32 to 128 pixels with $4, 8$ and 8 orientation bins are used, which results in a 960 dimensional image descriptor.

Such global representations can be used in order to train a classifier for recognizing a complete image. The GIST descriptors of different images can also be compared using a distance measure, like the Euclidean distance, in order to retrieve similar images. For example, in [OT01, DJS+09] the task of retrieving images depicting a similar scene is addressed. In [XHE+10], scene images are classified into one of 397 scene categories based on local and also global image representations.

2.2.1.2 *Local Binary Patterns*

Local Binary Patterns (LBPs) are pixel-wise local features and perhaps the simplest form of an edge descriptor. The traditional approach compares a pixel with each of its eight neighbors [OPH96]. The comparisons can be interpreted as a binary pattern where a lower or equal intensity compared to the center pixel equals to 0 and a higher gray value intensity equals to 1. This binary pattern can then be interpreted as one decimal number as there are at most $2^8 = 256$ binary patterns. An example of this process is given in figure 2.1. The advantage of this interpretation is that it allows to describe a region or a complete image by a histogram of local binary patterns.

While this descriptor is not the most powerful one and quite sensitive to local noise, e.g., JPEG compression artifacts, it has two advantages: it is comparably robust against the general intensity level of the image (i.e. a bright or dark image). Furthermore, it is computationally very efficient. It only requires intensity comparisons and integer arithmetic.

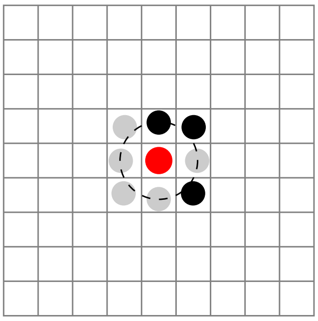


Figure 2.2: Example of an LBP descriptor computed at the red pixel: (bright gray) lower intensity, (black) higher intensity. The resulting binary pattern is **1 1 0 1 0 0 0 0** which results in the decimal value *208*.

There are several extensions to this method: rotation invariant binary patterns rotate the pattern so that it takes its minimal decimal value [OPM00]. Uniform binary patterns consider only those patterns with at most two transitions. In [OPM00] it has been shown that these account for most texture patterns. Therefore, all non-uniform patterns are represented by a single code reducing the number of different patterns to 58. Generalized binary patterns change the radius and number of points used for comparison, which however may require interpolation of intensity values and is, therefore, computationally more expensive [OPM00].

2.2.1.3 *SIFT & HoG*

For a long time, the most popular local feature representations were based on gradient information [MS05]. The Scale Invariant Feature Transform (SIFT) descriptor has been proposed in conjunction with a keypoint detector in [Low99]. A set of prominent points, which are preferably viewpoint invariant, are detected in an image. These are then described by their local neighborhood. The proposed neighborhood description has later on become known as the SIFT descriptor. The size of the neighborhood is typically dependent on the scale at which a keypoint has been detected and the descriptor is rotated in the direction of the main gradient.

The feature descriptor is illustrated in figure 2.3. A local image patch (thus a small area within the image) is subdivided into 4×4 cells. For each pixel within these cells, the local gradient is computed. The orientations of these gradients are then quantized into 8 bins of equal size and an orientation histogram is computed. Each orientation is weighted by the gradient's magnitude when accumulating the gradients into the histogram. Furthermore, an additional weighting factor is introduced that assigns a higher weight to pixels that are closer to the center of the descriptor. Finally, all 8 histogram bins from the 4×4 are concatenated which results in a 128 dimensional feature vector. In an evaluation of many different feature descriptors it has been shown that the SIFT descriptor is a very powerful and robust local feature descriptor for many applications [MS05]. As a result it is one of the most widely used local feature descriptors. In many applications the keypoint detection

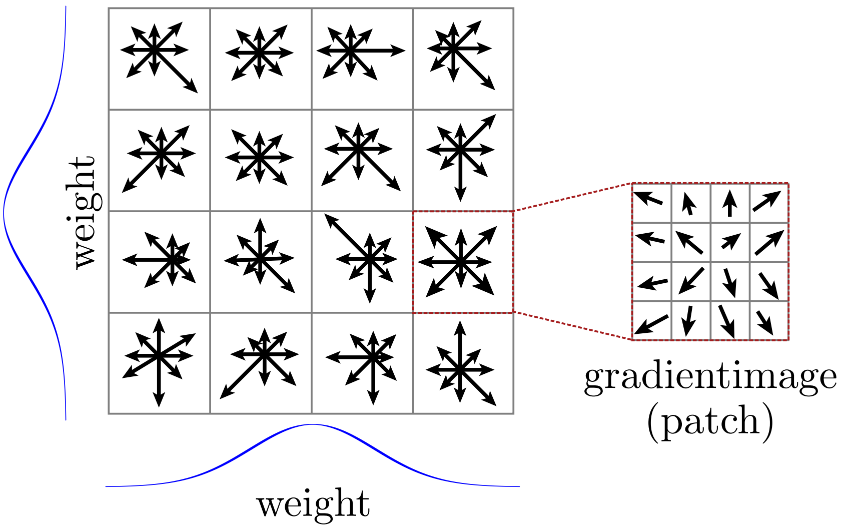


Figure 2.3: Example of a SIFT descriptor after [Low99]. Based on a gradient image – here symbolized by a 4 × 4px image patch on the right – binned gradient histograms are computed. Each cell in the descriptor considers small a patch of the gradient image. The gradients are weighted based on their distance to the descriptors' center – here symbolized by the weights along each axis in blue.

has been replaced by a dense grid where all descriptors are oriented in an upward direction and varying neighborhood sizes are computed [FFP05, LSP06, CLVZ11].

The Histogram of oriented gradients (HoG) descriptor can be seen as a generalization of the SIFT descriptor [DT05]. It subdivides a local image patch into a set of cells and computes the gradient orientations. The orientations are either in a range of $0° - 180°$ or $0° - 360°$ as for the SIFT descriptor. These orientations are also accumulated into a histogram based on evenly sized bins and weighted based on the gradient magnitudes. Groups of neighboring cells can be summarized into blocks which are then L2 normalized. Finally, the descriptors of all overlapping blocks are concatenated for the final feature representation. HoG descriptors exist with different parameterizations changing the number of orientations, cells or blocks. As they were introduced for person and object detection, the setup can be adjusted with respect to the objects' shape (cf. [DT05, FGMR10]). An example of a HoG descriptor using fine or coarse cells is shown in figure 2.4.

These gradient based local feature descriptors are very powerful descriptors that store many details about a local neighborhood. Originally, these were used in order to match each descriptor separately to its counterpart, i.e., in an image showing the same object from a different viewpoint [Low99]. However, this is computationally expensive. More interestingly, these gradient based features descriptors are often the basis for learned intermediate representations such as the Bag-of-Features (BoF) representation.

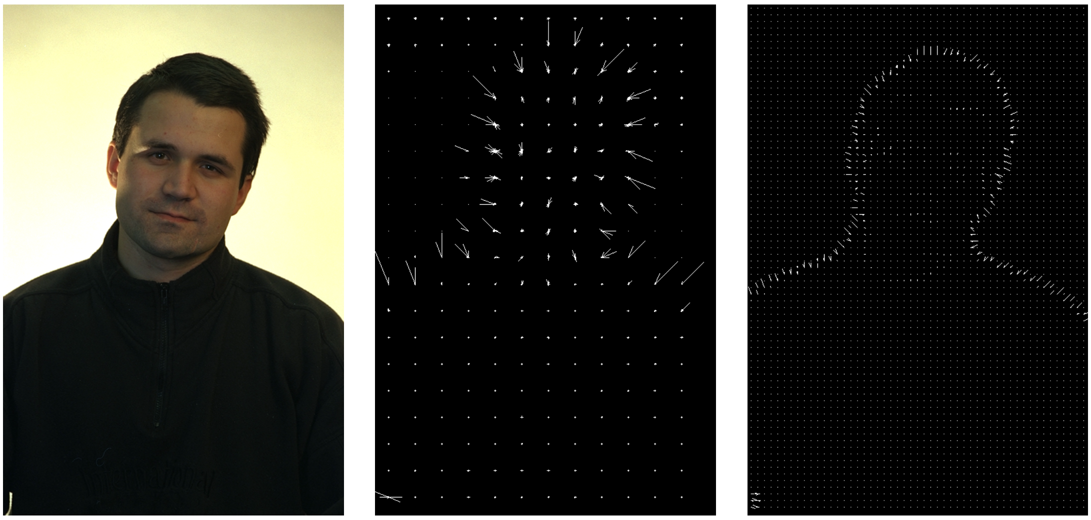


Figure 2.4: Example of a HoG descriptor. (left) original image from the Feret database [PMRR00]. (middle) coarse HoG cells. (right) smaller HoG cells showing finer details.

2.2.2 *Learned Representations*

In contrast to handcrafted feature representations where expert knowledge is used in order to create a feature representation that describes an image or a region in an image, it is also possible to learn a feature representation from a given dataset. This can either be done unsupervised (i.e. by clustering) or in a supervised manner so that additional class information is included in the learning process. Prominent examples are Bag-of-Features (BoF) representations and Neural Networks. BoF representations learn an intermediate representation from a set of handcrafted features in an unsupervised manner. Neural Networks directly learn a set of features in conjunction with a classifier in an end-to-end system and typically in a supervised manner.

2.2.2.1 *Bag-of-Features*

The BoF representation originates from the Bag-of-Words (BoW) principle from document classification [BYRNO11]. Here, all words that occur in a document are mapped to their word stems (e.g. *walking* and *walked* to *walk*). Furthermore, stop words such as 'the' or 'is' are removed. Then the occurrences of the words in a document are counted and the resulting histogram of word counts, also referred to as a term-vector, is used for distinguishing different categories of documents. As this approach discards the ordering of the words it is referred to as a 'Bag' of words.

This principle was adapted to image classification by generalizing it to arbitrary features instead of words [VZ02, SZ03, WAC^{+}04, CDF^{+}04]. An overview of the BoF principle for image classification is given in [OD11]. A set of local image features is extracted from the training data and then these are clustered using a k-Means algorithm [Llo57, Mac67] in order to obtain an intermediate representation. This essentially replaces the well defined stemming process in the BoW representation. The advantage of this generalization is that

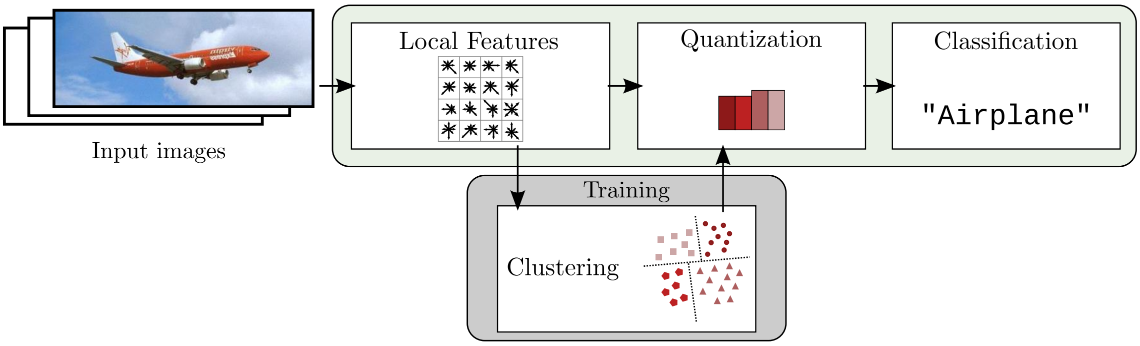


Figure 2.5: Overview of the BoF principle. Local features are computed (i.e. on a dense grid). The features from a set of training images are clustered, yielding a set of representatives, the so-called visual words. For a given image the local features are quantized with respect to the visual words. The occurrences of the different visual words are counted. Airplane image taken from the Caltech101 database [FFFP04].

the representatives that result from the clustering process are learned in an unsupervised manner. Using the analogy with the BoW principle, these representatives are often referred to as visual words and the complete set $\boldsymbol{V}$ as the visual vocabulary or codebook. Similar to using a certain dictionary in the BoW, the number of representatives $|\boldsymbol{V}|$ in the visual vocabulary is a free parameter.

After computing the codebook, for a given image the local image features are quantized. The occurrences of the different representatives are counted which results in a histogram, the BoF representation. The value for the histogram entry i is thus given by

$$h(i) = \frac{1}{K} \sum_k \delta_{\mathrm{hq}}(i, \boldsymbol{X}_k) \tag{2.1}$$

with

$$\delta_{\mathrm{hq}}(i, \boldsymbol{x}) = \begin{cases} 1 & \text{if } \operatorname*{argmin}_j d(\boldsymbol{V}_j, \boldsymbol{x}) = i \\ 0 & \text{otherwise,} \end{cases} \tag{2.2}$$

where $\boldsymbol{X}_k$ is a matrix containing all local feature descriptors with K being the number of descriptors computed in a given image and $d(\boldsymbol{V}_j, \boldsymbol{x})$ computes the distance between the centroid $\boldsymbol{V}_j$ and the feature vector $\boldsymbol{x}$. A typical distance measure is, for example, the Euclidean distance. The steps in the BoF pipeline are illustrated in figure 2.5. Arbitrary local feature descriptors can be incorporated in the BoF principle. Most commonly, the SIFT descriptor is used (cf. [MS05]).

There are two different approaches for computing the set of local features. For retrieval tasks that aim toward finding similar images of one exact instance (e.g. a building or painting) it is often beneficial to use an interest point detector [PCI$^+$07]. The detector allows to find very specific regions at different scales and viewpoints and, therefore, allows to find similar images for a single instance. For classification tasks it is more beneficial to use a dense grid, which implicitly assumes that the whole image is of interest. It provides a better generalization over a set of images belonging to one class as the higher number of

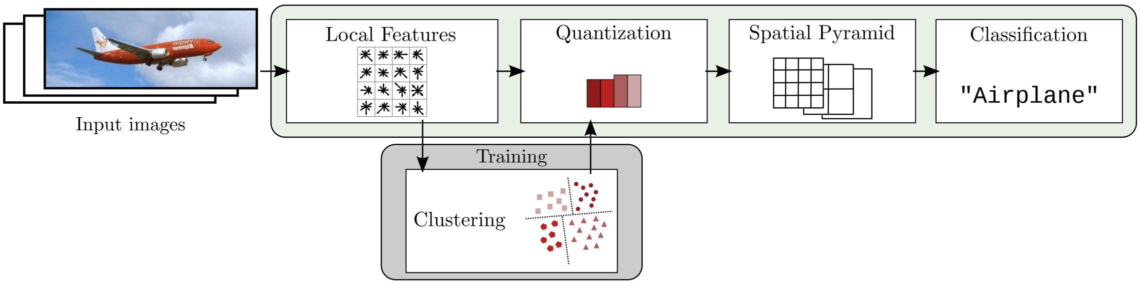


Figure 2.6: Overview of the BoF principle including spatial pyramids. The image is subdivided in different tiles in a quadtree like manner forming different scales. A BoF representation is computed for each tile. Finally, the resulting histograms are concatenated into a single feature vector. Airplane image taken from the Caltech101 database [FFFP04].

descriptors compared to an interest point detector allows for obtaining more information about the given class [FFP05]. A class is usually not only described by very specific details, but also by more uniform areas that are not covered by an interest point detector. Since the number of features that are computed greatly varies based on the image size, especially when using a grid, the counts in the histogram are often normalized into relative frequencies.

Spatial Information

A major drawback of the BoF principle is that all spatial information about the image is lost when computing the histogram. This is overcome by the spatial pyramid principle [GD05, LSP06] as shown in figure 2.6. In this approach an image is subdivided in a quadtree like manner creating a set of tiles at multiple scales. A BoF representation is computed for each of these tiles. The influence of a single local feature descriptor is often weighted based on the scale so that finer scales have more influence [LSP06]. Finally, the histograms of all tiles are concatenated into a single feature vector.

This principle is based on the assumption that the object of interest or the view of the scene is roughly centered and that the orientation is important. In practice this usually works well mostly due to the so-called photographer bias (a human photographer usually centers a shot on the object of interest; cf. [SS13]) but also due to the fact that the orientation of real world objects rarely changes, e.g., a car is rarely upside down.

Feature Quantization and Encoding

Besides the loss of spatial information, the quantization of features with respect to the visual vocabulary also discards information from the data. This is both an advantage and a disadvantage. It allows for generalization, but several details that have been described by the local feature vectors are lost. In order to overcome this limitation several approaches that allow for quantizing or encoding the data in a more meaningful manner than hard quantization have been proposed:

Soft quantization avoids the hard assignment to one centroid (cf. [CLVZ11]). A Gaussian mixture model (GMM) is fitted to the set of training features using the expectation maximization (EM) algorithm (cf. [DHS01]) instead of clustering the local features using a k-Means algorithm. This results in a mixture model of $|\boldsymbol{V}|$ components with $\boldsymbol{V}_i = (\boldsymbol{\mu_i}, \boldsymbol{\Sigma_i})$ being a tuple of mean and covariance. For each local feature vector the posterior distribution over the GMM's components is computed. The distributions for each local feature are then summed up in order to derive a histogram representation so that

$$h(i) = \frac{1}{K} \sum_k \delta_{\mathrm{sq}}(i, \boldsymbol{X}_k) \tag{2.3}$$

with

$$\delta_{\mathrm{sq}}(i, \boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu_i}, \boldsymbol{\Sigma_i}) / \sum_j \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu_j}, \boldsymbol{\Sigma_j}) \quad . \tag{2.4}$$

Each local feature is, therefore, softly assigned to the GMM's components instead of a single centroid.

The *Vector of Locally Aggregated Descriptors (VLAD)* encoding computes a hard assignment to one cluster center, but then instead of simply assigning the sample to the cluster, it encodes the distance to the centroid [JDSP10]. For each dimension of the local feature descriptor the element-wise distance is added to the VLAD representation. Hence, a detailed local statistic is computed by

$$\boldsymbol{h}(i) = \frac{1}{K} \sum_k \delta_{\mathrm{vq}}(i, \boldsymbol{X}_k) \tag{2.5}$$

with

$$\delta_{\mathrm{vq}}(i, \boldsymbol{x}) = \begin{cases} \boldsymbol{V}_j - \boldsymbol{x} & \text{if } \operatorname*{argmin}_j d(\boldsymbol{V}_j, \boldsymbol{x}) = i \\ 0 & \text{otherwise.} \end{cases} \tag{2.6}$$

For local feature descriptors with dimension $|\boldsymbol{x}|$, a $|\boldsymbol{V}| \times |\boldsymbol{x}|$ dimensional feature representation is created.

The *Fisher Vector (FV)* encoding combines the idea of VLAD encoding with soft quantization [PSM10]. Again a GMM is fitted to the data using the EM algorithm. Then, for each local feature the posterior distribution is computed. The difference between the mean value of a component and the local feature is weighted based on the posterior distribution so that

$$\boldsymbol{h}(i) = \frac{1}{K} \sum_k \delta_{\mathrm{fq}}(i, \boldsymbol{X}_k) \tag{2.7}$$

with

$$\delta_{\mathrm{fq}}(i, \boldsymbol{x}) = \delta_{\mathrm{sq}}(i, \boldsymbol{x}) \; (\boldsymbol{V}_i - \boldsymbol{x}) \quad . \tag{2.8}$$

The FV encoding also computes a $|\boldsymbol{V}| \times |\boldsymbol{x}|$ dimensional feature representation. Both, the FV and the VLAD representation store a statistic of the variance between the features represented by a codebook element rather than storing only the visual word occurrences.

Supervised codebook learning trains the codebook with additional class information [LR09, PGF14]. In [PGF14] the GMM approach for soft quantization is extended to a supervised learning approach. Codebooks are learned for the samples of each class independently and then concatenated in order to form a super codebook. In [LR09] it is proposed to train codebooks for BoF representations in a supervised manner based on an information loss minimization. Moreover, the posterior distribution of the classes is also encoded for each centroid of the codebook which in turn allows for inferring labels for unknown samples.

Classification of BoF Histograms

After computing BoF histograms these can be used as input to a classifier or the histograms can be compared using a distance measure, i.e., in retrieval tasks. For classification the most common choice is using an SVM [CLVZ11], but in principle every classifier is applicable. As the BoF representations became more high dimensional, linear SVMs became often already powerful enough to distinguish different classes. Basically, most BoF representations are both sparse and high dimensional so that a kernel SVM, which computes a scalar product between feature vectors in a high dimensional space, can be avoided.

Nevertheless, it could be shown that kernel SVMs may be beneficial even in very high dimensional spaces [VGVZ09]. There are a few popular choices for non-linear kernels. One choice is the histogram intersection kernel [GD05, LSP06] that evaluates the difference of two histograms $\boldsymbol{h_i}$ and $\boldsymbol{h_j}$ by

$$\kappa_{hi}(\boldsymbol{h_i}, \boldsymbol{h_j}) = \sum_k min(h_{i,k}, h_{j,k}) \quad . \tag{2.9}$$

In [VGVZ09] different kernels, including the χ^2 kernel and the Hellinger kernel were compared. The former one evaluates the kernel function

$$\kappa_{\chi^2}(\boldsymbol{h_i}, \boldsymbol{h_j}) = \sum_k \frac{2\, h_{i,k}\, h_{j,k}}{h_{i,k} + h_{j,k}} \tag{2.10}$$

which is shown to work well for histogram data. The latter one evaluates the kernel function

$$\kappa_{hellinger}(\boldsymbol{h_i}, \boldsymbol{h_j}) = \sum_k \sqrt{h_{i,k}\, h_{j,k}} \quad . \tag{2.11}$$

The Hellinger kernel is named after the corresponding distance metric which can be formulated as the Euclidean norm of the square root vectors

$$d_{hellinger}(\boldsymbol{h_i}, \boldsymbol{h_j}) = ||\sqrt{\boldsymbol{h_i}} - \sqrt{\boldsymbol{h_j}}|| \quad . \tag{2.12}$$

This turns out to be a huge technical advantage as the square rooting of two histograms can be pre-computed so that a linear SVM can be used.

Distance Measures for Comparing BoF Histograms

Besides the traditional classification setup, where BoF representations are classified by an SVM, these representations can also be directly compared using a distance measure. Several distance or similarity metrics have been applied in the literature [SZ03, PCI+07, SF15].

These include the Manhattan or the Euclidean distance, the cosine similarity, which computes the cosine of the angle α between two vectors

$$d_{cos}(\boldsymbol{h_i}, \boldsymbol{h_j}) = \cos(\alpha) = \frac{\boldsymbol{h_i}\boldsymbol{h_j}}{||\boldsymbol{h_i}||\,||\boldsymbol{h_j}||} \tag{2.13}$$

or the Bray-Curtis distance

$$d_{BC}(\boldsymbol{h_i}, \boldsymbol{h_j}) = \frac{\sum_k |h_{i,k} - h_{j,k}|}{\sum_k h_{i,k} + h_{j,k}} \quad . \tag{2.14}$$

In contrast to the also well known χ^2 distance, the Bray-Curtis distance also accounts for entries where both histogram bins at the same index equal zero (cf. [SF15]).

Each of these metrics and kernels are good candidates for evaluating discrete distributions such as BoF representations. In practice a good choice is often determined empirically and based on informal experiments. Most works showed that there is a strive for simplicity in order to keep the computational costs for very high dimensional data comparably low [VGVZ09, CLVZ11].

2.2.2.2 *Neural Networks*

While the BoF principle learns an intermediate representation based on handcrafted features, Deep Neural Networks (DNNs) allow for learning a classifier and a set of feature representations in a supervised manner directly from the input image. In Computer Vision, the most common DNN architectures are CNNs. Their structure allows for learning different levels of abstractions from simple features like colors or orientations to motifs and combinations of these motifs that form complex structures. The last of these help identify different classes. Approaches that are based on DNNs showed state-of-the-art results in scene and object classification [LBH15].

The Perceptron and Multi-layer Perceptrons

The development of Neural Networks dates back to the invention of the perceptron [Ros58]. Similar to other successful approaches in Computer Vision, the perceptron is biologically inspired. It is a simplified mathematical model of the neurons in the human brain. Put very simply, the human brain consists of a large set of neurons. Each of these neurons is connected to many other neurons. Once a neuron is active, it fires by sending an impulse through it's *axon* to it's *boutons*. These boutons are then connected to thousands of other neurons. Each neuron collects and effectively sums all inputs from nearby neurons via its *dendrites*. If the resulting value is greater than a firing threshold, the neuron itself fires [Nie15]. Similarly, an artificial neuron, the so-called perceptron, takes a set of inputs x_i. Each value is multiplied with a weight w_i and the result is summed up. Furthermore, an additional bias w_0 is added. If the summation surpasses a threshold, the perceptron outputs one, otherwise the output is set to zero. The perceptron is illustrated in figure 2.7.

Many perceptrons can be combined in a layer so that they work in parallel [MP69]. When connecting several layers in a hierarchical architecture such that each perceptron in a given

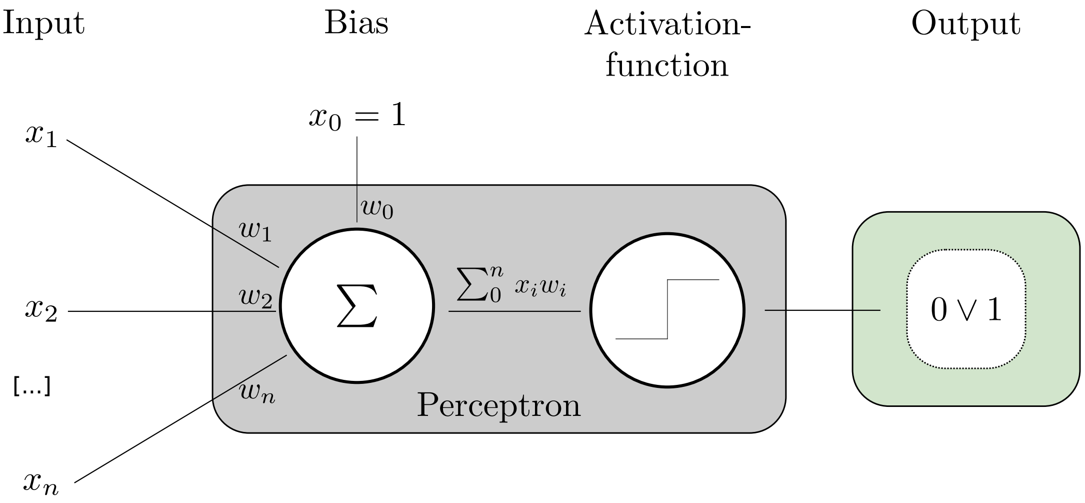


Figure 2.7: Sketch of the perceptron, following the idea of [Ros58]. Given an input vector $\boldsymbol{x}$ each value x_i is multiplied with the respective weight w_i and then summed up. Furthermore an additional bias w_0 is added. If the output surpasses a threshold, the output is one. Otherwise the output is zero.

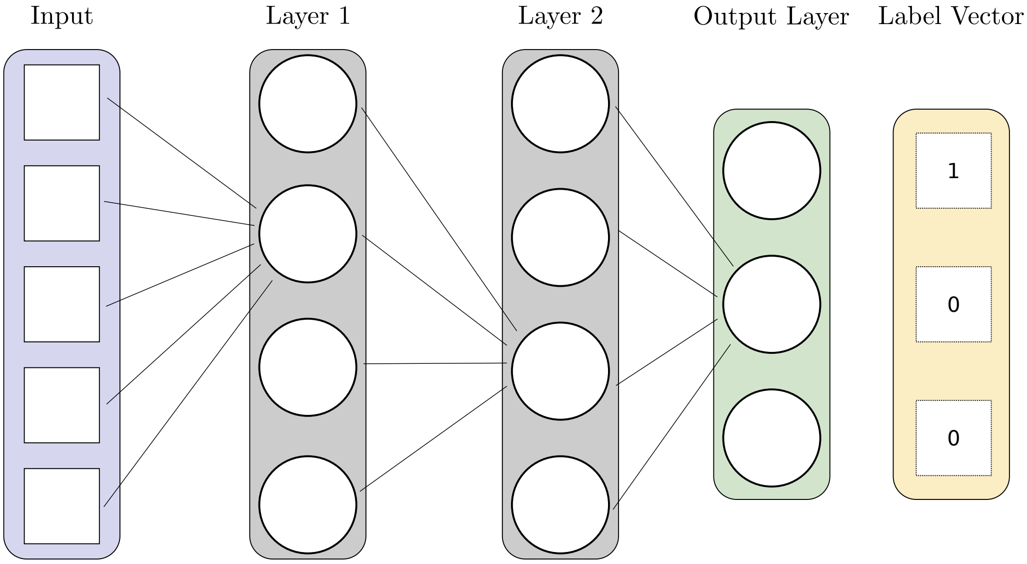


Figure 2.8: Sketch of a multi-layer perceptron, following the idea first mentioned in [MP69]. In practice, each neuron is connected to the output of each neuron in the previous layer. For the input layer, each neuron is connected to each input value. After evaluating the multi layer perceptron (MLP), the output, here for three classes, can be compared to a ground truth vector.

layer is connected to the output of the perceptrons in the previous layer, as illustrated in figure 2.8, a so-called multi layer perceptron (MLP) is constructed (cf. [Nie15]). The connection of many layers allows for approximating arbitrary (*Borel measurable*) functions [LBH15]. For image classification, the first layer can, for example, be connected to every pixel in a given image. The output layer is then comprised of C output neurons so that one neuron should be active for a given class c. This encoding is also known as a one hot encoding (cf. [Nie15]). Thus, forwarding the information of the input image through the network allows for a classification. This is also referred to as the forward pass (cf. [Nie15]).

Training Neural Networks

In order to understand the development of DNNs it is important to understand how training MLPs works. Let $\boldsymbol{x^l}$ be the output vector after the l^{th} layer of the network. The final output vector of an MLP with depth L is then $\boldsymbol{x^L}$. Given that the last layer implements a one hot encoding, an error between the output of a neural network and the desired output can be computed. When training a neural network, the goal is to minimize this error at the output of the network for a given training set. A typical measure for this error is, for example, the mean squared error:

$$\epsilon_{MSE} = \frac{1}{2}\sum_{i=1}^{C}(y_i - x_i^L)^2 \tag{2.15}$$

with $\boldsymbol{y}$ being a binary label vector where the annotated class is encoded as a one and all other class indices are set to zero. The backpropagation algorithm (cf. [Nie15]) uses this error and gradient descent for adapting the neural network's weights in order to produce the desired output for a given input. The output x_j^l of neuron j in layer l is computed by:

$$x_j^l = \varphi(\chi_j^l) \quad \text{with} \quad \chi_j^l = \sum_k x_k^{(l-1)} w_{kj} \tag{2.16}$$

where $x_k^{(l-1)}$ is the output of neuron k in the previous layer, w_{kj} is the weight between neuron k and j and φ is the activation function of the neuron. Computing the partial derivative $\frac{\partial \epsilon}{\partial w_{ij}}$, the error at the output can be partially accounted to each of the network's weights. Then each weight is updated using

$$w_{ij}^{(t+1)} = w_{ij}^t + \eta \frac{\partial \epsilon}{\partial w_{ij}} \tag{2.17}$$

where η is the so-called learning rate. The learning rate is a factor that is typically set to $\eta < 1$ so that the weights are slowly adapted in the direction of a local optimum. In order to solve the partial derivative, the chain rule can be applied:

$$\frac{\partial \epsilon}{\partial w_{ij}} = \frac{\partial \epsilon}{\partial x_j^l}\frac{\partial x_j^l}{\partial \chi_j^l}\frac{\partial \chi_j^l}{\partial w_{ij}} \quad . \tag{2.18}$$

Taking a closer look at the three terms, the third one can be solved by

$$\frac{\partial \chi_j^l}{\partial w_{ij}} = \frac{\partial}{\partial w_{ij}} \sum_k x_k^{(l-1)} w_{kj} = x_i^{(l-1)} \quad . \tag{2.19}$$

The second term,

$$\frac{\partial x_j^l}{\partial \chi_j^l} = \frac{\partial}{\partial \chi_j^l} \varphi(\chi_j) \ , \tag{2.20}$$

is dependent on the activation function φ, which in general requires the activation function to be differentiable. On a technical level it is sufficient that a gradient can be defined for every input value of the activation function. This also explains why a thresholding function is not a good choice for a networks' activation function as its derivative is mostly zero with a non-differentiable point at the threshold value. No error would be propagated backwards. As a result, differentiable functions like the sigmoid function

$$\varphi_{sig}(x) = \frac{1}{1 + e^{-x}} \quad \text{with} \quad \varphi'_{sig}(x) = \varphi_{sig}(x)(1 - \varphi_{sig}(x)) \tag{2.21}$$

have been used as an activation function [Nie15]. The last remaining term of equation 2.18 is a special case as it can be solved straight forward in the last layer where

$$\delta_j = \frac{\partial \epsilon}{\partial x_j^L} \tag{2.22}$$

is the derivative of the error function with respect to x_j^L. For example, for the mean squared error function

$$\delta_j = \frac{\partial \epsilon_{MSE}}{\partial x_j^L} = (y_j - x_j^L) \quad . \tag{2.23}$$

However, for an intermediate layer the error must be considered as a function of all inputs and, therefore, the subsequent layers. The term can be solved by

$$\frac{\partial \epsilon}{\partial x_j^l} = \frac{\partial \epsilon(\chi_0^{l+1}, \chi_1^{l+1}, \ldots)}{\partial x_j^l} \tag{2.24}$$

and thus for an intermediate neuron

$$\delta_j = \varphi'(\chi_j^l) \sum_k \delta_k w_{kj} \tag{2.25}$$

is also dependent on the error in the subsequent layer.

Before applying the backpropagation algorithm for training neural networks, the weights are typically randomly initialized [GB10, HZRS15b]. There are a few different approaches for the actual backpropagation algorithm. For example, the error can either be computed after a single forward pass or accumulated over many forward passes. In practice, stochastic gradient descent is often applied. It computes the gradient after processing a so-called mini-batch [LBH15]. It is assumed that this batch is a reasonably large representative of the training set and, therefore, approximates the statistics of the data. As the gradient is only an approximation, a so-called momentum is often used. When using momentum, the update of the network's weights in equation 2.17 is computed based on a weighted mean of the current gradient as well as the last gradient. Stochastic gradient descent with momentum is much more robust than using a single sample, which usually only represents a single class, and much more efficient than using the complete training set.

Neural networks of arbitrary size can be trained using the backpropagation algorithm. However, when scaling neural networks to deep architectures there are a few pitfalls that need to be considered when using backpropagation. These will be discussed in the following.

Deep Neural Networks

When stacking many layers in a multi-layer perceptron these became recently known as Deep Neural Networks (DNNs) [LBH15]. The field of research concerned with deep networks is termed deep learning. Although the definitions vary, one could even consider every network with more than three layers as a deep architecture. For comparison, BoF architectures extract a set of handcrafted features from the input image, then learn a representation on top of these features and finally a classifier [CLVZ11]. Including the classifier such a traditional architecture consists of three layers so that the networks with more than three layers are considered as a deep architecture. Note however that modern CNN architectures have eight [KSH12] to several hundred layers [HZRS15a].

For training these deep architectures there are a few challenges that need to be considered: First, the sigmoid activation function does not work very well in such deep architectures. The issue is that using the standard parameterization $\varphi'_{sig}(x) <= 0.25$ which means that when applying the chain rule to the sigmoid function as shown before, the gradient quickly becomes very small. This issue is known as the vanishing gradient [LBH15] and has been overcome by the ReLU activation [GBB11] function

$$\varphi_{relu}(x) = max(0, x) \ , \tag{2.26}$$

where the gradient is either one or zero, which avoids a vanishing gradient. For the non-differentiable point at zero the derivative is mostly just set to zero as well [GBB11]. Although ReLUs are mostly linear activation functions, the non-linearity in the network is generated by the number of paths through the network that can be active or inactive for a given input. The number of paths grows exponentially with each layer in the network. Thus arbitrary non-linear functions are simply approximated by an exponentially large number of linear functions [LBH15].

Second, the large number of parameters in such networks requires a large number of training samples, which in turn requires a lot of training time and in most cases also a tremendous labeling effort. The long training times have been overcome by recent advances in graphics processing unit (GPU) development which significantly reduced the computation time [Nvi15]. The availability of training images has been tackled by large efforts in the research community which led to the creation of large image corpora, like ImageNet [RDS+15] with millions of samples being available. These large datasets allow for training a network on several million samples and then using the weights from this training in order to adapt it to another, similar task with a smaller number of samples. This initial training process is also termed pre-training and the adaption step is also known as fine-tuning [RASC14].

Although the number of available training samples has been increased, one major issue is that with millions of parameters in deep networks they can easily overfit to the training data. They learn the training data perfectly but do not generalize to test data very well anymore. Hence, the third crucial aspect for deep neural networks has been applying regularization techniques. Two notable regularization techniques are dropout and data augmentation. Dropout sets the output of a given percentage of neurons in a layer to zero during training time. Thus removing them from the backpropagation and creating some variability in the training in order to avoid overfitting. This approach works well with ReLUs which already set the output of several neurons to zero [SHK+14]. Data augmentation describes the process of generating additional samples by modifying the original training data in order to reduce overfitting. In Computer Vision, examples for data augmentation are vertical mirroring, slight rotations of the image, adding Gaussian noise or taking random crops from the image [KSH12, SZ14].

Convolutional Neural Networks

The most prominent example of DNNs are Convolutional Neural Networks (CNNs) [LBD+89] which show state-of-the-art results in Computer Vision. These networks make use of weight sharing in order to reduce the number of parameters in the network. Each neuron is only connected to a small region of the previous layer. This input region to the neuron is moved over the layer in a sliding window approach, which resembles a convolution. This reduces the number of parameters dramatically compared to a multi-layer perceptron that is fully connected to the complete previous layer. It can be shown that the first layers of CNNs learn filters that represent edges and color blobs with the structures becoming more and more complex the deeper the network grows [ZF14]. In their architecture most CNNs follow a general concept:

- The first layers are convolution layers. In the first layer, the filters are moved over the image content and in the later layers over the previous layers' output feature maps. The later in the network the convolution is, the more and more complex structures are represented by each of the neurons [ZF14].

- Typically, the size of the resulting feature maps is reduced by pooling layers after a given number of convolutions. These layers apply pooling operations, e.g., max or average pooling to a small region in the feature map. For example, a small $N \times N$ region can be replaced by the largest activation within this region. This reduces the computational complexity and at the same time the network becomes invariant toward small translations and noise [SZ14, LBH15].

- The combination of convolution and pooling layers is then followed by fully connected layers, similar to the traditional MLP [KSH12, SZ14].

- For classification tasks, the last layer is typically using a one hot encoding [KSH12, SZ14, LBH15].

Instead of using arbitrary output values, the output of the last layer in most CNNs is represented as a pseudo-probability. For a one hot encoding, as used for classification tasks, the softmax function

$$f_{\text{softmax}}(x_i^L) = \frac{e^{x_i^L}}{\sum_j e^{x_j^L}} \quad \forall i \in [1, C] \tag{2.27}$$

is commonly used. The error is then computed by

$$\epsilon_{\text{softmax}} = -log(f_{\text{softmax}}(x_y^L)) \tag{2.28}$$

with y being the class label.

For an arbitrary binary target vector, each output value is processed by a sigmoid function. The error can then be computed by the cross-entropy loss

$$\epsilon_{CL} = -\frac{1}{N}\sum_{i=0}^{N} y_i \, log(x_i^L) + (1 - y_i) log(1 - x_i^L) \tag{2.29}$$

where $\boldsymbol{x^L}$ is the networks output and $\boldsymbol{y}$ a binary vector indicating the class label [Nie15]. Although, it is not an intuitive error function, like the mean squared error, the cross-entropy is always positive and tends toward zero if the networks' output approaches the desired output. It has been shown that in contrast to the mean squared error it is more stable with respect to the gradients [Nie15].

CNN architectures

Although most CNN architectures follow the concept described in the previous paragraph, there are some architectures that were especially influential. Two of the most influential ones are the Alexnet [KSH12] and the VGG networks [SZ14], which are sketched in table 2.1.

The AlexNet was the first network architecture that has successfully been applied to large scale object recognition tasks [KSH12]. Although all building blocks for the success of deep learning were already introduced, it was the first CNN that combined all techniques that are known to make deep learning successful. The architecture contains five convolution layers of the sizes 11 × 11, 5 × 5 and three 3 × 3 convolutions. The weight sharing allows for a deep architecture, which by the time it was first introduced still had to be split up to two GPUs [KSH12]. Furthermore, All neurons make use of ReLu activation functions. Dropout is employed in the fully connected layers and training has been done on a large dataset, the ImageNet [RDS$^+$15], using additional data augmentations in order to increase the variability and to avoid overfitting. Using this combination of techniques the AlexNet was the first deep network which outperformed traditional approaches on the ImageNet object recognition challenge by a large margin (cf. [RDS$^+$15]).

Building on the success of the AlexNet, the VGG networks attempted to scale networks to even deeper architectures. In [SZ14] three architectures with 13, 16 and 19 weighted layers have been introduced. Here, a weighted layer denotes a layer that includes neurons,

<table>
<tr><th>AlexNet</th><th>VGG16</th></tr>
<tr><td colspan="2">input 224 × 224</td></tr>
<tr><td>conv11-96</td><td>conv3-64
conv3-64</td></tr>
<tr><td colspan="2">maxpool</td></tr>
<tr><td>conv5-256</td><td>conv3-128
conv3-128</td></tr>
<tr><td colspan="2">maxpool</td></tr>
<tr><td>conv3-384
conv3-384
conv3-256</td><td>conv3-256
conv3-256
conv3-256</td></tr>
<tr><td colspan="2">maxpool</td></tr>
<tr><td></td><td>conv3-512
conv3-512
conv3-512</td></tr>
<tr><td></td><td>maxpool</td></tr>
<tr><td></td><td>conv3-512
conv3-512
conv3-512</td></tr>
<tr><td></td><td>maxpool</td></tr>
<tr><td colspan="2">fc-4096*
fc-4096*
fc-#classes</td></tr>
<tr><td colspan="2">softmax</td></tr>
</table>

Table 2.1: Overview of different CNN architectures. (left) AlexNet as introduced in [KSH12]. (right) VGG16 as introduced in [SZ14]. Here, the following notation is used: *conv* represents a convolution layer with the next two numbers being the filter size (denoted as side length) and the number of neurons in the layer. *fc* represents a fully connected layer which is followed by the number of neurons in the layer. *maxpool* represents a 2×2 maximum pooling. *softmax* represents the computation of pseudo probabilities at the last layer of the network. Dropout is applied to the layers marked with an asterisk.

e.g., a convolution or a fully connected layer. An unweighted layer is a heuristic operation such as maximum pooling. The 16 layer architecture is shown in table 2.1. All VGG architectures use the same approach, making the filter size smaller and stacking more convolutions on top of each other. The rationale behind this is that two stacked 3×3 filters have observed an area of 5×5 pixels in the original image, which is termed the receptive field size. Furthermore, stacking small convolution filters means that less parameters have to be learned by the network which eases the learning process and in turn allows for incorporating more layers. For example, for a single channel input image, two 3×3 filters have a receptive field size of 5×5, but require only 18 weights instead of 25 of a 5×5 filter. Although a 5×5 filter could learn more different filter types, it has been shown that the stacked 3×3 filters learn a meaningful feature representation [SZ14]. The filters are more constrained but as long as only a few hundreds or thousands of filters are required for the intermediate feature representations this is not an issue for the networks' expressiveness.

Extensions and recent trends

One of the remaining limitations of most networks is the fixed input size of the architectures. For example, both the AlexNet and the VGG16 used an input size of 224 × 224px. The reason is that the fully connected part of the network requires the size of the feature maps at the previous layer to be always the same. As a result either crops or transformed images are used as the input for the network [KSH12, SZ14]. There are two solutions for this issue. The first is the Spatial Pyramid Pooling layer, which follows the idea of the spatial pyramid ([LSP06], see section 2.2.2.1) and builds a quadtree structure on top of the feature map computed by the last convolution layer. The quadtree pooling is performed for each filters' responses separately. Therefore, a fixed size representation is obtained [HZRS15c]. This allows to use arbitrary sized images as input to the network since the convolutions are slided over the complete image, being independent of the image size. Then the fixed size representation of the spatial pyramid Pooling layer allows to connect a fully connected part. It has been shown that especially for objects with a high or low aspect ratio this is a beneficial approach [HZRS15a].

The second solution is to completely omit the fully connected part of the network [SDBR15]. In [SZ14] it has already been shown that a fully connected layer can technically be transformed into a convolution layer. For example, for an input size of 224 × 224, the last convolution of the VGG16 yields a 7 × 7 feature map with 512 features. Hence, each fully connected neuron can be replaced by a neuron that computes a 7 × 7 convolution over 512 channels. Each neuron in the second fully connected layer is connected to the output of the 4096 neurons from the previous layer. It can therefore be replaced by a 1 × 1 convolution over 4096 channels and so on [SZ14]. When evaluating the network on input sizes that are larger than the original input size, multiple outputs will be computed for each class, which can, for example, be averaged [SZ14]. This idea was picked-up in [SDBR15] where a fully convolutional network has been designed. Furthermore, the Residual Networks are also a fully convolutional architecture [HZRS15a]. In addition, these networks introduce skip connections between the layers. These connections help the networks' learning process so that some layers may just learn the identity matrix for some of the feature responses, i.e., responses that indicate classes that are easily recognizable, while still learning weights for other feature responses, i.e., details for very complex classes. As a result these networks can be constructed even deeper with several hundreds or thousands of layers [HZRS15a]. Residual networks outperform the shallower CNN architectures without skip connections and, therefore, show state-of-the-art performance.

2.2.2.3 *Discussion*

Very deep network architectures like most of the recent CNN architectures are more powerful than shallow architectures that build on handcrafted features. Since the last few years CNNs show state-of-the-art results in many recognition tasks. Especially, the results achieved in object recognition tasks revolutionized the field of Computer Vision (cf. [KSH12, SZ14, RDS+15]). When learning a CNN, a classifier is learned in conjunction with multiple feature representations in an end-to-end system. The learned feature representations are, therefore, based on the data and not based on experts' design decisions. Hence, they are typically better suited for the given task and provide a more meaningful representation.

However, for many practical applications, CNNs still have two shortcomings: First, due to the large number of parameters, they do often require a tremendous amount of labeled samples or at the very least a large number of synthetic, but still realistic, training samples. Second, they are computationally more expensive and require GPUs in order to be trained and evaluated efficiently. The first point is tackled by dropout [SHK+14], data augmentation and large datasets. Furthermore, many pre-trained models are available which can be used for fine-tuning a CNN instead of training it based on a random initialization [KSH12, SZ14]. In Computer Vision most models are pre-trained on the ImageNet dataset which contains several million labeled images [RDS+15]. In libraries such as *caffe*, many pre-trained models are available for various task [JSD+14]. However, even with data augmentation, dropout and the availability of pre-trained models, several thousand samples would be desirable for training the network.

The second aspect that limits their application for many tasks is that these can be found in embedded domains, such as automotive, robotics or smart environments. Here, low cost and resource efficient embedded solutions are a main requirement. While embedded GPU chips like the Jetson tx1 and tx2 enter the market, they are still very expensive [Nvi15]. Traditional digital signal processors (DSPs) [Smi13] are typically much less expensive. Nvidia's most recent change in their licenses, prohibiting the usage of low cost GPUs in data centers, may also impact other domains [Mos18]. Hence, approaches that build on handcrafted features still play an important role in several applications.

In the work presented in this thesis both domains will be of interest. While some early work solely focus on handcrafted features, later work will also take into account the advantages that can be obtained when incorporating DNNs.

2.3 OBJECT DETECTION

The task of object detection is more complex than object classification tasks. While classification tasks are merely concerned with the presence of an object, detection tasks are also interested in the location of an object within an image. Typically, a single detector is trained for each of the classes. This makes allows for removing or adding object detectors of new classes without re-training the complete set of object classes. The location is often determined based on bounding boxes which indicate the object's location in an image.

Similar to the classification tasks, there are different approaches to this task. On the one hand there are approaches that follow the traditional pattern recognition pipeline and that are based on handcrafted features. For object detection, these handcrafted features are usually HoG descriptors which are combined into object templates. These HoG detectors and their more flexible extension, the deformable part-based models (DPMs) are explained in section 2.3.1 and section 2.3.2 respectively. On the other hand there are also methods that incorporate DNNs for detecting objects. These methods are discussed in section 2.3.3. Finally, a brief overview over the field of scene parsing, where a pixel-wise analysis of images is provided, is given in section 2.3.4.

2.3.1 *HoG Detectors*

One of the most popular person and object detectors is the HoG detector [DT05]. It builds on the principle of the HoG descriptor as discussed in section 2.2.1.3. For a set of image crops showing either an object class or background, HoG descriptors are computed. The background samples are obtained by hard negative mining in the training set, which means that the selected background samples do not overlap with any of the object classes (cf. [DT05, FGMR10]). As the background is usually less structured and more arbitrary than the object itself, the number of background samples is typically much larger than the number of foreground samples.

Following the principle of the HoG descriptor, a fixed number of cells is computed in which the gradient orientations are binned. Furthermore, sets of cells are summarized into so-called blocks. The blocks may be overlapping and each block is $L2$ normalized. The orientation histograms of all blocks are concatenated into the feature descriptor for the object or background sample. The setup for cells and blocks may vary for the detectors of different classes. Given the HoG representation for the foreground and background samples, an SVM is trained for each object class. The SVM classifies a sample as part of the given class or as background. For the actual object detection, a sliding window is moved over the image on multiple scales. This multi-scale analysis allows for detecting objects at different sizes as well as adding a confidence value to detections. A sample that is only detected in one scale and in none of the neighboring scales is most likely an incorrect detection. As the last step a non-maximum suppression is used in order to find the best candidate from a set of overlapping detections and to obtain the final bounding box [DT05].

The HoG detector has been shown to be very successful in person detection, especially in surveillance scenarios where people are mostly standing and have a fixed aspect ratio

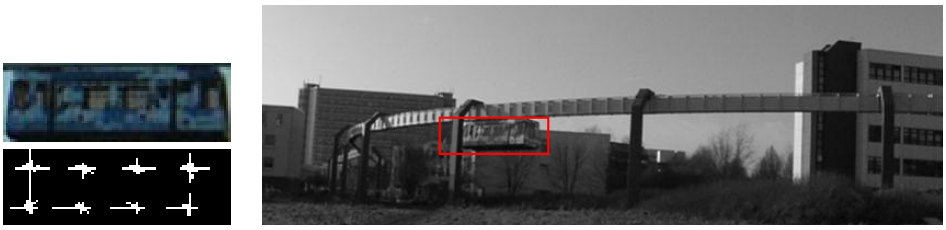


Figure 2.9: Example of a HoG detector. (left) a foreground sample and its HoG descriptor. (right) An exemplary detection of the object.

and appearance. Similar results can be obtained for well structured objects, like cars or trains. An example of a foreground sample, its HoG representation, and a detection in an image is shown in figure 2.9. For general object detection it is an issue that parameters like the number of cells and blocks in x- and y-direction as well as the number of bins need to be determined. Furthermore, the basic HoG detector does not account for deformations, which has been tackled by the deformable part-based models.

2.3.2 *Deformable Part-based Models*

Deformable part-based models (DPMs) tackle some of the issues that the basic HoG detector has. They account for different parts within a given object instance and partially also for the parameterization issues of the HoG detector [FGMR10].

The idea is that an object is subdivided into a set of parts. It still contains the original HoG template, termed the root filter, but this root filter is then subdivided into a set of parts. These parts are initialized by heuristic rules, i.e., the number of parts and they are typically symmetrically positioned along one of the objects axis. The parts are then optimized so that they fit regions of maximal energy in terms of the HoGs' magnitudes. This is done for all positive samples so that a distribution of the parts and, therefore, a deformation cost can be derived for each part. As for the HoG detector, an SVM is trained for each of the parts using a set of foreground samples and hard negatives.

For detecting an object a similar approach as for the HoG detector is used [FGMR10]. The root filter and all part filters are moved over the image at multiple scales. The part filters are typically evaluated at a finer scale than the root filter. Hence, for the root filter and for each part a response map is computed indicating the probability that the respective object or one of its parts is present at a given location. The response maps of the parts are transformed based on the deformation costs so that the transformed response map indicates the position of the root filter. Finally, all response maps are combined into one response map with detection scores for the respective object. These detection steps are illustrated in figure 2.10. Hence, a bounding box for the object is predicted based on its root filter and its parts. In the last step, a non-maximum suppression is applied.

The DPMs are more robust toward deformations of objects than simple HoG detectors and, therefore, showed results that significantly outperformed the HoG detectors [EEVG+15]. DPMs were also combined with additional context models based on co-

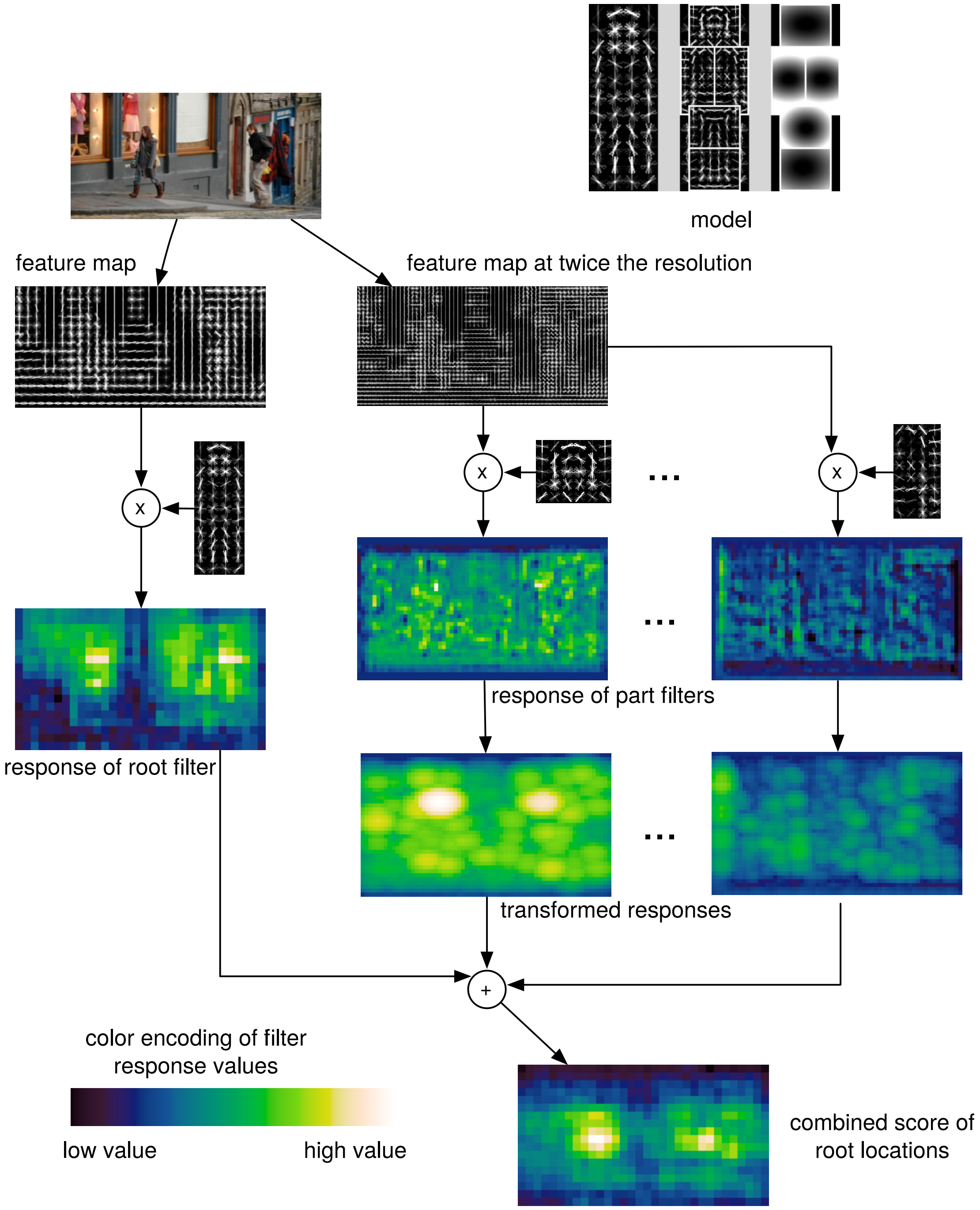


Figure 2.10: Overview of the object detection steps using DPMs. A root filter and multiple part filters are evaluated independently which then results in a combined response map indicating the location of an object. Illustration taken from [FGMR10] © 2010, IEEE

occurences and spatial priors [FGMR10, CTW12] which further improved their performance.

2.3.3 *Regional CNNs*

Besides the approaches that are based on handcrafted features, recently approaches that incorporate deep networks also emerged in the field of object detection. Regional Convolutional Neural Networks (R-CNNs) were the first approach to successfully incorporate deep networks in an approach for object detection. They extend traditional CNNs (see section 2.2.2.2) by defining image crops which potentially contain an object [GDDM16].

Since an exhaustive search would be computationally too expensive, a set of regions of interest is considered for evaluation. These regions are then passed through the CNN. The bounding box of a region of interest is either assigned one of the object class labels or the region is classified as background. For computing a set of region proposals, bottom-up methods like selective search [GDDM14] or supervised approaches like Region Proposal Networks (RPNs) can be used [RHGS15]. Based on the networks' architecture, the input regions are either directly processed by the network (i.e. if a spatial pooling layer is part of the network architecture) or resized in order to fit the input size. In [GDDM14], the classification output is generated by a softmax layer or by training SVM classifiers for each of the classes. In the latter case the second to last fully connected layer is used as a feature vector. This feature vector is then used for the SVM training. However, the advantage of the former approach is that it allows for an end-to-end training which is beneficial as it optimizes both the feature representation as well as the classifier in an integrated manner. During training time the ground truth annotations can be used in order to extract crops of the positive samples. Furthermore, background samples are extracted from the training set in areas without any object. Typically, the CNN is pre-trained on ImageNet and then fine-tuned to the object classes that need to be detected [GDDM16].

Since the R-CNN proposes a straightforward approach that evaluates a set of regions of interest with respect to a given set of object classes, one of the most crucial points is the selection of these candidates. In the following, two approaches that are proposed in the literature, selective search [GDDM14] and RPNs [RHGS15] are discussed. An overview of both approaches is given in figure 2.11.

Selective Search

Selective Search is a bottom-up algorithm that generates region proposals. Bottom-up region proposals have no information about the objects of interest and work purely based on heuristics in an unsupervised manner. These heuristics are mostly based on color or contrast cues [Naß16].

Selective Search builds on the hierarchical graph-based segmentation approach introduced in [FH04]. The graph-based segmentation is used in order to compute an over-segmentation of the image. The selective search algorithm groups these regions in a hierarchical manner based on their properties. The key idea is to use complementary similarity measures. These include size, color, texture and how well the two regions can be combined on a bounding

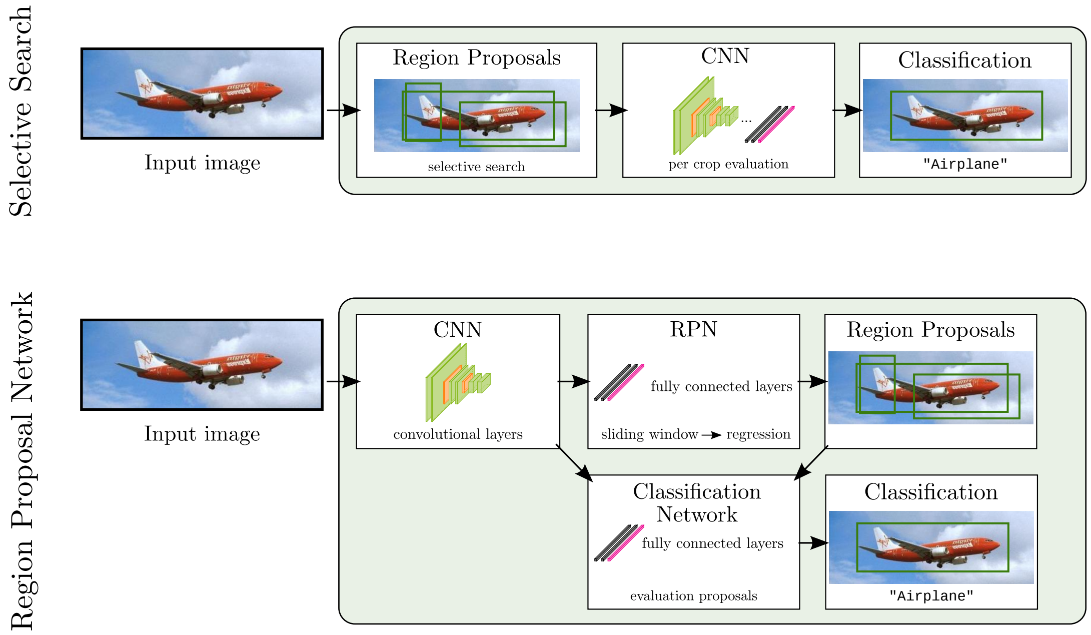


Figure 2.11: Comparison of different region proposal approaches for Regional Convolutional Neural Networks. (Top) A heuristic, i.e. selective search, is used for proposing region candidates which are then evaluated by a CNN. (Bottom) The integrated approach using a RPN. The convolutional layers are evaluated on the complete image and a small regression network is evaluated on the feature maps in order to compute region proposals. The same feature maps are then cropped, resized and used for classifying the proposals. Airplane image taken from the Caltech101 database [FFFP04].

box level. The approach can, further, be combined with various color spaces that are evaluated independently of each other. Finally, all similarity measures are combined and the two most similar regions are iteratively merged, which yields a hierarchical grouping. Each segment of the original segmentation and all segments from the hierarchical grouping are candidates for a region of interest. Since many region proposals can be computed and the quality of bottom-up proposals cannot be expected to be highly accurate many region proposals are evaluated by the R-CNN. In [GDDM14] it has been empirically determined that 2000 region proposals per image show a good trade-off between computation time and accuracy.

Region Proposal Networks

In [RHGS15] a faster version of the R-CNN is proposed. The selective search is replaced by a supervised approach for generating region proposals, the so-called Region Proposal Network (RPN). Therefore, the CNN is split after the last convolution layer. On the one side there is a shallow regression network, the RPN, on the other side there is the fully connected part of the original CNN for object recognition which predicts the class labels. Hence, the first convolution layers can be pre-computed for the complete image instead of processing each region proposal independently.

Based on the feature map of the last convolution layer, the RPN predicts the position and shape of a bounding box and and objectness-score. This regression is computed at each point of the feature map and each proposal is then classified as being an object or as background. The result is similar to the unsupervised selective search as a set of region proposals is generated. However, the regression is trained based on the presence of an arbitrary object in the training set. This is an advantage as the region proposals are already aware of the set of classes. Hence, a region proposal contains an arbitrary object that belongs to the set of objects of interest instead of just an arbitrary area of interest which is computed by a bottom-up approach. In [RHGS15] the number of region proposal could, therefore, be reduced from the 2000, which were used when applying selective search, to a few hundred proposals.

After the region proposals have been computed, the regions are extracted from the pre-computed feature map of the last convolution layer. Then, the forward pass through the network is evaluated for the fully connected part of the classification network. The classification network predicts the presence of any of the classes or assigns the background label to the bounding box. Since the classification network contains more neurons than the RPN it is more accurate and able to remove false positives from the RPN regression. This combination of region proposals and their evaluation speeds-up the evaluation tremendously [RHGS15].

Extensions and recent trends

Similar to classification tasks, several recent approaches focus on fully convolutional networks. The results of the R-CNN could be further improved by incorporating a residual network. In [HZRS15a] it could be shown that the higher descriptiveness, which is achieved by adding more layers to the network, directly translates to the task of object detection. Besides R-CNNs, state-of-the-art results were achieved by the single shot multibox detector [LAE+16]. A fully convolutional network is evaluated and the feature maps at different scales are used as input for a bounding box prediction. For each feature map one bounding box is predicted at each position within the feature map. The prediction contains a location and a class confidence. Using non-maximum suppression, the best scoring boxes are then used for prediction.

Furthermore, it could be shown that fully convolutional networks can be applied for weakly supervised object detection. In [ZKL+16] a fully convolutional network is trained with image level annotations. The output of the last feature map is then used in order to derive class activation maps which indicate the presence of classes in a given image. By applying thresholding on these activations, a bounding box can be predicted without the requirement of bounding box annotations for training.

2.3.4 *Scene Parsing*

A more detailed detection result is provided by so-called scene parsing approaches [Sze10]. The idea is that these provide a pixel-wise annotation for each image and, therefore, segment objects within the scene. There are two approaches for this task, either labels are transferred from a set of annotated images [LYT09, TL10] or alternatively a region-wise or even pixel-wise classification is performed. For the first approach the set of classes doesn't have to be known beforehand making the scene parsing methods open world tasks [SdSB13]. The latter one is more constrained with respect to a set of classes. These approaches are also described as semantic segmentation methods.

One of the first approaches used a conditional random field that incorporates texture, color, location and edge information about each of the patches in order to assign a label [SWRC06]. In order to describe the texture of a patch, textons are learned. In [LYT09], the SIFT flow algorithm has been introduced in order to transfer labels from similar scenes to a query image. Basically, a SIFT descriptor is computed for each pixel in a scene which is then compared to other scene images in an optical-flow like manner. The similarity between scenes and the computed flow allows for transferring the labels. The approach is, therefore, independent of the set of classes. Since the computational effort that is required in the SIFT flow algorithm is comparably high, the so-called superparsing approach [TL10] incorporated an oversegmentation based on so-called superpixels. Given a query image, a set of similar scene images is retrieved from an annotated dataset based on a combination of global scene descriptors such as the GIST of a scene and BoF representations (see section 2.2.1 and section 2.2.2). For each superpixel in the query image the closest superpixels from the set of retrieved images are computed based on a combination of handcrafted and learned features, such as color, shape, a BoF representation and many more. This allows for assigning a label to each superpixel in the query image. In [TL13] the approach was extended in order to also incorporate object detectors and boost the recognition rate of some objects. Similar to the superpixel approach, in [BFP09] a 3D representation is introduced that relies on columns of similar depth. These columns are called stixels which are then classified into different categories. As the focus of the stixel representation is in the automotive domain these are typically roads, cars or other obstacles.

There are also approaches that incorporate CNNs. While the approach introduced in [FCNL13] still builds on superpixels and then extracts features based on a CNN for a label transfer, there are also CNN approaches which use convolutions followed by deconvolutions in order to compute a finegrained classification of an image [LSD15, BHC15, NHH15].

A bottleneck for all approaches that generate a very fine-grained analysis is the requirement of having datasets containing annotations of the same granularity. Although there are approaches to provide more public images at this level of detail, e.g. [MCL$^+$14, COR$^+$16], it remains a very tedious and costly process. For example, in the cityscapes dataset [COR$^+$16] only a portion of the images is annotated at pixel level while several more images are annotated at a less detailed level and may allow for supporting the training of a scene parsing algorithm.

2.4 CONTEXT IN OBJECT DETECTION AND CLASSIFICATION

The approaches for classification and detection of scenes and objects that are discussed in the former sections are solely based on the idea that these entities can be recognized based on visual information. It is however known that contextual information can help in the task of recognizing objects [DHH+09, CTW12, ZZRFF15]. This section discusses different types of context for visual recognition tasks, ranging from co-occurrences to including additional modalities like external text sources.

2.4.1 *Types of Context*

Various forms of context that can improve visual recognition tasks have already been investigated in [DHH+09]. The context information can either be of visual nature or it can be based on additional modalities, such as semantic knowledge.

Visual context can be obtained in a local or global manner. The most local context is a pixel context, e.g., the pixels surrounding a bounding box. The most global form of context are global image descriptors like the GIST of a scene.

Other context cues are a combination of visual and semantic information. One example of a combination of visual information and additional context cues is object level context like the presence, appearance or location of different objects in a scene. The object presence or location is typically derived from an object detector's output [VF15] and the visual recognition is then combined with additional context information. Prominent examples for such context information are the typical position or co-occurrence of objects, which are learned from the training set. These forms of context are by far the most popular ones [FGMR10, CTW12, PXSH14, LNH14, VF15]. Some popular approaches that model context based on objects are described in section 2.4.2. Another example are visual attributes that describe the content of an image, e.g., whether the content of the image looks natural or man-made or whether a scene is indoors or outdoors. These visual attributes can be recognized by separate classifiers that predict the presence of a single attribute by classifying the complete image [PXSH14]. The information derived from the presence or absence of certain attributes can then be used in order to derive information about certain object classes. Methods that incorporate visual attributes are discussed in section 2.4.3.

More sophisticated semantic context information can be derived from external data sources. For example, natural language processing approaches can be applied for extracting additional semantic knowledge from large text corpora [EFC+11]. Lexical databases like the WordNet Tree can also be leveraged for incorporating additional semantical knowledge [MO95]. Approaches that transfer knowledge from external sources are discussed in section 2.4.4.

There are many more modalities that can be used as additional context information [DHH+09]. Some popular examples of context cues are:

- **Depth (3D context)**: This contains all information regarding the 3D layout of a scene, including surface orientations or occlusions. For example, in [HEH08] it has been shown how object detection can be improved by the scene layout.
- **Geographic context**: Some examples of geographic context are the terrain type or population density. A very detailed form of geographic context are GPS coordinates that are attached to the image. This can, for example, be used for tagging photo collections [CLH08].
- **Temporal context**: The most obvious form of temporal context are time stamps that are associated with an image. In combination with GPS coordinates, it can be assumed that photos taken at the same place and at the same time show similar content [CLH08]. In video data temporal context could simply refer to nearby frames.
- **Weather context**: Some examples of weather context are the current/recent precipitation, wind speed and direction, temperature, fog or haze, etc. This is especially interesting in panoramic/overview images, i.e., surveillance cameras monitoring a larger region as discussed in [NN02].
- **Illumination context**: Illumination context is comprised of the sun direction, sky color, cloud cover, shadow contrast and many more. The influence of such conditions is, for example, discussed in [LNE08]. Typically, there is also some interaction between illumination and weather context.

While some of these are of more importance for image processing techniques than scene and object recognition, the exemplary applications show that most context information can also be beneficial for recognition tasks.

2.4.2 *Object-level Context*

Especially the object level approaches that define context based on the dependencies and co-occurrences of different objects are pursued in several works, e.g., [FGMR10, CTW12, VF15]. Most of these works have in common that a considerable effort went into training a state-of-the-art object detector and the results of various detections are combined in order to obtain a context descriptor. In [FGMR10, CTW12] these were DPMs that build on HOG features (see section 2.3.2). Nowadays, R-CNNs [GDDM16, RHGS15] show state-of-the-art performance in object detection (see section 2.3.3).

The first prominent example using object co-occurrences has been introduced in [FGMR10]. A part-based model detector is evaluated for each object class on a given image. The resulting bounding box scores are then re-ranked using a stacked SVM classifier. For each bounding box, a feature vector consisting of the initial score, the bounding box coordinates, and the maximal detection scores for each object category in the image, is constructed. This vector is then classified by an SVM and the prediction score of the SVM is used as the new score for the bounding box in question. The re-scoring yields a significant improvement compared to the plain part-based model detector [FGMR10].

The work presented in [CTW12] uses positive and negative correlations between objects as well as a spatial prior. The model is built in a hierarchical tree structure which is especially useful in tasks with many objects. The model integrates both a global scene descriptor as well as the detection results from a DPM. In [VF15] the detection score of a bounding box is also re-ranked. The re-ranking is based on the position of the box within the scene, modeling a spatial prior, as well as the position of other bounding boxes. Hence, the spatial co-occurences between different types of objects modeled.

2.4.3 *Attributes*

Besides prior knowledge about objects and their co-occurrences, attributes are a prominent form of context information. Attributes, such as *natural, manmade, indoor* or *outdoor*, are descriptive labels that are directly associated with the image. The idea of several of these attributes is similar to the idea behind the GIST of a scene ([OT01]; see also section 2.2.1.1). Several works show that these descriptive attributes can be recognized by image classification (see section 2.2) and furthermore allow for transferring knowledge without explicitly annotating a specific class or object label [RSS11, LNH14, PXSH14, ZZRFF15].

In [PXSH14] scene images are associated with a set of very diverse attributes. It could be shown that these attributes partition the space of all scenes and support the high level understanding of scene images. Visual attributes can be recognized and support the recognition of scenes as these are often correlated. Similarly, in [ZZRFF15] a so-called *knowledge base system* is built that relates scenes with attributes and affordances. It is shown that the association of scenes with attributes and affordances allows for improving the predictions of scenes and their attributes as well as answering visual queries. Textual queries are parsed with respect to attributes so that the best matching images can be retrieved. In [LNH14] attributes that are associated with a set of images or classes are used for uncovering unknown classes and describing them in terms of their attributes. Such tasks with no training samples for given object classes are also referred to as zero-shot learning. Similarly, given a very small set of training samples, attributes can be used in order to transfer class labels to unknown images [RSS11].

Most approaches learn to recognize visual attributes based on a set of training images where the attributes are directly annotated. Each of the attributes is then predicted independently by a classifier, i.e., using an per attribute SVM [LNH14, PXSH14]. While these attributes form a detailed description of the image, it is often a tedious task to annotate various attributes in the training data. Unlike other applications, where a finite set of attributes can be used to describe an entity, the set of attributes that occurs in natural scenes may not be obvious and may not even be limited.

2.4.4 *External Modalitites*

While co-occurences, visual attributes and spatial priors still rely on information that is directly derived from the images, a complementary idea is adding further modalities for context. As discussed in section 2.4.1 there are several types of additional modalities that can be incorporated. The most prominent of these modalities is text. One example are image captions or additional tags that are provided for the images [LMB+14].

Such multi-modal approaches allow for learning connections between the visual and textual domains. This allows for answering visual questions [WSvdH+16], like *'what is the person in the image holding?'*, visual queries [ZZRFF15], like *'show me images with cats'*, or the captioning of images or videos [RWT+13, DAG+15] All three tasks attempt to bridge the gap between the visual domain and a textual description of images. In order to allow for learning connections between these domains, several datasets provide additional textual information. The Microsoft COCO dataset provides a detailed description of the scene in order to train algorithms that caption images [LMB+14].

Similarly, the Visual Genome provides a dataset that combines descriptions and various attributes that can be used in order to describe the scene or the objects within it [KZG+17]. Such descriptions and attributes are often combined with external sources since annotating images with attributes is tedious. The use of external data sources allows for scaling recognizers to a larger number of classes and attributes. In [RSS11] it has been shown that visual attributes can be derived from external sources such as Wikipedia or Yahoo.

2.5 SCENE AND OBJECT RECOGNITION TASKS

For each of the methods discussed in the previous sections, a set of tasks and benchmark datasets that are commonly used in the research community can be found. In the following some of the most important datasets and the evaluation protocols, which are used for assessing the performance of image classification and detection methods, will be introduced. The related work of the presented contributions will refer to these datasets. Some of these datasets will also be used for the evaluation of the methods presented in this thesis. It is therefore important to understand the characteristics of the different datasets in order to interpret the results. Especially the SUN dataset is of interest for the presented contributions as it provides different sets of annotations for scenes and objects.

2.5.1 *Object Classification*

Object classification tasks are concerned with classifying an object that is roughly centered in a given image or an image crop. The tasks are usually designed around a fixed set of object classes and each object belongs to exactly one class. Over the past years three major tasks were tackled by researchers:

- **Caltech-101:** the Caltech-101 was probably the first image classification task with a large number of classes [FFFP04]. Images of 101 objects classes were downloaded using Google's image search. All objects are centered in the images. There is no clutter or occlusion. However, some of the images contain only sketches depicting the object. A few examples are shown in figure 2.12. The dataset is also unbalanced. Some of the classes are visually very distinctive and are, therefore, *easy* to recognize. The number of samples in these classes is over-proportionally large.

- **Caltech-256:** the Caltech-256 is an extension to the Caltech-101 benchmark with 256 object classes [GHP07]. This task was designed to be a more complicated version of its successor. A main reason was that the Caltech-101 dataset has been solved by many algorithms. Although being much larger in terms of classes and the number of images, the dataset suffers from most of the limitations, namely the missing clutter and occlusions, that could already be observed in the Caltech-101 dataset.

- **ImageNet-ILSVRC:** the ImageNet Large Scale Visual Recognition Challenge builds on a large corpus of images in the ImageNet collection [DDS+09, RDS+15]. Starting in the year 2010 several challenges were released. One of the first tasks was a classification task that is comprised of 1,000 classes. The images still show one prominent object but in a more realistic and often cluttered environment.

All tasks can easily be evaluated based on the classification error. As the number of samples per class are relatively unbalanced, especially in both Caltech versions, the error is typically computed in a class-wise manner and then the average is reported for the whole dataset:

$$\epsilon_{\text{classification}} = \frac{1}{C} \sum_{c} \frac{\text{false}_c}{\text{correct}_c + \text{false}_c} \tag{2.30}$$

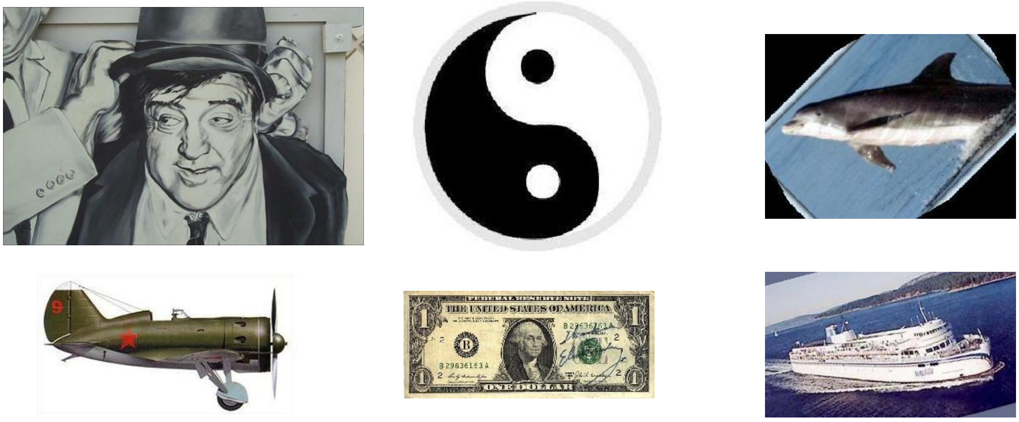


Figure 2.12: Example images from the Caltech 101 dataset. The images are clearly focused on one object category.

where C denotes the number of classes in the dataset. Due to the more cluttered images in the ILSVRC task, some images contain instances of other classes in the background. The ILSVRC classification task is, therefore, evaluated based on the top k error rate (i.e. $k = 5$) [RDS+15]. This extension considers the k highest predictions so that the prediction for an image is considered as correct if the correct label is in these k predicted classes.

As the requirement of having an object centered in an image is quite unrealistic in most scenarios, the object recognition tasks have been replaced by object presence prediction and detection tasks. Methods allowing for assigning one label to an image are rather applied to the various scene classification tasks discussed in the following.

2.5.2 *Object Detection*

The task of object detection is not only concerned with the question which object is shown but also where it is located within an image. The most common metric is bounding box based detection. Here, a predicted detection is required to overlap with a given ground truth annotation based on a pre-defined overlap threshold. There are three major tasks in the field of object detection:

- **VOC:** the Pascal Visual Object Challenge is one of the most influential object detection tasks in the community. There have been different releases of the task between 2007 and 2012 [EEVG+15]. Although the task died together with its creator Mark Everingham[1], it is still widely used. There are two major tasks which are both centered around 20 object categories: *person, bird, cat, cow, dog, horse, sheep, aeroplane, bicycle, boat, bus, car, motorbike, train, bottle, chair, dining table, potted plant, sofa, tv/monitor*. First, the presence detection which is just concerned with predicting the presence of each of the object classes in an image. This task is very similar to the object classification tasks described above. However, it is more difficult

[1] http://www.bmva.org/w/doku.php?id=obituaries:mark_everingham

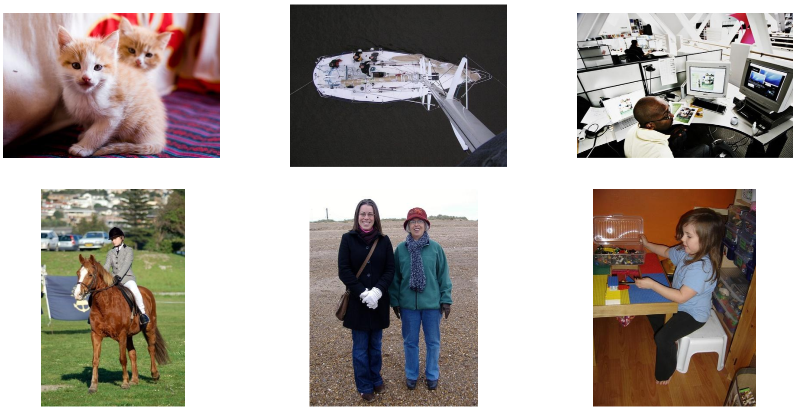


Figure 2.13: Examples images from the VOC object detection benchmark. The images are focused on a few entities/objects in a natural context.

as the images may show multiple objects and more cluttered scenes. The second prominent task is the detection task in which a bounding box for each of the object instances of the 20 classes has to be predicted.

- **ImageNet-ILSVRC:** the classification task gradually merged into a task that also requires an additional location of the object in the image [RDS+15]. A second task is designed similar to the VOC object challenge and concerned with the detection of 200 different object classes in more complex scenes.
- **Microsoft COCO:** the Common Object in Context dataset is concerned with recognizing arbitrary objects in their scene context [LMB+14]. It is comprised of 80 different object classes and several instances in each image. It provides an instance detection as well as a detailed segmentation ground truth. For all images five captions describing the scene with natural language are provided.

The most prominent evaluation metric for detection tasks is the average precision. This assumes that multiple objects can occur in a scene. The performance is then assessed for each class independently. Given a probability for each class to be present in an image, a threshold τ is chosen. Based on this threshold the prediction is accepted or not and precision and recall can be computed. When denoting $\mathrm{GT}_{c,\tau}, \mathrm{EST}_{c,\tau}$ and $\mathrm{COR}_{c,\tau}$ as the number of ground truth, estimated and correct object predictions for class c when using threshold τ, the precision $\mathcal{P}$ and the recall $\mathcal{R}$ can be defined as

$$\mathcal{P}_{c,\tau} = \frac{\mathrm{COR}_{c,\tau}}{\mathrm{EST}_{c,\tau}} \quad \text{and} \quad \mathcal{R}_{c,\tau} = \frac{\mathrm{COR}_{c,\tau}}{\mathrm{GT}_{c,\tau}} \quad . \tag{2.31}$$

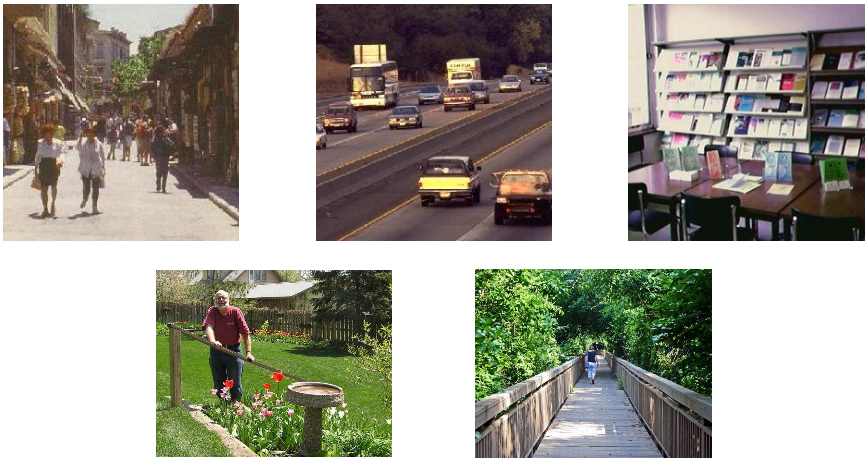


Figure 2.14: Examples for different scene images from the SUN 397 database. From top left to bottom right: *street, highway, library, cottage garden, boardwalk.*

When adjusting the threshold this results in a precision-recall curve, where the precision typically decreases with an increasing recall. The average precision (AP) is then the subsampling of this curve. For example,

$$AP_c = \frac{1}{11} \sum_{\tau \in \{0,0.1,..1\}} max_{\tau':\tau' \geq \tau} \, \mathcal{R}_{c,\tau'} \tag{2.32}$$

subsamples the curve in 0.1 steps. This measure has been used for the VOC challenge until 2011 [EEVG+15]. A more precise approximation considers all possible thresholds τ and has been used in the VOC2011 & 2012 task as well as the Microsoft COCO and ILSVRC dataset [EEVG+15, LMB+14, RDS+15]. Finally, the mean average precision (mAP) is computed by averaging the AP of all classes.

When the task is to detect the location of the object, again the average precision per class is computed. Here, a result is considered as a correct detection if there is a sufficient overlap

$$\mathcal{O}(\mathcal{B}_{pred}, \mathcal{B}_{gt}) = \frac{\mathcal{B}_{pred} \cap \mathcal{B}_{gt}}{\mathcal{B}_{pred} \cup \mathcal{B}_{gt}} \tag{2.33}$$

between a ground truth bounding box $\mathcal{B}_{gt}$ and the predicted bounding box $\mathcal{B}_{pred}$. Most commonly, an overlap $\mathcal{O} > 0.5$ between the bounding boxes is considered as a correct detection [EEVG+15].

2.5.3 *Scene Classification*

Similar to the classification of objects, scene classification tasks are concerned with classifying an image into one class out of a given set of classes. Here, natural scenes are depicted in the images. The scenario is therefore less artificial than the object recognition tasks with roughly centered objects. A scene typically consists of many objects and has a more complex structure. Over the years a few different tasks have been investigated.

- **15 Scenes:** Starting with less than ten scene categories this gray scale dataset was iteratively extended to 15 scene categories [FFP05, LSP06]. The task is to assign a test image to one of these categories.
- **SUN:** The **S**cene **Un**derstanding corpus is a large collection of natural scene images. There are several branches of the dataset which provide different levels of detail with respect to the annotations [XHE+10, XEH+14]. All images are annotated via crowd sourcing using the *LabelMe* toolbox (cf. [BT12]). The *SUN 397 dataset* is similar to the 15 Scenes task but on a much larger scale with 397 different scene categories and at least 100 samples for each of them. The *SUN2012 Scene and Object Dataset in Pascal VOC format* contains a much smaller set of scene images. It represents a subset of the *SUN 397* images and some additional ones. Additional annotations for as many as 4,919 object classes are provided for these images. Finally, the *SUN2009 Context* branch represents an much smaller set of images and annotations, but provides a detailed ground truth annotation for 200 object classes occurring in these images. The object instances within these natural scenes are not only well defined objects, but also highly deformable structures.
- **Places:** The *Places dataset* and its extended version, the *Places2 dataset*, is a large dataset focusing on natural scene images. The goal was to create a scene centric dataset with a size that is similar to the object centric ImageNet [ZLX+14]. The updated Places2 dataset contains ten million images from 434 semantic categories for scene recognition [ZKL+16]. The dataset is complementary to the SUN dataset. PCA-based duplicated removal (cf. [TFF08]) has been performed in order to ensure that the two databases do not contain the same images.

For the classification of scene images again the error rate is used for evaluation. For the more complicated tasks including objects the same metrics as discussed for the object detection tasks are applied.

I want you to attend the conference [...]
Oh, and if you mess up, don't forget to lie about who your advisor is ...
- Prof. Smith, PhD Comic 301, 10.05.2002

3 BAG-OF-FEATURES USING FEATURE AUGMENTATION

The following work has been discussed in two different contexts. The application to image classification has been introduced at ICIP - 2013 [GRF13]. The application to acoustic event detection has been presented at GCPR - 2015 [GPF15]. The approach to acoustic event detection is furthermore based on a previous publication at ICASSP 2014 [PGF14]. The works in the field of acoustic event detection have been summarized in a journal publication in the TASLP 2017 [GPF17]. The integration of this contribution in the context of this thesis is shown in figure 3.1.

In this chapter the idea of feature augmentation is discussed. Additional information is encoded directly at feature level and, therefore, also incorporated in an intermediate representation like the Bag-of-Features (BoF) representation. This is a huge advantage when applying distance measures and complex learning approaches. On the one hand, distance measures like the Euclidean distance are less meaningful in very high dimensional and sparse feature spaces [AHK01]. On the other hand, a low dimensional representation is also computationally more efficient to evaluate.

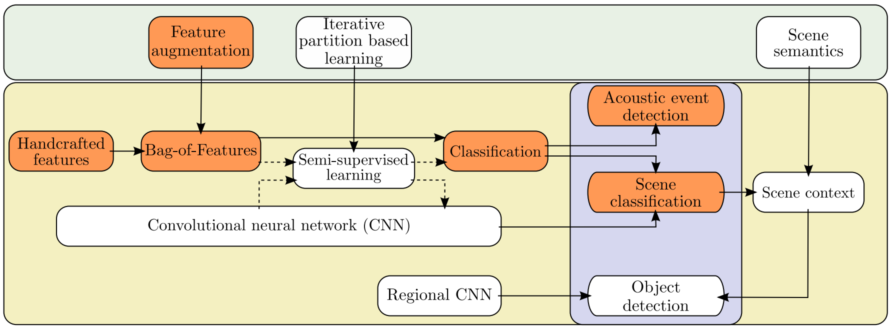


Figure 3.1: Integration of the *Bag-of-Features using Feature Augmentation* in the context of this thesis (relevant topics are highlighted in orange). The chapter deals with the topic of feature augmentation for handcrafted features in classification tasks. Applications are shown for image classification as well as acoustic event detection.

In the plain BoF representation, all information about the local features' position, e.g., their spatial position within an image or the temporal information within a time series, is lost. In the work presented here, additional information of spatial or temporal nature is introduced at feature level. Then, when building a BoF representation on top of these features, an adaptive codebook is computed. The centroids vary for different spatial or temporal areas. As a result, the centroids describe the data more accurately which in turn allows for reducing the dimensionality of the representation without losing descriptiveness.

This chapter is structured as follows. In the following section the related work is discussed. This includes the spatial pyramid approach, which re-introduces spatial information after computing the BoF representation (already briefly discussed in section 2.2.2.1), as well as other approaches for feature augmentation. Then, the proposed approach for feature augmentation is introduced. The approach is evaluated for object and scene classification tasks, where spatial information is encoded at feature level. Furthermore, the approach is also evaluated for acoustic event detection (AED), where the additional information is of temporal nature.

3.1 RELATED WORK

Three works are of special interest regarding the presented feature augmentation approach. All three are concerned with the classification of natural images. The first one is the spatial pyramid approach which has already been briefly introduced in section 2.2.2.1. The approach is discussed in detail in section 3.1.1 It is important related work since it introduced the concept of spatial information in the field of BoF representations for Computer Vision applications [LSP06]. However, the information is introduced at a later stage, after the codebook computation, which often increases the dimensionality tremendously. The other two approaches are concerned with the integration of spatial information in Fisher Vector (FV) frameworks [SPD12, KVJ11]. Both works pursue the same goal but take different approaches. In [KVJ11] the FV encoding is reformulated to also take spatial information into account. In [SPD12] additional information is also introduced at feature level before computing the codebook for the BoF representation. The two approaches are discussed in section 3.1.2. The related work ends with a discussion of recent developments and the strive towards deep learning.

3.1.1 *Spatial Pyramids*

The most traditional approach for re-introducing spatial information in a BoF approach for image classification is the spatial pyramid [LSP06]. As described in section 2.2.2.1 the BoF does not include any spatial information. However, the spatial position of features may carry important information. The spatial pyramid subdivides the image in a quadtree-like manner in order to reintroduce such information. This assumes that the entity of interest is roughly centered in the image and not heavily rotated or occluded. In most natural images and especially photography this is a valid assumption (cf. [SS13]).

The first steps are the same as for creating a BoF representation. Local features are computed for each of the training images. These features are clustered and quantized. In [LSP06] Scale Invariant Feature Transform (SIFT) descriptors are computed on a dense grid and rather small vocabulary sizes of $|V| = 200$ to $|V| = 400$ are used. At this stage there is no spatial information within the feature representation, it is known how many visual words of which type occur in an image, but not where they are located. Then, the spatial pyramid representation is created using a quadtree-like structure by iteratively subdiving the image. On the first level there is only a single quadtree cell which corresponds to the whole image. On the second level this cell is subdivided into four rectangles of equal size, which are then again subdivided on the next levels. Thus, each level l has 2^{l-1} cells. A BoF histogram is computed for each cell in the quadtree, as seen in figure 3.2. The histograms of all cells in the quadtree are concatenated which yields one large feature vector. In [LSP06] typical choices for the parameterization of the pyramid are introduced. Since the dimensionality of the data grows exponentially at each level, most commonly three, but no more than four levels are used. For example, 200 visual words and three levels result in a 4200 dimensional vector. The histograms of the coarser levels are assigned lower weights than the finer levels of the quadtree, because the bins in the finer levels provide

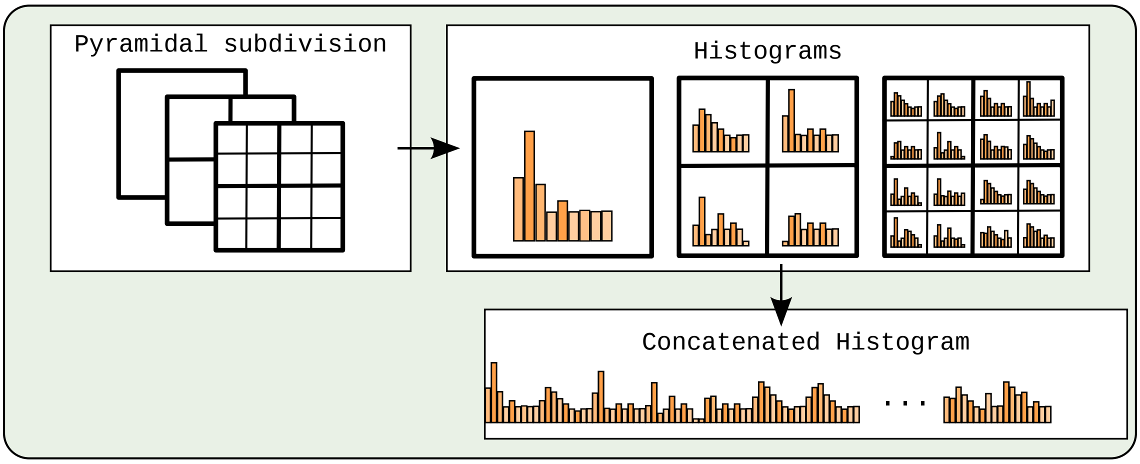


Figure 3.2: Example of constructing a three level spatial pyramid. For each cell a histogram indicating the occurring visual word types is created. The histograms of all cells are weighted and concatenated.

more detailed information about a feature's location. Hence, the BoF histograms at each level are weighted by $w_l = \frac{1}{2^{L-l}}$. At last, the histograms are normalized by the total weight of all features in the image. This creates a histogram of relative visual word frequencies and allows to compare images with varying numbers of visual words more accurately.

The comparison between these high dimensional representations can be performed using a distance measure like the Euclidean distance or using the so-called *pyramid match kernel* [GD05]. Two images are compared by matching the cell histograms $\boldsymbol{h_i}$ and $\boldsymbol{h_j}$ at equal positions in the quadtrees with each other:

$$d(\boldsymbol{h_i}, \boldsymbol{h_j}) = \sum_k min(h_{i,k}, h_{j,k}) \tag{3.1}$$

Thus, the minimum number of features that belong to the same visual word are counted. As the visual words in the finer levels l are assigned a higher weight, the similarity of those levels is more important. The pyramid matching can be implemented as a kernel function for a Support Vector Machine (SVM), which allows for an efficient handling of the very high dimensional data [GD05].

The spatial pyramid has been incorporated in many classification approaches in the following years (cf. [XHE+10, CLVZ11]). As the dimensionality of codebooks and, therefore, also of the final representation became more high dimensional, it has often been used in conjunction with linear SVMs assuming that these high dimensional spaces already allow for a linear separation. For example, in [CLVZ11] representations with as much as 21000 dimensions are evaluated.

3.1.2 *Fisher Vectors Using Spatial Information*

There are two approaches for introducing spatial information into a FV framework. The first one uses spatial modeling based on Gaussian mixture models (GMMs) (cf. [DHS01]) and the second one uses a feature augmentation approach similar to the approach presented in this thesis.

In [KVJ11] the spatial information is integrated into the FV encoding by reformulating the computation of the posterior probabilities for each of the components. Each local descriptor $\boldsymbol{a}$ is associated with its relative location $\boldsymbol{\rho}$ within the image. This yields a tuple $(\boldsymbol{a}, \boldsymbol{\rho})$ with $\boldsymbol{\rho} = (\frac{m}{M}, \frac{n}{N})$ for an image of size $M \times N$ and m, n being the pixel coordinates. This tuple is then integrated into a FV framework by not only fitting a GMM for the appearance feature $\boldsymbol{a}$, but also fitting Gaussians for the features' location $\boldsymbol{\rho}$.

The approach that is solely based on the appearance (see also section 2.2.2.1) first computes the prior

$$p(v) = \pi_v \tag{3.2}$$

and then further computes the posterior probability $p(\boldsymbol{a}|v)$. For an appearance feature $\boldsymbol{a}$, the soft assignment to a component v is computed by

$$p(\boldsymbol{a}|v) = \mathcal{N}(\boldsymbol{a}|\boldsymbol{\mu}_{\boldsymbol{v}}^{\boldsymbol{0}}, \boldsymbol{\Sigma}_{\boldsymbol{v}}^{\boldsymbol{0}}) \quad . \tag{3.3}$$

Here, $\boldsymbol{\mu}_{\boldsymbol{v}}^{\boldsymbol{0}}$ and $\boldsymbol{\Sigma}_{\boldsymbol{v}}^{\boldsymbol{0}}$ are the mean and covariance of the appearance features associated with the component v. Analogous, the posterior probability $p(\boldsymbol{\rho}|v)$ can be evaluated for the spatial location. A Gaussian can be fitted to the locations $\boldsymbol{\rho}$ of the descriptors associated with the visual words v

$$p(\boldsymbol{\rho}|v) = \mathcal{N}(\boldsymbol{\rho}|\boldsymbol{\mu}_{\boldsymbol{v}}^{\boldsymbol{1}}, \boldsymbol{\Sigma}_{\boldsymbol{v}}^{\boldsymbol{1}}) \quad , \tag{3.4}$$

with mean $\boldsymbol{\mu}_{\boldsymbol{v}}^{\boldsymbol{1}}$ and $\boldsymbol{\Sigma}_{\boldsymbol{v}}^{\boldsymbol{1}}$ being the mean and covariance of the locations associated with the visual word v. This is then integrated in a joint model

$$p(\boldsymbol{a}, \boldsymbol{\rho}|v) = \mathcal{N}(\boldsymbol{a}|\boldsymbol{\mu}_{\boldsymbol{v}}^{\boldsymbol{0}}, \boldsymbol{\Sigma}_{\boldsymbol{v}}^{\boldsymbol{0}})\,\mathcal{N}(\boldsymbol{\rho}|\boldsymbol{\mu}_{\boldsymbol{v}}^{\boldsymbol{1}}, \boldsymbol{\Sigma}_{\boldsymbol{v}}^{\boldsymbol{1}}) \tag{3.5}$$

using the probability product. Finally, the model is evaluated for all $|\boldsymbol{V}|$ components in the GMM. Similar to the FV framework, this yields a weighted representation of the differences between a visual word and the centroids.

In the experiments shown in [KVJ11], a principle component analysis (PCA) reduced SIFT descriptor with 64 dimensions has been used for representing the local image patches. Each SIFT descriptor is furthermore associated with its relative location within the image. An SVM with a histogram intersection kernel is used for classification. The experiments have been performed on the 15-Scenes dataset and the Pascal VOC2007 presence prediction task. It could be shown that the presented approaches are able to outperform the spatial pyramid using the same sizes for the vocabulary [KVJ11].

The approach that is most similar to the one presented in this thesis is the one introduced in [SPD12]. The authors argue that computing a centroid that represents a local patch in the BoF representation is basically an averaging of local statistics. The FV framework additionally stores the variance of local features that are represented by a component. It is argued that for tasks like object classification there are two shortcomings of such representations (cf. [BPL10]). The first one is that a finite set of local patch statistics with high variance is averaged. The second one is the variation in the object-dependent information between different images of the same class (i.e. the amount of background that is also included in the image). In order to address these issues the local features on which the FV representations is based are modified by a feature augmentation approach.

It is shown that the variance between the local patch statistics can be decreased if they are limited to specific locations or scales. Therefore, an appearance feature descriptor $\boldsymbol{a}$ that describes a local image patch is extended by its relative position in the image as well as information about the patches scale. From that a feature descriptor

$$\boldsymbol{x} = (\boldsymbol{a},\ \rho_0 - 0.5,\ \rho_1 - 0.5,\ \log(\vartheta_i) - \log(\sqrt{MN})) \tag{3.6}$$

is derived, where $\boldsymbol{\rho}$ is the relative pixel coordinate of the local patch descriptor and ϑ_i the scale of the descriptor. Using the same approach as for the SIFT computation [Low04], the scaled image is derived from convolving the input image with a Gaussian of variance σ_i.

In order to address the second issue, a local feature weighting is derived. A feature vector that is associated with an object should be weighted higher than one that is associated with the background. Therefore, the objectness measure, as introduced in [ADF10] is applied to the image. In this approach a set of bounding boxes is generated in a bottom-up approach. It is then evaluated how likely it is that these boxes contain an actual object. After evaluating several bounding boxes, a relative weight of an object to be present at this location can be computed for each location $\boldsymbol{\rho}$ in the image. In [SPD12, ADF10], 1000 windows per image have been evaluated. Each local descriptor is then weighted with the respective objectness weight. As the FV framework already weights the features according to the posterior probability of a given component (see section 2.2.2.1), the objectness weight is integrated as an additional factor.

In the experiments presented in [SPD12], a PCA-reduced SIFT descriptor with 80 dimensions has been used for representing the local image patches. The three least significant dimensions have been replaced with the three location and scale dimensions shown in equation 3.6. It could be shown that the presented approaches are able to outperform the spatial pyramid approach without increasing the dimensionality of the BoF representation.

Both approaches are similar to the work presented in this thesis. Especially the work presented in [SPD12] pursues a similar idea. The approach has been independently discovered and has been published at the time the method proposed in this thesis has already been submitted. A major difference is the application in a FV framework which allows for the integration of continuous pixel coordinates. However, the computation of a FV representation requires a large number of features that can be associated with each of the components in order to estimate the local variances.

3.1.3 *Recent Developments*

At the time the proposed approach and the related work were developed, BoF approaches showed state-of-the-art results for image classification [CLVZ11]. With the surprising success of Convolutional Neural Networks (CNNs) at the 2012 ImageNet classification challenge [KSH12], Computer Vision research began to focus on the field of deep learning.

Over the past years, CNNs were successfully applied to almost all prominent image recognition tasks, such as object classification [KSH12, SZ14, HZRS15a], object detection [HZRS15a, GDDM16], face recognition [PVZ15] or semantic segmentation [LSD15]. Traditional approaches are usually outperformed on these tasks. A comparison between BoF representations, including feature augmentation, and CNNs for image classification can be found in [CSVZ14]. For different pattern recognition domains, it took longer to adapt the trend of deep learning that evolved in Computer Vision. For example, for the acoustics domain, the shift toward deep learning did not become prominent before 2016 [vdODZ+16, AVT16, VMH+17]. Especially, the lack of large training corpora is a main obstacle. On some tasks with limited training data, traditional approaches are still able to keep up with deep neural networks [GPF17].

While research nowadays focuses on deep learning, traditional pattern recognition pipelines, like the BoF approach, can still be found in many applications. Especially, for applications with limited training data or constrained computational resources these approaches are still of interest. A discussion can be found in section 2.2.2.3.

3.2 QUANTIZED FEATURE AUGMENTATION

The central idea of the proposed feature augmentation method is adding contextual information to the local feature descriptors. This is done by appending additional quantized information $\boldsymbol{s}$ to a local feature descriptor $\boldsymbol{a}$ and therefore constructing a new feature vector

$$\boldsymbol{x} = (a_0, a_1, \ldots, a_I, s_0, \ldots, s_J) \quad . \tag{3.7}$$

In the case of image classification the contextual information could be quantized spatial coordinates resembling the tiling of a spatial pyramid. In case of sequential data this contextual information could be quantized time coordinates. Those two applications are discussed in section 3.2.1 and section 3.2.2 respectively.

Applying the BoF principle, all modified descriptors are clustered in order to form an augmented codebook. After the clustering, the codebook entries do not only represent feature descriptors that are similar, but similar local features in the same contextual regions. As for most BoF approaches, Lloyd's k-Means algorithm is used for clustering [Llo82]. The augmented codebook is then used for quantizing the features of each training sample. The training samples are therefore described by a histogram which already contains contextual information. In the last training step these histograms are used for training a classifier. In the test case, the local feature descriptors of the test samples are again augmented and then quantized with respect to the augmented codebook. The resulting histograms are then classified.

3.2.1 *Image Classification Using Spatial Visual Words*

Applied to image classification, a set of local image descriptors is extracted based on a dense grid. Typically, SIFT descriptors are used which describe the appearance of a small patch based on a gradient histogram (see section 2.2.1.3). The approach can also be applied to any other local feature descriptor. Following the notation of visual words, the cluster centroids that are derived when computing the BoF representation are referred to as *spatial visual words* and the codebook as the *spatial visual vocabulary.* In the following two different augmentations are presented. The first one resembles the quadtree structure of the spatial pyramid and the second one introduces a polar representation. The resulting processing steps, including the subsequent classification, are illustrated in figure 3.3.

3.2.1.1 *XY Representation*

Spatial pyramids subdivide an image in a quadtree-like manner, which implicitly uses the assumption that the visual object is roughly centered in the image. Using the same assumption, the direct translation of this approach to feature level is implemented by adding quantized pixel coordinates as a spatial feature. In [LSP06] it has been shown that the most important information of the spatial pyramid is contained within its top level. This representation can be approximated using spatial visual words with the spatial feature

$$\boldsymbol{s} = (q(\rho_0), q(\rho_1)) \quad , \tag{3.8}$$

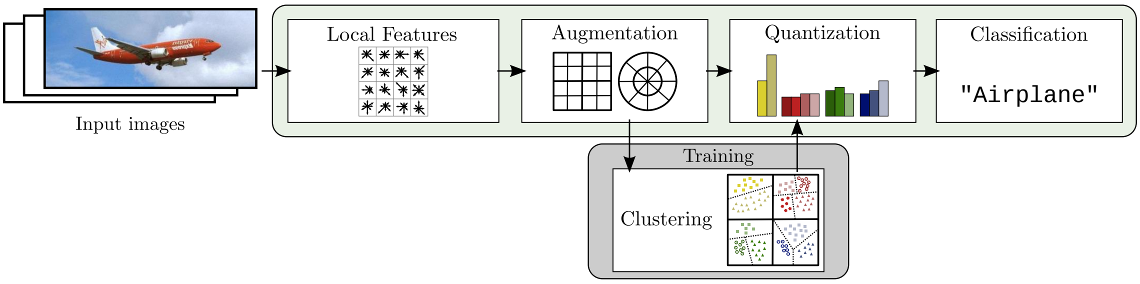


Figure 3.3: Overview of the proposed image classification approach using feature augmentation in a Bag-of-Features framework. Local features, which describe small regions within the image, are computed. These features are augmented with quantized spatial information in either a quadtree or polar representation. The resulting features are then clustered and quantized creating a histogram representation. These histograms are then used for classification. Airplane image taken from the Caltech101 database [FFFP04].

where $q(\cdot)$ denotes the quantization and $\boldsymbol{\rho}$ defines the tuple of relative pixel coordinates. The new feature vector is therefore described by

$$\boldsymbol{x} = (a_0, a_1, \ldots, a_I, \boldsymbol{s}) \quad . \tag{3.9}$$

For each of the regions represented by the spatial quantization, similar appearance features form clusters in the feature space. Note that in order to achieve this behavior the values of the pixel coordinates need to dominate the appearance features. Hence, the local patch descriptor is divided by the average descriptor length

$$\overline{a} = \frac{1}{I} \sum_i \sum_j A_{i,j} \quad , \tag{3.10}$$

with $\boldsymbol{A}$ being the matrix of all I appearance descriptors. Thus, the sum of all dimensions becomes approximately one. As a result, pixel coordinates that are larger than the values of the local appearance descriptors are appended for the spatial feature $\boldsymbol{s}$. For example, for a 2×2 subdivision the four subregions can be represented by the values $[(0,0),(0,1),(1,0),(1,1)]$ for the spatial feature $\boldsymbol{s}$. Thus, by adding these quantized coordinates, different adapted codebook entries will be computed for different spatial regions during the clustering.

When computing a subdivision into B tiles with a codebook size of $|\boldsymbol{V}|$ using the spatial pyramid, the final representation will have the dimension $B * |\boldsymbol{V}|$. This representation is resembled by quantizing the pixel coordinates into B bins using the feature augmentation and computing a codebook of $B * |\boldsymbol{V}|$ spatial visual words. The important difference is that the spatial pyramid uses the same codebook for all subregions, while the proposed codebook has specific entries for each region. Assuming that very high dimensional codebooks contain redundant information, it should then be possible to create a smaller spatial visual vocabulary with the same descriptiveness as the higher dimensional spatial pyramid representation. Thus, reducing the overall size of the representation.

3.2.1.2 *Polar Representation*

Besides approximating the spatial pyramid approach, spatial visual vocabularies can easily be adapted to different spatial quantization techniques. Here, a polar coordinate representation that consists of two parts is proposed: First, the distance r toward the center of the image and second the angle α to an upward axis are encoded. The spatial feature is then described by

$$. \tag{3.11}$$

Again, all values are quantized into a finite set of regions and the local feature descriptor is divided by the average descriptor length. This yields a representation in which the regions close to the center of the image are smaller than the regions in the corners of the image. In most natural images, the regions in the center are also more likely to show an entity of interest whereas the regions in the corner are more likely to contain background clutter (cf. [SS13]).

3.2.1.3 *Image Classification*

For both augmentation approaches, the subsequent steps of the BoF approach can directly be applied. Quantizing all augmented feature descriptors within an image yields one histogram per image. Then, using a set of annotated training images, a classifier is trained. As the BoF representations for images are typically very high dimensional, it is often assumed that the samples are linearly separable (cf. [CLVZ11]). Therefore, a linear SVM can, for example, be used for classification.

3.2.2 *Acoustic Event Detection Using Temporal Acoustic Words*

The concept of encoding additional information can also be adapted from image classification to other domains. One example is acoustic event detection (AED) where the temporal location and type of an acoustic event is detected in an audio signal. The audio signal is sampled and features are computed for very short frames of a fixed number of sampling values. In order to deal with the diverse nature of acoustic events, a specialized BoF approach has been developed. An overview is shown in figure 3.4. The approach is based on a sliding window approach that is moved forward in time. For a given window i, a set of feature vectors $\boldsymbol{X}_i = (\boldsymbol{x_1} \dots \boldsymbol{x_K})$ is calculated for all K frames within this window. All features in this set are augmented with additional information about their position within the window. Here, instead of encoding the spatial location of a local feature within an image, its temporal location t within a time window of ω milliseconds is encoded. To compute the BoF representation, these features are then softly quantized by a GMM that has been trained in a supervised manner. Finally, a maximum likelihood classification is applied.

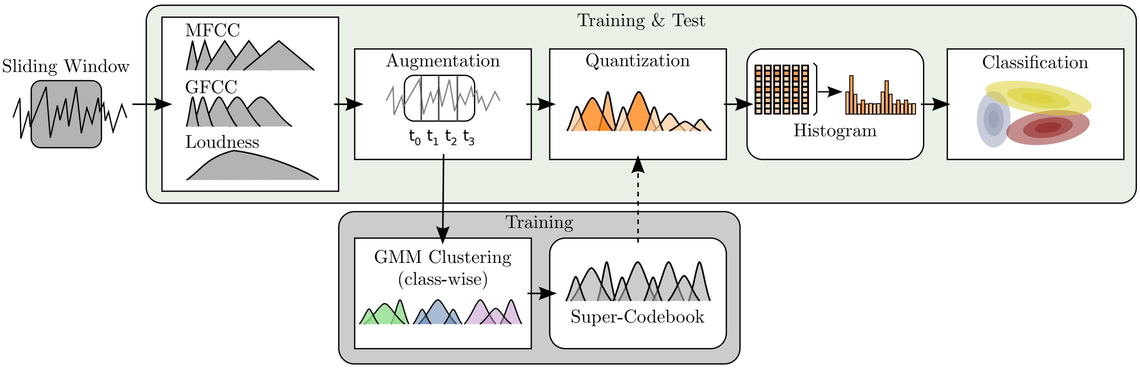


Figure 3.4: Overview of the proposed method for acoustic event detection: Given a time series of an acoustic signal, a sliding window is applied to the signal. For each window MFCCs, GFCCs and a loudness feature are computed. The resulting feature vector is augmented by a quantized time coordinate with respect to the window. A GMM is applied for clustering the features of each class separately in order to learn a supervised codebook. Finally, all features are quantized and the resulting histogram is classified by a maximum likelihood classifier.

3.2.2.1 *Feature Representation and Temporal Augmentation*

A local feature vector $\boldsymbol{x_k}$ describes a single frame, which consists of $1,024$ sampling values. Each local feature vector is comprised of Mel frequency cepstral coefficients (MFCC), gammatone frequency cepstral coefficients (GFCC) and the perceptual loudness. The feature vectors are computed with a frame overlap of 50%. For sound and especially speech processing, MFCC are one of the most widely used features [DM80]. The input signal is filtered by a triangular mel frequency filter bank. From that the GFCCs were derived [SSW07]. The filterbank of the MFCCs is replaced by linear phase gammatone filters. The perceptual loudness is derived from the A-weighted magnitude spectrum. For the resulting feature set, a basic whitening step is performed by subtracting the mean and dividing by the standard deviation of the training data. Thus, creating a set of training features with zero mean and unit variance.

When processing the time windows consisting of K frames, the feature vectors are augmented in the same way as for the image classification. Here, instead of a spatial feature, temporal information is added. Given a fixed window of ω ms in length, it is subdivided into B bins of equal size so that the time t is quantized into a value of $[1, .., B]$. The quantized time coordinate $b = q(t)$ is then appended to the local feature vectors. Thus, an augmented feature vector $\boldsymbol{x_k}$ consists of 13 MFCCs $\mathcal{M}$, 13 GFCCs $\mathcal{G}$, loudness $\mathcal{L}$ and the temporal index b:

$$\boldsymbol{x_k} = (\mathcal{M}_1, \ldots, \mathcal{M}_{13}, \mathcal{G}_1, \ldots, \mathcal{G}_{13}, \mathcal{L}, b)^T \tag{3.12}$$

3.2.2.2 *Bag-of-Super-Features*

Applying the BoF principle, a codebook of *acoustic words* is estimated from the training set. In contrast to image processing, where a 128 dimensional SIFT descriptor is used for the underlying feature representation, the size of the local feature vector is much smaller.

As a result, it has been empirically determined in [PGF14] that the optimal size for a vocabulary is also smaller. When using small vocabulary sizes, hard quantization may discard important details and, therefore, a supervised approach in combination with soft quantization is applied (see section 2.2.2.1).

The supervision is achieved by computing codebooks of size Z for all C classes in Ω separately. These codebooks are then concatenated in order to form a large super-codebook. Based on the super-vector construct used in speaker identification [TCHJH12], this method for constructing a BoF representation, is referred to as a *Bag-of-Super-Features* [PGF14].

For the soft quantization Z means and variances $\boldsymbol{\mu_{z,c}}, \boldsymbol{\Sigma_{z,c}}$ are estimated for each class c by applying the expectation maximization (EM) algorithm to all training windows $\boldsymbol{X}_i \in \Omega_c$ (cf. [DHS01]). All means and deviations are concatenated into a super-codebook $\boldsymbol{V}$ with $|\boldsymbol{V}| = Z \cdot C$ elements with

$$\boldsymbol{V}_{j=(c \cdot Z+z)} = (\boldsymbol{\mu_{z,c}}, \boldsymbol{\Sigma_{z,c}}) \quad , \tag{3.13}$$

where the index j is computed from the class index c and the Gaussian index z as $j = c \cdot Z + z$. Using this super-codebook, a soft quantization of a feature vector $\boldsymbol{x_k}$ can be computed as described in section 2.2.2.1. For a given window $\boldsymbol{X}_i$, a histogram $\boldsymbol{h_i}$ is computed by evaluating all K frames within the window. These histograms are then used as a feature representation of the i^{th} window and as an input for a classifier.

3.2.2.3 *Acoustic Event Classification*

For learning a classifier, the probability $P(\boldsymbol{V}_j|c)$ of an acoustic word $\boldsymbol{V}_j$ to occur in a given class c is estimated. For a given acoustic word $\boldsymbol{V}_j$ and a class c, all training samples $i \in \Omega_c$ are evaluated by

$$P(\boldsymbol{V}_j|c) = \frac{w + \sum_{i \in \Omega_c} \boldsymbol{h_i}(j)}{w\,|\boldsymbol{V}| + \sum_{u=1}^{|\boldsymbol{V}|} \sum_{i' \in \Omega_c} \boldsymbol{h_{i'}}(u)} \ , \tag{3.14}$$

where $\boldsymbol{h_i}(j)$ returns the histogram's value for centroid $\boldsymbol{V}_j$ in the window i. Additionally, w is a weighting factor for smoothing. In practice $w = 1$ (Laplace) or $w = 0.5$ (Lidstone) are commonly used. In the following, Lidstone smoothing with $w = 0.5$ will be applied. Hence, the probability $P(\boldsymbol{V}_j|c)$ is estimated by the fraction of the acoustic word $\boldsymbol{V}_j$ which occurs in windows associated with class c with respect to all acoustic words occurring in any of the windows of class c.

When classifying a given window, all classes are assumed to be equally likely. Thus maximum likelihood classification is used. For the BoF histogram $\boldsymbol{h_k}$ of a single test window k, the posterior probability is estimated using the relative frequency of all acoustic words

$$P(\boldsymbol{h_k}|c) = \prod_{j}^{|\boldsymbol{V}|} P(\boldsymbol{V}_j|c)^{\boldsymbol{h_k}(j)} \ . \tag{3.15}$$

The class c with the maximum posterior probability is chosen and the respective label is assigned to this window.

3.2.2.4 *Acoustic Event Detection*

Due to its simplicity, the BoF approach allows for a rapid computation. It can, therefore, be easily applied to the task event detection. The classification window is evaluated in a sliding window approach that is moved forward for one frame at a time. The recognition result is used for the frame that is centered in the window so that context information is available for a short time before and after the frame. As the window has a length of ω milliseconds, there is a processing delay of only $\omega/2$ ms. A major advantage of the BoF approach is that it is not only an online approach but also able to run in real time. The evaluation can be performed in 20% real time on a single intel i7-3770 CPU.

3.3 EXPERIMENTS

In the following section the proposed feature augmentation is evaluated for both tasks, image classification and acoustic event detection (AED). In the field of image classification a detailed parameter evaluation of the BoF framework and the feature augmentation is given. Furthermore, a comparison of the proposed feature augmentation with a basic BoF approach, the spatial pyramid and the FV framework is presented. For the acoustic event detection the relevant parameters and design decisions are evaluated and a comparison with the state-of-the-art in online acoustic event detection is given.

3.3.1 *Significance Testing*

In order to assess the significance of the differences in performance, a statistical test, namely the permutation test (also known as randomization test; [Goo00, OG10]) is evaluated. The permutation test is chosen as it makes no assumptions about the distribution of the data. It determines whether two sets of samples are generated by the same distribution.

For recognition tasks, these two sets could be the performance measure of two algorithms A and B. For example, two sets of classification results consisting of *true* or *false* values. Each of these sets produces a mean value, i.e., the classification accuracy. If the origin of these Boolean values is mixed up, i.e., whether a correct or false classification is generated by algorithm A or B, the respective mean values should change. Now the difference of the mean values of algorithm A and B are compared again. The chance of producing a larger difference of means after shuffling is very small if the two sets were generated by different distributions [Goo00]. An example is shown in figure 3.5 on the right. The mean values of the two different algorithms become more similar after a permutation. If the two algorithms behave the same, the means remain the same before and after the permutations, as shown in figure 3.5 on the left.

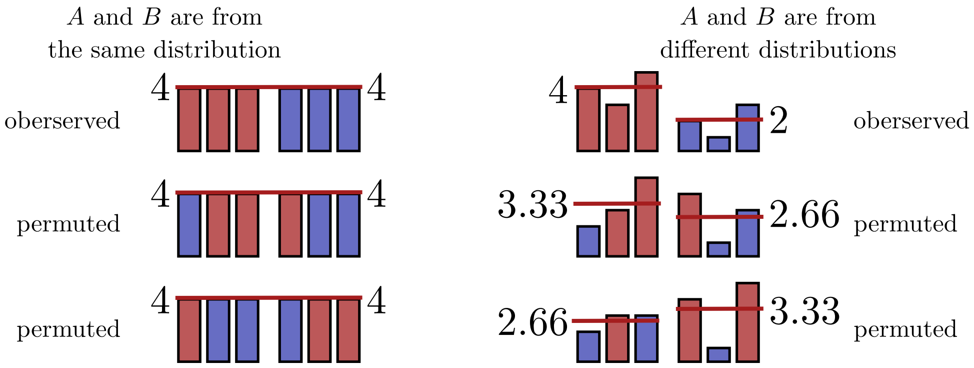

A and B are from the same distribution
A and B are from different distributions
oberserved
permuted
permuted
4
4
4
4
4
4
4
2
3.33
2.66
2.66
3.33
oberserved
permuted
permuted

Figure 3.5: Example of two sets of numbers generated by algorithm A (red) and B (blue). If A and B are from the same distribution (left), the mean value remains the same after permuting the labels. If they are different (right), their mean values become more similar after permuting the labels.

Given two sequences $\boldsymbol{x_i}$ and $\boldsymbol{x_j}$, i.e., classification results produced by the algorithms A and B, the observed difference of the two mean values can be computed by

$$d_{\text{obs}} = |\frac{1}{K}\sum_{k=0}^{K-1} x_{i,k} - \frac{1}{K}\sum_{k=0}^{K-1} x_{j,k}| \quad . \tag{3.16}$$

Analogously, the difference after a permutation can be computed, which is denoted by d_{p} in the following. The significance is measured by determining the p-value as

$$p = \frac{|\{d_{\text{p}} > d_{\text{obs}} | d_{\text{p}} \in \boldsymbol{\mathcal{Q}}\}|}{|\boldsymbol{\mathcal{Q}}|} \quad , \tag{3.17}$$

where $\boldsymbol{\mathcal{Q}}$ is the set of all possible distances after a permutation of the labels. For two sequences of the same length, there are $\binom{2|\boldsymbol{\mathcal{Q}}|}{|\boldsymbol{\mathcal{Q}}|}$ possible permutations. Note that for a large number of test samples, an exact permutation test is not feasible since the evaluation of all permutations $d_{\text{p}} \in \boldsymbol{\mathcal{Q}}$ is impossible in a reasonable amount of time. Therefore, the exact statistic is approximated by a finite amount of permutations with

$$|\boldsymbol{\mathcal{Q}}| = \frac{p(1-p)}{\sigma_p^2} \quad , \tag{3.18}$$

where σ_p is the desired upper bound of the standard deviation of the p-value induced by the approximate statistic [Goo00]. As the p-value is unknown beforehand, the worst case can be approximated using $p = 0.5$. Setting σ_p to 0.001, 250 000 permutations are required in order to assess the significance of the results.

Application in the experiments

In the following, each classification experiment is repeated a given number of times before significance testing. Given, for example, two different parameterizations that shall be compared, each experiment is repeated the given number of times with the same parameters. The average sequence of Boolean values (indicating the correctness of the classification) of these repetitions, which is generated by a given parameterization, is taken as a basis for the significance test. Thus, accounting for deviations by random factors. Then, the permutation test is evaluated for both parameterizations as described above. The difference is considered as significant if $p < 0.05$ and it is considered as highly significant if $p < 0.01$.

3.3.2 *Image Classification*

For the image classification tasks, SIFT features are extracted as the underlying local appearance feature representation. The features are densly sampled on a regular grid. For the baseline experiment, these are quantized in order to compute a BoF representation which is evaluated in a classification task using an SVM. The baseline is then compared to a pyramid scheme [LSP06] and the proposed feature augmentation. First a detailed

	Accuracy [%]			
	Linear SVM		Hist.-Intersection Kernel	
Cell size	freq. hist.	sqrt. hist.	freq. hist.	sqrt. hist.
4 px	35.7 ± 1.2%	40.3 ± 1.0%	30.2 ± 0.8%	39.4 ± 1.0%
8 px	37.2 ± 0.9%	43.4 ± 1.3%	36.2 ± 1.0%	42.4 ± 0.9%
12 px	38.6 ± 1.1%	**44.4 ± 0.9%**	38.9 ± 0.8%	42.4 ± 0.8%
16 px	**38.7 ± 0.8%**	43.7 ± 0.8%	**39.7 ± 1.4%**	**42.5 ± 0.8%**

Table 3.1: Influence of different descriptor sizes on the Caltech-101 dataset. A stride of 8 px and a codebook size of 200 have been used. The experiments were repeated ten times and the mean and standard deviation are reported.

parameter evaluation is given on the Caltech 101 dataset (described in section 2.5.1), the methods are then also evaluated on the 15-Scenes dataset (described in section 2.5.3).

3.3.2.1 *Caltech-101*

The Caltech-101 dataset contains images depicting objects from 101 different categories. For each class there are at least 31 images available. For the evaluation the same setup as in [LSP06] is used. The accuracy is computed for each class individually and the average over all classes is reported. This is important since the number of samples in the Caltech dataset is highly unbalanced and some of the large classes tend to be easily recognizable. 30 samples of each class are randomly chosen for training. In [LSP06] up to 50 samples are chosen for testing in order to reduce the computational effort. In the following, all remaining samples are used for testing. All experiments are repeated ten times and the mean results with standard deviation are reported.

Parameter Evaluation

First, the influence of different sizes for the SIFT descriptors as well as SVM kernels is investigated. A linear SVM is compared to an SVM using a histogram intersection kernel, as introduced in [LSP06]. The histograms are computed as relative frequencies as well as square rooted frequencies, which is commonly used in combination with a linear SVM and resembles the so-called Hellinger kernel [VGVZ09]. All SIFT descriptors are computed using the fast dense grid implementation of the the vlfeat library [VF10]. A stepsize of 8 pixels is used and SIFT descriptors with varying cell sizes of 4 to 16 px are computed. Hence, the total sidelength of the four cells in the SIFT descriptor is between 16 and 64 px. The results in table 3.1 show that larger descriptors improve the results with the best ones obtained using a cell size of 12 or 16 px. This differs from the setup used in [LSP06] where smaller descriptors with a cell size of 4 are used. While the histogram intersection kernel [GD05] works well for frequency histograms, the combination of a linear SVM and square rooted histograms, shows significantly better results ($p \leq 10^{-5}$). In the following, the cell size is fixed to 16 px so that the descriptor has an overall size of 64 px. Furthermore, the stride is set to 8 px which is $1/8^{\text{th}}$ of the descriptor size.

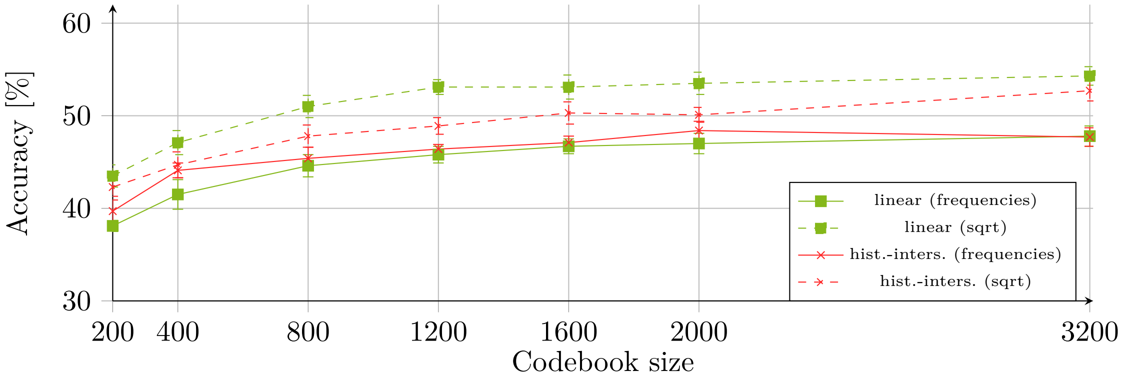


Figure 3.6: Influence of different codebook sizes on the Caltech-101 dataset. A stride of 8 px and a descriptor cell size of 16 px have been used. The experiments were repeated ten times and the mean and standard deviation are reported.

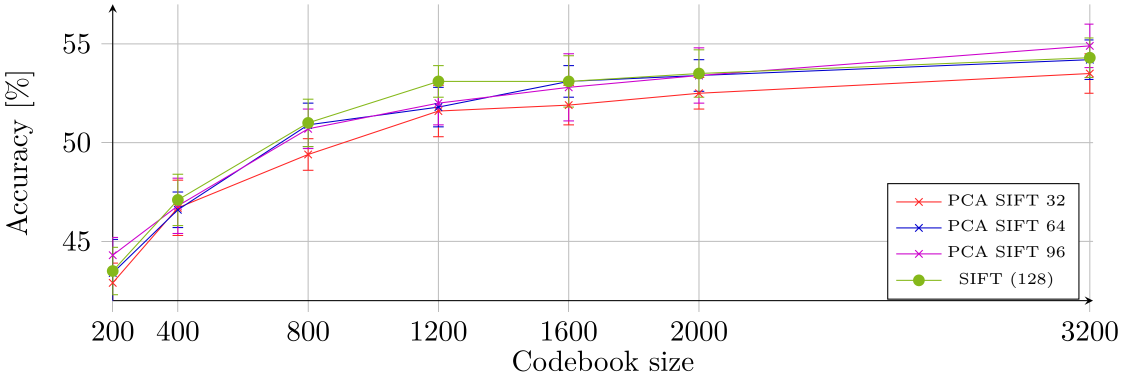


Figure 3.7: Influence of reducing the dimensionality of SIFT descriptors using PCA on the Caltech-101 dataset. All experiments use a linear kernel and square rooting. The experiments were repeated ten times and the mean and standard deviation are reported.

Based on this cell size for the SIFT descriptors, different codebook sizes are evaluated for both the SVM using the histogram intersection kernel and the linear SVM with the results shown in figure 3.6. It can be observed that choosing a larger codebook size improves the results at the cost of a higher computational effort. Note that increasing the codebook size by 200 centroids in the beginning significantly improves the recognition rates ($p \leq 0.011$). However, there is a turning point where no significant improvement can be achieved anymore by increasing the codebook size. For example, neither the differences between a codebook size of 1200 and 1600 nor 1600 and 2000 are significant anymore ($p \geq 0.141$). While further increasing the size of the codebook allows to describe the descriptors in the training data more accurately, these large codebooks do not allow for a better generalization to the unknown test data. Hence, the recognition rate converges or in the case of the histogram intersection kernel even deteriorates at a codebook size of 3200.

Besides the descriptor size and the codebook size, several works noted that reducing the dimensionality of the descriptors using PCA is an important parameter which can improve the results [PSM10, SPD12, SPMV13]. Thus, the influence of a dimensionality reduction

Dimensionality	1	2 × 2	1, 2 × 2	4 × 4	1, 2 × 2, 4 × 4
Method	200	800	1000	3200	4200
Pyramid Lin. Sqrt.	43.4 ± 1.2	54.6 ± 1.0	53.7 ± 1.0	57.6 ± 1.0	57.9 ± 0.5
Pyramid Hist.-Int.	39.7 ± 1.2	52.8 ± 0.8	53.0 ± 0.9	56.7 ± 1.1	57.9 ± 0.9

Table 3.2: Accuracy [%] of different configurations of the spatial pyramid on the Caltech-101 dataset. The top row indicates the tiling configuration for the spatial pyramid which yields the respective dimensionality of the feature representation. The experiments were repeated ten times and the mean and standard deviation are reported.

using PCA is evaluated. The results are shown in figure 3.7. Although it can be seen as an advantage that the dimensionality of the local feature descriptors can be reduced, making it easier to quantize the features with respect to large vocabularies, overall no significant improvement can be observed for the compressed descriptors. However, descriptor sizes of 64 or 96 after a dimensionality reduction using PCA perform almost similar to the complete 128 dimensional SIFT descriptor. This may be a more important advantage for VLAD or Fisher Vector representations where a $|\boldsymbol{x}| \times |\boldsymbol{V}|$ dimensional feature representation is computed with $|\boldsymbol{x}|$ being the dimensionality of the local feature vectors. Thus, for the proposed approach, the complete 128 dimensional SIFT descriptors are used.

Spatial information

With the determined parameter configuration the effect of reintroducing spatial information into the BoF representation is investigated. First, the spatial pyramid approach from [LSP06] is re-evaluated for comparison. In contrast to the original publication, a linear SVM with square rooted histograms is used as this showed superior results compared to the histogram intersection kernel. For comparison, the results using a histogram intersection kernel as in the original publication are also shown in table 3.2. For all pyramid representations a codebook size of 200 is used. Note that the results deviate from the performance reported in [LSP06], where classification rates of up to 64.6% are reported. These differences can, for example, be caused by the different parameterization or the limited number of test samples in the original evaluation. Similar to the observations in [LSP06] it can be seen that the top level of the pyramid stores the most important information for categorization. The recognition rates for the complete pyramid and the pyramid's top level are similar despite the higher dimensionality. Interestingly, the top level even outperforms the complete pyramid in some cases.

The results of the spatial pyramid are then compared to the spatial quantization techniques, which are described in section 3.2.1.1 and 3.2.1.2 respectively. Furthermore, a plain BoF approach is also shown. The comparison is shown in table 3.3. Perhaps most surprisingly, increasing the dimensionality of the plain BoF representation allows for closing the gap between the plain representation and the pyramid scheme. Although the improvement is significant in all cases ($p \leq 0.011$), adding spatial information does not outperform the plain representation by a large margin. The results also show that at the same dimensionality the augmentation approach always outperforms the spatial pyramids. The result is significant ($p \leq 0.018$) for a dimensionality greater or equal to 1000. Besides

	Dimensionality	1	2 × 2	1, 2 × 2	4 × 4	1, 2 × 2, 4 × 4
Method		200	800	1000	3200	4200
xy	BoF (lin, sqrt)	43.4 ± 1.2	51.0 ± 1.3	51.3 ± 0.7	54.3 ± 1.0	55.2 ± 0.9
	spatial pyramid	—	54.6 ± 1.0	53.7 ± 1.0	57.6 ± 1.0	57.9 ± 0.5
	augmented 2 × 2	**48.3 ± 1.0**	**55.1 ± 0.8**	**56.1 ± 1.1**	**59.9 ± 1.0**	59.7 ± 1.1
	augmented 4 × 4	47.0 ± 0.6	54.1 ± 0.9	55.8 ± 1.6	59.1 ± 1.0	**60.1 ± 0.8**
radial	augmented 2	49.2 ± 1.0	**56.1 ± 0.9**	56.4 ± 1.1	59.6 ± 0.8	60.1 ± 0.9
	augmented 3	**50.0 ± 0.9**	55.7 ± 1.1	**56.8 ± 1.7**	**60.7 ± 0.9**	**60.9 ± 1.0**
	augmented 4	42.7 ± 1.2	53.2 ± 1.2	54.3 ± 1.1	58.3 ± 0.9	58.5 ± 1.0

Table 3.3: Accuracy [%] on the Caltech-101 dataset for different approaches which incorporate spatial information. The top row indicates the tiling configuration for the spatial pyramid which yields the respective dimensionality of the feature representation. The experiments were repeated ten times and the mean and standard deviation are reported.

the accuracy improvement compared to the pyramid, the size of the codebook and, therefore, the overall representation can be adjusted easier in the feature augmentation approach than in the spatial pyramid. In the proposed augmentation approach the dimensionality of the final representation is correlated with the size of the codebook whereas the pyramid uses the same codebook multiple times. This is an important observation since it allows to achieve the same accuracy with a more compact feature representation.

The results in table 3.3 also allow for investigating the spatial tiling setups in more detail. It can be seen that the radial setup outperforms the quadtree configuration and allows for further improving the results with the best setup being a radial tiling with three quantization values for both the radius and the angle.

3.3.2.2 *15 Scenes*

The 15 Scenes dataset contains images depicting natural scene images from 15 different categories. At least 200 images are available for each of the categories. The same parameter configuration as for the Caltech-101 dataset is used for evaluating the proposed approach on this dataset. Following the setup described in [LSP06], 100 images per class are randomly chosen for training and the remaining ones for testing. Again, each experiment is repeated ten times and the mean results with standard deviation are reported.

Spatial Information

The results for different setups including spatial information are shown in table 3.4. Interestingly the results indicate that the plain BoF representation which contains the complete image carries very important information. A reason may be that a scene image may not have a clear focus in the center of the image in contrast to the objects shown in the Caltech-101 dataset. The plain BoF even outperforms the spatial pyramid representation when considering the same dimensionality. When increasing the codebook size to more than 200, the feature augmentation with 2 × 2 tiles outperforms the other approaches in all configurations. Nevertheless, the improvement compared to a plain representation is not significant ($p \geq 0.8$).

Dimensionality	1	2 × 2	1, 2 × 2	4 × 4	1, 2 × 2, 4 × 4
Method	200	800	1000	3200	4200
BoF (lin, sqrt)	**69.6 ± 0.8**	72.6 ± 0.6	73.3 ± 0.8	75.2 ± 0.5	75.4 ± 0.8
spatial pyramid	—	72.3 ± 0.8	73.2 ± 0.6	72.3 ± 0.6	74.1 ± 0.8
Augmented xy 2 × 2	69.5 ± 0.8	**72.9 ± 0.6**	**73.5 ± 0.7**	**75.4 ± 0.9**	**75.6 ± 0.5**
Augmented xy 4 × 4	63.4 ± 0.5	69.9 ± 0.9	70.9 ± 0.5	73.1 ± 0.7	74.0 ± 0.6
Augmented radial 2	68.1 ± 0.6	72.0 ± 1.0	72.7 ± 0.8	74.9 ± 0.7	75.1 ± 0.5
Augmented radial 3	66.7 ± 0.7	72.2 ± 0.7	72.4 ± 0.9	74.9 ± 0.5	75.2 ± 0.8

Table 3.4: Accuracy [%] on the 15 Scenes dataset for different approaches which include spatial information. The top row indicates the tiling configuration for the spatial pyramid which yields the respective dimensionality of the feature representation. The experiments were repeated ten times and the mean and standard deviation are reported.

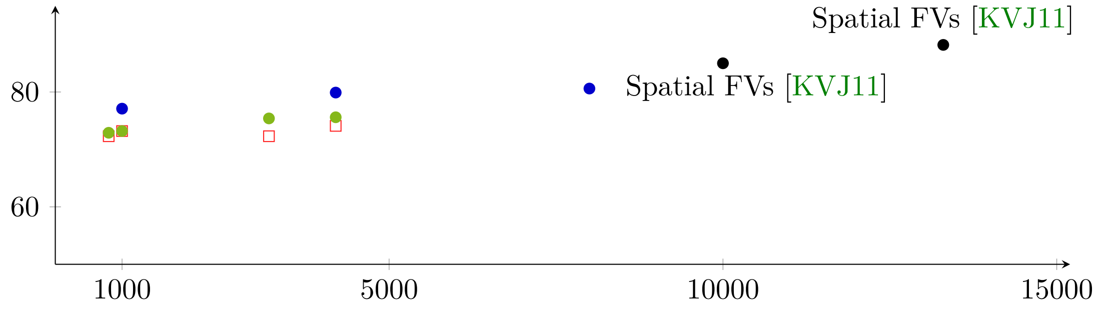


Figure 3.8: Accuracy [%] vs dimensionality on the 15 Scenes dataset for different approaches including results from the literature. (red squares) spatial pyramid, (green) feature augmentation 2 × 2, (blue) feature augmentation 2 × 2 using SIFT features extracted at multiple scales.

Comparison with results from the literature

In addition, a comparison with results as reported in the literature, including the results obtained by the Spatial Fisher Vectors from [KVJ11], is performed. In [KVJ11] a multi-scale analysis extracting SIFT descriptors with different descriptor sizes have been used. Starting at a size of 20 × 20px for the SIFT descriptor, the size has been increased by a factor of 1.2 using eight different scales. Furthermore, the SIFT descriptors are reduced in their dimension to 64 using PCA. Therefore, an additional multiscale-setup is evaluated for the proposed approach. As no improvement has been observed using the PCA reduced descriptors this approach has been resembled by extracting SIFT descriptors at multiple scales without the dimensionality reduction.

The results are shown in figure 3.8. The feature augmentation (green) outperforms the spatial pyramid (red). The additional setup with descriptors extracted at multiple scales (blue) further improves the results significantly ($p < 4 \cdot 10^{-6}$ for $4,200$ dimensions). Nevertheless, the accuracy of the plain BoF approaches falls short of the results obtained by the much higher dimensional FV representations in [KVJ11]. Note however, that the proposed approach yields a good performance using a comparably low dimensional representation.

3.3.3 *Acoustic Event classification*

The acoustic event detection (AED) task is evaluated on the DCASE Office Live dataset. The dataset was used for a recognition challenge in 2013 [GSB+13]. The development set, which consists of a training set and a publicly available test set, is still a very popular benchmark. The task is to detect the acoustic events in these sequences and classify them correctly.

As the proposed method generates a fine grained analysis with one classification result per frame, the evaluation is done in a frame-wise manner (cf. [GSB+13]). Similar to the object detection tasks (see section 2.5.2), the precision $\mathcal{P}_c$ and recall $\mathcal{R}_c$ are computed for each of the foreground classes and compared with the ground truth for each frame. From those the average F-Score over all foreground classes is computed by

$$\mathcal{F}_1 = \frac{1}{C}\sum_{c=1}^{C}\frac{2\,\mathcal{P}_c\mathcal{R}_c}{\mathcal{P}_c+\mathcal{R}_c} \quad . \tag{3.19}$$

3.3.3.1 *DCASE Office Live*

The dataset is comprised of a variety of indoor sounds that could occur in an office scenario. There are 16 sound classes *alert, clearthroat, cough, doorslam, drawer, keyboard, knock, laughter, mouse, pageturn, pendrop, phone, printer, speech, switch, keys* that have to be detected and additional *silence* recordings. The training set consists of segmented sequences for each of the 16 classes with a total length of 18:49 minutes. The test set consists of three scripted sequences with a total length of 5:21 minutes. For each of the test sequences two annotations, provided by different persons, are available. All experiments were repeated ten times using different codebooks each time. The evaluation is performed over all sequences and both annotations. The results are, therefore, averaged over all three scripts, both annotations and ten codebook generations. Note that the differences in the scripts lead to a larger variance as the scripts differ in their difficulty (on average around 3% F-Score).

Rather than using a prior classification step to eliminate silence and background noise, which is done in several systems (cf. [TMZ+07]), the rejection class $c = 0$ is trained with recordings where no event occurred. Since there are no recordings for *silence* in the training data but only in the test sequences, the silent portions from the other two scripts were used in order to train silence class for each of the scripts.

Parameter Evaluation

First, the codebook size is evaluated using varying sizes for $Z = 20, 30, 40, 50$ as well as a varying number of temporal tiles for the feature augmentation. A window size of 600 ms has been used. The results are shown in table 3.5. It can be observed that independently of the number of tiles small codebooks of 30 or 40 acoustic words per class yield good results. The performance deteriorates for larger codebooks. This is in contrast to most image classification applications where much larger codebooks are used and increasing the codebook size almost always improves the performance.

Tiles \ Z	20	30	40	50
2	55.7±4.2	55.3±3.3	54.2±2.9	52.9±3.0
4	55.5±3.8	55.4±3.3	54.7±2.8	54.4±3.3
6	56.0±4.2	**56.3±3.1**	56.1±2.7	55.5±3.0
8	55.9±4.2	55.8±3.4	55.8±2.6	55.2±2.8

Table 3.5: F-scores [%] and standard deviation for different codebook sizes on the DCASE Office Live dataset. The experiments were repeated ten times and the mean and standard deviation over all three scripts and both annotations are reported.

		windowlen [s]				dimensionality	
	tiles	0.3	0.6	0.9	1.2	Z	$\lvert V\rvert$
temporal	2	52.3±2.7	55.3±3.3	53.6±3.9	50.2±3.5	30	510
	4	52.7±3.2	55.4±3.3	54.7±3.5	51.7±4.7	30	510
	6	52.6±3.1	**56.3±3.1**	56.1±2.8	53.5±3.3	30	510
	8	52.1±3.3	55.8±3.4	56.0±2.9	54.2±3.1	30	510
pyramid	2	52.0±4.1	55.8±3.9	**55.9±2.3**	53.7±2.2	15	510
	4	51.1±4.3	54.1±5.0	53.4±3.1	50.6±2.9	8	544
	6	50.9±5.0	52.7±5.3	51.5±4.2	48.4±2.8	5	510
	8	50.1±5.0	52.7±4.2	51.0±3.6	47.4±3.7	4	544

Table 3.6: F-scores [%] and standard deviation for different temporal strategies and different window lengths and tilings on the DCASE2013 Office Live development set. The experiments were repeated ten times and the mean and standard deviation over all three scripts and both annotations are reported. The best F-score for the temporal augmentation as well as the temporal pyramid is highlighted.

In the following, a closer comparison with a temporal pyramid scheme that uses two layers is evaluated. At the bottom layer the pyramid consists of B tiles. The second layer of the pyramid is derived by a maximum pooling of the underlying tiles (as introduced in [PGF14]). The results are shown for a different number of tilings and a varying length of the classification window in table 3.6. When comparing the results with a pyramid of approximately the same dimensionality and using the same window length, the temporal augmentation always outperforms the pyramid configurations. Note that the pyramid approach does not allow for setting the exact same size of the final representation for all configurations. As a result it can be said that the proposed feature augmentation provides a compact yet powerful representation for AED. Furthermore, it can be seen that independent of the tiling parameter a window length of 600 ms shows very stable results. The best result for the temporal processing is achieved with a window size of 600 ms and 6 tiles.

Finally, as the proposed maximum likelihood classifier is not a common choice, especially when considering the origin of the BoF methods in image classification where the SVM is a de-facto standard, different classifiers have been compared: a linear SVM, a k-Nearest Neighbor classifier and a Random Forest. The k-Nearest Neighbor classifier considers 20 neighbors and the Random Forest has been trained using 1000 trees with a maximum

classifier	F-Score
Maximum Likelihood	**56.3±3.1**
SVM Linear	40.8± 7.5
Random Forest	33.1± 5.5
k-Nearest Neighbor	30.8± 9.5

Table 3.7: F-Score and error rates [%] with standard deviations for different classifiers on the DCASE Office Live dataset. The experiments were repeated ten times and the mean and standard deviation over all three scripts and both annotations are reported.

depth of 15. The average F-Score for each classifier is shown in table 3.7. The results show that the maximum likelihood classifier outperforms the other classification approaches. The ML classifier intuitively works well since it derives a probability for an acoustic word to occur for a given class (cf. equation 3.15). This probabilistic approach nicely integrates with the supervised codebook learning which already generated class specific codebook entries. Note that the supervised codebook learning is an important step for improving the recognition results (cf. [PGF14]) which will also be shown by the comparison with figures from the literature.

Comparison with results from the literature

In order to compare the BoF approach with methods from the literature the results of the DCASE 2013 Office Live challenge [GSB+13] challenge are used. Two approaches from the literature have been reimplemented. First, a supervised GMM is evaluated that scores the current features under a model for each event class [ADP07]. Second, the Bag-of-Audio-Words (BoAW) approach from [PA12] using an SVM with histogram intersection. This approach uses unsupervised clustering.

Furthermore, figures as reported in the literature are shown. An extension of the GMM approach using an additional foreground-background separation step which has been introduced in [VBK+13] is shown. This approach showed state-of-the-art results for online acoustic event detection on the DCASE challenge. In addition, the best performing hidden Markov model (HMM) approach for offline event detection and the result of the challenge baseline which uses non-negative matrix factorization (NMF) are shown [GSB+13]. All results are shown in table 3.8. Please note that for the figures from the literature it is neither known which scripts and annotations have been used for the evaluation nor how often the experiments were repeated.

Several observations can be made from the results. The NMF baseline has been outperformed by a large margin by all presented approaches. There is a huge performance gap between the supervised BoF approaches and the unsupervised BoAW. Here, it can be seen that the supervised learning is an important step in the processing pipeline. The basic GMM approach can be interpreted as a method that simply omits the clustering and quantization step of the BoF approaches. The results show that the BoF approaches outperforms the GMM method and even shows a performance similar to more sophisticated GMM method presented in [VBK+13]. It can be seen that the proposed approach shows state-of-the-art

	Method	F-Score [%]
proposed	BoF temporal augmentation	56.3 ± 3.4
re-implemented	BoF plain	55.0 ± 3.1
	GMM [ADP07]	53.6 ± 4.2
	BAW [PA12]	46.9 ± 7.0
	BoF FV encoding	25.9 ± 5.6
literature	GFB HMM [SCS+13] (*)	76.0
	BG-FG GMM [VBK+13]	56.3
	NMF baseline [GSB+13]	20.6

(*) offline approach.

Table 3.8: Comparison of different methods and literature values on the DCASE Office Live dataset with the proposed Bag-of-Features approaches. F-scores in percent. The re-implemented results are also averaged over all three scripts, both annotations and 10 runs each.

results for online acoustic event detection. Note however that offline approaches which are mostly using HMMs are able to outperform the presented online methods. For example, the approach presented in [SCS+13] achieves an F-Score of 76%.

Besides these figures from the literature, a plain BoF approach without any temporal information, as introduced in [PGF14], and an approach using a FV encoding have also been evaluated. The results are also shown in table 3.8. It can be seen that the temporal augmentation is able to improve the performance compared to the plain approach. Furthermore, it is interesting that the FV encoding which performs very well in the image classification tasks does not perform well for AED. A reason may be that the sliding window contains only a very limited number of local features which appear to be not sufficient to estimate a local statistic encoding the variance around the centroids.

3.4 CONCLUSION

In this chapter a novel approach for feature augmentation has been introduced. Additional information is encoded at feature level so that it can easily be integrated in an intermediate representation such as the BoF representation. The approach has been applied to two domains: image classification and acoustic event detection. Quantized spatial or temporal information is added to the local features before computing a BoF representation. Such information is typically lost when creating a BoF representation. Re-introducing this information at feature level causes the clustering step in the BoF pipeline to compute centroids which are specific for different spatial or temporal areas. As a result the data can be described more accurately by the learned BoF representation.

In the application to image classification two approaches for encoding spatial information at feature level have been introduced. The first approach uses quantized coordinates in order to resemble the top level of a spatial pyramid. The second configuration creates a radial setup. Both approaches assume that the object of interest is roughly centered within the image or an image crop that is processed. This assumption is valid for many natural images due to the photographer bias (cf. [SS13]). This property is even more exploited by the radial setup in which the spatial tiles are finer in the center of the image.

In the application to acoustic event detection the additional information is of temporal nature. A sliding window is moved over a temporal sequence of sounds in order to detect certain acoustic events. A quantized temporal coordinate is appended to the local feature vectors adding information about the feature's temporal location within the sliding window and, therefore, capturing the temporal dynamic of acoustic events.

The results on the two tasks showed that feature augmentation is able to improve the performance of BoF methods. Furthermore, it could be shown that augmentation outperforms the well established spatial pyramid approach [LSP06] when considering representations of the same dimensionality. This is a crucial advantage when comparing feature representations based on distance measures since these become less meaningful in high dimensional, sparse feature spaces (cf. [AHK01]).

The independent development of an approach encoding spatial information in a Spatial Fisher Vector framework for image classification in [KVJ11] as well as the feature augmentation approach in [SPD12] confirms these findings. Especially, the approach in [SPD12] follows a similar idea and showed good results on the Pascal VOC dataset [EEVG+15] for predicting object presences. Comparing the results of the proposed augmentation to the results reported in [KVJ11], it has to be noted that the higher descriptiveness of Fisher Vectors is a huge advantage in image classification. It could however be shown that this advantage is not transferable to other applications like AED. The focus of the proposed method has been on obtaining a compact representation that achieves similar performance to the pyramids while reducing the dimensionality. With the extensive evaluation of BoF and CNN approaches for object classification presented in [CLVZ11] and [CSVZ14], feature augmentation became very popular in many methods for computer vision tasks.

Perhaps it's time we came up with a plan for how you're going to finish your thesis. [...]
I'm going to tell you that you need to finish your thesis and you're going to do it.
- Prof. Smith, PhD Comic 1365, 03.09.2010

4 SEMI-SUPERVISED ANNOTATION LEARNING

The foundation of the following work has been published in IJPRAI - 2015 [GF15]. It is based on previous work of Jan Richarz and Szilard Vajda [RVF12, VJF11] with whom a joint paper has been published in Pattern Recognition - 2014 [RVGF14]. The extension using CNNs incorporates an idea from [GSF17], which has been published at ICIP - 2017. The integration of this contribution in the context of this thesis is shown in figure 4.1.

In this chapter a novel approach for semi-supervised annotation learning in large image collections is introduced. The focus is on collections displaying natural scene images. The goal is to learn a visual classifier for natural scene images that is trained with minimal human effort. In the work presented here, the principle of active learning is applied. Thus, a subset of a previously unlabeled dataset is selected and presented to a human annotator, the so-called human in the loop. It is assumed that gathering data is easy, but labeling is expensive. This can be safely assumed for most image data. It is further assumed that adding the information from the unlabeled samples improves the accuracy when training a classifier, which is a fundamental assumption of every semi-supervised learning approach.

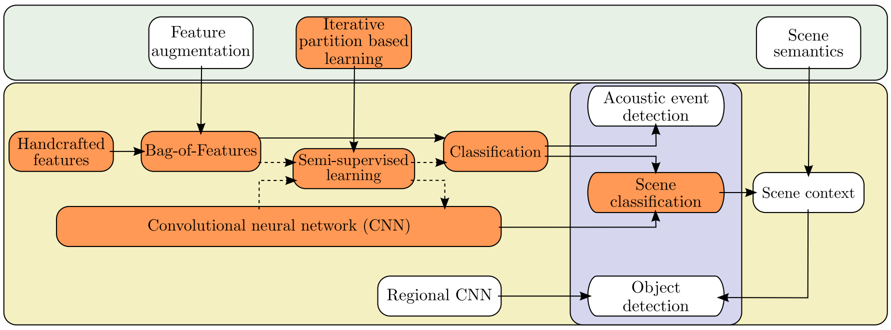


Figure 4.1: Integration of the *Semi-Supervised Annotation Learning* in the context of this thesis (relevant topics are highlighted in orange). The chapter deals with a semi-supervised learning approach for image classification. It incorporates traditional handcrafted features as well as features derived from CNNs.

In the proposed work, a clustering in multiple views, i.e., feature representations, is computed. These clusters are then refined by iteratively adding new centroids and repartitioning the data based on cluster evaluation measures. All centroids are iteratively labeled by the human in the loop. As a result, the unlabeled dataset only gets partially annotated. The samples for labeling are selected in a way that allows for propagating their labels to the remaining unlabeled samples based on the partitioning with high certainty. In the final labeling phase the samples are labeled based on a voting over the different views. Samples where no label can be assigned with a high certainty remain unlabeled.

This chapter is structured as follows. In the next section, the related work is reviewed. The method is based on the ideas of cluster-based annotation learning (CBA) [VJF11] and retrieval-based annotation learning (RBA) [RVF12]. Both approaches compute multiple views on the data in order to improve the robustness. Furthermore, the label propagation approach from [ZBL+04], which is a popular approach for semi-supervised learning, including its extension using active learning [EFS13], is reviewed and recent approaches including semi-supervised learning using Convolutional Neural Networks (CNNs) [HMC17, DVG16] are discussed. Then, the proposed method for semi-supervised learning in natural scene images is introduced. The approach is evaluated on two datasets for scene classification. The experiments include handcrafted features, where a final classifier is trained based on a single feature representation, i.e., a Bag-of-Features (BoF), as well as CNN features, where a deep neural network is fine-tuned based on the dataset that has been labeled in a semi-supervised manner.

4.1 RELATED WORK

Semi-supervised learning is a widely studied field (cf. [Zhu06]). Many approaches provide a theoretical framework for semi-supervised learning and are not specifically concerned with the recognition of images. Semi-supervised learning for image recognition is a difficult task as the representations are typically very high dimensional and the images are visually diverse such that there is a high intra-class variance.

The approach that is proposed in this thesis is concerned with recognizing scene images where a single scene label is assigned to an image. Compared to tasks like weakly supervised learning ([DAF12]; see section 2.1), the task of semi-supervised learning for image classification has not been very popular in recent years (cf. [HMC17]). The proposed approach is based on two approaches, which as a result are also the most prominent related works [VJF11, RVF12]. Both approaches were applied to images containing handwritten characters. The approaches are discussed in section 4.1.1 and section 4.1.2 respectively. A comparison of both approaches is given in [RVGF14]. Furthermore, the framework provided in [ZBL+04] and its extension including active learning [EFS13] for image classification are discussed in section 4.1.3. The related work ends with a brief discussion of recent developments of semi-supervised learning approaches for image classification.

4.1.1 *Cluster Based Annotation Learning*

The cluster-based annotation learning (CBA) has been introduced in [VJF11] where it has been applied to the task of handwritten character image labeling. The approach relies solely on the idea of clustering in multiple feature representations. An overview of the processing steps are given in figure 4.2.

Given a set of unlabeled samples, U views on the data are computed. Each view is given by a different feature representation. In [VJF11], raw pixel values, principle component analysis (PCA) or a feature set derived from an auto-encoder have been used. These features are then clustered using Lloyd's k-Means algorithm. The centroids in each representation are then manually labeled. Each sample in the cluster inherits the label that has been assigned to the centroid. When computing K clusters in each of the U representations, the labeling effort is reduced to $K \cdot U$. Thus a label matrix $\boldsymbol{Y}$ is derived from the clustering such that for the sample i a label vector $[Y_{i,1}, \dots, Y_{i,U}]^T$ with $Y_{i,u} \in \{1, \dots, C\}$ is computed. An entry is assigned the class label c if the sample i belongs to a cluster whose centroid is assigned the label c in the feature representation u. Using unanimity voting, a final labeling $\hat{y}_i$ of all samples derived as

$$\hat{y}_i = \begin{cases} c & \text{if } Y_{i,u} = c \quad \exists c \quad \forall\, u \in [1 \dots U] \\ \emptyset & \text{otherwise.} \end{cases} \tag{4.1}$$

Thus, a sample remains unlabeled if no unanimity is achieved. The learned annotations $\hat{\boldsymbol{y}}$ are then used in order to train a final classifier. This final classifier is therefore trained with a low manual annotation effort and is evaluated on a separate test set.

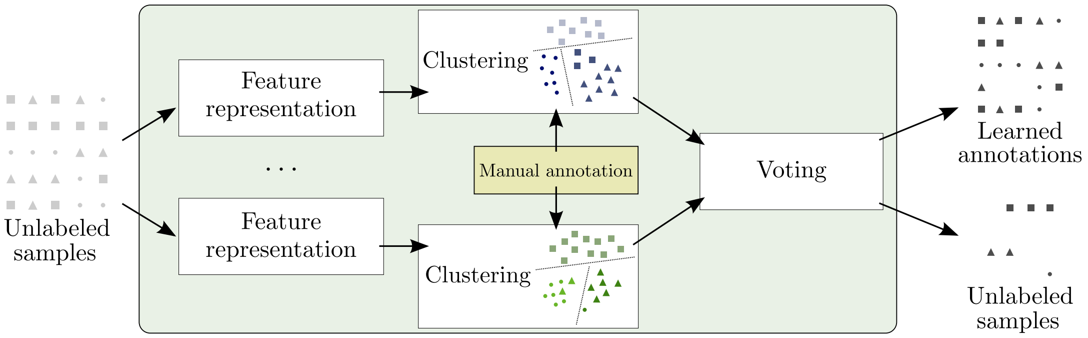


Figure 4.2: Overview over the cluster-based annotation learning (CBA). Given multiple views on the data, i.e. different feature representations, the samples are clustered in each representation independently. Each cluster is assigned a label by a human in the loop and the labels are assigned to every sample in that cluster. After labeling each cluster, the samples are assigned a label based on a unanimity voting over all views resulting in a partially labeled sample set. The labeled samples are then used for training a classifier.

In [VJF11] it has been shown that for the task of handwritten character recognition the labeling effort can be reduced to 0.4% of the data in the MNIST dataset. Using a total of three feature representations and training a simple k-Nearest Neighbor classifier, similar performance to a supervised case is achieved. It has been shown that especially for large classes with a low intra class variance high accuracy can be achieved by CBA [RVGF14]. Since the clustering approach uses Lloyd's k-Means, it heavily relies on the Euclidean distance between the samples. It is, therefore, crucial that the features are able to separate the classes quite well in an Euclidean space.

While CBA is able to reduce the manual labeling effort it has to be noted that a shortcoming of this approach is that the number of manual annotations grows with the number of clusters, but also with the number of feature representations.

4.1.2 *Retrieval Based Annotation Learning*

The shortcoming of the CBA, in which the number of annotations is dependent on the number of feature representations, is addressed in the retrieval-based annotation learning (RBA) [RVF12]. An overview of the proposed approach is illustrated in figure 4.3. A single sample is used as a query in a given set of U feature representations. Each sample is labeled manually. An ordered list of similar samples is determined based on a distance measure, i.e., the cosine distance. Each list is truncated based on a threshold τ_u for the u^{th} feature representation. Samples with a distance smaller than the threshold are assigned the same label as the query, resulting in a label matrix $\boldsymbol{Y}$ as for the CBA. Again, a unanimity voting is performed on these labels (see equation 4.1) and samples which are unanimously voted for are added to the pool of labeled samples after each retrieval operation. In addition two matrices of label weights $\boldsymbol{W}$ and $\boldsymbol{W'}$ are created for each sample. The weight $W_{i,c}$ is increased by the number of truncated retrieval lists the sample i occurs in every time a query is assigned the label c. The weight W'_i is increased by one every time the sample i occurs in at least one of the retrieval lists independently of the queries' label.

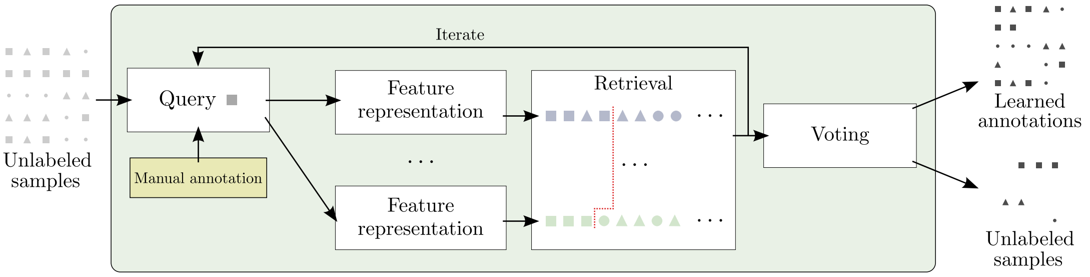


Figure 4.3: Overview over the retrieval-based annotation learning (RBA). For the retrieval-based approach, samples are represented in multiple feature representations. A sample is selected as a query and the most similar samples are retrieved in all representations. The query is labeled by a human in the loop and the label is propagated to samples that occur in all retrieval lists. After a given number of retrieval operations the remaining samples are assigned labels based on the retrieval lists they occurred in so far. Samples where no label can be assigned with high certainty remain unlabeled. Based on the resulting partially labeled sample set, a final classifier is trained.

The next queries are then determined based on a heuristic: a query is randomly selected from the samples that occurred least frequent in the retrieval lists. After a given number of retrieval operations K, the manual labeling is stopped. Usually quite a large number of samples still remain unlabeled after this procedure. Therefore, a soft voting is performed for the remaining samples.

$$\hat{\boldsymbol{y}}_i = \begin{cases} \underset{c}{\operatorname{argmax}}\ \boldsymbol{W}_{i,\cdot} & \text{if} \quad (\frac{K}{(K-1)W_i'} \ \max \boldsymbol{W}_{i,\cdot}) > \tau_{\text{vote}} \\ \emptyset \quad \text{otherwise.} \end{cases} \tag{4.2}$$

The maximal vote count which is non-unanimous is $(K-1)/K$, thus the maximum count any of the remaining unlabeled samples can have is $\frac{(K-1)W_i'}{K}$. The condition above thus represents a normalized confidence score for class c which has to be above a given threshold τ_{vote}. If the condition if fulfilled the sample is assigned the label with the highest score otherwise it remains unlabeled. As for the cluster based approach, the labeled samples are used for training a final classifier.

The approach has also been evaluated for the task of handwritten character recognition. In [RVF12] a fixed number of labels has been used for annotating samples from two sets of images. On the MNIST dataset 0.3% of the complete training set has been labeled. On a more diverse set of characters, about 1.9% of the data has been labeled. It could be shown that on the more diverse dataset the retrieval based approach works better than the cluster based method.

While RBA has the advantage that the number of manual annotations is independent of the number of feature representations, it is difficult to decide on the threshold values which are required. In [RVF12] the cosine distance has been used such that fixed thresholds τ_u could be chosen within a $[0, 1]$ value range. However, the choice is still a parameter that is very sensitive for different feature representations and difficult to optimize.

4.1.3 *Learning with Local and Global Consistency*

A well known approach to semi-supervised learning is the learning with local and global consistency (LLGC), which has been introduced in [ZBL+04]. In this approach labels are propagated across the dataset. The propagation iteratively spreads the label from a given sample to its neighbors until a global convergence is reached and all samples have been labeled. The approach is independent of the initial number of labeled samples.

Each sample is associated with a binary label vector so that a matrix $\boldsymbol{Y}$ is created which contains the labels of all samples in the dataset. An entry $Y_{i,j}$ is one if sample i is labeled as class j. The label vector contains only zeros if a sample is unlabeled. Similarly, a matrix $\hat{\boldsymbol{Y}}$ is constructed for the propagation results with $\hat{\boldsymbol{Y}}^t$ being the intermediate result at time t. The dataset is considered as a fully connected graph with weighted bi-directional edges between all samples. The weights are based on the Euclidean distance between the samples with

$$W_{i,j} = \exp(-||\boldsymbol{x}_i - \boldsymbol{x}_j||^2/2\sigma^2) \tag{4.3}$$

so that the weight is larger for samples which are closer to each other. The diagonal elements $W_{i,i}$ are set to zero. This weight matrix is then used in order to iteratively update the propagation matrix by

$$\hat{\boldsymbol{Y}}^{t+1} = \eta \boldsymbol{W'} \hat{\boldsymbol{Y}}^t + (1-\eta)\boldsymbol{Y} \quad \text{with} \quad \boldsymbol{W'} = \boldsymbol{G}^{-1/2}\boldsymbol{W}\boldsymbol{G}^{-1/2} \quad , \tag{4.4}$$

where η is a learning rate. The labels are updated with weight η such that the original label vector of a sample is kept with weight $(1-\eta)$. This formulation allows to propagate labels over the complete dataset. Unlabeled samples do not have an initial label so that they only propagate the intermediate result $\hat{\boldsymbol{Y}}^t$ after the first iteration. For an unlabeled samples, the term $(1-\eta)\boldsymbol{Y}$ will be zero for all classes and is therefore neglectful. The process can be continued until a global convergence is reached.

As the label propagation does not guarantee a local smoothness, the authors in [ZBL+04] add an additional smoothness constraint. The constraint states that the labels of samples which are close to each other should not change too much. Furthermore, this constraint also considers the initial labels as trustworthy and thus the propagated labels should not deviate too much from their initial labeling.

In [ZBL+04] it could be shown that the proposed approach works for multiple tasks such as digit recognition or text classification. Promising results are achieved even when labeling as few as only one example per class.

Extension to Learning with Local and Global Consistency

The LLGC has been extended in several works, like [GHY08] or [EFS13]. For example, in [GHY08] the class labels are incorporated into the label propagation so that samples with the same label do not propagate the labels between each other. Furthermore, it is proposed to incorporate a class prior in the propagation, assuming that it can be estimated beforehand.

In context of the proposed work, the most interesting extension has been proposed in [EFS13]. The authors extend the proposed approach by a sample selection strategy and active learning in order to scale the approach to large datasets. The underlying assumption is that it is computationally not feasible to include all unlabeled samples in the graph structure which is used for learning in [ZBL+04]. Instead the graph structure is using a subset of samples, e.g., the labeled samples, and two methods for enriching the structure are proposed. The first one is density based, assuming that samples in dense regions are representative for a certain class. In the spirit of lowering the computational burden, the second one is a k-Nearest Neighbor based approximation of the dense approach. The graph structure is enhanced around the initially labeled training samples. If the target domain, i.e., the test set is known in the form of feature vectors with unknown labels, the authors propose to also enhance the structure in the vicinity of the test set. However, in reality this will rarely be the case. Furthermore, the active learning approach from [EFS12] is included, which also uses an uncertainty and density modeling. Uncertainty is measured by an entropy measure over the propagated labels which identifies regions where the labels do not agree. Density is measured by the weights from the matrix described in equation 4.3 divided by the number of neighbors. Both criteria are combined in order to find samples in high density regions with high uncertainty. These are then labeled by a human in the loop.

In the evaluation it has been shown that the proposed approaches for enriching the graph structure and the active learning are improving the semi-supervised label propagation. Results are presented for large datasets, including up to 130000 samples from ImageNet. It is however worth noting that major improvements are achieved by sampling around both, training and testing samples.

4.1.4 *Recent Developments*

With the focus of research on deep learning and the availability of large datasets for supervised learning in the research community, semi-supervised learning has not been a popular paradigm (cf. [HMC17]). While there are several semi-supervised approaches that apply semi-supervised learning for label propagation on graphs [ZBL+04, LC09] or methods for traditional classifiers, like SVMs [TBZT07, FWT09, EFS13], there has not been much work after the rise of deep learning.

Note that recently, tasks like weakly supervised learning (see section 2.1), became very popular [DAF12]. In weakly supervised approaches the level of supervision that is given for training a detector is lower than its desired output. Typically, an image is annotated with respect to the presence of certain classes and the discriminative information is learned during training such that at test time, both the presence and the location of an entity can be predicted. These approaches are based on the fact that assigning a label for the presence of an entity is easier than annotating a complete bounding box for different instances [BRFFF16]. Prominent examples for weakly supervised object detection apply conditional random fields (CRFs) or CNNs and include [DAF12, ZKL+16, KL16]. In [DAF12] a CRFs is used in order to find similarities between images containing the same object classes, with localizations being based on regions proposals. In [ZKL+16] the activations from a

CNN, which is solely trained based on image labels, are leveraged for detecting objects. This approach is extended by the use of micro-annotations for a more accurate localization in [KL16]. In a recent work by the author of this thesis and colleagues, it could also be shown that point-wise localizations (cf. [BRFFF16]) in combination with point-wise mutual information can be leveraged for a more accurate localization [GSF18].

Regarding semi-supervised learning for image classification, there are only a few recent works. Some good examples are [HMC17, WJW+17, DVG16]. All of these deal with semi-supervised learning methods incorporating CNNs. In [HMC17], a network is trained on labeled as well as unlabeled images. Feature representations are derived from an intermediate layer of the network. It is assumed that these representations are similar for samples of the same class. Therefore, representations of unlabeled samples can be associated with the class of similar representations of labeled samples. A loss function is defined that weights labeled and unlabeled samples according to similarity of their feature representation. The evaluation is performed on datasets with a limited number of classes like MNIST, Google Street View House Numbers and a ten class object classification dataset, where promising results are achieved. In [WJW+17], an auxiliary clustering is introduced in order to model the structure of labeled and unlabeled images. A two-stream residual network is proposed that learns not only a classification function, but also attempts to learn a function that resembles the auxiliary clustering. For the classification function a weighted cross entropy loss is applied that considers labeled as well as unlabeled samples. In order to incorporate the unlabeled samples, weighted labels are derived based on pairwise similarity constraints. It is shown that the simultaneous training of both tasks helps to improve the network's performance. The approach in [DVG16], directly follows the method proposed in [DVG13]. Based on consistency criteria, a training dataset that consists of labeled as well as unlabeled samples is clustered into sets of different prototypes. A single prototype consists of multiple clusters and a classifier is learned for each one of the prototypes. For a given sample, the classifier of each prototype is evaluated and the resulting similarities are concatenated in order to compute a new feature embedding. This new feature embedding, therefore, considers information from all samples. In conjunction with the class information from the labeled samples this representation is used for training a final classifier. While the work in [DVG13] used traditional handcrafted features, the latter work employed features from a pre-trained CNN [DVG16].

In the following, it will be shown that the proposed approach for semi-supervised learning is applicable for traditional image classification approaches, i.e., using handcrafted features and Support Vector Machines (SVMs), but it can also be applied for training CNNs in a semi-supervised manner. Similar to [WJW+17, DVG16], a two-staged approach is followed. For example, the approach from [DVG16] uses labeled and unlabeled samples in order to learn a new feature representation which is then used as input for learning a final classifier. In the proposed approach, features are clustered independently in multiple views and actively labeled in order to derive a partially labeled sample set which is then used for training a final classifier. In case of the handcrafted features, a single representation is chosen for training the final classifier. Alternatively a CNN is directly fine-tuned based on the partially labeled dataset.

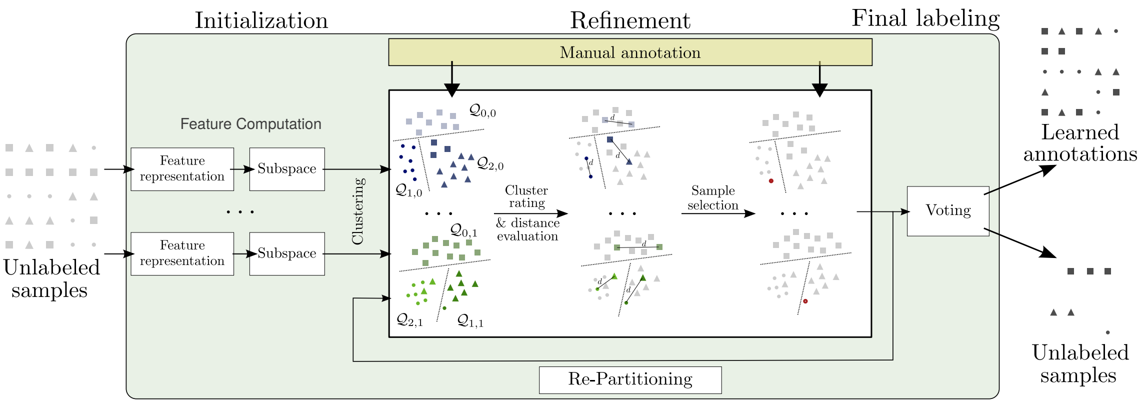


Figure 4.4: Overview of the iterative partition-based scene learning approach. The method is initialized by clustering the data independently in multiple feature representations. Each feature representation has previously been reduced in it's dimensionality by projecting it into a subspace. The clustering is iteratively refined based on a multi-view cluster scoring and distance evaluation. Only a few representative samples, namely the centroids are manually labeled so that labels are inferred for the unlabeled samples, resulting in a partially labeled sample set.

4.2 ITERATIVE PARTITION-BASED SCENE LEARNING

In the following section the proposed iterative partition-based learning approach is presented. The method assumes that data has been collected without any labels and then attempts to infer knowledge from the dataset in a semi-supervised manner. The method is illustrated in figure 4.4 and consists of three main steps:

1. The learning algorithm is initialized by computing different feature representations. The feature vectors of each representation are projected into a subspace in order to create a compact representation by reducing their dimensionality. Each of these representations is considered as a different view on the data and clustered independently. The centroids of each cluster are manually labeled.

2. The partitions which are created by the initial clustering are then refined by iteratively adding new partitions. The goal is to add partitions in regions where little knowledge about the samples can be inferred from the current partitioning of the feature spaces. Thus, a multi-view cluster and distance evaluation is computed in order to find these regions. The centroids of the new partitions are also manually labeled.

3. The labeled partitions are then used for inferring labels for the remaining unlabeled samples, resulting in a partially labeled dataset. Samples where no class can be assigned with high certainty remain unlabeled.

The iterative refinement can be terminated after a certain percentage of the dataset is labeled or after a fixed number of manual labeling operations. The learned annotations are then used for training a visual classifier. The main goal of the semi-supervised learning approach is to reduce the required manual labeling effort. Thus, the final classifier is expected to have a similar recognition rate as a classifier trained in a supervised manner, but in the semi-supervised case fewer samples were labeled for training the classifier.

4.2.1 *Feature Representations*

For the initialization of the method, U feature representations are computed. These representations are considered as different views on the data which serve to improve the robustness of the learning approach.

There are multiple choices for these feature representations. On the one hand, handcrafted features can be computed, like the GIST of a scene or an LBP histogram (see section 2.2.1). Local handcrafted features can also be integrated in a learned representation, like the Bag-of-Features. For example, the method introduced in chapter 3 allows for computing a compact feature representation that can be incorporated in the learning process. On the other hand, feature representations can also be derived from a CNN. A model that is pre-trained on a general image recognition task, such as ImageNet, is evaluated on the target task in order to derive a feature representation for the semi-supervised learning. If the target task is coarsely related to the pre-training dataset, a pre-trained network can be used as a general feature extractor [RASC14]. Thus, the activations from an intermediate layer are used as a feature representation for a given image.

4.2.2 *Initialization*

Since all of the aforementioned representations are very high dimensional, they are neither ideal for creating partitions using common clustering algorithms, nor for computing meaningful cluster scores (cf. [PHL04]). Thus, each feature representation is reduced in its dimensionality by projecting the features into a subspace. A lower dimensional representation, which is non-sparse, is desirable for several reasons:

1. Common clustering approaches assume that the underlying feature space is Euclidean. However, the Euclidean distance is less meaningful in sparse, high dimensional spaces (cf. [AHK01]).

2. Cluster evaluation methods that measure the overlap between different cluster partitions are based on the same assumption. They are, therefore, also less meaningful in sparse representations.

3. It is computationally more efficient to cluster datasets consisting of several thousand samples in a lower dimensional feature space.

On an abstract level, there are two alternative approaches for creating a lower dimensional subspace. On the one hand there are transformations that compute new features by computing linear combinations of a given feature representation and uncovering a latent structure. On the other hand there are feature selection methods which attempt to find the most relevant dimensions of a given feature representation.

While feature selection methods are often used in the context of subspace clustering, they assume that not all dimensions carry relevant information. They are also often applied in applications where interpretability is a crucial issue (cf. [PHL04]). For typical image representations that include a feature learning, the learned representations are already abstract features that are often difficult to interpret. Although these features are derived from the dataset, they do often contain redundancies and some of the features might be correlated (cf. [SHK+14, RGF17]). It is therefore appropriate to compute a transformation [PHL04].

In the context of Bag-of-Words representations, these subspaces are often referred to as topic spaces. It has been shown that, up to a certain point, topic space transformations allow for creating lower dimensional feature representations without loosing descriptiveness (cf. [BZM06, DDF+90]). In the context of the proposed iterative partition-based learning approach two alternatives, latent semantic indexing (LSI) and multidimensional scaling (MDS), are considered. The former one is a common topic representation for BoF representations and the latter one is a traditional approach for uncovering an underlying subspace of arbitrary feature representations.

Latent Semantic Indexing

Singular value decomposition (SVD), also referred to as latent semantic indexing (LSI), is a well studied subspace representation for histogram data [DDF+90]. Let $\boldsymbol{X_u}$ be the feature matrix of the u^{th} feature representation, where each column corresponds to a feature representation of a single scene image. LSI computes the decomposition of $\boldsymbol{X_u}$ as

$$\boldsymbol{\mathcal{U}\Sigma\mathcal{V}}^T = \text{svd}(\boldsymbol{X_u}) \tag{4.5}$$

such that $\boldsymbol{\mathcal{U}}$ and $\boldsymbol{\mathcal{V}}$ are orthogonal matrices, containing the left and right singular vectors, and $\boldsymbol{\Sigma}$ is a diagonal matrix containing the singular values. Using this decomposition, $\boldsymbol{X_u}$ can be approximated Choosing the K largest singular values from $\boldsymbol{\Sigma}$ and discarding the smaller ones, the matrix $\boldsymbol{X_u}$ is approximated by

$$\hat{\boldsymbol{X}}_{\boldsymbol{u}} = \boldsymbol{\mathcal{U}}_{\cdot,1:K} \;\cdot\; \boldsymbol{\Sigma}_{1:K,1:K} \;\cdot\; \boldsymbol{\mathcal{V}}^T_{\cdot,1:K} \tag{4.6}$$

and thus reduced in its dimensionality. The K features in the approximated matrix are a weighted combination of the previous features. Following the terminology of the visual words, these are also referred to as topics.

Multidimensional Scaling

Multidimensional scaling (MDS) is a traditional approach to determining an underlying feature space. The goal is to derive an embedding in a lower dimensional space that keeps the relative distances between the samples in order. Thus, in the ideal case MDS derives a monotonic embedding. The traditional MDS approach has been proposed in [Tor52]. In this approach, a target coordinate matrix $\boldsymbol{Y}$ shall be derived as the Eigenvalue decomposition of $\bar{\boldsymbol{O}} = \boldsymbol{YY'}$. Therefore, the following steps are performed:

1. A square matrix of pairwise Euclidean distances $\boldsymbol{D}$ between all training samples is computed. In the following, the squared matrix $\hat{\boldsymbol{D}} = \frac{1}{2}\boldsymbol{D}^2$ will be used. Given this distance matrix, the problem is to derive the matrix $\hat{\boldsymbol{O}}$.

2. As $\bar{\boldsymbol{O}}$ is a-priori unknown, the so-called *double centering* is applied, deriving a matrix $\boldsymbol{O} = -\boldsymbol{J}\hat{\boldsymbol{D}}\boldsymbol{J}$. Here, $\boldsymbol{J} = \boldsymbol{I} - n^{-1}\mathbf{11'}$ with n being the number of samples. As a result a single entry $O_{i,j} = \hat{D}_{i,j} - \frac{1}{n}\sum(\hat{\boldsymbol{D}}_{i,\cdot}) - \frac{1}{n}\sum(\hat{\boldsymbol{D}}_{\cdot,j}) - \frac{1}{n^2}\sum(\hat{\boldsymbol{D}}_{\cdot,\cdot})$ and thus the squared proximity subtracted by the average in its row, its column and the complete matrix.

3. The Eigenvectors and Eigenvalues of $\boldsymbol{O}$ are computed and the k largest ones are used for the embedding. The embedding of training samples is then the multiplication of the Eigenvectors and Eigenvalues.

For embedding an additional sample $\boldsymbol{x}$, a distance vector $\boldsymbol{d}_x$ containing the distance to all training samples is computed. The embedding is then computed by subtracting $\boldsymbol{d}_x^2$ from the mean of the pairwise distance matrix from the training set $\boldsymbol{\mu}_D$ and multiplying it with the normalized Eigenvectors $\boldsymbol{\mathcal{V}}$ by: $0.5\ (\boldsymbol{\mu}_D - \boldsymbol{d}_x^2) \cdot \boldsymbol{\mathcal{V}}$.

As this process is usually not suitable for large amounts of data, the so-called landmark MDS is often used as an approximation. This faster version of MDS uses a random subset of the training samples in order to determine the embedding. The remaining samples are then embedded as described above.

When applying MDS to arbitrary feature representations like the feature descriptor of a scene image, using the Euclidean distance is a strong constraint. In the non-metric MDS, first proposed in [She62], this constraint is removed. It has been shown that an arbitrary distance function can be used for computing the distance matrix $\boldsymbol{D}$. From this matrix a monotonic embedding is determined that keeps the original relations based on the given distance function. In the case at hand distance measures like the cosine distance may be of interest. Note that the target space is still a Euclidean space. MDS is, therefore, a very intuitive choice if the data shall be clustered using k-Means which assumes an Euclidean space.

Initial Clustering

After computing the subspace representations, the partitioning of the feature spaces is initialized. For initialization the concept of CBA is adopted by clustering the data of each subspace representation independently. Any method that allows for partitioning a set of feature vectors in an unsupervised manner is applicable. It is however important that the clustering allows for determining a meaningful centroid. In the following, Lloyd's k-means clustering is applied [Llo82].

In every feature representation, K partitions $\boldsymbol{\mathcal{Q}}_{\boldsymbol{u},\boldsymbol{k}}$ of the data are created. Each partition is then manually labeled. Ideally, the centroid of each partition is presented to a human annotator for labeling. As the centroids do not necessarily match with existing samples, the samples closest to the centroids are chosen. Using the manually assigned labels, a matrix $\boldsymbol{Y}$ is created which assumes that the labels are propagated to all samples within a partition. For the i^{th} sample, the labels assigned in the different feature representations are represented by C-dimensional binary vectors

$$[Y_{i,u,1}, \dots, Y_{i,u,C}]^T \in \{0, 1\}, \quad u = 1, \dots, U \quad , \tag{4.7}$$

where C is the number of classes. An entry in $Y_{i,u}$ is set to one if the label of the sample's partition equals c. It is set to zero for all other class indices.

As a result, the number of manual annotations which are required for the initialization is reduced to $U \cdot K$. If samples of a certain class are well separated from those of other classes, they are often already correctly labeled by the initial partitioning of the feature spaces. However, for tasks like natural scene recognition it is rarely the case that the clusters are well separated. For the majority of samples it is rather difficult to assign a class due to the high intra class variability and ambiguities. It is, therefore, interesting to further explore the feature space by iteratively refining the initial partitioning.

4.2.3 *Identifying Regions for Refinement*

In order to refine the partitioning, it is important to determine regions where the certainty of the initial labeling is comparably low. The general idea is therefore similar to the framework proposed in [EFS12]. To achieve this goal, two assumptions are made: First, a sample's label confidence is low if it is in partitions that are not well separated from other ones or if it belongs to partitions with varying labels in the different feature representations. Second, since the labeling is based on the partitions' centroid, the uncertainty is higher the further away from a centroid a sample is.

Based on the first assumption, the separation of clusters is evaluated. Typical choices to evaluate the separateness of clusters include evaluation measures like the Dunn index [Dun73], the Silhouette Score [Rou87] or the Davies Bouldin measure [DB79]. Here, a modified version of the Dunn is proposed. The Dunn index, as proposed in [Dun73], computes the ratio of the minimal inter cluster distance and the maximal intra cluster distance in order to evaluate clustering algorithms. Thus, the higher a Dunn index is, the better is the separation of the partitions that are computed by the clustering. In the proposed version, the Dunn index is evaluated for each partition separately and also incorporated the preliminary label information:

$$\operatorname{dunn}(u, k) = \frac{d_{\text{inter}}(\boldsymbol{\mathcal{Q}}_{\boldsymbol{u,k}}, u)}{d_{\text{intra}}(\boldsymbol{\mathcal{Q}}_{\boldsymbol{u,k}})} \tag{4.8}$$

where d_{intra} represents the intra partition distance and d_{inter} represents the distance between two partitions that have different labels:

$$d_{\text{intra}}(\boldsymbol{\mathcal{Q}}) = \frac{1}{|\boldsymbol{\mathcal{Q}}|} \sum_{\boldsymbol{x_i} \in \boldsymbol{\mathcal{Q}}} \underset{\boldsymbol{x_j} \in \boldsymbol{\mathcal{Q}}}{max} \; d(\boldsymbol{x_i}, \boldsymbol{x_j}) \tag{4.9}$$

$$d_{\text{inter}}(\boldsymbol{\mathcal{Q}}, u) = \frac{1}{|\boldsymbol{\mathcal{Q}}|} \sum_{\boldsymbol{x_i} \in \boldsymbol{\mathcal{Q}}} \underset{\boldsymbol{x_j} \notin \boldsymbol{\mathcal{Q}};\ \boldsymbol{Y}_{i,u,\cdot} \neq \boldsymbol{Y}_{j,u,\cdot}}{min} \; d(\boldsymbol{x_i}, \boldsymbol{x_j}) \tag{4.10}$$

As the semi-supervised labeling already provides labels for the clusters, it is beneficial to only consider partitions with different labels when evaluating the distance between the partitions. As a result, partitions which are close to each other, but have the same label do not influence the modified Dunn index negatively. For example, if a well separable class has a large number of samples, these samples might be represented by several partitions due to the initialization of the clustering algorithm. It is, however, not desirable to create a new partition in this region of the feature space.

Besides the pure separability of the clusters it is also of interest to consider the cluster's labels. Each sample is therefore associated with the product of the modified Dunn index and the labels that were assigned to the respective partitions in the different feature representations:

$$d_{\text{dunn}}(i) = \max_c \frac{1}{U} \sum_u \text{dunn}(u, q(\hat{\boldsymbol{X}}_{u,i})) \cdot \boldsymbol{Y}_{i,u,\cdot} \tag{4.11}$$

where $q(\cdot)$ assigns a sample to its partition. By computing the maximum over all classes, samples that belong to partitions with a high Dunn index and where the different feature representations agree on a label will be assigned a high combined score. Samples that belong to partitions with a low Dunn index and where the labels do not agree will be assigned a low combined score. This integration of the current labels is an important aspect of the learning process as it is desirable to add as much information as possible to the active learning. Reliable knowledge about the classes is obtained from the already labeled samples in the different feature representations.

The second assumption, considers the uncertainty higher the further away from a cluster center a sample is. It is evaluated by a distance measure:

$$d_{\text{dist}}(i) = \frac{1}{U} \sum_u \frac{1}{1 + d(\hat{\boldsymbol{X}}_{u,i}, q(\hat{\boldsymbol{X}}_{u,i}))} \tag{4.12}$$

The function is evaluated for each sample independently. It computes the inverse distance to the respective cluster centers in each feature representation. While d can be an arbitrary distance function, the Euclidean distance will be used in the following.

Both terms are then combined in a target function that is evaluated for all samples:

$$i^* = \underset{i}{argmin}\ w\ d_{\text{dunn}}(i) + (1-w)\ d_{\text{dist}}(i) \quad , \tag{4.13}$$

with w being a weighting factor. In practice, both values will between zero and one for a region with high uncertainty and thus a uniform weighting of both terms has shown to work well. The sample with the lowest combined score is chosen to be labeled by an annotator and is added as an additional centroid in all feature representations. Similar to the RBA the additional labeling effort is therefore reduced one sample at a time and independent from the number of feature representations.

In summary, the Dunn term causes the minimization to focus on regions where the ensemble does not agree and the partitions are not well separated. This term can be interpreted as contextual information, where the neighborhood of a sample is taken into account in form of its respective partition. The distance term helps to select a single sample for the refinement. It focuses on samples that are farther away from their respective centroids. As a result the samples computed by this refinement strategy will focus on either outliers in regions where no knowledge can be inferred or on class borders where the ensemble disagrees.

4.2.4 *Update of the Partitions*

After labeling a new sample, the partitions are updated by re-assigning all samples to the centroids. The update can be computed by the same update rule as used in the clustering processes, creating a new partitioning of the feature spaces. In case of k-means clustering each sample is assigned to the closest centroid, yielding a new Voronoi tesselation of the feature space.

Updating the partitions also includes an update of the label matrix $\boldsymbol{Y}$ as the labels for the samples may have changed after the update. Furthermore, the update of the partitions also requires to re-evaluate the Dunn scores. While this is computationally expensive, it can be efficiently solved by updating only those scores that are influenced by the re-partitioning:

- The distance d_{intra} can only change for those partitions that formerly contained samples which are assigned to the newly added partition.

- The distance between the partitions is more difficult, as it contains the constraint that only partitions with a different label are considered. Given that only partitions with labels from a subset of classes $\mathcal{C}$, which includes the newly assigned label, have changed by adding a new partition, it is possible to constrain the computation. If a partition has not been changed by the refinement, then its distance d_{inter} cannot change if its label $c \notin \mathcal{C}$. The nearest sample i with a different label would be the same, no matter what the actual label of sample i is.

In practice, these constraints are able to improve the computation time of the Dunn index by up to 80%, depending on the number of partitions and classes in the dataset.

Typically, the partitions that are computed by adding new centroids in the refinement steps are rather small. It will, however, be shown that they contain valuable information for training a visual recognizer. The refinement process is iterated until either all scores are sufficiently high, a given percentage of the dataset is labeled or a fixed number of labeling operations has been performed.

4.2.5 *Label Propagation*

In a final voting step the label matrix $\boldsymbol{Y}$ is used for assigning labels to the so far unlabeled samples of a dataset. Applying majority voting for the i^{th} sample results in an ensemble decision for a specific class label

$$y_i^{max} = \begin{cases} \underset{c}{\operatorname{argmax}} \sum_{u=1}^{U} Y_{i,u,c} & \text{if } \underset{c}{\max}\, Y_{i,u,c} > \frac{U}{2} \\ \emptyset & \text{otherwise ,} \end{cases} \tag{4.14}$$

where $\emptyset$ is a rejection class. Thus, labels are only assigned for samples where the class membership is determined with a majority agreement and, therefore, with a sufficient certainty. If the goal is a very precise labeling (e.g. tagging images) other voting schemes, like unanimity voting, might be considered.

The samples that were assigned a label by the semi-supervised learning algorithm will be used for training a visual classifier. Although the majority voting is not as precise as, for example, a unanimity voting, it is beneficial to label more samples at the cost of a slightly noisy set of labels. Nevertheless, the rejection process is an important difference to several semi-supervised learning algorithms that aim toward labeling all samples and, therefore, tend to make mistakes in regions of the dataset where not enough knowledge can be inferred from the manually labeled samples.

4.3 EXPERIMENTS

In this section the proposed method for iterative partition-based annotation learning (PBA) in large image collections is evaluated on two datasets comprised of natural scene images. The first one is the 15 Scenes dataset [LSP06] and the second one is the SUN397 dataset [XEH+14], a branch of the SUN corpus that contains a large collection of natural scene images. A detailed description of both datasets can be found in section 2.5.3.

In the experiments, all training samples are considered as initially unlabeled. These samples are then subsequently labeled by a human in the loop. In order to do so either the proposed approach, a semi-supervised method from the literature or supervised learning using a random subset of labeled samples is applied. Note that the figures in the following evaluations will show the results up to a certain percentage of labeled training samples.

Furthermore, in the experiments, handcrafted feature descriptors are evaluated alongside features that are derived from a CNN. The latter ones are expected to show better results, as these features are learned on large image corpora such as ImageNet and show state-of-the-art performance [RASC14]. The purpose of these experiments is to demonstrate the integrability of these feature into the proposed semi-supervised learning approach. A detailed description of the handcrafted features as well as the feature computation based on CNNs can be found in section 4.3.1 and section 4.3.2. The experiments for the two datasets are presented in the subsequent section 4.3.3 and section 4.3.4.

4.3.1 *Handcrafted Feature Representations*

In order to determine meaningful feature representations for the experiments, a set of handcrafted features is chosen. These features will be evaluated in a preliminary experiment on the 15 Scenes dataset. The following five handcrafted feature representations are considered in the evaluation:

- **SIFT xy tiling (c.f. section 3.2.1.1):** SIFT descriptors are extracted on a dense grid. A step size of 5 px is chosen and features are extracted at multiple scales using bin sizes of $4, 6, 8, 10$ px. The descriptors are quantized into a codebook of spatial visual words. Following the experiments in section 3.3, a 2×2 tiling is chosen for the spatial modeling. For the BoF representation, square rooted histograms are used. While the experiments in section 3.3 showed that large codebooks of several thousand spatial visual words yield a high classification accuracy, informal experiments determined that such high dimensional representations will not be beneficial for the learning process as it relies on clustering and a dimensionality reduction. Therefore, a codebook size of $1,000$ spatial visual words is chosen.

- **SIFT radial tiling (c.f. section 3.2.1.2):** the same descriptors as for the xy tiling are used. Again a BoF representation with $1,000$ spatial visual words and square rooted histograms is computed. Instead of a 2×2 tiling, a radial-tiling with two circles is computed.

- **SURF xy tiling:** SURF descriptors [BTG06] are extracted on a dense grid. The same parameters as for the SIFT representation are chosen and a BoF representation is computed.
- **LBP Histograms:** a histogram of rotation invariant Local Binary Patterns [OPM00] is computed. At each pixel 12 comparison points are chosen on a circle with a radius of one. The intensity values for the comparison points are computed by a bilinear interpolation of the pixel intensities. A pyramid scheme is built that computes an LBP histogram for 3×3 tiles and a histogram of the complete image is derived using max pooling.
- **GIST:** the GIST of a scene is computed using a spatial envelope representation [OT06], as described in section 2.2.1.1. The descriptor is computed using three channels (RGB) and three scales with 12, 12 & 4 orientations. This yields a $1,344$ dimensional representation.

4.3.2 *CNN-based Feature Representations*

Besides the set of handcrafted features, it is also considered to derive features from a CNN. In order to apply the proposed iterative partition-based annotation learning (PBA), different views on the data have to be derived from the network. While different pre-trained networks or different network architectures could be used to compute these views, this requires the training and subsequent evaluation of multiple deep networks. It is, however, also possible to derive multiple feature representations from a single network. Following the approach of [GG16, GSF17], it is assumed that a deep network that is trained with dropout, resembles an ensemble classifier which is based on the different paths through the network. Dropout is applied at test time so that different feature representations are sampled from the network. In [GG16] this approach is used for modeling deviations and uncertainty associated to predictions of the network. In practice, this also allows to compute multiple feature representations of the same input data using the same pre-trained network.

In the following, a VGG16 network that is pre-trained on the ImageNet corpus (see section 2.5.1) is chosen for the evaluation [SZ14]. Dropout is applied to the first fully connected layer (typically denoted as *fc-6*) and the features from the second fully connected layer (*fc-7*) are then chosen as a feature representation.

4.3.3 *15 Scenes Dataset*

In the first experiments, the approach is evaluated on the 15 Scenes dataset. This rather small dataset contains natural scene images from 15 different categories. In contrast to the supervised experiments, 200 images per class (= 3000 in total) are chosen as unlabeled training samples in order to have a relatively large collection of unlabeled training samples. The remaining 1485 samples are used for testing.

SIFT xy	SIFT radial	SURF xy	LBP hist	GIST
82.1 ± 1.1	82.6 ± 0.7	68.0 ± 0.7	68.8 ± 1.1	72.0 ± 1.0

Table 4.1: Accuracy [%] of the different handcrafted features on the 15 Scenes dataset in a supervised experiment. For each class 200 training samples (= 3000 in total) have been used. The experiments were repeated five times and the mean and standard deviation are reported.

Evaluation of handcrafted features

In a preliminary experiment, the quality of the handcrafted feature representations is evaluated in a supervised experiment on the 15 Scenes dataset. All 3000 training images are used in this experiment in order to derive an upper baseline for the quality of the handcrafted features. The experiment is repeated five times and the mean and standard deviation are reported in table 4.1.

The results show that the BoF representations which are built on SIFT features, alongside with the global GIST representation, are the best performing features. For the following experiments, a combination of three different feature representations will be chosen, as the number of labels in the initialization relies on clustering in multiple feature representations and, therefore, the number of manual labels increases with the number of feature representations. This choice is also in accordance with previous work, i.e. [RVGF14]. Based on the results from the supervised experiment, a combination of SIFT and GIST feature representations is assumed to be beneficial for the semi-supervised learning. This assumption has also been confirmed by an exhaustive evaluation of feature combinations in informal experiments. Hence, for the further experiments this setup is considered.

Initialization

For the initialization of the PBA two parameters are of interest. First, the dimensionality reduction algorithm and the target size of its representation. Second, the number of labeling operations which are assigned to the initial clustering step and the number of labeling operations that are performed in the refinement phase.

As the handcrafted feature representations that are based on local SIFT descriptors follow the BoF principle, LSI is applied for computing a topic space. LSI is a common approach for word histograms. For the target size, setups with $25, 50, 75$ and 100 topics are evaluated. It is assumed that the clustering step is assigned 90% of the labeling operations and the remaining 10% of the labeling operations are used for refining the partitions. The number of labeling operations that are used for the refinement are further investigated in the following. After labeling the samples in the training set in a semi-supervised manner, an SVM is trained on these samples using SIFT features with a 2×2 tiling. The SVM is then used as a final classifier. The results in figure 4.5 show that the best performance can be achieved with a low number of topics. More specifically, the best results are achieved with 25 or 50 topics. It is also shown that in comparison to an approach that does not use a dimensionality reduction, the recognition rate can be improved while at the same time reducing the computation time. It has to be noted that with an increasing number of samples, the partitions become smaller and thus the advantage of having a compact

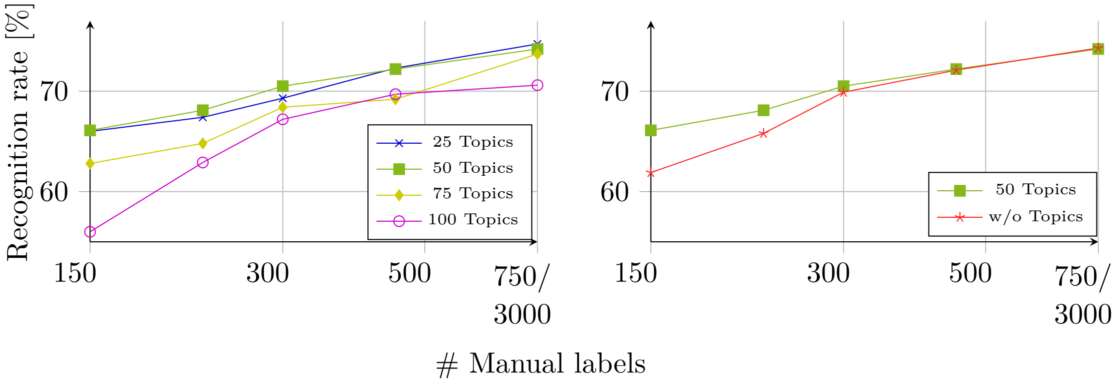


Figure 4.5: Recognition rates of an SVM on the 15 Scenes dataset. The SVM is trained on samples that are labeled in a semi-supervised manner by PBA. Labels are assigned for 5% up to 25% of the training set. (Left) Different sizes for the topic space that is computed by LSI. (Right) The topic space compared to running the method without any dimensionality reduction. The experiments were repeated five times and the mean is shown.

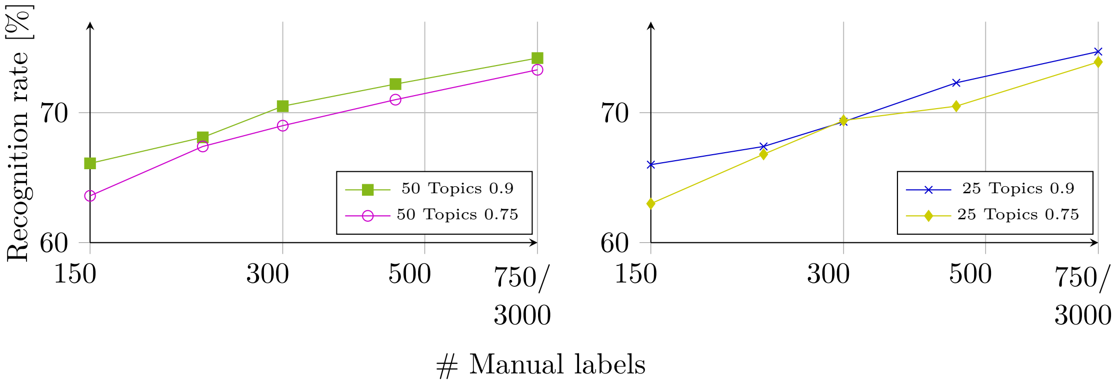


Figure 4.6: Recognition rates of an SVM on the 15 Scenes dataset. The SVM is trained on samples that are labeled in a semi-supervised manner by PBA. Labels are assigned for 5% up to 25% of the training set. (Left) Initializations using 75% and 90% of the labeling operations for the initial clustering and 50 topics computed by LSI. (Right) Initializations using 75% and 90% of the labeling operations for the initial clustering and 25 topics computed by LSI. The experiments were repeated five times and the mean is shown.

representation diminishes as only a few relatively close samples need to identified based on the distance measure. It can be assumed that the dimension reduction will be more important on larger datasets.

In a second experiment, the number of labeling operations that need to be assigned to the iterative refinement are further investigated. The two best performing topic space sizes, using 25 and 50 topics are evaluated. Two cases, one using 10% and one using 25% of the labeling operations for the refinement, are evaluated. Again an SVM is used as a final classifier. The results are shown in figure 4.6. It can be observed that the approach performs better if more labeling operations are assigned to the initial clustering step.

Both configurations perform better when using only 10% of all labeling operations in the refinement. This can be explained by the fact that each feature representation is clustered independently and thus the clustering step requires more labels than the refinement. Based on these findings, the size of the topic space is set to 50 and 10% of the labeling operations are used for refining the partitions in the following experiments.

Comparison with results from the literature

Based on the preliminary experiments, the iterative partition-based annotation learning (PBA) is compared to methods from the literature. The proposed approach is, therefore, compared to supervised classifier training and three semi-supervised approaches: cluster-based annotation learning (CBA), retrieval-based annotation learning (RBA) (see section 4.1.1 and section 4.1.2; [RVGF14]) as well as the graph-based learning with local and global consistency (LLGC) (see section 4.1.3; [ZBL$^+$04]). The semi-supervised methods from the literature were re-implemented. Furthermore, results reported in [DVG13, DVG16] are shown as comparison with another method that incorporated both handcrafted features as well as CNN-based features. All approaches, including the supervised learning, are evaluated using the same training and test sets. This is a difference to other semi-supervised setups, where additional unlabeled samples are added to a pool of labeled training samples. As for the former experiments, an SVM using SIFT features with a 2×2 tiling is used as a final classifier. For the supervised classifier training a given number of samples is randomly chosen from the pool of unlabeled samples and assigned its label. For LLGC these samples are used for initializing the graph structure and thus the subsequent label propagation. Note that drawing random samples implicitly resembles the class distribution (i.e. a uniform distribution) of the dataset. The semi-supervised learning has no knowledge about the true class distribution.

The results for all methods are shown in figure 4.7. The top figure shows the comparison with the other semi-supervised learning approaches. The proposed PBA outperforms the other algorithms and performs very well when the dataset is only sparsely annotated. With an increasing number of samples, LLGC is able to yield a similar performance. Most likely the larger number of initial samples allows for a robust initialization of the graph structure and, therefore, a robust labeling of all samples. For sparsely labeled datasets the rejection of samples where no label can be assigned with high certainty is an important advantage of the algorithm. In comparison with the two baseline algorithms CBA and RBA, the proposed methods outperforms both of them. Especially the purely distance-based RBA is significantly outperformed as long as the dataset is sparsely labeled.

In comparison with supervised classifier training there is a break-even point. For sparsely labeled datasets, the additional information that is gathered from the semi-supervised learning is beneficial. The more samples are labeled, the more reliable is the supervised learning, as no incorrect labels can be introduced to the classifier training. It can be observed that this break-even point is at roughly 25% of labeled samples within the training set. This is a good result considering that most semi-supervised setups work with a much larger set of unlabeled samples since it is usually easy to obtain additional unlabeled samples.

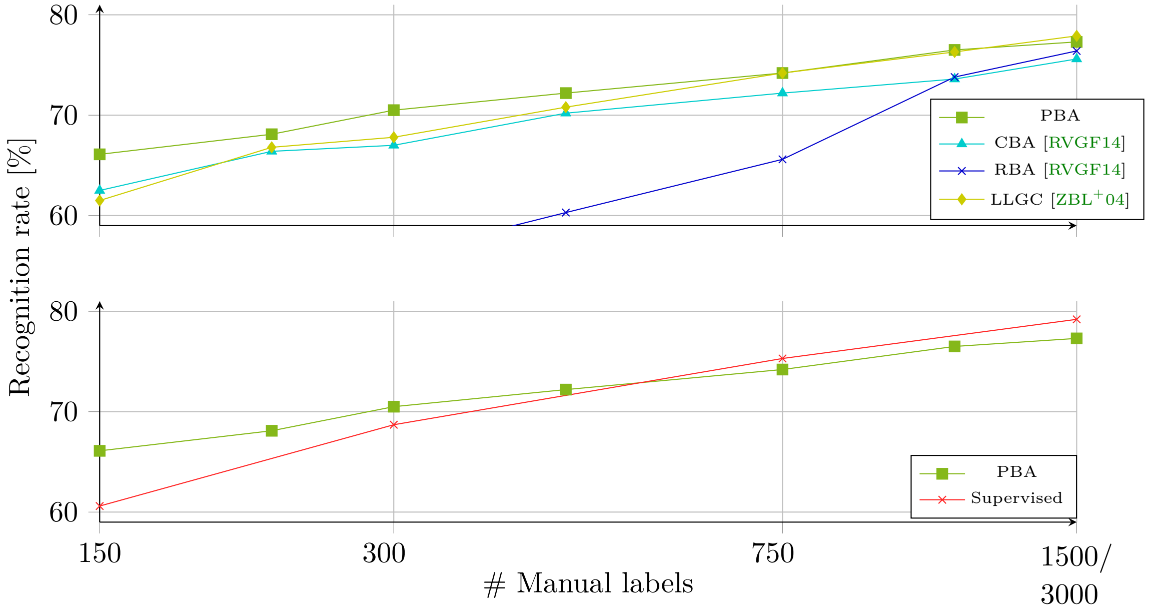


Figure 4.7: Recognition rates of an SVM on the 15 Scenes dataset. The SVM is trained on samples that are labeled in a semi-supervised manner. Labels are assigned for 5% up to 50% of the training set. (Top) Comparison of PBA with CBA, RBA and LLGC. (Bottom) Comparison with supervised classifier training. The experiments were repeated five times and the mean is shown.

Furthermore, the results of PBA can be approximately compared to the ones reported in [DVG13], where a similar two-stage approach to semi-supervised is proposed. A feature embedding is learned based on prototypes derived from labeled as well as unlabeled samples. This embedding is used in order to train a final classifier, e.g., an SVM or a logistic regression. Note that the evaluation protocol in [DVG13] slightly deviates from the evaluation described above. Most prominently, a fixed number of samples is selected per class so that the training samples are more balanced than in the proposed evaluation protocol. A combination of pyramidal HoG features, the GIST and LBPs are used as input features for the semi-supervised learning. The feature representations from [DVG13] should be similar in performance (cf. table 4.1). The comparison is shown in table 4.2. While the

		Reported in [DVG13]			
# samples	per class	Laplacian SVM	Logistic Regression	Lin. SVM	PBA
150	10	62.4	67.1	66.2	66.1
759	50	69.7	75.6	73.8	75.3

Table 4.2: Recognition rates [%] on the 15 Scenes dataset compared to the results reported in [DVG13]. The exact results were no longer available and were thus derived from figure 2 in [DVG13]. Note that the two evaluation setups deviate slightly from each other. Nevertheless, the feature representations and initial sample selection can be considered similar enough for an approximate comparison.

results in [DVG13] deviate depending on the final classifier, it can nevertheless be observed that the proposed PBA is in a similar range.

With respect to the labeling error it can be said that training a classifier is to some extent robust against incorrect labels. For 1500 labeling operations, as shown at the right-most point of the evaluation in figure 4.7, the PBA labels $80.9 \pm 0.8\%$ of the samples with $88.0 \pm 0.6\%$ of the learned labels being correct. Besides the robustness against incorrect labels, this also emphasizes why multiple feature representations are necessary as the best single view (SIFT radial) labels only $64.5 \pm 0.7\%$ of the samples correctly.

Features derived from CNNs

In the following experiments features are derived from the pre-trained CNN based on a VGG16 architecture. The network is pre-trained on ImageNet (see section 2.5.1). Features are extracted from the second fully connected layer. Dropout is applied at test time to the previous fully connected layer in order to generate different views on the data. For the final classifier, the pre-trained CNN is fine-tuned based on the samples that were labeled by the semi-supervised learning algorithm or the manually selected ones. The training images have been augmented using random translations $(0-5\%)$, flipping (50% chance) and Gaussian noise $(\sigma = 0.02)$ in order to avoid overfitting. In total 40 epochs or at least 100 iterations of batchsize 128 have been used for training. The learning rate has been set to $\eta = 10^{-4}$ for the first twenty epochs and is reduced to $\eta = 10^{-5}$ for the subsequent twenty epochs. As for the traditional features, the experiments are repeated five times.

In the first two experiments, design decisions that have been made for the handcrafted feature representations are reviewed. As LSI is not an obvious choice for computing a lower dimensional feature representation for feature vectors derived from a CNN, it is compared to the more general MDS. The results are shown in figure 4.8. It can be seen, that the results obtained by MDS are more stable than those achieved by LSI. MDS is evaluated with two different underlying distance measures, the cosine distance and the Bray-Curtis dissimilartiy. The best results are achieved using the Bray-Curtis dissimilarity. It can be observed that for the Bray-Curtis dissimilarity the size of the topic space does not appear to strongly influence the performance of the learning. However, informal experiments revealed that the results obtained with 75 topics show slightly better results for more than 750 labels. For the features derived from CNNs it might also be of interest to have a closer look at the voting scheme. Since the feature representations are derived from the same pre-trained network, it can be assumed that they are still to some extend similar to each other. Therefore, unanimity voting is compared to majority voting in figure 4.9, as it can be argued that for rather similar feature representations all three representations should agree on the label. The results show that no significant difference can be observed with respect to the final performance. It can be assumed that if the representations agree on a label almost always all three representations agree, whereas for samples with a high uncertainty the representations do completely disagree.

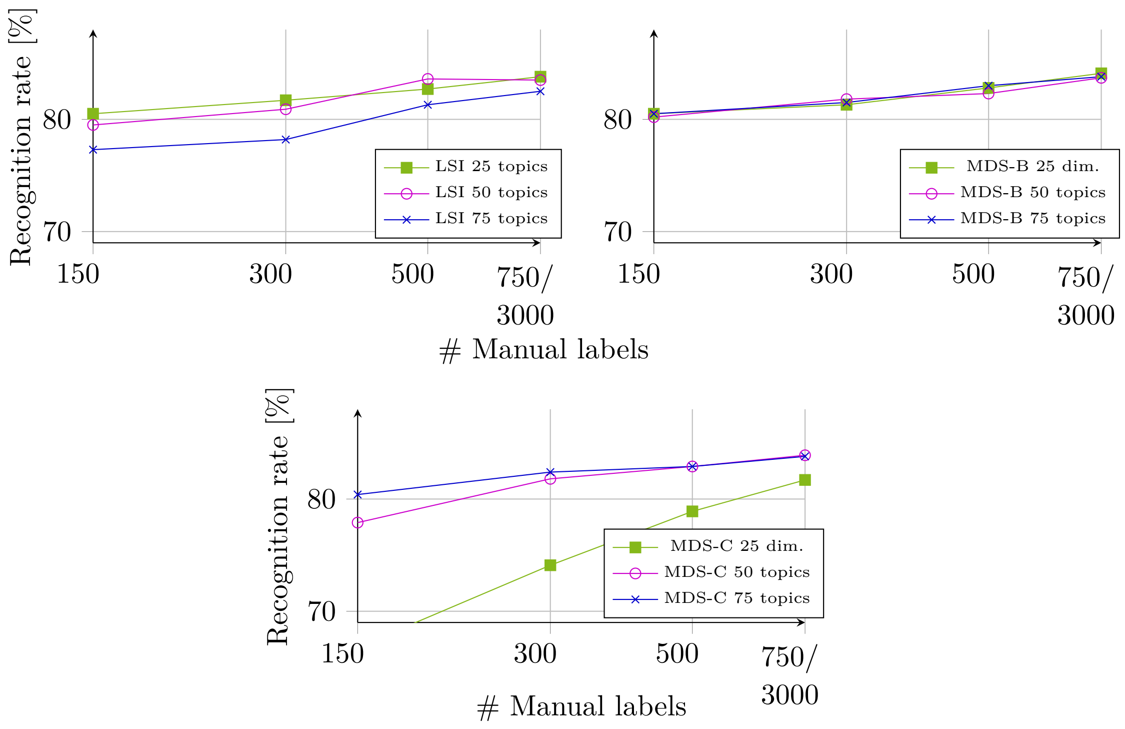


Figure 4.8: Recognition rates of a CNN on the 15 Scenes dataset. Features are initially extracted from a pre-trained network in order to label samples in a semi-supervised manner by PBA. Labels are assigned for 5% up to 25% of the training set. Using these features, the CNN is then fine-tuned to the task at hand. (Left) Topics computed by LSI. (Right) Topics computed by MDS using the Bray-Curtis dissimilarity. (Bottom) Topics computed by MDS using the cosine distance. The experiments were repeated five times and the mean is shown.

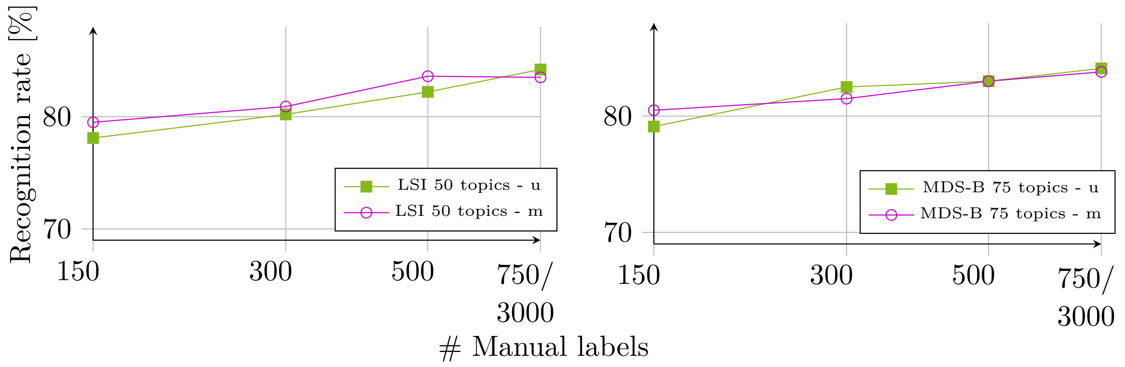


Figure 4.9: Recognition rates of a CNN on the 15 Scenes dataset. Features are initially extracted from a pre-trained network in order to label samples in a semi-supervised manner by PBA. Labels are assigned for 5% up to 25% of the training set. Using these features, the CNN is then fine-tuned to the task at hand. As the features derived from the same layer of a CNN using dropout at test time may be similar, unanimity voting is compared to majority voting. (Left) Topics computed by LSI. (Right) Topics computed by MDS. The experiments were repeated five times and the mean is shown.

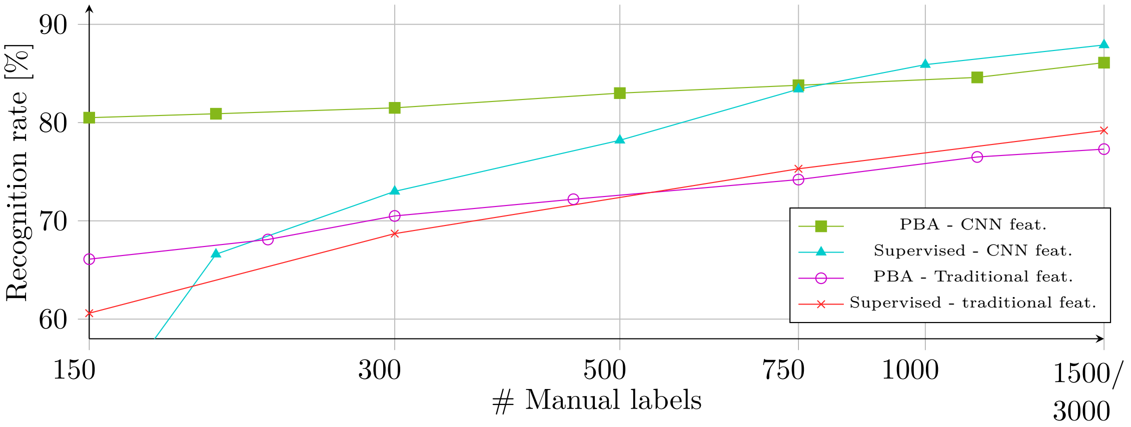


Figure 4.10: Comparison of traditional features and features derived from a CNN on the 15 Scenes dataset. Samples for training are either selected in a supervised manner or in a semi-supervised manner using PBA. Labels are assigned for 5% up to 50% of the training set. The experiments were repeated five times and the mean is shown.

In a last experiment on the 15 Scenes dataset, the best configuration of MDS using the Bray-Curtis dissimilarity and 75 topics with majority voting is compared to the traditional features. The results are shown in figure 4.10. The advantage of deriving features from a CNN becomes imminent. As for most Computer Vision tasks, the CNNs outperform the traditional approach by a margin. The results are directly transferable from supervised learning to semi-supervised learning. Not only for training a deep network from a random initialization, but also when fine-tuning a CNN it is a huge advantage to have more data for training the network. On a scarcely labeled dataset, the CNN that is fine-tuned in a supervised manner performs rather poorly, whereas the CNN that is fine-tuned based on the samples learned by PBA already outperforms the traditional approaches by a large margin. The results also show that fine-tuning a network is surprisingly robust toward labeling noise as some incorrect labels will be inferred from the semi-supervised labeling. The break-even point when supervised learning is more efficient than semi-supervised learning is slightly later than for the traditional features.

Furthermore, as for the traditional features, the results shown in figure 4.10 are approximately compared to the ones reported in [DVG13, DVG16], where a slightly different evaluation setup has been applied. As for the proposed approach, the method introduced in [DVG13] is also evaluated with CNN-based features in [DVG16]. However, instead of fine-tuning a CNN the final classifier is either a k-Nearest Neighbor, an SVM or Logistic Regression. The comparison is shown in table 4.3. Although the evaluation setup in this thesis is slightly more difficult, it can be observed that the PBA results are in a similar range. The performance of PBA is always in between the results of the three classifiers evaluated in [DVG16]. The difference to the best performing approach in [DVG16] is smaller when the dataset is scarcely labeled, confirming the observations made before.

		Reported in [DVG16]			
# samples	per class	KNN	Logistic Regression	Lin. SVM	PBA
150	10	77.4	82.9	82.2	80.5
759	50	82.1	87.5	86.6	83.8

Table 4.3: Recognition rates [%] on the 15 Scenes dataset compared to the results reported in [DVG16]. The exact results were no longer available and were thus derived from figure 4.2 in [DVG16]. Note that two evaluation setups deviate slightly from each other. Nevertheless, the results can be used for an approximate comparison.

4.3.4 *SUN397 Dataset*

In the following experiments, the approach is evaluated on the SUN397 dataset, a branch of the SUN corpus. The dataset contains natural scene images from 397 different categories and contains more than 100 000 images. It is one of the largest benchmarks for natural scene images. Following the setup described in [XHE+10], 50 samples from each category are randomly chosen for training. The remaining ones for testing so that for every category at least 50 samples remain in the test set. Hence, the training set contains 19850 samples in total. As for the 15 Scenes dataset, the experiments are repeated five times. Handcrafted features as well as features derived from a CNN are evaluated using the same configuration as for the former experiments. For both types of features the best parameter configuration that has been observed on the 15 Scenes dataset is used for the evaluation.

Evaluation of semi-supervised learning using handcrafted features

In the first experiments on the SUN397 dataset, the handcrafted features are evaluated. Note that the recognition rates for methods based on handcrafted features is very low on this task (cf. [XHE+10]), which demonstrates that the benchmark is an extremely challenging task. Therefore, the performance of the supervised baseline is low on this task and, consequently, semi-supervised learning on this dataset will also be extremely difficult. The results for PBA, LLGC and supervised training are shown in figure 4.11. In all cases an SVM is trained based on SIFT features as a final classifier. At least 397 labeling operations are evaluated as this equals the number of categories in the dataset. The results show that semi-supervised learning is also beneficial for challenging recognition tasks.

Given that this collection of natural scene images is much larger than the 15 Scenes dataset, it can be assumed that even with a small number of manual labels there is a sufficient amount of similar samples that can already be labeled with high certainty. Both semi-supervised learning approaches do therefore outperform the supervised learning while the number of manual labels is low. Even with roughly 45% of the training set being labeled the classifier that is trained based on samples labeled by PBA shows better recognition rates than the purely supervised learning. The recognition rates that can be achieved when using the PBA are also able to outperform LLGC at most times. It can therefore be concluded that the proposed method, which uses multiple views on the data and refines

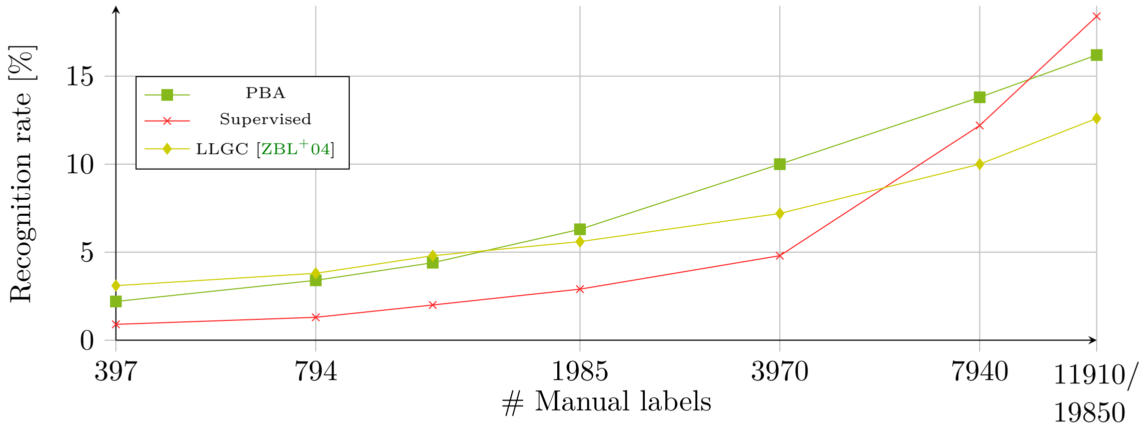


Figure 4.11: Recognition rates of an SVM on the 397 scene categories of the SUN397 dataset. The proposed PBA is compared training an SVM in a supervised manner and LLGC. Labels are assigned for 2% up to 60% of the training set. The experiments were repeated five times and the mean is shown.

the partitioning in regions without thorough knowledge about the samples, is useful for training a classifier on samples that are learned in a semi-supervised manner.

Features derived from CNNs

As for the 15 Scene dataset, features are also derived from a VGG16 network that is pre-trained on ImageNet. Features are extracted from the second fully connected layer using dropout at test time in order to generate different views on the data. For the final classifier, the same pre-trained CNN is fine-tuned based on the samples that were labeled by PBA or the ones that were labeled in a supervised manner. Again training images have been augmented using random translations $(0-5\%)$, flipping (50% chance) and Gaussian noise $(\sigma = 0.02)$ and 40 epochs or at least 100 iterations of batchsize 128 have been used for training. The learning rate has been set to $\eta = 10^{-4}$ for the first twenty epochs and is reduced to $\eta = 10^{-5}$ for the subsequent twenty epochs.

The results for the fine-tuned CNNs are shown in figure 4.12. The observations that could be made on the small 15 Scenes dataset show more clearly on the larger and more difficult SUN397 dataset. Ultimately, the CNNs outperform the traditional approach. However, while the dataset is only scarcely labeled, it is not possible to train a network in a supervised manner that is able to discriminate the large number of classes in the dataset. With less than 7940 samples (approx. 20 per class), the supervised CNN is outperformed by an SVM that is trained based on a BoF representation using SIFT features. As for the 15 Scenes dataset it is a huge advantage to fine-tune the network based on samples that were learned in a semi-supervised manner. The results that are achieved by the network that is fine-tuned based on samples that were labeled by PBA outperform the network that is fine-tuned in a supervised manner by a large margin. Even with only 397 labeled samples (approx. one per class), a recognition rate of 14.1% is achieved, which is still significantly

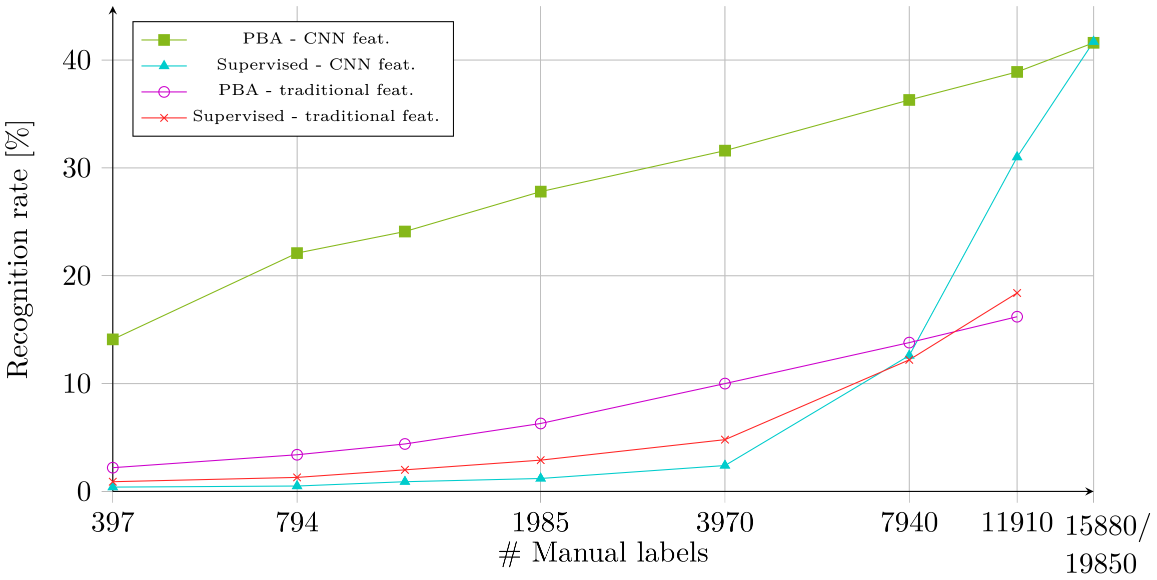


Figure 4.12: Comparison of traditional features and features derived from a CNN on the SUN397 dataset. Samples for training are either selected in a supervised manner or in a semi-supervised manner using PBA. Labels are assigned for 2% up to 80% of the training set. The experiments were repeated five times and the mean is shown.

better ($p \leq 10^{-5}$) than the supervised fine-tuning with 20 times as many labeled samples and a recognition rate of 12.6%. It is not until 80% of the dataset are being labeled that the performance of the supervised learning achieves the same results as the semi-supervised approach. At this point both networks show a performance of 41.7%. Note however that there is a sharp increase in the performance of the supervised approach starting at around 40% of the dataset being labeled and thus approximately 20 examples per class. When training a CNN with all 50 training samples per class, recognition rates of up to 63.2% are reported in the literature [ZKL+16]. This confirms the demand for a sufficient amount of labeled samples on complex image classification tasks, even when fine-tuning a pre-trained CNN.

4.4 CONCLUSION

In this chapter iterative partition-based annotation learning (PBA), a novel approach for semi-supervised learning in collections of natural scene images, has been introduced. The approach computes multiple feature representations as a foundation for a multi-view learning approach. Samples are clustered in each representation independently and labeled by a human in the loop. The approach is therefore an active learning approach. The clusters are refined iteratively, requesting further labels. The semi-supervised learning process terminates after a given number of labels has been assigned by the human in the loop. Finally, a voting over all views on the data is performed, resulting in a partially labeled set of training samples which are then used for training a final classifier.

Experiments were performed on two datasets, the small 15 Scenes dataset as well as the large SUN397 dataset. An evaluation of different parameterizations has been performed. A comparison with a traditional semi-supervised learning approach, a semi-supervised ensemble approach that uses either traditional or handcrafted features as well as it's two predecessors, CBA and RBA showed the effectiveness of the proposed approach. On scarcely labeled datasets, PBA outperforms supervised learning.

Furthermore, the application of traditional handcrafted features as well as features derived from a pre-trained CNN has been investigated. For the former case, multiple handcrafted feature representations have been evaluated and the learning has then been performed based on a combination of different BoF representations and a GIST representation. In the latter case, multiple feature representations are derived from a pre-trained CNN using dropout at test time. For the final classifier, the CNN is fine-tuned based on the samples that have been labeled in a semi-supervised manner. Unsurprisingly, features derived from a deep network are more powerful than traditional handcrafted feature representations. However, the experiments showed that the advantage of the proposed semi-supervised learning approach over supervised learning is even larger when fine-tuning a CNN. Even when fine-tuning a CNN based on pre-trained weights, i.e. derived from ImageNet, the amount of samples that is required for adapting the network may be comparably large. This is especially true on larger more complex tasks, like the SUN397 dataset. It could be shown that a network that is fine-tuned in a semi-supervised manner shows better results than a network that is trained in a supervised manner with 20 times more training samples.

Here, I made comments on the paper you wrote.
I got a little carried away with the red ink ...
- Prof. Smith, PhD Comic 690, 07.03.2006

5 APPLYING SCENE SEMANTICS FOR OBJECT PREDICTION

A preliminary version of the following work has been presented at VisApp - 2017 [GF17]. For the application to object detection an idea discussed in [GSF17] is integrated into the method, which has been presented at ICIP - 2017. The integration of this contribution in the context of this thesis is shown in figure 5.1.

In this chapter a novel approach for zero-shot object prediction is introduced. The work focuses on the semantic relations between scenes and objects. While most zero-shot methods rely on various visual attributes that describe a scene or an object [PXSH14, LNH14], here a more direct way is pursued. Instead of recognizing many attributes, a single entity, the scene category, is predicted. After recognizing the scene that is depicted in an image, the semantic relations between scenes and objects are used for deriving a probability that a certain object is present in a scene in a zero-shot manner.

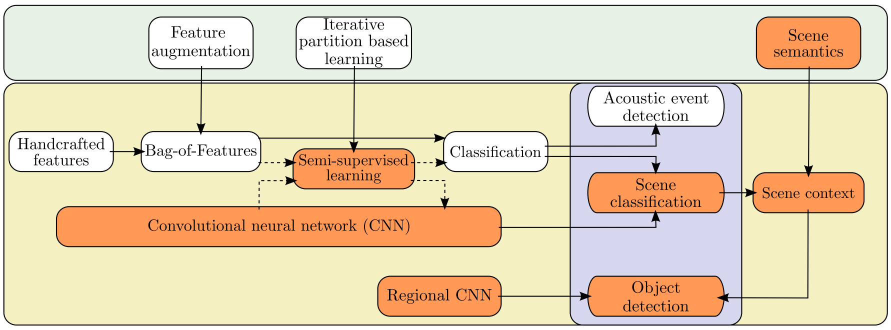


Figure 5.1: Integration of the *Scene Semantics for Object Prediction* in the context of this thesis (relevant topics are highlighted in orange). The chapter deals with a the combination of a visual scene recognizer that is based on a CNN and semantic knowledge from textual sources for the task of object detection. Applications for presence prediction as well as instance detection are shown. Furthermore, a combination with semi-supervised scene learning is shown.

Semantic knowledge can be a powerful source of information especially in scenarios with few or no annotated training samples. Most importantly, relations between scenes and objects can easily be obtained from external sources such as large scale text corpora from the web. They do, therefore, not require tremendous manual labeling efforts. In scenarios where labeling is scarce and visual recognition is difficult, scene knowledge and semantic information provide an important cue for object prediction.

Given that no visual information about the object classes is used, the proposed approach can also be seen as an object prior (see section 2.4.1; cf. [DHH$^+$09, CTW12]). Besides the task of object prediction, it will therefore be shown how these presence predictions can be incorporated in a context descriptor for a visual object detector (see section 2.1.2).

In the following, the related work regarding zero-shot recognition based on visual attributes or direct semantic comparisons and related work in the field of combining natural language and vision will be reviewed. The proposed approach for relating scenes and objects based on semantic co-occurrence will then be introduced. The approach is evaluated on the challenging SUN image corpus, which contains a wide range of visual scenes and objects, as well as the Pascal VOC2012 dataset for the object detection task.

5.1 RELATED WORK

There are two kinds of methods that are related to the proposed approach. The first ones are methods which rely on visual attributes in order to describe a visual scene or object. Thus, no training samples are given for the visual target classes. Such methods with no training are also referred to as zero-shot learning approaches. An overview of the related work regarding attributes is given in section 5.1.1. The second type of methods that is related to this work are the ones that combine natural language and vision, which are described in section 5.1.2. The presented approach combines both ideas in a novel way. As a result, an object prediction is made without any visual information about the object class, but including the visual information from the natural scene. The section ends with a discussion of recent related work that deals with the combination of visual and textual information.

5.1.1 *Zero-shot Learning with Attributes*

There are several methods for zero-shot learning based on visual attributes (cf. [PXSH14, LNH14]). In general, a set of visual attributes is defined and recognized within the image. Then, a set of previously unobserved classes is described in terms of their attributes. For example, an animal with the attributes *black, white* and *stripes* will most likely be a zebra. As a result, this also allows to tackle open set classes where a new class is defined by a new combination of attributes. In the evaluation of such zero-shot settings, one sample of a latent class is chosen and the goal is to retrieve all other samples of this class based on the attributes. This is repeated for all samples of the latent class. The evaluation is therefore very similar to an image retrieval task.

In [LNH14], different frameworks for zero-shot prediction are discussed. *Direct attribute prediction* is compared to *indirect attribute prediction*. An illustration of these frameworks and a multiclass classification (on the left side) is given in figure 5.2. In the direct approach, visual attributes are immediately recognized based on a set of image features. These attributes are then used for recognizing a set of previously unobserved target classes. In the indirect approach, a set of known classes is recognized using a multi-class classification. Then, a deterministic dependence between each of those classes and a set of attributes is assumed so that the attributes can be inferred from the result of the multi-class classification. Based on these attributes a set of formerly unobserved target classes is recognized. The direct attribute prediction requires to associate each image with a set of attributes. Depending on the number of attributes this may require a considerable annotation effort. The indirect attribute prediction has the advantage that a set of known classes can be used as a basis. It is only necessary to define a dependence between these classes and the attributes which may reduce the annotation effort. A detailed evaluation of several datasets is performed in [LNH14], with the most prominent one focusing on the recognition of attributes of animals. Each of the attributes is recognized independently. A nonlinear SVM is trained for each attribute. The learning is performed without any knowledge about the actual object classes. The attribute vector is then used in order to infer knowledge about object

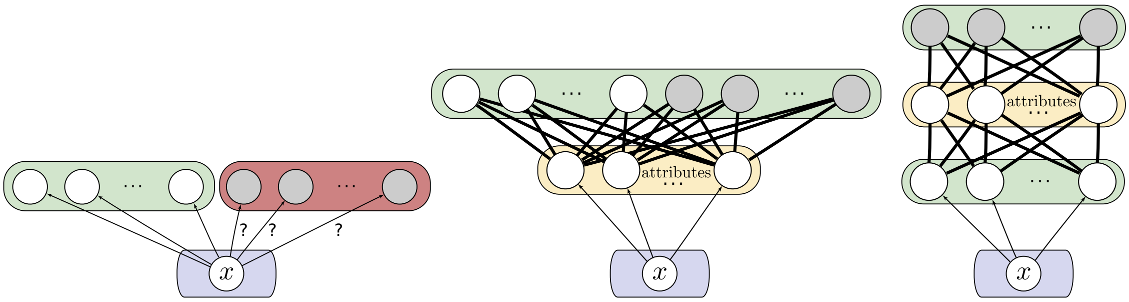


Figure 5.2: Comparion of multiclass classification, direct attribute prediction and indirect attribute prediction, based on [LNH14]. Given an input vector $\boldsymbol{x}$, the goal is to predict known (white) or previously unknown classes (grey). In a flat multiclass setting the unknown classes cannot be predicted. In the direct attribute prediction, a set of attributes is predicted based on the feature vector $\boldsymbol{x}$ and then known as well as previously unknown classes are described in terms of their attributes. In the indirect attribute prediction, a set of known classes is predicted and each of them is associated with a set of attributes. From these the prediction of a previously unknown class is derived.

classes in a retrieval setting as described above. For most evaluations, the direct attribute prediction shows better results for the task of zero-shot prediction than the indirect attribute prediction.

One prominent example of *direct attribute prediction* is [PXSH14], where a set of visual attributes is designed for the natural scene images in the SUN database. These attributes range from abstract definitions of functions and affordances, like *transporting things* or *construction*, over materials that are shown in the image to surface properties and ultimately the spatial envelope attributes, e.g., the *openness* of the scene (see section 2.2.1; [OT01]). In total 102 attributes are defined which describe a natural scene. Each of these attributes is recognized independently by a per-attribute classifier. In order to recognize the attributes, a set of features, containing histograms of oriented gradients, self-similarity, geometric and color features (cf. section 2.2.1; [XHE+10]), is computed for each image. For each attribute a linear SVM is learned based on this feature set. It is shown that attributes can be predicted with a high accuracy and that subsequently these attributes can be used for predicting scene classes. The evaluation is performed in both, a supervised as well as a zero-shot setting. In the supervised setting the predicted attribute representation is used as an input feature vector to a classifier. In the unsupervised zero-shot setting, similar samples of a latent class are uncovered based on their attributes as described above. While the results are promising, the accuracy in the supervised setting for scene classification is not superior to the accuracy of gradient based features [PXSH14]. It can, however, be assumed that the results can be improved with a more accurate attribute prediction.

The issue of an accurate attribute prediction is also tackled in further works. It could be shown that the approaches for direct attribute prediction can be improved based on the recent advances in Convolutional Neural Networks (CNNs). In [ZKL+16] a CNN is trained on a large dataset of natural scenes (the Places2 dataset; see section 2.5.3). An intermediate layer of the CNN is used as a feature extractor for training linear per class SVMs that predict the visual attributes of the SUN dataset. The results reported in [PXSH14], which are based on handcrafted features, could be outperformed by a margin. In a recent work by the author of this thesis and colleagues, it could be shown that the traditional per-attribute SVM could be replaced by a CNN that predicts multiple classes in a single forward pass [GSF16]. The accuracy can be significantly improved compared to the per class SVM.

Similar to these zero-shot methods, given a very small set of training samples, attributes can also be used in order to transfer class labels to unknown images [RSS11]. Attributes for images can either be learned directly via annotated training images or indirectly via additional sources of information such as Wikipedia or WordNet [LNH14, RSS11]. Examples of such knowledge transfer that combines natural language and vision are given in the following section.

5.1.2 *Combining Natural Language and Vision*

As annotating images with attributes is tedious, especially, the combination with natural language allows for scaling recognizers to a larger number of classes and attributes. In [RSS11] it is shown that attributes can also be mined from external sources instead of manually defining them. Attributes are derived for the object classes in the ImageNet dataset based on different sources. Text sources such as *Wikipedia*, *Yahoo Web*, *Yahoo Snippets* are considered as well as visual sources including tags such as *Yahoo Images*. The attributes are then related to the visual classes based on semantic measures such as the WordNet path length or web search hit counts [RSS$^+$10]. The association is then either performed on a class level or in a hierarchical approach based on the WordNet tree, which has been used for the creation of the ImageNet dataset. In [RSS11] the evaluation is performed in a supervised as well as a zero-shot setting. While it is not possible to outperform a visual one-vs-all recognition in the supervised case, good results are shown in the zero-shot setup using *direct attribute prediction* as in [LNH14]. It is however not possible to compare these attributes to a set of manually assigned ones as these are not available for the ImageNet classes.

Besides the attribute based zero shot recognition, a direct similarity measure of classes is also introduced [RSS11]. Instead of associating classes with attributes, a set of known classes is associated with each other based on semantic similarity. The approach is to some extent a simplification of the *indirect attribute prediction*, where a set of known classes is predicted in order to derive attributes. In [RSS11, RWT$^+$13] the probability distribution over a set of known classes is directly used in order to find similar classes. It is shown that this direct relation between classes may be beneficial instead of using visual attributes. This holds especially true, in a setting where all target classes are previously unobserved, whereas in mixed settings of known and unknown classes attributes may be more accurate [RWT$^+$13]. The application of the semantic relations can also be adapted to other tasks. In [RWT$^+$13] applications include associating objects and attributes for zero-shot recognition, visual action recognition or describing image and video content

In the following a similar approach will be presented where the semantic relation between visual scenes and objects is exploited. This is most related to the direct similarity introduced in [RSS+10, RSS11]. A set of known classes, in this case natural scenes, is predicted and then related to previously unobserved visual object classes based on their semantic relations.

5.1.3 *Recent Developments*

Recently, tasks that match images and text became of increasing interest in the Computer Vision community [AAL+15, DAZ+17, KJKFF17, HAR+17, WLHL18]. The most prominent tasks are image and video capturing, where given an image query the task is to retrieve a relevant textual description, and visual question answering, where either visual information is provided according to a textual query or a textual question regarding an image and it's content has to be answered.

For example, in [KJKFF17] relevant regions of an image are extracted based on a Region Proposal Network (RPN). These regions are then fed into a hierarchical Recurrent Neural Network (RNN) in order to create an image description. In [HAR+17], the opposite direction is considered. Given a textual question, the text is decoded using an RNN. Based on the textual decomposition, neural networks are chosen in order to analyze the image an then answer the visual question. For example, given the question *How many other things are of the same size as the green matte ball?*, the green ball has to be located by a visual recognizer and then objects of a similar size need to be determined. In [WLHL18] two branch networks are introduced for matching images and text. Given two inputs, a visual region and a description, both types of information are processed by independent branches of a Deep Neural Network (DNN). For training the network, both inputs are then combined in a loss layer. The same approach can then be used in order to compute a similarity between text and an image crop.

Furthermore, in [KZG+17], the so-called Visual Genome, a large dataset combining the visual and textual domain, has been introduced. For each image a complete scene graph of objects, attributes and their relations with each other is provided. This scene graph allows not only for predicting attributes and relations, but also for answering complex visual questions. Textual queries are parsed with respect to attributes so that for a given textual query the best matching images can be retrieved. An approach for an automatic generation of a scene graph is introduced in [XZCFF17]. A CNN and a RPN are used in order to recognize the scene and objects that are depicted within an image. Using graph inference, relations are derived from the predictions, yielding a complex scene graph.

The works on the Visual Genome are one of the few ones that consider the relations between scene and objects. In [XZCFF17] the goal is to derive a complex scene graph, describing the relations between a scene and the objects within this scene. Similarly, most tasks dealing with image captioning or visual question answering, consider the relation between entities within an image. In the proposed approach, relations will not be derived from an image. Instead it will be shown that semantic relations between scenes and objects can be leveraged for visual recognition.

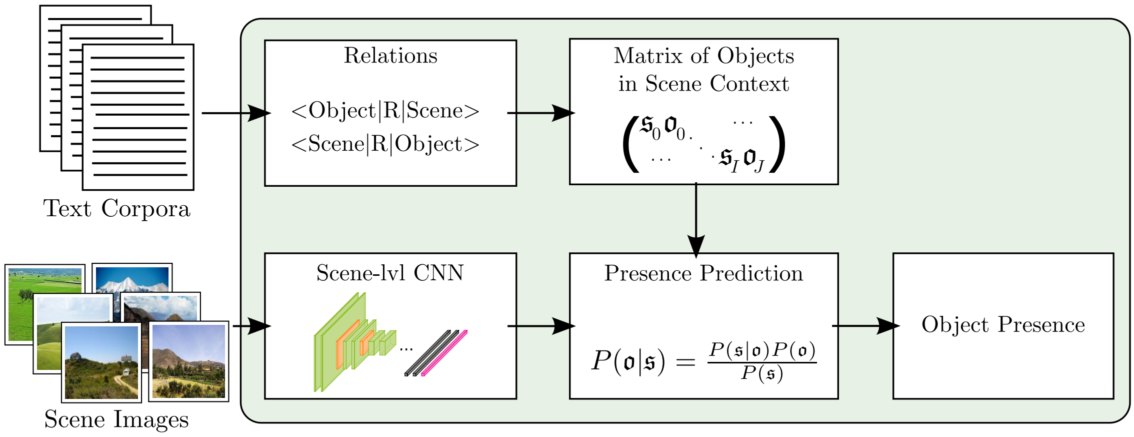

Text Corpora
Scene Images
Relations
<Object|R|Scene>
<Scene|R|Object>
Matrix of Objects in Scene Context
$\begin{pmatrix}\mathfrak{s}_0\mathfrak{o}_0 & \cdots \\ \cdots & \mathfrak{s}_I\mathfrak{o}_J\end{pmatrix}$
Scene-lvl CNN
...
Presence Prediction
$P(\mathfrak{o}|\mathfrak{s}) = \frac{P(\mathfrak{s}|\mathfrak{o})P(\mathfrak{o})}{P(\mathfrak{s})}$
Object Presence

Figure 5.3: Overview of the proposed zero-shot approach based on scene information. First, a matrix containing relation counts between scenes and objects is derived from text corpora. Then, for a given scene image a scene label is predicted. Using both information cues, a probability for the presence of object classes is derived.

5.2 SEMANTIC SCENE KNOWLEDGE FOR ZERO-SHOT OBJECT PREDICTION

In the proposed method for object prediction, relations between scenes and objects are derived from text sources and are then used for modeling top down knowledge. This knowledge is used in order to predict the presence of objects and replaces the visual information that is typically used for object prediction. An overview of the method is given in figure 5.3. On the textual side, large text corpora are analyzed with respect to possible scene and object categories. Using an information extraction approach, the relations between scenes and objects are extracted from the text source. From these relations, a matrix of objects in a scene context is created. The image is solely analyzed on scene level, which requires minimal annotation effort and no visual knowledge about the objects within the scene. Most importantly, no annotations are required with respect to the object classes. A statistical approach is used for predicting an object's presence based on the textual relations between scenes and objects. This is then used in combination with the scene prediction in order to predict the presence of an object in an image.

While most of the related work builds on a set of visual attributes which are recognized independently of each other, here, only the scene category is predicted. A probabilistic assignment to a set of scenes is a more simplistic approach and does not require the assignment of multiple visual attributes. The approach can, therefore, be interpreted as a direct semantic similarity. It is thus to some extend similar to the direct semantic similarity measures presented in [RSS+10, RSS11]. It extends these approaches by using semantic relations from text corpora to bridge the gap between visual scenes and objects.

5.2.1 *Obtaining Relations*

Given a large text corpus, information that is extracted from this corpus can be assumed to be roughly representative for the real world and thus may also be observed in natural scene images. Such text corpora can, for example, be obtained by crawling Wikipedia or any other source of textual information from the web [RSS11]. For the proposed approach, sentences including possible scene or object categories and their relations are of further interest. When assuming that the text corpus is representative for the real world, so are the relations between scenes and objects which are derived from the text.

In the following, relations based on the information extraction approach Reverb [EFC+11] are used. The corpus is split into single sentences. Reverb then extracts relations and their arguments from a given sentence based on two steps:

1. For each verb i in the sentence, the longest sequence of words is uncovered which starts at the verb i and satisfies both a syntactical and a lexical constraint. For the syntactical constraint the following regular expression is evaluated:

$$\mathfrak{v}|\mathfrak{v}\mathfrak{p}|\mathfrak{v}\mathfrak{n}^*\mathfrak{p}$$

$$\begin{aligned} \mathfrak{v} &= \text{verb particle? adv?} \\ \mathfrak{n} &= (\text{noun}|\text{adj}|\text{adv}|\text{pron}|\text{det}) \\ \mathfrak{p} &= (\text{prep}|\text{particle}|\text{inf. marker}) \end{aligned} \tag{5.1}$$

   Since some phrases may pass this constraint that will have only very few possible instances, the additional lexical constraint is applied. A dictionary consisting of 1.7 million relations from the English language is used in order to reject uncommon relations. Sequences which do not pass the syntactical or lexical constraint are discarded. Finally, overlapping sequences are merged into a single relation sequence.

2. Given a sequence which passed both, the syntactical and lexical constraint, its relation $\mathfrak{r}$, the nearest noun left and right of the relation $\mathfrak{r}$ are extracted. If two nouns can be observed for a relation $\mathfrak{r}$, this results in a triplet

$$\mathfrak{t} = (\text{noun1}, \mathfrak{r}, \text{noun2}) \ . \tag{5.2}$$

Simple examples relating scenes and objects may be *'A car drove down the street'* or *'Many persons were on the streets'*. Both relate an entity that is typically used in object detection (*car* or *person*) with a scene label (*street*). Based on the description above, the triplets

$$\begin{aligned} \mathfrak{t}_1 &= (\text{car}, \text{drive}, \text{street}) \\ \mathfrak{t}_2 &= (\text{person}, \text{is}, \text{street}) \end{aligned}$$

would be derived from the two sentences with the relations being *drive* and *is* in their stemmed presence form.

5.2.2 *Counting Relations*

Given a set of relations that were extracted from a text corpus and a vocabulary that defines a set of scene names and object names, a matrix $\boldsymbol{Q}$ describing the objects in scene context is created. In a general setting a vocabulary could be derived from frequently occurring words in a text corpus [EFC+11] or the WordNet tree [MO95]. For example, many Computer Vision corpora are based on the entities in the WordNet tree (cf. [RDS+15]). For a specific task, the vocabulary can also be based on a set of objects and scenes which are of interest and which are defined beforehand.

In the following $\mathfrak{S}$ and $\mathfrak{O}$ are referred to as the number of scene and object names in the vocabulary, so that $\boldsymbol{Q}$ has a size of $\mathfrak{S} \times \mathfrak{O}$. At the index $\mathfrak{s}, \mathfrak{o}$ it contains the number of relations between the respective scene $\mathfrak{s}$ and object $\mathfrak{o}$:

$$\begin{aligned} Q_{\mathfrak{s},\mathfrak{o}} &= \delta(\mathfrak{s},\mathfrak{o}) + \delta(\mathfrak{o},\mathfrak{s}) \quad \text{with} \\ \delta(i,j) &= |\{\mathfrak{t} = (i,\mathfrak{r},j) \mid \forall\, \mathfrak{r}\}| \quad . \end{aligned} \tag{5.3}$$

In the experiments, some rare cases of self-similarity were observed, in which a scene and an object name are the same (i.e. a scene in a street may also show the object road/street). In these cases the self-similarity is set to the maximum count observed.

Ultimately, this representation can be considered as a case of co-occurrence representation, which is frequently used in contextual modeling. Here, the co-occurrences are derived based on textual relations and without any knowledge about the true co-occurrence count, i.e., counts derived from a training dataset.

5.2.3 *Scene Classification*

For analyzing the image, a classifier has to be trained that is able to recognize one of the $\mathfrak{S}$ scene categories. Basically, any approach that allows for classifying a complete image can be applied for this task. It is desirable to have a probability distribution over the scene classes so that a probability $P(\mathfrak{s}|\mathcal{I})$ can be computed for a given image $\mathcal{I}$. In the following, a CNN that is based on the VGG16 network architecture is used for predicting the scene category [SZ14]. The network is pre-trained on a large image corpus, e.g., ImageNet or the Places2 dataset, and then adapted to the target task using a set of training images depicting the scene categories which are of further interest.

Note that this only requires annotating images with respect to the scene depicted in the image which requires only little manual effort. The annotation effort for a single scene label is much lower than labeling various attributes or the object classes occurring in an image (cf. [BRFFF16]). It is a descriptive abstraction which is similar to the zero-shot recognition based on a direct semantic relation (cf. [RSS11]) as it does not imply any visual knowledge about the objects within the scene.

The CNN can be fine-tuned in a supervised manner based on a previous initialization. Thus, the annotation effort with respect to the desired scene categories is relatively low.

In addition, the annotation effort can be reduced even further when training the scene classifier in a semi-supervised manner as it has been shown in chapter 4.

5.2.4 *Presence Prediction*

After extracting the relations between scenes and objects from a large text corpus and predicting the scene category, both sources of information can be combined in order to predict the presence of objects in a given image. Following the assumption that the text corpus is representative for the real world, it can further be assumed that the likelihood of an object to occur in a given scene is correlated with the number of textual relations between those entities. Thus the probability $P(\mathfrak{o}|\mathfrak{s})$ of an object $\mathfrak{o}$ to be present in a scene $\mathfrak{s}$ needs to be derived from the matrix $\boldsymbol{Q}$.

However, there is one limitation that needs to be considered when using the semantic relations. Not only the task of presence detection, but also the textual relations are concerned with the question whether at least one object instance can be observed in a given scene. Due to this many to one relation of objects that can occur in a single scene, it is not possible to recover an exact count of objects or scenes. Consider for example the sentence '*While many cars drove down the streets, there were even more motorbikes on these streets.*'. A relation would be observed for *car* and *street* and another relation would be observed for *motorbike* and *street*. Thus, if the Matrix $\boldsymbol{Q}$ contains N relation counts, these counts indicate the presence of at least N objects from O categories which may occur in a set of S scenes. However, the exact count of scenes which lead to the entries in the matrix cannot be recovered. Thus, a prior $P(\mathfrak{s})$ for a given scene category cannot be directly computed from the matrix. Since multiple objects can occur in a single scene, the probability $P(\mathfrak{o}|\mathfrak{s})$ of an object to occur in a scene can also not be directly estimated.

Most images are, however, constrained to shownd only one scene category so that given the observation of a set of object classes $P(\mathfrak{s}|\mathfrak{o})$ can be estimated from the counts, i.e. a *motorbike* and a *car* indicate a *street* scene. This allows to compute $P(\mathfrak{o}|\mathfrak{s})$ based on Bayes theorem as

$$P(\mathfrak{o}|\mathfrak{s}) = \frac{P(\mathfrak{s}|\mathfrak{o})P(\mathfrak{o})}{P(\mathfrak{s})} \qquad \text{with}$$

$$P(\mathfrak{s}|\mathfrak{o}) = \frac{Q_{\mathfrak{s},\mathfrak{o}}}{\sum_{\mathfrak{s}'} Q_{\mathfrak{s}',\mathfrak{o}}} \quad \text{and} \quad P(\mathfrak{o}) = \frac{\sum_{\mathfrak{s}'} Q_{\mathfrak{s}',\mathfrak{o}}}{\sum_{\mathfrak{s}'}\sum_{\mathfrak{o}'} Q_{\mathfrak{s}',\mathfrak{o}'}} \, . \tag{5.4}$$

The prior probability $P(\mathfrak{o})$ can be approximated by assuming that one relation count represents the presence of at least one object $\mathfrak{o}$. This approximation is sufficient since presence prediction tasks are also interested in the presence of one or more objects of a given class in an image. Since the prior probability $P(\mathfrak{s})$ for a certain scene cannot be derived from the matrix of objects in scene context, it is assumed to be uniformly distributed.

In order to be able to predict an object in a scene where no relations have been previously observed, unobserved events need to be handled. Therefore, the probability of an object $\mathfrak{o}$ to occur in a scene $\mathfrak{s}$ is smoothed by

$$P^*(\mathfrak{o}|\mathfrak{s}) = (1-w)\ P(\mathfrak{o}|\mathfrak{s}) + w\ P(\mathfrak{o}) \quad \text{with} \quad w = \frac{|\{Q_{\mathfrak{s},\mathfrak{o}}|Q_{\mathfrak{s},\mathfrak{o}} = 1\}|}{\sum_{\mathfrak{s}'}\sum_{\mathfrak{o}'} Q_{\mathfrak{s}',\mathfrak{o}'}}\ , \tag{5.5}$$

similar to the smoothing of probability distributions for statistical natural language processing (cf. [MS99]). The interpolation factor w is estimated based on the number of relations with only a single occurrence, as it is frequently done.

Assuming a perfect classification, all images depicting a scene $\mathfrak{s}$ would be given the same probability for an object to occur in the image. In order to model this more softly, $\hat{P}(\mathfrak{o}|\mathfrak{s})$ is sampled by drawing from a normal distribution so that

$$\hat{P}(\mathfrak{o}|\mathfrak{s}) \sim \mathcal{N}(P^*(\mathfrak{o}|\mathfrak{s}), \sigma(\boldsymbol{Q})) \quad . \tag{5.6}$$

Thus, the remaining degree of uncertainty that is associated with counts that are derived from a text source is modeled. In addition, the drawing from a normal distribution also allows for covering intra-scene variabilities. The variance σ is estimated based on the variance over all counts within the matrix $\boldsymbol{Q}$.

Finally, the scene classification and the probability for an object to occur in a scene category are combined in order to estimate the probability of an object $\mathfrak{o}$ to occur in a given image $\mathcal{I}$. The presence is then predicted by:

$$P(\mathfrak{o}|\mathcal{I}) = \sum_{s} \hat{P}(\mathfrak{o}|\mathfrak{s}) \cdot P(\mathfrak{s}|\mathcal{I})\ . \tag{5.7}$$

5.2.5 *Context for Object Detection*

The proposed approach for presence prediction can also be incorporated as a context descriptor for visual object detection. Similar to the approach of the context Support Vector Machine (SVM) in [FGMR10], the detection scores of an object detector can be re-scored based on this context descriptor. The approach can also be interpreted as a meta-recognition step (cf. [SRMB11]). In [FGMR10] a feature vector that consists of a bounding box location, a class score and the scores of other objects in an image is derived for each detection of a deformable part-based model (DPM). This feature vector is then used in order to compute a new score for each detection (see section 2.4).

Here, it is proposed to train a scene-context SVM. Let $\mathcal{B}$ be the detection window of an object detector, i.e., a Regional Convolutional Neural Network (R-CNN) [RHGS15]. For each detected bounding box a context descriptor is derived:

$$\boldsymbol{x} = (y_{\text{mean}}, y_{\text{std}}, P(o_1|I), .., P(o_n|I)) \tag{5.8}$$

Instead of using object co-occurrences as in [FGMR10], the object priors $P(o_i|I)$ from the semantic relations between scenes and objects are integrated for re-scoring the detections.

As CNNs are typically overconfident in their decision, the approach from [GSF16, GSF17] is integrated in the evaluation of the R-CNN's scores. Dropout is applied at test time, using a probability p_{drop} and the results from multiple forward passes is averaged, producing a mean y_{mean} and a standard deviation y_{std} for the bounding box $\mathcal{B}$. In the experiments p_{drop} is set to 0.1, following the results from [GSF16]. Thus, instead of using a single detection score, the mean and standard deviation from multiple forward passes are integrated in the context descriptor in order to model the uncertainty associated with the R-CNN's detections.

An SVM is trained independently for each object class. True positive and false positive detections are gathered by evaluating the detector on a training set. The false positives are collected by hard negative mining (cf. [FGMR10]) using only samples with an overlap below 30% of any positive bounding box. Platt's scaling is used in order to derive new probability scores [P+99].

5.3 EXPERIMENTS

In this section the proposed approach for predicting objects based on the relations between scenes and objects in a zero-shot manner is evaluated. Ideally, the evaluation requires a dataset that offers annotations for both scenes as well as object labels. Hence, the evaluation is based on different branches of the SUN corpus (described in section 2.5.3; [XEH+14]). Namely, the *SUN2012 Scene and Object Dataset in Pascal VOC format* and the *SUN2009 Context dataset* are used. These two branches of the SUN corpus are described in some more detail in the following:

> **SUN2012 Scene and Object Dataset in Pascal VOC format:** the dataset contains a subset of images which are taken from the SUN image corpus. While the more prominent *SUN397 dataset* is annotated with 397 different scenes labels, this branch of the SUN corpus contains additional annotations for 4 919 different object classes [XEH+14]. There are 16 873 images of which 11 426 are a subset of the *SUN397 dataset*. For the latter one, a scene label and object labels are available. For the remaining 5 447 images no scene annotations are provided.
>
> **SUN2009 Context:** this branch of the dataset also contains a subset 12 059 images and annotations for several object categories. In [CTW12] a benchmark for a supervised detection experiment has been defined which uses 107 object categories.

Both branches show a broad set of different scene and object categories. In contrast to traditional object detection tasks, like the Pascal VOC challenge [EEVG+15], there is a great variability with respect to the object's properties. While some of them are well defined (e.g. car or person), some others describe regions (sky, road, buildings) or highly deformable objects (river, curtain). Moreover, the annotations have been created with the LabelMe toolbox via crowd sourcing (cf. [BT12]) so that they are very noisy: There is no clear naming scheme, some of them contain descriptive attributes, like *person walking*, *table occluded*, *tennis court outdoor* and others mix singular and plural.

In order to relate the scene and object labels with natural text, all descriptive attributes are removed from the labels and all objects and scene labels are lemmatized based on the WordNet tree [MO95]. From the initial 4 919 object classes in the *SUN2012 Scene and Object Dataset*, 3 390 unique object labels which may occur in 377 different scene categories are remaining after this filtering. Most of the descriptive attributes are related to activities that are carried out by persons. Furthermore, several object labels are stemmed as the annotations mix singular and plural.

Note that although the lemmatized scene names may be semantically similar, they may be visually different. For example, *tennis court indoor* and *tennis court outdoor* are grouped into one scene category after removing the descriptive *indoor/outdoor* attributes. Therefore, the visual classifier that predicts the scene labels is trained on all 397 scene labels which are available for the SUN397 dataset. As a result, the probabilities for a lemmatized scene category are the sum of the original visual classes. The object presence is then predicted based on the relations between the lemmatized scene and object names.

After evaluating the zero-shot object prediction, a final evaluation is shown for a visual object detection task. The Pascal VOC2012 dataset is used for this task (see section 2.5.2). It is a common benchmark for object detection containing 20 different object classes. The well known R-CNN object detector is evaluated as a baseline [GDDM16]. The detected bounding boxes are re-scored based on the context descriptor and the SVM classifier described in section 5.2.5.

5.3.1 *Creating a Matrix of Objects in Scene Context*

In order to compute a matrix of objects in scene context, the OpenIE database has been queried. It contains over 5 billion extractions that have been obtained using Reverb on over a billion web pages[1]. Hence, a very diverse text corpus that captures the relations between a huge set of nouns has been used. Note that these web pages are in no way related to the visual recognition task. The vocabulary has been defined based on the task of the SUN2012 Scene and Object dataset. All lemmatized scene and object names are defined as labels which are of further interest with respect to the benchmark. Therefore, the vocabulary consists of the $\mathfrak{S} = 377$ scene names and $\mathfrak{O} = 3390$ object names. All possible combinations of scenes and objects were queried for which a total count of 1 375 559 relations have been extracted. The distribution of these relation counts is very long tailed, leaving a large set of unobserved events. In practice, these unobserved events can be considered as very unrealistic or rare co-occurrences of scenes and objects.

5.3.2 *SUN 2012 Scene and Object Dataset*

In order to recognize objects in a zero-shot manner based on their relation with a scene category, the probability $P(\mathfrak{s}|\mathcal{I})$ of a scene $\mathfrak{s}$ to be depicted in a given image $\mathcal{I}$ needs to be computed. Here, two different classification setups are evaluated:

Perfect classifier: it is assumed that the scene label is known beforehand, i.e., given by a human in the loop, or that training a perfect scene classifier with respect to the annotated scene labels would be possible. In order to simulate this case, $P(\mathfrak{s}|\mathcal{I})$ is set to one for the annotated scene label and set to zero otherwise.

CNN classification: a VGG16 network architecture is trained as described in section 5.2.3. The network is pre-trained on a large image corpus such as ImageNet or the Places dataset. The resulting weights are then used for the initialization of the subsequent training and the network is adapted to the task of scene classification in a supervised manner. All scene images from the SUN397 dataset that are not included in the SUN2012 Scene and Object dataset are used for training. The exclusion of the images from the SUN2012 Scene and Object dataset leaves a set of 97 304 training images. The training images have been augmented using random translations $(0 - 5\%)$, flipping (50% chance) and Gaussian noise $(\sigma = 0.02)$ in order to achieve a

[1] For a demo see http://openie.allenai.org/, last visited June 18th, 2017

better generalization. In total 15 epochs have been used for training, i.e., about 5 700 training iterations of batch size 256. The learning rate has been set to $\eta = 10^{-4}$ for the first ten epochs and is reduced to $\eta = 10^{-5}$ for the last five epochs.

Semi-supervised CNN classification: in addition to the VGG16 network that is trained in a supervised manner, an additional network is trained in a semi-supervised manner. Instead of using all 97 304 images for training only 11 910 samples ($\approx$ 12% and 30 labels per class) are labeled using iterative partition-based annotation learning (PBA), as introduced in chapter 4. The same augmentations as for the supervised case are used. Following the setup introduced in chapter 4, the network is trained for 40 epochs using the smaller subset of training samples that have been labeled in a semi-supervised manner. The learning rate has been set to $\eta = 10^{-4}$ for the first 20 epochs and is reduced to $\eta = 10^{-5}$.

As the ground truth labels in the SUN corpus are created by crowd sourcing (see section 2.5.3), they might be ambiguous. In [XHE+10] it has been shown that even human annotators deviate in their decision from the ground truth labels. Hence, a soft assignment to the set of scenes might yield a more meaningful input for the prediction. The oracle experiment using the ground truth annotations is therefore an interesting comparison on this benchmark.

5.3.2.1 *Scene Classification*

For the scene classification, the two initializations, based on a pre-training on ImageNet or the Places2 dataset, are compared (described in section 2.5.1). The recognition rates for the top predictions of the CNN on scene categories of the SUN2012 Scene and Object dataset are shown in table 5.1. All images of the SUN2012 Scene and Object dataset for which scene labels are available have been used for the evaluation. Similar to the ILSVRC benchmark, the top ranking predictions are evaluated and the classification is considered as correct if the ground truth label is in the top ranking predictions. The results show that pre-training on the more related Places2 dataset is beneficial. The highest scoring prediction yields an accuracy of 62.6%, whereas only 59.1% is achieved with the ImageNet pre-training. Considering that the correct result must be within the five highest scoring predictions an accuracy of 89.3% is achieved. The performance of the top five predictions emphasizes that the probabilistic assignment to a set of scene categories based on the CNN's predictions is a meaningful input for the proposed object prediction.

	recognition rate		
	# top predictions		
Initialization	1	3	5
ImageNet [RDS+15]	59.1%	79.3%	85.8%
Places2 [ZKL+16]	62.6%	83.3%	89.3%

Table 5.1: Recognition rate for the k highest scoring predictions of the scene label using a CNN.

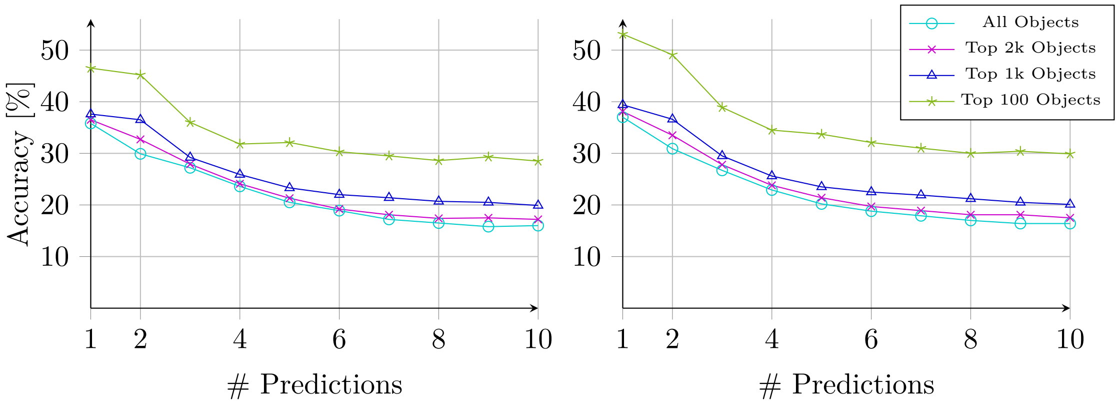


Figure 5.4: Accuracy for the top k object predictions on the SUN2012 Scene and Object Dataset. (left) The scene label is predicted by simulating a perfect classifier using the ground truth labelss. Thus, the object prediction is only based on the number of relations between the annotated scene and the set of objects. (right) The scene label is predicted by the CNN. Thus, each image is softly assigned to the scene categories.

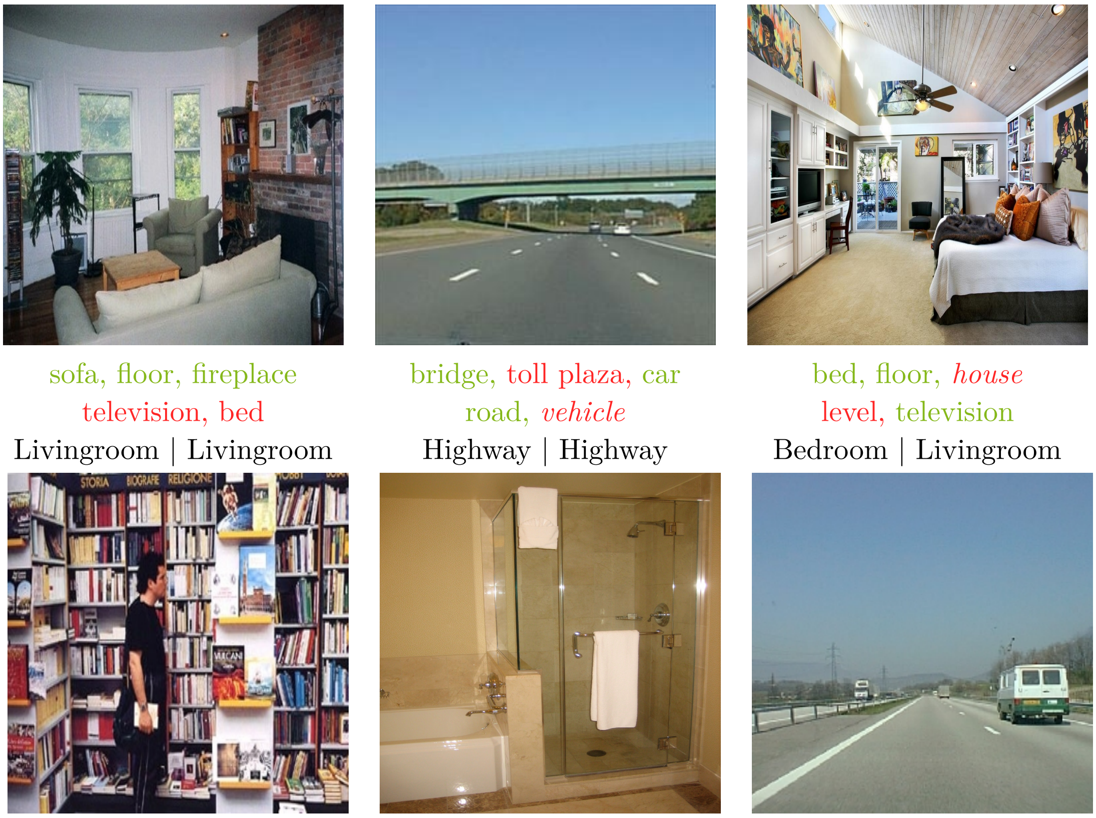
Based on this comparison of the two different pre-training datasets, the CNN that is pre-trained on the Places2 dataset is used in the following experiments for object prediction.

5.3.2.2 *Zero-shot Object Prediction*

For the task of object prediction, the accuracy for the most frequently occurring classes is evaluated. The top 100 up to all 3390 objects categories from the dataset are considered. The most frequently occurring object classes can safely be assumed to be present in most images, such as walls, while others are comparably rare. Since multiple objects can occur in a single image, the k highest scoring predictions are considered and their accuracy is evaluated. For example, for $k = 2$, both predictions are compared to the ground truth so that each one can be a correct or false prediction. Thus, in contrast to a top k single-class prediction, the accuracy will decrease while increasing the number of predictions k. Similar to the evaluation setup in [CTW12], only images with at least k annotated objects were used for the evaluation.

In figure 5.4 the results for the two different scene classifiers are shown: on the left side for simulating a perfect classifier based on the annotated scene labels and on the right side for predicting $P(\mathfrak{s}|\mathcal{I})$ using the CNN that has been pre-trained on the Places2 dataset. When evaluating the CNN, the highest ranking object predictions have a precision of up to 53.1% when considering a set of 100 objects and 37.0% when considering as many as 3 390 different object categories. These are promising results considering that no knowledge about the visual appearance of the objects is used. However, as mentioned before, some of the objects describing regional objects tend to be very general and can safely be assumed to occur in most scenes.

Interestingly, the simulation of a perfect scene classifier does not yield a superior accuracy compared to the results achieved by predicting the scene label using the CNN. This might be explained by the the findings reported in [XHE+10]. Scenes are often ambiguous, for



sofa, floor, fireplace
television, bed
Livingroom | Livingroom

bridge, toll plaza, car
road, *vehicle*
Highway | Highway

bed, floor, *house*
level, television
Bedroom | Livingroom

book, information, *people*
material, floor
Bookstore | Bookstore

shower, *bath*, floor
tub, bathtub
Bathroom | Bathroom

car, road, *vehicle*
people, parking
Highway | Highway

Figure 5.5: Exemplary results showing the five highest scoring object predictions: (green) correct (red) wrong (red & italic) wrong according to annotations, but can be seen in the image. In the bottom row: (left) Annotation (right) highest scoring CNN prediction.

example, a scene depicting a *cathedral*, a *church* or a *chapel* is hardly visually distinguishable. Thus, even humans do not achieve an agreement on the correct label. While these scenes are visually similar, they may also be similar on a semantic level. The CNN computes a soft probabilistic assignment to the scene classes which may in such cases be more robust than a single label. Furthermore, some information about the objects occurring in a scene may also be inferred from the CNN's prediction. For example, in [ZKL+15] it has been shown that CNNs predict scene classes based on dominant objects within the scene. Similarly, on the textual side there is also room for ambiguity. The distribution of objects in scene context is obtained from a very general external text source, which does not necessarily match the ground truth distribution that can be observed in the dataset. Thus, a mixture of relations from many scenes is more robust than the observed relations from a single scene class.

Exemplary results showing the five highest scoring object predictions for six image are shown in figure 5.5. Correct predictions are shown in green, incorrect predictions according

Objects	Perfect Classifier mAP [%]	Scene-level CNN mAP [%]	Chance mAP [%]	GT Distribution mAP [%]
Top 20	34.7	38.8	21.6	54.1
Top 40	29.5	33.2	13.9	47.1
Top 60	24.2	27.3	10.3	43.2
Top 80	21.5	24.2	8.5	39.2
Top 100	19.4	22.0	7.2	35.9

Table 5.2: Mean average precision for different sets of objects on the SUN2012 dataset in Pascal format. The presence predictions are based on the number of relations between the scenes and the objects: (1^{st} col.) simulation of a perfect classifier using the ground truth scene labels. (2^{nd} col.) scene labels predicted using the CNN. (3^{rd} col.) results of a prediction by chance – for comparison. (4^{th} col.) results of the proposed method when using the true number of scene-object relations derived from the dataset.

to the ground truth annotations are shown in red. On the bottom the annotated as well as the highest scoring scene label are given. Several of the incorrect predictions, which are not present in the image, are often at least plausible guesses of what else could be found in the given scene. For example, in the first image in the top row, a *television* is an object that occurs most frequently in a *livingroom*. Similarly, in the second image in the top row, the second best prediction is a *toll plaza* which is also frequently found on a highway. The third image is an interesting example where the highest scoring scene prediction of the CNN is incorrect. However, the prediction of *livingroom* instead of *bedroom* is not only visually but also semantically, related, as the wall on the left with the TV could also easily be placed in a livingroom. Here, the effect of the probabilistic assignment can also be observed. In the extracted relation matrix there is no relation between scene *livingroom* and the object *bed*. Nevertheless, the object class *bed* is still the highest scoring object prediction due to the influence of the other scene labels from the probabilistic assignment.

Besides these first qualitative observations, it can also be seen that the annotations are noisy and not at all complete. Although a large set of objects is annotated in the SUN corpus, some of the predictions that cannot be found in the ground truth annotations might be deemed as correct. These annotations are marked in red and italic. For example, the *livingroom* scene can be considered as showing a *house* and both *highway* scenes are clearly showing a *vehicle* which is, however, absent in the annotations as it has been annotated with a different label. Similarly, the example of the bathroom in the bottom row shows a typical example of ambiguity in natural language as well as in the provided annotations. Since the three labels *bath*, *tub* and *bathtub* are in the annotations, they are regarded as classes of interest and predicted based on the matrix of textual relations. Nevertheless, only one of these object labels is assigned to this image.

In order to provide a more detailed analysis, different sets of object categories are also evaluated based on the VOC mean average precision (mAP) criterion (see section 2.5.1; [EEVG+15]). The results for the 20 to 100 most common objects in the dataset is given in table 5.2. For comparison, the prediction by chance as well as an oracle experiment that is based on the ground truth distribution of the object classes is shown. The mAP

# top predictions	recognition rate [%]
1	47.9
3	69.5
5	77.0

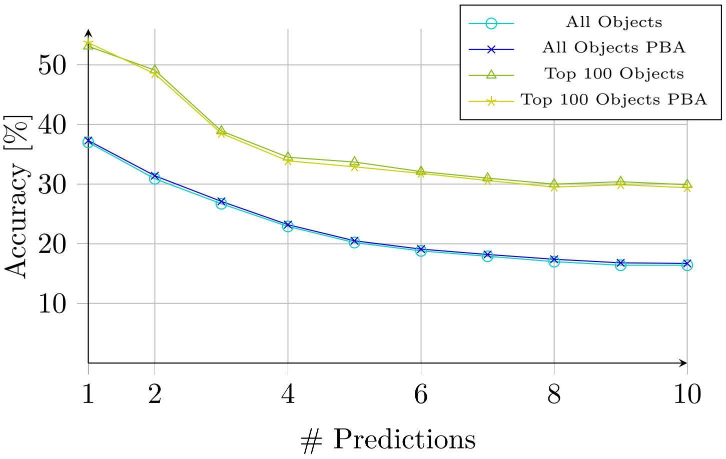


Table 5.3: Accuracy for the top k predictions on the SUN2012 Scene and Object Dataset using a CNN that is fine-tuned in a semi-supervised manner. (left) Prediction of the scene label (right) Prediction of object labels.

of predicting an object by chance is indicating how frequently these objects occur in the dataset. For the oracle experiment, the ground truth scene label and the distribution of all scenes and objects which can be observed in the dataset are used. The object class prediction is still estimated by using the proposed statistical approach. This can be seen as an indication of an upper bound for the performance that could be obtained by solely using the proposed approach of scene and object relations.

Two observations can be made from these evaluations. First, similarly to the previous evaluation, the probabilistic assignment to scene labels is more robust. Second, the oracle experiment reveals that the relations obtained from the additional text sources do not perfectly model the distribution in the dataset. Nevertheless, the results are well between the indicated upper and lower bounds. It can therefore be argued that the proposed method shows promising results. Especially given the fact that the relations are obtained from arbitrary websites. There is potential for improvement if the relation can be estimated in a more accurate manner, for example, by limiting the textual sources to ones that are related to the visual task.

5.3.2.3 *Zero-shot Object Prediction with Semi-supervised Scene Learning*

In order to reduce the manual effort even more, the CNN has been fine-tuned in a semi-supervised manner using PBA. The results for the highest scoring scene labels as well as the derived object classes are shown in table 5.3. The result shows some interesting findings. As the CNN is trained with only about 12% of the sample set being labeled, the accuracy is not as good as for the CNNs that have been trained in a supervised manner. There is a difference of about 14.7% for the top 1 accuracy and 12.3% for the top five accuracy. It is worth noting that this is still a good result considering the remarkably smaller number of labels that have been used for training the network. In total, 65565 images have been labeled by PBA with 64.4% accuracy. When predicting object classes within a scene image,

Method	Annotations	# Objects	mAP [%]
Part based Models [FGMR10]	Cropped objects	107	17.9
PbM + Context SVM [CTW12]	Cropped objects	107	23.8
PbM + Tree Context [CTW12]	Cropped objects	107	26.1
Scene-level CNN + Objects in Context	- (*)	107	19.4
PBA + Scene-level CNN + Objects in Context	- (*)	107	17.3
Scene-level CNN + Objects in Context	- (*)	104	20.1
PBA + Scene-level CNN + Objects in Context	- (*)	104	18.0

Table 5.4: Mean average precision for the object presence in the SUN2009 Context dataset. Entries are marked with (*), where the CNN predicts one of the 397 scenes from the SUN397 dataset without any knowledge about the objects.

the results are very promising as the difference between the two networks is neglectable. For example, the highest scoring prediction achieves an accuracy of 53.7% when considering the 100 most frequent object classes and 37.3% when considering all object classes within the dataset. The highest scoring predictions are present in the images with a similar accuracy. It can, therefore, be assumed that labels which are incorrectly assigned are not only visually but also semantically similar.

5.3.3 *SUN2009 Context Dataset*

The SUN2009 Context dataset is evaluated in order to emphasize the difficulty of detecting objects with a large variability, e.g., large regional or highly deformable objects. This subset of the SUN dataset contains 200 challenging object categories. In [CTW12] 107 of these have been used for evaluating object detectors that are based on DPMs. The remaining 93 categories could not be detected. The DPMs have been trained in a fully supervised manner including additional images from web sources. In [CTW12], a plain DPM detector has been compared to two other methods which combine a DPM detector with a re-scoring that is based on additional context models that have been learned from the data. Thus, the proposed approach can be directly compared to the supervised object detection methods.

5.3.3.1 *Zero-shot Object Prediction*

For the proposed object prediction approach, the same CNN as before is used. Thus, a CNN that is pre-trained on the Places2 dataset and then adapted to the SUN corpus is used for predicting the scene categories. The object classes are again predicted based on the proposed approach evaluating the relations between scenes and objects and thus without any visual information. As the original evaluation protocol contained three classes which were filtered by the stemming (*bottles, stones* and *rocks*), these classes cannot be recognized by the proposed method. The results for all 107 classes as well as the results for the remaining 104 classes after the stemming are shown in table 5.4.

The proposed approach, which is solely based on scene-level predictions, achieves comparable results to DPMs that are trained in a fully-supervised manner. Only with additional contextual information the DPMs are able to outperform the proposed approach. This clearly shows the requirement for contextual information, especially since the visual information in cluttered scenes is often limited, as shown in the SUN dataset. Although there is no evaluation of R-CNNs on this task, they have surpassed DPMs as the state-of-the-art in object detection (see section 2.3.3; [GDDM16]). It can therefore be assumed that they outperform the DPMs on this task as well. Nevertheless, the DPMs are powerful object detectors and it is interesting that a widely unsupervised approach can achieve similar results which shows that the relations between scenes and objects provide important cues for object prediction.

5.3.3.2 *Zero-shot Object Prediction with Semi-supervised Scene Learning*

Similar to the SUN 2012 Scene and Object dataset, the CNN that has been fine-tuned in a semi-supervised manner using PBA has been evaluated for the object recognition task on the SUN2009 Context dataset. The results are also shown in table 5.4. The network that is trained in a semi-supervised manner yields a mAP of 17.3% and 18.0% for the 107 and 104 object classes respectively. With a difference of 2.1% in the mAP compared to the network that has been trained in a supervised manner, the difference is more clear than for the top predictions on the SUN 2012 dataset. Nevertheless, considering that only a fraction of the scene labels have been used for training the network on the target task, the difference between the supervised and semi-supervised approach is not very large.

With a mAP of 17.3% the result is still in a similar range as the results of the DPMs. This is a remarkable result considering that only very little visual information from the target domain has been used for training the CNN and no visual information about the object classes at all.

5.3.4 *Pascal VOC2012 Dataset*

After evaluating the zero-shot object prediction, a final evaluation is shown for a visual object detection task. Here, the Pascal VOC2012 dataset is used (see section 2.5.2). It is a common benchmark for object detection containing 20 different classes. The well known R-CNN is evaluated as a baseline detector using the publicly available model [GDDM16]. Hence, the result is purely achieved at test time without further training of the visual object detector.

The detected bounding boxes are re-scored based on the context descriptor and a per class SVM as described in section 5.2.5. For deriving the object priors, the CNN is trained on the Places2 dataset and fine-tuned on the semantic scene categories from the SUN dataset in a supervised manner. The CNN is evaluated on the Pascal VOC dataset. The object classes closest to the 20 Pascal VOC classes are evaluated using the matrix of objects in scene context (described in section 5.3.1). Thus, the probability $P(\mathfrak{o}|\mathcal{I})$ is computed for all 20 classes of the Pascal VOC dataset and incorporated in the context descriptor together with the R-CNN's scores. A sampling of ten forward passes and a dropout of $p_{\text{drop}} = 0.1$ is used in order to derive a mean and variance for the R-CNN's scores. The training set of the VOC2012 dataset is purely used for training the scene-context SVMs.

class	R-CNN [GDDM16]	R-CNN + context-SVM [FGMR10]	R-CNN + scene-context
Aeroplane	80.0	**80.5**	79.8
Bicycle	73.7	**75.3**	74.8
Bird	68.9	69.0	**70.0**
Boat	46.4	44.7	**47.6**
Bottle	45.0	46.9	**47.4**
Bus	**79.1**	79.0	**79.1**
Car	68.2	68.9	**69.0**
Cat	83.1	82.0	**83.8**
Chair	39.7	40.0	**41.1**
Cow	62.5	58.7	**64.6**
Diningtable	50.8	**52.3**	**52.3**
Dog	80.9	81.5	**82.1**
Horse	70.8	70.0	**71.3**
Motorbike	**77.5**	76.7	**77.5**
Person	76.2	76.4	77.3
Pottedplant	31.9	32.7	**35.1**
Sheep	66.5	66.0	**67.7**
Sofa	56.3	52.6	**58.4**
Train	73.5	**75.1**	73.5
TV monitor	62.6	62.6	**62.9**
mAP	64.7	64.6	**65.8**

Table 5.5: Object detection results as average precision (AP) [%] on the Pascal VOC2012 dataset. The left column indicates the respective class, the center column the plain evaluation of the R-CNN model, the right column the evaluation of the re-scored detection results using the scene-context SVM.

The SVMs are trained using a grid search for parameter optimization and using an RBF kernel.

The evaluation results on the VOC2012 validation set are shown in table 5.5. The results show that the scene context is beneficial as context information in a visual object detection task. The integration of additional scene-level information is able to improve the results purely by re-scoring the detections at test time. The mAP is improved by 1.1% at test time. For comparison, the results of the co-occurrence re-scoring, as introduced in [FGMR10], are also shown. Again a grid search has been performed for the per class SVMs. However, the approach is not able to improve the results, which emphasizes the difficulty of re-scoring detections, especially since CNNs are typically very confident in their decisions.

5.4 CONCLUSION

In this chapter a novel approach for predicting the presence of objects based on their semantic relation to scenes has been introduced. The approach works in a zero-shot manner and directly relates visual scenes with object classes. While the scenes are visually observed beforehand, no visual information about the object classes is required. The application of a semantic relation between scenes and objects is closely related to the work in [RSS11, RWT+13]. In these works it has been shown that if no information of the target classes is known beforehand, a direct semantic relation is favorable over visual attributes as an intermediate layer.

In the proposed approach, a CNN is trained on a set of visual scene categories. At the same time the semantic relations between these scenes categories and a large set of object classes are extracted from a large textual corpus. Based on the information extraction approach presented in [EFC+11], arbitrary textual sources can be analyzed. For a given test image, a probability distribution over the scene classes is predicted. The visual and the textual information is then combined using a statistical approach.

In the experiments, information gathered from a billion arbitrary websites has been analyzed, yielding more than a 1.3 billion semantic relations between a complex set of scenes and objects. It has been shown that the relations between scenes and objects are a powerful source of information. It is possible to predict the occurrences for as many as 3390 objects without visually observing them beforehand. The scene predictions made by the CNN yield an accurate distribution over the scene classes which in turn allows for predicting the object classes within a given image. On the SUN2012 dataset, the highest ranking prediction achieved an accuracy of more than 50%. Considering some qualitative results it has been revealed that the crowd sourced annotations of the SUN dataset are relatively noisy, exhibiting several missing objects or ambiguous naming schemes. The predictions of the proposed approach often include objects which are simply omitted in the annotations. Several predictions which are incorrect with respect to the annotations are at least plausible guesses at a semantic level. Furthermore, an oracle experiment showed that textual sources which are more focused on the task at hand could improve the performance.

On the complex task of the SUN2009 context dataset, it has been shown that the object prediction shows a similar performance compared to visual object detectors that were trained in a fully supervised manner. Especially on tasks which contain cluttered scenes and not very well structured objects, the visual classification is very difficult.

When training the scene recognition CNN in a semi-supervised manner using PBA, further interesting observations can be made. While the recognition for the scene classification task is considerably lower than for the network that is trained in a supervised manner, the object classes on the SUN 2012 dataset are predicted without a significant difference in the accuracy. Similar results are obtained on the SUN2009 context dataset. While there is a difference in the mAP between the supervised and the semi-supervised network training, there is no tremendous difference between the two. This allows to predict object classes while using almost no visual information from the target domain and, therefore, only a very limited annotation effort.

This allows for several applications of the proposed approach. For example, one could think of a scenario with a human in the loop, where the highest scoring object classes are suggested to a human annotator for verification. This could reduce the manual effort for tagging images as a verification requires only a small amount of time (cf. [BRFFF16]).

In a final experiment it has been shown that the object predictions can also be incorporated as additional information for an object detector. Similar to the context SVM introduced in [FGMR10], the predictions of an R-CNN are re-evaluated in a meta-recognition step. Here, it could be shown that the detection accuracy of an R-CNN can be improved purely at test time by integrating contextual information.

I've gotten three data points in the last 60 days
- Unknown grad student, PhD Comic 1648, 25.10.2013

6 CONCLUSION

This thesis introduced methods that lower the manual annotation effort for visual scene and object recognition. As the work of this thesis has been done between 2012 and 2018, it deals with traditional pattern recognition approaches as well as methods from the field of deep learning. The contributions made in this thesis are three-fold and range from feature augmentation over semi-supervised learning for natural scene classification to zero-shot object recognition.

Feature Augmentation

First, a novel method for incorporating additional information at feature level and subsequently integrating this information in a Bag-of-Features (BoF) representation has been introduced. The additional information can, for example, be of spatial or temporal nature, encoding a local feature's position within it's descriptor. The information is quantized and appended to the feature vector and thus integrated in the unsupervised learning step of the BoF representation. As a result more specific codebook entries are computed for different regions. It has been shown that the proposed approach allows for learning compact feature representations without reducing the accuracy of the subsequent classification. It's applicability has been shown in the field of image classification for natural scenes and objects as well as the field of acoustic event detection.

Semi-supervised Scene Learning

Second, a method for semi-supervised learning in large image collections has been proposed. The approach is based on multi-view learning and the active learning principle. The data is represented by multiple views which are clustered independently of each other. A human in the loop is asked to label each data cluster. The clusters are then iteratively refined based on cluster evaluation measures and additional labels are assigned to the dataset. Ultimately, a voting over all views creates a partially labeled sample set. This partially labeled sample set is then used for training an image classifier. The method has been applied to learning labels for natural scene images. For the semi-supervised learning it is desirable to have compact feature representations like the ones obtained by the proposed feature augmentation approach. Besides the application of traditional feature representations it has been demonstrated that the semi-supervised learning approach can also be applied to features derived from a deep network, i.e. a Convolutional Neural Network (CNN). Especially the application in the field of deep learning, which usually requires large amounts of labeled samples for training or even adapting the networks, showed that a powerful classifier can be learned with a minimal annotation effort.

Zero-shot Object Prediction

Third, a zero-shot object prediction has been proposed that combines visual and semantic information about natural scenes. A CNN is trained in order to distinguish different scene categories. Furthermore, the relations between scene categories and visual object classes are learned based on their semantic relation in large text corpora. The probability for a given image to show a certain scene is derived from the CNN and combined with the semantic relations based on a statistical approach. Thus, allowing to predict the presence of certain object classes in an image. It could be shown that especially in cluttered scenes the semantic information can be a powerful cue and that objects can be predicted in a zero-shot manner. It has furthermore be shown that the object predictions can be integrated in a visual object detector.

Summary

When combining these contributions, information about natural scene images and the object classes within these images can be gained without having the burden to manually label tremendous amounts of images beforehand. It could be shown that a scene classifier can be learned with minimal annotation effort and then subsequently be used in conjunction with semantic knowledge in order to predict the presence of object classes. In the experiments, the manual annotation effort for the scene classifier could be reduced by a factor of approximately ten. Thus, fine-tuning a deep neural network with only about 30 samples per class and still obtaining a high accuracy for scene classification. In conjunction with the semantic information, the presence of object classes could be predicted with up to 53.7% accuracy on a challenging dataset of natural scenes images without having any visual information about any of the object classes. The results of the network that is trained in a semi-supervised manner is thus similar to the supervised training.

The findings in this thesis illustrate the usefulness of the proposed approaches for many practical applications. When creating a visual recognition system for a novel task, one of the main burdens, the collection and annotation of data, can be reduced by the proposed methods.

7 BIBLIOGRAPHY

[AHK01] C. C. Aggarwal, A. Hinneburg, and D. A. Keim. On the surprising behavior of distance metrics in high dimensional space. In *Proc. International Conference on Database Theory*. Springer, 2001.

[ADF10] B. Alexe, T. Deselaers, and V. Ferrari. What is an object? In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2010.

[AAL$^+$15] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh. Vqa: Visual question answering. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[ADP07] J.-J. Aucouturier, B. Defreville, and F. Pachet. The bag-of-frames approach to audio pattern recognition: A sufficient model for urban soundscapes but not for polyphonic music. *The Journal of the Acoustical Society of America*, 122(2), 2007.

[AXE] AXES Project. Axes - acess to audiovisual archives, project page.

[AVT16] Y. Aytar, C. Vondrick, and A. Torralba. Soundnet: Learning sound representations from unlabeled video. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, 2016.

[BFP09] H. Badino, U. Franke, and D. Pfeiffer. The stixel world - a compact medium level representation of the 3d-world. In *Pattern Recognition: 31st DAGM Symposium Jena*. Springer, 2009.

[BHC15] V. Badrinarayanan, A. Handa, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labeling. *arXiv preprint arXiv:1505.07293*, 2015.

[BYRNO11] R. Baeza-Yates, B. Ribeiro-Neto, and Others. *Modern information retrieval*. Addison Wesley, 2 edition, 2011.

[BT12] A. Barriuso and A. Torralba. Notes on image annotation. *arXiv preprint arXiv:1210.3448*, 2012.

[BTG06] H. Bay, T. Tuytelaars, and L. V. Gool. Surf: Speeded up robust features. In *Proc. European Conference on Computer Vision (ECCV)*, 2006.

[BRFFF16] A. Bearman, O. Russakovsky, V. Ferrari, and L. Fei-Fei. What's the point: Semantic segmentation with point supervision. In *Proc. European Conference on Computer Vision (ECCV)*. 2016.

[BZM06] A. Bosch, A. Zisserman, and X. Munoz. Scene classification via pLSA. In *Proc. European Conference on Computer Vision (ECCV)*, 2006.

[BPL10] Y.-L. Boureau, J. Ponce, and Y. LeCun. A theoretical analysis of feature pooling in visual recognition. In *Proc. International Conference on Machine Learning (ICML)*, 2010.

[Cad17] C. Cadell. Google's AlphaGo clinches series win over Chinese Go master, 2017. http://www.reuters.com/article/us-science-intelligence-go-idUSKBN18L0LH.

[CLH08] L. Cao, J. Luo, and T. S. Huang. Annotating photo collections by label propagation according to multiple similarity cues. In *Proc. ACM International Conference on Multimedia*. ACM, 2008.

[CLVZ11] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. In *Proc. British Machine Vision Conference (BMVC)*, 2011.

[CSVZ14] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *Proc. British Machine Vision Conference (BMVC)*, 2014.

[CTW12] M. J. Choi, A. Torralba, and A. S. Willsky. A tree-based context model for object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(2), 2012.

[COR$^+$16] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. *arXiv preprint arXiv:1604.01685*, 2016.

[CDF$^+$04] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *Proc. European Conference on Computer Vision (ECCV)*, 2004.

[DVG13] D. Dai and L. Van Gool. Ensemble projection for semi-supervised image classification. In *Proc. IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2013.

[DVG16] D. Dai and L. Van Gool. Unsupervised high-level feature learning by ensemble projection for semi-supervised image classification and image clustering. *arXiv preprint arXiv:1602.00955*, 2016.

[DT05] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2005.

[DAZ+17] A. Das, H. Agrawal, L. Zitnick, D. Parikh, and D. Batra. Human attention in visual question answering: Do humans and deep networks look at the same regions? *Computer Vision and Image Understanding*, 163, 2017.

[DB79] D. L. Davies and D. W. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(2), 1979.

[DM80] S. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4), 1980.

[DDF+90] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6), 1990.

[DDS+09] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2009.

[DAF12] T. Deselaers, B. Alexe, and V. Ferrari. Weakly supervised localization and learning with generic knowledge. *International Journal of Computer Vision*, 100(3), 2012.

[DHH+09] S. K. Divvala, D. Hoiem, J. H. Hays, A. A. Efros, and M. Hebert. An empirical study of context in object detection. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[DAG+15] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-Term Recurrent Convolutional Networks for Visual Recognition and Description. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[DJS+09] M. Douze, H. Jégou, H. Sandhawalia, L. Amsaleg, and C. Schmid. Evaluation of gist descriptors for web-scale image search. In *Proc. ACM International Conference on Image and Video Retrieval*. ACM, 2009.

[DHS01] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern classification*. Pattern Classification and Scene Analysis: Pattern Classification. Wiley, 2001.

[Dun73] J. C. Dunn. A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. *Journal of Cybernetics*, 3(3), 1973.

[EFS12] S. Ebert, M. Fritz, and B. Schiele. Ralf - reinforced active learning formulation for object class recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[EFS13] S. Ebert, M. Fritz, and B. Schiele. Semi-supervised learning on a budget: scaling up to large datasets. In *Proc. Asian Conference on Computer Vision (ACCV).* Springer, 2013.

[EFC$^+$11] O. Etzioni, A. Fader, J. Christensen, S. Soderland, and M. Mausam. Open information extraction: The second generation. In *Proc. International Joint Conference on Artificial Intelligence (IJCAI)*, volume 11, 2011.

[EEVG$^+$15] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1), 2015.

[FCNL13] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8), 2013.

[FFFP04] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops.* IEEE, 2004.

[FFP05] L. Fei-Fei and P. Perona. A Bayesian hierarchical model for learning natural scene categories. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* IEEE, 2005.

[FGMR10] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9), 2010.

[FH04] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2), 2004.

[FWT09] R. Fergus, Y. Weiss, and A. Torralba. Semi-supervised learning in gigantic image collections. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, 2009.

[GG16] Y. Gal and Z. Ghahramani. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *Proc. International Conference on Machine Learning (ICML)*, 2016.

[GSB$^+$13] D. Giannoulis, D. Stowell, E. Benetos, M. Rossignol, and M. Lagrange. A Database and Challenge for Acoustic Scene Classification and Event Detection. In *Proc. European Signal Processing Conference (Eusipco)*, 2013.

[GDDM14] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[GDDM16] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Region-based convolutional networks for accurate object detection and segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(1), 2016.

[GB10] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proc. International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 9, 2010.

[GBB11] X. Glorot, A. Bordes, and Y. Bengio. Deep sparse rectifier neural networks. In *Proc. International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2011.

[Goo00] P. Good. *Permutation Tests - A Practical Guide to Resampling Methods for Testing Hypothesis*. Springer, 2 edition, 2000.

[GD05] K. Grauman and T. Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, volume 2, 2005.

[GHP07] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, 2007.

[Grz12] R. Grzeszick. Semi-supervised learning of visual object categories. Master's thesis, TU Dortmund University, 2012.

[GRF13] R. Grzeszick, L. Rothacker, and G. A. Fink. Bag-of-Features Representations using Spatial Visual Vocabularies for Object Classification. In *Proc. International Conference on Image Processing (ICIP)*, 2013.

[GF15] R. Grzeszick and G. A. Fink. An Iterative Partitioning-based Method for Semi-supervised Annotation Learning in Image Collections. *International Journal of Pattern Recognition and Artificial Intelligence*, 2015.

[GPF15] R. Grzeszick, A. Plinge, and G. A. Fink. Temporal Acoustic Words for Online Acoustic Event Detection. In *Proc. German Conference on Pattern Recognition (GCPR)*, Aachen, Germany, 2015.

[GSF16] R. Grzeszick, S. Sudholt, and G. A. Fink. Optimistic and pessimistic neural networks for scene and object recognition. *arXiv preprint arXiv:1609.07982*, 2016.

[GPF17] R. Grzeszick, A. Plinge, and G. A. Fink. Bag-of-Features Methods for Acoustic Event Detection and Classification. *IEEE/ACM Transactions Audio, Speech and Language Processing*, 25(6), 2017.

[GSF17] R. Grzeszick, S. Sudholt, and G. A. Fink. Optimistic and Pessimistic Neural Networks for Object Recognition. In *Proc. IEEE International Conference on Image Processing*, 2017.

[GF17] R. Grzeszick and G. A. Fink. Zero-shot Object Prediction Using Semantic Scene Knowledge. In *Proc. International Conference on Computer Vision Theory and Applications (Visapp)*, 2017.

[GSF18] R. Grzeszick, S. Sudholt, and G. A. Fink. Weakly supervised object detection with pointwise mutual information. *arXiv preprint arXiv:1801.08747*, 2018.

[GHY08] J. Gui, D.-S. Huang, and Z. You. An improvement on learning with local and global consistency. In *Proc. International Conference on Pattern Recognition (ICPR)*. IEEE, 2008.

[HMC17] P. Haeusser, A. Mordvintsev, and D. Cremers. Learning by association-a versatile semi-supervised training method for neural networks. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017.

[HZRS15a] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[HZRS15b] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proc. IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2015.

[HZRS15c] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9), 2015.

[HEH08] D. Hoiem, A. A. Efros, and M. Hebert. Putting objects in perspective. *International Journal of Computer Vision*, 80(1), 2008.

[HAR+17] R. Hu, J. Andreas, M. Rohrbach, T. Darrell, and K. Saenko. Learning to reason: End-to-end module networks for visual question answering. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2017.

[JDSP10] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2010.

[JSD+14] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional Architecture for Fast Feature Embedding. In *Proc. ACM International Conference on Multimedia*, 2014.

[KL16] A. Kolesnikov and C. H. Lampert. Improving weakly-supervised object localization by micro-annotation. In *Proc. British Machine Vision Conference (BMVC)*, 2016.

[KVJ11] J. Krapac, J. Verbeek, and F. Jurie. Modeling spatial layout with fisher vectors for image categorization. In *Proc. IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2011.

[KJKFF17] J. Krause, J. Johnson, R. Krishna, and L. Fei-Fei. A hierarchical approach for generating descriptive image paragraphs. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017.

[KZG$^+$17] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L. Li, D. A. Shamma, M. Bernstein, and L. Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1), 2017.

[KSH12] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, 2012.

[LNE08] J.-F. Lalonde, S. G. Narasimhan, and A. A. Efros. What does the sky tell us about the camera? In *Proc. European Conference on Computer Vision (ECCV)*. Springer, 2008.

[LNH14] C. H. Lampert, H. Nickisch, and S. Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(3), 2014.

[LSP06] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[LR09] S. Lazebnik and M. Raginsky. Supervised learning of quantizer codebooks by information loss minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(7), 2009.

[LBD$^+$89] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4), 1989.

[LBH15] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553), 2015.

[LMB$^+$14] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *Proc. European Conference on Computer Vision (ECCV)*. Springer, 2014.

[LYT09] C. Liu, J. Yuen, and A. Torralba. Nonparametric scene parsing: Label transfer via dense scene alignment. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2009.

[LC09] W. Liu and S.-F. Chang. Robust multi-class transductive learning with graphs. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2009.

[LAE$^+$16] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *Proc. European Conference on Computer Vision (ECCV).* Springer, 2016.

[Llo57] S. Lloyd. Least squares quantization in PCM's. Technical report, Bell Telephone Labs, Murray Hill, NJ, 1957.

[Llo82] S. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2), 1982.

[LSD15] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[Low99] D. G. Lowe. Object recognition from local scale-invariant features. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, volume 2, 1999.

[Low04] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2), 2004.

[Mac67] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proc. of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, 1967.

[MS99] C. D. Manning and H. Schütze. *Foundations of statistical natural language processing.* MIT Press, 1999.

[MS05] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10), 2005.

[MO95] G. A. Miller and Others. WordNet: a lexical database for English. *Communications of the ACM*, 38(11), 1995.

[MP69] M. Minsky and S. Papert. *Perceptrons.* MIT press, 1969.

[Mos18] S. Moss. Nvidia updates geforce eula to prohibit data center use, 2018. http://www.datacenterdynamics.com/content-tracks/servers-storage/nvidia-updates-geforce-eula-to-prohibit-data-center-use/99525.fullarticle.

[MCL$^+$14] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille. The role of context for object detection and semantic segmentation in the wild. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[NN02] S. G. Narasimhan and S. K. Nayar. Vision and the atmosphere. *International Journal of Computer Vision*, 48(3), 2002.

[Naß16] F. Naße. *Die Detektion interessanter Objekte unter Verwendung eines objektbasierten Aufmerksamkeitsmodells.* PhD thesis, Dissertation, Dortmund, TU Dortmund University, 2016.

[Nie15] M. A. Nielsen. *Neural Networks and Deep Learning.* Determination books, 2015. http://neuralnetworksanddeeplearning.com; no print available.

[NHH15] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[Nvi15] Nvidia. GPU-Based Deep Learning Inference: A Performance and Power Analysis. Technical report, Nvidia, 2015.

[OD11] S. O'Hara and B. A. Draper. Introduction to the bag of features paradigm for image classification and retrieval. *arXiv preprint arXiv:1101.3354*, 2011.

[OG10] M. Ojala and G. C. Garriga. Permutation Tests for Studying Classifier Performance. *Journal of Machine Learning Research*, 11, 2010.

[OPH96] T. Ojala, M. Pietikäinen, and D. Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern recognition*, 29(1), 1996.

[OPM00] T. Ojala, M. Pietikäinen, and T. Mäenpää. Gray scale and rotation invariant texture classification with local binary patterns. In *Proc. European Conference on Computer Vision (ECCV).* Springer, 2000.

[OT01] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3), 2001.

[OT06] A. Oliva and A. Torralba. Building the gist of a scene: The role of global image features in recognition. *Progress in Brain Research*, 155, 2006.

[PA12] S. Pancoast and M. Akbacak. Bag-of-Audio-Words Approach for Multimedia Event Classification. In *Proc. Interspeech*, 2012.

[PVZ15] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *Proc. British Machine Vision Conference (BMVC)*, 2015.

[Par16] R. Parloff. Why deep learning is suddenly changing your life, 2016. http://fortune.com/ai-artificial-intelligence-deep-machine-learning/.

[PHL04] L. Parsons, E. Haque, and H. Liu. Subspace clustering for high dimensional data: a review. *ACM SIGKDD Explorations Newsletter*, 6(1), 2004.

[PXSH14] G. Patterson, C. Xu, H. Su, and J. Hays. The SUN attribute database: Beyond categories for deeper scene understanding. *International Journal of Computer Vision*, 108(1-2), 2014.

[Per10] P. Perona. Vision of a Visipedia. *Proceedings of the IEEE*, 98(8), 2010.

[PSM10] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *Proc. European Conference on Computer Vision (ECCV)*. Springer, 2010.

[PCI$^+$07] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[PMRR00] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss. The FERET evaluation methodology for face-recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10), 2000.

[P$^+$99] J. Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3), 1999.

[PGF14] A. Plinge, R. Grzeszick, and G. A. Fink. A Bag-of-Features Approach to Acoustic Event Detection. In *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014.

[PBE$^+$06] J. Ponce, T. L. Berg, M. Everingham, D. A. Forsyth, M. Hebert, S. Lazebnik, M. Marszalek, C. Schmid, B. C. Russell, A. Torralba, and Others. Dataset issues in object recognition. In *Toward category-level object recognition*. Springer, 2006.

[RASC14] A. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. CNN features off-the-shelf: an astounding baseline for recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2014.

[RHGS15] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, 2015.

[RVF12] J. Richarz, S. Vajda, and G. A. Fink. Annotating handwritten characters with minimal human involvement in a semi-supervised learning strategy. In *Proc. International Conference on Frontiers in Handwriting Recognition*, Bari, Italy, 2012.

[RVGF14] J. Richarz, S. Vajda, R. Grzeszick, and G. A. Fink. Semi-Supervised Learning for Character Recognition in Historical Archive Documents. *Pattern Recognition, Special Issue on Handwriting Recognition*, 47(3), 2014.

[RSS$^+$10] M. Rohrbach, M. Stark, G. Szarvas, I. Gurevych, and B. Schiele. What helps where – and why? Semantic relatedness for knowledge transfer. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2010.

[RSS11] M. Rohrbach, M. Stark, and B. Schiele. Evaluating knowledge transfer and zero-shot learning in a large-scale setting. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* IEEE, 2011.

[RWT$^+$13] M. Rohrbach, Q. Wei, I. Titov, S. Thater, M. Pinkal, and B. Schiele. Translating Video Content to Natural Language Descriptions. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2013.

[Ros58] F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6), 1958.

[Rou87] P. J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20, 1987.

[RGF17] F. M. Rueda, R. Grzeszick, and G. A. Fink. Neuron Pruning for Compressing Deep Networks using Maxout Architectures. In *Proc. German Conference on Pattern Recognition (GCPR)*, 2017.

[RDS$^+$15] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3), 2015.

[SPD12] J. Sánchez, F. Perronnin, and T. De Campos. Modeling the spatial layout of images beyond spatial pyramids. *Pattern Recognition Letters*, 33(16), 2012.

[SPMV13] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek. Image classification with the fisher vector: Theory and practice. *International Journal of Computer Vision*, 105(3), 2013.

[SS13] B. Schauerte and R. Stiefelhagen. How the distribution of salient objects in images influences salient object detection. In *Proc. International Conference on Image Processing (ICIP)*, 2013.

[SRMB11] W. J. Scheirer, A. Rocha, R. J. Micheals, and T. E. Boult. Meta-recognition: The theory and practice of recognition score analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8), 2011.

[SdSB13] W. J. Scheirer, A. de Rezende Rocha, A. Sapkota, and T. E. Boult. Toward open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7), 2013.

[SCS$^+$13] J. Schröder, B. Cauchi, M. R. Schädler, N. Moritz, K. Adiloglu, J. Anemüller, S. Doclo, B. Kollmeier, and S. Goetze. Acoustic event detection using signal enhancement and spectro-temporal feature extraction. Technical report, IEEE AASP Challenge: Detection and Classification of Acoustic Scenes and Events, 2013.

[SSW07] Y. Shao, S. Srinivasan, and D. Wang. Incorporating auditory feature uncertainties in robust speaker identification. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2007.

[She62] R. N. Shepard. The analysis of proximities: Multidimensional scaling with an unknown distance function. *Psychometrika*, 27(2), 1962.

[SWRC06] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *Proc. European Conference on Computer Vision (ECCV).* Springer, 2006.

[SZ14] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[SZ03] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2003.

[Smi13] S. Smith. *Digital signal processing: a practical guide for engineers and scientists.* Newnes, 2013.

[SDBR15] J. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. Striving for simplicity: The all convolutional net. In *Proc. International Conference on Learning Representations (ICLR), workshop track*, 2015.

[SHK$^+$14] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1), 2014.

[SF15] S. Sudholt and G. A. Fink. A modified isomap approach to manifold learning in word spotting. In *Proc. German Conference on Pattern Recognition (GCPR)*, Lecture Notes in Computer Science, 2015.

[Sze10] R. Szeliski. *Computer Vision: Algorithms and Applications.* Springer Science & Business Media, 2010.

[TBZT07] F. Tang, S. Brennan, Q. Zhao, and H. Tao. Co-tracking using semi-supervised support vector machines. In *Proc. IEEE International Conference on Computer Vision (ICCV).* IEEE, 2007.

[TCHJH12] H. Tang, S. M. Chu, M. Hasegawa-Johnson, and T. S. Huang. Partially supervised speaker clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(5), 2012.

[TMZ$^+$07] A. Temko, R. Malkin, C. Zieger, D. Macho, C. Nadeu, and M. Omologo. {CLEAR} Evaluation of Acoustic Event Detection and Classification Systems. In *Multimodal Technologies for Perception of Humans*, volume 4122 of *Lecture Notes in Computer Science.* Springer Berlin Heidelberg, 2007.

[TL10] J. Tighe and S. Lazebnik. Superparsing: Scalable Nonparametric Image Parsing with Superpixels. In *Proc. European Conference on Computer Vision (ECCV)*. Springer, 2010.

[TL13] J. Tighe and S. Lazebnik. Finding things: Image parsing with regions and per-exemplar detectors. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2013.

[Tor52] W. S. Torgerson. Multidimensional scaling: I. theory and method. *Psychometrika*, 17(4), 1952.

[TFF08] A. Torralba, R. Fergus, and W. T. Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11), 2008.

[VJF11] S. Vajda, A. Junaidi, and G. A. Fink. A Semi-Supervised Ensemble Learning Approach for Character Labeling with Minimal Human Effort. In *Proc. International Conference on Document Analysis and Recognition (ICDAR)*, Beijing, China, 2011.

[vdODZ$^+$16] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu. Wavenet: A generative model for raw audio. In *Proc. ISCA Speech Synthesis Workshop*, 2016.

[VZ02] M. Varma and A. Zisserman. Classifying images of materials: Achieving viewpoint and illumination independence. In *Proc. European Conference on Computer Vision (ECCV)*. Springer, 2002.

[VF10] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. In *Proc. International Conference on Multimedia*. ACM, 2010.

[VGVZ09] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In *Proc. IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2009.

[VF15] A. Vezhnevets and V. Ferrari. Object localization in ImageNet by looking out of the window. *arXiv preprint arXiv:1501.01181*, 2015.

[VMH$^+$17] T. Virtanen, A. Mesaros, T. Heittola, A. Diment, E. Vincent, E. Benetos, and B. M. Elizalde. *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017)*. Tampere University of Technology. Laboratory of Signal Processing, 2017.

[VBK$^+$13] L. Vuegen, B. V. D. Broeck, P. Karsmakers, J. F. Gemmeke, B. Vanrumste, and H. V. Hamme. An MFCC-GMM Approach for Event Detection and Classification. Technical report, IEEE AASP Challenge: Detection and Classification of Acoustic Scenes and Events, 2013.

[WLHL18] L. Wang, Y. Li, J. Huang, and S. Lazebnik. Learning two-branch neural networks for image-text matching tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[WAC$^+$04] J. Willamowski, D. Arregui, G. Csurka, C. R. Dance, and L. Fan. Categorizing nine visual classes using local appearance descriptors. *Illumination*, 17, 2004.

[WSvdH$^+$16] Q. Wu, C. Shen, A. van den Hengel, P. Wang, and A. Dick. Image Captioning and Visual Question Answering Based on Attributes and Their Related External Knowledge. *arXiv preprint arXiv:1603.02814*, 2016.

[WJW$^+$17] S. Wu, Q. Ji, S. Wang, H.-S. Wong, Z. Yu, and Y. Xu. Semi-supervised image classification with self-paced cross-task networks. *IEEE Transactions on Multimedia*, 2017.

[XHE$^+$10] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. SUN database: Large-scale scene recognition from abbey to zoo. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2010.

[XEH$^+$14] J. Xiao, K. A. Ehinger, J. Hays, A. Torralba, and A. Oliva. SUN Database: Exploring a Large Collection of Scene Categories. *International Journal of Computer Vision*, 2014.

[XZCFF17] D. Xu, Y. Zhu, C. B. Choy, and L. Fei-Fei. Scene graph generation by iterative message passing. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017.

[ZF14] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *Proc. European Conference on Computer Vision (ECCV)*. Springer, 2014.

[ZKL$^+$15] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Object detectors emerge in deep scene CNNs. In *Proc. International Conference on Learning Representations (ICLR)*, 2015.

[ZKL$^+$16] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016.

[ZLX$^+$14] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, 2014.

[ZKL$^+$16] B. Zhou, A. Khosla, A. Lapedriza, A. Torralba, and A. Oliva. Places: An image database for deep scene understanding. *arXiv preprint arXiv:1610.02055*, 2016.

[ZBL+04] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, 2004.

[Zhu06] X. Zhu. Semi-supervised learning literature survey. *Technical report, Computer Science, University of Wisconsin-Madison*, 2006.

[ZZRFF15] Y. Zhu, C. Zhang, C. Ré, and L. Fei-Fei. Building a Large-scale Multimodal Knowledge Base System for Answering Visual Queries. *arXiv preprint arXiv:1507.05670*, 2015.