KTH Computer Science
and Communication

Data Driven Visual Recognition

OMID AGHAZADEH

Doctoral Thesis
Stockholm, Sweden, 2014

TRITA-CSC-A 2014:10
ISSN-1653-5723
ISRN-KTH/CSC/A-14/10-SE
ISBN 978-91-7595-197-3

Computer Vision and Active Perception laboratory
School of Computer Science and Communication
KTH Royal Institute of Technology
SE-100 44 Stockholm, Sweden

Tryck: Eprint AB 2014

Abstract

This thesis is mostly about supervised visual recognition problems. Based on a general definition of categories, the contents are divided into two parts: one which models categories, and one which is not category based. We are interested in data driven solutions for both kinds of problems.

In the category-free part, we study novelty detection in temporal and spatial domains as a category-free recognition problem. Using data driven models, we demonstrate that based on a few reference exemplars, our methods are able to detect novelties in ego-motions of people, and changes in the static environments surrounding them.

In the category level part, we study object recognition. We consider both object category classification and localization, and propose scalable data driven approaches for both problems. A mixture of parametric classifiers, initialized with a sophisticated clustering of the training data, is demonstrated to adapt to the data better than various baselines such as the same model initialized with less subtly designed procedures. A non-parametric large margin classifier is introduced and demonstrated to have a multitude of advantages in comparison to its competitors: better training and testing time costs, the ability to make use of indefinite/invariant and deformable similarity measures, and adaptive complexity are the main features of the proposed model.

We also propose a rather realistic model of recognition problems, which quantifies the interplay between representations, classifiers, and recognition performances. Based on data-describing measures which are aggregates of pairwise similarities of the training data, our model characterizes and describes the distributions of training exemplars. The measures are shown to capture many aspects of the difficulty of categorization problems and correlate significantly to the observed recognition performances. Utilizing these measures, the model predicts the performance of particular classifiers on distributions similar to the training data. These predictions, when compared to the test performance of the classifiers on the test sets, are reasonably accurate.

We discuss various aspects of visual recognition problems: what is the interplay between representations and classification tasks, how can different models better adapt to the training data, *etc.* We describe and analyze the aforementioned methods that are designed to tackle different visual recognition problems, but share one common characteristic: being data driven.

Keywords: Visual Recognition, Data Driven, Supervised Learning, Mixture Models, Non-Parametric Models, Category Recognition, Novelty Detection.

Acknowledgments

First and foremost, I'd like to thank my parents and particularly my partner Sara, without the support of whom I would not have been able to start and finish my PhD studies.

I'd like to thank Jan-Olof Eklundh for introducing computer vision to me and getting me interested in the field by inspiring discussions around various related topics; Josephine Sullivan for introducing statistical learning and machine learning to me during my Master's studies; and Stefan Carlsson for introducing projective geometry to me during the same time.

I'd like to thank Stefan for supporting my interest in various topics that I touched upon during my doctoral studies. I would not have had the same understanding of what computer vision is, and what it should be about, without Stefan caring about the big-picture aspects of the research we have done together. I'd also like to thank Josephine for supporting me in the beginning of my PhD studies, and for caring about the details of the research we did together.

I'd also like to thank all my colleagues in CVAP. Patric Jensfelt played an important role in creating the Systems, Control and Robotics Master's programme, during which I got to know and get interested in the great research environment in CVAP. Danica Kragic helped maintain the inspiring atmosphere in CVAP. As a result, I have had many constructive discussions with my close colleagues and friends there. Figure 1 demonstrates my recollection of such discussions. Thank you for this (sorted by the degree of colleagues' nodes in the graph): Hossein, Stefan, Josephine, Heydar, Oscar, Miro, Vahid, Jan-Olof, Carl-Henrik, Sobhan, Babak, Magnus, Ali, Peter, Mårten, and Niklas. I have also enjoyed the company and support of various other colleagues including Christian, Hedvig, Florian, Friné, Alper, Marin, Oskar, Marianna, Andrzej, Xavi, Virgile, Alessandro, Jeanette, Renaud, Gert, Matthew, Martin, Cheng, Mikael, Püren, Atsuto, Lazaros, and many others. I'd also like to thank Sara, Axonn, Oscar, Jan-Olof, Hossein, and Hedvig for helping me proofread and/or improve this thesis.

I have compiled a list of authors I have cited the most in the 5 papers included in this work. According to that list, the work presented here has been mostly inspired by (in alphabetical order): Pedro Felzenszwalb, Jitendra Malik, Deva Ramanan, Andrea Vedaldi, and Andrew Zisserman.

Last but not least, I'd like to acknowledge funders who supported my PhD studies. Most of the works in this thesis were funded by Swedish Foundation for Strategic Research (SSF); within the project VINST (Wearable Visual Information Systems), and by the European Commission KIC: EIT ICT labs.

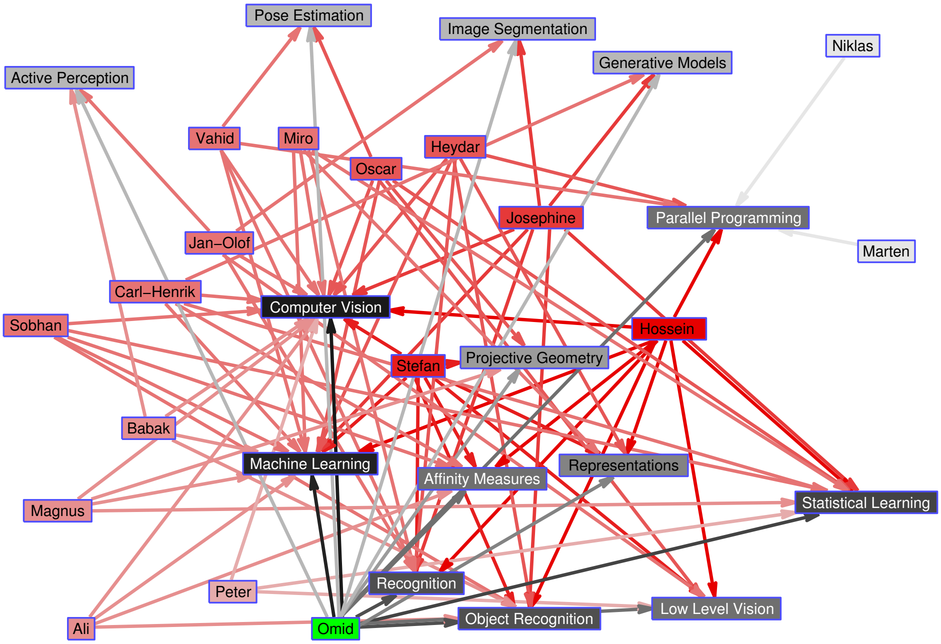


Figure 1: Visualization (of a sparsified version) of the research aspect of my Academic social network within CVAP. An edge in between a colleague's node and a topic represents my recollection of more than one discussions related to the topic between me and the corresponding colleague. According to the figure, I have had most discussions around *Computer Vision*, *Machine Learning*, *Statistical Learning*, *Recognition*, and *Object Recognition*; mostly with Hossein, Stefan, Josephine, Heydar, and Oscar.

List of Papers

This thesis is based on the following papers:

[A] Omid Aghazadeh, Josephine Sullivan and Stefan Carlsson. Novelty Detection from an Ego-Centric Perspective. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[B] Omid Aghazadeh, Josephine Sullivan and Stefan Carlsson. Multi View Registration for Novelty/Background Separation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[C] Omid Aghazadeh, Hossein Azizpour, Josephine Sullivan and Stefan Carlsson. Mixture Component Identification and Learning for Visual Recognition. In *European Conference on Computer Vision*, 2012.

[D] Omid Aghazadeh and Stefan Carlsson. Properties of Training Data Predict the Performance of Classifiers. In submission to: *International Journal of Computer Vision*, 2014.

[E] Omid Aghazadeh and Stefan Carlsson. Large Scale, Large Margin Classification using Indefinite Similarity Measures. In submission to: *Neural Information Processing Systems*, 2014.

In addition to the papers [A]-[E], the following papers have been produced in part by the author of this thesis:

- Omid Aghazadeh and Stefan Carlsson. Properties of Datasets Predict the Performance of Classifiers. In *British Machine Vision Conference*, 2013.
- Ali Sharif Razavian, Omid Aghazadeh, Josephine Sullivan and Stefan Carlsson, Estimating Attention in Exhibitions Using Wearable Cameras. In *International Conference on Pattern Recognition*, 2014.

Contents

Part I

Introduction

Chapter 1

Introduction to Visual Recognition

Constructing algorithms which can interpret and acquire understanding of the contents of images and videos, has been the holly grail of the computer vision field. Such algorithms, tackling visual recognition, have been identified as a core component of AI systems which are to interact with people. This interaction might come in different forms such as providing information regarding contents of images, searching large collection of images for instances of object categories, and learning individual preferences. Although humans perform recognition tasks rather trivially and almost without any conscious effort, thanks to a billion years of evolution of multi-cellular life forms, developing algorithms which match human vision has proven to be extremely complex.

Visual recognition is not only a complex and interesting academic problem. With the advances in computer hardware, and also in visual recognition algorithms, we have reached a point where industry is taking serious steps towards including visual recognition in their next generation AI systems. The reason that visual recognition has gained much attention from both academia and industry is the ever increasing amounts of data and the ever improving power to process them. What might make construction of robust visual recognition algorithms possible is the ever growing production of better digital cameras – available in various forms such as dSLRs, mobile, and wearable cameras – giving rise to huge collections of images and video footage, and the ability to have data driven models learn different concepts from such data collections. The consequence is that acquiring and processing extremely large bodies of images and video footage have become an extremely fruitful research direction, with the potential to transform the AI technology in the 21st century.

Pursuing the dream of many computer vision scientists and experts, this thesis is dedicated to visual recognition. That is, we seek computational approaches for recognition of visual patterns. We define what we exactly mean by *visual recognition*, and discuss how different *representations* and *classifiers* affect it in Section 1.1. We discuss and define what we mean by *categories* in Section 1.2, and introduce the two types of problems we consider in this work: *category-free* and *category level* recognition. For both the category-free and category level recognition problems, we consider *data driven* approaches. After discussing what parametric and non-parametric classifiers are in Section 1.3, we elaborate on what is meant by *data driven*, and how it differs from non-parametric classification, in Section 1.4.

To summarize, what we investigate in this thesis is:

Learning, from training data, to classify visual patterns; given (a large amount of) prior knowledge.

The prior knowledge plays a significant role in visual recognition, and it covers very general aspects and assumptions about *e.g.* the natural world, the imaging process, and the distribution of exemplars sharing particular characteristics under specific representations.

This chapter introduces the background for the problems studied in this work. Category-free recognition is more thoroughly discussed in Chapter 2. Chapter 3 covers category level recognition. Summaries of the included papers are given in Chapter 4. Chapter 5 concludes the thesis.

1.1 Representation, Classification and Recognition

Throughout this thesis, we adopt a definition of recognition which differs from what is defined as recognition in cognitive sciences, biological vision, and also in the mainstream computer vision. In computer vision, recognition usually refers to the study of problems such as object recognition, action recognition, and scene recognition; and problems such as learning to detect boundaries of objects in natural images is not considered to be doing recognition. Such categorization of problems is suitable for fields such as biological (human) vision, where different cells and neural circuits that affect human vision and perception play different roles in transforming the signals captured in retina to a semantic understanding of the natural world. In computer vision, such distinct circuitries for extracting signals from still images or videos do not exist. Furthermore, there is no reason for computer vision to try to adopt a systems approach similar to how signals are transformed in human brain, as 1) computational algorithms are not constrained (or even guided) by evolution, and 2) we still do not have a clear understanding of human vision in a way that allows construction of algorithms that mimic it in a computationally feasible and efficient manner.

As a result of this difference between computational and biological systems, we argue that unless the goal is to exactly mimic the human vision, there is no reason to try to define computer vision tasks in the same way that they are carried out in the human brain *i.e.* based on the same inputs, outputs, and in the same way the input is transformed to the output. A high level computational task might be implemented in various ways, and one might pick solutions which have lower computational complexities, higher accuracies, or have other specific desired properties

[1]. As a result of this freedom in implementation of high level visual tasks, the question is then the following: what can be considered a high level visual task? We consider any visual task which is derived by (computational) learning a high level visual task. This is in contrast with the current mainstream definition based on biological counterparts of the tasks being (significantly) constrained by evolution. In other words, we adopt a computationally motivated definition for recognition, as opposed to a biologically inspired one.

The Oxford English dictionary defines *recognition* as:

> *"Identification of a thing or person from previous encounters or knowledge."*

In computer vision, we are mostly interested in visual patterns as the 'thing' to recognize. In [10], pattern recognition is described as:

> *"The field of pattern recognition is concerned with the automatic discovery of regularities in data through the use of computer algorithms and with the use of these regularities to take actions such as classifying the data into different categories."*

Pattern recognition includes problems such as regression, classification, and structure prediction. In this thesis, the focus is on the learning aspect of classification of visual patterns.

Is visual recognition the same as pattern recognition techniques applied to visual patterns? Our short answer is no; with the following motivation. Unlike in traditional pattern recognition and machine learning, where the representation of data is usually fixed and given, the RGB pixel values of images do not constitute a good representation suitable for pattern classification. For example, images acquired by pinhole cameras, represent a projection of the natural world. This projection can perturb properties that exist in the natural world *e.g.* perpendicular lines are not necessarily perpendicular when projected, or give rise to new ones in images *e.g.* occlusions. However, many properties are preserved such as continuity and linearity. What makes visual recognition particularly hard, is the lack of a good representation of visual patterns which highlights the important information and discards what is irrelevant. The consequence is that the models that are used to classify visual patterns cannot rely on these properties to be reflected in the representation. As a result, the models and/or the representations need to be adapted particularly for visual recognition. [2]

[1] This has been pointed out by Marr [29] in the context of his computational theories.

[2] It is crucial to note that for any representation, a measure of (dis) similarity needs to be specified. For example, the χ^2 distance is a better dissimilarity measure for histogram representations, in comparison to the Euclidean distance. Similarly, a Gaussian RBF kernel is more appropriate than a linear kernel, as a measure of similarity for representations which result in linearly non-separable data distributions. Therefore, when we refer to representations, we assume that they are accompanied by suitable (dis) similarity measures. In this regard, the choices regarding suitability of (dis) similarity measures are as equally important as the algorithms for deriving and extracting the representations from images, videos, *etc.* For the sake of brevity, in this thesis we

Although there have been works on how to derive suitable representations, there has not been any consensus even on what constitutes an ideal representation. There are many ways to acquire representations from images, each highlighting specific aspects and suppressing other sources of variance in images; see Marr's argument on the implementation of computational theories [29]. The most common approaches, arguably inspired by Marr's representational scheme [29], follow this pipeline: construct some local features based on some properties of images, and acquire a global representation based on some sort of aggregation of these local features. The list below gives some examples of such an approach:

- HOG [13] where HOG cells are particularly aggregated and normalized local image gradients, and the global representation is acquired by globally concatenating the HOG cells on a 2D spatial grid. This representation mostly captures shape, and is invariant mostly to illumination changes and very small local deformations.

- Bag of Words [39] where local regions are somehow identified *e.g.* dense sampling on a grid, or interest regions [30], and somehow described *e.g.* via SIFT [28], or learned descriptors [37]; and a representation is acquired by somehow aggregating these local descriptors *e.g.* in a spatially ignorant manner [39], or on hierarchies of 2D spatial grids [25]. Depending on the choice of descriptors, this approach can result in a representation invariant to local and global transformations of the input image. For example a plain bag of densely sampled SIFT descriptors will be invariant to global translations and rotations, a plain bag of affine covariant interest regions described with SIFT is invariant to global affine transformations, and the same approaches aggregated on 2D spatial grids are only invariant to local deformations.

- Hyper features [1] which are similar to the bag of words approach, but the feature extraction, coding and representation is repeated multiple times in a spatially aware manner.

- Attribute based mid-level representations [17] which describe images, or regions of interest, by pooling responses of some (semantically meaningful) classifiers.

For a rather long period of time, image representation usually referred to an approach similar to one of the aforementioned ones, potentially based on different local features, regions, coding techniques, pooling techniques, *etc.* The representation was defined, derived, and fixed, prior to learning a classifier. However, the rather recent deep learning approach defines representation in an alternative way.

The deep convolutional representation learning, mostly based on Yann Lecun's convolutional networks [45], gained significant attention from the computer vision

use 'representation' to refer to both *i.e.* we also consider approaches such as metric learning to be (implicitly or explicitly) related to representations.

community with the success of [23] in ImageNet [14]. In [23], consecutive convolutions with locally (shared) convolutional kernels, followed by particular normalizations and max-poolings constitute local features, and concatenation of convolution responses constitutes the (intermediate) representation. This step is repeated multiple times, and the invariance to local deformations is aggregated towards larger and more global invariance to deformations/transformations with the addition of consecutive convolutional layers. The convolutional response of the last convolutional layer, or the response of the next (fully connected) layer, is said to constitute a representation suitable for recognition.

The convolutional deep learning approach is similar to Marr's representational approach in that there is a sequence of increasingly complex representations that achieve the desired generalization and invariance. However, unlike Marr's approach and other traditional representational schemes, only a description of what will constitute a representation is specified in the method, and the parameters of the convolutional kernels are learnt jointly with the classifier coupled with a categorization task [3]. This is in contrast to the other representational approaches that are usually decoupled from particular tasks they might be needed for.

The question is, if we need to consider representation and classification two different components required for recognition, or if they are inter-connected and non-separable? In other words, is the representational part of the deep learning approach the RGB pixel values, and in that case the convolutional layers will be a part of the classifier, or is the representation the convolutional part of the network, and thereby the fully connected multi layer network at the end of the deep learning architecture constitutes the classifier? Similarly, are the particular choices for traditional representations (HOG, BoW, *etc.*) classifier choices, and if they are, the representation will be the RGB pixel values, or are the representations separate from the choice of classifiers? Answering these questions is outside the scope of this thesis. However, we provide a framework in Section D which can be used to select representational schemes which are more suitable for particular datasets and recognition tasks.

While there exist works which suggest that the representation acquired from deep learning might be suitable for various kinds of recognition tasks [36, 32], the deep learning framework was not as popular and accessible when these studies were performed. Therefore, in this thesis we do not investigate the deep learning representation. Instead, in the category-free part, various representations are considered and evaluated for different recognition tasks. Similarly, in the category level part, the representation is fixed to a slightly modified version of HOG described in [18].

[3] There are works which use regressors in the last layer [41]. There is no limit on what can be connected to the network, as long as its (sub) gradients can be computed and used in the back-propagation.

1.2 Category Free and Category Level Recognition

The definition of categories is a complicated and a rather sensitive topic, specially because in cognitive sciences, what comprises a category is directly translated to our understanding of the reasoning capabilities of humans, and our understandings of the world surrounding us [24]. While there are more complex definitions of categories in various fields such as cognitive science and psychology, we are only interested in a simplified definition of categories which can be used in a computational framework. Consequently, we take on a rather traditional definition of categories; one which Lakoff relates to 'the objectivist view' [24].

Empirical results of our methods (discussed later in Chapter 3 and in Section C and Section D) qualitatively suggest that exemplars of what we consider as categories are not equal, that is, we implicitly make use of graded category memberships [24]. The inequality of exemplar-category memberships might have various reasons such as 1) the biases (photographer and selection bias [40]) involved in the sampling process of exemplars, or 2) the difference in the centrality of the exemplars to the category. Although our empirical results suggest that exemplars do not exhibit equal category memberships, and classifiers might be better off ignoring some exemplars or model those which are more 'central', we do not claim to have followed or avoided prototype theory [24]. Specifying the reason for this inequality, and similarly, stating an exact definition of categories irrespective of computational aspects, are outside the scope of this thesis. Instead, we consider a simplified definition of categories, specifically the one the Oxford dictionary provides.

The Oxford English dictionary defines *category* as:

> *"A class or division of people or things regarded as having particular shared characteristics."*

where it defines *characteristic* as:

> *"A distinguishing feature or quality."*

Depending on how one defines 'shared characteristics', categories in computer vision can be defined in various ways. For example, sharing characteristics might be defined as

- being visually similar to a mountain
- can be used to sit on
- being harmful to the Ozone layer
- having 6 legs

According to this definition of categories, the resulting classifications will be semantically meaningful.

In this regard, what we mean by category-free recognition is a classification problem in which none of the classes form any specific categories; interpreted with regards to the given definition of categories. Particularly, what we consider in the category-free recognition is novelty detection *i.e.* classification to 'common' and 'uncommon'. None of the 'common' or 'uncommon' classes match the given definition of a category: being 'common' or 'uncommon' in a given dataset has nothing to do with any particular characteristic that 'common' or 'uncommon' exemplars share.

In the category level recognition, we model categories *e.g.* 'cars', 'people', and 'plants'. We do not assume any particular characteristic to be shared between all exemplars of the same category. However, we assume that groups of exemplars from the same category exhibit some sort of similarity in some visual characteristic *e.g.* shape, texture, or color. Obviously, the kind of visual characteristics a category supposedly has puts some constraints on the type of information/features that are to be extracted from images. For example, a color-invariant feature/representation is inappropriate for modelling a category whose most distinctive characteristic is a color. By fixing the representation to HOG, we essentially model the shapes of (prototypes of) categories.

1.3 Parametric and Non-Parametric Classifiers

Consider a binary classification problem. If we know the distributions the exemplars of each class are sampled from, we can directly use a model of these distributions to derive a decision criterion which optimizes the expected value of a given loss function. When these known distributions have specific forms that can be specified by some parameters independent of the actual data *e.g.* Gaussian, Laplacian, and Poisson, the distribution is referred to as a parametric distribution.

On the contrary, if such a family of distributions does not exist, one can utilize mixtures of parametric distributions *e.g.* mixture of Gaussians, or non-parametric distributions *e.g.* Parzen window (kernel density estimation), to approximate the unknown distributions[4]. Both mixtures of parametric and non-parametric distributions can approximate arbitrary distributions with any desired accuracy [5]. The main difference is in the way each is constructed: the complexity of mixture of parametric models is controlled by parameters: the number of mixture components and the family of mixtures, and the complexity of non-parametric models is regulated mostly by the actual data. In other words, non-parametric distributions have less assumptions about the structure of the data, and consequently, they store and re-use the training data to represent the actual distributions; while in the mixture

[4] We do not consider mixtures of parametric distributions, a third category in addition to parametric and non-parametric distributions. While they are parametric by nature, we will argue that the distinction between non-parametric distributions and mixtures of parametric distributions fades away when considering complex enough mixtures of parametric distributions.

[5] More accurate approximations, in general, demand more samples from the distributions that are to be approximated.

of parametric case, parameters of the mixture models are learnt from the training data, which is discarded afterwards. Therefore, mixtures of parametric models are usually more expensive to 'train', but cheaper to 'test'.

Similarly, if the decision function – for the classification task – can be represented with reasonable accuracy by parametric functions *e.g.* polynomials and exponentials, the parametric functions can be used to define the decision rule, and the resulting classifier is called parametric. Alternatives are mixtures of parametric classifiers *e.g.* piecewise quadratic, and non-parametric classifiers *e.g.* nearest neighbor classifier and kernelized SVM equipped with RBF kernels. Similar to parametric and non-parametric distributions, non-parametric classifiers store and re-use copies of the actual data points, and they are usually more expensive than parametric and mixture of parametric classifiers to 'test' and cheaper to 'train'.

There is a crucial interplay between complexity of representations and complexity of classifiers. Given a classification task with arbitrary complex class distributions, the more complex the representation is, the less complexity is required from the classifier. For example, one can map linearly non-separable data to a higher dimensional space where the data is more likely to be linearly separable. This can be seen as increasing the complexity of the representation, which is traded off by using a simpler classifier in the higher dimensional space. The kernelized SVMs equipped with RBF kernels, though non-parametric and extremely non-linear, are linear classifiers in an infinite dimensional space. The complexity of the classifier not only is reduced by its linearity, but also by the max-margin constraint of the SVMs, which regulates extra degrees of freedom that are not required to separate the data [35]. Similarly, kernel embedding approaches follow similar reasonings; see *e.g.* [43].

Another example is the representation acquired from the deep learning framework [23]. The system takes as input RBG pixel values, and outputs a 1000 way classification based on approximately 60 million parameters distributed in multiple layers. The trade-off can be easily seen in this case: depending on what layer one defines the representational part to finish, the classification part is started from the next layer forward. In other words, if the representational part is seen as RGB pixel values, the coupled classifier is extremely complex and consists of multiple layers of convolutions, followed by a few fully connected layers, followed by softmax functions. On the contrary, if the representational part is seen as all the convolutional layers followed by two fully connected layers, the coupled classifier is only the softmax at the end of the network.

What makes non-parametric classifiers particularly interesting is that they can adapt their complexity based on the size of the training data: the more training data is provided to them, the more complex decision boundaries they can learn. Despite this great property, their test time complexities make them rather inapplicable to large scale scenarios. Various attempts have been made to reduce the test time complexity of non-parametric classifiers, which mainly involve approximations [11], or directly controlling the complexity of the classifier [26, 44]. We discuss the latter more in Section E.

1.4 Data Driven Recognition

The term *data driven* has been used to emphasize on data-adaptability of non-parametric classifiers, when compared to basic parametric classifiers which cannot adapt to arbitrary distributions. As argued in the previous section, mixtures of parametric classifiers have been shown to be able to adapt to arbitrarily complex data distributions, provided that the model's complexity is not over-regulated *i.e.* a large number of mixture components is allowed. Also, classifiers exist which are not non-parametric, and therefore parametric, and can adapt to arbitrary data distributions *e.g.* random forests, artificial neural networks, *etc.* Therefore, a more modern definition of what constitutes data driven learning would involve classifiers / regressors complex enough to perform the required recognition tasks with any desired level of accuracy, given enough training data.

We have already motivated that the classifier / regressor is tightly coupled with the representation it builds upon. Additionally, we have argued in [3, 5] that the right kind of data for data driven methods will most likely have to satisfy some qualitative constraints. We discuss in Section D that for any complex enough model, there is a very crucial factor which is usually neglected: the distribution of the training data provided to the model. We argue that by considering the data as a design parameter, a new kind of recognition is brought to life, which actually concerns with 'tuning the training data'. In other words, we will argue that classifiers might learn better from a subset of training data, and they could potentially select, from a large pool of candidate training data, exemplars which will help the classifier generalize better.

Although we motivate and partially demonstrate this in Section D, we think that more solid theoretical or experimental results are needed to verify the extent of validity of this hypothesis. Therefore, our definition of data driven in this thesis will be one which acknowledges the importance of 'right kind of data' but does not specify how such data might be acquired:

> ***Data Driven Learning****: A process involving the design and training of classifiers/regressors and representations, which can perform required recognition tasks with any desired performance, given a sufficiently large number of suitable training data.*

It is worth emphasizing that a desired recognition performance dictates requirements from training data in addition to requirements from the model. For example very high categorization performances require

- non-gradual category memberships *e.g.* high resolution images of cars vs plants
- either
  - very strong priors about the natural world *e.g.* rich enough representations which lead to separable data and a very small Bayes risk

  - a complex enough machinery for learning the representations and classifiers + sufficiently many training exemplars from which a representation and a classifier can be learnt that can match the required categorization performance

In other words,

- Problems such as ‘young’ vs ‘old’ will not achieve very high recognition performance as the ‘young’ and ‘old’ categories are not mutually exclusive.
- In absence of rich representations that provide rather certain information about occlusions, lighting conditions, pose, *etc.*
  - A linear classifier using HOG representation cannot accurately classify cars from arbitrary view points, no matter how much data is used.
  - A view point dependent representation requires many samples from various view points and a complex classifier that can adapt to the (potentially) multi modal distribution of positives and negatives, with some mechanism for occlusion reasoning
  - An algorithm which learns a view point independent representation and learns a complex classifier that can correctly classify occluded exemplars under the learnt representation, will require many examples of objects in different view points – potentially augmented with correspondences in multiple views – exhibiting various occlusion patterns

Chapter 2

Category Free Recognition

This chapter introduces the category-free recognition problems we consider in this thesis. Section 2.1 overviews category-free recognition problems frequently encountered in computer vision. Section 2.2 discusses novelty detection as a recognition problem. The category-free part of this work mostly involves novelty detection in wearable visual systems, which we introduce and motivate in Section 2.3.

2.1 Category Free Recognition in Computer vision

Many problems in computer vision are category free. For example, object boundary detectors such as [9, 27], involve category-free recognition of pixels in images which correspond to the contours of the objects in the natural images. Having corresponding pixels in images end up on salient contours of the object is not a characteristic that parts of objects in real world share. Similarly, non-semantic segmentation methods such as [7, 6], are category-free, and so are general object tracking systems *e.g.* [21]. Interpreted according to the definition of recognition discussed in the previous chapter, in all these problems some visual pattern from a training set is being re-identified: visual patterns corresponding to pixels on prominent boundaries of objects, visual patterns that correspond to one prominent object within images, and the visual pattern corresponding to the object that is to be tracked.

2.2 Novelty Detection as a Recognition Problem

As motivated earlier, novelty detection is a form of category-free recognition. Novelty detection and outlier detection are sometimes considered the same. We make a subtle distinction between the two. Outlier detection is mostly about identifying data points which do not seem to be similar to the majority of data points. The main reasons for a point to become an outlier might be sensory or manual labelling errors, or not agreeing with a model which explains a significant majority of the data points. Novelty detection on the contrary aims to identify patterns which are believed to be of interest, and the reason for identification of those patterns is not believed to be a sensory error, or mis-labelling of the data. In other words, the aim of the outlier detection is to have a model which clearly explains the data, while the aim of novelty detection is identification of rare patterns.

In order to detect novelties, one needs to identify the novel pattern to belong to a certain group or category, and then identify it as being novel in that group or category. In essence, non-novel patterns need to be recognized in order for the novel patterns to be classified as novel. This can be identified in the works that we present in Section A and Section B, where the novel temporal segments are found within sequences of images which partially share structure with non-novel sequences, and novel spatial segments are found within images which partially share structure with non-novel images.

2.3 Novelty Detection in Wearable Visual Systems

Digital cameras are becoming smaller, processors are becoming more powerful and energy efficient, communications are becoming faster, storage units are becoming smaller and support more capacity; and all of them are becoming cheaper. The result is that light devices equipped with small cameras, good processors, plenty of storage capacity, and fast communications are becoming cheap to produce. Many similar devices have existed for quite a few years now: Microsoft SenseCam, Muvi Atom, and GoPro; some have been recently produced: Google glass; and more will be developed and mass marketed soon.

Most of these devices are designed to be continuously worn and/or record lengthy video footage. Each hour of video footage, recorded at 30 HZ, comprises more than 100,000 images. When sub-sampled at 1 HZ, 8 hours of daily recording results in over a million images each year. Storing, processing, or just viewing these images, as might be required by applications such as life logging or memory therapy, will be associated with huge costs in terms of storage, computation, or attention time. An automatic selection process will be crucial to help manage such a large body of images.

Novelty detection can be used to filter out what is most common in such large datasets, leaving only those which are in some sense rare. We propose two different novelty detection frameworks based on wearable footage. The first one, presented in Section A, explores novel ego-motion detection in sequences of images of a subject walking from a metro station to work on a daily basis. We show that the novelties the system detects reflect events such as 'running into a friend', 'meeting a friend', 'giving directions', and 'buying ice-cream'. The second one, presented in Section B, seeks novel spatial patterns in images of roughly the same physical place *e.g.* in front of the metro station. We show that what is mostly novel in images of the same outdoors urban areas are 'people', 'bicycles', 'cars', *etc.*, which are not parts of a static environment. We believe these results to reflect a useful 'filtering process', which allows the rest of the sequences to be summarized as 'the usual'.

Chapter 3

Category Level Recognition

This chapter introduces the problems related to category level recognition that we study in this work. Section 3.1 overviews the category level recognition problems frequently encountered in computer vision. Section 3.2 overviews the most prominent object recognition systems. Section 3.3 discusses data driven object recognition, and what our works contribute to it. In Section 3.4 we discuss modelling the interplay between representations, classifiers, and the data.

3.1 Category Level Recognition in Computer Vision

Many problems in computer vision are category based. For example, scene recognition [31] aims to classify images of scenes to categories such as 'indoors', 'cinema', and 'park'. In object recognition, the aim is to either determine if an object from categories such as 'chair', 'person', and 'car' exists within an image [16, 15], or to localize such instances within images [16, 18]. Semantic segmentation [20] is about partitioning pixels of images according to the categories they belong to. Similarly, in pose estimation [46, 22, 8] and action recognition [12], the 'human' category is the focus of the model, and in face recognition [38] categories such as 'person 1', 'person 2', *etc.* are considered.

3.2 A Brief Overview of Object Recognition Systems

The category level part of this thesis investigates the object recognition problem. In cognitive neuroscience, object recognition is defined as the ability to perceive objects' (visual) properties such as shape, texture, and color, and assign semantic attributes to them. In this work, we are not concerned with the semantics associated with objects, other than the categories they belong to. Additionally, since we use the HOG representation, we mostly model shape rather than color or texture.

Object recognition in computer vision is usually divided into three distinct problems: 1) instance level recognition, where the goal is to identify the same object in novel images, 2) object category classification, where the goal is to determine if instances of different categories exist within an image, and 3) object category localization which aims to localize instances of categories within images. The complexities of these problems are usually considered to be in the same order.

Instance level recognition can be performed via local feature matching [28], potentially followed by a geometric verification step *e.g.* [39], or by other means of modelling the appearance *e.g.* [21]. The classification problem usually involves holistic reasoning based on (explicit or implicit) global representations [23, 15], and therefore the training and testing samples usually do not exhibit significant scale variation within images. The detection problem is usually considered the most complex, as detectors are expected to handle significant position and scale variations of objects within images. The inference process usually needs to classify all the bounding boxes with varying aspect ratios and scales over an image. The consequences are: 1) the training procedure becomes much more expensive than holistic approaches, as all bounding boxes that do not overlap significantly with the provided ground truth bounding boxes define 'negative' samples, and 2) the inference procedure for each image becomes very expensive which renders expensive classifiers rather impractical. There are approaches that do not implement scanning window classifiers, but rather use more sophisticated approaches for pruning the search space, or speeding up the computations *e.g.* [42, 33, 19]. Nevertheless, the detection problem is considered to be more complex than the classification problem (when dataset sizes are comparable).

In this thesis, we consider both classification and detection problems. In Section C we present our object detection model. Section E describes our object classification system.

3.3 Data Driven Object Recognition

Object Detection: As motivated earlier, we consider mixtures of parametric classifiers as data driven methods, provided that they have a sufficiently large number of mixture components.[1] This suggests that such models need to adapt their complexity to the training data distributions. Cross-validation can be used to tune the number of mixture components; provided that these models can learn mixtures with different number of components equally well.

As the optimizations involved in training mixture models are non-convex in the absence of fixed data-mixture component associations (latent associations), these models are sensitive to initializations. We consider a data driven clustering step, based on sophisticated visual similarity measures, to provide the initial data-mixture component associations within the mixture learning framework. The

[1] The deformability of part based models such as [18] can be potentially considered equivalent to compressing many rigid templates in one deformable component [47]. However, deformable models such as [18] usually limit the deformability of parts and strongly penalize highly deformed part configurations. In this sense, even deformable part based models require many global mixture components in order to be able to handle high intra-class variation.

approach is shown to adapt the mixture models to the distributions of data better than the simple and common alternative of clustering the data based on the aspect ratio of bounding boxes. This is discussed further in Section C.
Object Classification: One of the most popular non-parametric classifiers is the kernelized SVM equipped with RBF kernels. The RBF kernel performs particularly well when the Euclidean distance on the input representation is a reasonable measure of dissimilarity. The main problem with such a configuration is the expensive training and testing procedures associated with it, in addition to the inappropriateness of the Euclidean distance, or other metrics [34], for most of the representations of visual patterns. In Section E, we introduce a non-parametric classifier which has better training and testing costs (computational and memory), while it does not assume metrics, or more accurately positive (semi-definite) similarity measures. What is particularly interesting about this model is that it can be equipped with deformable/invariant similarity measures which are indefinite.

3.4 Modelling Categorization Problems

Due to the interplay between representations, classifiers, and the training data, and due to a lack of understanding about what constitutes a good representation and classifier configuration for particular data distributions, cross-validation has been the dominant approach to select representations and classifiers which perform well on particular training sets. The lack of objective measures for describing and characterizing distributions of data has contributed to the practice of trying various combinations of representations and classifiers, and selecting the configuration which performs well on held out data.

The cross-validation procedure does not provide much insight on how to select new configurations for unseen distributions of data, nor does it provide any objective measure of what kind of performance one can expect with certain representation - classifier combination without going through the expensive training-testing procedure for various combinations [47, 40].

In Section D, we introduce our solution to this problem: we aim to model various factors that affect a recognition system, which in addition to the data-describing measures that we introduce, is able to characterize distributions of data. For example, it can quantify semantic characteristics of training data *e.g.* intra-class variation, connectivity, *etc.*, and predict the test performance of specific representation/classifier choices with reasonable accuracy.

Chapter 4

Summary of Papers

A Novelty Detection from an Ego-Centric Perspective

This paper presents a solution for *temporal* segmentation of novelties in ego motion of a person walking from a metro station to work [6]. The novelty detection presented in the paper is performed in a non-parametric category-free manner. The sequence of images acquired by sub-sampling a new query video are registered to the stored reference sequences and the temporal segments which cannot be registered are identified as novelties. This is demonstrated in Figure 1, and an example is depicted in Figure 2.

The registration is performed by aligning sequences according to a pairwise similarity of ego motion defined on the frame level. As a sequence of similar view points reflects similar ego-motions in the same environment, the similarity between view-points is utilized in a dynamic time warping algorithm to register sequences.

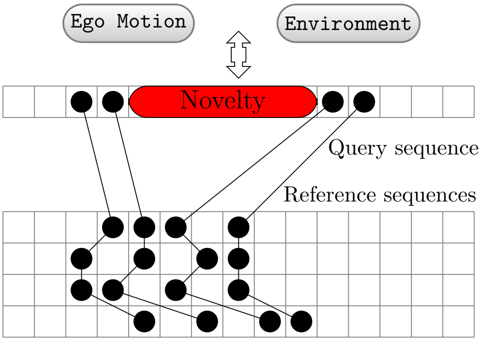


Figure 1: Novelty detection via sequence alignment. Each block represents an image, and each row represents frames sub-sampled from a video. Links represent correspondences between frames.

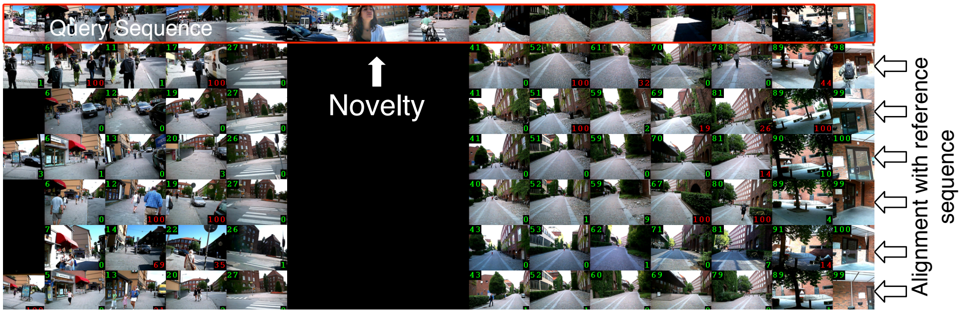


Figure 2: Reference sequences are aligned with the query sequence and novelty is detected

Similarity between view-points is approximated by the number of matching local descriptors between pairs of frames. As local appearance based descriptors are ambiguous by nature, Epipolar geometry is utilized to ensure that only the local features that correspond roughly to the same point in the 3D world are matched together.

Experimental parts of the paper suggest that such an approach can identify deviations from an implicitly learned model of 'normal' ego motions in the same environment. Storing approximately 5 reference sequences was shown to be sufficient in order to recognize environment-specific 'normal' ego motions, which in turn results in detection of novelties.

B Multi View Registration for Novelty/Background Separation

This paper presents a solution for *spatial* segmentation of novelties in multiple images of the same enviornment [7]. Similar to the previous paper, the environmental setup is that of a person walking from a metro station to work, and recognition is performed in a non-parametric and category-free manner. Given a query image and multiple reference images of the same enviornment, reference images are spatially registered to the query image, and novelty is defined as the regions in the query image which cannot be explained by the registered reference images. This is depicted in Figure 3.

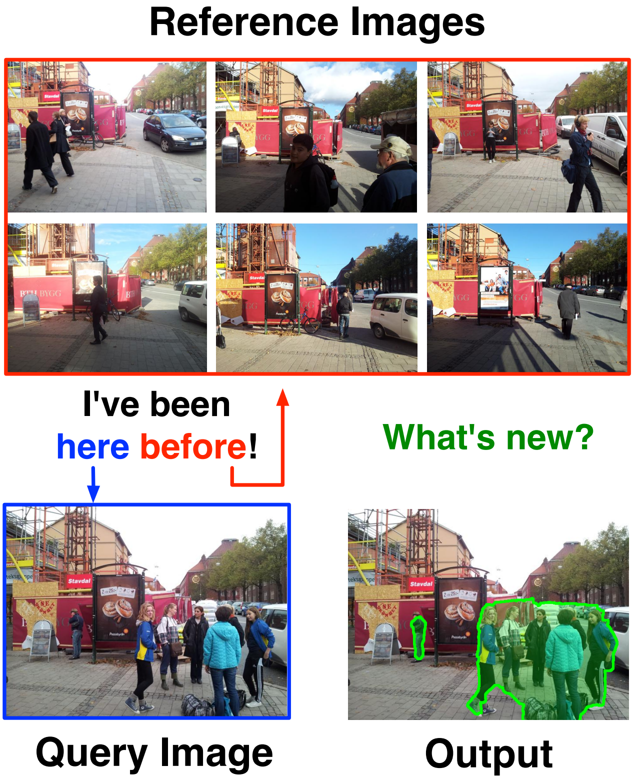


Figure 3: Our system takes as input a query image and multiple reference images. We assume all these images are of the same environment taken from approximately the same view point but at different times. The algorithm segments out objects in the query image which are not part of the environment. The bottom right figure shows the computed segmentation.

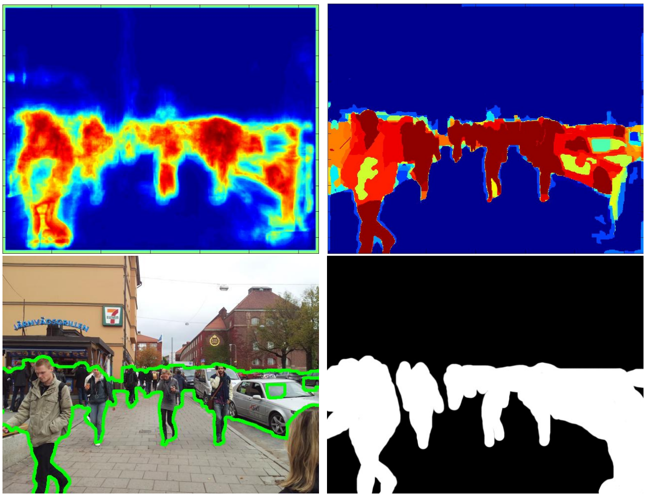


Figure 4: From top left to bottom right: initial probability of novelty, final probability of novelty, final segmentation, and the ground truth labelling of novelties in an example.

The reference images are registered and warped towards the query frame separately. The appearance-based residuals of warping errors from reference images are then aggregated in a fixed-length vector, which in turn approximates an initial probability of novelty via a regression function. The parameters of this regressing function are learnt in a supervised way, and the resulting estimate is used in multiple iterated graph cuts segmentation procedures with different parameter settings representing different priors. The solutions from each of the segmentation procedures are then aggregated into a final probability of novelty. The parameters of this regressing function are also learnt in a supervised manner, and a final non-iterated graph cuts segmentation produces the output of the algorithm. The regressed probabilities of novelties, the final output and the defined ground truth for an example query image are depicted in Figure 4.

Experimental results of the paper suggest that the proposed method detects spatial novelties in the query image such as ‘cars’, ‘bicycles’, ‘people’, etc. It is shown that the method can produce reasonable outputs, given roughly 5 reference images for a query image.

C Mixture Component Identification and Learning for Visual Recognition

This paper proposes a framework for learning mixtures of rigid templates targeting category level recognition, where the data-component associations are initialized through a sophisticated clustering of category-specific exemplars [2]. Each category is partitioned into a fixed number of clusters, based on pairwise affinity of exemplars within the category. Mixtures of linear classifiers are then learnt based on these clusters in a binary (one vs rest) manner. This is demonstrated in Figure 5.

The pairwise similarity measure used in the clustering step performs feature selection on the exemplar level via discriminative reasoning. This was shown to perform better than coarser measures of similarity such as similarity in aspect ratio of the bounding boxes, or appearance based measures that do not perform feature selection. The similarity measures considered in the paper were rigid *i.e.* they did not model deformability of the exemplars. Consequently, the resulting mixture components considered are rigid and not deformable.

The clustering algorithm considered does not assume any shape or property for the clusters, but it requires the number of clusters to be determined apriory. Although there are ways to determine the number of clusters based on distributions

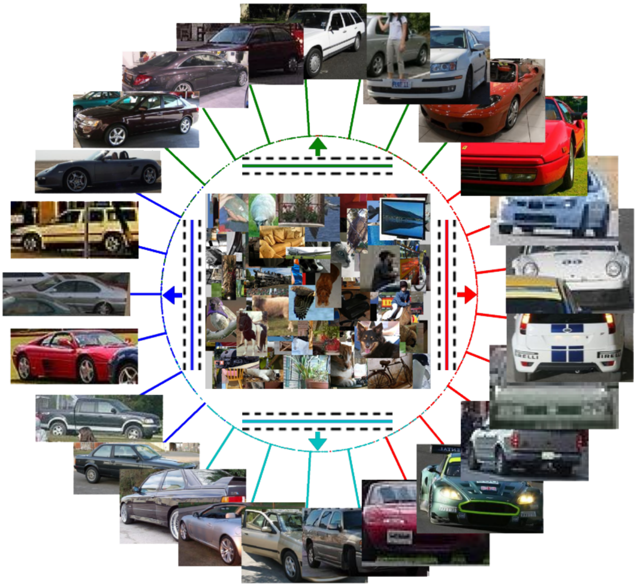


Figure 5: The high level overview of our approach. We group visually similar positive instances together and for each cluster, learn a linear classifier which separates the cluster from all negative data. Each color represents a different cluster.

of the data, they are not studied in this paper. As a part of the clustering step, an embedding of the data is acquired which is qualitatively shown to match our perceptual evaluation of similarity, given that images have sufficient resolutions, and that the considered category does not exhibit “too much intra-class variation”[1].

The training of the mixture model involves a non-convex optimization problem, which in turn makes the model sensitive to initial data-component associations provided to the optimization process. It is shown that the clustering equipped with the feature-selecting similarity measure constitutes the best initialization among the ones that were considered. The resulting model was shown to outperform all other non-deformable models that are based on the same feature that was used in our study. Figure 6 visualizes the clusters and the filters learnt for each cluster for the ‘car’ category.

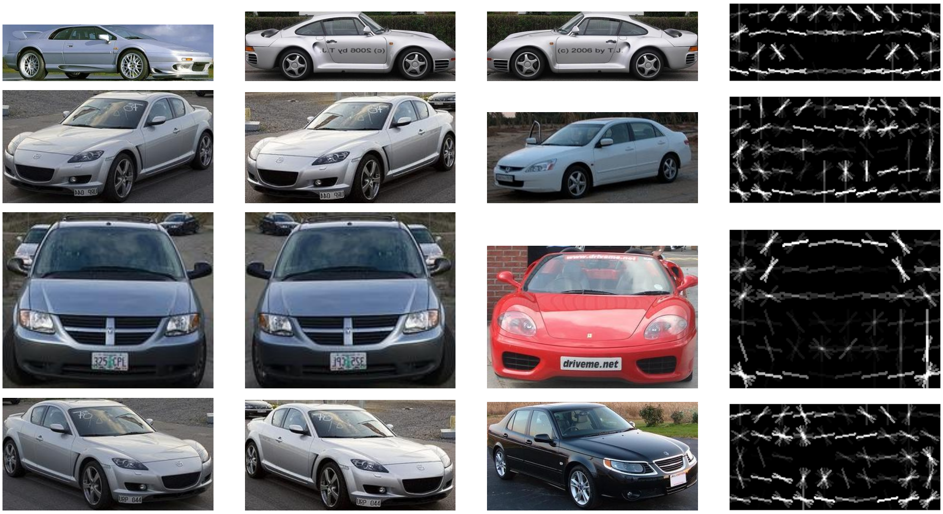


Figure 6: Visualization of the clusters and the filters learnt for each cluster.

[1] This observation motivated the study in the next paper, where we investigated what comprised “too much intra-class variation”, and how to measure it.

D Properties of Datasets Predict the Performance of Classifiers

Inspired by the observations regarding the intra-class variation and the test performance in the previous paper, this paper aims to quantify the dependencies between test performance, properties of the datasets, and other factors that affect the test performance [5]. The paper demonstrates that these dependencies can be modelled in a way that under some reasonable assumptions, the resulting model can predict the test performance that particular classifiers will achieve when trained on rather arbitrary training sets. The model that has been used in the paper is demonstrated in Figure 7.

A rather critical assumption in the model is that training sets and test sets are sampled from the same distribution. The distributions of training exemplars, and similarly the testing exemplars, are quantified via aggregations of affinity measures on the training set. A regressing function is then defined which maps these aggregated measures to the test performance. As a training step of this model, the parameters of this regressing function are learnt in a supervised manner *i.e.* the aggregating measures are computed on different training sets, models are trained on the training sets and tested on the corresponding test sets, and the test performances are provided to the learning algorithm as ground truth regression targets.

The test performances predicted by this model are shown to rather accurately agree with the novel test performances. While the model does not consider all the factors involved in the recognition process, we show that it can reasonably predict the majority of the variation in the observed test performances. In particular, the 'connectivity' of the training exemplars is shown to play the most significant role in determining the test performance. The correlation of the 'connectivity' measure to the test performances is shown to be stronger than that of the 'intra-class variation' or other measures. This suggests that the existing models have specific requirements

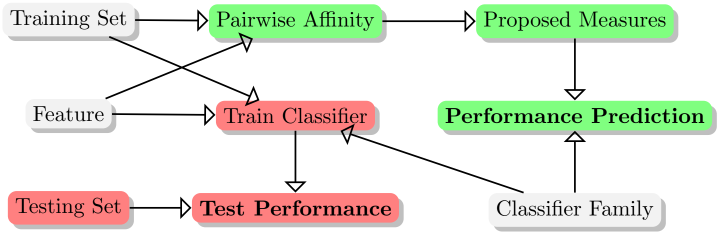


Figure 7: The training-testing process (red boxes) and the proposed test performance prediction process (green boxes). The direction of arrows determines the flow of information and also the dependencies. Both procedures are dependent on the white boxes.

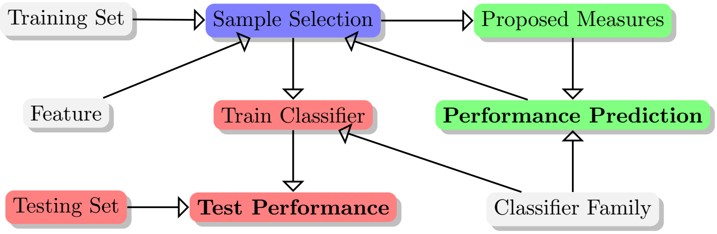


Figure 8: Illustration of the proposed procedure for sample selection.

about the training data. In other words, the existing models will perform better if they are provided with the right kind of training data.

Being able to quantify the quality of a training set, the next step explored in the paper is to modify the training set in a way which best suits particular classifiers, see Figure 8. Due to the assumptions associated with the model used in the paper, the modifications to the training sets are required to be small *i.e.* radical changes to the training set violate the assumptions of the model, making the resulting predictions invalid. However, the small changes to the training set that are acquired from the current model agree substantially with our expectations in that, what is being suggested to remove from the training set are *outliers* that are not *connected* to the

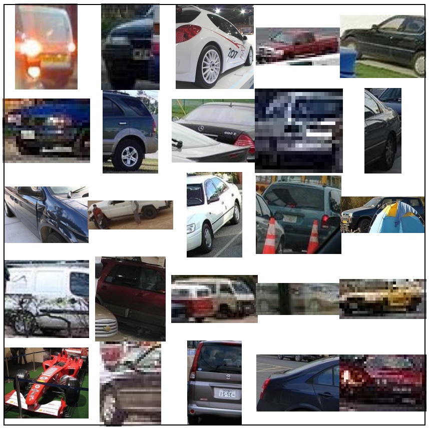
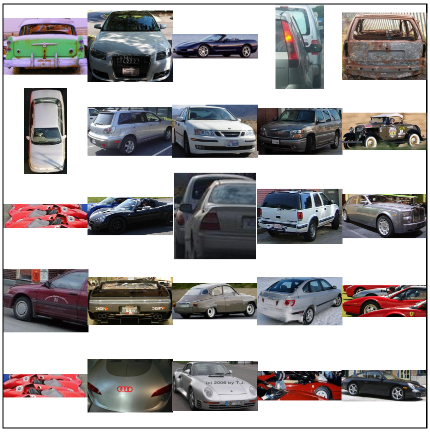
(Worst Outliers) Remove to gain 1% AP	(Best Inliers) Remove to lose 1% AP
(Remove 97) APPred=56, AppxAPPred=55 	(Remove 72) APPred=55, AppxAPPred=53 

Figure 9: Demonstration of *Automatic Dataset Selection*. For the 'car' class of Pascal VOC 2007, exemplars are shown which upon removal from the training set result in 1% change in the predicted test performance.

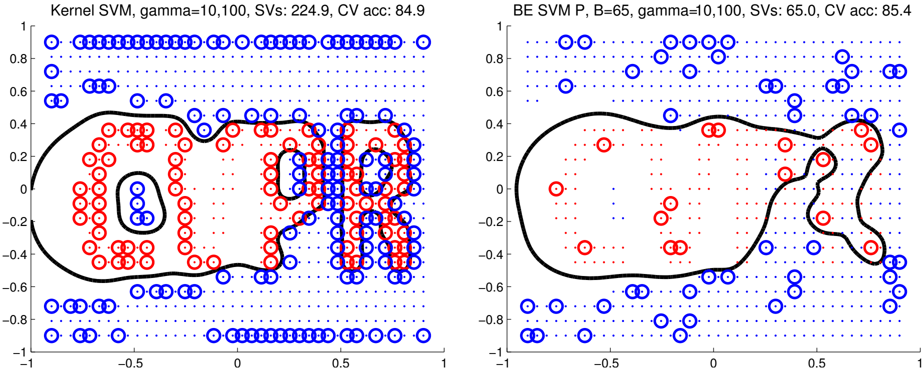
rest of the training set; as reflected by the affinity measure. The exemplars which are labelled *best inliers* on the other hand, are those which keep different groups of training exemplars, distributed in form of clusters in the ‘configuration space’, *connected*. In other words, these exemplars ensure that enough *support* in rather *critically undersampled* areas of the ‘configuration space’ is retained. Figure 9 depicts qualitative results for such a procedure.

E Large Scale, Large Margin Classification using Indefinite Similarity Measures

This paper proposes a scalable large-margin non-parametric categorizer equipped with deformable indefinite similarity measures [4]. The model, named Basis Expansing SVM (BE-SVM), is based on a normalization of empirical kernel maps based on a restricted set of bases. The resulting optimization procedure is convex, and in fact, general fast approximate linear SVM solvers are used to optimize the model's parameters.

The pairwise similarity measures used in the paper are generalizations of invariant kernels, which search for the optimal (global) translation and (local) deformations which maximize the similarity between pairs of instances. The measures are more expensive than the RBF kernels, and they are indefinite *i.e.* not positive (semi) definite. However, it is shown that the negative eigenvectors of the resulting similarity matrices hold significant discriminative information. This suggests that metric restrictions on measures are not necessarily optimal for classification.

The empirical kernel maps are used to acquire a fixed length representation of exemplars based on the given similarity measures. The representation is acquired by evaluating the similarity of the instance to a fixed set of bases. Various basis selection strategies are investigated which did not exhibit significant change in the performance on the CIFAR-10 dataset. It can be expected that on more challenging datasets such as Pascal VOC, the basis selection strategy will be of much more



(a) Kernelized SVM (b) BE-SVM primal objective (reduced)

Figure 10: Demonstration of kernelized SVM and BE-SVM using two Gaussian RBF kernels. Figure 10(b) is based on 10% of the data randomly selected as bases. 10 fold cross validation accuracy and the number of support vectors are averaged over 20 scenarios based on the same problem but with different spatial noises. The visualization is on the noiseless data for clarity. Best viewed electronically.

significance, as motivated by the work presented in the previous section.

Although it is expensive to evaluate this fixed length representation, it has the benefit of simplifying the resulting optimizations step. The computational complexities and memory requirements for training and testing of the proposed model are compared to that of the kernelized SVM based on positive definite kernels, and the proposed model is shown to have similar or better computational and memory requirements.

The experimental results on the CIFAR-10 dataset suggest that the proposed model equipped with proper invariant similarity measures outperforms the kernelized SVM based on the optimal parameter setting. It is shown that the model outperforms the competitors given the same number of supporting exemplars, or the same number of model parameters. Figure 10 depicts the proposed model in comparison to kernelized SVM on 2D toy data.

Chapter 5

Discussion and Conclusions

In this thesis, we investigated two types of recognition problems: category level recognition which models categories (and the inherent semantics associated with them), and category-free recognition which does not assume any semantics associated with what is to be recognized.

In the category-free part, we considered novelty detection in spatial and temporal domains. We showed that having access to roughly 5 reference exemplars allows a non-parametric model to perform novelty detection within the problem domains that we considered. Since the view-point change was limited in our scenarios, we expect this number to increase when significant view point changes are to be addressed by the system. However, as motivated, the data driven approach can overcome these limitations, simply by making use of more of the right kind of training data, and without any significant changes to the proposed models.

In the category-level part, we demonstrated how particular mixture models can be better adapted to the training data. Particularly, we showed that by using careful clustering of the training data, and using these clusters as initialization for the mixture models, more complex mixture models can be utilized which adapt better to the training data *i.e.* they can become more data driven.

We also demonstrated how to make use of invariant/deformable similarity measures in a non-parametric manner, while achieving reduced training and testing costs associated with non-parametric models. Particularly we showed that deformable similarity measures can play a significant role in designing scalable non-parametric classifiers.

Finally, we demonstrated how a model of recognition systems can be constructed which can characterize distributions of data under specific representations, and can quantify the relation between characteristics of data and expected recognition performance on similar test data. We believe this to be a first step towards gaining a more detailed insight on the interplay between data, representations, and the test performance of recognition systems. A more sophisticated version of such models would allow us to optimize data as a design parameter *i.e.* they could be used to automatically ensure that the training data satisfies particular qualitative requirements which are shown to significantly correlate to recognition performance.

Future Work

In the category-free part, it will be interesting to investigate other types of novelties that can be extracted from large scale data, potentially acquired from wearable cameras. A more detailed insight about how the human brain selects memories would be beneficial for the continuation of the same research path. This would involve inter-disciplinary research between psychologists, neuro-scientists, and computer vision scientists/experts.

In the category level recognition part, it will be interesting to apply the framework introduced for analyzing representations, classifiers, and data distributions, to representations other than HOG, in order to gain a more detailed insight on how different representations change properties of the data: do they reduce the intra-class variation, increase data connectivity, or change other measurable properties of the training data?

Developing a more complete model of category recognition systems – one which allows large modifications to the training data – will also be beneficial. This would result in systems which can select optimal training data most suitable for categorizing desired target distributions.

Bibliography

[1] A. Agarwal and B. Triggs. Hyperfeatures - multilevel local coding for visual recognition. In *European Conference on Computer Vision*, 2006.

[2] O. Aghazadeh, H. Azizpour, J. Sullivan, and S. Carlsson. Mixture component identification and learning for visual recognition. In *European Conference on Computer Vision*, 2012.

[3] O. Aghazadeh and S. Carlsson. Properties of datasets predict the performance of classifiers. In *British Machine Vision Conference*, 2013.

[4] O. Aghazadeh and S. Carlsson. Large scale, large margin classification using indefinite similarity measures. In *Submitted to: Neural Information Processing Systems*, 2014.

[5] O. Aghazadeh and S. Carlsson. Properties of training data predict the performance of classifiers. *Sumbitted to: International Journal of Computer Vision*, 2014.

[6] O. Aghazadeh, J. Sullivan, and S. Carlsson. Novelty detection from an egocentric perspective. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[7] O. Aghazadeh, J. Sullivan, and S. Carlsson. Multi view registration for novelty/background separation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[8] K. Alahari, G. Seguin, J. Sivic, and I. Laptev. Pose estimation and segmentation of people in 3d movies. In *IEEE International Conference on Computer Vision*, 2013.

[9] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33:898–916, 2011.

[10] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics).* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. ISBN 0387310738.

[11] O. Boiman, E. Shechtman, and M. Irani. In defense of nearest-neighbor based image classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[12] P. Bojanowski, F. Bach, I. Laptev, J. Ponce, C. Schmid, and J. Sivic. Finding actors and actions in movies. In *IEEE International Conference on Computer Vision*, 2013.

[13] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[14] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[15] O. Duchenne, A. Joulin, and J. Ponce. A Graph-matching Kernel for Object Categorization. In *IEEE International Conference on Computer Vision*, 2011.

[16] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[17] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[18] P. F. Felzenszwalb, R. B. Girshick, D. A. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1627–1645, 2010.

[19] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR*, 2013.

[20] A. Ion, C. Sminchisescu, and J. Carreira. Probabilistic Joint Image Segmentation and Labeling by Figure-Ground Composition. *International Journal of Computer Vision*, 2014. Submitted.

[21] Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-learning-detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34:1409–1422, 2012.

[22] V. Kazemi, M. Burenius, H. Azizpour, and J. Sullivan. Multi-view body part recognition with random forests. In *British Machine Vision Conference*, 2013.

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Neural Information Processing Systems*, 2012.

[24] G. Lakoff. *Women, Fire and Dangerous Things: What Categories Reveal About the Mind.* University of Chicago Press, 1987. ISBN 9780226468037.

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

[26] Y. J. Lee and O. L. Mangasarian. Rsvm: Reduced support vector machines. In *Data Mining Institute, Computer Sciences Department, University of Wisconsin*, 2001.

[27] M. Leordeanu, R. Sukthankar, and C. Sminchisescu. Generalized Boundaries from Multiple Image Interpretation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014. Submitted.

[28] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, pages 91–110, 2004.

[29] D. Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information.* Henry Holt and Co., Inc., New York, NY, USA, 1982. ISBN 0716715678.

[30] K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *International Journal of Computer Vision*, pages 63–86, 2004.

[31] S. Naderi-Parizi, J. G. Oberlin, and P. F. Felzenszwalb. Reconfigurable models for scene recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[32] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[33] N. Razavi, J. Gall, P. Kohli, and L. J. Van Gool. Latent hough transform for object detection. In *European Conference on Computer Vision*, 2012.

[34] W. J. Scheirer, M. J. Wilber, M. Eckmann, and T. E. Boult. Good recognition is non-metric. *CoRR*, 2013.

[35] B. Schölkopf, R. Herbrich, and A. J. Smola. A generalized representer theorem. In *Computational Learning Theory*, 2001.

[36] A. Sharif-Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. *CoRR*, abs/1403.6382, 2014.

[37] K. Simonyan, A. Vedaldi, and A. Zisserman. Learning local feature descriptors using convex optimisation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014.

[38] J. Sivic, M. Everingham, and A. Zisserman. “Who are you?” – learning person specific classifiers from video. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[39] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *IEEE International Conference on Computer Vision*, 2003.

[40] A. Torralba and A. A. Efros. Unbiased Look at Dataset Bias. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[41] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. *CoRR*, 2013.

[42] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In *IEEE International Conference on Computer Vision*, 2009.

[43] A. Vedaldi and A. Zisserman. Efficient additive kernels via explicit feature maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011.

[44] C. Williams and M. Seeger. The effect of the input density distribution on kernel-based classifiers. In *International Conference on Machine Learning*, 2000.

[45] Q. Z. Wu, Y. LeCun, L. D. Jackel, and B. S. Jeng. on-line recognition of limited vocabulary chinese character using multiple convolutional neural networks. In *IEEE International Symposium on circuits and systems*, 1993.

[46] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[47] X. Zhu, C. Vondrick, D. Ramanan, and C. C. Fowlkes. Do we need more training data or better models for object detection? In *British Machine Vision Conference*, 2012.

Part II

Included Publications

Paper A

Novelty Detection from an Ego-Centric Perspective

Omid Aghazadeh, Josephine Sullivan and Stefan Carlsson

Published in
IEEE Conference on Computer Vision and Pattern Recognition

Novelty Detection from an Ego-Centric Perspective

Omid Aghazadeh, Josephine Sullivan and Stefan Carlsson

Abstract

This paper demonstrates a system for the automatic extraction of novelty in images captured from a small video camera attached to a subject's chest, replicating his visual perspective, while performing activities which are repeated daily. Novelty is detected when a (sub)sequence cannot be registered to previously stored sequences captured while performing the same daily activity. Sequence registration is performed by measuring appearance and geometric similarity of individual frames and exploiting the invariant temporal order of the activity. Experimental results demonstrate that this is a robust way to detect novelties induced by variations in the wearer's ego-motion such as stopping and talking to a person. This is an essentially new and generic way of automatically extracting information of interest to the camera wearer and can be used as input to a system for life logging or memory support.

1 Introduction

In this paper we address the problem of selecting and storing relevant parts of the visual input collected from a continuously worn camera capturing images at video rate. This problem is partly dictated by applications such as life logging [3, 9, 1] and memory support systems for the disabled [5]. Especially in the design of efficient memory support, there is a large potential advantage in the automatic selection of relevant moments of one's daily visual experience.

Memory selection depends on several factors relating to the complex state of the human observer and these are not primarily related to vision. Given just the visual input, however, we can ask ourselves which moments of the input we would like to capture and store and if there are any rules that can be formulated for this.

It is generally accepted that *novelty* is very central in deciding whether to remember something or not. It is a very natural criterion for selection both on pure data storage grounds as well as for the purely subjective reasons of later inspection of stored images. Heuristically novelty can be measured as the deviation from some

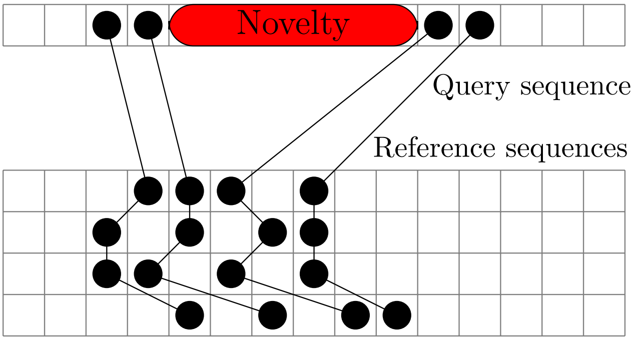


Figure 1: Novelty detection via sequence alignment.

standard background. The less variation there is in the background the easier it will be to detect novelty. One way to ensure that the background variation is limited is to choose a specific context within which novelty is selected.

Here we choose the simple context of the daily repeated activity of going to work. The collected video sequences from various days therefore contain image frames captured from approximately the same location. The influence of the day-to-day variation of these locations can be further reduced by aligning corresponding frames from different days using appearance and geometry information in the image frames. The content of a recorded sequence depends on two main factors: 1) the ego motion of the person wearing the camera and 2) the environment in which the sequence is captured. If there is a sufficient variation in one of these factors, this leads to the inability to register some or all of the sequence to previously stored sequences. This inability is taken as a measure of novelty. Ideally variations such as the person deviates from his/her daily path or stops to do some shopping or a street being shut off should be captured by our system.

This work extends previous studies based on wearable cameras in two main ways: 1) We use a very small (4cm high) camera that captures image at video rate for one hour and stores it on a memory stick. 2) Video is captured from daily repeated activities such as going to work and we develop algorithms for the automatic frame to frame registration of sequences recorded on different days. 3) We define novelty based on the absence of a good registration between a sequence and stored reference sequences.

The rest of paper is organized as follows. We begin by presenting in section 2 the details of our sequence alignment algorithm. This algorithm establishes frame to frame correspondences between two sequences. Section 3 then describes how the correspondences between sequences can be utilized to detect novelties. Afterwards, we present evaluation of the components of the proposed algorithm in section 4. Section 5 shows the results of the novelty detection algorithm and finally, section 6 concludes the paper.

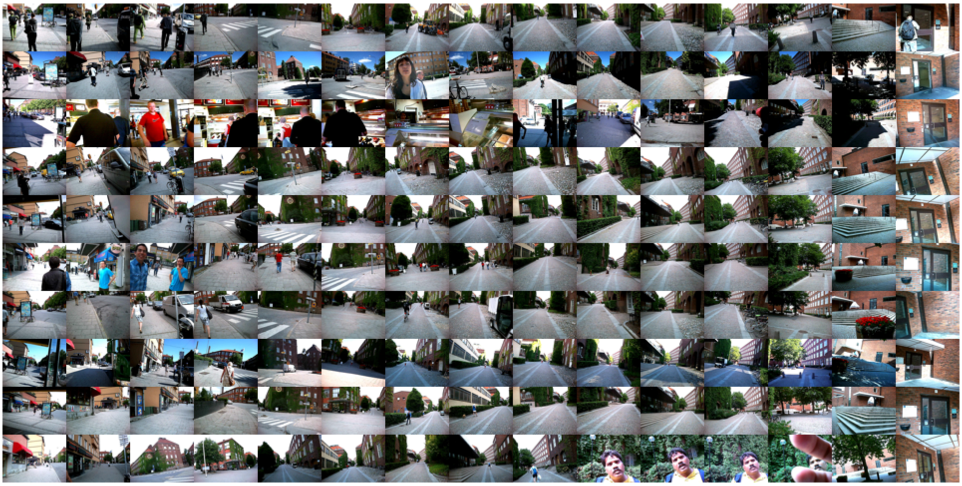


Figure 2: Each above row shows a highly temporally sub-sampled sequence from our dataset. Each sequence corresponds to a different day and captures what the subject experienced visually on his way to work.

2 Sequence alignment

Figure 2 displays 10 sequences from our dataset. Each row corresponds to one sequence and is of the subject walking from the a metro station to his work place. All our sequences are frames sampled from 25Hz videos at 1Hz. We wish to put the frames of one sequence s_1 in correspondence with another sequence s_2. As the sequences we capture have temporal continuity characteristics and repeated underlying structures, a natural way to establish correspondences is with Dynamic Time Warping (DTW). This algorithm requires a measure of similarity between each frame of s_1 and each frame of s_2 and the rest of this section is mainly devoted to how we compute this.

2.1 Appearance based cues

The most straightforward approach to define a measure of similarity between two sequences is to represent each frame with a fixed length vector and compare the representative vectors with a kernel such as polynomial or minimum intersection kernel. In order to represent frames with a fixed length vector, a common approach is to model the distribution of some local visual words[1] disregarding their spatial information.

[1] We use the terms visual words and features interchangeably in this article.

Local features are fixed length description of some local interest regions localized in different areas of an input image. SIFT [7] and its variations are one of the most commonly used region descriptors. Various methods detect interest regions based on different criteria such as the determinant of the Hessian or the Harris tensor. A thorough study of region descriptors and interest region detectors is performed in [12]. Alternatively, it is possible to densely sample the SIFT features on multiple scales from a spatial grid over the image.

The local features are afterwards aggregated in a fixed length vector representing the entire image. The Bag of Features(BoF), inspired by text processing techniques, clusters features from many images to C clusters and models the frequency of assignments of the features in each image to one of cluster centers. This gives rise to a sparse C dimensional vector for each image regardless of the dimensionality of the features themselves. Recently, the Vector of Locally Aggregated Descriptors (VLAD) [6] was introduced that aggregates all the feature vectors assigned to the same cluster center to reach a vector of the same dimension as the visual words and performs the same for all cluster centers. This leads to a dense dC dimensional feature vector where d is the dimension of the local features.

We use the fast and efficient fixed length representation of the image to find the nearest neighbors of each frame of a query sequence in reference sequences. We will compare the performance of the VLAD and BoF aggregation methods on interest region based and dense sampling of SIFT features in our dataset in section 4.1.

2.2 Geometric similarity

The appearance features, described in the previous subsection, highlight pairs of frames which contain the same local structures. However, they do not guarantee that the matched local structures occur in a geometrically consistent way. The features can be considered as geometrically consistent if there is a global transformation or there are certain constraints are fulfilled between the matched features' locations encoding the relative position and orientation of the camera viewpoints. The tried and tested way to check this, especially when one may encounter large displacements and rotations between the views, is via epipolar geometry and estimation of the Fundamental matrix [8].

Thus we estimate the epipolar geometry between two views. Our measure of similarity is then defined as the percentage of inliers, with respect to the estimated fundamental matrix, in an initial set of putative matches. It should be noted we use this measure of similarity between two frames as an absolute score in $[0, 1]$, not a means for re-ranking [10], which is independent of the other images.

Estimating epipolar geometry robustly and efficiently

The images we capture are of dynamic environments and from a moving, twisting platform. Therefore we frequently have to match views with significant amounts of occlusion and significantly different viewpoints. We thus estimate the fundamental matrix from a sparse set of noisy correspondences and robust estimation via a RANSAC variant.

Unfortunately RANSAC based methods require an exponential number of trials in the minimum number of points required to fit the model and worse than exponential trials in the ratio of outliers to inliers. Given the large amount of data we have to process, a careful implementation w.r.t. the computational demands is required. Therefore we

- use Prosac [2] as it provides a significant speed up on RANSAC in the presence of a large number of outliers but where some inliers can be readily identified,
- reduce the minimum number of correspondences required to estimate the fundamental matrix from the standard 7 [4] to 5 by using the method suggested in [11] (though it does give up to 10 solutions),
- reduce the number of false correspondences in the initial putative set by choosing distinctive correspondences. As suggested in [7], we compute for each feature in one view the ratio of the Euclidean distance to its nearest neighbor and second nearest neighbor in the other view. These scores are sorted into ascending order and the first 250 features and its nearest neighbor match, w.r.t. this ordering, make up the putative set.

Another issue which has to be addressed is that the epipolar constraint is relatively weak (it maps a point in one view to a line in the other). To accurately judge the correctness of a hypothesized fundamental matrix in the presence of many incorrect correspondences additional constraints are needed. To this end we enforce that inliers must also be consistent with a homography mapping the local feature locations from one frame to the other. This homography consistency constraint is only weakly enforced and is achieved by using Prosac with a loose definition of inlier to robustly estimate a homography. Then only the matches which are consistent with this estimated homography are maintained and used for the fundamental matrix estimation. Algorithm 1 summarizes the complete implementation and the second row of figure 3 depicts the stages of the fundamental matrix estimation.

Algorithm 1 Computation of geometric similarity.

```
  **INPUT**: Features $\mathcal{F}_1, \mathcal{F}_2$ extracted from images $I_1, I_2$
  **OUTPUT**: Similarity measure $F_{GV}(I_1, I_2) \in [0, 1]$
  $P \leftarrow N$ best putative matches between $\mathcal{F}_1$ and $\mathcal{F}_2$
  $H_L \leftarrow$ PROSAC 4 points loose Homography($P$)
  $P_H \leftarrow$ inliers of $P$ to $H_L$
  $E \leftarrow$ PROSAC 5 point Essential Matrix($P_H$)
  $P_{HE} \leftarrow$ inliers of $P_H$ to $E$
  $F_{GV}(I_1, I_2) \leftarrow f_s(P_{HE}, P)$
```

The final geometric similarity measure

Once the fundamental matrix has been estimated and used to define a set of final point correspondences between the two views, we can calculate the geometric similarity score. In this work we define this as

$$f_s = \min\left(1, \alpha \max\left(0, \frac{|P_{HE}|}{|P|} - \beta\right)\right) \tag{1}$$

where $|P|$ is the number of correspondences in the initial putative set and $|P_{HE}|$ is the number of final inliers found. The α and β are non-negative scalars which are learnt from training data. The role of β is to force the average matching score towards 0 for images which contain no overlap, while α scales the score with the aim that when images of the same scene are matched they achieve a score of around 1.

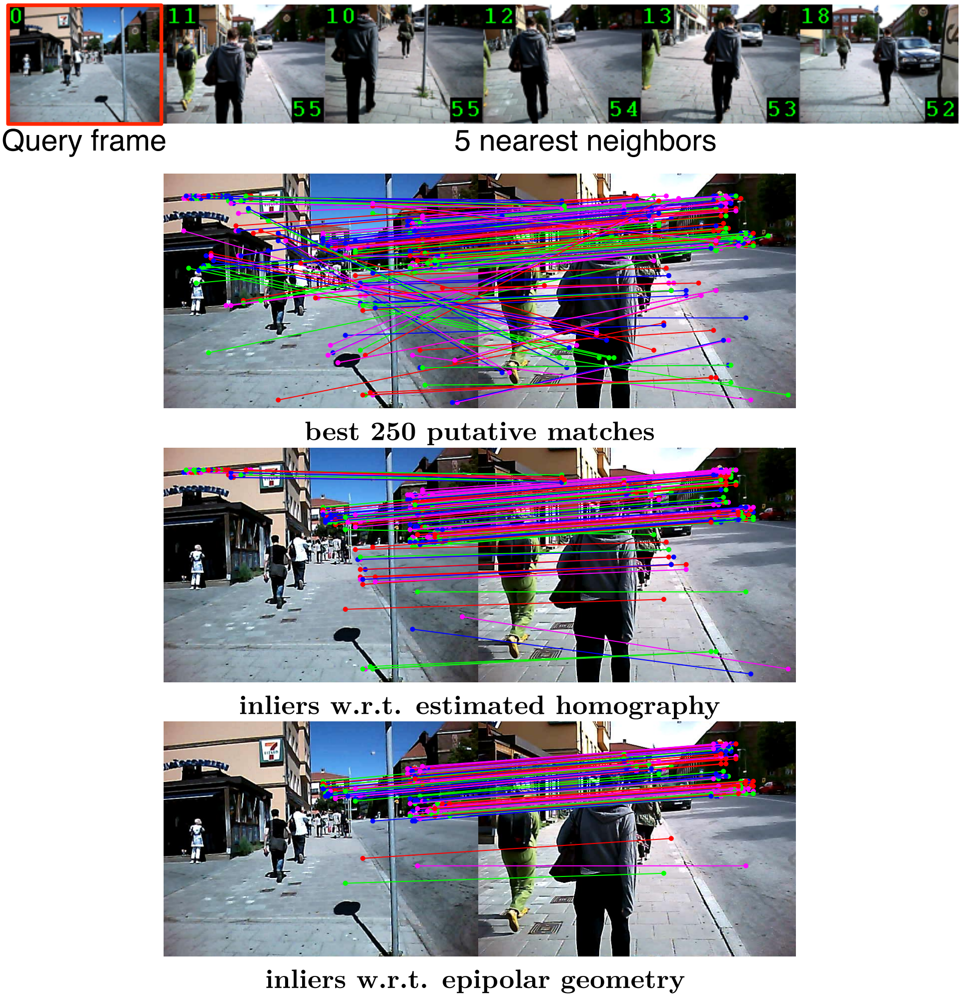
2.3 Dynamic time warping

Once one can measure similarity between two frames, using our geometric similarity measure, the temporal alignment of sequences is straightforward. There are just a couple of steps involved. First the similarity matrix containing the similarity between any pairwise frames is formed and turned into a cost matrix by mapping the similarities to costs using a zero-mean Gaussian with standard deviation σ_c. Then temporal alignment is calculated via dynamic time warping on the cost matrix. Computing alignment in this fashion though straightforward is extremely slow as evaluating each entry in the cost matrix requires calculating the computationally expensive geometric similarity score. Clearly, it is not necessary to compute every entry, we just need to compute those which will have low costs.

These low cost entries can be easily identified, similar to [10], by utilizing the fast and efficient nearest neighbor search using the previously described appearance based fixed length representation, of the frames to find the k nearest neighbors in s_2 of each frame in s_1. Evaluation of the geometric similarity is then limited to k evaluations for each frame in s_1. As the same local features are used in the fixed length representation and in the geometric similarity evaluation, we expect the relevant low cost entries to be computed while ignoring the high cost entries. Figure 4 shows for one particular alignment example what proportion of geometric scores from the full matrix are actually computed and how the entries on the ground truth alignment path have been identified by the k nearest neighbor search.

The minimum cost path connecting the first and last entry of the cost matrix is denoted by a set of ordered pairs $\delta_{s_1,s_2} = \{(i_1, j_1), \ldots, (i_L, j_L)\}$ with $i_1 \leq i_2 \leq \cdots \leq i_L$ and similarly for the j's. We then define the *match cost* of a frame i in sequence s_1 to sequence s_2 as

$$\lambda(i, \delta_{s_1,s_2}) = \begin{cases} C_{i_k,j_k} & \text{if } \exists\ (i_k, j_k) \in \delta_{s_1,s_2} \text{ s.t. } i == i_k, \\ & i_k - i_{k-1} = 1 \ \textbf{ and } j_k - j_{k-1} = 1 \\ 1 & \text{otherwise} \end{cases} \tag{2}$$



best 250 putative matches

inliers w.r.t. estimated homography

inliers w.r.t. epipolar geometry

Figure 3: The top row shows the 5 nearest neighbors in a reference sequence to the query frame. The bottom rows show the stages taken to establishing epipolar geometry between a query frame and a nearest neighbor. The initial correspondences are successively filtered by a robustly estimated homography and then the estimated epipolar geometry.

where C_{i_k,j_k} is the value of the cost matrix at entry (i_k, j_k). Note the defined *match term* is unique for each frame due to the form of the path returned by dynamic time warping.

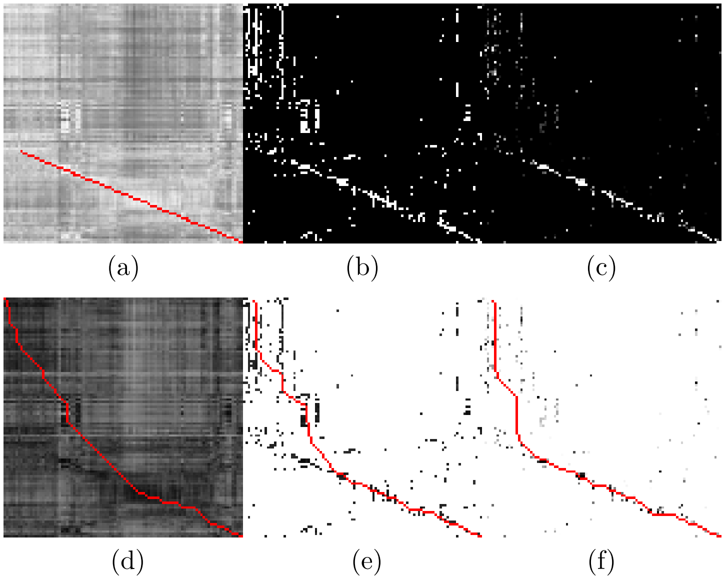


Figure 4: The similarity matrices calculated affect the ability to successfully align a sequence s_1 with another sequence s_2. **Top row:** (a) The full appearance similarity matrix and the ground truth registration between the two sequences is overlayed in red. (b) Sparse sampling of the appearance similarity matrix, using the 5 nearest neighbor per query frame (c) Sparse geometric similarity matrix, the geometric similarity is computed at non-zero entries of the b) matrix. **Bottom row:** The results of DTW applied to (d) dense appearance based cost matrix, (e) sparse appearance based cost matrix (f) sparse geometric similarity based cost matrix. Note how the final registration is closest to the ground truth.

3 Novelty Detection

Once sequences can be aligned and correspondences can be established between their frames, then the quality of the alignments can be used for novelty detection. The crucial point is that novelties induce poor quality alignments. We therefore align a query sequence with the training sequences and search for frames within the test sequence which do not have good correspondences in any or very few other sequences. Figure 1 illustrates such a situation.

Having aligned all sequences to a query sequence, for each frame of the sequence we compute the minimum match cost for each frame of the query sequence:

$$E(s_t^{(i)}) = \min_{s_r \in S} \lambda(i, \delta_{s_q,s_r}) \tag{3}$$

where S represents the set containing the reference sequences. If a frame has a good correspondence in at least one of the reference sequences, the entity $E(s_q^{(i)})$ will have a small value, otherwise it will have a bigger value close to 1. Therefore, we can directly threshold the *minimum match cost* to find novelties. A temporal smoothing of the minimum match cost E is applied prior to thresholding to reduce the effect to the multifarious sources of noise. We smooth the E's with a Gaussian mask with $\sigma_N = 2$ and then threshold them with $\theta_N = e^{-\frac{1}{2^3 \sigma_c^2}}$ to detect novelties. This threshold is chosen as corresponds to a cost associated with a geometric similarity of 0.5.

4 Evaluation of the similarity matching

In this section we evaluate the quality of the performance of the constituent parts of the algorithm to compute the similarity between frames - the nearest neighbor search based on matching appearance and the geometric similarity scoring. It is crucial that these attain a certain level of performance to ensure that sequences can be registered in the presence of non-interesting variations. To help us do this we have manually annotated all sequences with a total of 9 different labels representing the location each frame of each sequences belongs to.

4.1 Nearest Neighbor search

The nearest neighbor search based on appearance features plays a critical role in creating the appropriate sparse cost matrix. Therefore we want to optimize its design and quantify its performance. There are numerous possible choices for the exact form of the features used and how they are compared as expounded in section 2.1. We limit, influenced by recent literature, our investigations to

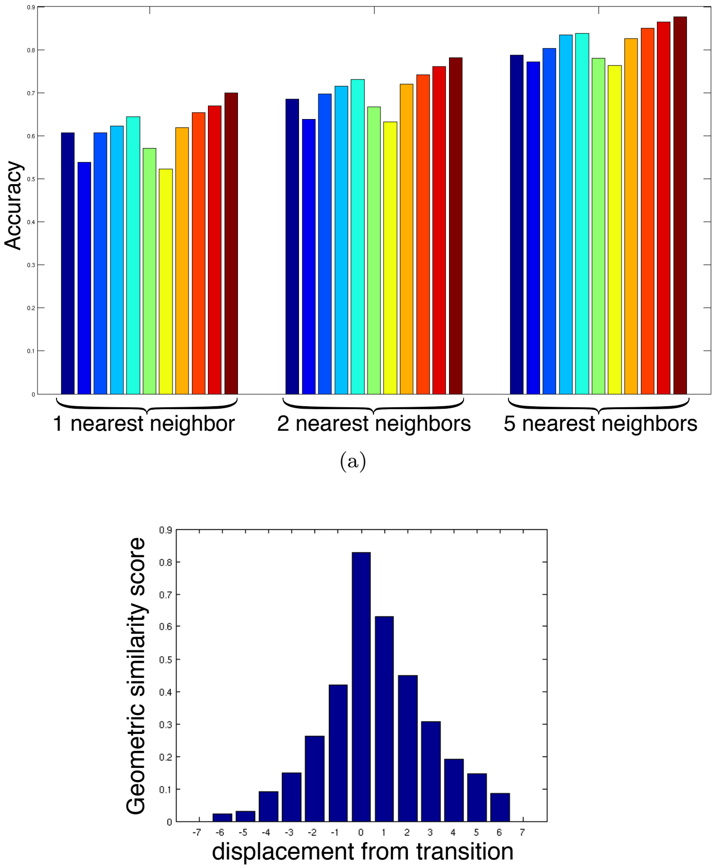
- fixed length vector representations of the image with either BoF or VLAD descriptors built from SIFT features,
- the standard set of interest region detectors, see figure 5(a), including a dense sampling[2].

Similarity between two images is then computed with the minimum intersection kernel for the BoF vectors and a polynomial kernel of degree one to compare VLAD vectors. When both representations are used, we use linear combination of the kernels with equal weights.

We then compare the label of a *query frame* with that of its K nearest neighbors and compute the proportion of the retrievals over the data set which return at least one correct label. Figure 5(a) shows the results of this experiment as the number of nearest neighbors returned and the image feature design varies. It can be observed that the dense sampling outperforms more specific interest region detection.

Guided by these results, we use the combination of the BoF and VLAD vectors with the color and gray variation in the final system. With this method, 88% of

[2] We use the implementation of dense SIFT features [13] with 4 scales and skip parameter of 6 pixels.



(a)

(b)

Figure 5: **(a)** The accuracy of image matching for differing interest region detectors and numbers of nearest neighbours. Methods (from left to right): VLAD+HessianAffine, VLAD+MSER, VLAD+HarrisAffine, VLAD+Dense(gray), VLAD+Dense(color), BoF+HessianAffine, BoF+MSER, BoF+HarrisAffine, BoF+Dense(gray), BoF+Dense(color), VLAD+BoF+Dense(gray+color). **(b)** The average of 100 F_{GV} values on local windows around the true correspondences.

the time at least one of the 5 nearest neighbors to the query frame will correspond to a high similarity entry in the final cost matrix.

4.2 Geometric Similarity

There are many parameters that affect the performance of the geometric similarity function such as the number of fixed initial putative matches N, the thresholds θ_H and θ_E on the reprojection error for the estimated homography and essential matrix used to define inliers and the number of PROSAC iterations T_H and T_E used in estimating the homography and essential matrices. Although it is possible to find the configuration of the parameters by exhaustive search, such an approach would be extremely computationally expensive. Instead, we fixed the parameters and structure of F_{GV} empirically: we used $N = 250$, $\theta_H = 1$, $\theta_E = 0.01$, $T_H = 100$ and $T_E = 25$.

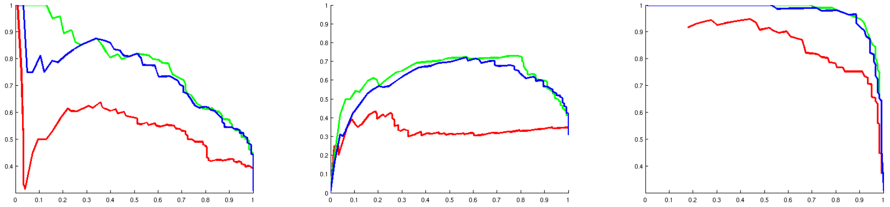
We evaluated the performance of the geometric similarity function using the dense sampling of the SIFT features and interest region detectors and found the dense sampling approach to perform better in terms of robustness and accuracy. This happens as 1) too many interest regions are found around the dynamic objects in the scene and these do not have a correspondence in the other frame and 2) too few interest regions are found in many regions which do not contain strong texture/gradients *e.g.* the the pavement in a relatively low resolution image. In these cases, it is no surprise that dense sampling approach can better capture information from the entire image.

The F_{GV} scores of a frame at a label transition matched to each frame in a local time window around a label transition to the same label as our target frame are computed and recorded. This process is repeated for all such transition frames and time windows. Figure 5(b) depicts the average result of this computation. On average F_{GV} maps the correct correspondence (the transition point)to a number close to 1 while its value drops monotonically relatively quickly with the displacement from the transition point. The appearance based fixed length representations would have a much slower drop and would not be able to precisly locate the label transitions as precisely or unambiguously.

5 Results

For the experiments in this paper, we used a data set of 31 sequences of the subject walking from metro station to work. In addition to the labelings mentioned earlier, we also manually defined temporal segments of the sequences in which something happened that either did not happen in the other sequences or it was infrequent *e.g.* subject meeting with a friend. The labelings resulted to 4 of the 31 sequences containing novel segments. Below, we present the results of the suggested algorithm trying to detect these 4 temporal segments.

Figure 7 depicts the intermediate and final results of novelty detection for a sequence containing novelty(the subject meets a friend). Due to limited space, we show the final picture containing 15 samples of the sequence(Figure 7(a)) vs 6 reference sequences. It can be observed that the method is able to detect both segments that were manually labeled as novel segments in addition to one false



(a) (b) (c)

Figure 6: Precision-recall curves for novelty detection. Each figure uses a different cost matrix: (a) dense appearance , (b) sparse appearance, (c) sparse geometric. The red, green and blue curves show when 1, 6 and 10 reference sequences are used.

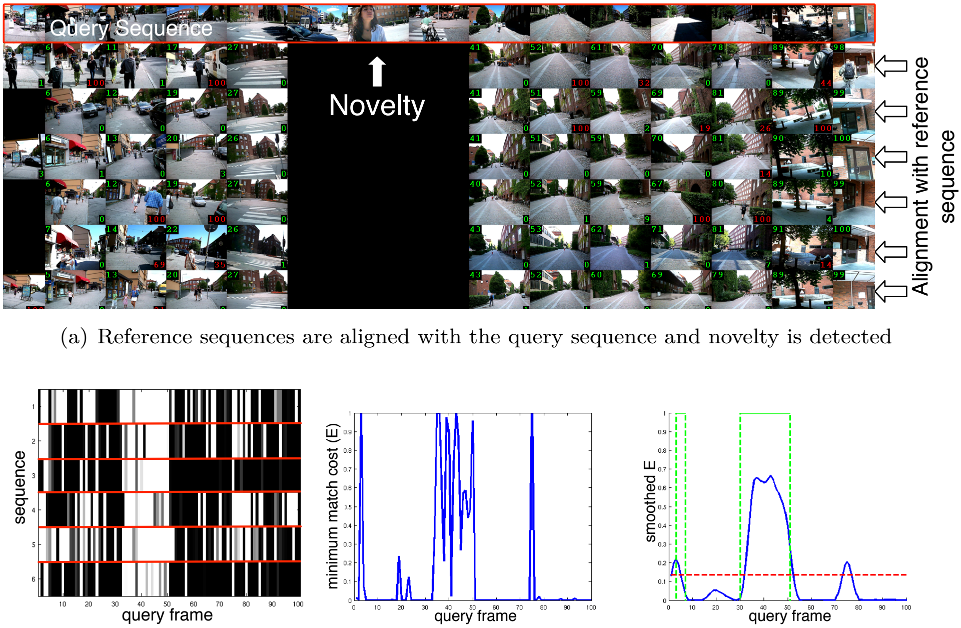
detection of a segment containing 4 frames(Figure 7(d)). The false detection is due to a very strong change in the lighting leading to a few overly bright frames; this inevitably leads to significant changes in local features which then prevents the algorithm to establish correct correspondences for those frames. Figure 8) depicts the results of novelty detection on the remaining 3 sequences that contain novel segments.

The accuracy of the novelty detection on 400 frames(4 sequences sub sampled to contain 100 frames) for which we had the ground truth manually labeled, are measured and depicted in figure 6. It can be observed that using dense the appearance costs leads to better accuracy compared to its sparse version. The figure also suggests that using the method with geometric costs outperforms the use of the appearance based costs with a strong margin. The high average precision of the results using geometric costs with as few as 6 reference sequences($AP \geq 0.96$)(green and blue curves in Figure 6(c)), suggests that the method is accurate and reliable for the purpose of novelty detection while being robust to various environmental changes such as view point and illuminations changes as well as occlusions.

6 Conclusions and Future Work

We have demonstrated a system that is able to automatically extract novel events in the context of video captured from a camera continuously worn by a person who repeats a daily activity. The sequences manually annotated contain (subjectively) a total of four different novel events. All these novelties were automatically detected without any false positives. As far as we know this is the first systematic study of novelty detection of this kind where a repeated activity is used as background. These results indicate that potentially interesting applications of automatic memory selections should be possible especially in constrained environments like the kind considered here.

The frame-to-frame registration of the video captured from one day to another



(a) Reference sequences are aligned with the query sequence and novelty is detected

(b) (c) (d)

Figure 7: A detected novelty - *the subject meets a friend.* (a) The query frames without correspondences in the reference set, the black images below, are detected as novelties. Due to sub sampling only 3 of the 23 detected novelty frames are shown. (b) The *match cost* (λ) between each frame of the query sequence and the reference sequences it has been aligned to. Darker values correspond to lower costs. (c) The *minimum match cost* (E). (d) The smoothed *minimum match cost*. The red line shows the automatic threshold θ_N and the green curve the ground truth labeling of novelty. The large peak corresponds to the novelty displayed in figure (a).

is possible, just using appearance and geometric cues, as we have constrained the variation in these sequences to those experienced by human wearer. This makes it possible to define a background relative to which novelty is measured.

In the future, we want to consider longer individual sequences captured over longer time periods. These will encompass many more activities in differing environments and will undoubtedly require a more complex description and representation of the captured background. Registration at a more abstract semantic level as opposed to the appearance/geometric level exploited in this paper will be needed. Novelty detection at a semantic level will allow disambiguation between false positives generated by changes in appearance and geometry induced by non-relevant

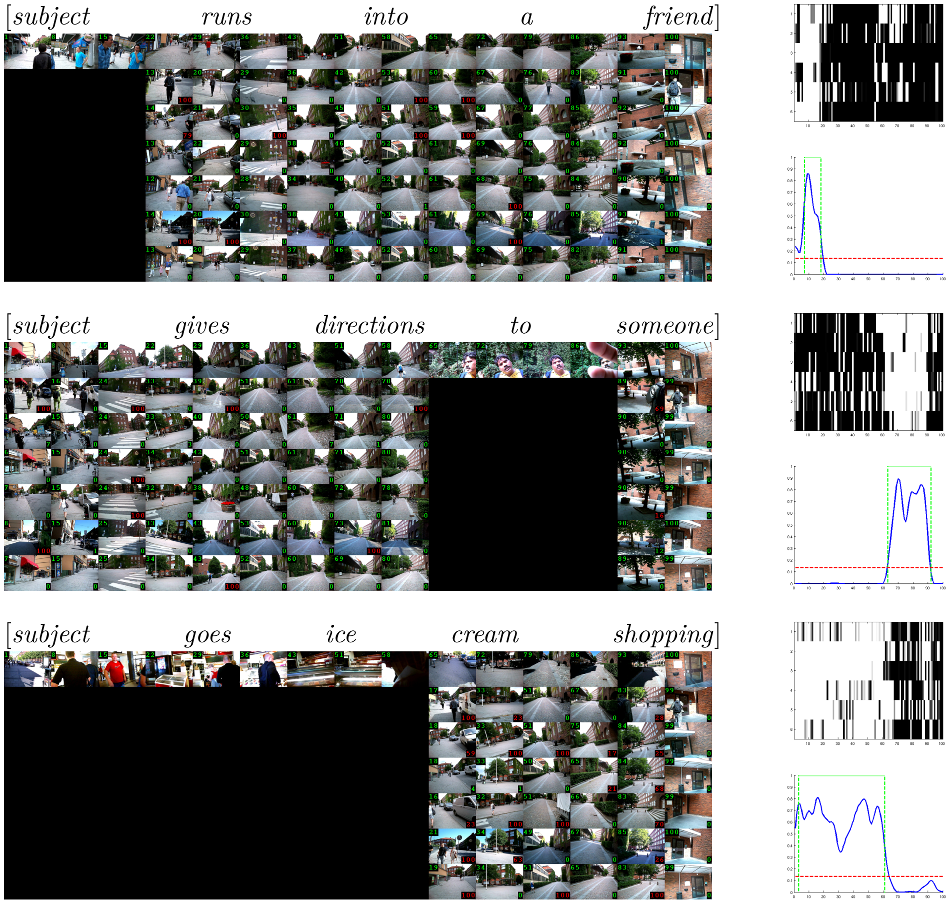


Figure 8: Detected novelties in 3 sequences containing novelty and the corresponding *match costs* and smoothed *minimum match costs* on the right side.

variation of the environment or the ego-motion.

The central problem is the ability to measure similarity of recorded background with the actual captured video. In this sense the problem of novelty detection is intimately related to the general problem of similarity learning and the structuring of visual manifolds. We believe that the analysis of video captured from an ego-centric perspective can serve as an important test case for the study of these problems.

Acknowledgements: This work was supported by The Swedish Foundation for Strategic Research in the project "Wearable Visual Information Systems".

References

[1] Ulf Blanke and Bernt Schiele. Daily routine recognition through activity spotting. In *International Symposium on Location and Context Awareness*, 2009.

[2] Ondrej Chum and Jiri Matas. Matching with prosac " progressive sample consensus. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 220–226, 2005.

[3] Aiden Doherty and Alan F. Smeaton. Automatically augmenting lifelog events using pervasively generated content from millions of people. *Sensors*, pages 1423–1446, 2010.

[4] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004.

[5] Steve Hodges, Lyndsay Williams, Emma Berry, Shahram Izadi, James Srinivasan, Alex Butler, Gavin Smyth, Narinder Kapur, and Ken Wood. Sensecam: A retrospective memory aid. In *Proceedings of the 8th International Conference on Ubicomp*, pages 177–193, 2006.

[6] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[7] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, pages 91–110, 2004.

[8] F. Schaffalitzky and A. Zisserman. Automated scene matching in movies. In *Proceedings of the Challenge of Image and Video Retrieval*, pages 186–197, 2002.

[9] Bernt Schiele, Nicky Kern, and Albrecht Schmidt. Recognizing context for annotating a live life recording. *Personal and Ubiquitous Computing*, page 251–263, 2007.

[10] J. Sivic and A. Zisserman. Video Google: Efficient visual search of videos. In *Toward Category-Level Object Recognition*, pages 127–144. 2006.

[11] Henrik Stewénius, Christopher Engels, and David Nistér. Recent developments on direct relative orientation, 2006.

[12] Tinne Tuytelaars and Krystian Mikolajczyk. *Local Invariant Feature Detectors: A Survey.* 2008.

[13] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms, 2008.

Paper B

Multi View Registration for Novelty/Background Separation

Omid Aghazadeh , Josephine Sullivan and Stefan Carlsson

Published in
IEEE Conference on Computer Vision and Pattern Recognition

The layout has been revised

Multi View Registration for Novelty/Background Separation

Omid Aghazadeh , Josephine Sullivan and Stefan Carlsson

Abstract

We propose a system for the automatic segmentation of novelties from the background in scenarios where multiple images of the same environment are available e.g. obtained by wearable visual cameras. Our method finds the pixels in a query image corresponding to the underlying background environment by comparing it to reference images of the same scene. This is achieved despite the fact that all the images may have different viewpoints , significantly different illumination conditions and contain different objects - cars, people, bicycles, etc. - occluding the background. We estimate the probability of each pixel, in the query image, belonging to the background by computing its appearance inconsistency to the multiple reference images. We then, produce multiple segmentations of the query image using an iterated graph cuts algorithm, initializing from these estimated probabilities and consecutively combine these segmentations to come up with a final segmentation of the background. Detection of the background in turn highlights the novel pixels. We demonstrate the effectiveness of our approach on a challenging outdoors data set.

1 Introduction

A mobile surveillance system or a person with a wearable camera often moves in a geographically limited environment over extended time periods. The problem of identifying the background pixels of a scene from this environment is an interesting and challenging one especially as the background will vary and be partially occluded by different temporary objects each time it is viewed. However, the ability to perform such background/novelty detection would greatly facilitate visual memory processing of wearable camera footage and the monitoring of areas with mobile surveillance systems. In this paper we focus on wearable camera footage and propose a system for detecting these temporary/novel objects in locations repeatedly visited by a person wearing a camera.

We consider images captured over time periods of days and during this time both substantial nuisance and interesting variations can occur in the environment. Given a query image captured from a specific location on a certain day and reference

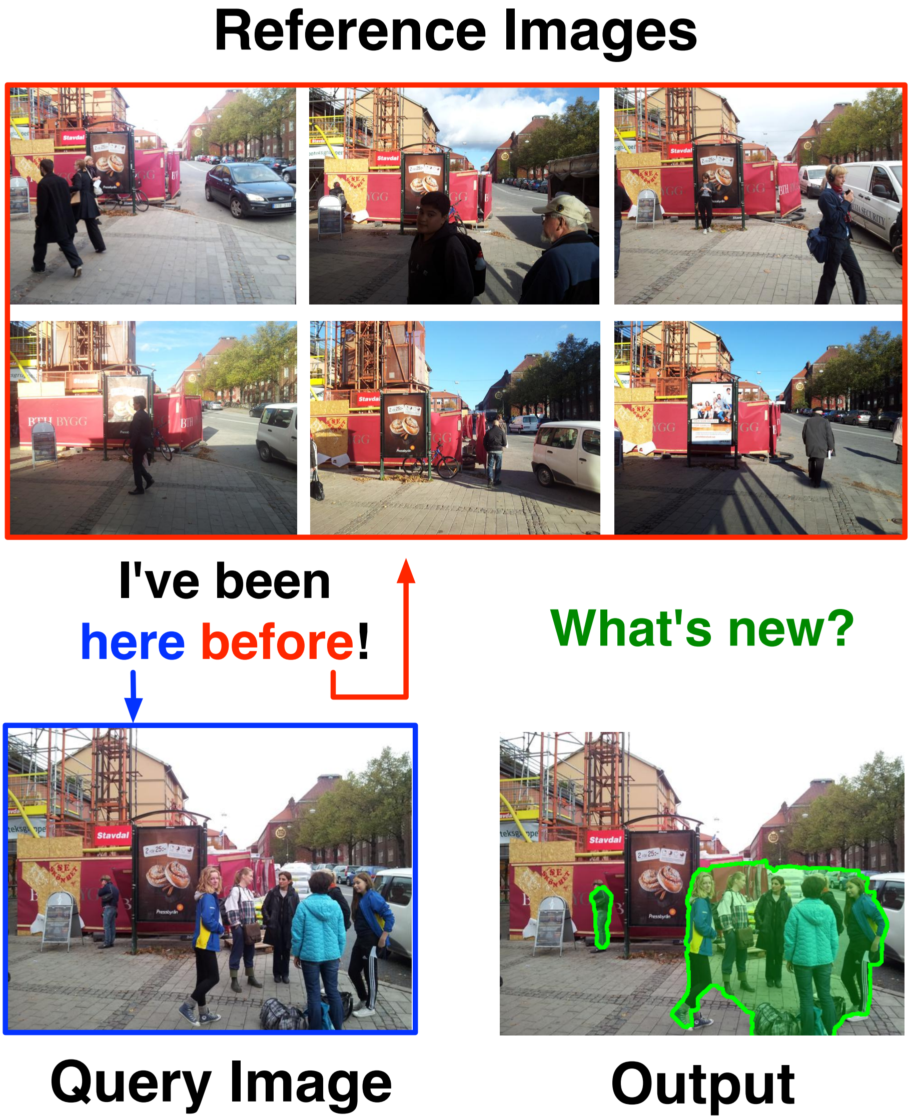


Figure 1: Our system takes as input a query image and multiple reference images. We assume all these images are of the same environment taken from approximately the same view point but at different times. The algorithm segments out objects in the query image which are not part of the environment. The bottom right figure shows the computed segmentation.

images captured from previous days at approximately the same location, we aim to *distinguish between the novel and background pixels in the query image.*

This is not a trivial task. All the images examined are captured on different days and will have a potentially large variation in their illumination and shading in combination with relatively large variations in their viewpoints. And also in each image there will be different temporary objects occluding the background. Therefore, it is not feasible to build one clean background image and perform background subtraction. We instead associate background pixels with those which can be consistently and reliably *matched* to the reference images. Our system has two main steps. The first probabilistically classifies each pixel in query image as background or not from appearance consistency features extracted from dense correspondences to the reference images. While the second stage is two-class segmentation of the query image guided by the output of the background classification and consistency of image appearance. Note that the system implicitly relies on the geometric constraints that the query and stored images are captured from approximately the same location.

This problem differs significantly from traditional problems of foreground background segmentation with stationary surveillance cameras where the main source of background variation is changes in illumination. Since we use static images widely separated in time we cannot exploit camera motion constraints as in [4]. Contrary to [14], our images neither allow 3D modelling of the background nor detailed geometric analysis [17] to be used. Our reliance on appearance matching and two class segmentation allows for robust exploitation of our highly varying background images. The use of multiple static images contrasts with segmentation methods using optical flow [1, 16] and allows us to segment out novelty that is not necessarily foreground with high disparity. Co-segmentation approaches [7], though related, are not suitable due to the significant appearance variations in the background - the object of constant appearance for co-segmentation - across the images.

The main contribution of the paper is a robust and generic novelty detection algorithm whose parameters are automatically learnt from annotated data. This allows for the detection of many time-varying scene components such as people, cars... in a robust way that mimics the performance of specifically designed object detection algorithms.

The organization of the paper is: In section 2 we introduce our method for novelty/background segmentation, in section 3 we quantitatively and qualitatively evaluate the proposed method and we conclude the paper in section 4.

2 Foreground/Background Segmentation

As previously stated we have a set of reference images and a query image taken of the same scene. All these images have been captured at different times and relatively different viewpoints. Our goal is to identify background pixels in the query image. This is achieved by summarizing comparisons of the query image to each reference image as follows:

1. Estimate the probability of each pixel **not** belonging to background which we term the *probability of novelty* from the dense correspondences found between the query image and each reference image.
2. Produce multiple segmentations of the query image, given the probabilities of novelty, by varying the parameter settings of the segmentation process.
3. Combine all the segmentations probabilistically to produce a final classification of the query image pixels.

We now describe each step in more detail.

2.1 Estimating the probability of novelty

Crucial to our algorithm's success is the computation of dense correspondences between the query image and each reference image. Establishing such correspondences, when each image has different parts of the scene occluded, is a hard problem. In fact establishing correspondences and occlusion estimation are closely related tasks - knowledge of the image correspondences makes estimation of the occlusions easier and vice versa.

Some authors have exploited this relationship by explicitly including occlusion estimation into their algorithms for finding image correspondences [12]. As such formulations usually rely on expectation-maximization like procedures, they are usually more susceptible to local minima. Therefore, occlusion estimation is usually ignored and more emphasis is instead put on imposing priors - such as smooth displacement fields - when calculating correspondences.

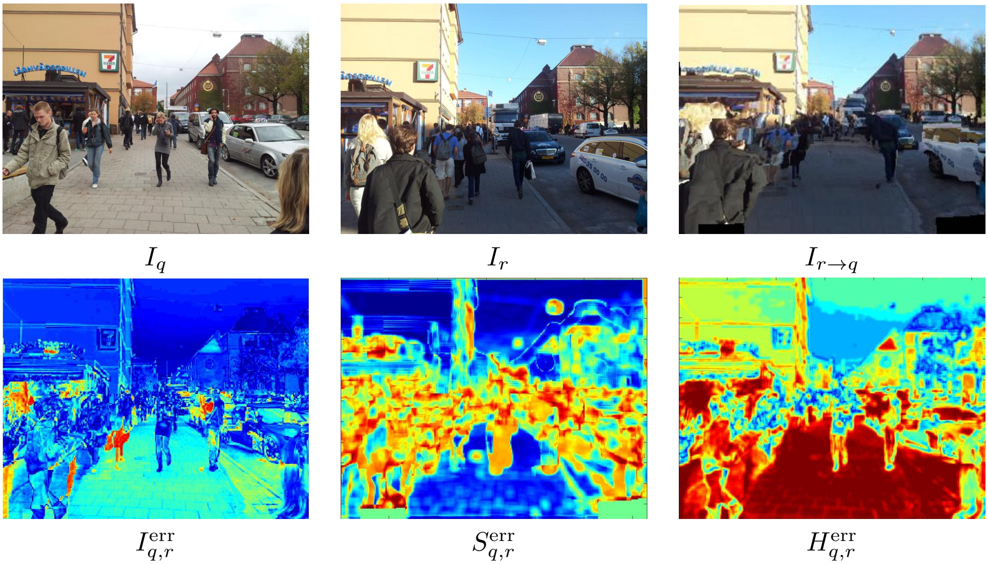
In this work, we do not aim to solve for both occlusions (which in our problem are mainly novelties) and the correspondences simultaneously. Instead, we aim to deduce the background pixels given some noisy correspondences between images. We use *SIFT Flow* [9] to establish such correspondences as we found it more robust to illumination changes, occlusions and large displacements compared to the methods we tried.

We first establish correspondences between the query image I_q and each reference image $I_r \in R$ where R is the set of reference images. Then, we compute the following features on each pixel of I_q using each I_r in turn:

$$
\begin{aligned}
I^{\text{err}}_{q,r,x} &= \|I_{q,x} - I_{r\to q,x}\| \\
S^{\text{err}}_{q,r,x} &= \|S_{q,x} - S_{r\to q,x}\| \\
H^{\text{err}}_{q,r,x} &= \sum_c QC^A_{0.5}\left(H(I_q,x,c), H(I_{r\to q},x,c)\right)
\end{aligned}
\tag{1}
$$

where

- $I_{i,x}$ is the color (CIE Lab) of pixel x in image I_i,



I_q I_r $I_{r\to q}$

$I^{\text{err}}_{q,r}$ $S^{\text{err}}_{q,r}$ $H^{\text{err}}_{q,r}$

Figure 2: The features used to calculate the probability of novelty $\tilde{P}$ for pixels in the query image I_q when compared to a reference image I_r. $I_{r\to q}$ is I_r warped towards I_q using SIFT Flow. The corresponding pixels of I_q and I_r are then compared via $I_{r\to q}$ as in equation (1).

- $S_{i,x}$ is the SIFT [10] computed at pixel x of I_i,
- $H(I_i, x, c)$ is the histogram of channel c intensity values of the pixels inside a rectangular region centered at pixel x in I_i and $QC^A_m(.,.)$ is the distance between two histograms computed using the Quadratic Chi kernel with respect to the parameter m and the similarity matrix A [13],
- $I_{r\to q}$ denotes image I_r warped towards I_q.

The measure H^{err} dubbed *Normalized Bagged Similarity* measures neighborhood similarity of pixels similar to Normalized Cross Correlation while unlike NCC it is invariant to the ordering of the pixels and also, it can be made invariant to nonlinear transformations of the intensities using proper histogram normalization techniques and proper similarity matrices (see supplementary material). NBS can be computed very efficiently by the use of Integral Histograms and its computations can be parallelized very efficiently by the use of GPUs. Table 1 describes the properties of the features and Figure 2 shows the features evaluated on an example case.

We compute these three measurement types at multiple scales and stack the resulting feature vectors into $\bar{F}_{q,r,x}$. We then compute for each pixel x at a fixed scale, the algebraic mean, harmonic mean and minimum of each response in $\bar{F}_{q,r,x}$

		Properties of the Feature		
Feature	**Source**	Neigh.	Corr. Sens.	Illum Inv.
I^{err}	Color	0	1	0
S^{err}	Sift	1	1	1
H^{err}	Hist	1	0	0,1

Table 1: Features used in the estimation of the *probability of novelty* and their properties. **Neigh.** is 1 if the feature captures information in the neighborhood of a pixel. **Corr. Sens.** is 1 if the feature is affected considerably by small errors in the correspondences. **Illum Inv.** is 1 if the feature is invariant to illumination changes. Different normalizations of H^{err} can make it sensitive or invariant to illumination changes.

with respect to the reference images I_r. The resulting feature vector, $F_{q,x}$, for each pixel in I_q is 78 dimensional. This feature vector is used to estimate the probability of novelty as follows.

We use logistic regression to map a pixel's feature vector, $F_{q,x}$, to a scalar between 0 and 1 estimating the pixel's *posterior probability* of being not background. The parameters of this regression function are learnt from our manually annotated ground truth data (see Section 3) which provides many pixel feature vectors and their associated labelling as background or not. L_2 regularization is imposed during learning and LibLinear [6] is used to ensure training takes a reasonable time given the large number of training examples examined (approximately 3 million) which are collected by sub sampling the data every 6th pixel in each direction. In the rest of this paper, we refer to the results of this logistic regression (the *probability of novelty*) evaluated at pixel x in the image i with $\tilde{P}_i(x)$. Figure 4 (top left) depicts a typical evaluation of $\tilde{P}$ on a query image.

2.2 Segmenting out the background

Using the estimated probability of novelty $\tilde{P}$, we iterate between segmentation of the query image's pixels into background and novel regions and updating our models describing the features associated with the background and novel pixels. We do this in a manner similar to Grab Cut [15]. An important difference, though, is that we initialize our foreground and background models automatically from the probability maps indicated by $\tilde{P}$. This iterative process can be viewed as a variant of Expectation Maximization.

For the maximization step, we use an energy minimization approach to segment the images into novelty and background regions. We use Graph cuts [8, 2, 3] to perform the minimization as we use an appropriate energy function in the popular form of a sum of unary and pairwise terms.

$$E(l) = \sum_{x \in \mathcal{X}} D_x(l_x) + \lambda \sum_{(x,y) \in \mathcal{N}} V_{x,y}(l_x, l_y) \tag{2}$$

where l is a binary labelling assigning each pixel $x \in \mathcal{X}$ a label $l_x \in \{0, 1\}$. Here $D_x(l_x)$ is the data term and determines the cost of assigning the label l_x to pixel x in image I. $\mathcal{N}$ is the set of pairs of neighbouring pixels (8 connectivities) and $V_{x,y}(l_x, l_y)$ is the pairwise smoothness (regularization) term and determines the cost of assigning different labels to neighbouring pixels x and y.

A popular choice of the smoothness term is the Ising prior weighted by some dissimilarity measure to relax the smoothness constraint at image discontinuities. We utilize a similar approach and use a parallelized version [5] of the gPb detector [11] - which utilizes GPUs to estimate the boundaries of objects in natural images - to encourage the cut to go through those boundaries. Therefore, our pairwise term is

$$V_{x,y}(l_x, l_y) = \frac{1}{\|x - y\|} \, [l_x \neq l_y] \, e^{-\frac{|I_B(x) - I_B(y)|^2}{2\sigma^2}} \tag{3}$$

where [.] is the Iverson bracket and $I_B(x)$ denotes the response of the gPb detector at pixel x.

We define the data term to be

$$D_x(l_x) = -\log P(l_x \,|\, f_x) \tag{4}$$

where f_x is a feature descriptor of the pixel x and $P(l_x \,|\, f_x)$ represents the posterior probability of label l_x (novelty or background) conditioned on observing feature f_x. More details of how we estimate this posterior probability are now given.

Let $l^{(t)}$ represent the current best estimate of the pixel labellings. Define $\mathcal{X}_k^{(t)} = \{x \,|\, l_x^{(t)} = k\}$ for $k \in \{0, 1\}$ to be the set of pixels with label k according to labelling $l^{(t)}$. For the expectation step, we collect some statistics about the distribution of some features in I_q conditioned on the current estimate of the segmentation[1]. The features we use for the segmentation are (a subset of) the color, a dimensionality reduced version of the sift feature vector (to 3 dimensions) and the position of each pixel. We use *Kernel Density Estimation* to estimate the likelihood $P(f_x \,|\, l_x)$

$$P(f_x \,|\, l_x) = \frac{1}{|\mathcal{X}_{l_x}^{(t)}| \, h^d} \sum_{y \in \mathcal{X}_{l_x}^{(t)}} K\left(\frac{f_x - f_y}{h}\right) \tag{5}$$

where d is the dimensionality of the f_x (8 in case of all 3 features), h is the bandwidth(window width) and $K(\mu)$ is the multivariate Gaussian density function with

[1] In the first iteration, we collect statistics only from the pixels whose $\tilde{P}$ is more than a desired margin m_0 away from 0.5. This way, we can collect the initial statistics about segments with the desired level of certainty and avoid collecting statistics from uncertain regions if $m_0 > 0$.

Prior name	$P(l_x = 1) \propto$	$P(l_x = 0) \propto$
P_H	$\sum_x \tilde{P}(x)$	$\sum_x (1 - \tilde{P}(x))$
P_{SF}	$\tilde{P}(x)$	1
P_S	$\tilde{P}(x)$	$1 - \tilde{P}(x)$

Table 2: The three types of priors used for the labelling of a pixel.

identity covariance matrix evaluated at μ. Whitening the feature data is performed before any likelihood computations are made. We sub-sample the pixels on a fixed grid, evaluate a homogeneous KDE on the same subset of pixels and use bilinear interpolation to estimate the likelihood maps on all pixels. The KDE evaluation is quadratic in the number of (sub sampled) pixels and can be parallelized very efficiently by the use of GPUs. Evaluating the likelihood maps at each iteration takes around 1 second on an NVIDIA GTX 470 for a sub sampling of once every 3 pixels in both directions.

To convert the estimated likelihoods to posteriors, based on $\tilde{P}$, we consider three types of class priors for each pixel: a uniform prior (P_H) and two spatially varying priors (P_{SF} and P_S). Table 2 shows the details of these priors. The prior P_{SF} allows more deviation from the relatively noisy probability estimates in $\tilde{P}$ compared to P_S which strictly promotes the segmentation suggested by $\tilde{P}$. Note the way we define the posterior is different to [1] as we do not marginalize over model parameters but instead use a pixelwise prior computed from $\tilde{P}$. We then, use the negative log of the posterior $P(l_x \mid f_x) \propto P(f_x \mid l_x)P(l_x)$ as the data term:

$$D_x(l_x) = -\log\left(P(f_x \mid l_x)P(l_x)\right) + \log Z_x \tag{6}$$

where the normalization factor is

$$Z_x = \sum_{k \in \{0,1\}} P(f_x \mid l_x = k)P(l_x = k) \tag{7}$$

Figure 3 shows the different segmentations achieved using the different priors P_H, P_{SF} and P_S on the pixel labels. In this example the parameters were set to $m_0 = 0.1, \lambda = 5, h = 0.5$ and each f_x was composed of pixel x's color, dimensionality reduced sift representation and its position. We iterate between the expectation and maximization steps until the solution converges for a maximum of 25 iterations.

2.3 Combining Multiple Segmentations

The segmentation process of the previous section will converge to a stable segmentation. However, the final segmentation achieved will greatly depend on the setting of the explicit and implicit parameters in the energy function defined in equation (2). The explicit parameter corresponds to the regularization parameter λ, while

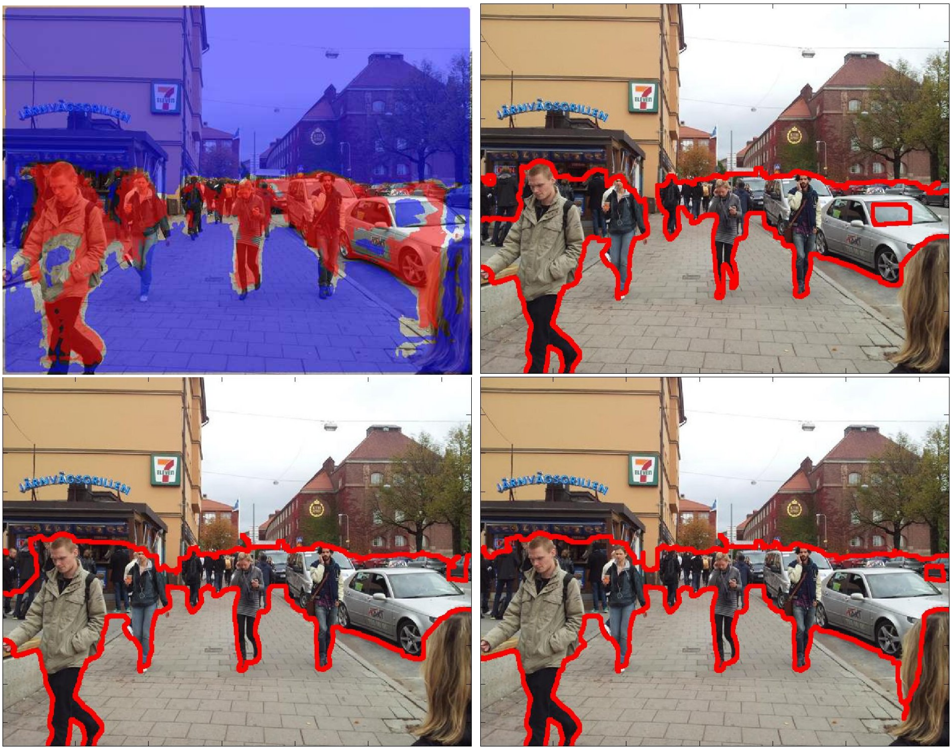


Figure 3: Segmentation results with different class priors: from top left to bottom right: initializing with $m_0 = 0.1$ and segmentation results using P_H, P_{SF} and P_S class priors. In the figure illustrating the initialization, regions inside blue and red boundaries represent initial estimates of background and novelty regions. The margin $m_0 = 0.1 > 0$ on $\tilde{P}$ (refer to Figure 4) leads to gaps between the regions.

the implicit parameters include the initialization margin m_0, the bandwidth of the KDE h in the likelihood function $P(f_x \mid l_x)$, the features extracted to define f_x and the prior used in the calculation of the posterior $P(l_x \mid f_x)$. For clarity let $\mathcal{S} = \{\lambda, h, \ldots\}$ denote the set of all the parameters which influence the segmentation process and $\mathbf{s}$ a vector containing the values assigned to each parameter in $\mathcal{S}$.

The question then is which $\mathbf{s}$ should we use when we segment a new image? We could potentially use the $\mathbf{s}$ which optimizes performance on a validation set. However, the choice made in this way will be highly influenced by the images in the validation set and how performance is measured and also the best parameter setting can vary drastically across individual images. Ideally, we want to perform multiple segmentations, corresponding to $\mathbf{s}_1, \ldots, \mathbf{s}_K$, and aggregate the results. One drawback of this approach is the extra computational cost if K segmentations must be performed and this becomes computationally impractical for a large K. Another issue is how to aggregate the results.

We propose the following solution. We start with a large pool $\{\mathbf{s}_1, \ldots, \mathbf{s}_K\}$ of parameter settings ($K = 50$ in the experiments). Each image in our training set is segmented K times, once for each $\mathbf{s}_k$. Then for a pixel x in a training image we get a binary vector of length K whose kth entry is l_x and l_x is its labelling returned by the segmentation process with parameter setting $\mathbf{s}_k$. We then, learn a logistic regression function with L_1 regularization which maps this binary vector to a probabilistic estimate of its ground truth labelling. The parameter controlling the regularization, in the regression learning, is set to ensure a sparse solution is found. An immediate consequence of this sparse solution is that only a small proportion of the original K segmentations need to be computed when a novel image is encountered. We denote the evaluation of this learnt logistic function on image i at pixel x with $\hat{P}_i(x)$. The top right image of 4 shows an example of a computed $\hat{P}(x)$.

The final segmentation of the query image is found by minimizing an energy function similar to 2 but with the data term based on $\hat{P}$:

$$\hat{D}_x(l_x) = \begin{cases} -\log\left(1 - \hat{P}(x)\right) & \text{if } l_x = 0 \\ -\log\left(\hat{P}(x)\right) & \text{if } l_x = 1 \end{cases} \tag{8}$$

We use Graph Cuts to minimize this energy globally[2]. The bottom left image of Figure 4 shows the final segmentation found for a query image.

3 Experiments

3.1 Data Set

Our data set consists of 12 images of 12 different places making a total of $12 \times 12 = 144$ images. Figure 5 shows 3 images of one of the places from our data set. Note that as the images of the same place were captured on different days, they contain significant (non-linear) changes in lighting conditions - strong shadows and bright regions appear and disappear and occlusions and viewpoints change between images.

The definition of novelty depends on *the memory* we provide the system i.e. which images are used as reference images to detect novelties in a query image. But it also, from the design of our system, depends on the manual annotations we provide for training. However, an accurate annotation is very expensive to obtain manually and is subject to choices made by the annotator. Annotators were not given strict rules but were simply asked to annotate what they thought was not a part of the environment in disjoint subsets of images. They did not consider

[2] This minimization step is not iterated as the data term is fixed.

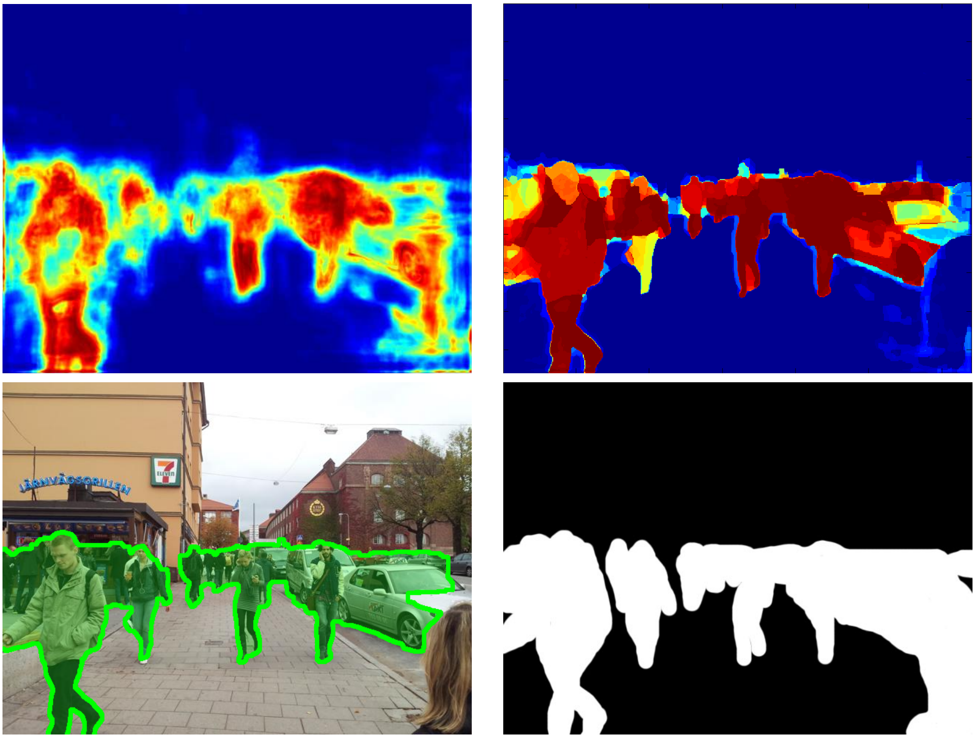


Figure 4: Evaluating the logistic regression function $\hat{P}$ combining multiple segmentation results and the final segmentation acquired from $\hat{P}$. From top left to bottom right: $\tilde{P}$, $\hat{P}$, final segmentation and the ground truth labelling.

what actually changes in the other images in our data set. Therefore, we do not have entirely consistent annotations that strictly follow objective rules: in some annotations, we have strong shadows labeled as novelty while in some cases, some parts of the environment that appear multiple times at the same physical place are labeled as novelty. While it is impossible for any algorithm to agree completely with the ground truth, we expect a reasonable algorithm to statistically agree with the majority of the annotations.

In the following evaluations, we divide our images into training and testing sets, use the training set to fit our models and to cross validate its parameters and we report the results on the testing set.

3.2 Estimating the probability of novelty

Figure 6(a) shows the results of using different combination of features in computing $\tilde{P}$. The beginning capital letters in the figure denote which features are used e.g. `I`

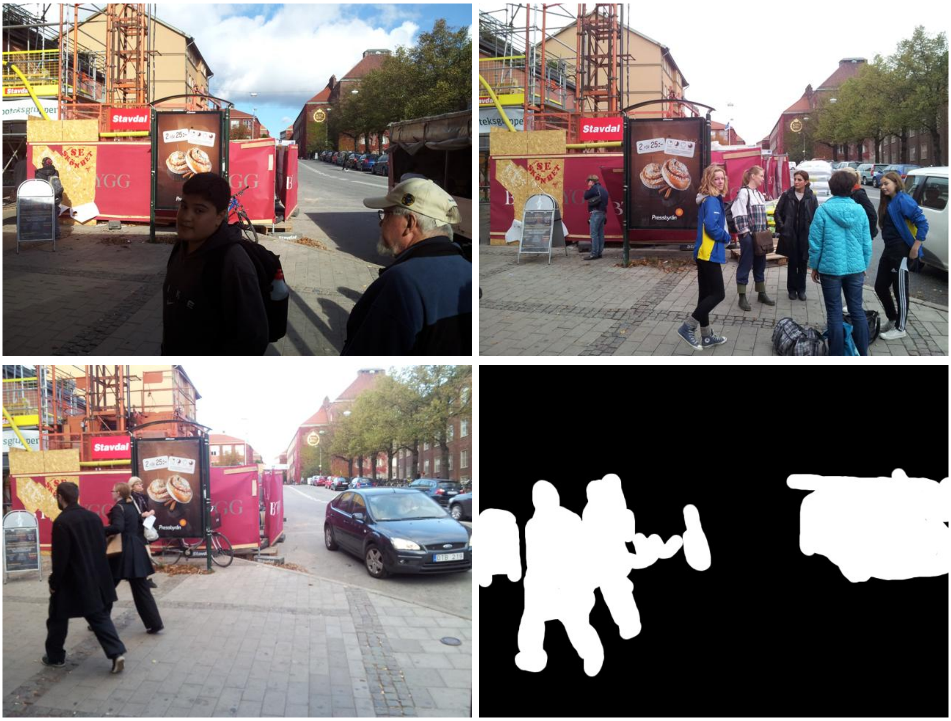
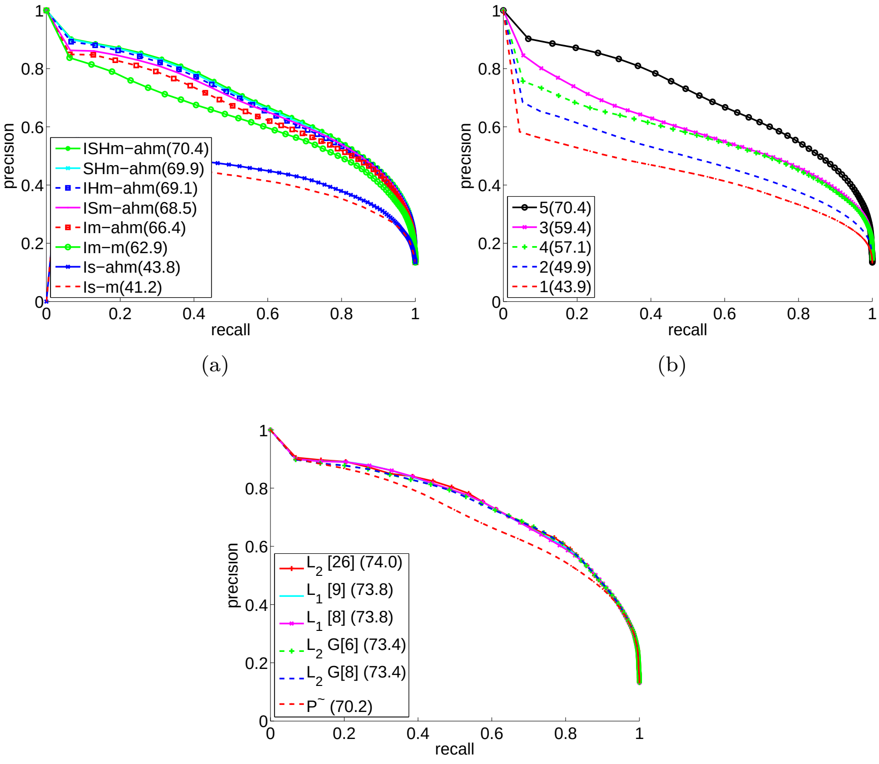


Figure 5: Three images of the same place from our data set and the ground truth labeling of the last image. Note the variation in lighting conditions, strong shadows, occlusions and changes in viewpoints.

denotes the I^{err} measure and `ISH` refers to the combination of I^{err}, S^{err} and H^{err}. The subsequent letter refers to a single scale "`s`" or a multi scale "`m`" version of the mentioned features. The final letters after the "`-`" sign ("`a`", "`h`" and "`m`") refer to the aggregation function applied to different pairwise error measures (the algebraic mean, harmonic mean and the minimum respectively).

It can be observed that by taking the minimum of the most basic measure, I^{err}, over 5 different reference images `Is-m`, the Average Precision (AP) of 41.2 can be achieved. By including more aggregating functions, the harmonic and algebraic means, the AP improves to 43.8 `Is-ahm` while by considering the multi scale version of the same measure `Im-m`, the AP improves considerably to 62.9. Using a multi scale version of the same feature I^{err} with multiple aggregation `Im-ahm` function achieves an AP of 66.8. Therefore, we use multiple aggregations and multi-scale versions of the features in the remaining part of the evaluations.

To evaluate the contribution of each feature, we report the performance measure when the feature is removed from the feature pool: in order to evaluate the



(a)

(b)

(c)

Figure 6: Quantitative evaluation of the individual pixel classifier. The effect of using different features and different combinations of features (a), the effect of using a different number of reference frames (b) and the final probability measure $\hat{P}$ combining multiple segmentations and $\tilde{P}$ using 5 reference frames (c).

	Parameter Settings					
Feature	h	λ	m_0	$P(l_x)$	log	**Acc**
CSP	0.66	10	0.4	P_{SF}	1	**91.86**
CSP	0.5	1	0.4	P_{SF}	0	**91.76**
CSP	0.5	10	0.3	P_{SF}	1	**91.75**
CSP	0.66	10	0.4	P_{SF}	0	**91.72**
CSP	0.5	1	0.3	P_{SF}	0	**91.71**
CSP	0.75	10	0.4	P_{SF}	1	**91.69**
CSP	0.5	0.5	0.3	P_{SF}	0	**91.67**
CSP	0.5	10	0.2	P_{SF}	1	**91.56**
CSP	1	10	0.4	P_{SF}	1	**91.50**
CSP	0.5	5	0.1	P_{SF}	1	**91.43**

Table 3: Evaluation of different parameter settings for the segmentation process. The pixel-wise accuracy of the 10 best performing settings are presented. Compare with the accuracy of thresholding $\tilde{P}$ (the initialization for the segmentations) at 0.5 : 90.64.

contribution of I^{err} measure, we report the performance of `SHm-ahm` and compare it to a logistic regression based on all three measures `ISHm-ahm` with an AP of 70.4. We expect features with more information to have more contribution to the performance of `ISHm-ahm`. Therefore, the results suggest that the H^{err} measure contains more information than the other two: AP of 68.5 for `ISm-ahm` compared to 69.9 for `SHm-ahm` and 69.1 for `IHm-ahm`. For the rest of the evaluations, we use the entire feature pool (78 dimensions) unless stated otherwise.

Figure 6(b) shows the results of using a different number of reference frames to compute $\tilde{P}$. Using only one reference frame (one pairwise comparison) results in an AP of 43.9 while increasing the number of reference frames increases performance. Due to computational issues we do not consider using more than 5 reference frames (AP of 70.4) but the figure suggests that increasing the "memory" of the system i.e. by increasing the number of reference images compared to a query image, the performance of the system increases.

3.3 The Segmentation Method

Table 3 shows quantitative evaluation of the segmentation step using the 10 best performing parameter settings from the 50 we tried where *best* is defined relative to the pixel-wise accuracy measure. From the results the following observations can be made. All the three feature measurement types used in the KDE likelihood computations have a positive role in improving the segmentation. One should avoid using information from uncertain regions ($m_0 > 0$) when initializing the likelihood model and that P_{SF} performs better than the other two priors imposed on the pixel label.

It should be emphasized here that our annotations do not match the data exactly. Large brush strokes were used to manually label the novelties, therefore our annotations over-estimate the extent of the true novelties. Our annotations therefore agree more with smoother and slightly over extended estimations. Therefore, by fitting boundaries of the segmentation to their exact locations, we have probably decreased the accuracy measure compared to a slightly over extended estimation! This probably accounts for the small quantitative improvements in accuracy and AP measures over the estimations achieved by thresholding $\tilde{P}$.

3.4 Combining Multiple Segmentations

Figure 6(c) shows a quantitative evaluation of the combination of multiple segmentations approach. The figure presents the results for combinations of $\tilde{P}$ with different segmentations using different priors. The L_2 [26] refers to an L_2 regularized logistic regression fitted to 25 of the best performing parameters, L_2 G[x] to the greedy selection of x out of the best 8 and L_1 [x] to an automatic feature selection of x features using L_1 regularization.

It can be observed that the suggested approach efficiently combines different segmentations (compare $\tilde{P}$ with the rest) and that L_1 regularization based feature selection outperforms the greedy approach for the same level of sparsity in the solution (compare L_1 [8] and L_1 [9] with L_2 G[8] and L_2 G[6]). In summary, we can achieve more than 3.5 percent increments on the AP measure by combining multiple segmentations. However, the argument we made earlier about the over-extension of the ground truth labelling still holds here and therefore, we believe the true gain to be greater than is reflected in these numbers.

4 Discussions and Conclusions

Figure 7 shows some qualitative results of our method. While most of the results are quite compelling and convincing, some depict the limitations of the method. In particular, as is the case with any correspondence method, large homogeneous regions cause problems as they are ambiguous to register. While our method can overcome incorrect established correspondences to a reasonable extent, the algorithm will have difficulty in detecting novel textureless segments occluding textureless background regions if the wrong correspondences are established consistently across different reference images. This probably accounts for most of the missed novelties.

Although our method is robust to illumination and moderate view point changes, it cannot cope with large changes in the appearance such as strong textures induced by strong shadows. However, as more reference images are added to the system e.g. with the passage of time in wearable systems, scenes will be represented under

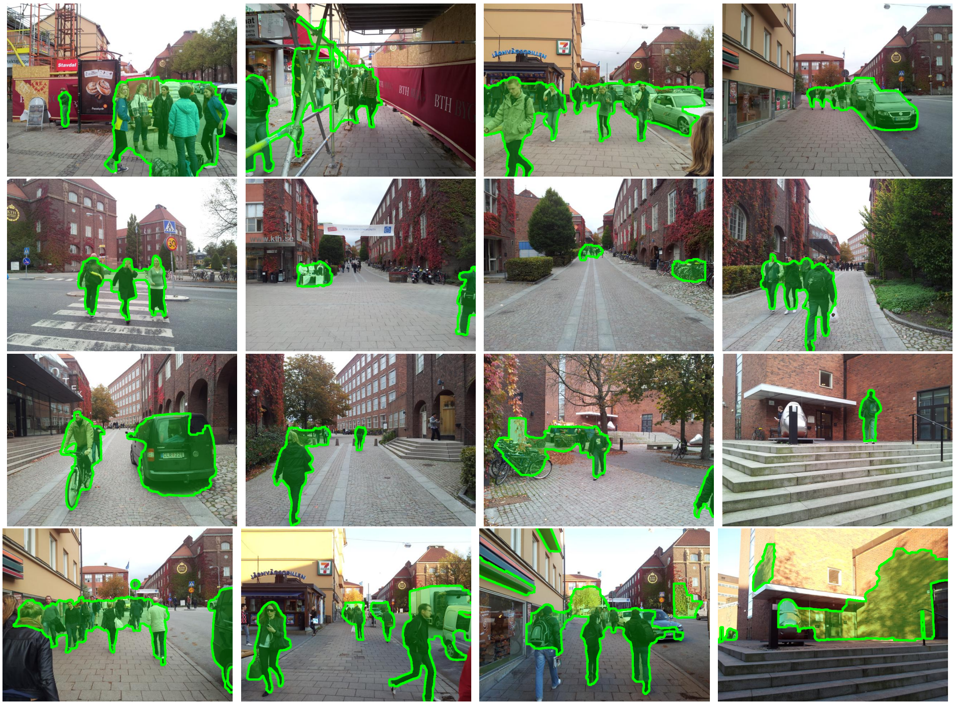


Figure 7: Qualitative results of our algorithm. The first three rows show one result per different place that we have collected data (12 places in total). The last row shows some failure cases where most likely either parts of objects are missed or strong changes in appearance (e.g. strong shadows) are detected as novelties.

various illumination conditions and view points and this issue will become less important. Figure 6 (middle) provides evidence for this argument.

In conclusion, we presented a system which uses multiple images of the same environment captured at different times, viewpoints and lighting conditions to implicitly learn a background model and segment out the novel objects. As for future work, it would be interesting to also consider temporal information and to consider an extra constraint of consistency across different view points. Using such an approach, we would be able to explicitly learn the underlying 3D model and its projection in each view point, which would allow us to make a dense 3D model of the environment and to automatically remove the novelties, and fill them in with the learnt background model.

Acknowledgements: We want to thank Jan-Olof Eklundh for his constructive comments about the segmentation process. This work has been funded by the Swedish Foundation for Strategic Research (SSF); within the project VINST.

References

[1] C. Bibby and I. Reid. Robust real-time visual tracking using pixel-wise posteriors. In *European Conference on Computer Vision*, 2008.

[2] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *PAMI*, 2004.

[3] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *PAMI*, 2001.

[4] A. Bugeaue and P. Perez. Detection and segmentation of moving objects in highly dynamic scenes. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[5] Bryan Catanzaro, B. Su, Narayanan Sundaram, Yunsup Lee, Mark Murphy, and Kurt Keutzer. Efficient, high-quality image contour detection. In *IEEE International Conference on Computer Vision*, 2009.

[6] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, pages 1871–1874, 2008.

[7] Armand Joulin, Francis R. Bach, and Jean Ponce. Discriminative clustering for image co-segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1943–1950, 2010.

[8] Vladimir Kolmogorov and Ramin Zabih. What energy functions can be minimized via graph cuts. *PAMI*, 2004.

[9] Ce Liu, Jenny Yuen, Antonio Torralba, Josef Sivic, and William T. Freeman. Sift flow: Dense correspondence across different scenes. In *European Conference on Computer Vision*, pages 28–42, 2008.

[10] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, pages 91–110, 2004.

[11] Michael Maire, Pablo Arbelaez, Charless C. Fowlkes, and Jitendra Malik. Using contours to detect and localize junctions in natural images. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[12] A. S. Ogale, C. Fermuller, and Y. Aloimonos. Motion segmentation using occlusions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 988–992, 2005.

[13] Ofir Pele and Michael Werman. The quadratic-chi histogram distance family. In *European Conference on Computer Vision*, 2010.

[14] T. Pollard and J. L. Mundy. Change detection in a 3-d world. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[15] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. "GrabCut": interactive foreground extraction using iterated graph cuts. In *SIGGRAPH*, 2004.

[16] P. Sundberg, T. Brox, M. Maire, P. Arbelaez, and J. Malik. Occlusion boundary detection and figure/ground assignment from optical flow. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[17] A. Taneja, L. Ballan, and M. Pollefeys. Image based detection of geometric changes in urban environments. In *IEEE International Conference on Computer Vision*, 2011.

Supplementary Materials

4.1 Data Set

Figure 8 shows 7 examples of one of the places in our data set. Despite the fact that we had more than these number of images per place, in this paper, we did not use all of them: we used 4 images for training/cross validation and 3 for testing purposes. Note that using 4 images for training means that we have $4 \times \binom{6}{5} = 24$ different choices for training using 5 reference images, and similarly, $3 \times \binom{6}{5} = 18$ different choices for the testing purposes per place, which is more than sufficient for training / testing purposes. The main reason for this is the combinatorial costs in increasing the maximum number of images e.g. considering 8 images per place in total and dividing it into 4 training and 4 testing images, we would have had $4 \times \binom{7}{5} = 84$ choices for training and the same for testing - per place. This number increases to $6 \times \binom{12}{5} = 4752$ in case of using all the 12 images and 5 reference images which would have been much more expensive to deal with. For other numbers of reference images we randomly picked the same number of training and testing cases (24, 18) e.g. in case of 3 reference images - from the possible $4 \times \binom{6}{3} = 80$ training cases - we randomly picked 24.

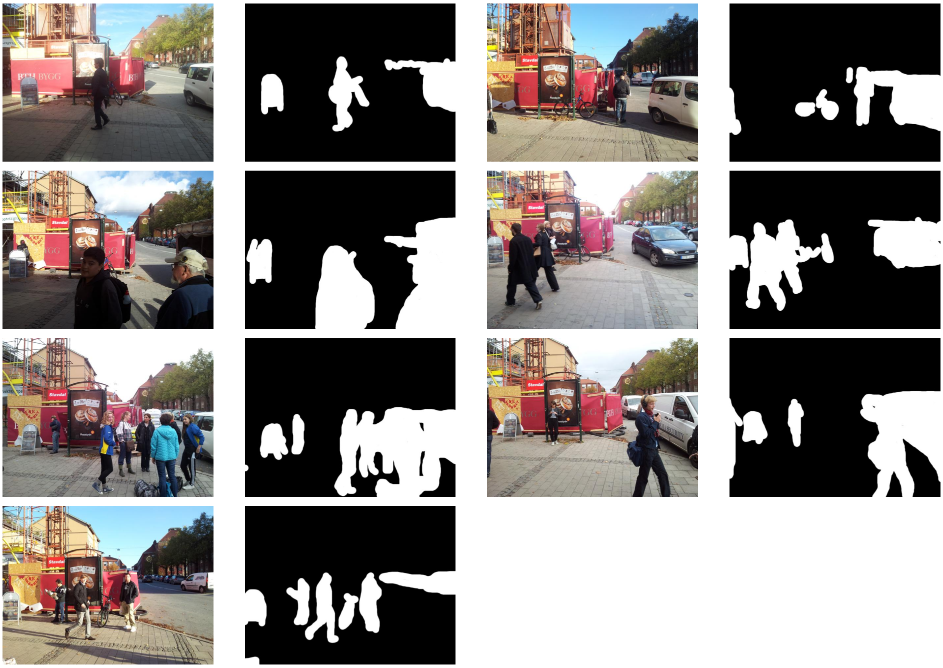


Figure 8: 7 images of one of the places from our data set and the manually defined ground truth for each image

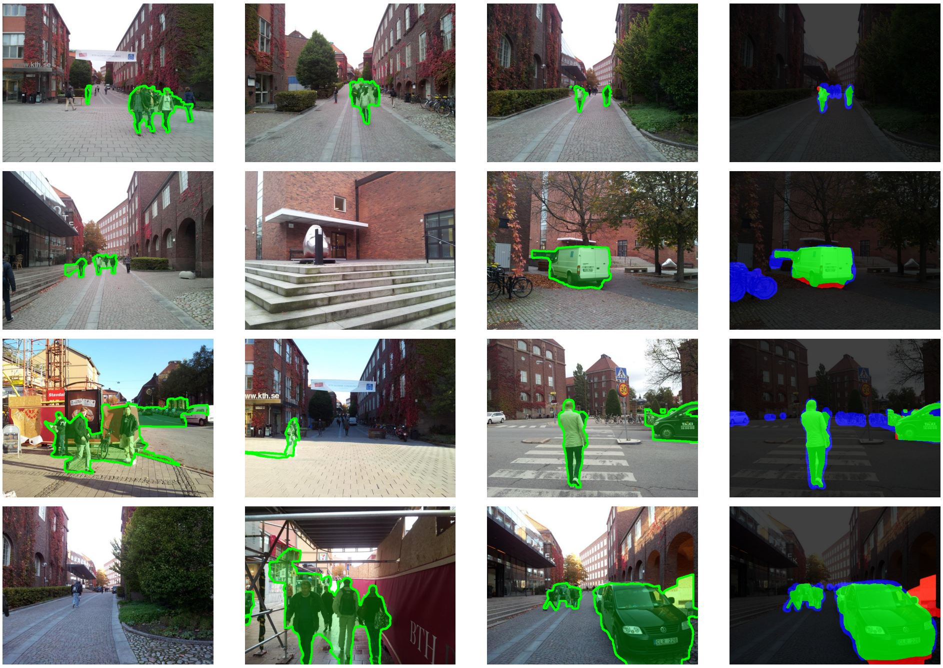


Figure 9: More qualitative results on the first 3 columns. The last column depicts the segmentation of the 3rd column imposed on the ground truth: black and green represent correctly detected background and novelty, and red and blue represent background detected as novelty and novelty detected as background respectively. Note the over extended definition of novelties in our ground truth. Best viewed electronically.

4.2 More Qualitative Results

Figure 9 depicts more qualitative results. The same behavior as in the results in the paper can be observed: In addition to the general appealing behavior of the algorithm, we have some occasional missing novelties and false detections. We expect the results to improve if temporal information is additionally considered or a true multi view registration - which satisfies the projective geometry in all the views simultaneously - is formulated and solved for.

The last column in Figure 9 compares the segmentations in the 3rd column to our manually labelled ground truth. Note the over extension of the manual labellings with respect to the exact boundaries of novelties.

4.3 Technical Details

Here, we clarify the meaning of the "log" column in Table 3 in the paper. It is 1 if the negative log of the posterior was used in the data term of the energy function (similar to Equations 4, 6 and 8 in the paper) and 0 if the posterior itself was used. From the table it is evident that good solutions can be found without the use of the sensitive log operator if proper priors are considered to convert likelihoods to posteriors and if proper bandwidths are used in the likelihood estimations. However, as expected, large bandwidths leading to very smooth likelihoods e.g. $h > 0.66$, require the sensitive log operator to be able to discriminate between the novelties and the background i.e. to guide the segmentation to converge to the desired solutions. As smaller bandwidths can prevent over-smooth likelihood maps, they can be discriminative without relying on the log operator.

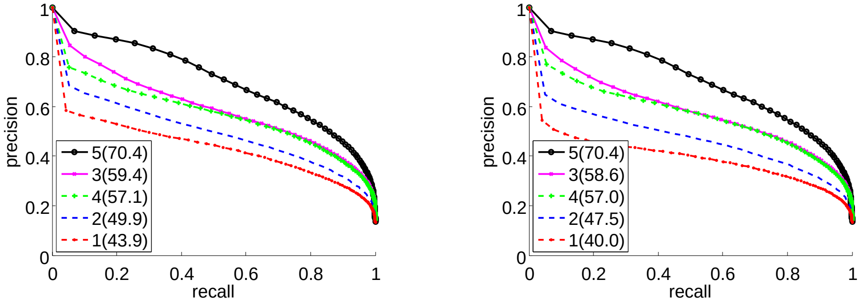
4.3.1 Feature Vectors Used in $\tilde{P}$

Algorithm 2 shows the feature extraction process for $\tilde{P}$.

Algorithm 2 Feature Extraction Algorithm for $\tilde{P}$

```
INPUT: $I_q$, $R = \{I_{r_1}, ..., I_{r_n}\}$, $R_{\rightarrow q} = \{I_{r_1 \rightarrow q}, ..., I_{r_n \rightarrow q}\}$,$\Sigma_a = \{\sigma_{a_1}, ..., \sigma_{a_{na}}\}$, $\Sigma_s = \{\sigma_{s_1}, ..., \sigma_{s_{ns}}\}$, $\sigma_{SF}$
OUTPUT: $F_q$
for $I_r \in R$ do
  $\bar{F}_{q,r} \leftarrow \emptyset$
  for $\sigma_a \in \Sigma_a$ do
    $\bar{F}_{q,r} = \bar{F}_{q,r} \times G_{\sigma_a} * \|S_q^{(\sigma_{SF})} - (S_r^{(\sigma_{SF})})_{\rightarrow q}\|$
    $\bar{F}_{q,r} = \bar{F}_{q,r} \times G_{\sigma_a} * \|I_q - I_{r \rightarrow q}\|$
    for $\sigma_s \in \Sigma_s$ do
      $\bar{F}_{q,r} = \bar{F}_{q,r} \times G_{\sigma_a} * \|S_q^{(\sigma_s)} - S_{r \rightarrow q}^{(\sigma_s)}\|$
    end for
  end for
  for $\sigma_s \in \Sigma_s$ do
    $\bar{F}_{q,r} = \bar{F}_{q,r} \times \sum_c QC_{0.5}^A \left(H_{SI}^{(\sigma_s)}(I_q, ., c), H_{SI}^{(\sigma_s)}(I_{r \rightarrow q}, ., c)\right)$
    $\bar{F}_{q,r} = \bar{F}_{q,r} \times \sum_c QC_{0.5}^A \left(H_{SV}^{(\sigma_s)}(I_q, ., c), H_{SV}^{(\sigma_s)}(I_{r \rightarrow q}, ., c)\right)$
  end for
end for
$F_q^{AM} = \frac{1}{|R|} \sum_r \bar{F}_{q,r}$
$F_q^{HM} = \frac{|R|}{\sum_r \frac{1}{\bar{F}_{q,r}}}$
$F_q^M = \min_r \bar{F}_{q,r}$
$F_q = F_q^{AM} \times F_q^{HM} \times F_q^M$
```

where



(a)

(b)

Figure 10: Evaluation of $\tilde{P}$ when trained on the data using different reference image numbers and tested on the corresponding testing set (left) and (right) when trained using the train data with 5 reference images and tested on testing data with different number of reference images.

- we used $F_1 \times F_2$ to refer to the concatenation of feature vectors F_1 and F_2 ,
- $G_{\sigma_a} * X$ refers to the convolution of X with a Gaussian kernel with standard deviation σ_a,
- σ_{SF} is the scale the SIFT feature vectors in Sift Flow were computed on,
- A is the similarity matrix for Quadratic Chi kernel. We used the following band limited similarity matrix $A_{i,j} = \frac{1}{1+|i-j|}[|i-j| < 4]$ where the [] is the Iverson bracket,
- $H_{SI}^{(s)}$ and $H_{SV}^{(s)}$ denote the shift invariant and shift variant histograms - of intensities inside a square region of size $(2s+1) \times (2s+1)$ - respectively,
- Σ_a and Σ_s define window sizes (standard deviations for Gaussian windows and window length for histogram computations) for spatial aggregation and scale computations respectively,
- S_q^s defines the sift vector on scale s computed on I_q.

We used $\Sigma_a = \{2, 4, 8, 16\}$ and $\Sigma_s = \{2, 4, 8\}$ in the paper which results in $3|\Sigma_a| = 12$ dimensions for I^{err}, $3|\Sigma_a|(1 + |\Sigma_s|) = 48$ dimensions for S^{err} and $3|\Sigma_s|2 = 18$ for H^{err} feature (a total of 78 dimensions). Note the superior performance of the NBS feature (H^{err}) compared to the rest of the Sift based features (S^{err}) despites its lower dimensionality (Figure 6 (left) in the paper).

4.3.2 Normalized Bagged Similarity

The computation of NBS can be made very efficient using Integral Histograms. Normalizing channels between $[0, 1]$ and quantizing each channel into $N = 32$ bins and using linear interpolation, we compute the IH of the image and compute the histogram of a given width centered around a given point by 2 histogram additions and 2 subtractions.

In order to build invariance into NBS, we compute the statistics of the regions on which the histograms are obtained (from the histograms themselves)

$$\begin{array}{lclcl} \mu(H) & \approx & E[i] & = \sum i\,P(i) & \approx \sum_{n=0}^{N-1} \frac{n}{N-1} H(n) \\ \sigma^2(H) & \approx & E[(i-E[i])^2] & = \sum (i-\mu)^2\,P(i) & \approx \sum_{n=0}^{N-1} \left(\frac{n}{N-1} - \mu(H)\right)^2 H(n) \end{array} \tag{9}$$

where $E[i]$ denotes the first moment of the intensities of the pixels inside a region described by the histogram H. To make NBS Shift Invariant, we shift (each bin of) the histogram by the approximated first moment ($\mu(H)$) and interpolate the target - in the re-sampled bin locations from $[-1,1]$ - by linear interpolation. The same approach is used for the affine invariant version but, the target is normalized by the second moment as well and the bins are then re-sampled from $[-3,3]$[3]. However, as the discriminativeness of the measure becomes less as the invariance level increases, we did not include affine invariant version of NBS in the computation of $\tilde{P}$ and instead, we used both Shift Invariant and Shift Variant versions of the NBS in the feature pool. We also experimentally found out that the shift variant version is more discriminative and suits our problem more.

It is also possible to exhaustively search for a shift in one of the histograms that minimizes a distance measure between the two. However, we found such an approach to be computationally more demanding - specially if the distance measure is expensive to evaluate e.g. non diagonal similarity matrices in Quadratic Chi kernels - without any specific advantages.

4.3.3 Estimating $\tilde{P}$

Figure 10(a) shows the results of training the logistic regression function using different number of reference images (the same as Figure 6(b) in the paper) and Figure 10(b) shows the result of the logistic function learnt using the training set with 5 reference images but evaluated on the test sets using different number of reference images. It can be seen that the decrements in the performance gets smaller and smaller when the number of reference images are increased (3.9, 2.4, 0.8, 0.1) which suggests that

- The logistic function being learnt gets more and more independent of the training data as the reference image set size increases.
- The regression process is perhaps converging to an optimal function irrespective of the number of reference images (in training and testing times) as enough data is provided to the method. The figure provides strong support for this idea.

[3] with the assumption of Gaussian distribution of intensities, 3 standard deviation covers 0.997 of the space.

Paper C

Mixture Component Identification and Learning for Visual Recognition

Omid Aghazadeh, Hossein Azizpour, Josephine Sullivan and Stefan Carlsson

Published in
European Conference on Computer Vision

Mixture Component Identification and Learning for Visual Recognition

Omid Aghazadeh, Hossein Azizpour, Josephine Sullivan and Stefan Carlsson

Abstract

The non-linear decision boundary between object and background classes - due to large intra-class variations - needs to be modelled by any classifier wishing to achieve good results. While a mixture of linear classifiers is capable of modelling this non-linearity, learning this mixture from weakly annotated data is non-trivial and is the paper's focus. Our approach is to identify the modes in the distribution of our positive examples by clustering, and to utilize this clustering in a latent SVM formulation to learn the mixture model. The clustering relies on a robust measure of visual similarity which suppresses uninformative clutter by using a novel representation based on the exemplar SVM. This subtle clustering of the data leads to learning better mixture models, as is demonstrated via extensive evaluations on Pascal VOC 2007. The final classifier, using a HOG representation of the global image patch, achieves performance comparable to the state-of-the-art while being more efficient at detection time.

1 Introduction

Object class detection and recognition is a major challenge within computer vision. It has been most successfully tackled with the approach: learn a discriminant function from labelled data sets of positive and negative examples [5]. The decisions about the form of this discriminant function and how it should be learnt are critical. These decisions require one to consider that the appearance of images of the same object class can vary significantly due to clutter, lighting, view-point of the camera and intra-class variation. There is also a strong bias imposed by photographers with their preferences for specific viewpoints and illuminations. These variations and biases lead to a multi-modal distribution of the positive class irrespective of representation. Combined with the almost uniform distribution of the negative class, this results in non-linear decision boundaries. This paper addresses this non-linearity with a mixture of discriminative functions which exploit the multi-modal nature of the positive class.

In order to be able to scale the method to large data set and reduce memory and computational costs of both the training and the testing phases, instead of using non-linear mappings of the data [16, 19, 17] or utilizing the invariances inherent in more complex representations e.g. [6], we focus on the use of mixture of linear discriminants. Here each classifier effectively distinguishes one mode of the positive class distribution from the background[5, 9]. This framework is attractive as the simplicity of the component classifiers serve to regularize the overall classifier and avoid over-fitting.

However, learning such mixture of classifiers is not trivial when the association of each positive training example to a mode of its class distribution is unknown, the case when one only has weakly annotated data. How to achieve this learning robustly in this scenario is the main motivation of this paper. One can try to perform an optimization which simultaneously finds the assignment of each positive training example to a mixture and learns each discriminative classifier. But this is a non-convex and expensive optimization problem bedeviled by local minima. Instead we propose to de-couple the association of the positive examples to the mixture components and the discriminative learning of the classifiers.

We regard the problem as consisting of two stages. The first is associating each example with a mode - for which we use the term *Mixture Component Identification (MCI)* - while the second is learning the mixture of classifiers given the associations which we refer to as *Mixture Component Learning (MCL)*. Figure 1 illustrates our approach: we group visually similar positive samples of a class together and learn linear classifiers for each group of samples.

We show in the experimental section that such a grouping results in learning better classifiers per cluster which in turn improves the performance of a detection system. Extensive experiments are performed on the Pascal VOC-2007 data set where the configuration settings of our algorithm are thoroughly tested. The contributions of this work are: 1) to promote the use of unsupervised clustering - based on visual similarities - in mixture modeling, for the purpose of visual recognition and 2) to propose a new robust visual similarity measure using a representation derived from exemplar SVMs[11].

Following a review of the related work, the organization of the rest of the paper is: section 2 introduces our method, our experiments and results are described and interpreted in section 3 and the paper is concluded in section 4.

Related work

Related to our work are all the works which address different sources of variations such as view point [15, 10], articulation [18] and sub-categories [1]. We aim to address the sources of variations without explicitly modelling any and without using any extra supervision, in a way that leads to better performance in the detection task. Therefore, we implement a discriminative framework - to perform well in the detection task - combined with a rather generative reasoning - to address the variations - for careful initialization of the discriminative model. A rather similar

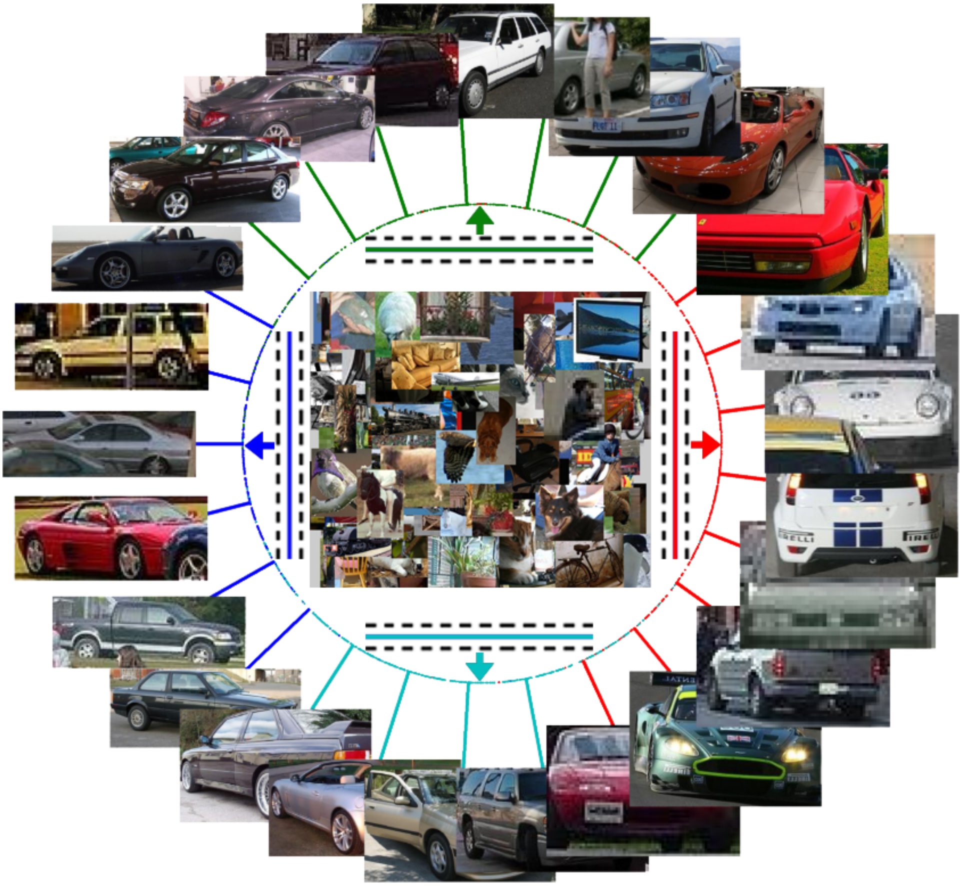


Figure 1: **The high level overview of our approach.** We group visually similar positive instances together and for each cluster, learn a linear classifier which separates the cluster from all negative data. Each color represents a different cluster.

argument can be found in [7] and a similar approach for a different problem is taken in [12].

Previous works have often utilized mixture models and - either explicitly or implicitly - dedicated mixture components to modes of the aforementioned multimodal distributions e.g. [8, 9, 7, 5]. Unlike the greedy optimization steps in boosting based approaches, we use the latent SVM formulation of [5] - which is essentially a mixture of linear SVMs - for our MCL step. The latent SVM formulation minimizes a convex objective once the latent variables, which include the data-component associations, are fixed. However, once the latent variables are allowed to vary, the problem is non-convex and is referred to as semi-convex [5]. This non-convexity makes the latent SVM initialization-dependent.

Most similar to our work is [7], which - in the unsupervised case - initializes a latent SVM using a clustering of the positive examples. In comparison to our work: 1) the similarity measure in [7] does not perform any feature selection and therefore is clutter sensitive, 2) the focus of [7] is view-point classification and therefore, very little experiment is done in the direction of object recognition, 3) the objective being minimized in [7] is slightly different: ℓ^2 regularization for large number of components leads to over-regularization for the same cache size; therefore the variables C_{Neg} and C_{Pos} are included in (3) of [7] which probably require extra cross-validation while Cs in our case are fixed for different number of components, thanks to the max regularization.

Unsupervised MCI is possible either by explicitly using a generative model or by unsupervised clustering of the positive data. Current approaches in the second direction include the clustering according to the Aspect Ratio of the bounding boxes [5], a combination of HOG and AR similarity [7] and the recent ensemble of exemplar SVM approach [11] which essentially treats each positive sample as a mixture component. The AR clustering is a very crude estimate of the visual similarity of the data and therefore, clusters based on aspect ratio do not necessarily contain visually similar samples. HOG based similarity - without feature selection - is sensitive to clutter, as it will be shown later in sections 2 and 3. Therefore, linear combination of the two - as suggested in [7] - cannot overcome the mentioned shortcommings. On the other hand, MCL based on one positive sample inherently cannot generalize well. We now describe how to measure and utilize visual similarity to group the positive data and learn a mixture model with one linear classifier per cluster which discriminates better than the former and generalizes better than the latter.

2 Visual Similarity Based Mixture Model Learning

2.1 Mixture Learning Framework

Our learning framework consists of two de-coupled steps: MCI and MCL. The MCI step, given a desired number of components c, assigns to each training example, x_i, a mixture component number $m_i \in \{1, \ldots, c\}$. We further describe the elements of the MCI step in sub-sections 2.3 and 2.2.

The MCL step, given the data-component associations, learns a model for each component using a latent SVM [5] formulation. The training data in this step consists of the following. There is a set of positive examples and their associated mixture components $\mathcal{D}_p = \{(x_1, m_1), \ldots, (x_N, m_N)\}$, a set $\mathcal{D}_n = \{x'_1, \ldots, x'_{N'}\}$ of negative examples and finally a set $\mathcal{Z}(x)$ containing all the candidate bounding boxes which overlap more than 50% with the annotated bounding box of x[1].

Let $\Phi(x, z)$ denote the modified HOG [2] feature vector of [5] extracted from the bounding box z. The MCL step learns the parameters $\boldsymbol{\beta} = (\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_c)$ by minimizing the objective function:

$$L(\boldsymbol{\beta}) = \frac{1}{2}\max_i \|\boldsymbol{\beta}_i\|^2 + C\sum_{i=1}^{N} \max\big(0, 1 - f^+_{\boldsymbol{\beta}}(x_i, m_i)\big) + C\sum_{i=1}^{N'} \max\big(0, 1 + f^-_{\boldsymbol{\beta}}(x'_i)\big) \quad (1)$$

where the scalar C controls the relative weight of the regularization with respect to the hinge loss and

$$f^+_{\boldsymbol{\beta}}(x, m) = \max_{z \in \mathcal{Z}(x)} \boldsymbol{\beta}_m \cdot \Phi(x, z), \qquad f^-_{\boldsymbol{\beta}}(x) = \max_m f^+_{\boldsymbol{\beta}}(x, m). \quad (2)$$

In (2), the data-mixture component associations ("m_i" s) for positive samples were fixed to those found in the MCI step. The "m_i" s can also be treated as latent variables. This increases the non-linearity of the objective function which in turn increases the number of local minima. However, we expect a careful initialization to result in better minima. This is empirically validated later in section 3.

We use a slightly modified version of [4] to optimize (1) and unless stated otherwise, we use the same parameters as in the original implementation.

2.2 Measuring Visual Similarity

To perform successful clustering one must have a good way of measuring similarity between examples. This is a tricky task as background and foreground clutter affect the appearance of an object instance within its bounding box. Hence, to robustly measure the visual similarity between two examples from the same visual class one needs to disregard the irrelevant clutter.

We use the recently developed exemplar SVM [11] to suppress this clutter. The aim of the exemplar SVM is to learn a classifier which best separates a single positive example from the large set of negative examples. The classifier learnt based on this premise effectively performs feature selection on that particular example. It suppresses the uninformative detail inside the bounding box, see figure 3, which is not useful when discriminating it from the negative class. The exact details of how we robustly measure visual similarity now follow.

Let $\{\mathbf{w}_i \mid i = 1, \ldots, n\}$ be the set of n sparse basis filters (in this paper these filters correspond to the weights of the exemplar SVMs learnt for each training example). Each one is applied linearly to the feature extracted from the image patch in x defined by the bounding box z as $\mathbf{w}_i \cdot \Phi(x, z)$. A calibration process is then required to ensure the scores from the different basis filters are comparable. This is achieved with the sigmoid function and we define our basis functions as

$$F_i(x, z) = \frac{1}{1 + \exp(-\alpha_i(\mathbf{w}_i \cdot \Phi(x, z) - \gamma_i))} \quad (3)$$

[1] Here, the set of valid bounding boxes should be a function of the dimensionality of the corresponding filter. This was neglected in the notations for the sake of brevity.

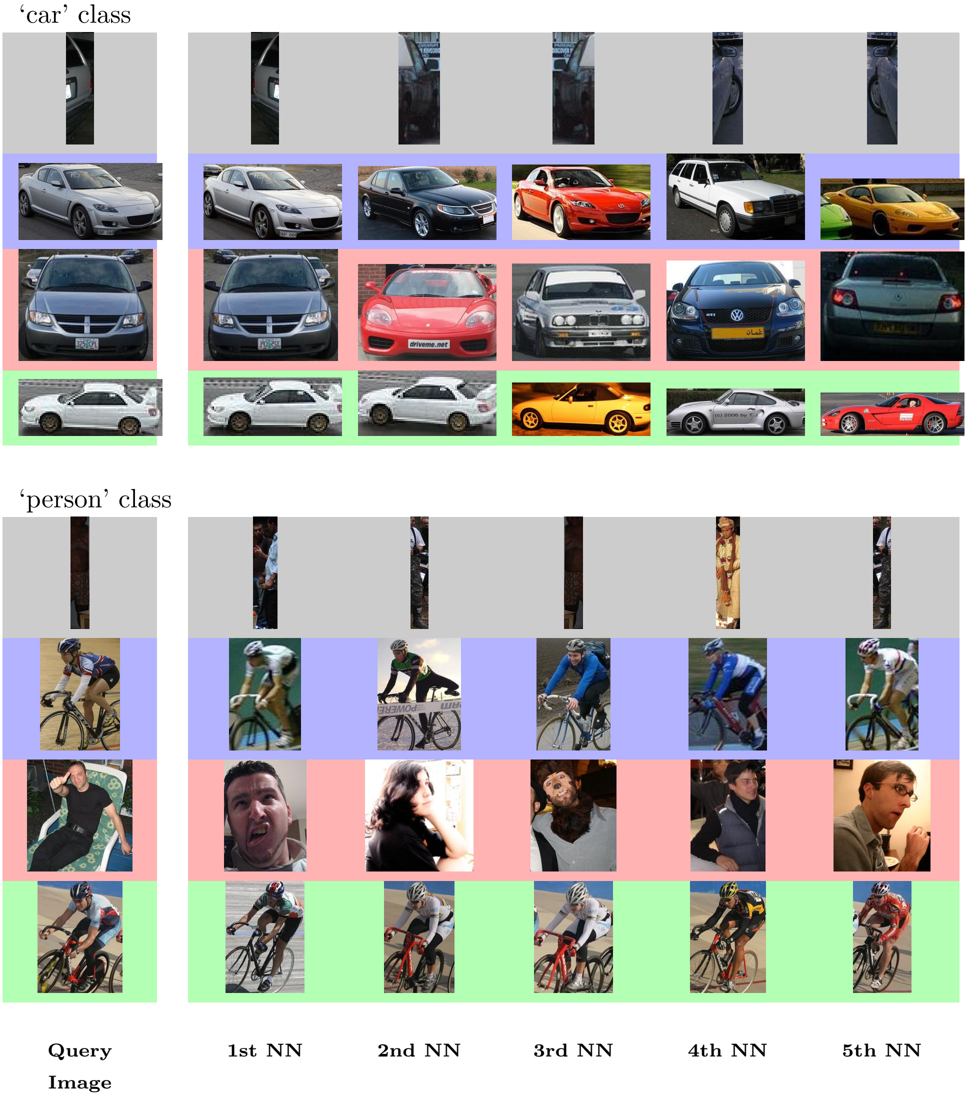


Figure 2: **Nearest neighbors produced by different visual similarity measures.** The similarity measures within each block, from top to bottom are: HOG similarity without feature selection, K^E, K^E_{MI} and K^E_{MMI}. The leftmost column shows the image with highest similarity to its 10 nearest neighbors and to its right are its 5 nearest neighbors. Note how feature selection based on exemplar SVM results in better measures of visual similarity.

where α_i and γ_i are the calibration parameters learnt as in [11][2] and $\mathbf{w}_i$ is the i-th sparse basis filter. Let $E_i(x)$ be the maximum score of $F_i(.,.)$ over the valid latent positions of x:

$$E_i(x) = \max_{z \in \mathcal{Z}(x)} F_i(x, z) \tag{4}$$

This maximization process corresponds to finding the best alignment over scale and translation, the search is over bounding boxes of different size and position, of the sparse filter with the test image patch and can be found via convolution.

If there is a one-to-one association between the basis functions and the positive training examples, which is the case if an exemplar SVM is trained for each positive example, we can directly use the bases to evaluate *visual structural similarity* between the i-th and j-th positive training instance. Assuming the same order for the bases and the positive examples in this case, we can define

$$K^E(x_i, x_j) = \frac{1}{2}\left(E_i(x_j) + E_j(x_i)\right) \tag{5}$$

where symmetry is achieved by averaging between two model responses. However, if a one-to-one association between the bases and the positive training samples does not exist or cannot be established, other measures need to be utilized as the K^E measure cannot be evaluated on such cases. Let $\mathbf{E}_x = (E_1(x), \ldots, E_n(x))$ be the vector of all basis functions aligned and evaluated on x. With this new fixed length representation of x, we can utilize any kernel to measure similarity between two instances without directly associating either of the instances with the bases. Applying the Intersection Kernel on this representation, the visual similarity between two image patches becomes:

$$K^E_{\mathrm{MI}}(x, y) = \sum_{i=1}^{n} \min\left(E_i(x), E_i(y)\right) \tag{6}$$

As a specific example is usually visually similar to only a limited number of examples, averaging (mean pooling) the intersection measure on all the bases will unnecessarily smoothen out the responses. Therefore, if the responses of the bases are calibrated *with respect to each other*[3], we can make use of measures which are more sensitive to the responses of the bases. Therefore, we utilize ℓ^∞ on the intersection measures and define the K^E_{MMI} as the max pooling of the intersections:

$$K^E_{\mathrm{MMI}}(x, y) = \max_i \min\left(E_i(x), E_i(y)\right) \tag{7}$$

Figure 2 shows the top nearest neighbors using each similarity measure evaluated on several classes. Similar to the results reported in [11], feature selection according

[2] We used the models provided by the authors.

[3] We need to emphasize here that while the exemplar SVMs in [11] are not calibrated with respect to each other, we found out the independent calibrations to be sufficiently accurate to be used in K^E_{MMI} (7).

to the exemplar SVMs results in better visual similarity measures which in turn leads to visually more appealing nearest neighbors. It is evident from the figure that unlike K^E_{MMI} which is sensitive to subtle variations in basis responses, the averaging behavior of K^E_{MI} does not result in visually appealing nearest neighbors if the class exhibits high variations.

Let $L = \frac{1}{N} \sum_{x \in \mathcal{D}_p} |\mathcal{Z}(x)|$ be the average number of latent positions over the positive training set and D be the average dimensionality of the linear weights of the basis filters . The computational complexity of evaluating a full affinity matrix using K^E is $\mathcal{O}(Dn^2L)$. Assuming the same number of bases as positives i.e. $N = n$, the computational complexity of evaluating a full affinity matrix using K^E_{MI} and K^E_{MMI} is $\mathcal{O}(Dn^2L + n^3)$. However, as usually $DL \gg n$, the dominating factor is still the convolutions which makes the computational complexity of all measures equivalent. We now describe how these similarity measures can be used to identify mixture components.

2.3 Mixture Component Identification via Unsupervised Clustering

With our similarity measure K, we can cluster our positive data using spectral clustering [13]. We construct fully connected similarity graphs and use the similarity measure as the affinity measure s.t. $\mathbf{W} = (w_{ij})$ and $w_{ij} = K(x_i, x_j)$. Let $\mathbf{L}_{sym}$ denote the symmetric normalized Laplacian:

$$\mathbf{L}_{sym} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\, \mathbf{W}\, \mathbf{D}^{-\frac{1}{2}} \tag{8}$$

where $\mathbf{D}$ is the degree matrix - a diagonal matrix with diagonal entries $d_{ii} = \sum_j w_{ij}$. In order to identify c components, we compute the first $c + 1$ eigenvectors $\bar{\mathbf{u}}_0, \bar{\mathbf{u}}_1, \ldots, \bar{\mathbf{u}}_c$ of $\mathbf{L}_{sym}$ and ignoring the first eigenvector, construct $\bar{\mathbf{U}} = (\bar{\mathbf{u}}_1, \ldots, \bar{\mathbf{u}}_c)$. Let $\mathbf{U}$ be the matrix obtained by normalizing the rows of $\bar{\mathbf{U}}$: $u_{ij} = \bar{u}_{ij} / \left(\sum_k \bar{u}^2_{ik}\right)^{\frac{1}{2}}$. We refer to the i-th row of $\mathbf{U} \in \mathbb{R}^{n \times c}$ as $\mathbf{u}_i$ and the mapping - according to K and c - from x_i to $\mathbf{u}_i$ as the (c, K)-*spectral projection*.

The ℓ^2 distance is well suited to the spectral projection ($\mathbf{u}$) representation and therefore, as suggested in [13], k-means on this representation gives a good clustering of the data. The 2D coordinates of the instances in Figure 1 depict the $(2, K^E_{\text{MMI}})$-spectral projection of a subset of the car examples. It can be observed that the ℓ^2 distance on this representation reflects the visual similarity between instances: points close in this space are expected to be visually similar. Because of this fact, we can measure the quality of a cluster by computing the average distance between two samples in the cluster. The colors in Figure 1 reflect the association of samples to the top 4 clusters from the 5 clusters produced by k-means on the $(5, K^E_{\text{MMI}})$-spectral projection of the data. The 5th cluster had a high average distance measure as it mainly contained everything which was not visually similar to samples of any of the other clusters and therefore, it was omitted for visualization purposes.

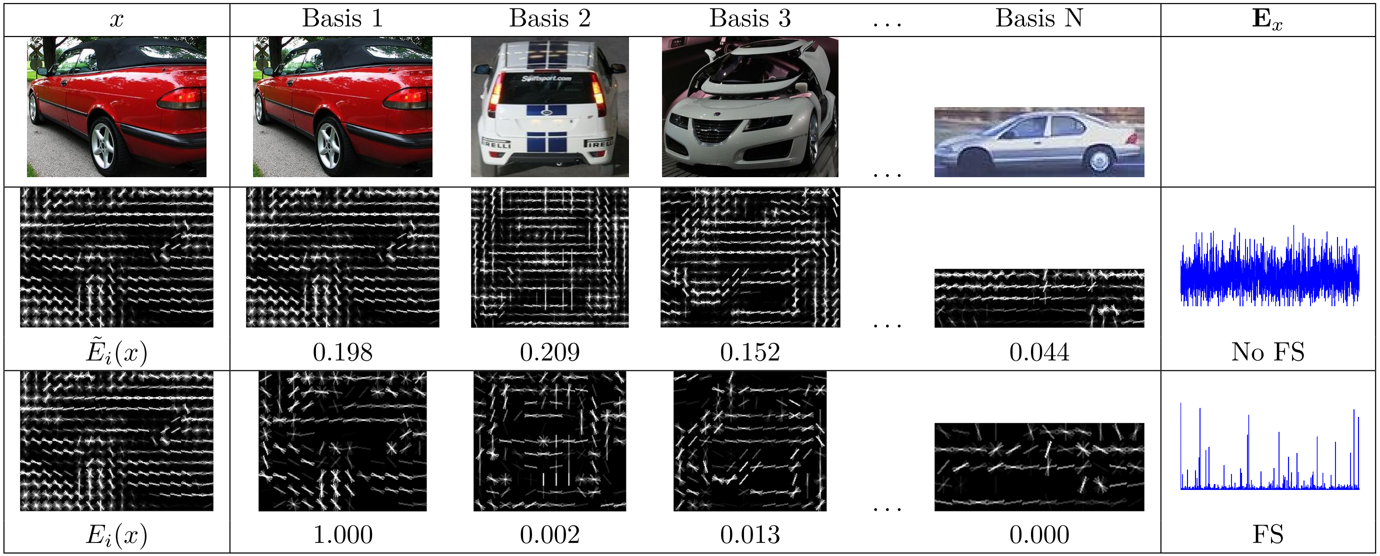


Figure 3: **Visualization of x projected onto a set of basis filters.** In this figure the feature vector, $\Phi(x, z)$, extracted from example x is projected onto two different sets of basis filters. The first is a non-sparse basis and corresponds to the original HOG feature representation of each training example, while the second is a sparse one based on its exemplar SVM weight vector. The suppression of the clutter in the sparse basis allows for a more precise matching w.r.t. visual similarity (compare E_i s with $\tilde{E}_i$ s).

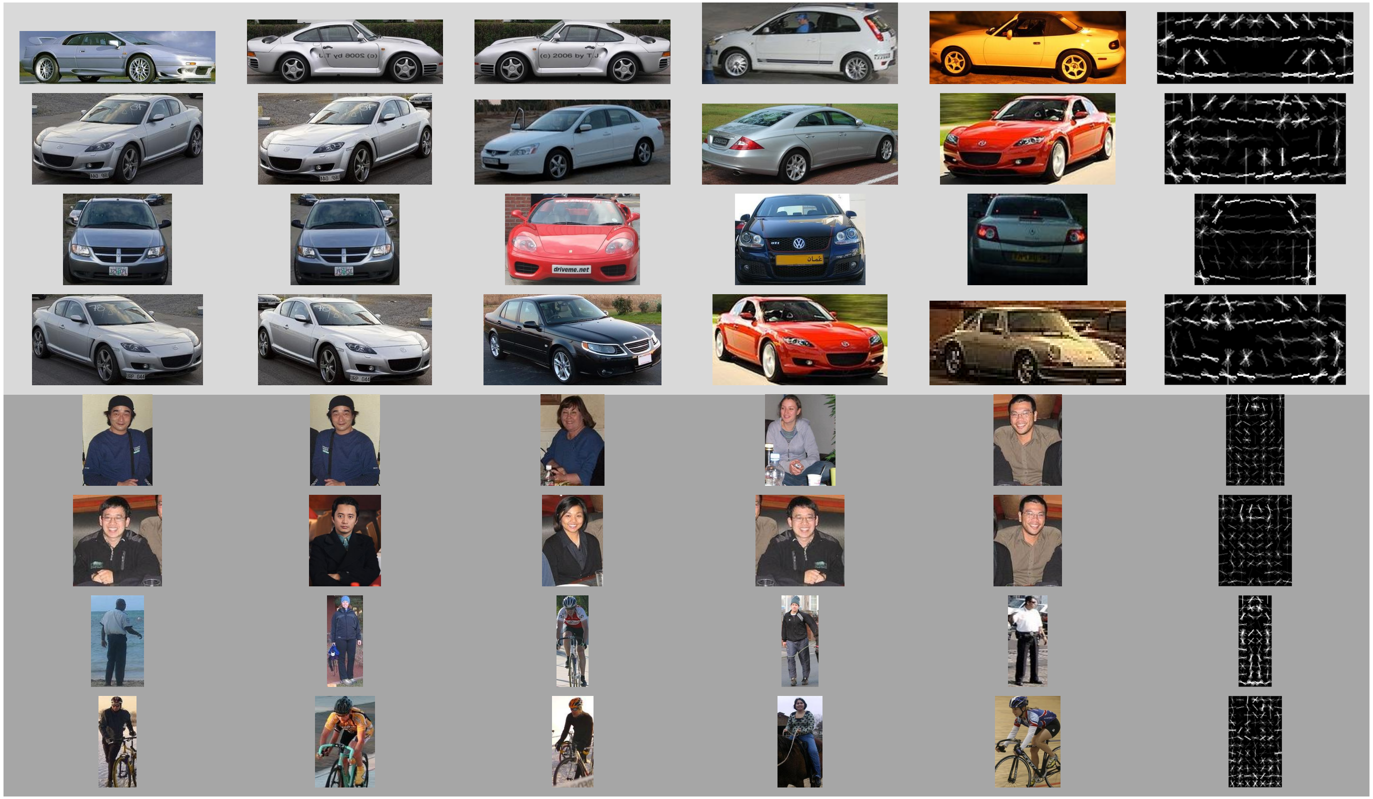


Figure 4: **Visualization of the clusters.** Each row shows the top 5 samples of the top 4 clusters based on the highest average kernel similarity after $(5, K^E_{\text{MMI}})$-spectral clustering of the car and person classes (see text for details). The last column depicts the positive weights of the model learnt for each cluster in the MCL step.

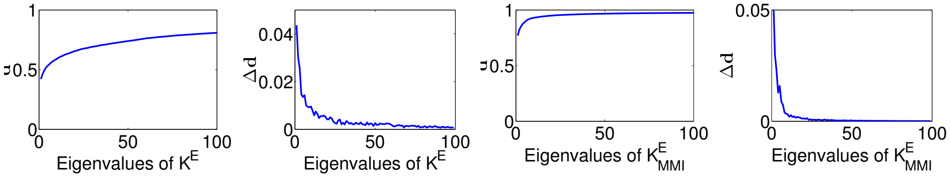


Figure 5: **Rank analysis of K^E and $K^E_{\mathbf{MMI}}$:** average of the (oredered) eigenvalues of $\mathbf{L}_{sym}$ ($\bar{\mathbf{d}}$) and its derivative ($\boldsymbol{\Delta}\bar{\mathbf{d}}$) when using K^E and K^E_{MMI} as visual similarity measures.

Example clusters found using the K^E_{MMI} similarity measure are shown in figure 4. Shown are the top 5 samples of the top 4 from the 5 clusters for four classes and the filters (the $\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_c$ from equation (1)) learned for each cluster. Here, the top sample refers to samples with the highest average visual similarity, using K^E_{MMI}, to all instances associated with the same component. The top cluster is considered as the cluster with the highest average visual similarity between the samples assigned to the cluster. It can be observed that the MCI step groups together examples that are visually similar.

It is worth noting that while the K^E and K^E_{MMI} visual similarity measures are not kernels i.e. they do not result in positive definite affinity matrices, they can be utilized in the spectral clustering as the spectral projection utilizes (the normalized version of) the largest eigenvalues of the affinity matrix. Let $\tilde{\mathbf{d}}$ refer to the vector of ordered eigenvalues of the $\mathbf{L}_{sym}$ and $\bar{\mathbf{d}}$ refer to the average $\tilde{\mathbf{d}}$ values over all classes in Pascal VOC 2007. Figure 5 shows $\bar{\mathbf{d}}$ and its derivative when using K^E and K^E_{MMI} as similarity measures. It can be observed that K^E results in higher rank affinity matrices leading to lower rank normalized Laplacians; which means that K^E_{MMI} is potentially preferable for coarser clusterings (less number of clusters). This is also experimentally validated later in Figure 6.

3 Experiments

Data set: We evaluate our method on the Pascal VOC 2007 [3] data set, training on the train + validation set, and testing on the test set and using the Average Precision (AP) and mean Average Precision (mAP) as performance measures. We report the performance of the MCI + MCL framework based on different visual similarity measures and different number of mixture components. Therefore we review the visual similarity measures considered and our acronyms for them : 1) aspect ratio (AR) as a very crude measure of visual similarity, 2) visual similarity *without feature selection (HOG)*: linear kernel on HOG feature vectors with latency on the position and scale and 3) visual similarity *with feature selection* (K^E, K^E_{MI} and K^E_{MMI}). A '+L' in the results denotes an MCL step with latent data-component association, initialized from the data-component associations of the MCI step.

Performance vs number of components: Figure 6 (top) shows the *mAP vs the number of mixture components* when different visual similarity measures are used in the MCI step. We point out the following observations: *1)* Clustering based on AR performs well only for low numbers of components i.e. 3 and 5 components. Unlike other visual similarity measures however, it fails to provide good initializations when the non-linearity of the objective increases. *2)* Latent (positive) data-component association is beneficial almost consistently (with the exception of AR:5). The extra non-linearity introduced to the objective via this latent formulation is initialization dependent (compare K^E_{MMI}+L and AR+L). *3)* Feature selection in visual similarity measure improves the performance (compare HOG with K^E_X). *4)* The performance tends to improve when more mixture components are utilized in combination with MCI based on visual similarities.

We did not experiment with higher number of mixture components mainly because of the computational expense. We observed, though, that the performance of K^E_{MMI} - which outperforms all other measures consistently up to and including 10 mixture components - degrades after 10 components while the smoother measures K^E and K^E_{MI} continue to benefit from more mixture components. The main reason of failure in these cases is the domination of the ℓ^2 distance in the k-means clustering (after the spectral projection step) by the eigenvectors associated with large eigenvalues of the normalized Laplacian (small eigenvalues of the affinity matrix which tend to be noisy). Addressing this issue is out of the scope of this work but, a potential solution is to use less eigenvectors than the desired number of clusters, in the spectral clustering step.

Performance vs Model Complexity: Figure 6 (bottom) shows the performance of the MCI + MCL framework vs the models' parameters (averaged over the 20 classes) using K^E_{MMI} (magneta) and AR (green) visual similarities and for models with 3, 5 and 10 mixture components. In the figure, a '-F' refers to a model without the flip heuristic[4] and a '+S' refers to a finer (2 scale) HOG representation instead of a coarse representation (the same scale as the root filters in [4]). Additionally, the performance of the model is shown if an oracle were available to tell the model the optimal number of mixture components for each class (assign each class a number $c_i \in \{3, 5, 10\}$); shown with a '+O' in the legend entries.

The analysis of the figure is as follows: *1)* Models with the flip heuristic outperform equally complex models based on the same similarity measure and without the flip heuristic (compare K^E_{MMI}+L+S-F with the rest of models based on K^E_{MMI}). The reason for this is probably the reduced degrees of freedom imposed on the model using the flip heuristic which prevents model from over-fitting. *2)* The use of an oracle (the '+O' entries) improves the performance of a coarse representation by

[4] In [4], for each mixture component by default two filters are learnt that are flipped horizontally with respect to each other i.e. a 3 component mixture contains 6 (root) filters. This constraint essentially reduces the degrees of freedom in comparison to a model with the same number of filters without the flip constraint.

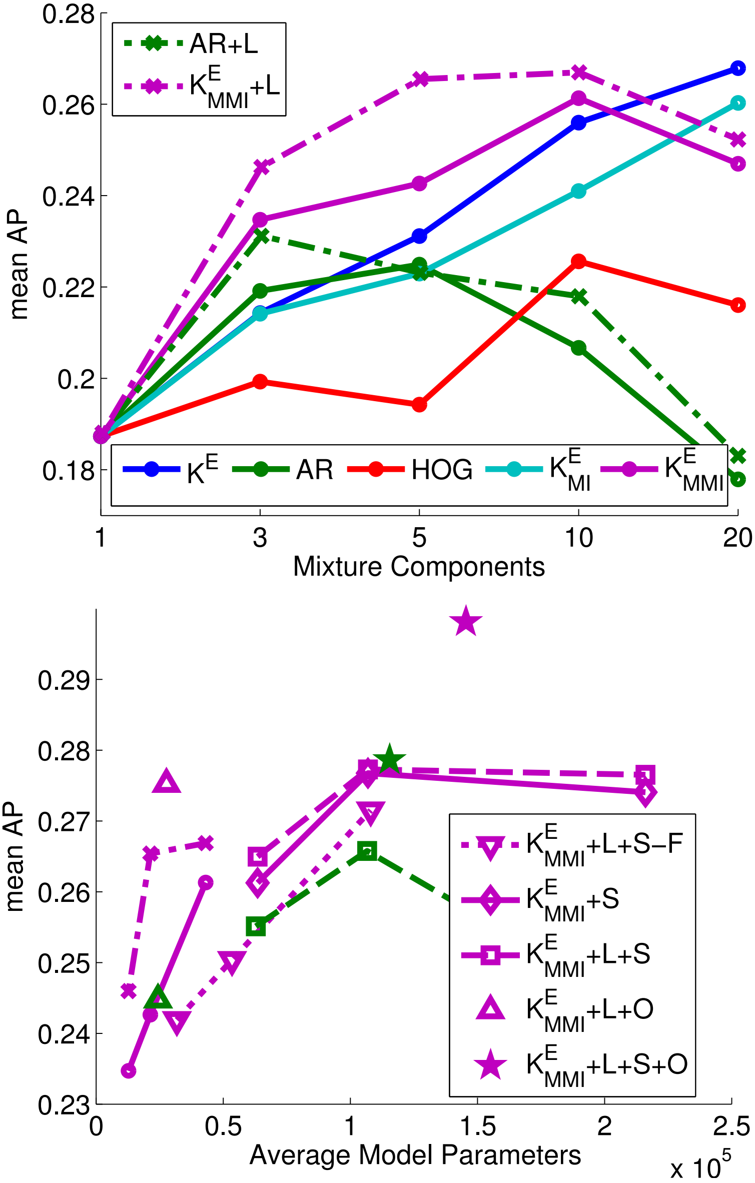


Figure 6: The performance of the MCI + MCL framework using different visual similarity measures on Pascal VOC 2007 classes. Top: results achieved by varying the number of components for each visual similarity measure. Bottom: performances vs model complexity for 3, 5 and 10 component mixture models in different configurations(see text for details).

approximately 0.01 mAP and that of a fine representation by approximately 0.02 mAP in case of K^E_{MMI} and by 0.015 mAP and 0.012 mAP in case of AR. These are encouraging results for future work on adapting/estimating the number of mixture components and at the same time emphasize on the use of subtle visual similarity measures: K^E_{MMI}+L+O and K^E_{MMI}+L+S+O perform 0.03 mAP and 0.02 mAP better than their AR based counterparts. *3)* Fine scale representation improves the performance of K^E_{MMI} by approximately 0.01, but improves that of AR by 0.025 in case of 3 components and 0.035 in case of 5 components. Nevertheless, AR+L+S+O is only 0.003 better than K^E_{MMI}+L+O, while it is more than 4.1 times more complex!

Performance vs Intra-Class variation: In order to analyze the performance of our models in presence of different bias and variation levels of the positive classes, we need to be able to approximate the intra-class variation[5]. In the following, we assumed the intra-class variation is negatively correlated with the performance of K^E_{MMI}+L+O and we made our arguments reasonably invariant to the actual measure we used to approximates intra-class variation by considering the ordering of the classes instead of the exact measured values. This makes the estimates invariant to any monotonic transformation of the measure. It is worth mentioning that similar overall conclusions can be drawn using other reasonable measures e.g. the performance of a one component latent SVM model or the results of AR+L:3, leads to similar overall conclusions.

Figure 7 (top) shows how the performance of K^E_{MMI}+L decreases when intra-class variation increases. The solid lines are fitted to the actual data depicted by dashed lines via linear regression. Higher bias (simpler) models are expected to work better when intra-class variation is large and sufficient data is not available for the classifier to efficiently learn the discriminative structures. As expected, more complex models perform worse in presence of larger intra-class variation: slope of the lines increases when more mixture components are utilized and also, a 5 component model performs better than a 10 component model on classes with more intra-class variation than 'diningtable'. At the same time 1 and 3 component models are almost consistently outperformed by 5 and 10 components; except the last 3 classes: bird, dog and plant which probably require other representations, more data or more supervision.

Figure 7 (bottom) shows how K^E_{MMI}+L compares with AR+L:3. It can be observed that in all cases, the gain has a positive slope i.e. improvement gets more as intra-class variation increases. However, the slope decreases when the complexity of the model increases. Considering the slope and intercept, we can conclude that K^E_{MMI}+L with 5 and 10 components almost consistently outperform AR+L:3.

Comparison to related works: Table 1 shows the performance of the MCI+MCL framework using 2 configuration settings on each class of the data set compared to the ESVM approach and 3 part based models. It can be observed that without

[5] Here, we neglect the effect of the inter-class variations.

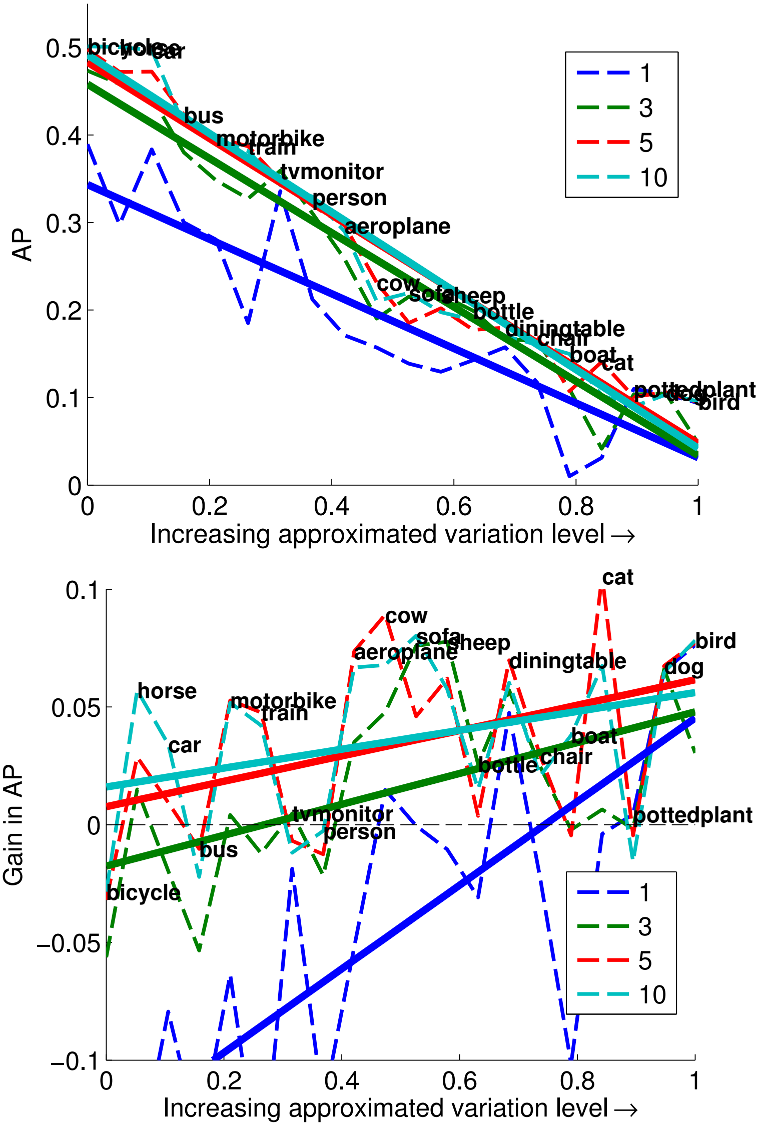


Figure 7: Performance (of K^{E}_{MMI}) vs approximate intra-class variation level on (top) and AP gains in comparison to AR+L:3 (bottom).

using parts, we outperform the state-of-the-art part based models - based on the HOG representation - in 2 classes and outperform 2 part based models in mean AP. It should be noted that although the training process is expensive for a visual similarity based MCI step, the testing phase consists of convolutions of linear filters learnt in the MCL step; without any dynamic programming step to account for deformation of the parts. This, without requiring a cascade or hierarchical model, is cheaper and better paralellizable compared to part based models and more sophisticated approaches such as [16]. Furthermore, the same framework can potentially be utilized to train better root filters for any part-based model and to provide better initialization for their non-convex optimization.

Acknowledgements: This work has been funded by the Swedish Foundation for Strategic Research (SSF); within the project VINST (Wearable Visual Information Systems) and by the European Commission KIC: EIT ICT labs. We thank PDC for letting us use their supercomputers, without which the thorough experiments presented in this work were impossible to perform. We also would like to thank Carl Henrik Ek for the constructive discussions around the topic.

4 Conclusions

In this paper, we introduced the MCI + MCL mixture learning framework and promoted the use of visual similarity measures for the MCI step. We performed extensive evaluations of the proposed framework based on different visual similarity measures on the Pascal VOC 2007 data set. The framework achieved very promising results, outperforming the bases we used - the exemplar SVMs - in the detection task and 2 part based models without using parts.

Future work includes estimating the optimal number of clusters for each class, automatic refinement of the "junk" clusters - clusters which contain samples not similar to those of any other cluster's; but not sharing any structural similarities, investigating the use of other methods for the purpose of feature selection, and learning the mixture of discriminants with methods other than the latent SVM.

References

[1] Aharon Bar-Hillel and Daphna Weinshall. Subordinate class recognition using relational object models. In *Neural Information Processing Systems*, 2006.

[2] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[3] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[4] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester. Discriminatively trained deformable part models, release 4. http://people.cs.uchicago.edu/ pff/latent-release4/.

[5] Pedro F. Felzenszwalb, Ross B. Girshick, David A. McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010.

Table 1: Results on the Pascal VOC 2007 data set. LDPM , CFHPM and DTDPM-R4 are part based models. Without any post-processing and without using parts, we outperform state of the art in 2 classes and two part based models in mean AP.

Method Class	aero	bicycle	bird	boat	bottle	bus	car	cat	chair	cow	mAP
ESVM+Co-occ[11]	.208	.480	.077	.143	.131	.397	.411	.052	.116	.186	.227
LDPM [5]	.290	.546	.006	.134	**.262**	.394	.464	.161	.163	.165	.262
CFHPM [14]	.277	.540	.066	.151	.148	.442	.473	.146	.125	.220	.269
DTDPM-R4 [4]	.289	**.595**	**.100**	.152	.255	**.496**	**.579**	**.193**	**.224**	**.252**	**.323**
K^E_{MMI}+L:10	.290	.501	.096	.150	.189	.411	.497	.103	.160	.210	.267
$\mathbf{K^E_{\text{MMI}}}$**+L+S+O**	**.333**	.536	.096	**.156**	.229	.488	.515	.163	.163	.200	.298
Method Class	table	dog	horse	bike	person	plant	sheep	sofa	train	monitor	mAP
ESVM+Co-occ[11]	.111	.031	.447	.394	.169	.112	.226	.170	.369	.300	.227
LDPM [5]	**.245**	.050	.436	.378	.350	.088	.173	.216	.340	.390	.262
CFHPM [14]	.242	**.120**	.520	.420	.312	.106	**.229**	.188	.353	.311	.269
DTDPM-R4 [4]	.233	.111	**.568**	**.487**	**.419**	**.122**	.178	**.336**	**.451**	**.416**	**.323**
K^E_{MMI}+L:10	.170	.103	.500	.396	.330	.090	.198	.220	.382	.343	.267
$\mathbf{K^E_{\text{MMI}}}$**+L+S+O**	.238	.110	.553	.438	.369	.107	.227	.235	.386	.410	.298

[6] Kristen Grauman and Trevor Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *IEEE International Conference on Computer Vision*, 2005.

[7] Chunhui Gu and Xiaofeng Ren. Discriminative mixture-of-templates for viewpoint classification. In *European Conference on Computer Vision*, 2010.

[8] Chang Huang, Haizhou Ai, Yuan Li, and Shihong Lao. Vector boosting for rotation invariant multi-view face detection. In *IEEE International Conference on Computer Vision*, 2005.

[9] Tae-Kyun Kim and Roberto Cipolla. Mcboost: Multiple classifier boosting for perceptual co-clustering of images and visual features. In *Neural Information Processing Systems*, 2008.

[10] Joerg Liebelt, Cordelia Schmid, and Klaus Schertler. Viewpoint-independent object class detection using 3d feature maps. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[11] Tomasz Malisiewicz, Abhinav Gupta, and Alexei A. Efros. Ensemble of exemplar-svms for object detection and beyond. In *IEEE International Conference on Computer Vision*, 2011.

[12] Sobhan Naderi Parizi, John G. Oberlin, and Pedro F. Felzenszwalb. Reconfigurable models for scene recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[13] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Neural Information Processing Systems*, 2001.

[14] M. Pedersoli, A. Vedaldi, and J. Gonzalez. A coarse-to-fine approach for fast deformable object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[15] Min Sun, Hao Su, Silvio Savarese, and Li Fei-Fei. A multi-view probabilistic model for 3d object classes. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[16] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In *IEEE International Conference on Computer Vision*, 2009.

[17] A. Vedaldi and A. Zisserman. Efficient additive kernels via explicit feature maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011.

[18] Yi Yang and Deva Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[19] Jianguo Zhang, Marcin Marszalek, Svetlana Lazebnik, and Cordelia Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *International Journal of Computer Vision*, 2007.

Supplementary Materials

Figures 8 - 11 depict the $(2, K^{E}_{\mathrm{MMI}})$-Spectral Projection of the aeroplane 8, bicycle9, bus 10 and person 11 classes. The figures are acquired using the same approach as for Fig. 1 in the paper: 2D coordinates are acquired by $(2, K^{E}_{\mathrm{MMI}})$-Spectral Projection and colors represent associations to top 4 out of 5 clusters (acquired by k-means on the $(5, K^{E}_{\mathrm{MMI}})$-Spectral Projection representation). Here, similar to the paper, "top clusters" are those which have highest average kernel similarity between samples assigned to them.

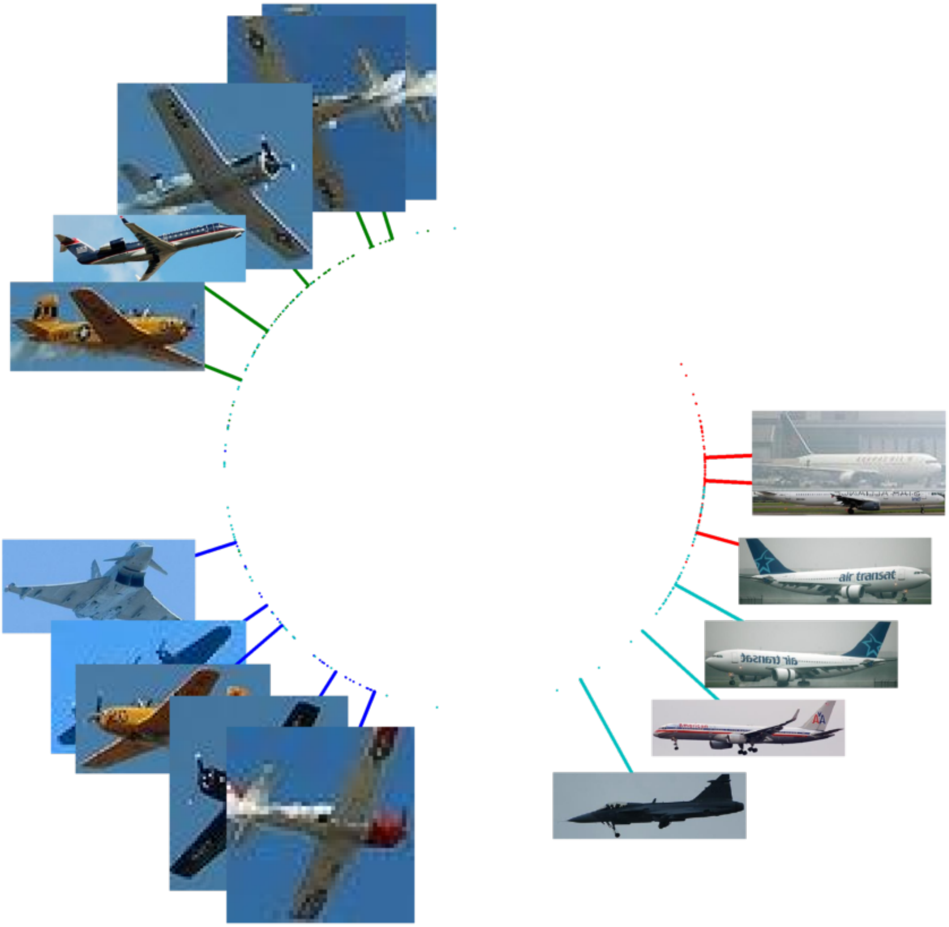


Figure 8: Visualization of $(2, K^{E}_{\mathrm{MMI}})$-Spectral Projection of the aeroplane class.

It can be observed that for the classes with small intra class variation e.g. mainly viewpoint variation, the 1 dimensional degree of freedom in $(2, K^{E}_{\mathrm{MMI}})$-Spectral Projection representation (angle) captures the variation in the underlying variation source. However, if the class has high intra class variation e.g. in case of person: articulation and sub-

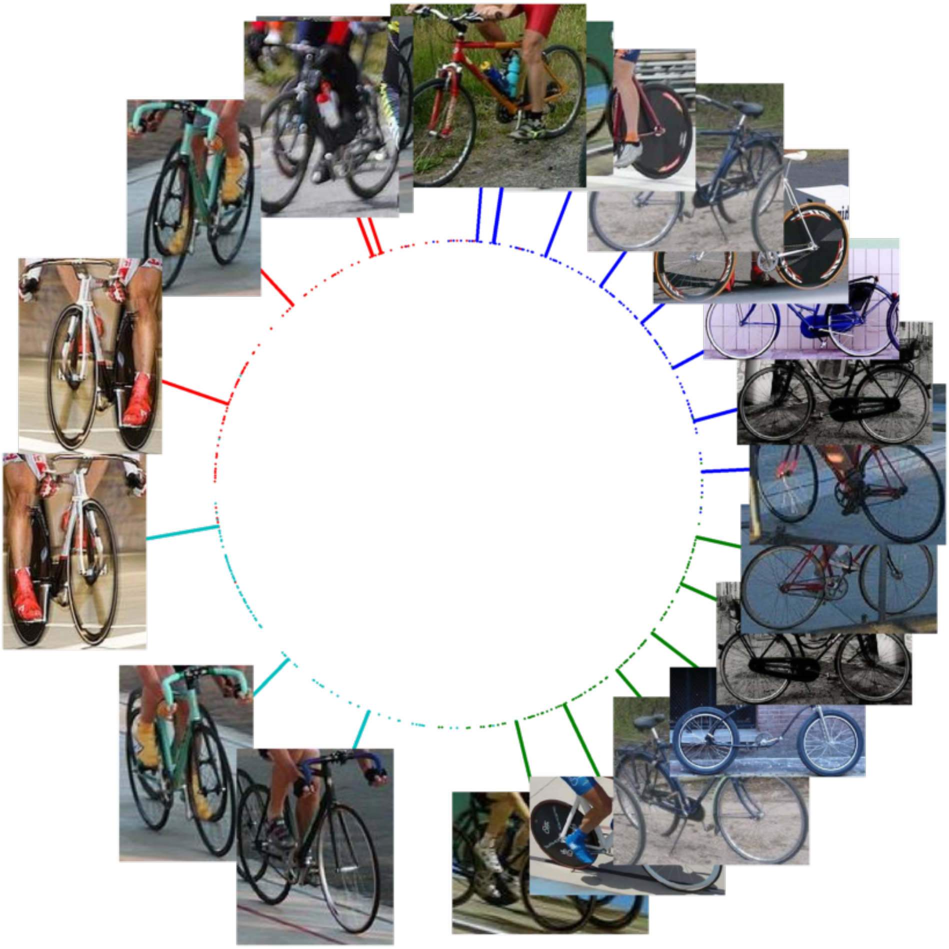


Figure 9: Visualization of $(2, K^{E}_{\mathrm{MMI}})$-Spectral Projection of the bicycle class.

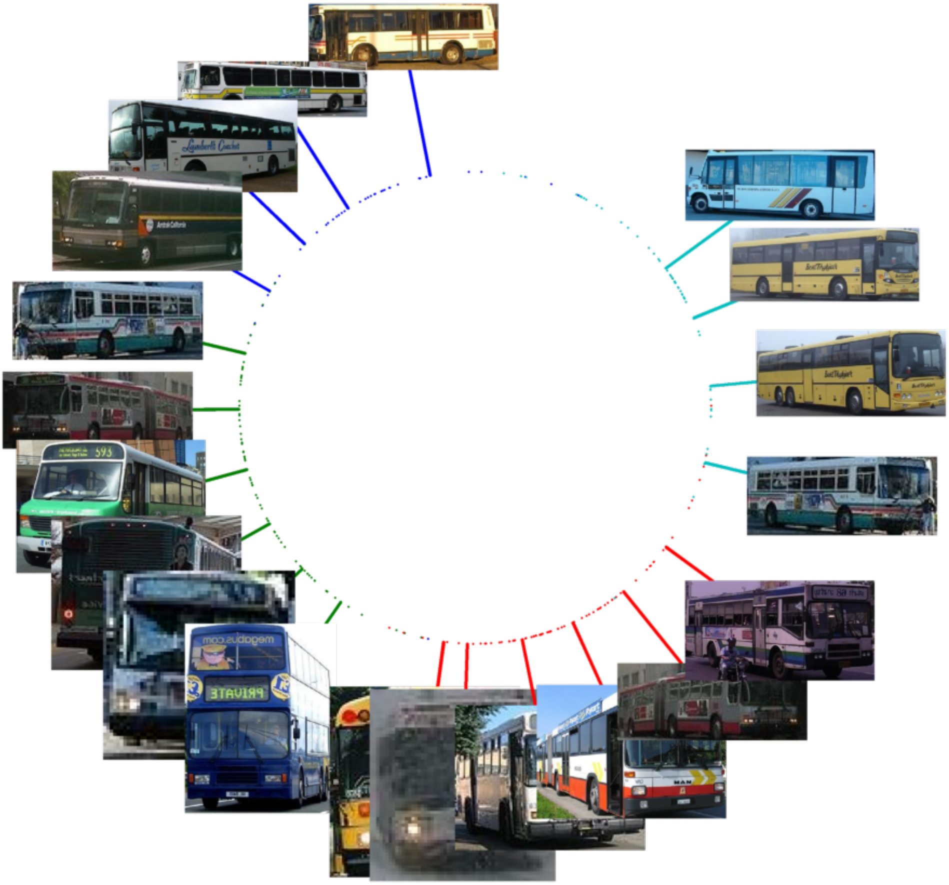


Figure 10: Visualization of $(2, K^E_{\text{MMI}})$-Spectral Projection of the bus class.

category (standing, riding a bike, sitting, etc), the 1 dimensional degree of freedom is basically not sufficient to capture a smooth transition between variations. However, by projecting the data to higher dimensions e.g. $\{(5, K^E_{\text{MMI}}), (10, K^E_{\text{MMI}})\}$-Spectral Projection (used for clustering), the ℓ_2 distance becomes a good approximation of the visual similarity. This can be verified by looking at the cluster centers and the learnt filters for each cluster (see Figures 18 and 19). The same fact can be observed for other classes: aeroplane (12, 13), bicycle (14, 15) and bus (16, 17). In Figures 12 - 19, “top image of a cluster” refers to images associated with a cluster that have highest average kernel similarity to other images associated with the same cluster.

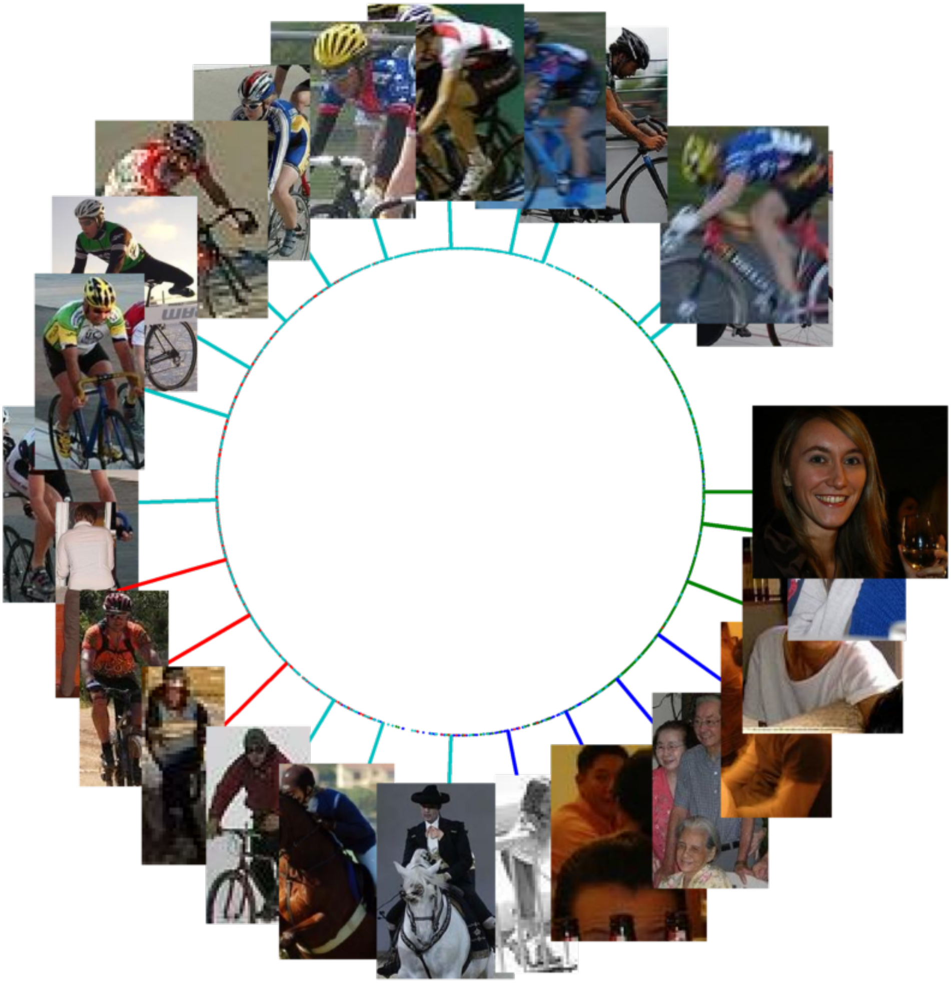


Figure 11: Visualization of $(2, K^E_{\text{MMI}})$-Spectral Projection of the person class.

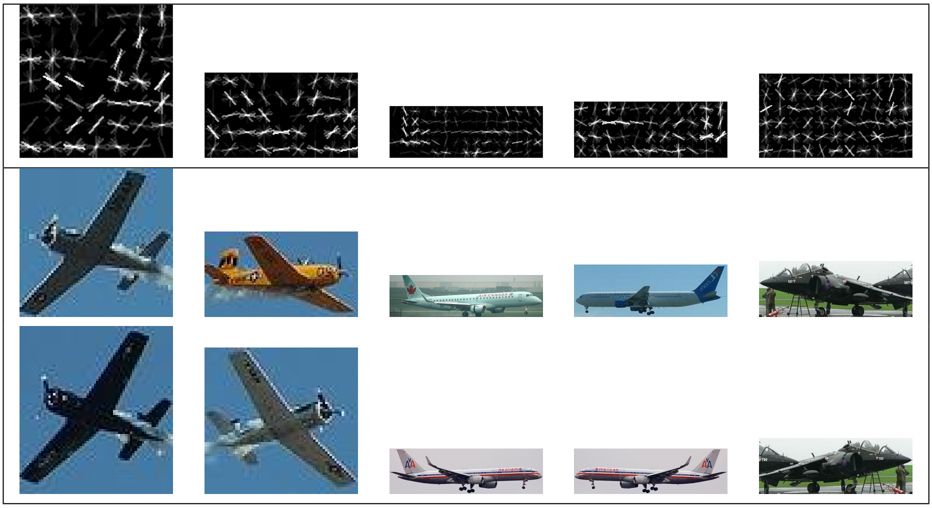


Figure 12: Filters learnt for aeroplane class in the MCL step with 5 components (first row). Below each component, the top two images associated with the component are depicted.

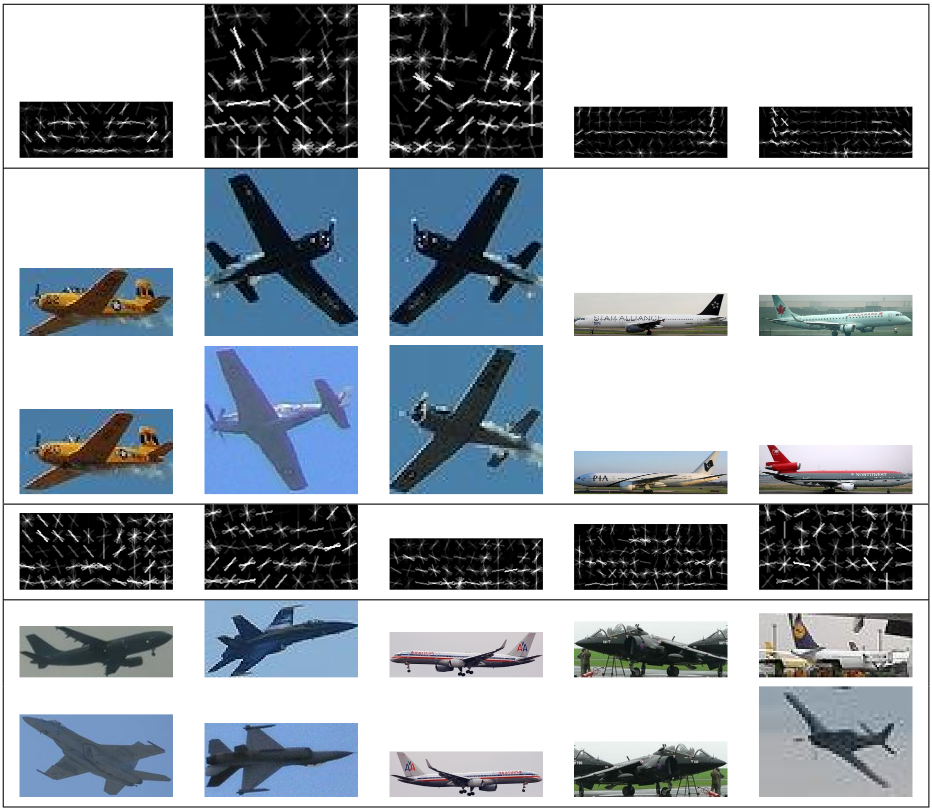


Figure 13: Filters learnt for aeroplane class in the MCL step with 10 components (first and fourth rows). Below each component, the top two images associated with the component are depicted.

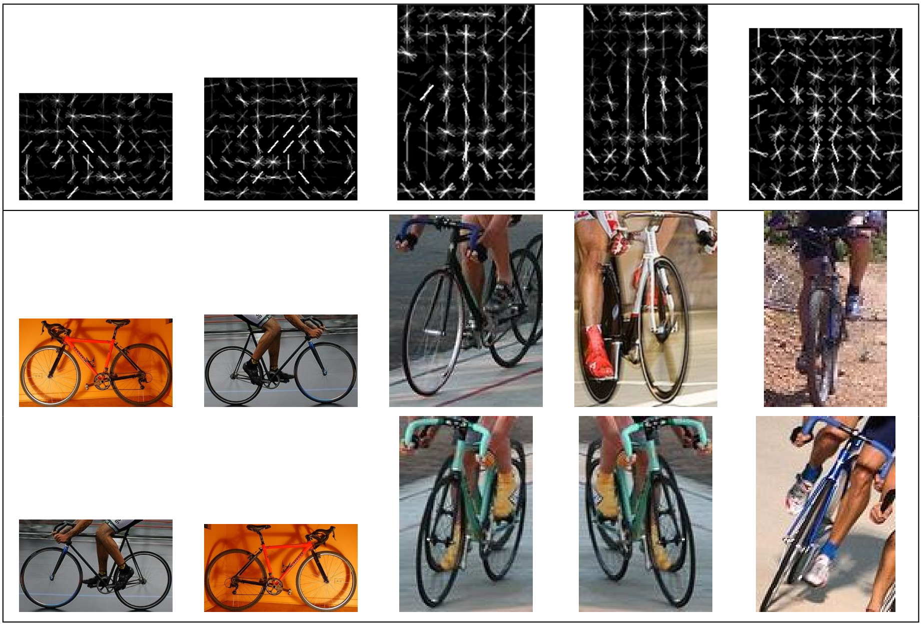


Figure 14: Filters learnt for bicycle class in the MCL step with 5 components (first row). Below each component, the top two images associated with the component are depicted.

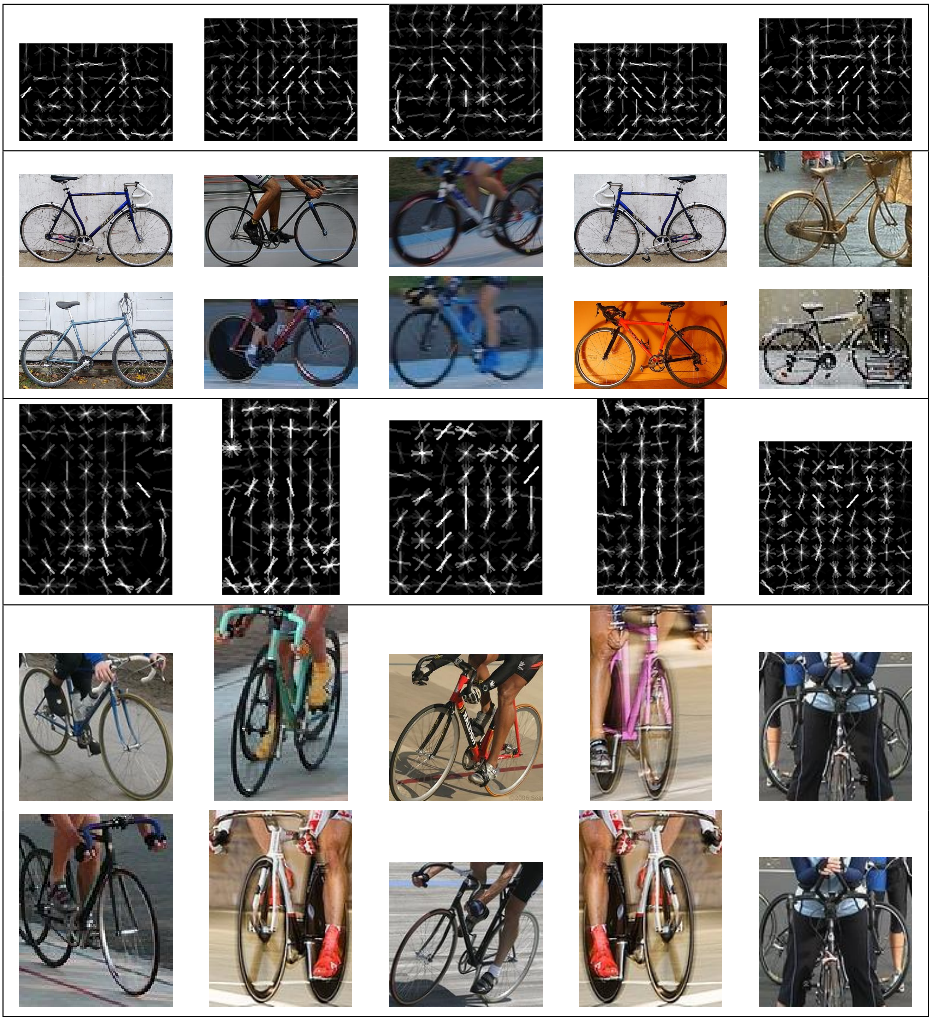


Figure 15: Filters learnt for bicycle class in the MCL step with 10 components (first and fourth rows). Below each component, the top two images associated with the component are depicted.

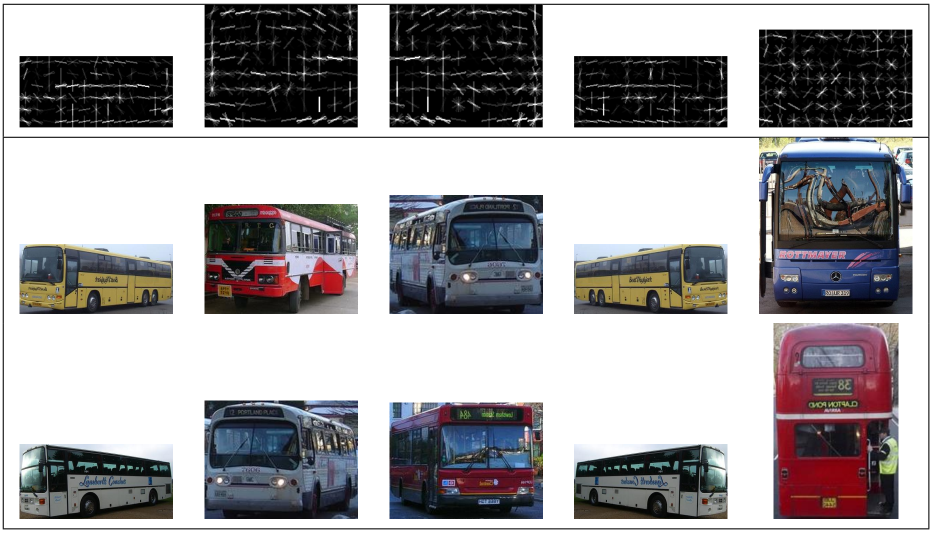


Figure 16: Filters learnt for bus class in the MCL step with 5 components (first row). Below each component, the top two images associated with the component are depicted.

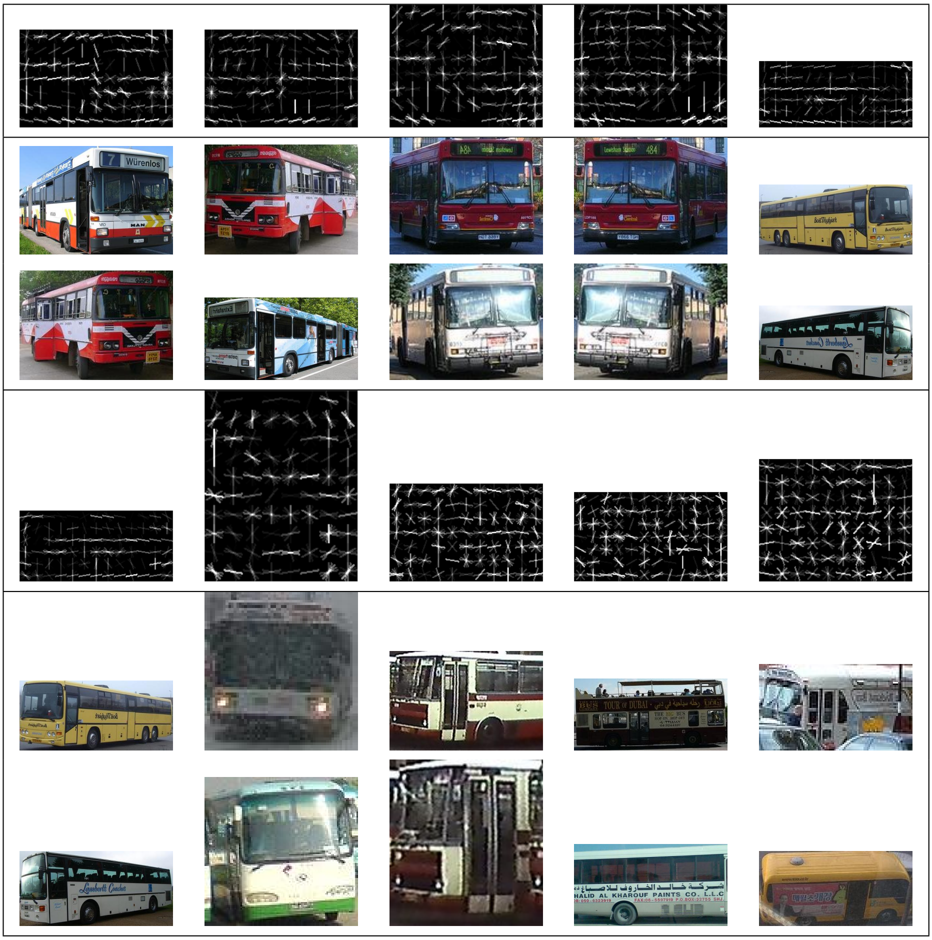


Figure 17: Filters learnt for bus class in the MCL step with 10 components (first and fourth rows). Below each component, the top two images associated with the component are depicted.

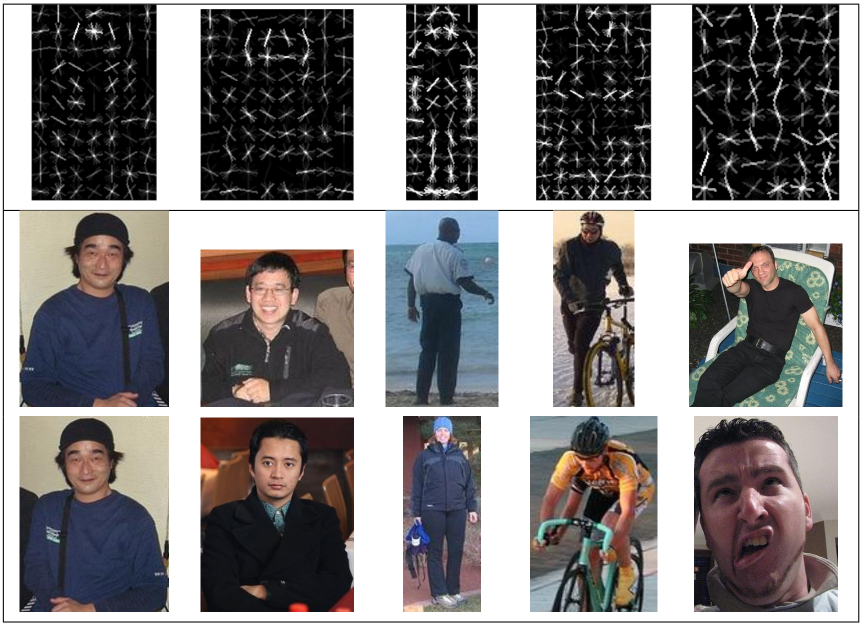


Figure 18: Filters learnt for person class in the MCL step with 5 components (first row). Below each component, the top two images associated with the component are depicted.

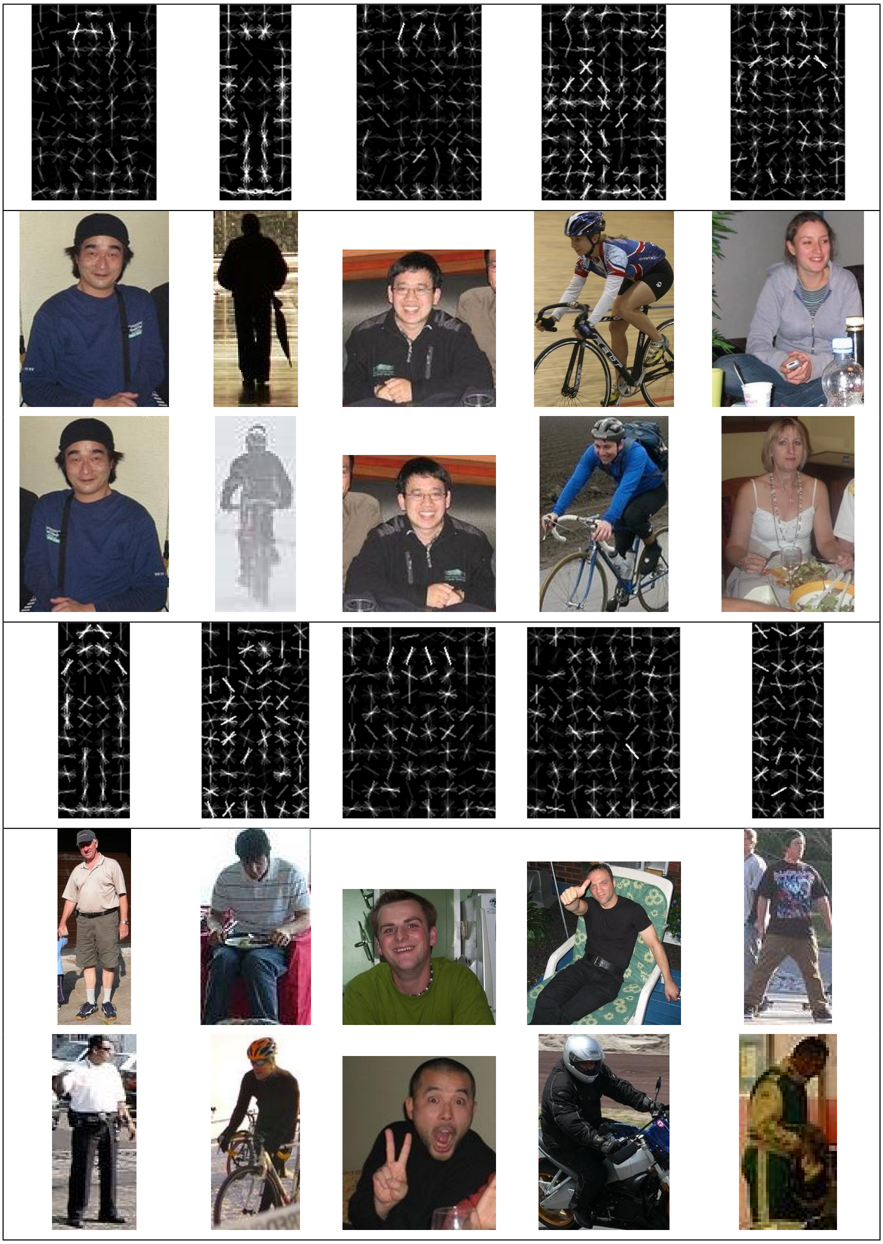


Figure 19: Filters learnt for person class in the MCL step with 10 components (first and fourth rows). Below each component, the top two images associated with the component are depicted.

Paper D

Properties of Training Data Predict the Performance of Classifiers

Omid Aghazadeh and Stefan Carlsson

Submitted to:
International Journal of Computer Vision

Properties of Training Data Predict the Performance of Classifiers

Omid Aghazadeh and Stefan Carlsson

Abstract

It has been shown that the performance of classifiers depends not only on the number of training samples, but also on the quality of the training set [20, 18]. The purpose of this paper is to 1) provide quantitative measures that determine the quality of the training set, and 2) provide the relation between the test performance and the proposed measures. We introduce data-describing measures that are derived from pairwise affinities between training exemplars of the positive class, and have a generative nature. We show that the performance of the state of the art methods, on the test set, can be reasonably predicted based on the values of the proposed measures on the training set. These measures open up a range of potential applications for visual recognition, enabling us to analyze the behavior of the learning algorithms w.r.t the properties of the training data. This will in turn enable us to devise rules for automatic selection of training data that maximize the quantified quality of the training set and thereby improve recognition performance.

1 Introduction

The most important component in the construction of modern classification algorithms has proved to be the data supplied, especially in terms of quantity [8]. While computer vision has benefited from more data over the years, as pointed out in [20], data has not had the same impact on computer vision field as on other fields such as text and speech. The main reason for this is believed to be the large intra-class variability of visual classes resulting from the variation in conditions under which images are created. However, no measure of intra-class variation has been proposed that can relate to the performance of classifiers.

Intra-class variation results in complex distributions of the data, which in turn result in non-linear decision boundaries between the classes. The overlap between these distributions, together with the assumptions of models about the data, results in non-separability of the classes. We have observed many advancements in

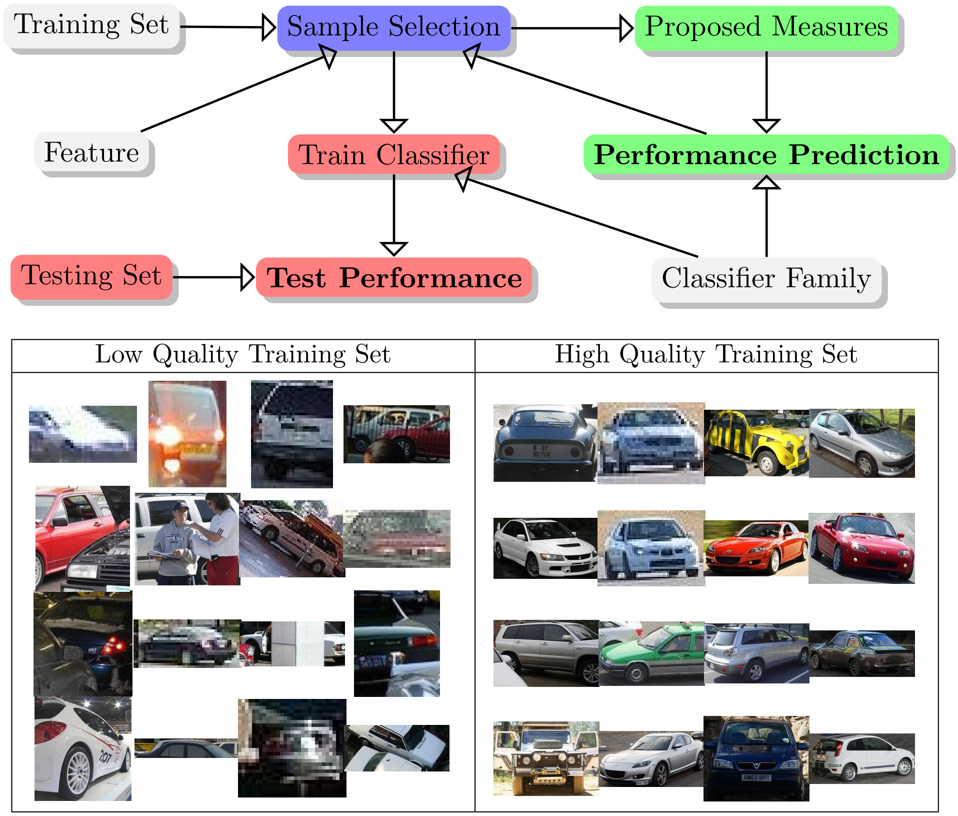


Figure 1: Top: illustration of the proposed procedure. The direction of arrows reflects the information flow and the dependencies. The red boxes comprise the traditional training/testing procedure while the green boxes are proposed in this paper. Bottom: (right) illustration of automatic sample selection (the blue box) using the HOG feature. The low quality set (left) is intentionally generated for comparison. Both set are automatically generated from the "car" class of Pascal VOC 2007, using the measures proposed in this paper.

modelling the non-linearity of the decision boundaries [19, 7, 16, 1]. However, identifying and alleviating the effect of outliers has not got the same attention – at least in SVM based formulations. Models are expected to automatically identify and ignore the resulting outliers – as optimizing the 0-1 loss would naturally do – despite the fact that the popular hinge loss is affected by gross outliers [20].

It has generally been assumed that increasing the size of the training set would overcome these problems. Some observations however seem to contradict this. [20] challenges the idea that more training data always leads to better performance. For a selection of state of the art ($s.o.a.$) classifiers, it is demonstrated that performance can decrease, which is attributed to the increased inclusion of outliers that distort the classification decision boundary. It is then suggested that "clean" data is crucial for current s.o.a learning algorithms, but no automatic way of obtaining clean data was proposed.

Related to this is the fact that performance of classification in benchmark tests such as Pascal-VOC is highly dependent on class and does not correlate well with the amount of data. The question then arises: What properties of the distribution of the exemplars in these classes are responsible for this? Is it possible to come up with measures based on the distributions that would predict the classification performance?

The fact that the distribution of training data can influence the performance of classification has been demonstrated in a dramatic way in [18] where it is pointed out that most data sets are biased in the sense that classifiers trained on a specific data set do not perform as well on other data sets. This is often a consequence of the fact that these data sets were collected with a specific objective in mind, but even the sets designed for the specific purpose of evaluating classification algorithms such as Pascal-VOC suffer from this. The authors propose cross-data set recognition performance as a measure of the bias of a data set. Such a measure will reflect the similarity between the distributions of samples in the training set of the source data set and that of the test set of the target data set. Despite the plausibility of such a measure, it has a few shortcomings. Firstly, it is model dependent in that a specific model needs to be trained and tested across data sets and unless this is to be exploited directly [10], it is not a desirable property. Secondly, the discriminative measure does not provide guidelines for *automatic sample selection* in order to avoid such biases.

It is therefore the objective of this paper to

1. quantify the properties of the training data such as intra-class variation

2. analyze how performance of s.o.a. classifiers vary with such measures and provide insight on the interplay between properties of training sets and the performance of classifiers.

This approach is generative in the sense that it makes predictions based on its descriptions of the positive training set. Such a generative approach – in contrast to the discriminative approach of [18] – will naturally and *automatically* determine what [20] refers to with “cleanness” of the data. In a longer perspective, it will allow us to devise rules of selection of data and classifier models that will maximize classification performance. In other words, we propose to consider data selection procedures as an active tool for the construction of classifiers. Figure 1 visualizes this.

The rest of this paper is organized as follows. A review of related works is given in section 1.1. Section 2 describes the model we use to quantify the quality of the training set, and the assumptions the model makes about the data. In section 3 we propose to use this model to improve the training set. Experimental setup and results are given in section 4. We discuss the proposed model, its assumptions and limitations in section 5. Section 6 concludes this paper.

1.1 Related Works

It is generally realized that there is a strong relation between intra-class variation and classification performance. Many works have been proposed that more or less trade coverage of the visual class for recognition performance. Modelling subclasses, e.g. ‘frontal car’ instead of the entire ‘car’ class, reduces the intra-class variation and results in better classification / localization performance.

Poselets [2] have been proposed which model specific body parts in specific configurations, which are tight in the image appearance and consequently result in modelling very little intra-class variation. Similarly, visual phrases [17] were shown to improve the recognition performance of classifiers which modelled two or more classes in a specific relation, in comparison to modelling the two classes independently. This specific relation between two or more classes, e.g. ‘person riding a bike’ or ‘person lying in beach’, restricts the intra-class variation of the modelling process; hence the observed improvements in the recognition. Relational phraselets [5] is closely related to visual phrases and benefits from the same facts. In [15], it was shown that avoiding the modelling of ‘cat’ and ‘dog’ bodies – and only modelling their faces – results in significantly superior recognition performances. Such a procedure reduces the intra-class variation, and consequently results in better test performances.

Despite all these efforts, no automatic way has been proposed to identify a procedure which automatically reduces intra-class variation. The main reason for this has to do with the fact that no measure has been proposed to quantify intra-class variation in relation to classification performance. Latent mixture models [7] has been proposed that dedicate mixture components to tight clusters, each modelling a part of intra-class variation [1]. Such an approach is model-dependent and initialization sensitive [1]. Furthermore, while it is copes with intra-class variation better than a single component SVM [4], it does not rectify the effects of intra-class variation.

Also related to our work is how algorithms deal with outliers. Boosting based approaches, unless a non-convex loss function which is robust to outliers is utilized, are sensitive to outliers [11]. SVM based approaches were argued to be affected by gross outliers [20], unless a proper non-convex loss is utilized. Latent mixture models [7] were argued to be able to assign outliers to a 'junk component', and leaving the rest of the components unaffected by them [1, 20]. However, the same argument as for the intra-class variation applies here.

We propose a model based approach for measuring intra-class variation and detecting gross-outliers. However, in contrast to existing approaches, the definition of intra-class variation and gross-outliers in our approach does not depend on specific classifier models. In other words, our model defines outliers in relation to other exemplars in a set, in contrast to defining outliers as those which cannot be explained by particular classifiers. Consequently, our approach does not need to train classifiers for every set, and verify explainability of exemplars through the trained classifiers. The result is a more efficient and more flexible way of analyzing training sets and determining outliers.

2 Quantifying the Quality of a Training Set

In this section we describe the proposed procedure. We start by motivating the use of local pairwise similarity measures and emphasizing the necessity of feature selection in section 2.1. Section 2.2 describes how to measure statistical properties of the training set at multiple scales. We elaborate on how these measures can be linked to the quality of the training set – and thus to test performance – in section 2.3.

2.1 Measuring Visual Structural Similarity via Discriminative Feature Selection

Ideally, in order to characterize the statistical properties of a visual class, one would like to measure the distribution of a feature vector that contains information relevant only to the class and discards all kinds of clutter contained in an image. This would however require a perfect method of feature selection which is not available. The best alternative is to assess local properties of the manifold of image exemplars within the class. Global properties then have to be inferred from the integration of these local characterizations. We will show later, in the experiments section, that the integration of these local properties does not amplify "noise", and it results in analysis more robust than it is solely based on the local properties.

The local analysis can be performed via the use of e.g. local pairwise affinities between exemplars in the data set. A similarity measure can be said to be *local* when it returns a high value if an only if the structure is sufficiently and significantly similar between the two exemplars. The RBF kernel is an example of a local similarity measure that does not perform feature selection.

Similarity should ideally refer to similarity at the level of visual class which requires a complete localization and extraction of the image content related to the class under consideration. This is an extremely complex task by itself, and we will restrict ourselves to a more limited objective that aims to enhance the contribution of visual class to the similarity measure.

The class specific visual similarity measures introduced in [1] use the calibrated exemplar SVMs [12] to perform feature selection when evaluating similarities. The measures are based on the modified version of the HOG feature [4] introduced in [7]. The exemplar SVM weights tend to "push the positive example as far away from the negative data as possible"; thus reasoning out the background, clutter and the noise in the HOG representation. Due to the specific type of feature selection in the similarity measures of [1], namely the projection of y onto the exemplar SVM weight of x, the distance between x and y is lost in a locality preserving way. Such visual similarity measures tend to have a high value if and only if x and y both have the same structure, hence the name visual structural similarity and the locality preserving property.

Using such visual similarity measures, the search for correspondences can be restricted in an unsupervised manner in contrast to the supervised framework of [20]. Consequently, we make use of the $K^{E}_{\text{MMI}}(.,.)$ measure [1] and exploit the aforementioned properties of the similarity measure. We have made the evaluated similarity measure on Pascal VOC 2007 publically available at `http://www.csc.kth.se/~omida/wearable/MMOR/Clustering_NN_Results_ESVM_HOGL_MI_MMI.html`.

2.2 Multi-Scale Analysis of the Data

Discriminative analysis requires specification of a positive and a negative set. The discriminative feature selection embedded in the class specific similarity measure - through the use of exemplar SVMs - already knows what does not belong to classes, locally in the space. As a result, by measuring only the (locally) discriminative properties of the positive set, we will implicitly model the properties of the negative set. Therefore, in the rest of this paper, we will concentrate on the properties and descriptions of the positive set, and only implicitly model the negative set.

Given a class $\mathcal{C}$ with n positive samples $\mathcal{C} = \{p_1, ..., p_n\}$ and a pairwise similarity measure $K^{(\mathcal{C})}(.,.) \in [0,1]$ we analyze the data on local, semi-local and global scales. On each scale, we measure the first and second order statistics – mean and variance – of different quantities that are described below. For the sake of brevity, we drop the superscript $(\mathcal{C})$ in the following whenever possible.

2.2.1 Local Scale

On the local scale, the quantity in question is the similarity of a sample to its nearest neighbor where the nearest neighbor is defined as the most similar sample. Formally, we define

$$K_L(p_i) = \max_{p_j \neq p_i} K(p_i, p_j) \tag{1}$$

to be a measure of local connectivity of p_i to its nearest neighbor. Therefore, the first and second moments on this scale

$$\begin{aligned} \mu_L &= \tfrac{1}{n} \textstyle\sum_{i=1}^{n} K_L(p_i) \\ \sigma_L^2 &= \tfrac{1}{n} \textstyle\sum_{i=1}^{n} K_L(p_i)^2 - \mu_L^2 \end{aligned} \tag{2}$$

roughly measure the average connectivity and average variation of connectivity around positive samples.

2.2.2 Semi-Global Scale

The moments on the semi-global scale collect statistics of the pairwise similarity values. Therefore, the moments

$$\begin{aligned} \mu_S &= \tfrac{1}{n^2} \textstyle\sum_{i=1}^{n} \sum_{j=1}^{n} K(p_i, p_j) \\ \sigma_S^2 &= \tfrac{1}{n^2} \textstyle\sum_{i=1}^{n} \sum_{j=1}^{n} K(p_i, p_j)^2 - \mu_S^2 \end{aligned} \tag{3}$$

compute global statistics of all the pairwise (local) similarity values; hence the name semi-global. The further the points are from each other, the smaller these measures become. Hence, what is measured on this scale is the (lack of) intra-class variation.

2.2.3 Global Scale

On this scale, the goal is to measure how points are distributed globally w.r.t each other. This involves measuring the distance between points that might be far away. Due to the locality property, the similarity measure loses information about large distances between points. Therefore, we have to resort to multiple local steps to approximate the global distance. We suggest the following procedure to compute the geodesics distance, on the manifold of the positive training exemplars, without explicitly specifying the manifold.

We construct a full graph with each node corresponding to one positive training exemplar. An edge between p_i and p_j in the graph are weighted according to

$$w_{ij} = D_L(p_i, p_j) = 1 - K(p_i, p_j) \tag{4}$$

When p_i and p_j are very similar, the weight between them is going to be very small and the shortest path will be the one directly from p_i to p_j. However, when points are far away, the direct path has a large value, and is unlikely to be the shortest path between the two. In such cases, the shortest path will take multiple steps, using low-valued edges which correspond to highly similar exemplar. Consequently, the shortest path between points on such a graph approximates the geodesics distances on an implicit manifold of image exemplars.

Let $\mathcal{P}(p_i, p_j)$ refer to the set of all paths between p_i and p_j. The global distance between the two points as determined by $\mathbf{p} \in \mathcal{P}(p_i, p_j)$ is

$$D_P(\mathbf{p}) = \sum_{k=2}^{\dim \mathbf{p}} D_L(\mathbf{p}_{k-1}, \mathbf{p}_k) \tag{5}$$

Measure	Scale	Semantic
μ_L	Local	Connectivity
μ_S	Semi-Global	Lack of Variation
μ_G	Global	Intra-Class Variation
μ_P	Global	Connected Variation

Table 1: Semantics of the first order moments.

Let $\mathbf{s}(p_i, p_j)$ refer to the shortest path between p_i and p_j:

$$\mathbf{s}(p_i, p_j) \quad = \underset{\mathbf{p} \in \mathcal{P}(p_i, p_j)}{\arg\min} \; D_P(\mathbf{p}) \tag{6}$$

Also let $P_G(p_i, p_j)$ refer to the length of $\mathbf{s}(p_i, p_j)$, and $D_G(p_i, p_j)$ refer to the global distance between p_i and p_j – as approximated by the shortest path:

$$\begin{aligned} P_G(p_i, p_j) &= \dim \mathbf{s}(p_i, p_j) - 1 \\ D_G(p_i, p_j) &= D_P\left(\mathbf{s}(p_i, p_j)\right) \end{aligned} \tag{7}$$

The moments

$$\begin{aligned} \mu_G &= \tfrac{1}{n^2} \textstyle\sum_{i=1}^{n} \sum_{j=1}^{n} \; D_G(p_i, p_j) \\ \sigma_G^2 &= \tfrac{1}{n^2} \textstyle\sum_{i=1}^{n} \sum_{j=1}^{n} \; D_G(p_i, p_j)^2 - \mu_G^2 \end{aligned} \tag{8}$$

measure how far the points are away from each other. As a result, they measure intra-class variation.

Similarly, the moments

$$\begin{aligned} \mu_P &= \tfrac{1}{n^2} \textstyle\sum_{i=1}^{n} \sum_{j=1}^{n} \; P_G(p_i, p_j) \\ \sigma_P^2 &= \tfrac{1}{n^2} \textstyle\sum_{i=1}^{n} \sum_{j=1}^{n} \; P_G(p_i, p_j)^2 - \mu_P^2 \end{aligned} \tag{9}$$

measure the number of linked local steps between pairs of points. Two factors affect these measures:

1. global distances between pairs of points

2. the number of exemplars which help linking multiple local steps to decrease the global distance between points

Consequently, these measures reflect ‘connected intra-class variation’.

Table 1 summarizes the mentioned semantics. Figures 2 and 3 depicts two set for each first order moment; one with a high value and one with a low value. The contrast between these pairs of sets qualitatively verifies the assigned semantics. We will describe later in section 3 how to generate such sets automatically.

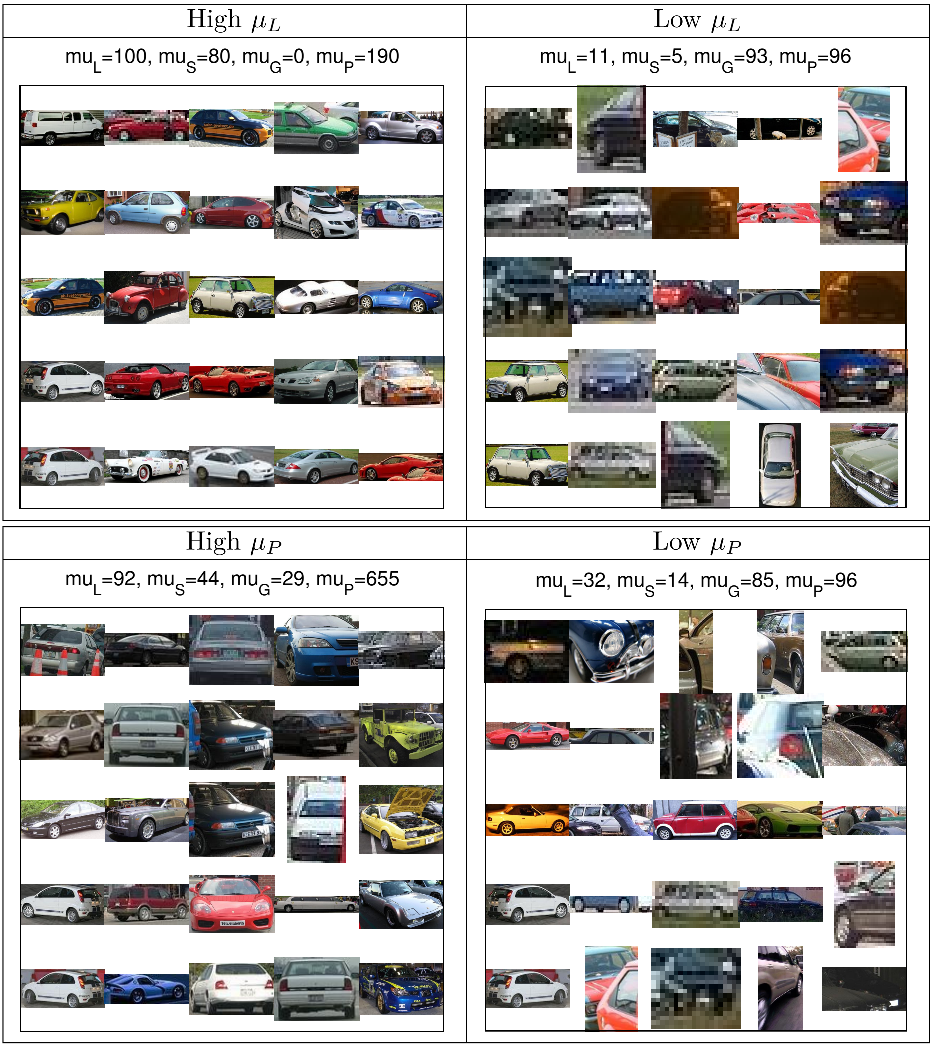
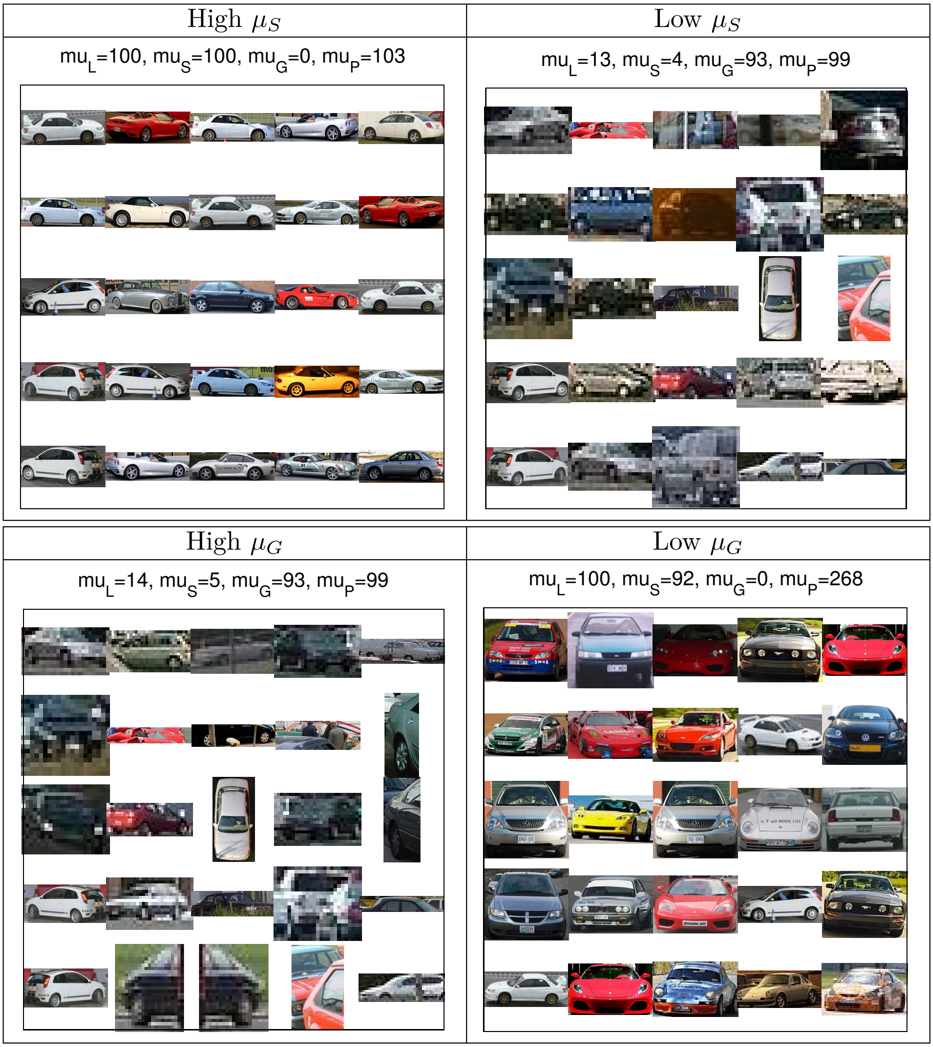

High μ_L
mu$_L$=100, mu$_S$=80, mu$_G$=0, mu$_P$=190
Low μ_L
mu$_L$=11, mu$_S$=5, mu$_G$=93, mu$_P$=96
High μ_P
mu$_L$=92, mu$_S$=44, mu$_G$=29, mu$_P$=655
Low μ_P
mu$_L$=32, mu$_S$=14, mu$_G$=85, mu$_P$=96

Figure 2: Demonstrations of sets with low and high first order connectivity measures. The measures are scaled by a factor of 10^2 for better readability.



High μ_S

mu$_L$=100, mu$_S$=100, mu$_G$=0, mu$_P$=103

Low μ_S

mu$_L$=13, mu$_S$=4, mu$_G$=93, mu$_P$=99

High μ_G

mu$_L$=14, mu$_S$=5, mu$_G$=93, mu$_P$=99

Low μ_G

mu$_L$=100, mu$_S$=92, mu$_G$=0, mu$_P$=268

Figure 3: Demonstrations of sets with low and high semi-global and global variation. The measures are scaled by a factor of 10^2 for better readability.

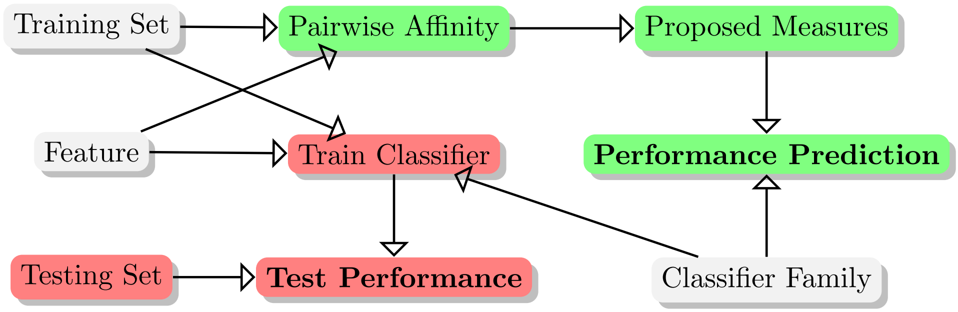


Figure 4: The training-testing process (red boxes) and the proposed test performance prediction process (green boxes). The direction of arrows determines the flow of information and also the dependencies. Both procedures are dependent on the white boxes.

2.3 Test Performance Prediction by Analyzing the Training Set

In this section, starting from a formalization of the usual training-testing process, we will derive an expression which will relate a description of the training set to the test performance. We then use the measured moments as descriptions of the training sets and establish the relation between the proposed measures and the test performance. Figure 4 visualizes this.

Consider a family of models $\mathcal{M}$ e.g. the DPM of [7]. Let $M(\mathcal{C}) \in \mathcal{M}$ refer to the process of training a model from the family $\mathcal{M}$ on the set $\mathcal{C}$. Also let the process of testing such a model on a test set $\mathcal{C}_{TST}$ – resulting in average precision $AP_{\mathcal{M}}^{(\mathcal{C})}$ – be described by

$$AP_{\mathcal{M}}^{(\mathcal{C})} = \tau\left(M(\mathcal{C}_{TR}), \mathcal{C}_{TST}\right) \tag{10}$$

where $\tau(M, \mathcal{C})$ evaluates the model M on $\mathcal{C}$ i.e. the detection process. Let $\mu^{(\mathcal{C})} \in \mathbb{R}^8$ be the vector of moments computed on a set $\mathcal{C}$. If a function $\hat{f}_{\mathcal{M}}(.,.)$ can be found that is associated with a small approximating errors in

$$AP_{\mathcal{M}}^{(\mathcal{C})} = \hat{f}_{\mathcal{M}}\left(M(\mathcal{C}_{TR}), \mu^{(\mathcal{C}_{TST})}\right) + \epsilon_{\hat{f}_{\mathcal{M}}} \tag{11}$$

then we can say that $\mu^{(\mathcal{C}_{TST})}$ is a good description of the test set.

The trained model $M(\mathcal{C}_{TR}) \in \mathcal{M}$ depends on the training set $\mathcal{C}_{TR}$ and the classifier family $\mathcal{M}$ – observable also in figure 4. Replacing the dependency on the training set with a description of the training set we get

$$AP_{\mathcal{M}}^{(\mathcal{C})} = f_{\mathcal{M}}\left(\mu^{(\mathcal{C}_{TR})}, \mu^{(\mathcal{C}_{TST})}\right) + \epsilon_{f_{\mathcal{M}}} \tag{12}$$

Both $f_{\mathcal{M}}(\mu^{(\mathcal{C}_{TR})}, \mu^{(\mathcal{C}_{TST})})$ and $\hat{f}_{\mathcal{M}}(M(\mathcal{C}_{TR}), \mu^{(\mathcal{C}_{TST})})$ have the same dependencies as they both already depend on $\mathcal{M}$.

Assuming what the empirical risk minimization approaches assume – that the training set and the test set are drawn from the same distribution – we approximate the description of the test set by that of the training set i.e. $\mu^{(\mathcal{C}_{TST})} \approx \mu^{(\mathcal{C}_{TR})}$.

Hence, we can say that if there exists $\tilde{f}_{\mathcal{M}} : \mathbb{R}^8 \rightarrow [0, 1]$ such that the prediction error $|\epsilon_{\tilde{f}_{\mathcal{M}}}|$ is sufficiently small for a variety of classes where

$$AP_{\mathcal{M}}^{(\mathcal{C})} = \tilde{f}_{\mathcal{M}}\left(\mu^{(\mathcal{C}_{TR})}\right) + \epsilon_{\tilde{f}_{\mathcal{M}}} \tag{13}$$

then:

1. $\mu^{(\mathcal{C})}$ is a reasonably accurate description of $\mathcal{C}$.

2. $\tilde{f}_{\mathcal{M}}(.)$ establishes the relation between test performance and the proposed measures.

Let $\mathcal{R} = \{\mathcal{M}_1, ..., \mathcal{M}_r\}$ denote a set of family of models and
$\mathbf{v}^{(\mathcal{C})} = \left(f_1^{(\mathcal{C})}, \ldots, f_{n_v}^{(\mathcal{C})}\right)^T$; $f_i^{(\mathcal{C})} : \mathbb{R}^8 \rightarrow \mathbb{R}$ be a vector of n_v predictors where each predictor is a function of the 8 measured moments.

We now search for $\bar{f}_{\mathcal{R}} : \mathbb{R}^{n_v} \rightarrow \mathbb{R}$ which minimizes the average L_2 norm of the prediction errors $\epsilon_{\bar{f}_{\mathcal{R}}}$[1]. We assume a sigmoid structure for $\bar{f}_{\mathcal{R}}$ which is linear in $\mathbf{v}$

$$\bar{f}_{\mathcal{R}}(\mathbf{w}_{\mathcal{R}}; \mathbf{v}) = \left(1 + \exp\left\{-\mathbf{w}_{\mathcal{R}}^T \mathbf{v}\right\}\right)^{-1} \tag{14}$$

Given a data set $\mathcal{D} = \{\mathcal{C}_1, \ldots, \mathcal{C}_D\}$, we solve for $\mathbf{w}_{\mathcal{R}}^{(\mathcal{C}_{CV})} = \arg\min_{\mathbf{w}} \mathcal{L}(\mathbf{w}, \mathcal{C}_{CV})$ where

$$\begin{aligned}\mathcal{L}(\mathbf{w}, \mathcal{C}_{CV}) =& \lambda \|\mathbf{w}\|^2 \\ &+ \sum_{\mathcal{M} \in \mathcal{R}} \sum_{\mathcal{C} \in \mathcal{D} \setminus \{\mathcal{C}_{CV}\}} \|AP_{\mathcal{M}}^{(\mathcal{C})} - \bar{f}_{\mathcal{R}}(\mathbf{w}; \mathbf{v}^{(\mathcal{C})})\|^2\end{aligned} \tag{15}$$

Afterwards, $\mathbf{w}_{\mathcal{R}}^{(\mathcal{C}_{CV})}$ is used to predict the test performance for $\mathcal{C}_{CV}$ and this cross-validating procedure is performed for all $D = 20$ classes of Pascal VOC 2007 [6]. We also add a bias term to (15) – which was omitted here for the sake of clarity – and found $\lambda = 10^{-3}$ to be optimal after centering and normalizing the predictors.

3 Constrained Dataset Selection

In this section, we consider data selecting procedures based on the measures proposed in section 2.2.

[1] The reason for the $\mathcal{R}$ subscript – instead of $\mathcal{M}$ in (13) – is the dependency of the function $\bar{f}_{\mathcal{R}}(.)$ on a set of families of models $\mathcal{R}$ rather than one particular family $\mathcal{M}$.

3.1 Greedy Set Selection

Let $\mu^{(\mathcal{S})}$ refer to a vector of 8 moments evaluated on the set $\mathcal{S} \subseteq \mathcal{C}$. Given a desired criterion $g : \mathbb{R}^8 \to \mathbb{R}$, we search for the set $\mathcal{S}$ which optimizes

$$s(\mathcal{S}) = g\left(\mu^{(\mathcal{S})}\right) \tag{16}$$

The optimization is combinatorial, and therefore we resort to greedy optimization procedures. In each step of optimization, we can either add exemplars to or delete exemplars from $\mathcal{S}$. We consider two types of scenarios, each with a specific type of constraint:

1. *Fixed Cardinality*: the goal is to optimize

$$\begin{array}{rcl} \mathcal{S}_F & = & \underset{\mathcal{S}\subseteq\mathcal{C}}{\arg\max}\, s(\mathcal{S}) \\ \text{s.t.} & & |\mathcal{S}| = n_f \end{array} \tag{17}$$

The optimization process in this case is initialized with a small set, and exemplars are greedily added to the set according to

$$\mathcal{S}_A^{(k+1)} \quad = \mathcal{S}_A^{(k)} \cup \left\{ \underset{i\in\mathcal{C}\setminus\mathcal{S}_A^{(k)}}{\arg\max}\, s\left(\mathcal{S}_A^{(k)} \cup \{i\}\right) \right\} \tag{18}$$

As most measures require at least 3 samples, we initialize $\mathcal{S}_A^{(2)}$ with the two samples with maximal $K_L(.)$s (1). $\mathcal{S}_A^{(n_f)}$ determines the solution to the fixed cardinality problem.

2. *Largest Set*: the goal is to optimize

$$\begin{array}{rcl} \mathcal{S}_L & = & \underset{\mathcal{S}\subseteq\mathcal{C}}{\arg\max}\, |\mathcal{S}| \\ \text{s.t.} & & s(\mathcal{S}) \geq \tau \end{array} \tag{19}$$

The optimization process in this case is initialized with $\mathcal{S}_R^{(n)} = \mathcal{C}$, and exemplars are greedily removed from the set according to

$$\mathcal{S}_R^{(k-1)} \quad = \mathcal{S}_R^{(k)} \setminus \left\{ \underset{i\in\mathcal{S}_R^{(k)}}{\arg\max}\, s\left(\mathcal{S}_R^{(k)} \setminus \{i\}\right) \right\} \tag{20}$$

The biggest k for which $s(\mathcal{S}_R^{(k)}) \geq \tau$ is satisfied determines the solution $\mathcal{S}_R^{(k)}$ to the largest set problem.

For example, figures 2 and 3 were generated using fixed cardinality optimization procedure.

3.2 Subsets that Maximize Test Performance

The relation between test performance, training set and test sets were formalized in section 2.3 (12). We assumed similar distributions of the exemplars in the training set and the test set, which resulted in (13). This assumption is less correct as the training set becomes more dissimilar to the test set. This will usually be the case when the training set is modified and the test set is kept fixed. Therefore, two scenarios can happen in general:

1. The testing set is modified in the same manner as the training set. The assumption is valid and (13) can be used.

2. The testing set is kept fixed. It has to be modelled e.g. by (12), in order to be able to select a training set which suits it.

The first scenario has a trivial solution in absence of extra constraints: the less variation in the training set, the more the predicted performance. Therefore, we consider the second scenario.

Given a training set $\mathcal{C}_{TR}$, the goal here is to find a subset which improves the test performance. Let $\mathcal{S}_P \subseteq \mathcal{C}_{TR}$ refer to the desired subset. Similar to section 2.3 we assume $\mu^{(\mathcal{C}_{TR})} \approx \mu^{(\mathcal{C}_{TST})}$. If a function such as $f_{\mathcal{M}}(.,.)$ (12) is available which relates the generalization performance of a description of a set to another one's, we could find the subset via

$$\mathcal{S}_P = \underset{\mathcal{S} \subseteq \mathcal{C}_{TR}}{\arg\max} f_{\mathcal{M}} \left(\mu^{(\mathcal{S})}, \mu^{(\mathcal{C}_{TR})} \right) \tag{21}$$

In absence of such a function, we propose an alternative approach. If the description of the training set is similar to the description of the subset, that is if $\mu^{(\mathcal{C}_{TR})} \approx \mu^{(\mathcal{S})}$, we can replace the dependency on $\mu^{(\mathcal{C}_{TR})}$ and model the test performance as a function of the description of the evolving set:

$$\begin{array}{rcl} \mathcal{S}_P & = & \underset{\mathcal{S} \subseteq \mathcal{C}_{TR}}{\arg\max} \tilde{f}_{\mathcal{M}} \left(\mu^{(\mathcal{S})} \right) \\ \text{s.t.} & & \| \mu^{(\mathcal{C}_{TR})} - \mu^{(\mathcal{S})} \| \leq \epsilon_\mu \end{array} \tag{22}$$

Obviously, such an approach will be valid only when $\mathcal{S}_P$ and $\mathcal{C}_{TR}$ are similar i.e. when $\epsilon_\mu \approx 0$. In other words, extrapolations based on this approximate approach becomes more and more uncertain as ϵ_μ becomes bigger. Therefore, we restrict the study to small changes in the training set. A small change in the training set can be encoded by a constraint on the cardinality of $\mathcal{S}_P$, or via a constraint on the change in the predicted test performance. We choose the latter and optimize for

$$\begin{array}{rcl} \mathcal{S}_P & = & \underset{\mathcal{S} \subseteq \mathcal{C}_{TR}}{\arg\max} \bar{f}_{\mathcal{R}} \left(\tilde{\mathbf{w}}_{\mathcal{R}}; \mathbf{v}^{(\mathcal{S})} \right) \\ \text{s.t.} & & \bar{f}_{\mathcal{R}} \left(\tilde{\mathbf{w}}_{\mathcal{R}}; \mathbf{v}^{(\mathcal{S})} \right) - \bar{f}_{\mathcal{R}} \left(\tilde{\mathbf{w}}_{\mathcal{R}}; \mathbf{v}^{(\mathcal{C}_{TR})} \right) \leq \epsilon_P \end{array} \tag{23}$$

where – similar to section 2.3 – $\mathbf{v}^{(\mathcal{C})} : \mathbb{R}^8 \to \mathbb{R}^{n_v}$ is a vector of predictors describing $\mathcal{C}$, and

$$\tilde{\mathbf{w}}_{\mathcal{R}} = \arg\min_{\mathbf{w}} \lambda \|\mathbf{w}\|^2 + \sum_{\mathcal{M}\in\mathcal{R}} \sum_{\mathcal{C}\in\mathcal{D}} \|AP^{(\mathcal{C})}_{\mathcal{M}} - \bar{f}_{\mathcal{R}}(\mathbf{w}; \mathbf{v}^{(\mathcal{C})})\|^2 \tag{24}$$

We solve (23) via a largest set greedy optimization (section 3.1) with

$$\begin{aligned} \tau &= \bar{f}_{\mathcal{R}}\left(\tilde{\mathbf{w}}_{\mathcal{R}}; \mathbf{v}^{(\mathcal{C}_{TR})}\right) + \epsilon_P \\ g\left(\mu^{(\mathcal{S})}\right) &= \bar{f}_{\mathcal{R}}\left(\tilde{\mathbf{w}}_{\mathcal{R}}; \mathbf{v}^{(\mathcal{S})}\right) \end{aligned} \tag{25}$$

4 Experiments

We provide regression and correlation analysis which determine the relation between the proposed measures and the test performance. We use *Spearman's rank correlation coefficient* (Spearman's ρ) [13] as it is non-parametric and thus, invariant to any monotonic transformation of the variables. This makes Spearman's ρ particularly useful for highlighting non-linear dependencies.

4.1 Reference Methods

The reference methods we have considered are the following:

1. (D4): deformable part based model of [7]. The results are of release 4 of the software without bounding box prediction and context re-scoring.

2. (D5): release 5 of DPM [7] with bounding box prediction and contextual re-scoring.

3. (RT): K^E_{MMI}+L+S+O [1] – a two scale mixture of rigid templates which relies on an oracle for the optimal number of fixed templates.

4. (RT10): K^E_{MMI}:10+L [1] – a single scale mixture of 10 rigid templates.

5. (E): exemplar SVM(ESVM) [12]. The co-occurrence re-calibration results are reported.

6. (CF): the coarse to fine part based model [16].

7. (LHSL): the latent 3-scale part based model of [19].

The average performance of the reference set based on 7 methods is 0.2899. Table 2 shows the test performance of the reference methods and figure 5 shows the correlation between their test performances.

	plane	bicycle	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	monitor	mAP
D4	30	57	10	17	25	48	55	18	22	25	23	11	58	48	42	12	19	32	45	41	32
D5	37	62	12	18	29	55	60	26	21	26	27	15	61	51	45	14	22	38	49	44	**35**
RT	33	54	10	16	23	49	52	16	16	20	24	11	55	44	37	11	23	24	39	41	30
RT10	29	50	10	15	19	41	50	10	16	21	17	10	50	40	33	9	20	22	38	34	27
E	21	48	8	14	13	40	41	5	12	19	11	3	45	39	17	11	23	17	37	30	23
CF	28	54	7	15	15	44	47	15	13	22	24	12	52	42	31	11	23	19	35	31	27
LHSL	29	56	9	14	29	44	51	21	20	19	25	13	50	38	37	15	20	25	37	39	30

Table 2: Test performance of the reference methods. Results are rounded off for better readability.

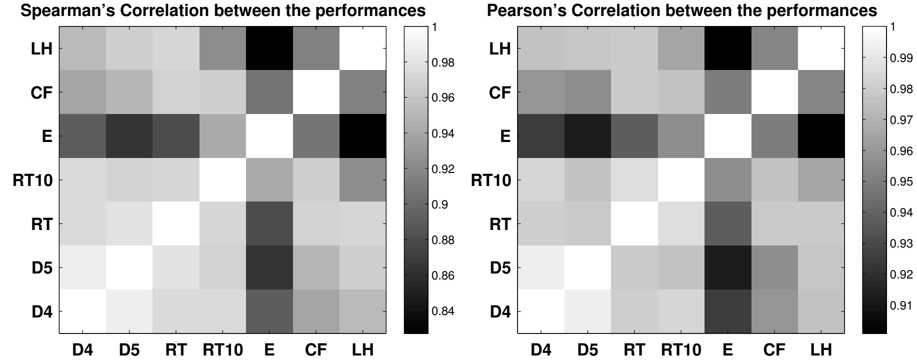


Figure 5: Correlations between test performance of reference methods. The average Spearman's correlation is 0.948 while the average Pearson's correlation is 0.967.

4.2 The Measured Moments

Figure 6 shows the Spearman's correlation and Pearson's correlation between the measures. It can be observed that the measures are correlated and that the dependencies are mostly linear. Particularly, the semi-global and global measures seem to be significantly correlated. We provide the following explanation for this observation.

Low global connectivity implies low-length shortest paths, which results in similarity of semi-global and global measures. In the extreme case – where the shortest paths are all of length 1 – global measures and semi-global measures will become the same. This mainly reflects the overall low global connectivity of the Pascal VOC 2007[2]. The strong correlation between local and global connectivity (μ_L and μ_P), (μ_S and μ_G) and (σ_S and σ_G) supports this hypothesis.

Table 3 shows the ordering that each measure induces on the classes of Pascal VOC 2007. It can be observed that the measured moments tend to more or less agree on the quality of the training set. For example, 'bird' is the class with the least local and global connectivity (μ_L and μ_P), and it exhibits the most intra-class variation ($1-\mu_S$ and μ_G). On the contrary, 'car' has the best one-nearest neighbors (local connectivity) and is ranked second in global connectivity (multiple nearest neighbors). It exhibits the least intra-class variation.

Table 4 shows the correlation between the performance of the reference methods and the proposed measures. It can be seen that the only factor that has a negative correlation with the objective, is the intra-class variation (μ_G). In absence of any other information, local and global connectivity (μ_L and μ_P) seem to have stronger effects on the test performance than bias or intra-class variation. Moreover, the

[2] We augmented Pascal VOC 2007 with its left-right flipped version. This essentially doubled the size of the dataset, and increased the connectivity measures. The conclusion regarding the overall low global connectivity would be even stronger without this modification.

	plane	bicycle	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	monitor
μ_L	14	19	**1**	9	10	15	**20**	3	8	11	2	4	18	16	13	5	7	6	12	17
σ_L	**20**	17	2	11	12	18	13	3	7	10	5	**1**	15	16	9	4	8	6	19	14
μ_S	5	18	**1**	7	2	17	**20**	6	12	11	4	13	15	14	16	3	8	9	10	19
σ_S	13	19	2	10	6	17	**20**	5	**1**	12	7	3	16	15	9	4	11	8	14	18
μ_G	15	2	**20**	14	17	6	**1**	16	9	10	18	8	4	5	7	19	12	13	11	3
σ_G	13	19	3	8	9	15	**20**	5	**1**	11	7	2	17	16	12	4	10	6	14	18
μ_P	12	**20**	**1**	9	11	16	19	4	7	10	3	6	18	17	13	2	8	5	14	15
σ_P	12	19	**1**	10	11	18	16	3	7	9	5	4	**20**	17	13	2	8	6	14	15

Table 3: Pascal VOC 2007 classes ranked w.r.t the proposed measures.

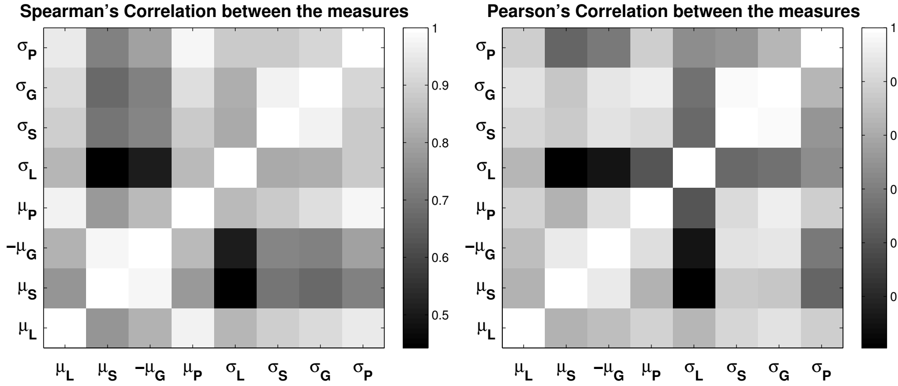


Figure 6: Spearman's Correlation and Pearson's Correlation between the measures. The average Spearman's correlation between the measures is 84.3 while the average Pearson's correlation is 82.9.

f	D4	D5	RT	RT10	E	CF	LHSL	mean	min
μ_S	71	70	71	75	68	71	68	70.5	67.5
μ_G	-75	-73	-74	-80	-74	-75	-71	-74.6	-71.1
σ_L	78	76	78	82	84	79	76	79.0	75.9
μ_L	88	85	86	90	90	86	85	87.2	85.0
σ_S	83	84	87	90	93	91	83	87.4	82.6
μ_P	90	89	89	93	90	90	87	89.6	87.1
σ_G	88	88	91	92	91	93	88	90.0	87.6
$\sigma_\mathbf{P}$	92	90	92	94	91	92	88	**91.3**	**88.3**

Table 4: Correlation of the measures with the performance of the reference methods.

second order moments seem to be more crucial to analyze than the first order ones. σ_P in absence of any other information is the best predictor of how much contemporary algorithms can learn from a class.

4.3 Test Performance Prediction by Analyzing the Training Set

Table 5 demonstrates the results of the approach proposed in section 2.3 using different predictors, shown on the top row. In the table, $\mathbf{m}_X$ refers to a vector of first and second order moments at scale X, together with their inverses. For example,

$$\mathbf{m}_{GP} = \left(\mu_G, \mu_P, \sigma_G, \sigma_P, \mu_G^{-1}, \mu_P^{-1}, \sigma_G^{-1}, \sigma_P^{-1}\right)^T$$

Also in the table, $\mathbf{v} = n$ refers to the number of positive training sample for each class used as a predictor of the test performance, and $\mathbf{v} = 1$ predicts the test performance of a class by averaging the other 19 observed test performances. The middle row shows the scaled root mean squared error (RMSE), while the average correlation to the performance of the reference methods is reflected in the bottom.

It can be observed that the size of the training set is a poor predictor of its quality. That the use of data-describing measures significantly improves the predictions, suggests that 1) the quality of the training set determines the test performance with a reasonable accuracy, and 2) $\bar{f}_{\mathcal{R}}(.)$ (14) quantifies the quality of the training data.

That the size of the training set does not quantify the quality of the training set, suggests that "big data" should meet some quality requirements in order to be useful for visual recognition – at least in case of HOG feature and linear classifiers. The same has been concluded in [20] where the "cleanness of data" was emphasized. Among the proposed measures, those based on the global scale analysis – connectivity and variation – seem to be able to explain the majority of the observed performances. As an evidence for this hypothesis we point out the superiority of the predictions based on the global measures – $\mathbf{m}_{GP}$ in table 5, and the strong correlation of these measures with the test performance of reference methods – as reflected in table 4. This hypothesis consequently suggests that *"big connected data"* will satisfy the quality constraints on "big data".

Furthermore, the global connectivity measures correlate stronger with the test performances and predict them better than the rest of the measures – observable in tables 4 and 5 . This suggests that the effects of intra-class variation can be rectified by ensuring good connectivity between samples. This also promotes the *"big connected data"* hypothesis.

Figure 7 shows the predicted test performances and the mean absolute error (MAE) of the predictions, using the $\mathbf{m}_{GP}$ predictor. While the relevance of the predicted performances is evident, there are variations in test performances that the predictions do not quite capture. Example of such cases are the D5 – the deformable part based model based on contextual re-scoring, and E – the exemplar SVM approach based on co-occurrence re-calibrations, which also utilizes contextual re-scoring. Part of this is due to the differences in how reference methods utilize training data. Table 6 shows model specific prediction of test performances where the same procedure as in section 2.3 is repeated for each reference method independently. As expected, dependency of each method on the data is best learnt by studying how the performance of the method itself depends on the data, in contrast to studying a reference set. On average, 0.04 AP of the test performances are not explained by the current procedure. More discussion on this is deferred to section 5.

4.4 Dataset Selection

Evaluating global measures on a set scales cubically with the cardinality of the set. Consequently, the use of global measures on largest set problems is prohibitive. The local measures were shown to be able to approximate the global measures reasonably well, in tables 4 and 5. Therefore, we base the analysis in this section

Criterion \ $\mathbf{v}$	$\mathbf{m}_L$	$\mathbf{m}_S$	$\mathbf{m}_G$	$\mathbf{m}_P$	$\mathbf{m}_{PL}$	$\mathbf{m}_{SG}$	$\mathbf{m}_{SL}$	$\mathbf{m}_{GP}$	$\mathbf{m}_{LSGP}$
10^3 RMSE	79	86	77	63	64	80	80	**62**	65
Corr to AP	87	84	89	88	89	88	86	**92**	92

Criterion \ $\mathbf{v}$	n	1
10^3 RMSE	171	159
Corr to AP	-82	-97

Table 5: Evaluation of test performance prediction based on all reference methods.

	D4	D5	RT	RT10	E	CF	LHSL
MAE	4.5	5.3	3.5	3.3	4.2	3.6	4.0
Corr	89.7	92.3	93.3	93.6	89.5	93.7	89.6

Table 6: Evaluation of test performance prediction specific to each reference method. Both measures are scaled by 10^2 for better readability.

on the use of local measures, which scale quadratically with the size of the training set.

The use of local measures results in a cubic overall complexity of each iteration of the largest set optimization and is still prohibitive for the ‘person’ class with 9380 samples. Hence, we sub-sample from the person class a total of 2500 randomly chosen samples. This inevitably violates the assumption about small modifications of the original training set, unless the randomly selected 2500 samples represent the test set as accurately as the 9380 samples.

Figures 8 and 9 (right) depict the dataset selection procedure for $\epsilon_P = 0.01$. For comparison, the same figures (left) depict exemplars which upon removal from the training set result in 1% decrement in predicted test performances. It can be observed that gross outliers and strong inliers can be identified by this approach, without training and testing specific models. Qualitatively, most gross outliers either

1. are significantly truncated
2. are significantly occluded
3. are taken from a significantly low quality image, are noisy or too small
4. have been captured from viewpoints which do not have enough “support” in the training set.

The latter is related to photographer and selection biases discussed in [18].

Figure 10 shows the changes the automatic dataset selection induces on the training set. $\Delta|\mathcal{C}|$ refers to the number of exemplars removed from the training set, and $\Delta AP(G)$ refers to the predicted change in the test performance when all

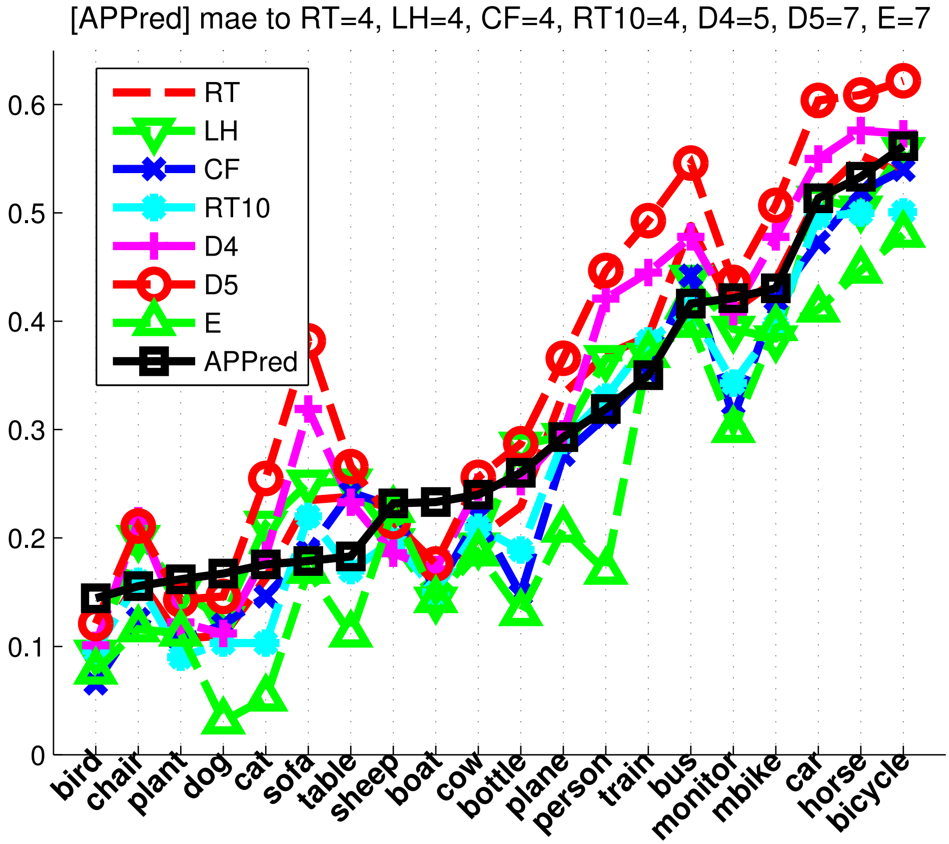


Figure 7: *Test Performance Prediction* of Pascal-VOC 2007 classes ('APPred') and the performance of the reference methods. Best viewed electronically and in color.

the measures are used as predictors. According to the predictions made by local measures, on average 55 samples have to be removed from the training sets to achieve 1% improvements in AP. According to not only the local measures, but also the semi-local and global ones, the actual change is going to be 0.7%. It can be observed that the selected subsets consistently result in better local and global connectivity (μ_L and μ_P), and less semi-global and global intra-class variation (μ_S and μ_G).

Decrements in intra-class variation are to be expected when exemplars are removed from training set. Exemplars which are not connected (linked) to the rest of the training set decrease global connectivity μ_P. Consequently, the strong correlation between connected variation μ_P and test performance suggests removal of such exemplars from the training set. The same can be concluded from figures 10,

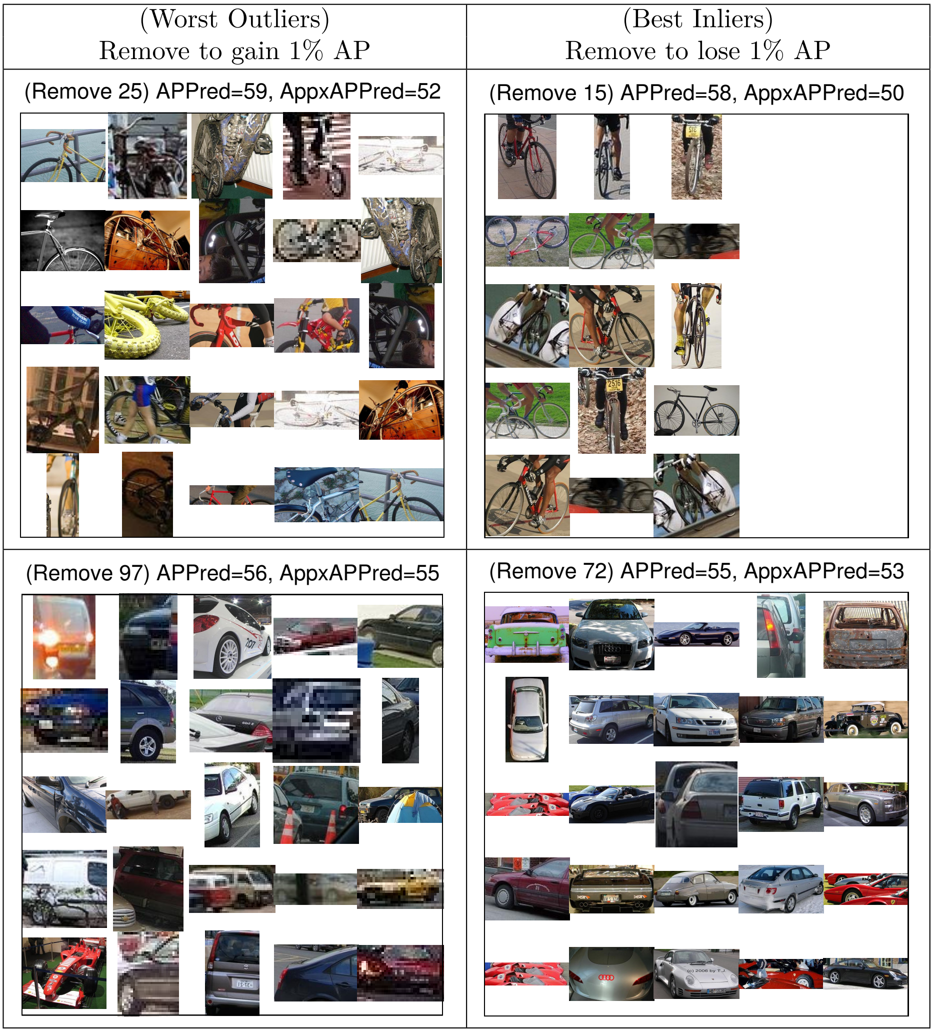

(Worst Outliers)
Remove to gain 1% AP
(Best Inliers)
Remove to lose 1% AP
(Remove 25) APPred=59, AppxAPPred=52
(Remove 15) APPred=58, AppxAPPred=50
(Remove 97) APPred=56, AppxAPPred=55
(Remove 72) APPred=55, AppxAPPred=53

Figure 8: Demonstration of *Automatic Dataset Selection.* For the two classes with best global connectivity μ_P, that are ‘bicycle’ and ‘car’, exemplars are shown which upon removal from the training set result in 1% change in the test performance.

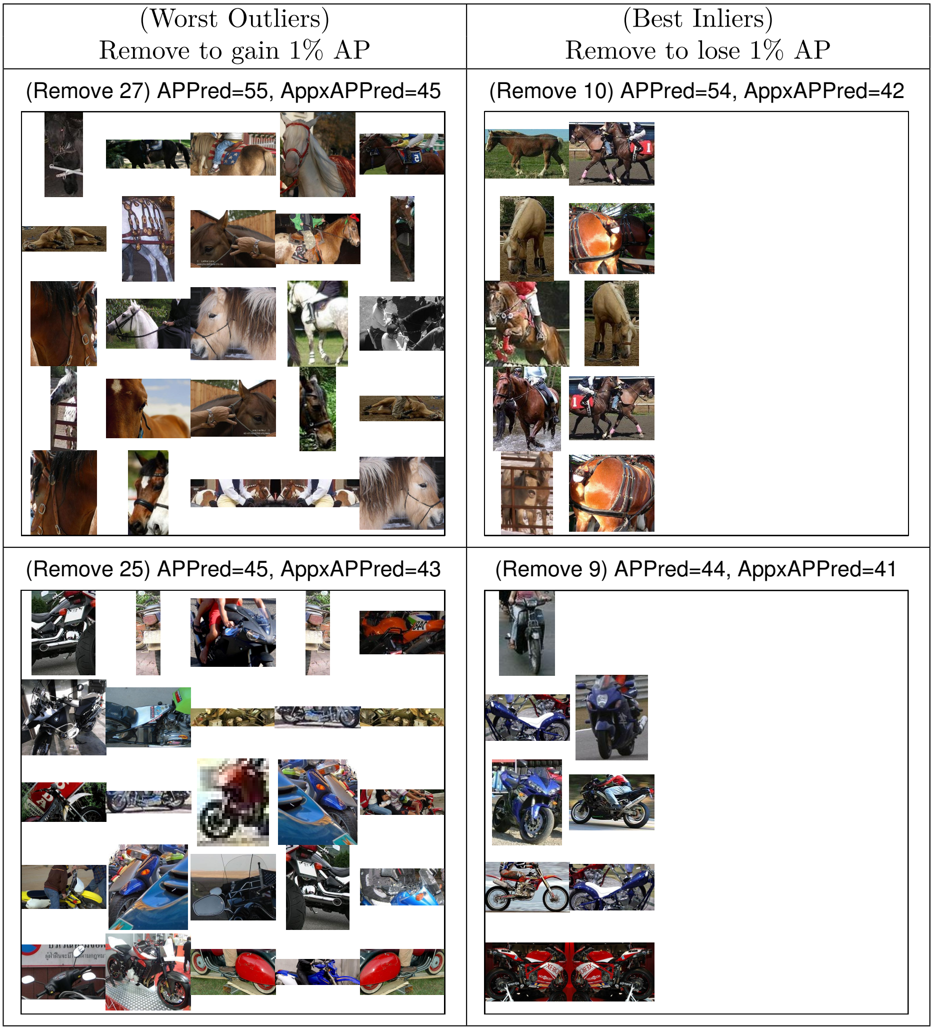

(Worst Outliers)
Remove to gain 1% AP
(Best Inliers)
Remove to lose 1% AP
(Remove 27) APPred=55, AppxAPPred=45
(Remove 10) APPred=54, AppxAPPred=42
(Remove 25) APPred=45, AppxAPPred=43
(Remove 9) APPred=44, AppxAPPred=41

Figure 9: Demonstration of *Automatic Dataset Selection* (continued). For the next classes with best global connectivity μ_P, that are ‘horse’ and ‘motorbike’, exemplars are shown which upon removal from the training set result in 1% change in the test performance.

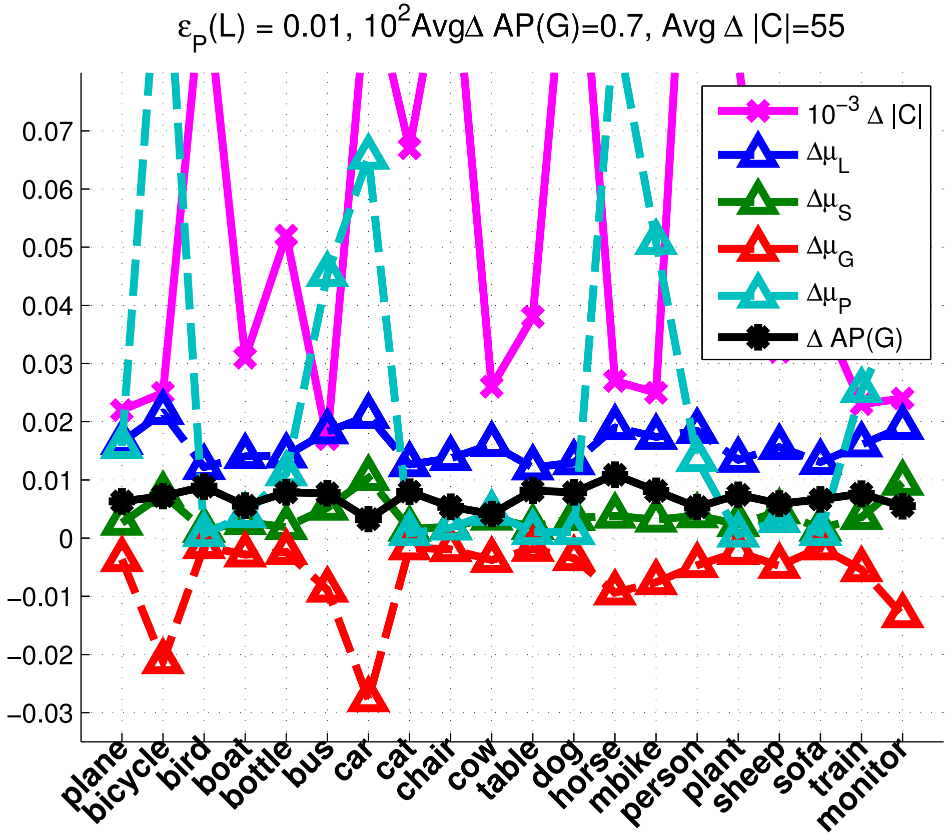


Figure 10: Changes to the training set induced by automatic dataset selection. See text for analysis. Best viewed electronically and in color.

8, and 9.

5 Discussion

5.1 Towards modelling the interplay between features, classifiers, training data, and test performance

As pointed out in section 4.3, the predicted test performances in some cases do not quite match the actual outcomes. This might be due to

1. factors that affect the test performance but are not related to the quality of the training set e.g. a significant difference between the distribution of the training set and that of the test set

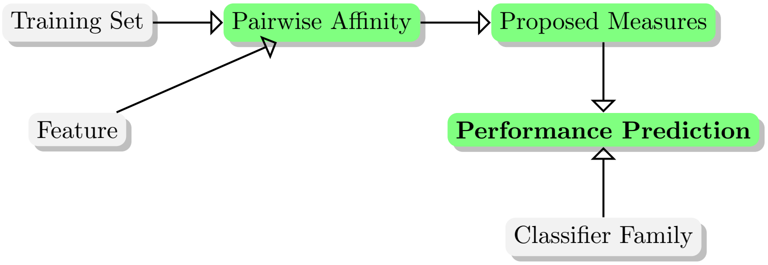


Figure 11: Our model of the dependencies between features, classifier families, training data, and test performance. The test set is assumed to have a distribution similar to that of the training set's.

2. variations in the similarity values which do not reflect similarity in the class level
3. a source of variability in the training set that the proposed measures do not model e.g. contextual information

While measuring the extent of correctness of each of these hypotheses is outside the scope of this study, investigating them is a promising and important direction for future works.

The assumption that the training set and test set have identical, or at least very similar, distributions is the core assumption of many learning algorithms. It will be interesting to verify to what extent for different classes this core assumption holds by measuring the properties of the test set and comparing it to those of the training set. This will automatically determine if a training set is a fair representation of a test set.

In the paper, we mostly focused on analyzing how the test performance varies with properties of the training data while keeping the feature, the similarity measure, the proposed measures and the classifier families fixed. However, the same methodology allows us to model the complete interplay between these factors.

Figure 11 demonstrates the dependencies between features, classifier families, training data and test performance. By modelling all dependencies at the same time, that is by modelling the predicted performance as a function of all these variables, one could attempt to "optimize" all the variables involved. By varying one or more factors and keeping the rest fixed, one could "optimize" the varying variables (boxes in the figure). Here "optimization" refers to a search process which results in more accurate predictions of test performances. For example, the same proposed procedures can be utilized to select, among a set of **similarity measures**, the one which results in more accurate test performance predictions, while all other factors – the feature, training set, test set, classifier families, and the proposed measures – are kept fixed. As another example, given all the factors but the **feature**, one could select, from a set of possible features, the one which maximizes the predicted test performances, without actually training any classifiers using that feature. Similarly, given a feature, similarity measure,... one should be able to propose the optimal **classifier** [3]. This would be a first systematic approach toward automatic selection of the optimal feature, classifier family, and the **training set**. Hence, this seems the most promising direction to explore further.

Investigating these directions will enable us to model the part of performance variations that currently our model cannot explain.

5.2 Scalability

The most expensive part of our analysis is the evaluation of the pairwise similarity measure. The computational cost of K^{E}_{MMI} [1] was reported to be cubic, given the trained and calibrated exemplar SVMs. The expensive training process of exemplar SVMs [12] could be avoided by using the recent 'who' features [9] based on LDA instead of an SVM formulation. The LDA analysis assumes a global Gaussian shape for the negative set, which is less flexible than local approach of the exemplar SVM. We expect this flexibility to result in less accurate selection of discriminative features, which consequently results in less accurate similarity measures. Nevertheless, this trade-off would be inevitable in large scale scenarios.

Global measures were shown to be the most informative ones. We used the Floyd-Warshall's algorithm [3] to find the shortest path between all pairs of samples, which has a cubic computational complexity. Using sparse graphs instead of the full ones we utilized, and using the Johnson's algorithm [3] instead of the Floyd-Warshall algorithm, can reduce the complexity to a super-quadratic one. However, as the moments are affected by the structure of the graph, and as optimizing the computational complexity has not been the focus of this study, we have not tried such an approach. Similarly, Laplacian embedding [14] can result in a sub-cubic, super-quadratic complexity but with an additional advantage; that is avoiding the heuristic definition of the distances based on similarity values (4). Unfortunately, such an approach is restricted to positive semi-definite similarity measures, which is not the case with the indefinite similarity measure we utilized.

As we demonstrated earlier in tables 4 and 5, local measures can approximate global measures with reasonable accuracy. Approximating the global measures with local and semi-global ones, and by using a similarity measure with quadratic complexity, cheaper but reasonable approximations to the proposed procedures can be achieved.

5.3 Dataset Selection

In this paper, we mostly analyzed test performance as a function of intra-class variation and connectivity of the training set. We assumed similar distributions for training and test sets and build upon this assumption. This assumption essentially

[3] Automatically proposing the optimal classifier in case of the HOG feature and Pascal VOC 2007 seems not particularly challenging at the moment(see table 2).

avoids addressing the bias problem. While in [18] cross dataset biases are analyzed, we argue that the selection biases affect the same dataset, although not as strongly as cross dataset biases. The set bias is inevitably a function of two sets, and a more elaborate model which considers two sets, e.g. (12), is required to address it. While this work avoided doing so, the model we propose can be extended to consider two sets.

A model based on two sets not only will be beneficial for modelling and measuring set biases, but it also will allow more significant modifications in the set selection scenario (section 3.2). However, training such models requires more data than a model based on one set requires. Here, proper data would involve different reference methods trained on different training sets, and each trained model tested on different sets. This is another promising direction to explore further in future.

The dataset selection procedure described in section 3 has been proposed mainly as proof of concepts. Qualitatively, results match our expectations and intuitions. Quantitatively, the accuracy of the predictions have to be verified. This involves a significant amount of training and testing steps, which will be very expensive if a wide range of family of models is considered. Therefore, it was avoided in this paper. It seems that the data required for training a two-set-based model would be sufficient to verify these predictions quantitatively.

6 Conclusions

This study proposes data-describing measures that link the quality of the training set to the test performance of classifiers. This essentially quantifies the claim on "Unreasonable effectiveness of data" [8] for s.o.a classifiers, and makes it possible to automatically measure the "cleanness of data" [20]. This implies that it should be possible to devise rules for automatic selection of training data that maximize the quality of the training set and consequently increase the test performance. Furthermore, the strong impact of the connectivity measure on the test performances suggests that "big connected data" might rectify the effects of intra-class variation.

References

[1] Omid Aghazadeh, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Mixture component identification and learning for visual recognition. In *European Conference on Computer Vision*, 2012.

[2] Lubomir Bourdev and Jitendra Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *IEEE International Conference on Computer Vision*, 2009.

[3] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms.* McGraw-Hill Higher Education, 2009.

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[5] Chaitanya Desai and Deva Ramanan. Detecting actions, poses, and objects with relational phraselets. In *European Conference on Computer Vision*, 2012.

[6] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[7] Pedro F. Felzenszwalb, Ross B. Girshick, David A. McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010.

[8] Alon Y. Halevy, Peter Norvig, and Fernando Pereira. The unreasonable effectiveness of data. *IEEE Intelligent Systems*, 2009.

[9] Bharath Hariharan, Jitendra Malik, and Deva Ramanan. Discriminative decorrelation for clustering and classification. In *European Conference on Computer Vision*, 2012.

[10] Aditya Khosla, Tinghui Zhou, Tomasz Malisiewicz, Alexei Efros, and Antonio Torralba. Undoing the damage of dataset bias. In *European Conference on Computer Vision*, 2012.

[11] Philip M. Long and Rocco A. Servedio. Random classification noise defeats all convex potential boosters. *Machine Learning*, pages 287–304, 2010.

[12] Tomasz Malisiewicz, Abhinav Gupta, and Alexei A. Efros. Ensemble of exemplar-svms for object detection and beyond. In *IEEE International Conference on Computer Vision*, 2011.

[13] J.L. Myers and A.D. Well. *Research Design and Statistical Analysis.* Lawrence Erlbaum Associates, 2003.

[14] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Neural Information Processing Systems*, 2001.

[15] O. M. Parkhi, A. Vedaldi, C. V. Jawahar, and A. Zisserman. Cats and dogs. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[16] M. Pedersoli, A. Vedaldi, and J. Gonzalez. A coarse-to-fine approach for fast deformable object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[17] Mohammad Amin Sadeghi and Ali Farhadi. Recognition using visual phrases. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[18] Antonio Torralba and Alexei A. Efros. Unbiased Look at Dataset Bias. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[19] Long Zhu, Yuanhao Chen, Alan L. Yuille, and William T. Freeman. Latent hierarchical structural learning for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[20] Xiangxin Zhu, Carl Vondrick, Deva Ramanan, and Charless C. Fowlkes. Do we need more training data or better models for object detection? In *British Machine Vision Conference*, 2012.

Paper E

Large Scale, Large Margin Classification using Indefinite Similarity Measures

Omid Aghazadeh and Stefan Carlsson

Submitted to:
Neural Information Processing Systems

Large Scale, Large Margin Classification using Indefinite Similarity Measures

Omid Aghazadeh and Stefan Carlsson

Abstract

Despite the success of the popular kernelized support vector machines, they have two major limitations: they are restricted to Positive Semi-Definite (PSD) kernels, and their training complexity scales at least quadratically with the size of the data. Many natural measures of similarity between pairs of samples are not PSD *e.g.* invariant kernels, and those that are implicitly or explicitly defined by latent variable models. In this paper, we investigate scalable approaches for using indefinite similarity measures in large margin frameworks. In particular we show that a normalization of similarity to a subset of the data points constitutes a representation suitable for linear classifiers. The result is a classifier which is competitive to kernelized SVM in terms of accuracy, despite having better training and test time complexities. Experimental results demonstrate that on CIFAR-10 dataset, the model equipped with similarity measures invariant to rigid and non-rigid deformations, can be made more than 5 times sparser while being more accurate than kernelized SVM using RBF kernels.

1 Introduction

Linear support vector machine (SVM) has become the classifier of choice for many large scale classification problems. The main reasons for the success of linear SVM are its max margin property achieved through a convex optimization, a training time linear in the size of the training data, and a testing time independent of it. Although the linear classifier operating on the input space is usually not very flexible, a linear classifier operating on a mapping of the data to a higher dimensional feature space can become arbitrarily complex.

Mixtures of linear classifiers has been proposed to increase the non-linearity of linear classifiers [10, 1]; which can be seen as feature mappings augmented with non-linear gating functions. The training of these mixture models usually scales bilinearly with respect to the data and the number of mixtures. The drawback is the non-convexity of the optimization procedures, and the need to know the (maximum) number of components beforehand.

Kernelized SVM maps the data to a possibly higher dimensional feature space, maintains the convexity, and can become arbitrarily flexible depending on the choice of the kernel function. The use of kernels, however, is limiting.

Firstly, kernelized SVM has significantly higher training and test time complexities when compared to linear SVM. As the number of support vectors grows approximately linearly with the training data [22], the training complexity becomes approximately somwehere between $\mathcal{O}(n^2)$ and $\mathcal{O}(n^3)$. Testing time complexity scales linearly with the number of support vectors, thus it is bounded by $\mathcal{O}(n)$.

Secondly, the positive (semi) definite (PSD) kernels are sometimes not expressive enough to model various sources of variation in the data. A recent study [21] argues that metric constraints are not necessarily optimal for recognition. For example, in image classification problems, considering kernels as similarity measures, they cannot align exemplars, or model deformations when measuring similarities. As a response to this, invariant kernels were introduced [6] which are generally indefinite. Indefinite similarity measures plugged in SVM solvers result in non-convex optimizations, unless explicitly made PSD, mainly using eigen decomposition methods [3]. Alternatively, latent variable models have been proposed to address the alignment problem *e.g.* [9, 25]. In these cases, the dependency of the latent variables on the parameters of the model being learnt mainly has two drawbacks: 1) the optimization problem in such cases becomes (at best) semi-convex, which is a form of non-convexity, and 2) the cost of training becomes much higher than the case without the latent variables.

This paper aims to address these problems using explicit basis expansion. We will show in section 2 that the resulting model: 1) has better training and test time complexities than kernelized SVM models, 2) can make use of indefinite similarity measures without any need for removal of the negative eigenvalues, which requires the expensive eigen decomposition, 3) can make use of multiple similarity measures without losing convexity, and with a cost linear in the number of similarity measures.

Our contributions are: 1) proposing and analyzing Basis Expanding SVM (BESVM) regarding the aforementioned three properties, and 2) investigating the suitability of particular forms of invariant similarity measures for large scale visual recognition problems.

2 Basis Expanding Support Vector Machine

We review linear and kernelized SVM in section 2.1, and related approaches for speeding up kernelized SVM in section 2.2. In section 2.3, we present the suggested model and its properties.We present our indefinite invariant similarity measures in section 2.4. We discuss the multi class formulation in section 2.5, and in section 2.6 we compare our approach to related work.

2.1 Background: SVM

Given a dataset $\mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n) | x_i \in \mathcal{X},\ y_i \in \{-1, 1\}\}$ the SVM based classifiers learn max margin binary classifiers. The SVM classifier is $f(x) = \langle w, x \rangle \geq 0$ [1]. The w is learnt via minimizing $\frac{1}{2}\langle w, w \rangle + C \sum_i \ell_H(y_i, f(x))$, where $\ell_H(y, x) = \max(0, 1 - xy)$ is the Hinge loss. Any positive semi definite (PSD) kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ can be associated with a reproducing kernel hilbert space (RKHS) $\mathcal{H}$, and vice versa, that is $\langle \psi_{\mathcal{H}}(x), \psi_{\mathcal{H}}(y) \rangle = k(x, y)$, where $\psi_{\mathcal{H}} : \mathcal{X} \to \mathcal{H}$ is the implicitly defined feature mapping associated to $\mathcal{H}$ and consequently to $k(.,.)$. Representer theorem states that in such a case, $\psi_{\mathcal{H}}(w) = \sum_i \gamma_i k(., x_i)$ where $\gamma_i \in \mathbb{R}\ \forall i$.

For a particular case of $k(.,.)$, namely the linear kernel $k(\mathbf{x}, \mathbf{y}) = \mathbf{x} \cdot \mathbf{y}$ associated with an Euclidean space, linear SVM classifier is

$$f_l(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} \geq 0 \tag{1}$$

where $\mathbf{w}$ is given by minimizing the primal SVM objective

$$\frac{1}{2}\|\mathbf{w}\|^2 + C \sum_i \ell_H(y_i, f_l(\mathbf{x}_i)) \tag{2}$$

More generally, given an arbitrary PSD kernel $k(.,.)$, the kernelized SVM classifier is

$$f_k(x) = \sum_i \alpha_i k(x, x_i) \geq 0 \tag{3}$$

where α_is are learnt by minimizing the dual SVM objective

$$\frac{1}{2}\alpha^{\mathrm{T}}\mathbf{YKY}\alpha - \|\alpha\|_1,\ 0 \leq \alpha_i \leq C,\ \alpha^{\mathrm{T}}\mathbf{y} = 0 \tag{4}$$

where $\mathbf{Y} = \mathrm{diag}\,(\mathbf{y})$.

The need for positiveness of $k(.,.)$ is evident from (4) where the quadratic regularizing term depends on the eigenvalues of $\mathbf{K}_{ij} = k(x_i, x_j)$. In case of indefinite $k(.,.)$s, the problem becomes non-convex and the inner products need to be redefined, as there will be no associating RKHS to indefinite similarity measures. Various workarounds for indefinite similarity measures exist, most of which involve expensive eigen decomposition of the gram matrix [3]. A PSD kernel can be learnt from the similarity matrix, with some constraints *e.g.* being close to the similarity matrix where closeness is usually measured by the Frobenius norm. In case of Frobenius norm, the closed form solution is spectrum clipping, namely setting the negative eigenvalues of the gram matrix to 0 [3]. As pointed out in [4], there is no guarantee that the resulting PSD kernels are optimal for classification. Nevertheless, jointly optimizing for a PSD kernel and the classifier [4] is impractical for large scale scenarios. We do not go into the details of possible re-formulations

[1] We omit the bias term for the sake of clarity.

regarding indefinite similarity measures, but refer the reader to [19, 13, 3] for more information.

Linear and Kernelized SVM have very different properties. Linear SVM has a training cost of $\mathcal{O}(d_{\mathbf{x}}n)$ and a testing cost of $\mathcal{O}(d_{\mathbf{x}})$ where $d_{\mathbf{x}}$ is the dimensionality of $\mathbf{x}$. Kernelized SVM has a training complexity which is $\mathcal{O}(d_k(nn_{sv})+n_{sv}^3)$ [15] where d_k is the cost of evaluating the kernel for one pair of data, and n_{sv} is the number of resulting support vectors. The testing cost of kernelized SVM is $\mathcal{O}(d_kn_{sv})$. Therefore, a significant body of research has been dedicated to reducing the training and test costs of kernelized SVMs by approximating the original problem.

2.2 Speeding up Kernelized SVM

A common approach for approximating the kernelized SVM problem is to restrict the feature mapping of w: $\psi_{\mathcal{H}}(w) \approx \psi_R(w) = \sum_{j=1}^{J} \beta_j \psi_{\mathcal{H}}(z_j)$ where $J < n$. Methods in this direction either learn synthetic samples z_j [24] or restrict z_j to be on the training data [15]. These methods essentially exploit low rank approximations of the gram matrix $\mathbf{K}$.

Low rank approximations of PD $\mathbf{K} \succ 0$, result in speedups in training and testing complexities of kernelized SVM. Methods that learn basis coordinates outside the training data *e.g.* [24] usually involve intermediate optimization overheads, and thus are prohibitive in large scale scenarios. On the contrary, the Nyström method gives a low rank PSD approximation to $\mathbf{K}$ with a very low cost.

The Nyström method [23] approximates $\mathbf{K}$ using a randomly selected subset of the data:

$$\mathbf{K} \approx \mathbf{K_{nm}}\mathbf{K_{mm}^{-1}}\mathbf{K_{mn}} \tag{5}$$

where $\mathbf{K_{ab}}$ refers to a sub matrix of $\mathbf{K} = \mathbf{K_{nn}}$ indexed by $\mathbf{a} = (a_1, \dots, a_n)^{\mathrm{T}}$, $a_i \in \{0, 1\}$, and similarly by $\mathbf{b}$. The approximation (5) is derived by defining eigenfunctions of $K(.,.)$ as expansions of numerical eigenvectors of $\mathbf{K}$. A consequence is that the data can be embedded in an Euclidean space: $\mathbf{K} \approx \mathbf{\Psi_{mn}^{\mathrm{T}}}\mathbf{\Psi_{mn}}$, where $\mathbf{\Psi_{mn}}$, the Nyström feature space, is

$$\mathbf{\Psi_{mn}} = \mathbf{K_{mm}^{-\frac{1}{2}}}\mathbf{K_{mn}} \tag{6}$$

Methods exist which either explicitly or implicitly exploit this *e.g.* [14] to reduce both the training and test costs, by restricting the support vectors to be a subset of the bases defined by $\mathbf{m}$.

In case of indefinite similarity measures, $\mathbf{K_{mm}^{-\frac{1}{2}}}$ in (6) will not be real. In the rest of the paper, we refer to an indefinite version of a similarity matrix $\mathbf{K}$ with $\tilde{\mathbf{K}}$. In order to get a PSD approximation of an indefinite $\tilde{\mathbf{K}}$, the indefinite $\tilde{\mathbf{K}}_{\mathbf{mm}}$ (5) needs to be made PSD. Spectrum clipping, spectrum flip, spectrum shift, and spectrum square are possible solutions based on eigen decomposition of $\tilde{\mathbf{K}}_{\mathbf{mm}}$. The latter can be achieved without the eigen decomposition step: $\tilde{\mathbf{K}}_{\mathbf{mm}}^{\mathrm{T}}\tilde{\mathbf{K}}_{\mathbf{mm}} \succeq 0$.

If the goal is to find the PSD matrix closest to the original indefinite $\tilde{\mathbf{K}}$ with respect to the reduced basis set $\mathbf{m}$, spectrum clip gives the closed form solution. Therefore, when there are a few negative eigenvalues, the spectrum clip technique gives good low rank approximations to $\tilde{\mathbf{K}}_{\mathbf{mm}}$ which can be used by (5) to get a low rank PSD approximation to $\tilde{\mathbf{K}}$. However, when there are a considerable number of negative eigenvalues, as it is the case with most of the similarity measures we consider later on in section 2.4, there is no guarantee for the resulting PSD matrix to be optimal for classification. This is true specially when eigenvectors associated with negative eigenvalues contain discriminative information. We will experimentally verify in section 3.3 that the negative eigenvalues do contain discriminative information.

As an alternative, instead of aiming to approximate $\tilde{\mathbf{K}}$ with a PSD matrix, we aim to find a low rank PSD matrix which results in a linear discriminant that is competitive with the one learnt in the Nyström feature space based on a spectrum clip technique for making $\tilde{\mathbf{K}}_{\mathbf{mm}}$ PSD. In other words, we want to avoid modifying the eigenvalues of $\tilde{\mathbf{K}}_{\mathbf{mm}}$; which means that we want to avoid the normalization by $\tilde{\mathbf{K}}_{\mathbf{mm}}^{-1}$[2]. For example, one can replace $\mathbf{K}_{\mathbf{mm}}$ in (6) with the covariance of columns of $\mathbf{K}_{\mathbf{mn}}$. We experimentally found out that a simple embedding: scaling $\bar{\mathbf{K}}_{\mathbf{mm}}$ by the average ℓ_2 norm of its columns where $\bar{\mathbf{K}}_{\mathbf{mm}}$ is $\mathbf{K}_{\mathbf{mm}}$ with its rows centered, is competitive with the Nyström embedding (6) for PSD similarity measures, while outperforming it in case of indefinite ones that we studied. The embedding is presented in the next section; see (8).

2.3 Basis Expanding SVM

Basis Expanding SVM (BE-SVM) is a linear SVM classifier equipped with a normalization of the following explicit feature map

$$\tilde{\varphi}(\mathbf{x}) = [s(\mathbf{b}_1, \mathbf{x}), \dots, s(\mathbf{b}_B, \mathbf{x})]^{\mathrm{T}} \tag{7}$$

where $\mathcal{B} = \{\mathbf{b}_1, \dots, \mathbf{b}_B\}$ is an ordered basis set[3] which is a subset of the training data, and $s(.,.)$ is a pairwise similarity measure. The BE-SVM feature space defined by

$$\varphi(\mathbf{x}) = \frac{1}{\mathbb{E}_{\mathcal{X}}[\|\tilde{\varphi} - \mathbb{E}_{\mathcal{X}}[\tilde{\varphi}]\|]} \left(\tilde{\varphi}(\mathbf{x}) - \mathbb{E}_{\mathcal{X}}[\tilde{\varphi}]\right) \tag{8}$$

is similar to the Nyström feature space (6) with a different normalization scheme, as pointed out in section 2.2. The centralization of $\tilde{\varphi}(.)$ better conditions $\varphi(.)$ for a linear SVM solver, and normalization by the average ℓ_2 norm is most useful for combining multiple similarity measures.

The BE-SVM classifier is

$$f_{\mathcal{B}}(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\varphi(\mathbf{x}) \geq 0 \tag{9}$$

[2] Throughout the text, we will refer to this normalization with $\tilde{\mathbf{K}}_{\mathbf{mm}}^{-1}$ with the Nyström normalization.

[3] For the moment assume $\mathcal{B}$ is given. We experiment with different basis selection strategies later in section 3.4).

where $\mathbf{w}$ is solved by minimizing the primal BE-SVM objective

$$\frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_i \ell_H(y_i, f_{\mathcal{B}}(\mathbf{x}_i))^2 \tag{10}$$

An ℓ_1 regularizer results in sparser solutions, but with the cost of more expensive optimization than an ℓ_2 regularization. Therefore, for large scale scenarios, an ℓ_2 regularization, combined with a reduced basis set $\mathcal{B}$ is preferred to an ℓ_1 regularizer combined with a larger basis set.

Both kernelized SVM and BE-SVM are max margin classifiers in their feature spaces. The feature space of kernelized SVM $\psi_{\mathcal{H}}(.)$ is implicitly defined via the kernel function $k(.,.)$ while the feature space of the BE-SVM is explicitly defined. In order to derive the margin as a function of the data, we first need to derive the dual BE-SVM objective, where we assume a non-squared Hinge loss and unnormalized feature mappings $\tilde{\varphi}(.)$. Borrowing from the representer theorem and considering the KKT conditions of the primal, one can derive $\mathbf{w} = \sum_i y_i\beta_i\tilde{\varphi}(\mathbf{x}_i)$, and consequently derive the BE-SVM dual objective which is similar to the dual SVM objective (4) but with $\mathbf{K}_{ij} = \tilde{\varphi}(\mathbf{x}_i)^{\mathrm{T}}\tilde{\varphi}(\mathbf{x}_j)$. Let $\mathbf{S}_{BX}$ refer to the similarity of the data to the bases. We can see that the margin of the BE-SVM, given the optimal dual variables $0 \leq \beta_i \leq C$, is $\left(\beta^{\mathrm{T}}\mathbf{Y}\mathbf{S}_{BX}^{\mathrm{T}}\mathbf{S}_{BX}\mathbf{Y}\beta\right)^{-1}$, as opposed to $\left(\alpha^{\mathrm{T}}\mathbf{YKY}\alpha\right)^{-1}$ for the kernelized SVM, given the optimal dual variables $0 \leq \alpha_i \leq C$. Furthermore, $\mathbf{S}_{BX}^{\mathrm{T}}\mathbf{S}_{BX}$ is PSD, and that is BE-SVM's workaround for using indefinite similarity measures. We provide more analysis regarding the margin of BE-SVM in supplementary materials.

Using multiple similarity measures is straightforward in BE-SVM. The concatenated feature map $\varphi_M(\mathbf{x}) = \left[\varphi^{(1)}(\mathbf{x})^{\mathrm{T}}, \ldots, \varphi^{(M)}(\mathbf{x})^{\mathrm{T}}\right]^{\mathrm{T}}$ encodes the values of the M similarity measures evaluated on the corresponding bases $\mathcal{B}^{(1)}, \ldots, \mathcal{B}^{(M)}$. In this work, we restrict the study to the case that the bases are shared among the M similarity measures: *i.e.* $\mathcal{B}^{(1)} = \ldots = \mathcal{B}^{(M)}$. In such cases, it can be verified that in case of unnormalized features $\tilde{\varphi}^{(m)}(.)$, the corresponding Gram matrix will be

$$\begin{aligned}\tilde{K}(\mathbf{x}_i, \mathbf{x}_j) &= \textstyle\sum_{m=1}^{M} \tilde{K}^{(m)}(\mathbf{x}_i, \mathbf{x}_j)\\ &= \textstyle\sum_{m=1}^{M} \tilde{\varphi}^{(m)}(\mathbf{x}_i)^{\mathrm{T}}\tilde{\varphi}^{(m)}(\mathbf{x}_j)\\ &= \textstyle\sum_{m=1}^{M}\sum_{b=1}^{B} s^{(m)}(\mathbf{b}_b, \mathbf{x}_i)s^{(m)}(\mathbf{b}_b, \mathbf{x}_j)\end{aligned} \tag{11}$$

where $\tilde{K}^{(m)}$s are combined with equal weights, the value of each of which depends (locally) on how the similarities of $\mathbf{x}_i$ and $\mathbf{x}_j$ correlate with respect to the bases. In the case of normalized features, the centered values of each similarity measure is weighted by $(\mathbb{E}_{\mathcal{X}}[\|\tilde{\varphi} - \mathbb{E}_{\mathcal{X}}[\tilde{\varphi}]\|])^{-2}$ *i.e.* more global weight is put on (the centered values of) the similarity measures with smaller variances in similarity values.

While the BE-SVM's normalization of empirical kernel maps is not optimal for discrimination, it can be seen as a reasonable prior for combining different similarity measures. Utilizing such a prior, in combination with linear classifiers and ℓ_P regularizers, has two important consequences: 1) the centering helps reduce the correlation between dimensions and the scaling helps balance the effect of regularization on different similarity measures, irrespective of their overall norms, and 2) such a scaling directly affects the parameter tuning for learning the linear classifiers: for all the similarity measures (and combinations of similarity measures) with various basis sizes, the same parameter: $C = 1$ was used to train the classifiers. While cross-validation will still be a better option, cross-validating for different parameters settings – and specially when combining multiple similarity measures – will be very expensive and prohibitive. By using the BE-SVM's normalization, we essentially avoid searching for optimal combining weights for different similarity measures and also tuning for the C parameter of the linear SVM training. The normalization of BE-SVM is evaluated quantitatively in the supplementary materials.

2.4 Indefinite Similarity Measures for Visual Recognition

The lack of expressibility of the PSD kernels have been argued before *e.g.* in [3, 4, 21]. For example, similarity measures which are not based on vectorial representations of data are most likely to be indefinite. Particularly in computer vision, considering latent information results in lack of a fixed vectorial representation of instances, and therefore similarity measures based on latent information are most likely to be indefinite[4].

A few applications of indefinite similarity measures in computer vision are pointed out below. [6] proposed (indefinite) jitter kernels for building desired invariances in classification problems. [1] used indefinite pairwise similarity measures with latent positions of objects for clustering. [16] considers deformation models for image matching. [7] defines an indefinite similarity measure based on explicit correspondences between pairs of images for image classification.

In this work, we consider similarity measures with latent deformations:

$$s(x_i, x_j) = \max_{z_i \in \mathcal{Z}(x_i),\, z_j \in \mathcal{Z}(x_j)} K_I(\phi(x_i, z_i), \phi(x_j, z_j)) + \mathcal{R}(z_i) + \mathcal{R}(z_j) \tag{12}$$

where $K_I(.,.)$ is a similarity measure (potentially a PD kernel), $\phi(x, z)$ is a representation of x given the latent variable z, $\mathcal{R}(z)$ is a regularization term on the latent variable z, and $\mathcal{Z}(x)$ is the set of possible latent variables associated with x. Specifically, when $\mathcal{R}(.) = 0$ and $\mathcal{Z}(x)$ involves latent positions, the similarity measure becomes similar to that of [1]. When $\mathcal{R}(.) = 0$ and $\mathcal{Z}(x)$ involves latent positions and local deformations, it becomes similar to the zero order model of [16]. Finally, an MRF prior in combination with latent positions and local deformations gives a similarity measure, similar to that of [7].

The proposed similarity measure (12) picks the latent variables which have the maximal (regularized) similarity values $K_I(.,.)$s. This is in contrast to [6] where the

[4] Note that [25] and similar approaches use a PD kernel on fixed vectorial representation of the data, *given the latent information*. The latent informations in turn are updated using an alterantive minimization approach. This makes the optimization non-convex, and differs from similarity measures which directly model latent informations.

Training		
	Memory	Computation
K SVM	$nM\bar{d}_\phi + \frac{n^2}{C}$	$n^2M\bar{d}_K + \frac{n^3}{C}$
BE-SVM	$nM\bar{d}_\phi + nM\|\mathcal{B}\|$	$nC\|\mathcal{B}\|M\bar{d}_K$

Testing (per sample)		
	Memory	Computation
K SVM	$nM\bar{d}_\phi$	$nCM\bar{d}_K$
BE-SVM	$\|\mathcal{B}\|M\bar{d}_\phi$	$\|\mathcal{B}\|CM\bar{d}_K$

Table 1: Complexity Analysis for kernelized SVM and BE-SVM. The number of samples for each of the C classes was assumed to be equal to $\frac{n}{C}$. M is the number of kernels/similarity measures, $M\bar{d}_\phi$ is the dimensionality of representations required for evaluating M kernels/similarity measures, and $M\bar{d}_K$ is the cost of evaluating all M kernels/similarity measures.

latent variables were suggested to be those which minimize a metric distance based on the kernel $K_I(.,.)$. The advantage of a metric based latent variable selection is not so clear, while some works argue against unnecessary restrictions to metrics [21]. Also, if $K_I(.,.)$ is not PSD, deriving a metric from it is at best expensive. Therefore, the latent variables in (12) are selected according to the similarity values instead of metric distances.

2.5 Multi Class Classification

SVMs are mostly known as binary classifiers. Two popular extensions to the multi-class problems are one-v-res (1vR) and one-v-one (1v1). The two simple extentions have been argued to perform as well as more sophisticated formulations [20]. In particular, [20] concludes that in case of kernelized SVMs, in terms of accuracy they are both competitive, while in terms of training and testing complexities 1v1 is superior. Therefore, we only consider 1v1 approach for kernelized SVM. In case of linear SVMs however, 1v1 results in unnecessary overhead and 1vR is the algorithm of choice. A 1vR BE-SVM can be expected to be both faster and to generalize better than a 1v1 BE-SVM where bases from all classes are used in each of the binary classifiers. In case of 1v1 BE-SVM where only bases from the two classes under consideration are used in each binary classifier, there will be a clear advantage in terms of training complexity. However, due to the reduction in the size of the basis set, the algorithm generalizes less in comparison to a 1vR approach. Therefore, we only consider 1vR formulation for BE-SVM. Table 1 summarizes the memory and computational complexity analysis for 1v1 kernelized SVM and 1vR BE-SVM. Shown are the upper bounds complexities where we have considered n to be the upper bound on n_{sv}.

2.6 Related Work

There exists a body of work regarding the use of proximity data, similarity, or dissimilarity measures in classification problems. [18] uses similarity to a fixed set of samples as features for a kernel SVM classifier. [12] uses proximities to all the data as features for a linear SVM classifier. [11] uses proximities to all the data as features and proposes a linear program machine based on this representation. In contrast, we use a normalization of the similarity of points to a subset of the data as features for a (fast, approximate) linear SVM classifier.

3 Experiments

3.1 Dataset and Experimental Setup

We present our experimental results on CIFAR-10 dataset [17]. The dataset is comprised of 60,000 tiny 32×32 RGB images, 6,000 images for each of the 10 classes involved, divided into 6 folds with inequal distribution of class labels per fold. The first 5 folds are used for training and the 6th fold is used for testing. We use a modified version of the HOG feature [5], described in [9]. For most of our experiments, we use HOG cell sizes of 8 and 4, which result in $31 \times \frac{32}{8}^2 = 496$ and $31 \times \frac{32}{4}^2 = 1984$ dimensional representation of each of the images.

Due to the normalization of each of the HOG cells, namely normalizing by gradient/contrast information of the neighboring cells, the HOG cells on border of images are not normalized properly. We believe this to have a negative effect on the results, but as the aim of this paper is not to get the best results possible out of the model, we rely on the consistency of the normalization for all images to address this problem. A possible fix is to *e.g.* up-sample images and ignore the HOG-cells at the boundaries, but we do not provide the results for such fixes.

For all of the results in the paper, we center the HOG feature vector and scale feature vectors inversely by the average ℓ_2 norm of the centered feature vectors, similar to the normalization of BE-SVM (8). This results in easier selection of parameters C and γ for SVM formulations. Unless stated otherwise, we fix $C = 2$ and $\gamma = 1$ for kernelized SVM with Gaussian RBF kernels, and $C = 1$ for the rest. We use LibLinear [8] to optimize the primal linear SVM objectives with squared Hinge loss, similar to (10). For kernelized SVM, we use LibSVM [2]. We report multi-class classification results (0-1 loss) on the test set.

3.2 Baseline: SVM with Positive Definite Kernels

Figure 3 shows the performance of linear SVM (H4L and H8L) and kernelized SVM with Gaussian RBF kernel (K4R and K8R) as a function of number of parameters in the models. The number of parameters for linear SVM is the input dimensionality, and for kernelized SVM it is the sum of $n_{sv}(d_\phi + 1)$ where d_ϕ is the dimensionality of the feature vector the corresponding kernel operates on. The 5 numbers for each model are the results of the model trained on $1, \ldots, 5$ folds of the training data (each fold contains 10,000 samples). Figure 4 shows the performance kernelized SVM as a function of support vectors when trained on $1, \ldots, 5$ folds. Except the linear SVM with a HOG cell size of 8 pixels (496 dimensions) which saturates its performance at 4 folds, all models consistently benefit from more training data.

3.3 BE-SVM with Invariant Similarity Measures

The general form of the invariant similarity measures we consider was given in (12). In particular, we consider rigid and deformable similarity measures where the smallest unit of deformation/translation is a HOG cell.

The rigid similarity measure models invariance to translations and is given by

$$K_R(x, y) = \max_{\mathbf{z}_R \in \mathcal{Z}_R} \sum_{\mathbf{c} \in \mathcal{C}} \phi_C(x, \mathbf{c})^T \phi_C(y, \mathbf{c} + \mathbf{z}_R) \tag{13}$$

where $\mathcal{Z}_R = \{(z_x, z_y) | z_x, z_y \in \{-h_R, \ldots, h_R\}\}$ allows a maximum of h_R HOG cells displacements in x, y directions, $\mathcal{C} = \{(x, y) | x, y \in \{h_1, \ldots, h_H\}$ is the set of indices of h_H HOG cells in each direction, and $\phi_C(x, \mathbf{c})$ is the 31 dimensional HOG cell of x located at position $\mathbf{c}$. $\phi_C(x, \mathbf{c})$ is zero for cells outside x (zero-padding). $K_R(x, y)$ is the maximal cross correlation between $\phi(x)$ and $\phi(y)$.

The deformable similarity measure allows local deformations (displacements) of each of the HOG cells, in addition to invariance to rigid deformations

$$K_L(x, y) = \max_{z_R \in \mathcal{Z}_R} \sum_{\mathbf{c} \in \mathcal{C}} \max_{z_L \in \mathcal{Z}_L} \phi_C(x, \mathbf{c})^T \phi_C(y, \mathbf{c} + \mathbf{z}_R + \mathbf{z}_L) \tag{14}$$

where $\mathcal{Z}_L = \{(z_x, z_y) | z_x, z_y \in \{-h_L, \ldots, h_L\}\}$ allows a maximum of h_L HOG cell local deformation for each of the HOG cells of y.

We consider a maximum deformation of 8 pixels *e.g.* 2 HOG cells for a HOG cell size of 4 pixels. Regularizing global or local deformations is straightforward in this formulation. However, we did not notice significant improvements for the set of displacements we considered, which is probably related to the small size of the latent set suitable for small images in CIFAR-10.

Figure 1 shows the performance of BE-SVM using different similarity measures, when trained on the first fold. It can be seen that the invariant similarity measures improve recognition performance. Particularly, in absence of any other information, modelling rigid deformations (latent positions) seems to be much more beneficial than modelling local deformations. An interesting observation is that aligning the data in higher resolutions is much more crucial: all models (linear SVM, kernelized SVM, and BE-SVM) suffer performace losses when the resolution is increased from a HOG cell size of 8 pixels to 4 pixels. However, BE-SVM achieves significant performance gains by aligning the data in higher resolutions: compare H4L with H4(1,0) and H4(2,0), and H8L with H8(1,0).

We tried training linear and kernelized SVM models by jittering the feature vectors, in the same manner that the invariant similarity measures do (13), (14); that is

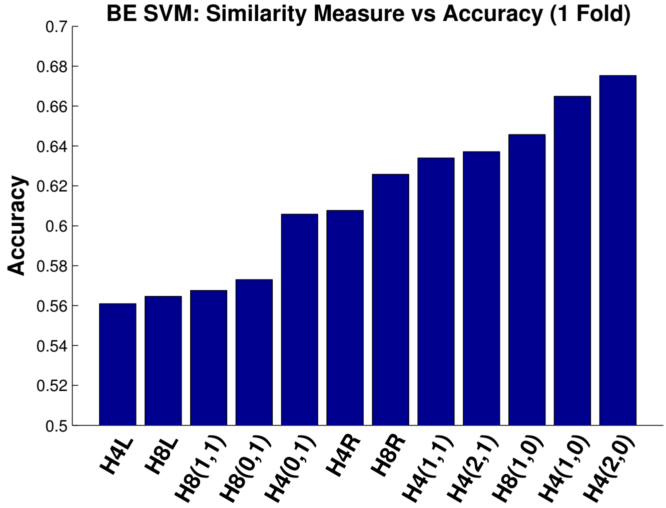


Figure 1: Performance of BE-SVM as a function of different similarity measures when trained on the first fold. An H4 (H8) refers to a HOG cell size of 4 (8) pixels. L and R refer to linear and Gaussian RBF kernels respectively, and (h_R, h_L) refers to a similarity measure with h_R rigid and h_L local deformations (13), (14).

to jitter the HOG cells with zero-padding for cells outside images. This resulted in significant performance losses for both linear SVM and kernelized SVM, while also siginificantly increasing memory requirement and computation times. We believe the reason for this to be the boundary effects; which are also mentioned in previous work *e.g.* [6]. We also believe that jittering the input images, in combination with some boundary heuristics (see section 3.1), will improve the test performance (while significantly increasing training complexities), but we do not provide experimental results for such cases.

3.4 Basis Selection

Figure 2 shows accuracy of BE-SVM using different similarity measures and different basis selection strategies; for a basis size of $B = 10 \times 100$ exemplars. In the figure, 'Rand' refers to a random selection of the bases, 'Indx' refers to selection of samples according to their indices, 'K KMed' refers to a kernel k-medoids approach based on the similarity measure, and 'Nystrom' refers to selection of bases similar to the 'Indx' approach, but with the Nyström normalization, using a spectrum clip fix for indefinite similarity measures(see section 2.2). The reported results for 'Rand' method is averaged over 5 trials; the variance was not significant. It can be observed that all methods except the 'Nystrom' result in similar performances.

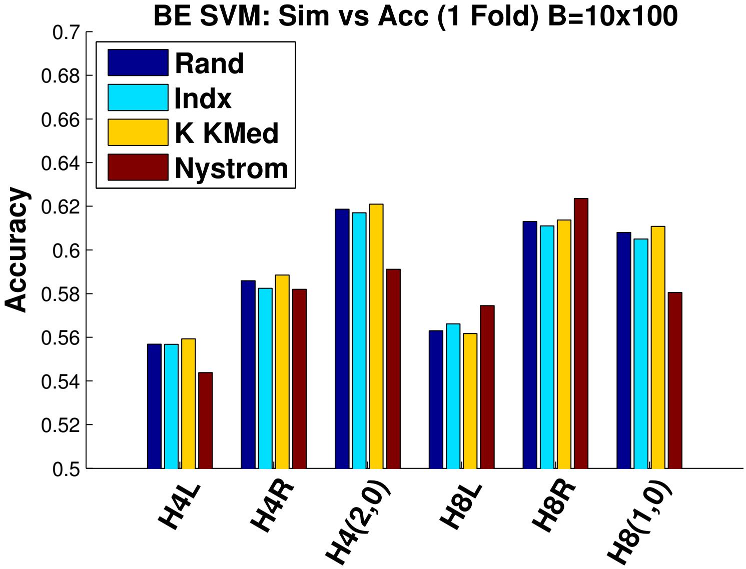


Figure 2: Performance of various basis selection strategies for BE-SVM using various similarity measures. See text for analysis.

We also tried other sophisticated sample selection criteria, but observed similar behaviour. We attribute this to little variation in the quality of exemplars in the CIFAR-10 dataset. Having observed this, for the rest of sub-sampling strategies, we do not average over multiple random basis selection trials, but rather use the deterministic 'Indx' approach.

The difference between normalization factors in BE-SVM and Nyström method (see section 2.2) is evident from the figure. The BE-SVM normalization tends to be superior consistently in case of indefinite similarity measures. For PSD kernels (H4L, H8L, H4R, and H8R) , the Nyström normalization tends to be better in lower resolutions (H8) and worse in higher resolutions (H4). We believe the main reason for this is to be lack of significant similarity of bases in higher resolutions in absence of any alignment. In such cases, the low rank assumption of $\mathbf{K}$ [23] is violated, and normalization by a diagonally dominant $\mathbf{K_{mm}}$ will not capture any useful information.

In order to analyze how the performance of BE-SVM depends on the eigenvalues of the similarity measures, we provide the following eigenvalue analysis. We compute the similarity matrix of the bases to themselves – corresponding to $\mathbf{K_{mm}}$ in (6)) – and perform an eigendecomposition of the resulting matrix. Table 2 shows the ratio of negative eigenvalues: 'NgRat'$=\frac{1}{B}\sum_i[\lambda_i < 0]$, and the relative energy

	H4(0,0)	H4(0,1)	H4(1,0)	H4(1,1)	H4(2,0)	H4(2,1)	CorNyst	CorBE
NgRat	.0	.26	.18	.25	.16	.30	.20	.61
NgEng	.00	.04	.05	.05	.04	.07	.33	.73

Table 2: Eigenvalue analysis of various similarity measures based on HOG cell size 4. See text for analysis.

of eigenvalues 'NgEng'=$\frac{\sum_i |\lambda_i|[\lambda_i<0]}{\sum_i |\lambda_i|[\lambda_i>0]}$ as a function of various similarity measures for $B = 10 \times 100$ and a HOG cell size of 4. The last two columns, namely 'CorNyst' and 'CorBE' reflect the correlation of the measured entities – 'NgRat' and 'NgEng' – to the observed performance of BE-SVM using the Nyström normalization and BE-SVM normalization. We used Pearson's r to measure the extent of linear dependence between the test performances and different normalization schemes. It can be observed that: 1) both normalization schemes have a positive correlation with both the ratio of negative eigenvalues and their relative energy, and 2) BE-SVM normalization correlates more strongly with the observed entities. From this, we conclude that negative eigenvectors contain discriminative information and that BE-SVM's normalization is more suitable for indefinite similarity measures. We also experimented with spectrum flip and spectrum square methods for the Nyström normalization, but they generally provided slightly worse results in comparison to the spectrum clip technique.

3.5 Multiple Similarity Measures

Different similarity measures contain complementary information. Fortunately, BE-SVM can make use of multiple similarity measures by construction. To demonstrate this, using one fold of training data and $B = 10 \times 50$, we greediy and in an incremental way augmented the similarity measures with the most contributing ones. Using this approach, we found two (ordered) sets of similarity measures with complementary information: 1) a low-resolution set $\mathcal{M}_1 = \{H8R, H8(1,0), H8(0,1)\}$, and 2) a two-resolution set $\mathcal{M}_2 = \{H8R, H4(2,0), H4(0,1), H8(1,0)\}$. Surprisingly, the two resolution sequence resembles those of the part based models [9], and multi resolution rigid models [1] in that the information is processed at two levels: a coarser rigid 'root' level and a finer scale deformable level.

We then trained BE-SVM models using these similarity measures for various sizes of the basis set, and for various sizes of training data. Figures 3 and 4 show these results, where the BE-SVM models are trained on all 5 folds. The shown number of supporting exemplars (and consequently the number of parameters) for BE-SVM are based on the size of the basis set. It can be seen that using a basis size of $B = 10 \times 250$, the performance of the BE-SVM using more than 3 two-

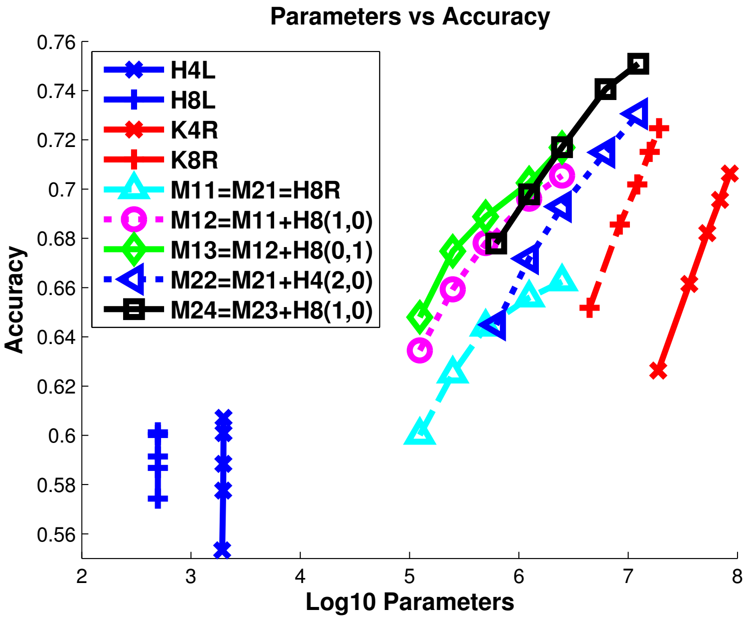


Figure 3: Performance of BE-SVM vs model parameters for various sizes of the basis set, using multiple similarity measures. Each curve for linear SVM (H4L, H8L) and kernelized SVM (K4R, K8R) represents the result for training on $1, \ldots, 5$ folds of training data. Each curve for BE-SVM shows the result for training model with a basis set of size $B = 10 \times \{25, 50, 100, 250, 500\}$ when trained on 5 folds of the training data.

resolution similarity measures surpass that of the kernelized SVM trained on all the data. Using low-resolution similarity measures, $B = 10 \times 500$ outperforms kernelized SVMs trained on up to 4 folds of the training data. Furthermore, it can be observed that for the same model complexity, as measured either by the number of supporting exemplars, or by model parametrs, BE-SVM performs better than kernelized SVM.

Measured by model parameters, BE-SVM is roughly 8 times sparser than kernelized SVM for the same accuracy. Measured by supporting exemplars, its sparsity increases roughly to 30. We need to point out that different similarity measures have different complexities *e.g.* H8(1,0) is more expensive to evaluate than K8R. However, when the bases are shared for different similarity measures, CPU cache can be utilized much more efficiently as there will be less memory access and more (cached) computations.

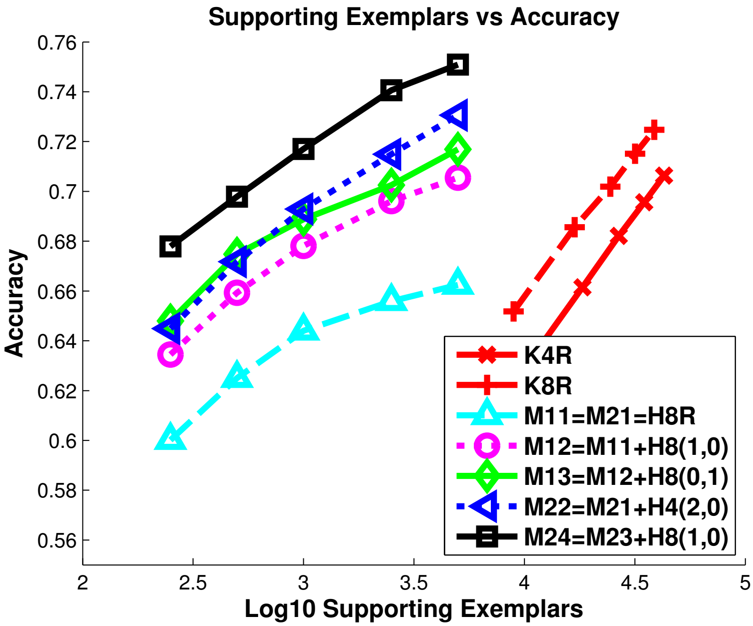


Figure 4: Performance of BE-SVM vs supporting exemplars/support vectors for various sizes of the basis set, using multiple similarity measures. See text for analysis.

4 Conclusion

We analyzed scalable approaches for using indefinite similarity measures in large margin scenarios. We showed that our model based on an explicit basis expansion of the data according to arbitrary similarity measures can result in competitive recognition performances, while scaling better with respect to the size of the data. The model named Basis Expanding SVM was thoroughly analyzed and extensively tested on CIFAR-10 dataset.

In this study, we did not explore basis selection strategies, mainly due to the small intra-class variation of the dataset. We expect basis selection strategies to play a crucial role in the performance of the resulting model on more challenging datasets *e.g.* Pascal VOC or ImageNet. Therefore, an immediate future work is to apply BE-SVM to larger scale and more challenging problems *e.g.* object detection, in combination with data driven basis selection strategies.

References

[1] Omid Aghazadeh, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Mixture component identification and learning for visual recognition. In *European Conference on Computer Vision*, 2012.

[2] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, pages 27:1–27:27, 2011.

[3] Yihua Chen, Eric K. Garcia, Maya R. Gupta, Ali Rahimi, and Luca Cazzanti. Similarity-based classification: Concepts and algorithms. *Journal of Machine Learning Research*, pages 747–776, 2009.

[4] Yihua Chen, Maya R. Gupta, and Benjamin Recht. Learning kernels from indefinite similarities. In *International Conference on Machine Learning*, 2009.

[5] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[6] Dennis Decoste and Bernhard Schölkopf. Training invariant support vector machines. *Machine Learning*, pages 161–190, 2002.

[7] Olivier Duchenne, Armand Joulin, and Jean Ponce. A Graph-matching Kernel for Object Categorization. In *IEEE International Conference on Computer Vision*, 2011.

[8] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 2008.

[9] Pedro F. Felzenszwalb, Ross B. Girshick, David A. McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1627–1645, 2010.

[10] Zhouyu Fu, Antonio Robles-Kelly, and Jun Zhou. Mixing linear svms for nonlinear classification. *Neural Networks*, pages 1963–1975, 2010.

[11] T. Graepel, R. Herbrich, Bernhard Schölkopf, A. Smola, P. Bartlett, K. Müller, K. Obermayer, and R. Williamson. Classification on proximity data with lp-machines. In *Neural Information Processing Systems*, 1999.

[12] Thore Graepel, Ralf Herbrich, Peter Bollmann-Sdorra, and Klaus Obermayer. Classification on pairwise proximity data. In *Neural Information Processing Systems*, 1998.

[13] Bernard Haasdonk. Feature space interpretation of svms with indefinite kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2005.

[14] Yuh jye Lee and Olvi L. Mangasarian. Rsvm: Reduced support vector machines. In *Data Mining Institute, Computer Sciences Department, University of Wisconsin*, 2001.

[15] S. Sathiya Keerthi, Olivier Chapelle, and Dennis DeCoste. Building support vector machines with reduced classifier complexity. *Journal of Machine Learning Research*, pages 1493–1515, 2006.

[16] Daniel Keysers, Thomas Deselaers, Christian Gollan, and Hermann Ney. Deformation models for image recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1422–1435, 2007.

[17] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[18] Li Liao and William S. Noble. Combining Pairwise Sequence Similarity and Support Vector Machines for Detecting Remote Protein Evolutionary and Structural Relationships. *Journal of Computational Biology*, pages 857–868, 2003.

[19] Cheng Soon Ong, Xavier Mary, Stéphane Canu, and Alexander J Smola. Learning with non-positive kernels. In *International Conference on Machine Learning*, 2004.

[20] Ryan Rifkin and Aldebaro Klautau. In defense of one-vs-all classification. *Journal of Machine Learning Research*, pages 101–141, 2004.

[21] Walter J. Scheirer, Michael J. Wilber, Michael Eckmann, and Terrance E. Boult. Good recognition is non-metric. *CoRR*, 2013.

[22] Ingo Steinwart. Sparseness of support vector machines – some asymptotically sharp bounds. In *Neural Information Processing Systems*, 2004.

[23] Christopher Williams and Matthias Seeger. The effect of the input density distribution on kernel-based classifiers. In *International Conference on Machine Learning*, 2000.

[24] Mingrui Wu, Bernhard Schölkopf, and Gökhan Bakir. A direct method for building sparse kernel learning algorithms. *Journal of Machine Learning Research*, pages 603–624, 2006.

[25] Weilong Yang, Yang Wang, Arash Vahdat, and Greg Mori. Kernel latent svm for visual recognition. In *Neural Information Processing Systems*, 2012.

Supplementary Materials

Margin Analysis of Basis Expanding SVM

As pointed out in the paper, we analyze the margin of BE-SVM in case of unnormalized features ($\tilde{\varphi}(.)$ instead of $\varphi(.)$)[5] and a non-squared Hinge loss. Given the corresponding dual variables, the margin of the BE-SVM was mentioned to be

$$M_{BE}(\beta) = \left(\beta^{\mathrm{T}}\mathbf{Y}\mathbf{S}_{BX}^{\mathrm{T}}\mathbf{S}_{BX}\mathbf{Y}\beta\right)^{-1} \tag{15}$$

as opposed to that of the kernelized SVM

$$M_{K}(\alpha) = \left(\alpha^{\mathrm{T}}\mathbf{Y}\mathbf{K}\mathbf{Y}\alpha\right)^{-1} \tag{16}$$

For comparison, the margin of the Nyströmized method is

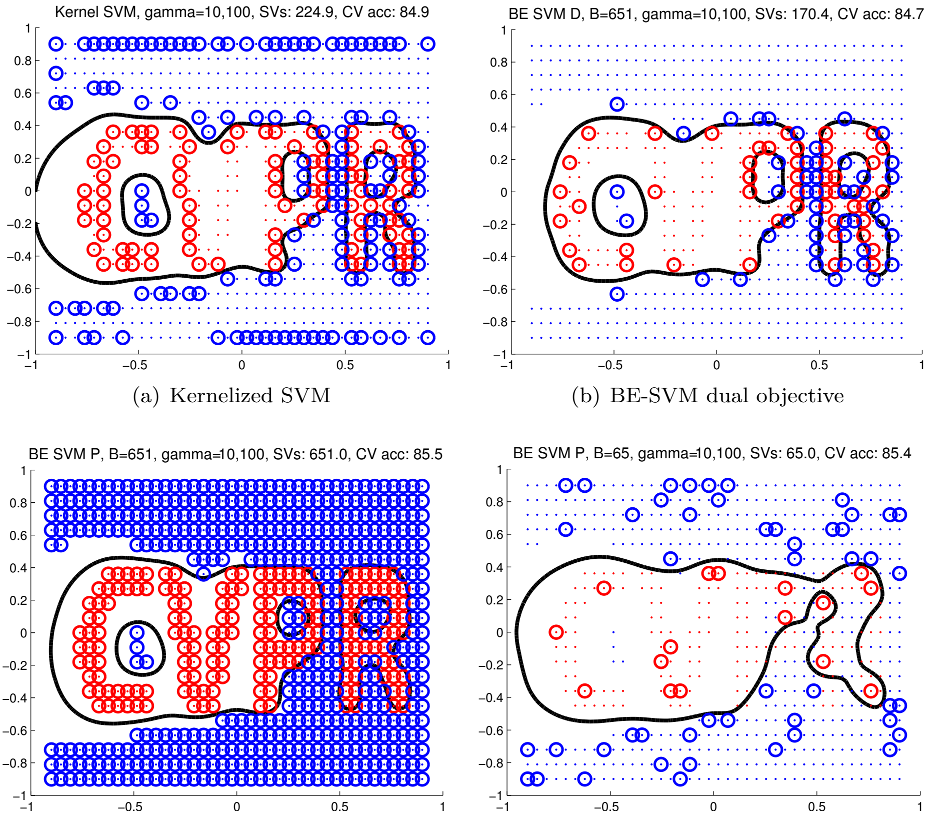
$$M_{N}(\alpha) = \left(\alpha^{\mathrm{T}}\mathbf{Y}\mathbf{K}_{XB}\mathbf{K}_{BB}^{-1}\mathbf{K}_{BX}\mathbf{Y}\alpha\right)^{-1} \tag{17}$$

BE-SVM vs Kernelized SVM: When $s(.,.) = k(.,.)$ and all training exemplars are used as bases, the margin of the BE-SVM will be $\left(\beta^{\mathrm{T}}\mathbf{Y}\mathbf{K}^2\mathbf{Y}\beta\right)^{-1}$. Comparing to the margin of SVM, for the same parameter C and the same kernel, it can be said that the solution (and thus the margin) of BE-SVM is even more derived by large eigenpairs, and even less by small ones. It is straightforward to verify $\mathbf{K}^2 = \sum_i \lambda_i^2 \mathbf{v}_i \mathbf{v}_i^{\mathrm{T}}$. Therefore, the contribution of large eigenpairs, that are $\{(\lambda_i, \mathbf{v}_i) | \lambda_i > 1\}$, to $\mathbf{K}^2$ is amplified. Similarly, the contribution of small eigenpairs, that are those with $\lambda_i < 1$, to $\mathbf{K}^2$ is dampened.

BE-SVM vs Nyströmized method: When $s(.,.) = k(.,.)$ and a subset of training exemplars are used as bases (reduced settings), the resulting margin of BE-SVM is $\left(\beta^{\mathrm{T}}\mathbf{Y}\mathbf{K}_{XB}\mathbf{K}_{BX}\mathbf{Y}\beta\right)^{-1}$. Comparing to the margin of the Nyströmized method, we can say that the most of the difference between the Nyströmized method and BE-SVM, is the normalization by $\mathbf{K}_{BB}^{-1}$.

For covariance kernels, that the Nyströmized method is most suitable for, $\mathbf{K}_{BB}$ is the covariance of the basis set in the feature space. Therefore, it can be said that the normalization by $\mathbf{K}_{BB}^{-1}$ essentially de-correlates the bases in the feature space. Although this is an appealing property, as pointed out in section 2.2 of the paper, there is no associating RKHS with indefinite similarity measures and the de-correlation in such cases is non-trivial. In case of covariance kernels, it can be said that BE-SVM assumes un-correlated bases, while bases are always correlated in the feature space. As larger sets of bases usually result in more (non-diagonal) covariances, the un-correlated assumption is more violated with large set of bases. The consequence is that in such cases, that are covariance kernels with large set of bases, BE-SVM can be expected to perform worse than the Nyströmized method. However, for sufficiently small set of bases, or in case of indefinite similarity measures, there is no reason for superiority of the Nyströmized method. In such cases and in practice, BE-SVM is competitive or better than the Nyströmized method. We provide quantitative analysis regarding this, later in section 4; in figure 8.

[5] Refer to equations (7) and (8) in section 2.3 of the paper.



(a) Kernelized SVM

(b) BE-SVM dual objective

(c) BE-SVM primal objective (approximate)

(d) BE-SVM primal objective (reduced, approximate)

Figure 5: Demonstration of kernelized SVM and BE-SVM using two Gaussian RBF kernels with $\gamma_1 = 10, \gamma_2 = 10^2$ and $C = 10$. 5(a) is based on equally weighted kernels. 5(b) is without normalization. 5(c) is with normalization. 5(d) is with normalization on 10% of the data randomly selected as bases. 10 fold cross validation accuracy and the number of support vectors are averaged over $i = 1 : 20$ scenarios based on the same problem but with different spatial noises. The noise model for i^{th} scenario is a zero mean Gaussian with $\sigma_i = 10^{-2}i$. The visualization is on the noiseless data for clarity. Best viewed electronically.

Demonstration on 2D Toy data

Figure 5 visualizes the use of multiple Gaussian RBF kernels in BE-SVM and kernelized SVM. We point out the following observations.

1) the BE-SVM primal objective (approximate) using an ℓ_2 regularization, does not

result in sparse solutions: all exemplars in 5(c) are used as support vectors (non-zero coefficients). However, based on the same objective, but in a reduced basis set setting (5(d)), the solution will be sparse by construction.

2) the dual objective of BE-SVM (exact) tends to result in sparser solutions as measured by non-zero support vector coefficient (compare 5(a) with 5(b)). We believe the main reason for this to be the modification of the eigenvalues as described in section 4. Note however that in order to classify a new sample, its similarity to all training data needs to be evaluated, irrespective of the sparsity of the BE-SVM solution (see equation (11) in the paper). In this sense, the BE-SVM dual objective results in completely dense solutions, similar to the primal BE-SVM objective. However, the solution can be made sparse by construction, by reducing the basis set, similar to the case with the primal BE-SVM objective. We do not demonstrate this here, mainly because our main focus is on the (approximate) primal objective.

3) due to the definition of the (linear) kernel in BE-SVM (see equation (11) in the paper), the solution of the BE-SVM has an inherent bias with respect to the (marginal) distribution of class labels. In other words, the contribution of each class to the norm of $\tilde{\varphi}(.)$, and consequently to the value of $\tilde{K}(.,.)$, directly depends on the number of bases from each class. Consequently, the decision boundary of BE-SVM is shifted towards the class with less bases: compare the decision boundaries on the left sides of 5(a) and 5(b). The centering step in normalization of BE-SVM (see equation (8) in the paper) helps alleviate this to some extent (compare the decision boundaries in 5(b) and 5(c)). In experiments on CIFAR-10 dataset, as the number of exemplars from different classes are roughly equal, this did not play a crucial role.

We provide the following speculations for the observed difference between the performance of kernelized SVM and BE-SVM. The solution of the kernelized SVM tends to be more accurate for low noise scenarios (similar to the depicted case), where a smooth function, as defined by the Gaussian RBF kernels, can separate the clean toy data with a larger margin in comparison to the cases based on perturbed data. On the contrary, in high noise scenarios, where the smoothness of the decision boundary is not particularly optimal, BE-SVM tends to perform better. We observed this when experimenting with the depicted 2D toy data. We need to point out that in higher dimensional cases, as is the case with the HOG features with hundreds of dimensions, samples are further away from each other, and the density is lower than it is in lower dimensional cases. In such cases, kernelized SVM would result in better solutions than a BE-SVM based on a full basis set: for a HOG cell size of 8 pixels, the performance of kernelized SVM based on one fold of training set is .65 where BE-SVM results in an accuracy of .63 (see figures 1 and 4 in the paper).

BE-SVM's Normalizations

In this section, we quantitatively evaluate the normalization suggested for BE-SVM (8), and compare it to a few other combinations. Particularly, we consider various normalizations of the HOG feature vectors, and similarly, various normalization schemes for the empirical kernel map $\tilde{\varphi}$ (7). We consider the following normalizations:

- No normalization (Unnorm)
- Z-Scoring, namely centering and scaling each dimension by the inverse of its standard deviation (Z-Score)

- BE-SVM normalization, namely centering and scaling all dimensions by the inverse average ℓ_2 norm of the centered vectors (BE-SVM)

We report test performances for all combinations of normalizations for the feature vectors and the empirical kernel maps, for two cases: 1) when $C = 1$, and 2) when the C parameter is cross-validated from $\mathcal{C} = \{10^{-1}, 10^0, 10^1\}$. In both cases, $|\mathcal{B}| = 10 \times 100$ bases were uniformly sub-sampled from the first fold of the training set (Indx basis selection described in Section 3.4).

Figure 6 shows the performance of BE-SVM in combination with different normalizations of the feature vectors and empirical kernel maps, and for different similarity measures. On top, reported numbers are for $C = 1$ while on the bottom, C is cross validated. It can be observed that the BE-SVM's normalization works best both for the feature and empirical kernel map normalizations. Although z-scoring is more suitable for linear similarity measures (compare BE-SVM + BE-SVM with Z-SCORE + BE-SVM in H4L, H8L, H4(x,y) and H8(x,y)), overall BE-SVM's normalization of the feature space works better than the alternatives. Particularly, in single similarity measure cases, it seems that normalizing the feature according to the BE-SVM's normalization is more important than normalizing the empirical kernel map. While the cross-validation of the C parameter marginally affects the performance, it does not change the conclusions drawn from the $C = 1$ case.

Figure 7 shows the performance of BE-SVM in combination with different normalizations of the feature vectors and empirical kernel maps, and for different combinations of similarity measures (the sequence of greedily augmented similarity measures $\mathcal{M}_2$: the set of two resolution similarity measures described in Section 3.5). It can be observed that BE-SVM's normalization of the kernel map is much more important and effective when combining multiple similarity measure (compare to Figure 6) .

These observations quantitatively motivate the use of BE-SVM's normalization with the following benefits, at least on the dataset we experimented on:

- It removes the need for cross-validation for tuning the C parameter, and mixing weights for different similarity measures.
- As the feature vector is centered and properly scaled, the linear SVM solver converges much faster than the unnormalized case, or when $C >> 1$.
- It results in robust learning of BE-SVM which can efficiently combine different similarity measures *i.e.* RBF kernels (H8R), and linear deformable similarity measures (H4(2,0), H4(0,1), H8(1,0)).

More Quantitative Analysis on CIFAR-10

Comparison of Nyström and BE-SVM Normalizations

Figure 8 shows accuracy of BE-SVM using different similarity measures and different basis selection strategies; for various sizes of the basis set. It can be seen that the difference between the performances of Nyström and BE-SVM normalizations becomes less significant for smaller basis set sizes, and more significant for larger ones. As argued in the paper, the Nyström normalization tends to be better for low resolution PSD kernels (H8R and H8L), but worse in other cases.

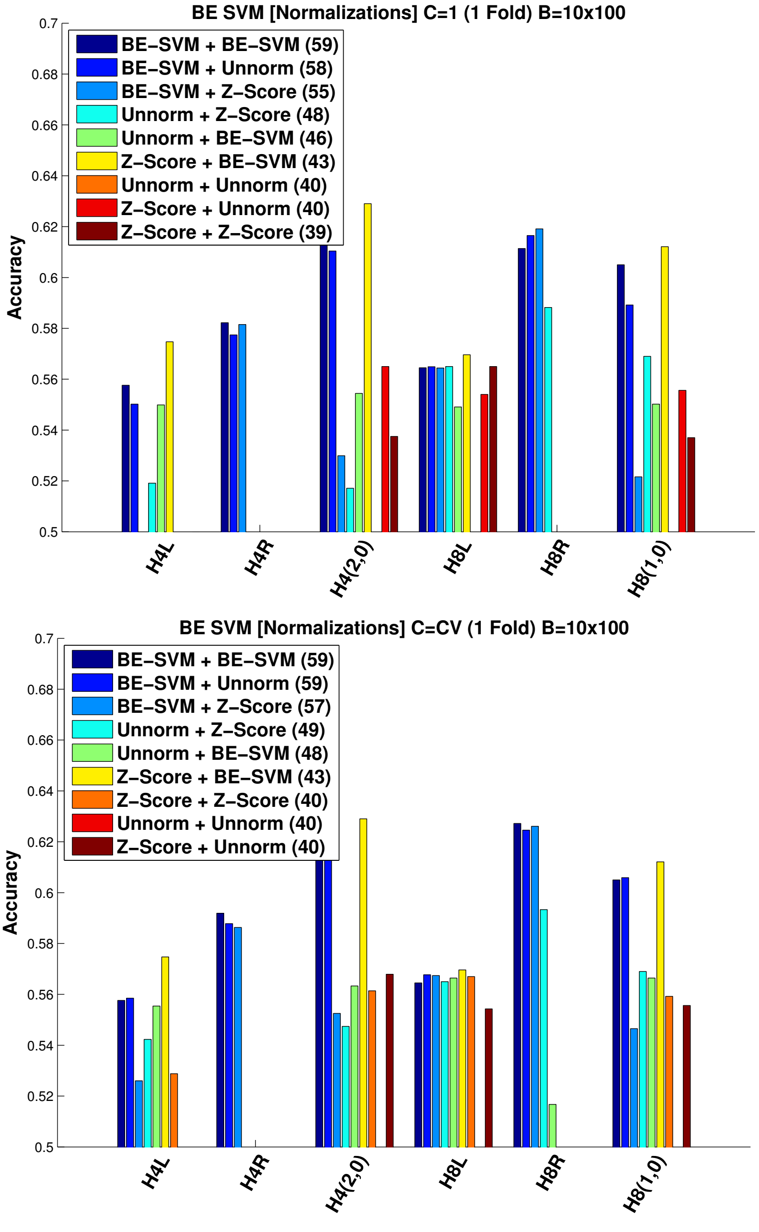
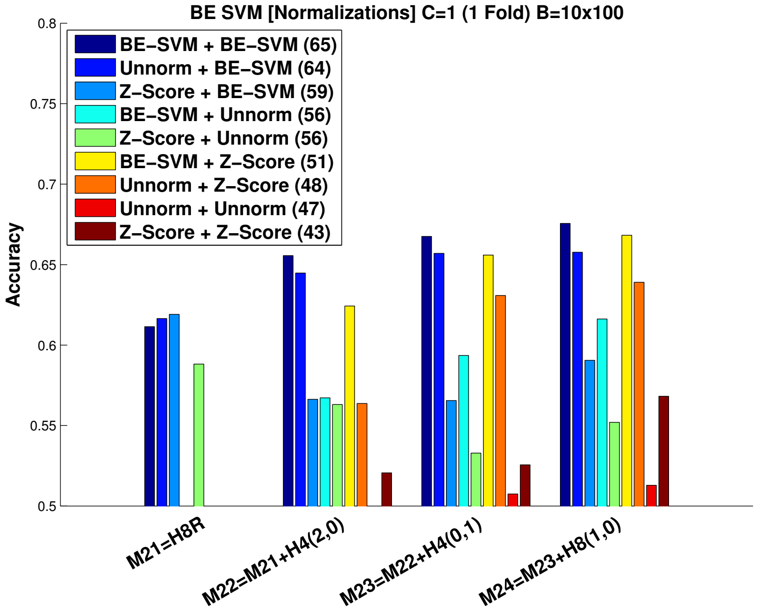


Figure 6: Performance of BE-SVM for different normalization schemes of the feature vector and the empirical kernel map, and different similarity measures. “F + K (P)” in the legend reflects using F and K normalization schemes for the feature vectors and the empirical kernel maps respectively, which results in the average test performance of P (averaged over the similarity measures).



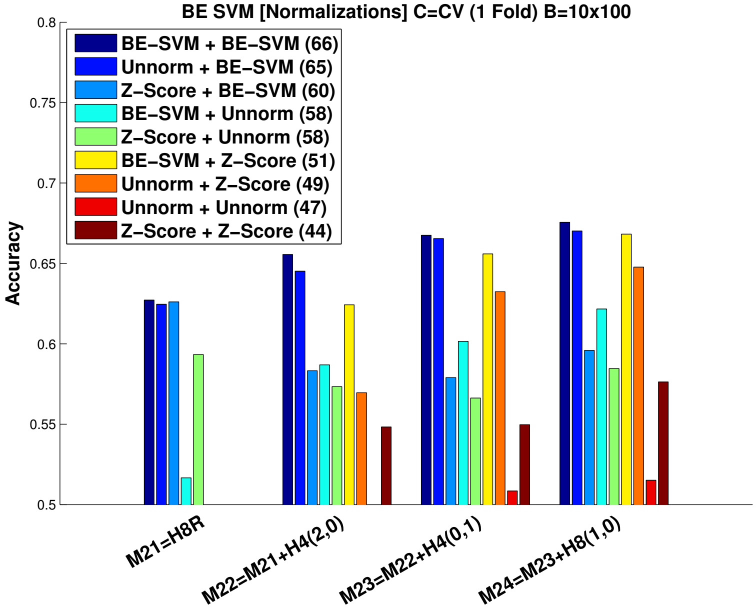


Figure 7: Performance of BE-SVM for different normalization schemes of the feature vector and the empirical kernel map, and different combinations of similarity measures. “F + K (P)” in the legend reflects using F and K normalization schemes for the feature vectors and the empirical kernel maps respectively, which results in the average test performance of P (averaged over the combinations of similarity measures).

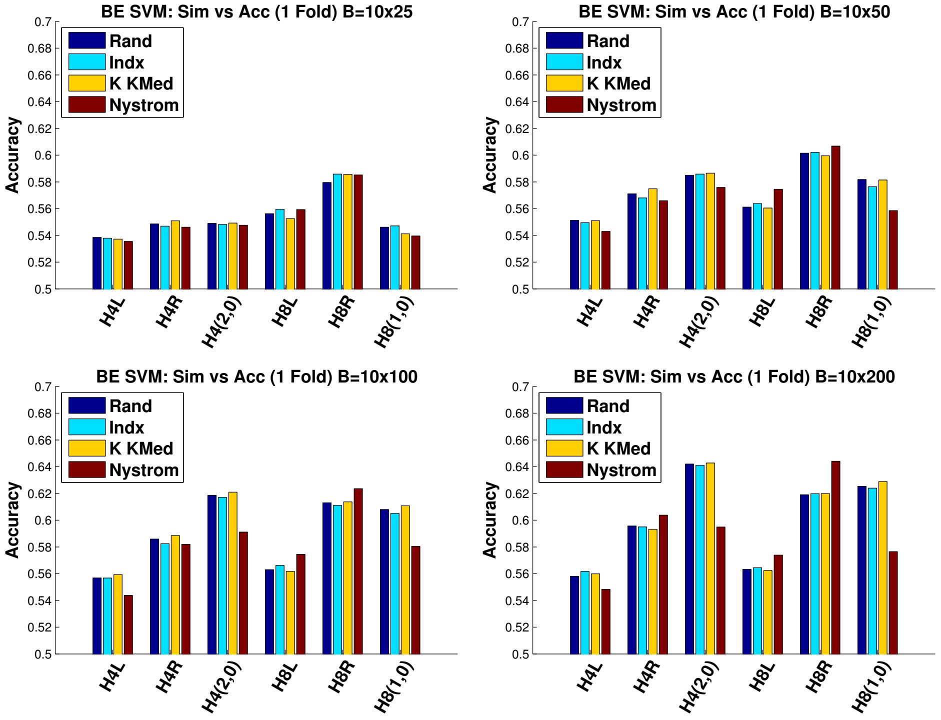


Figure 8: Performance of various basis selection strategies for BE-SVM using various similarity measures for various sizes of the basis set. The figure is similar to figure 2 in the paper, but with different basis set sizes.

Multiple Kernel Learning with PSD Kernels

We tried Multiple Kernel Learning (MKL) for kernelized SVM with PSD kernels. When compared to sophisticated MKL methods, we found the following procedure to give competitive performances, with much less training costs. Defining $K_C(.,.) = \alpha K_1(.,.) + (1-\alpha)K_2(.,.)$, our MKL approach consists of performing a line search for an optimal alpha $\alpha \in \{0, .1, \dots, 1\}$ which results in best 5-fold cross validating performance. Using this procedure, linear kernels were found not to contribute anything to Gaussian RBF kernels. The optimal combination for high resolution and low resolution Gaussian RBF kernels (K4R and K8R) resulted in a performance gain of less than 0.5% accuracy in comparison to K8R. We founds this insignificant, and did not report its performance, considering the fact that the number of parameters increases approximately 4 times using this approach.

Performance of Multiple Similarity Measures

Figure 9 shows the results of using multiple similarity measures with BE-SVM using the same approach as described in section 3.5 of the paper. It can be observed that using

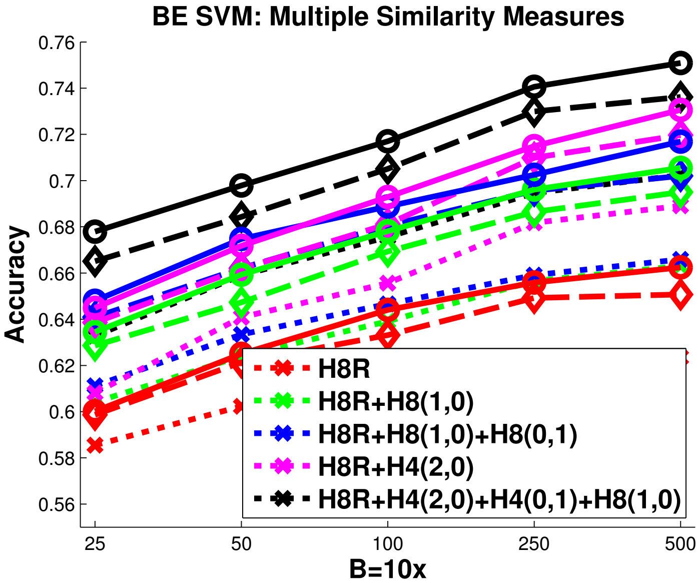


Figure 9: Performance of BE-SVM using multiple similarity measures for various sizes of the basis set. Results with dotted, dashed, and solid lines represent 1, 3, and 5 folds worth of training data. See text for analysis.

(invariant) indefinite similarity measures can significantly increase the performance of the model: compare the red curve with any other curve with the same line style. For example, using all the training data and a two resolution deformable approach results in 8-10% improvements in accuracy in comparison to the best performing PSD kernel (H8R). Furthermore, the two-resolution approach outperforms the single resolution approach by approximately 3-4% accuracy (compare blue and black curves with the same line style).