DISS. ETH NO. XXXX

Towards Cost-Effective and Performance-Aware Vision Algorithms

A dissertation submitted to

ETH ZURICH

for the degree of

Doctor of Science (Dr. sc. ETH Zürich)

presented by

Dengxin Dai
Master of Science
born November 15, 1986
citizen of China

accepted on the recommendation of

Prof. Dr. Luc Van Gool, examiner
Prof. Dr. Bernt Schiele, co-examiner

2016

DEDICATION GOES HERE...

Abstract

Computer vision has leaped forward during the last decade, and now is able to recognize objects of thousands of categories and reconstruct 3D scenes at city- or world-scale. However, the field still has to find means to keep up with the exploration of the massive amounts of data being captured on a daily basis. This is mainly due to the lack of sufficient training annotations and the lack of computational resources. The thesis is dedicated to mitigate the problem.

Firstly, we elaborate two strategies to reduce the annotation costs in order to train vision algorithms: (1) developing smart annotation approaches for efficient, large-scale annotations; and (2) learning better feature representations using unlabeled data; We develop algorithms for the strategies and show - in the context of recognition tasks - that they are able to considerably reduce the annotation costs for the training of recognition algorithms.

Secondly, in addition to reducing annotation cost, we also examine how to reduce the computational cost associated with the training and testing of vision algorithms. This research has lead to two contributions: (1) two efficient solvers for linear and kernel SVM+, significant speeding up the training process of SVM+ to explore privileged information; and (2) a method to allow computationally cheap features to imitate alternative features that perform better but are computationally more expensive. The imitation significantly improves the performance of the cheap features while retaining their efficiency.

Thirdly, as images keep growing in size, vision algorithms need to be more intelligent and self-aware of their performance. To this aim, we have developed two approaches: (1) Performance Forecasting to cheaply predict the success of vision algorithms for particular samples, which can be used for better resource allocation; and (2) Scale-Aware Image Segmentation to re-organize image segmentation hierarchies to better couple hierarchy depth and segmentation scale. The two methods also show potentials in reducing computational time of consecutive vision algorithms.

Zusammenfassung

Zusammenfassung geht hier...

Acknowledgements

Acknowledgement goes here.

Contents

1

Introduction

A picture is worth a thousand of words, and often comes with a story that the photographer wanted to deliver. We probably still remember the moments when our grandparents were telling the stories behind the old pictures at our home, be it about birthday parties, be it about holidays. The purpose of taking pictures has been extended dramatically in the last decades, especially since the invention of digital cameras and the Internet. The inventions make it effortless ever to take pictures, store them, and share them with friends and families. While still used to remember such special moments of our daily life, most images and videos are taken now to support many other practical applications — cameras have been widely used for public security, for product control in factories, for product description in E-commerce, for real-world object reconstruction, for robot control, to name a few.

1.1 Motivation

While the quantity and the quality of images taken everyday are steadily increasing, it is hard to take its full advantage if the data is unorganized. In computer vision, algorithms are developed to tackle the fundamental problems of processing and understanding the visual data, in order to better suits the application context. The most extensively researched problems include image classification, object detection, action recognition, human pose estimation, object tracking in videos, 3D object reconstruction. With algorithms from these sub-fields, images and videos can then be converted into semantic information so that the data can be used to its full potential.

The last decade has witnessed a great progress in computer vision, which comes as an outcome of three factors: 1) introduction of large-scale human annotations, such as ImageNet [Deng et al., 2009] for image classification, PASCAL VOC [Everingham et al., 2010] and MSCOCO [Lin et al., 2014] for object detection, KITTI [Geiger et al., 2013] for road scene understanding; 2) development of sophisticated, statistical learning approaches

including Support Vector Machines, Adaboost, Random Forests, and Deep Neural Networks; and 3) accessibility to powerful computing infrastructures such as the GPUs and many software frameworks, such as CuNN, Caffe, Torch, and Theano. These advantages have made the field achieved multiple milestones: computer vision is able to recognize objects of thousands of categories in high accuracy [Krizhevsky et al., 2012a; Simonyan & Zisserman, 2015], transcribe images/videos directly into natural languages [Vinyals et al., 2015], and generate realistic images [Gregor et al., 2015].

Despite the great achievement, the field is still left behind by the exploration of visual data. Visual content is exploding – millions of videos are generated and consumed every second, from the footage of surveillance cameras to the over two billion images and videos uploaded daily to social platforms. However, intelligent and insightful understanding of these data is still far from being reached. We believe there are at least three reasons behind this: 1) visual annotation is expensive to obtain; 2) algorithms in computer vision are often computationally heavy; and 3) developed methods often are not performance-aware.

The first reason is evidenced by the fact that most of the benchmarks still only cover a small portion of the visual data that is available in the wild, and are collected under quite controlled scenarios. The speed of visual annotation cannot match that of visual data acquisition — taking pictures, storing and sharing them are effortless nowadays, while annotating visual data in the form that machine can learn on is tedious and very time-intensive. For example, YouTube has 100 hours of videos (100 million frames) updated every minute, while it required 19 man-years to label 1.2 million internet images [Deng et al., 2009]. The annotations are done in the form of bounding boxes, rather than in more dedicated forms such as full segmentation masks.

The second reason follows the fact that visual data is very high-dimensional, rendering the algorithms computationally heavy in both training and testing. For instance, training a state-of-the-art image classifier [Simonyan & Zisserman, 2015; He et al., 2015] can take days or weeks even with modern GPUs, computing the optical flow of a standard video dataset [Karpathy et al., 2014] can simply take several days, and computing the similarity among images of a large dataset can be very expensive as well [Gong et al., 2013]. This heavy computation hinders the community from quickly exploring new models, easily deploying the trained models to power-limited devices, and readily scaling the developed models.

The last one is due to the trend that the community is mostly focused on developing the next best method for task X, and rarely on improving the self-awareness of the methods, be it a measure of model uncertainty [Kendall et al., 2015; Kondermann et al., 2008], be it the usefulness to down-streamed vision tasks [Yao et al., 2011; Jain et al., 2015]. This performance blindness leads to ineffective solutions in many situations, such as the same amount of resource is allocated to every image regardless of its complexity (difficulty), and the advance made in all the sub-fields cannot be synergized effectively.

1.2 Contributions

The aforementioned problems deliver a strong need to reduce the annotation cost and computational cost of current computer vision methods, and to learn to predict their performance. This thesis is dedicated to provide a collection of methods to attack these problems from multiple perspectives.

First, we elaborate two strategies to reduce the annotation cost:

- Developing efficient visual annotation approaches. A natural and efficient annotation method is developed for object recognition by letting annotators speak. Since drawing scribbles and speaking are very natural to human, our method unleashes the expressive ability of annotators and solves the *what* and *where* problems of object annotation both at the same time, leading to an approach which draws a good trade-off between recognition accuracy and annotation cost.
- Learning feature representation with unlabeled data. A new method is developed to learn a new feature representation on top of standard feature representations. The leaning takes advantage of discriminative learning and ensemble learning, and is able to generate new features specifically tailored to the data at hand.

Secondly, we examine how to reduce the computational cost associated with the training and testing of vision algorithms:

- Developing efficient training algorithms to the standard approach SVM+. SVM+ has shown excellent performance in visual recognition tasks for exploiting privileged information in the training data. We propose two efficient algorithms for solving the linear and kernel SVM+. Experiments show that our proposed algorithms achieve significant speed-ups to the state-of-the-art solvers for SVM+.
- Proposing Metric Imitation (MI), a method to allow computationally cheap features to imitate alternative features which perform better but are computationally more expensive. The leaned transformation significantly boost the performance of cheap features while retaining their efficiency.

Lastly, we investigate performance-aware vision algorithms by making the following contributions:

- Predicting how likely vision algorithms succeed on particular samples. It is true for every vision task that not all images are equally difficulty. We examine how to learn this on the task of example-based texture synthesis (ETS). ETS has been widely used to generate high quality textures of desired sizes from a small example.

However, not all textures are equally well reproducible that way. We predict how synthesizable a particular texture is by ETS and find that texture synthesizability can be learned and predicted efficiently.

- Learning scale-aware image segmentation. Hierarchical image segmentation provides segmentation at different scales in a single tree-like structure. However, they are not aware of the scale information of the regions in them. As such, one might need to work on many different levels of the hierarchy to find the objects in the scene. This work predicts the scales of the regions to modify their depth in the tree to better couple tree depth and region scale. The output of our method is an improved hierarchy, which improves the quality of the hierarchical segmentation representations.

- Evaluating the usefulness of image super-resolution methods to other computer vision tasks. Although it might be believed that image super-resolution is helpful for other vision tasks, this work has formalized the conception and conducted quantitative evaluation. The work can serve as an inspiration for the community to evaluate image super-resolution with respect to the helpfulness to other vision tasks, and to apply it as a pre-processing component if the input images are of low-resolution.

1.3 Organization

Since extensive research has been done in similar spirit, the thesis begins with Chapter 2 examining related work in a broad context. Our developed approaches are presented in Chapters 3 – **??**. They are written to be generally self-contained and can be read independently. Finally, Chapter 5 concludes this thesis. A detailed overview of the remaining chapters follows:

In Chapter 2, *Related Work*, we provide a short literature overview of previous art in the direction of reducing the annotation cost and computational cost of vision algorithms and towards self-aware algorithms.

In Chapter 3, *Efficient Visual Annotation by Speech Recognition*, we present our efficient visual annotation approach Draw&Tell and show its efficiency in the context of semantic image segmentation. In order to solve the *what* and *where* problems in visual annotation both at the same time, we let annotators speak the name of the object while they draw strokes on it. The speech is recognized by a speech recognition engine specifically trained for the purpose, and an extension to the fully convolutional neural network is made to learn from the stroke-based 'coarse' annotation. The approach draws a good trade-off between recognition accuracy and annotation cost. The work in this chapter was originally presented in [Dai et al., 2016a].

In Chapter 4, *Representation Learning with Unlabeled Data*, we present our motivation and method of learning a new feature representation with unlabeled data. We then evaluate the method in the context of semi-supervised image classification and image clustering. This work was originally presented in [Dai et al., 2012a] and in [Dai & Van Gool, 2013].

In Chapter **??**, *Fast Training Algorithms for SVM+*, we present two efficient algorithms for training the linear and kernel SVM+. New problems with fewer constraints are formulated in the dual domain, making the problem solvable efficiently by the SMO algorithm of one-class SVM. Experiments show that our proposed algorithms are significantly faster than the the state-of-the-art solvers for SVM+. This work was originally presented in [Li et al., 2016].

In Chapter **??**, *Metric Imitation for Efficient Distance Computation*, we present a method called Metric Imitation (MI) to efficiently compute the distance among images. MI learns a transformation to cheap features so that the L2 distance of the transformed features can approximate L2 distance of better-performing features which are more computationally expensive to compute. The method is evaluated on multiple vision tasks. The method was originally presented in [Dai et al., 2015a].

In Chapter **??**, *Performance Prediction: Succeed or Fail?*, we present our approach of predicting the success of example-based texture synthesis. To this aim, we collected a dataset with $21,302$ annotated textures and annotated them according to the synthesizability — the quality of synthesized results by texture synthesis methods. A set of relevant features are then defined to regress the value of synthesizability. Extensive experiments show that texture synthesizability is learnable. This work was originally presented in [Dai et al., 2014].

In Chapter **??**, *Performance Prediction: Under-, Properly-, or Over-Processed?*, we present a method to predict the scale of image segments relative to the corresponding objects, and to then apply the prediction to re-align the results from general hierarchical image segmentation so that the depth in the tree structure and the scale of the regions is better coupled. The output of our method is an improved hierarchy, which improves the quality of the hierarchical segmentation representations. The work was originally presented in [Chen et al., 2016].

In Chapter **??**, *Performance Evaluation: Helpful for Other Tasks?*, we evaluate the usefulness of image super-resolution methods to other computer vision tasks. Sixes state-of-the-art computer vision algorithms are evaluated on four popular computer vision tasks: boundary detection, semantic image segmentation, digit recognition, and scene recognition. The work confirms that image super-resolution is helpful for other vision tasks when the state-of-the-art approaches are used. The work was originally presented in [Dai et al., 2016b].

2

Related Work

The community has made great efforts in reducing the annotation cost and computational cost of vision algorithms by using a diverse set of techniques. There is also a large body of literature working towards self-aware vision algorithms. This section summarizes the related topics in a broad context. Related work to each of our specific work is discussed in the corresponding chapter.

2.1 Reducing Annotation Cost

Datasets play a critical role in computer vision. They qualitatively 'define' the learning tasks and guide research directions, which has been proved multiple times in the history of computer vision [Baker et al., 2011; Everingham et al., 2010; Deng et al., 2009]. We limit ourselves to annotations for object recognition. Training annotations for object recognition often come as bounding boxes or full segmentation masks (c.f. Fig.**??**). The most popular ones fall into this category: CamVid [Brostow et al., 2009] and Cityscapes [Cordts et al., 2016] for urban scenes, NYU [Silberman et al., 2012] for indoor scenes, PASCAL [Everingham et al., 2010] and COCO [Lin et al., 2014] for general objects, and PASCAL-Context [Mottaghi et al., 2014] for objects in context. Creating datasets for object recognition is very expensive even with excellent annotation tools [Bell et al., 2013; Russell et al., 2008a]. As a result, methods were proposed to reduce the cost. For instance, [Deng et al., 2014; Lin et al., 2014] exploit the hierarchical structures of object classes to reduce annotation space. Other popular techniques can be roughly grouped into the following categories (with overlaps).

Weakly Supervised Learning
As stated, visual annotation is time-consuming. Many works have developed algorithms to learn from weakly annotated training data. The following are the outstanding examples. [Prest et al., 2012] learns object detectors from weakly labeled videos. [Pathak et al., 2015b; Papandreou et al., 2015] trains convolutional neural networks for semantic image segmentation with image-level annotations. [Khoreva et al., 2016] trains boundary

detector with bounding-box annotations. [Bilen & Vedaldi, 2016] trains object detectors with the training data of image classification. This stream of work proves that with careful design of algorithms, weak supervision is able to yield quite competitive performance as the full supervision. Weak supervision consumes far less effort, which makes the training more affordable. Recent datasets such as Cityscapes [Cordts et al., 2016] start providing annotations of different quality. In the same vein, our work in Chapter 3 tries to learn from weak supervision for semantic image segmentation.

Transfer Learning

Transfer learning is to transfer knowledge learned from one task to another task, or from one domain to another domain, when training data is scarce for the latter scenario. Successful applications in vision include knowledge transfer from videos to images [Kulis et al., 2011; Gopalan et al., 2011; Fernando et al., 2013], knowledge transfer from known classes to unseen classes [Lampert et al., 2009], supervision transfer [Gupta et al., 2015] from annotated RGB images to other data modalities such as depth and flow, and supervision transfer [Hoffman et al., 2014] from classification task to detection task. Transfer learning has gained great success recently in object recognition and has become the standard procedure in training (fine-tuning) deep neural networks [Girshick et al., 2014a; Long et al., 2015]. It successfully lifts the requirement of large training set for the task at hand, and considerably reduces the training time. Our work in Chapter **??** can be considered as a special case of transfer learning by transferring learned manifold from one domain to another.

Semi-supervised Learning

Semi-supervised learning (SSL) aims at enhancing the performance of recognition systems by exploiting an additional set of unlabeled data. SSL is especially helpful when the labeled training data is limited. Due to its great practical value, SSL has a rich literature [Chapelle et al., 2006; Zhu & Goldberg, 2009]. The research in this vein can be classified into four groups based on their underlying techniques: (1) Self-training scheme [Blum & Mitchell, 1998; Guillaumin et al., 2010; Shrivastava et al., 2012], where the system iterates between training recognition models with current 'labeled' training data and augmenting the training set by adding its highly confident predictions in the set of unlabeled data; (2) Label propagation [Zhu et al., 2003; Zhou et al., 2004; Belkin et al., 2006; Fergus et al., 2009; Ebert et al., 2012], where a graph is defined with nodes representing data examples and edges reflecting their similarities, and label information propagates over the graph; (3) Classifier regularization [Joachims, 1999; Bennett & Demiriz, 1998; Leistner et al., 2009; Weston et al., 2012], by enforcing the boundaries to pass through regions with a low density of data samples. Our work in Chapter 4 bases on the assumption of manifold smoothness to learn a new feature representation, and shows pleasant performance on semi-supervised image classification.

Active Learning & Human in the Loop

Active learning or recently Human in the Loop lets human and the computer work collectively to train good vision algorithms. Smart policy needs to be designed or learned in

order to take full advantage of machine systems and to minimize the human effort to machine's uncertainties. The policy varies significantly from task to task: from suggesting the most informative samples to annotate [Vijayanarasimhan & Grauman, 2012; Freytag et al., 2014], to how many object to display [Yao et al., 2012], and to what questions to ask [Russakovsky et al., 2015b; Papadopoulos et al., 2016]. The great success of this technique in computer vision is evidenced by the large amount of academic publications, mainly on image recognition [Joshi et al., 2009a; Branson et al., 2010; Collins et al., 2008], semantic image segmentation [Vezhnevets et al., 2012], and object detection [Russakovsky et al., 2015b; Papadopoulos et al., 2016].

Supervision from Multimodal Data & Web Data
In addition to human annotated training data, other sources of supervision have been exploited to train vision algorithms as well, such as text from web pages or newspapers [Berg et al., 2004; Gupta & Davis, 2008], eye-tracking data [Papadopoulos et al., 2014]. Webly-supervised learning [Chen et al., 2013; Chen & Gupta, 2015a; Divvala et al., 2014] has gained extensive attention in the recent years, where visual recognizer is trained automatically by the images returned by image search engines such as Google, Flickers, and Bing. These works show the great potentials of scaling visual recognition to billion-sized scale, without using human annotations.

Unsupervised Feature Learning
Another group of work in the spirit of learning with limited annotations aims to learn middle- or high-level image representation in an unsupervised manner. The supervision often comes from intelligent exploration of prior knowledge or common sense knowledge. For instance [Coates et al., 2011; Dosovitskiy et al., 2014] generates surrogate classes by clustering or performing transformation to local patches, [Doersch et al., 2015] employees the spatial relationships of image windows in an image, [Wang & Gupta, 2015] exploits the tracking results of objects in videos, and [Agrawal et al., 2015] leverages the ego-motion of cameras. These systems are all able to learn good feature representations without using human annotations. Our work in Chapter 4 learns new feature representations by exploiting the assumption of manifold smoothness from unlabeled data, which is similar to [Coates et al., 2011; Dosovitskiy et al., 2014].

2.2 Reducing Computational Cost

In addition to reducing annotation cost, we also investigate how to reduce the computational cost of vision algorithms. There are numerous great techniques towards efficient vision algorithms, especially those targeted for mobile and robotic applications. A complete overview of the topic is beyond the scope of this thesis, so we only summarize a group that is mostly related to our work Metric Imitation in Chapter **??**.

Model Compression, Imitation and Distillation
Often, we are faced with a dilemma where the best performing model is too slow and too large, and the efficient alternatives are not as accurate. This is exactly the place where model compression [Bucilua et al., 2006] comes into play. Model compression is often given other names such as distillation or imitation [Hinton et al., 2014; Romero et al., 2014; Rusu et al., 2016]. The high-level idea is to learn a fast and compact model to approximate or imitate the functional behaviors of a better-performing model which is slow to run. [Bucilua et al., 2006] successfully compresses ensemble classifiers 'into' a compact neural network, and [Hinton et al., 2014; Romero et al., 2014; Rusu et al., 2016] distills the knowledge learned by a deeper or wider neural networks into a lighter neural networks. [Xu et al., 2015b] proves that it is also possible to imitate the effects of edge-aware filters designed with domain knowledge by an efficient convolutional neural network. The results of these works suggest that models of different complexity are needed for learning and evaluation if efficiency is concerned. Complex models are required to extract knowledge form noisy observations; Compact models then distill the learned knowledge from the trained complex models for efficient evaluation. Our work Metric Imitation shares the spirit.

2.3 Self-aware Vision Algorithms

Predicting Model Confidence, Uncertainty or Failure
Performance-blind vision algorithms can be disastrous, as they can fail saliently without even throwing a warning. As such, this research topic is increasingly gaining more attention, though not very popular yet. Notable examples of learning model uncertainty or confidence include [Kendall et al., 2015] for semantic image segmentation, [Kondermann et al., 2008; Aodha et al., 2013] for optical flow, [Kopf et al., 2012] for image completion, [Hartmann et al., 2014] for patch matching, [Park & Yoon, 2015] for stereo, and [Zhang et al., 2014] for multiple applications such as vanishing line estimation and memorability. Our work texture synthesizability (Chapter **??**) enriches this repository by adding another example texture synthesis. Performance-aware algorithms carry other benefits as well. For instance, it can speed up down-streamed algorithms by adaptively allocating computing resource based on the complexity of image samples. Cascading [Viola & Jones, 2001; Felzenszwalb et al., 2010] and active inference [Liu & He, 2015] serve as standing examples.

The Usefulness of Task X for Task Y
Computer vision is a very challgening task, and research has to be conducted in many sub-fields. These sub-fields are intertwined intrinsically, and should benefit from each other in principle. However, this does not always happen in practice, because building a system of one task on top of another task introduces extra modes of errors as well. Thus, it is necessary for the community to regularly evaluate whether sub-field X has

advanced to a level that sub-field Y can generally benefit from it if integrated. Excellent research has been continuously done in this direction. For instance, [Hoiem et al., 2008] proves that surface orientation estimation is already good enough to be helpful in guiding object detection; [Yao et al., 2011] confirms that pose estimation is able to help action recognition; [Jain et al., 2015] analyzes how object detection helps action recognition when the state-of-the-art detectors run over 15000 object classes; [Martinovic et al., 2015] shows that 3D reconstruction of regular buildings is already accurate enough to perform object recognition directly on it. Our work in Chapter **??** consolidates that image super-resolution by state-of-the-art approaches is helpful for general vision tasks.

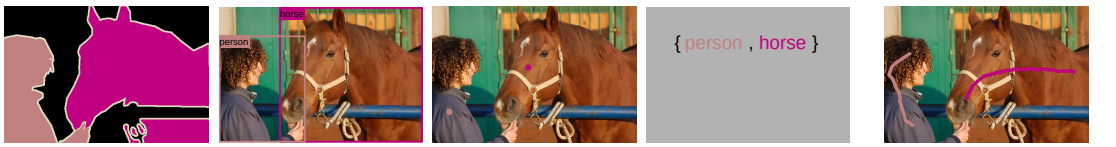
3

Efficient Visual Annotation with Speech Recognition

3.1 Introduction

Tremendous progress has been made during the last few years in object detection and semantic image segmentation due to (1) the success of training deep convolutional neural networks (CNNs) [Krizhevsky et al., 2012b; Donahue et al., 2014a; Simonyan & Zisserman, 2015] on large classification datasets and (2) the flexibility of transferring CNN models pre-trained for image classification to the task of object detection and semantic segmentation where not as much training data is available [Girshick et al., 2014a; Long et al., 2015; Chen et al., 2015]. Model transfer, often called fine-tuning, partially lifts the strong need for large training sets of CNNs. Yet, the fact remains that CNNs *intrinsically* call for large-scale training data to unleash their learning capacity. As such, we can expect that the limit of CNNs for the tasks is far from reached and that one of the obstacles is the cost of obtaining training samples.

Obtaining training data for object detection or semantic image segmentation consists of two main jobs: answering the *what* and *where* questions. *What* is to annotate what objects are present and *where* is to indicate their locations in the image. As to *what*, annotators usually need to choose class names from some form of menus with the mouse and/or keyboard [Bell et al., 2013]. The procedure can be costly especially when the number of classes considered is large. Imagine the amount of work for annotators to find the class name out of a list of hundreds of names for every object they annotate, not even to mention attributes, views, etc. In other systems [Bearman et al., 2015; Lin et al., 2014], binary questions ask whether object classes are present, which is also very expensive when the number of classes is large, though exploiting the hierarchy of classes may reduce the cost [Deng et al., 2014; Lin et al., 2014]. As to *where*, various forms of annotations have been used for training: from the full segmentation masks [Girshick et al., 2014a; Chen et al., 2015; Zheng et al., 2015], over bounding boxes [Xu et al., 2015a; Dai et al., 2015b; Pathak et al., 2015a] or just single points [Bearman et al., 2015], to nothing spatial at all



(a) mask [ ] (b) b-boxe [ ] (c) point [ ] (d) keyword [ ] **(e) scribble** [+]

Figure 3.1*: A comparison of our annotation (e) to other forms of annotations for semantic segmentation, ranging from (a) full segmentation masks, to (b) bounding boxes, to (c) single points, and to (d) image-level keywords. The devices to obtain these annotations are shown as well: others only use the mouse (+keyboard), while ours also uses voice input through a microphone.*

(image-level annotation) [Verbeek & Triggs, 2007; Papandreou et al., 2015]. As can be seen in Figure 3.1, the more specific the annotation, the more informative it is, but also the more expensive to obtain.

In this work , we present a novel annotation method, which strikes a balance between annotation cost and informativeness provided. In particular, annotators are asked to draw scribbles (strokes) on objects of interest, while saying aloud their class names (attribute if necessary). The scribbles are recorded to solve the *where* problem; the speech is recognized by a specially trained recognition engine to solve the *what* problem. In order to further increase the recognition accuracy of the scribbles (associating to object names), we integrate the speech recognition with a webly-supervised object recognition. The object recognition system is trained with images retrieved from image searching engine directly, and is applied to the image region where the scribble is drawn. An example of the annotation by our method is shown in Figure 3.1, along with that of other methods.

The method is inspired by two observations: (i) *what* and *where* are handled separately in previous methods. For instance, in [Bell et al., 2013] annotators first mask out a region and then assign a label to it; and in [Lin et al., 2014] annotators label the presence of objects in the first round, followed by positioning them in the second round. This separation is unnecessary and introduces extra cost, because the two problems are interdependent and solved together by the annotators vision. It seems the separation is due to the fact that only a single mode of input devices was used in the previous methods, which is the mouse (+keyboard). We lift this restriction by including the speech channel, which allows annotators to communicate with the computer more naturally and efficiently. (ii) Drawing scribbles is another natural and efficient ability, and has been widely used for interactive image segmentation. But to the best of our knowledge, it has not been applied to create training data for semantic image segmentation. We demonstrate that combining drawing and speaking can tackle both the *what* and *where* problems, leading to an efficient annotation method for semantic image segmentation. We call the method **Draw&Tell**.

Having obtained the annotations, we convert them to soft confidence maps of the corresponding classes by combining interactive image segmentation and ensemble learning.

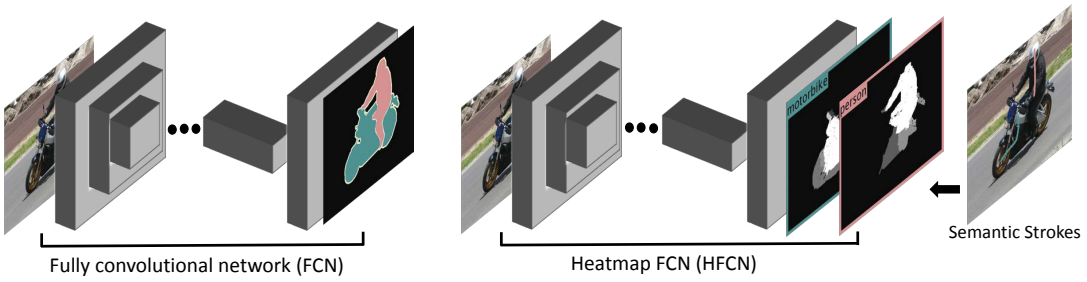


***Figure 3.2**: The framework of the standard fully convolutional network (FCN) [Long et al., 2015] and our heatmap-based FCN.*

We call these soft confidence maps semantic heatmaps. Finally, we extend the standard CNNs model [Long et al., 2015] to accommodate the soft semantic heatmaps as training data rather than the standard crisp labels. The pipeline of the method is shown in Figure 3.2 (right hand side), which also shows the standard fully convolutional network [Long et al., 2015] for ease of comparison. We show in experiments (i) that our annotation method is 11 times faster than pixel-wise annotation, and also faster than conventional image-level annotation; (ii) that our adapted CNNs trained with the annotations yields significantly better results than the CNNs trained with image-level training data, and yields results comparable with those of the CNNs trained with pixel-wise annotations; and (iii) that under the same annotation budget, our annotations, combined with our learning method, yield better results than the conventional precise annotations.

Our contributions are mainly: (i) a new method which combines webly-supervised vision and automatic speech recognition to efficiently create training data for semantic image segmentation, and (ii) a method to train CNNs with scribble-based training data, by converting scribbles to soft heatmaps and extending the standard neural networks. Introducing speech recognition into visual annotation is novel and our method can be used for other vision tasks as well, such as annotations for object detection, part detection, attribute detection and view estimation. The integration of webly-supervised object recognition and speech recognition can also serve as an example of combing vision and speech.

This chapter is organized as follows. Section 3.2 presents related work. Section 3.3 details the annotation method, followed by Section 3.4 which is devoted to the segmentation method. Section 3.5 then reports on the experiments and Section 3.6 concludes the paper.

3.2 Related Work

Previous work related to ours mainly falls into two groups: semantic image segmentation and integration of vision and language (speech).

3.2.1 Semantic Image Segmentation

Methods: There is a rich literature of semantic image segmentation (SIS) [Shotton et al., 2009; Krähenbühl & Koltun, 2011], and the field has made tremendous progress since CNNs were applied. The seminal R-CNN [Girshick et al., 2014a] and the follow-up systems [Long et al., 2015; Chen et al., 2015; Zheng et al., 2015] yield significantly better performance than previous methods. As noted above, SIS through CNNs has probably not come to full fruition yet due to the small size of the existing training sets. Several successful attempts have been made to use weaker supervision in order to reduce the annotation cost. For instance, [Pathak et al., 2015b] exploits multiple instance learning to train FCN [Long et al., 2015] with image-level annotations. [Papandreou et al., 2015; Dai et al., 2015b; Pathak et al., 2015a] leverage the power of object proposals to train FCN with object bounding-boxes. [Bearman et al., 2015] presents a system to train FCN with point annotation – objects are indicated by single points. In the same vein our work tries to train FCN with annotations that are efficient to obtain.

Supervision: Datasets play a critical role in computer vision. They qualitatively 'define' the learning tasks and guide research directions. Training annotations for SIS often come as full segmentation masks (c.f. Figure 3.1). The most popular ones fall into this category: CamVid [Brostow et al., 2009] for outdoor scenes, NYU [Silberman et al., 2012] for indoor scenes, PASCAL [Everingham et al., 2010] and COCO [Lin et al., 2014] for general objects, and PASCAL-Context [Mottaghi et al., 2014] for objects in context. Creating datasets for SIS is very expensive even with excellent annotation tools [Bell et al., 2013; Russell et al., 2008a]. As a result, methods were proposed to reduce the cost. For instance, [Deng et al., 2014; Lin et al., 2014] exploit the hierarchical structures of object classes to reduce annotation space. [Vijayanarasimhan & Grauman, 2012; Vezhnevets et al., 2012] exploit active learning techniques to suggest the most informative samples to annotate under a budget. [Papadopoulos et al., 2014] employs eye tracking systems to help create training data for object detection. Our proposal is to exploit speech recognition to help dataset creation for SIS.

3.2.2 Integration of Vision and Language (Speech)

Research interest in integrating vision and language has increased recently, e.g. for image/video caption generation [Vinyals et al., 2015; Guadarrama et al., 2013; Thomason et al., 2015; Karpathy & Fei-Fei, 2015; Kulkarni et al., 2013]. The objective is to learn from a corpus of sentences and images / video snippets to generate meaningful descriptions for new images. One of the main research topics is the alignment of representations from the two different domains: vision and language. In terms of language&vision understanding, our goal is more conservative, because the words we handle are restrictive and vision and language are aligned with human help. However, our purpose of the integration is different.

Speech-driven interfaces are going through a renaissance, with popular commercial products like Apple's Siri. The most relevant to our work are those integrating speech and vision. Pixeltone [Laput et al., 2013] and Image spirit [Cheng et al., 2014] are examples that use voice commands to guide image processing and semantic segmentation. The difference between image spirit and our work is that they use voice commands to refine labeling results and we use them to collect training data. Also, image spirit uses voice commands alone, while our method combines speech input with mouse interactions. There also is academic work [Srihari & Zhang, 2000; Kalashnikov et al., 2011; Hazen et al., 2007; Hoogs et al., 2003] and an app [smi, 2015] that use speech to provide image descriptions. Again, our purpose is very different.

3.3 Speech-based Annotation

3.3.1 The Draw & Tell Annotation Tool

This section describes our annotation method, which consists of two parts: drawing scribbles on the objects and telling the system their names. Drawing scribbles is straightforward and we follow popular interactive image segmentation (IIS) methods [Gulshan et al., 2010] to record the mouse tracks. As to the speech recognition, we draw on the state-of-the-art, end-to-end speech recognition systems [Graves & Jaitly, 2014], trained with bidirectional recurrent neural networks under the connectionist temporal classification loss function. The systems successfully avoid the significant human effort in creating the pronunciation dictionary. The characters are then decoded to words with a dictionary of desirable class names by the CTC Beam Search algorithm in [Graves & Jaitly, 2014]. Improving such a system itself is challenging and beyond the scope of this paper, thus we choose to treat the system as a black box and operate on an n-best list of possible object names for each utterance.

Speech recognition is a complex research area in its own right, and there are still problems in recognizing natural speech with high accuracy [Saini & Kaur, 2013]. We, however, want to use it to simplify the task of annotation, for which high accuracy is required. Fortunately, some constraints can be imposed in our case, which our annotation task naturally fulfills. The constraints and solutions are:

- Constraining the vocabulary and syntax of the utterances to ensure robust speech recognition. For our annotation, the vocabulary is restricted to the names of the objects of interest, and we can also instruct annotators to follow a specific syntax.
- Synchronizing the speech input with mouse input. We instruct the annotator to only say the object names while they draw the scribble on the corresponding object, not at any other time. The speech recognition engine uses this synchronization to better identify speeches relevant to the object being annotated.

Figure 3.3: *The interface of Draw&Tell. Annotators can draw scribbles on objects of interest and speak their names in the meanwhile. The names of all the classes of interest are shown at the bottom for reference. The names of the object are obtained by speech recognition and are shown right after the drawing of the scribbles. Annotators can correct the result if it is wrong.*

- Re-ranking the n-best words by integrating information from visual object recognition, as analyzing the visual content in the drawing area provides complementary information. We choose to learn the recognizer in a webly-supervised manner, where images retrieved from image search engines such as Google and Bing are used directly for the training. This integration is also interesting in the wider context of combining vision and speech. Section 3.3.2 elaborates the method.

As a fall-back solution, Draw&Tell provides ways to permit correction of the recognition errors, by either repeating the operation (drawing + speaking) for the same object; or typing the object name directly. The annotator will be asked to review the name list if the system fails multiple times to recognize the speech, mainly because the spoken words are

out of the dictionary. Our implementation is built on top of the open source code of [Maas et al., 2015]. The interface of Draw&Tell is shown in Figure 3.3.

3.3.2 Integration with Webly Supervised Object Recognition

As discussed in the previous section, speech recognition still has problem for high-precision applications like ours. Thus, we propose to re-rank the n-best words (object names) from the speech recognition engine by using visual information. The idea is straightforward as follows: once a scribble is drawn, a corresponding image region is cropped and fed into an object recognition system for visual recognition. The region is generated by choosing an object proposal out of 500 candidates yielded by edge-box [Zitnick & Doll, 2014]: (i) obtaining the enclosing bounding-box of the scribble, and (ii) choosing an object proposal with which the bounding-box has the highest intersection-over-union score. The recognition scores are then fused with the scores from speech recognition to obtain the name of the object.

The object recognition system can be trained with samples collected by conventional annotation methods, if available. We here explore the potential of training it with web images, due to (i) an enormous amount of visual data is online to use for free; and (ii) noisy recognition (up to some level) is acceptable here as it is only used to complement speech recognition, which itself already performs well for a small dictionary of pre-defined words, though not perfect.

We follow the footprint of [Chen & Gupta, 2015b] to train a webly-supervised RCNN [Girshick et al., 2014a] model for object recognition. Our RCNN is trained with a classification loss on images returned by Google directly – images returned by the same keyword are taken as of the same class. The rationale is that the top images returned by Google mostly come with a clean background and a single centered object of the query class. If needed, more sophisticated techniques can be used to prepare the training data, such as seeds-based co-segmentation [Chen & Gupta, 2015b]. In this work, we downloaded 2500 images for each class, and fine-tuned the VGG Net [Simonyan & Zisserman, 2015] pre-trained on ImageNet [Deng et al., 2009]. The integration of speech recognition and webly-supervised object recognition is done by simply fusing their posterior probabilities:

$$P(y|\mathbf{v},\mathbf{x}) = \sigma P(y|\mathbf{v}) + (1-\sigma)P(y|\mathbf{x}) \tag{3.1}$$

where $y \in \{1, ..., K\}$ denotes the object names with K the number of classes, $\mathbf{v}$ the voice represented as spectrograms, $\mathbf{x}$ the cropped image region, and σ a scalar value to balance the two terms. σ is set to 0.7 empirically in this work to put more weight on speech recognition as the object recognizer is trained with noisy data.

PASCAL VOC				MSCOCO			
PocketSphnix	Ours			PocketSphnix	Ours		
	Speech	Web	Combined		Speech	Web	Combined
71.2	77.4	51.0	92.1	57.3	60.0	39.6	75.8

***Table 3.1**: Recognition accuracy (%) of object names, where speech means our method without the help of the object recognition, web denotes only using the object recognition, and combined stands for our final method.*

3.3.3 Annotation Results

Three results are reported: (i) recognition accuracy of object names, (ii) annotation accuracy of the scribbles, and (iii) annotation speed of our method.

Recognition accuracy for object names: Our method is evaluated with a comparison to PocketSphinx [sph, a], a standard speech recognition engine. For fair comparison, we retrained a new dictionary and language model [1] for PocketSphinx as well so that it focuses on the object names of interest. The evaluation is conducted on the box annotation of PASCAL VOC 2012 [Everingham et al., 2010] and that of MSCOCO dataset [Lin et al., 2014]. An annotated object in the image is highlighted and its name is displayed. Annotators are asked to draw scribbles on the object while speak aloud the name displayed. Examples are shown in the supplementary material. The recognition result will be compared to the ground truth and the average accuracy is reported. 20 object instances are sampled for each object classes, leading to $400 = 20 \times 20$ objects for PASCAL VOC and $1760 = 20 \times 88$ objects for MSCOCO. Three annotators (graduate students) are asked to perform the annotation, and their results are averaged. Table 3.1 shows the results.

The results shows that our method performs significantly better than the pure speech recognition methods, namely the HMM-GMM based method PocketSphnix, and the bidirectional RNN based method [Graves & Jaitly, 2014; Maas et al., 2015]. Webly-supervised object recognition provides reasonably good results, but itself alone is still not enough to be used for the annotation task. By combining the strength of vision and speech, our method yields excellent results for the 20 classes of PASCAL VOC, and quite good results for the 88 classes of MSCOCO. The results suggest that our method is accurate enough to be used directly for annotation tasks of around 20 classes. For more classes like that in MSCOCO, people often annotate them in a hierarchical manner [Lin et al., 2014], leading to annotation tasks with smaller number of classes, which makes our method applicable as well. In the rest of this work, we will mainly evaluate our annotation method on the 20 classes of PASCAL VOC.

Annotation Accuracy: We report and analyze our annotation results on PASCAL VOC 2012 [Everingham et al., 2010] here. We annotated the 20 object classes of PASCAL

[1] They are trained by simply uploading the text corpus to CMUSphinx's web service [sph, b].

Anno-1.5				Anno-3			
object		pixel		object		pixel	
false neg.	false pos.	background	other obj.	false neg.	false pos.	background	other obj.
4.7	3.8	1.7	5.1	4.5	4.0	1.0	3.8

***Table 3.2**: Errors (%) of the annotation: false positives and false negatives at object-level, and the percentages of pixels drawing to the background and objects of other classes for the scribbles which are 'correct' at object-level.*

VOC for $12,031$ images, including all images in the training and validation set of the PASCAL VOC 2012 segmentation benchmark and all images in the augmented dataset from [Hariharan et al., 2011]. The scribbles for the background will be automatically sampled in order to reduce the annotation cost. Two annotators annotated the data, independently, with the same goal of drawing scribbles as much as possible on the objects, but on different budgets: the first one has 1.5 seconds per object, while the second one has 3 seconds. They will be referred as Anno-1.5 and Anno-3. Exemplar annotations are shown in the supplementary material.

The annotation results are listed in Table 3.2. Four types of errors are reported: two at object-level and two at pixel-level. At object-level, we assign the scribbles to their closest object masks (by the number of pixels shared) and check the consistency of their labels. False positives (FP) and false negatives (FN) are reported, which are fairly few. Also, we must be aware that objects of very small size may be confusing to annotators in terms of whether they need to be annotated. These cases contributed the most of these FP and FN cases. Scribbles that are 'correct' at object-level were evaluated for their accuracy at pixel-level. We specify two errors: the percentage of pixels drawn onto background as well as onto other objects. They are actually very small, which means for the correctly detected objects human annotators can spot their position and outline very accurately. Overall, the annotations are precise, because (i) scribbles are flexible, easily adaptable to different object layouts and (ii) drawing scribbles is very natural to human.

Annotation Speed: We compare the speed of our annotation method to four other popular annotation methods (c.f. Figure 3.1): full segmentation masks, bounding boxes, singular points on objects, and image-level keywords. Because all these methods need to loop over all classes to solve the *what* problem, the minimum time - browsing time - is constant for all. According to [Bearman et al., 2015], the time for browsing one image for one class is 1 second, which is consistent with what we found in experiments. The total browsing time per image is 20.0 seconds, which is also the time for image-level annotation. The time for other forms of annotations is the sum of this 20.0 seconds and the time for annotating 2.8 (on average) objects in each image. Point-based annotation [Bearman et al., 2015] costs 3.2 seconds for clicking points on the objects, resulting in 23.2 seconds in total.

For the time of drawing one bounding box, different numbers are reported, varying from 7 to 25.5 seconds [Russakovsky et al., 2015b; Dutt Jain & Grauman, 2013], probably

Mask	Bounding-Box	Point	Keyword	Ours (Speech-based Scribble)	
				Anno-1.5	Anno-3
104.8	39.9	23.2	20.0	6.6	11.1

***Table 3.3**: The annotation speed (seconds per image) of all methods, measured on PASCAL VOC.*

because the types of objects being handled and the quality of the annotation are different. We experimented with images from PASCAL VOC 2012 and drew bounding boxes for 200 randomly chosen objects. The annotation time per bounding box is 7.1 seconds on average, which is quite efficient compared to the numbers reported in the literature. Thus the total time for bounding-box based annotation per image is 39.9 $(20.0 + 2.8 \cdot 7.1)$ seconds. Similarly, we annotated the masks for 200 randomly sampled objects using the annotation tool developed in [Bell et al., 2013]. Annotating each mask takes 30.3 seconds on average. Thus, full-mask based annotation time is 104.8 $(20.0 + 2.8 \cdot 30.3)$ seconds.

For our annotations, we allocate 2.0 seconds for browsing the image, which was found sufficient in the annotation. Drawing one scribble takes 1.5 and 3.0 seconds for Anno-1.5 and Anno-3. The recognition error rate of speech recognition is about 8% for our task, which contributes to the correction time. Putting all together, the annotation time is 6.6 $(2 + 2.8 \cdot 1.5 \cdot (1 + 0.08 + 0.08 \cdot 0.08 + ...))$ seconds per image for Anno-1.5, and 11.1 seconds for Anno-3. All the numbers are listed in Table 3.3. According to the numbers, our annotation is the fastest one (faster than the image-level annotation as well), due to the help by the integration of speech recognition and webly-supervised object recognition, which lets annotators solve the *what* and *where* tasks of object annotation both at the same time. Some of these numbers are quite *rough* estimations, but they reflect the cost to a reasonable level of accuracy.

3.4 Semantic Image Segmentation

We follow recent methods [Girshick et al., 2014a; Long et al., 2015; Zheng et al., 2015] to fine-tune CNNs for semantic image segmentation. To this end, we convert the annotated semantic scribbles to semantic heatmaps – confidence maps of corresponding classes – and then adapt the fully connected CNNs (FCN) [Long et al., 2015] to accommodate the heatmap-based training data.

3.4.1 Scribbles to Heatmaps

We convert the annotated scribbles to semantic heatmaps of all classes considered. Let $I \in \mathbb{R}^{W \times H \times Z}$ be the input image, with size $[W, H]$ and depth Z (3 for RGB images), its semantic heatmaps are denoted by $P^k \in [0, 1]^{W \times H}$, where $k \in \{1, 2, ..., K\}$, and K is

(a) original scribbles (b) augmented scribbles (c) generated semantic heatmaps

***Figure 3.4**: An example of scribble augmentation and heatmap generation: (a) annotated scribbles; (b) augmented scribbles for the background; (c) generated semantic heatmaps for the two objects.*

the number of classes considered. For classes not present in the images, their heatmaps are simply set to zero. For classes which are present, we generate the heatmap for each class individually. The heatmap is generated by a combination of interactive image segmentation (IIS) and ensemble learning. In particular, we take the scribbles of the class considered as the scribbles for the foreground object in the context of IIS, and the scribbles on other objects as that for the background. For instance if the dog class in Figure 3.4(a) is considered, the scribble on the person is taken as the scribble for background. However, as the example shows, only having the one scribble for the entire background is wanting. We want to add scribbles sampled from the rest of the background as well, thus forming an augmented scribble set. See Figure 3.4(b) for the three additional scribbles. With all scribbles in the augmented set, we run the IIS method proposed in [Gulshan et al., 2010] to get one solution to the heapmap P^k (Figure 3.4(d)), with 1 indicating foreground and 0 background. To further increase the robustness, we run the method T times with different augmented scribbles to obtain T such solutions. The final heatmap (shown in Figure 3.4(c)) for class k is computed as the average of all the individual solutions:

$$P^k = 1/T \sum_{t=1}^{T} P_t^k. \tag{3.2}$$

This is inspired by ensemble learning. Below we present the scribble augmentation.

Scribble Augmentation

For each class, three (sufficient in practice) more scribbles are added. For simplicity, they are added sequentially. New scribbles should be far from the foreground for good separation, and far from existing background scribbles to be complementary. Below, we define the distance between the scribbles.

Given an image I, all paths that connect pixel a and pixel b are denoted by $\mathcal{Q}_{ab}$. Suppose we have a path Γ described by the pixels it passes through $\{\Gamma^1, \Gamma^2, ..., \Gamma^r\}$. The distance between the pixels Γ^1 and Γ^r along path Γ is defined as (following [Gulshan et al., 2010]):

$$D(\Gamma) = \sum_{j=1}^{r} \sqrt{d_{eu}(\Gamma^j, \Gamma^{j+1}) + \lambda \|\nabla I(\Gamma^j)\|^2} \ , \tag{3.3}$$

where $d_{eu}(\Gamma^j, \Gamma^{j+1})$ is the Euclidean distance between two consecutive pixels, and $\|\nabla I(\Gamma^j)\|^2$ is the gradient magnitude between (Γ^j, Γ^{j+1}), which is computed by the edge detector of the structured random forest [Dollár & Zitnick, 2013], to avoid texture edges. λ is used to balance the two terms.

Let's denote the existing scribbles as a set of pixels $\mathcal{E}$. Then, the geodesic distance of pixel a to $\mathcal{E}$ is defined as:

$$G(a) = \min_{b \in \mathcal{E}} \min_{\Gamma \in \mathcal{Q}_{ab}} D(\Gamma) \tag{3.4}$$

Without any specific preference, we fix the shape of the new scribble $\bar{\mathcal{E}}$ to a disk of radius $r = 20$ (a balance between localization and informativeness). Then, the geodesic distance from the scribble centered at position a to existing scribbles $\mathcal{E}$ is:

$$G(\bar{\mathcal{E}}_a) = \min_{c \in \bar{\mathcal{E}}_a} G(c). \tag{3.5}$$

The sampling probability for the next scribble is then defined as:

$$Pr(a) = \frac{\exp(G(\bar{\mathcal{E}}_a)^2/\sigma^2)}{\sum_{b=1}^{WH} \exp(G(\bar{\mathcal{E}}_b)^2/\sigma^2)} \tag{3.6}$$

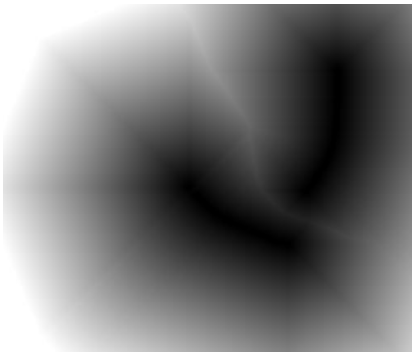
where σ is set adaptively to the average of all $G(\bar{\mathcal{E}}_a)$. $Pr(a)$ needs to be updated each time a new scribble is added. One example of $Pr(a)$ for all values of a is shown in Figure 3.5. The randomness introduced by this sampling allows for the ensemble approach. Note that other information such as objectness [Zitnick & Doll, 2014] can be exploited for sampling the background scribbles. We leave this as future work.

3.4.2 Heatmap-based FCN

FCN [Long et al., 2015] is designed to regress the 2D label map $O \in \{1, 2, ..., K\}^{W \times H}$, directly from the input image I. The output of FCN is K score maps S, one for one class, i.e. $S \in \mathbb{R}^{W \times H \times K}$. The class labels are obtained either by simply taking the best-scoring classes at each pixel [Long et al., 2015] or by using conditional random fields for further refinement [Papandreou et al., 2015]. FCN is trained to minimize the prediction error of cross-entropy over all pixels, and its optimization function is:

$$\mathcal{L} = \sum_{n=1}^{N} \sum_{a=1}^{WH} \mathcal{L}_{na}(S_{na}, O_{na}), \tag{3.7}$$

(a) existing scribbles



(b) probability map

***Figure 3.5**: Probability map for sampling the next new scribble (c.f. Equation 3.6).*

where N is the number of training images and $\mathcal{L}_{na}$ is the per-pixel loss function, which is commonly defined as:

$$\mathcal{L}_{na} = -\log\left(\frac{\exp(S_{na}^{O_{na}})}{\sum_{k=1}^{K}\exp(S_{na}^{k})}\right). \tag{3.8}$$

The loss function is only computed for pixels having ground truth labels. Otherwise, the loss function is set to zero.

Directly applying FCN to our training annotations is problematic, as our annotations are soft semantic heatmaps rather than crisp segmentation masks. To solve this, we extend the loss function in Equation 3.8 to the following:

$$\mathcal{L}'_{na} = \sum_{k=1}^{K} P_{na}^{k}\left[-\log(\frac{\exp(S_{na}^{k})}{\sum_{k'=1}^{K}\exp(S_{na}^{k'})})\right]. \tag{3.9}$$

By the adaptation, the loss function is modulated by the heatmaps of relevant classes – for each pixel, the most confident classes affect the loss function the most. It can be seen that the loss function on crisp segmentation masks in Equation 3.8 is a special case of our new loss function. The new loss function can be optimized the same way as used in standard FCN, with a modification to the loss layer. We call the model heatmap-based FCN (HFCN), and its pipeline is shown in Figure 3.2.

3.5 Experiments

We evaluate HFCN with different settings, and compare it to other competing methods. The goal is to show that training with scribbles-based annotations is a good trade-off between annotation cost and prediction accuracy.

Supervision	Image-level	Point	Bounding-box	Full-mask	Scribbles (ours)		
Method	ConsCNN	WhatPoint	CNN-EM	FCN-8s	HFCN-1.5	HFCN-3	HFCN-3+CRF
mIoU	35.3	42.7	60.6	62.7	56.2	61.9	64.1

***Table 3.4**: The results of different methods with varying levels of supervision on the validation set of PASCAL VOC 2012.*

3.5.1 Experimental Settings

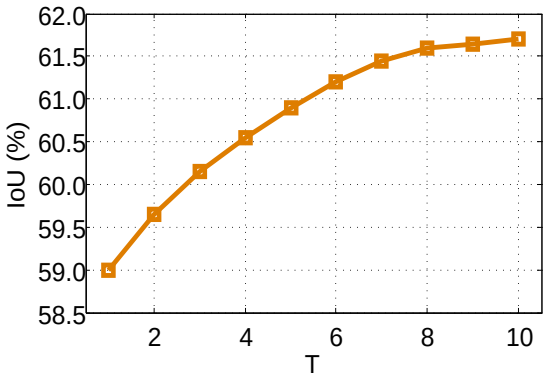
Dataset: We evaluate HFCN on PASCAL VOC 2012 [Everingham et al., 2010], which comes with three subsets: training (1464 images), validation (1449 images) and test (1456 images), having 20 object classes annotated in full segmentation masks. In order to keep the same settings with previous methods [Long et al., 2015; Pathak et al., 2015a; Papandreou et al., 2015; Bearman et al., 2015], we also extend the training set by adding the extra annotations created by Hariharan *et al.* [Hariharan et al., 2011] (excluding images from the original validation set), ending up with $10,582$ training images. The method is evaluated on the validation set and the test set, under the metric intersection-over-union (IoU) for all the 20 classes.

CNN: We adopt the FCN-8s [Long et al., 2015] model, as it has shown excellent performance and there is code available. The FCN model is adapted from the VGG network [Simonyan & Zisserman, 2015] pre-trained on the ILSVRC dataset [Russakovsky et al., 2015a]. For the optimization, we employ a procedure similar to FCN: the SGD solver is used, with an initial learning rate of 10^{-6}; the network is trained with $80,000$ iterations. HFCN is implemented with the Caffe framework [Jia et al., 2014].

3.5.2 Results

Scribble Augmentation. The scribble for the background is augmented in an ensemble sampling manner, which saves the annotation cost for the background; background is often large and scattered, so annotating it can be costly. The parameter T for the ensemble sampling is evaluated over 10 values: $[1, 2, ..., 10]$ for HFCN-3. The results is shown in Fig 3.6. As the figure shows, the performance increases with T from the beginning and then starts stabilizing. The scribble augmentation from each single round has its own weakness and the combination of multiple rounds 'cancels out' the weaknesses because the weaknesses from different rounds are different due to the randomness in the sampling. This property has been widely explored in the filed of ensemble learning. We use $T = 10$ for all the following experiments.

Validation Result: We first evaluate the method on the validation set. Table 3.4 show the results, where the results of several other methods are also reported for



Method	aerop	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mIoU
Image-level:																					
MIL-FCN	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	24.9
Img2Pix-CNN	25.4	18.2	22.7	21.5	28.6	39.5	44.7	46.6	11.9	40.4	11.8	45.6	40.1	35.5	35.2	20.8	41.7	17.0	34.7	30.4	32.6
Single-Point:																					
What's-point	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	43.6
Bounding-Box:																					
CNN-EM	64.4	27.3	67.6	55.1	64.0	81.6	70.5	76.0	24.1	63.8	58.2	72.1	59.8	73.5	71.4	47.4	76.0	44.2	68.9	50.9	60.8
BoxSup	80.3	31.3	82.1	47.4	62.6	75.4	75.0	74.5	24.5	68.3	56.4	73.7	69.4	72.5	75.1	47.4	70.8	45.7	71.1	58.8	64.6
Full-mask:																					
SDS	63.3	25.7	63.0	39.8	59.2	70.9	61.4	54.9	16.8	45.0	48.2	50.5	51.0	57.7	63.3	31.8	58.7	31.2	55.7	48.5	51.6
FCN-8s	76.8	34.2	68.9	49.4	60.3	75.3	74.7	77.6	21.4	62.5	46.8	71.8	63.9	76.5	73.9	45.2	72.4	37.4	70.9	55.1	62.2
Zoomout	81.9	35.1	78.2	57.4	56.5	80.5	74.0	79.8	22.4	69.6	53.7	74.0	76.0	76.6	68.8	44.3	70.2	40.2	68.9	55.3	64.4
DeepLab-CRF	83.5	36.6	82.5	62.3	66.5	85.4	78.5	83.7	30.4	72.9	60.4	78.5	75.5	82.1	79.7	58.2	82.0	48.8	73.7	63.3	**70.7**
Speech-Scribbles:																					
HFCN-1.5	72.1	32.1	63.2	47.0	59.7	72.1	70.4	72.8	21.6	58.2	41.6	68.2	58.2	71.3	71.5	41.4	56.8	32.1	64.5	52.1	56.4
HFCN-3	76.1	37.4	69.3	53.5	64.9	79.6	74.4	76.4	25.9	62.5	45.8	72.3	62.4	76.8	74.6	45.7	72.1	38.3	68.9	56.2	61.7
HFCN-3+CRF	78.7	37.5	71.7	52.8	64.5	78.6	77.8	79.7	25.9	65.6	49.9	75.1	65.9	79.4	76.6	48.5	74.9	39.9	73.9	59.5	63.9

***Table 3.5**: Comparison to other methods with different levels of supervision on PASCAL VOC 2012 test.*

comparison. For HFCN, we trained it with the two versions of annotations: scribbles from Anno-1.5 and Anno-3. They are referred hereafter by HFCN-1.5 and HFCN-3. Following the literature [Chen et al., 2015; Pathak et al., 2015a], we also added the refinement by CRF [Krähenbühl & Koltun, 2011] to Anno-3, which is referred by HFCN-3+CRF. It generally true from the table that stronger (more expensive) supervision leads to better performance. However, it also shows that with the scribbles by Anno-1.5, HFCN is already able to yield quite decent results. If the training data is upgraded to the scribbles of Anno-3, the results are comparable to that of FCN-8s [Long et al., 2015] and better than that of CNN-EM [Papandreou et al., 2015], though their training annotations are much more expensive to obtain (*c.f.* Table 3.3). Our method also shows significantly better results than the methods [Bearman et al., 2015; Pathak et al., 2015a] trained with comparable annotation cost. HFCN-3+CRF further improves the performance on top of HFCN-3, showing the benefits of combining graphical model and deep neural networks. Figure 3.7 shows several segmentation examples.

Test Results: We also evaluate HFCN on the test set. Table 3.5 shows all the results. The conclusions drawn on the validation set hold on the test set as well. Our method obtains results which are comparable to other methods trained with more expensive supervision, *i.e.* full masks and bounding boxes, and are better than methods trained with supervision

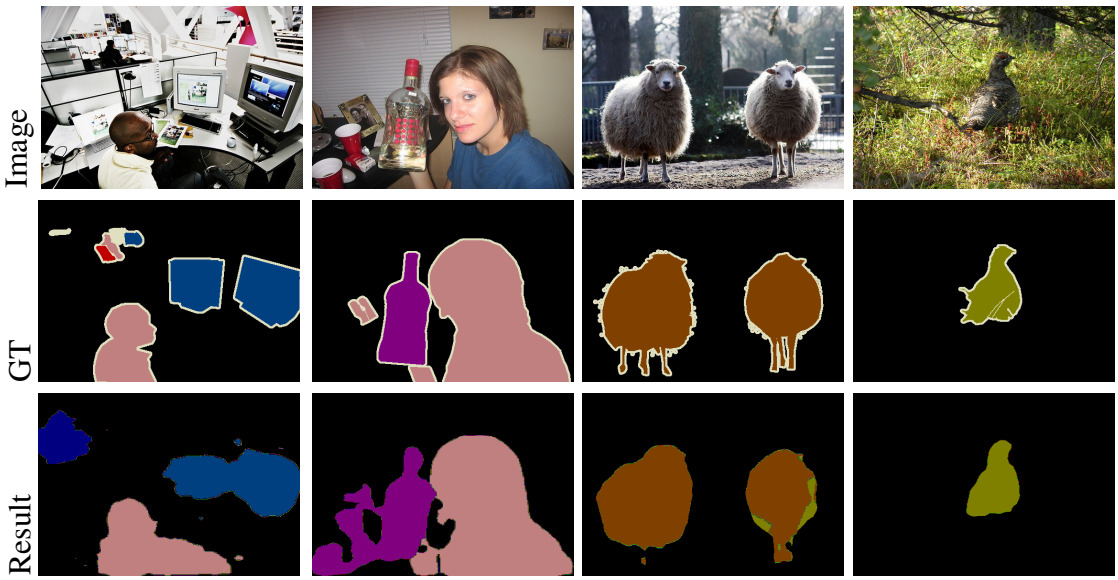


***Figure 3.7**: Examples of the segmentation results on the validation set of PASCAL VOC 2012.*

of comparable annotation cost. These annotation costs are only computed on the PASCAL VOC training images, however, some competing methods also use 'extra' supervision. For instance, Img2Pix-CNN [Pinheiro & Collobert, 2015] learns image prior from another larger set of images; BoxSup [Dai et al., 2015b] uses the MCG object proposal [Arbelaez et al., 2014] method, which is trained with full-mask supervision from the PASCAL VOC dataset. Also, the bounding boxes used by the two methods [Papandreou et al., 2015; Dai et al., 2015b] are generated from the labeled full segmentation masks, which may transfer some supervision from there because the generated bounding boxes are often tighter than those annotated directly by annotators. The performance of HFCN is improved the same way as other methods [Chen et al., 2015; Papandreou et al., 2015; Pathak et al., 2015a] by the CRF [Krähenbühl & Koltun, 2011] for further refinement.

Annotation Cost vs. Accuracy: We tabulate the annotation cost and the mIoU of all the methods considered. See Figure 3.8 for the results. From the plot, it is evident that HFCN strikes a good balance between annotation speed and prediction accuracy. The merit makes our method adaptable and applicable to new, customized tasks, which is beneficial to many real vision applications. As more and more forms of supervision are explored, we believe that an evaluation of annotation cost vs. accuracy is very necessary in order to clearly show the trade-offs made by different alternative approaches.

Weak Annotation vs. Strong Annotation. The development of methods with different forms of annotations naturally raises a question: under a fixed annotation budget, should one work for more weak annotations or fewer precise annotations? We answer this question by comparing the performance of HFCN-3 trained on 9900 images annotated by Anno-3 to that of FCN-8 trained on 900 images with full-mask annotations. The

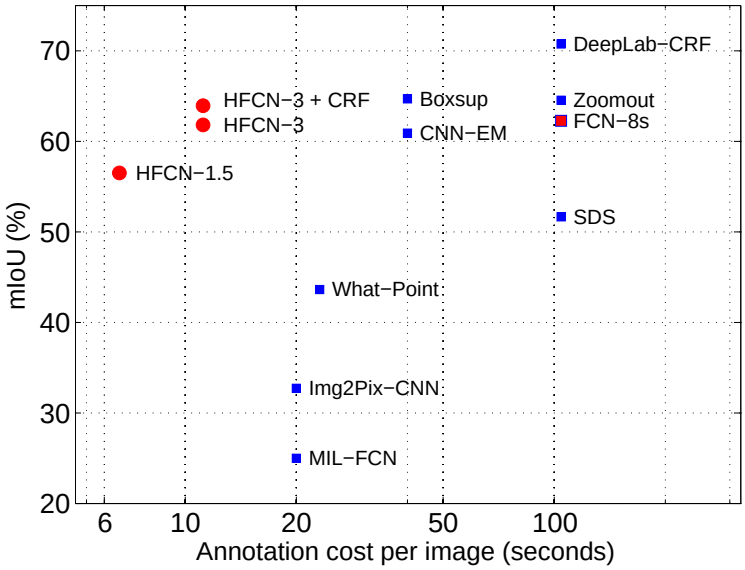


***Figure 3.8**: Annotation vs. segmentation performance (mIoU) of all the methods considered. Evaluated on PASCAL VOC 2012 test.*

training data for FCN-8 is randomly sampled from all the training data. 5 FCN-8 models are trained, each with its own training set, and their results are averaged. The final results (mIoU) are: 56.3% for FCN-8 and 60.9% for HFCN-3. The results are exciting, and they suggest that with the same amount of annotation effort, our method gathers more semantic information than the conventional more expensive annotation methods. In a wider context, the results suggest that gathering weak training data can be more helpful than gathering strong training data, if the same amount of annotation effort is given. Learning with a mixture of strong and weak annotations can be interesting as well, as shown in [Dai et al., 2015b]. We leave this as our future work.

Soft Heatmap vs. Hard Segmentation. We investigated whether the soft heatmaps are helpful. To this aim, a baseline is designed for comparison by simply taking the max object probability from the soft maps, and then passing the resulting object masks to FCN-8. We found that by doing so, the performance drops from 61.7 to 59.2. The advantage of soft heatmaps is that objects (parts) with uncertainties are left to be decided and explored by the FCN model that has been trained with objects of higher certainty. The trained FCN model is more knowledgeable than simple thresholding in distinguishing uncertain objects.

3.6 Conclusion

In this work , we developed an annotation method Draw&Tell to create training data for semantic image segmentation. Draw&Tell allows annotators to simply draw scribbles (strokes) on objects and speak their names in the meanwhile, solving the *what* and *where* problems once at the same time. We have proven experimentally that Draw&Tell is faster than other annotation methods, *e.g.* 11 times faster than full-mask annotation, 4 times

faster than bounding-box annotations, and 2 times than the image-level annotation. A method of integrating visual information and acoustic information is also proposed for robust object name recognition. This combination can serve as an example of integrating vision and speech, and inspires research in this direction. Furthermore, we proposed a method that allows CNNs models to learn from scribble-based training data, by converting scribbles to semantic confidence maps and extending standard CNNs to accommodate soft confidence maps. We showed in experiments that our annotation method, coupled with the learning method, yields significantly better results than competing methods trained with annotations of comparable cost, and yields comparable results to the methods trained with significantly more expensive annotations. Introducing speech recognition to visual annotation is helpful especially for tasks on mobile devices. This work is just a very first step in this direction.

4

Representation Learning with Unlabeled Data

Providing efficient solutions to image classification has always been a major focus in computer vision. Recent years have witnessed considerable progress in image classification. However, most popular systems [Lazebnik et al., 2006; H. Zhang & Malik, 2006; Bosch et al., 2007; Boiman et al., 2008; Quattoni & Torralba, 2009; Wang et al., 2010; Xiao et al., 2010; Yang et al., 2014] heavily rely on manually labeled training data, which is expensive and sometimes impossible to acquire. Despite substantial efforts towards 'pleasant' annotation by developing online games [Von Ahn, 2006] or appealing software tools [Russell et al., 2008b], collecting training data for recognition is still very time-consuming and tedious. The scarcity of annotations, combined with the explosion of image data, starts shifting focus towards learning with less supervision. As a result, numerous techniques have been developed to learn classification models with cheaper annotations. The most notable ones include semi-supervised learning [Fergus et al., 2009; Guillaumin et al., 2010; Dai & Van Gool, 2013], active learning [Jain & Kapoor, 2009; Joshi et al., 2009b], transfer learning [Quattoni et al., 2008; Pan & Yang, 2010], weakly-supervised learning [Prest et al., 2012; Dai et al., 2015a], self-taught learning [Raina et al., 2007], and image clustering [Sivic et al., 2005; Dai et al., 2010].

In this paper, we are interested in the problem of semi-supervised learning (SSL) for image classification, but our solution can be applied seamlessly to image clustering, as shown in Section 4.4. Given a small set of labeled data along with a large set of unlabeled data, SSL aims to learn a more accurate recognition system than that based on the labeled data set alone. Recent research in SSL has been quite successful [Fergus et al., 2009; Leistner et al., 2009; Kumar Mallapragada et al., 2009; Zhu & Goldberg, 2009; Chapelle et al., 2006; Ebert et al., 2010; Pitelis et al., 2014; Kingma et al., 2014], typically building upon the *local-consistency* assumption that data samples with high similarity should share the same label. This is also called *smoothness of manifold*, and it is often used to regularize the classifying functions learned from labeled samples or to propagate labels of labeled samples to unlabeled ones over a graph reflecting the manifold structure. Yet,

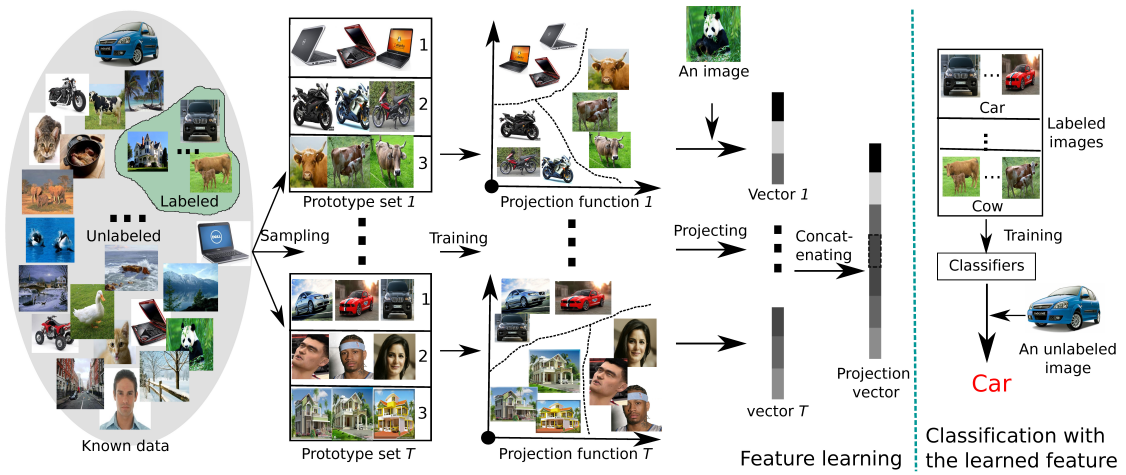


***Figure 4.1**: The pipeline of Ensemble Projection (EP). EP consists of unsupervised feature learning (left panel) and plain classification or clustering (right panel). For feature learning, EP samples an ensemble of T diverse prototype sets from all known images and learns discriminative classifiers on them for the projection functions. Images are then projected using these functions to obtain their new representation. These features are fed into standard classifiers and clustering methods for image classification and clustering respectively.*

these methods have their drawbacks: regularization-based methods [Joachims, 1999; Li & Zhou, 2011] are difficult to apply to large-scale problems; graph-based methods [Zhu et al., 2003; Zhou et al., 2004; Fergus et al., 2009; Liu et al., 2010] are difficult to be given out-of-sample extensions and lack ability for discriminative learning.

In this paper, we depart from the traditional paradigm and propose another route to SSL. Instead of regularizing the classifiers like most of the SSL methods [Bennett & Demiriz, 1998; Joachims, 1999; Kumar Mallapragada et al., 2009; Leistner et al., 2009], we learn a new feature representation using the labeled and unlabeled data. The new feature representation is learned in a discriminative way to capture not only the information of individual images, but also the relationships among images. The learning is conceptually straightforward and computationally simple. The learned features can be fed into any classifiers for the final classification of the unlabeled samples. Thus, the method is agnostic to the classifiers. This facilitates the deployment of SSL methods, as users often have their favorite classifiers and are reluctant to drop them. For image clustering, we apply the same feature learning methods to all provided images, and then feed the learned features to standard clustering methods such as k-means and Spectral Clustering. Below, we present our motivations and outline the method.

4.0.1 Motivations

People learn and abstract the concepts of object classes well from their intrinsic characteristics, such as colors, textures, and shapes. For instance, *sky* is blue, and a *football* is spherical. We also do so by comparing new object classes to those that have already been learned. For example, a *leopard* is similar in appearance to a *jaguar*, but is smaller. This paradigm of learning-by-comparison or characterization-by-comparison is part of Eleanor Rosch's prototype theory [Rosch, 1978], that states that an object's class is determined by its *similarity* to prototypes which represent object classes. The theory has been used successfully in transfer learning [Quattoni et al., 2008], when labeled data of different classes are available. An important question is whether the theory can also be used for SSL and image clustering when a large amount of unlabeled data is available. This paper investigates this problem.

To use this paradigm, we first need to create the prototypes automatically from the available data, both labeled and unlabeled. In keeping with Eleanor Rosch's prototype theory [Rosch, 1978], ideal prototypes should have two properties: 1) images in the same prototype are to be from the same class; and 2) images of different prototypes are to be from different classes. They guarantee meaningful comparisons and reduce ambiguity. Without access to labels of data samples, the prototypes have to be created in an unsupervised way, based on some assumptions. In addition to the widely-used *local-consistency*, we propose another one called *exotic-inconsistency*, which states that samples that are far apart in the feature space are very likely to come from different classes. The assumptions have been verified experimentally, and will be presented in Sec. 4.2.1. Based on these two assumptions, it stands to reason that samples along with their closest neighbors can be "good" prototypes, and such prototypes that are far apart can play the role of different classes. According to this observation, we design a method to sample the prototype set from all available images by encoding them on a graph with links reflecting their affinity.

The sampled prototypes are taken as surrogate classes and discriminative learning is yields projection functions tuned to the classes. Images are then linked to the prototypes via their projection values (classification scores) by the functions. Since information carried by a single prototype set is limited and can be noisy, we borrow ideas from ensemble learning [Rokach, 2010] to create an ensemble of diverse prototype sets, which in turn leads to an ensemble of projection functions, to mitigate the influence of the deficiencies of each training set. The idea is that if the deficiency modes of the individual training sets are different or 'orthogonal', ensemble learning is able to cancel out or at least mitigate their effect. This conjecture is verified with a simulated experiment in Sec. 4.2.2, and is also supported by the superior performance of our method in real applications. With the ensemble of classifiers, images are then represented by the concatenation of their classification scores – similarities to all the sampled image prototypes – for the final classification, which is in keeping with prototype theory [Rosch, 1978]. We call the method Ensemble Projection (EP). Its schematic diagram is sketched in Fig. 4.1.

4.0.2 Contributions

EP was evaluated on eight image classification datasets, ranging from texture classification, over object classification and scene classification, to style classification. For SSL, EP is compared to three baselines and three other methods. For image clustering, it is compared to the Convolutional Neural Network (CNN) feature of [Chatfield et al., 2014] with two standard clustering methods: k-means and Spectral Clustering. The experiments show that: (1) EP improves over the original features by learning with unlabeled data when standard classifiers are used, and outperforms competing SSL methods; (2) EP produces promising results for self-taught image classification where the unlabeled data does not follow the same distribution as the labeled ones, but rather from a random collection of images; (3) EP improves over the original features for image clustering.

This paper is an extension of our ICCV paper [Dai & Van Gool, 2013] and our ECCV paper [Dai et al., 2012b]. In addition to putting the two papers into the same framework, this paper has two new contributions. First of all, in the conference papers, the idea of EP was validated only with very general and 'cheap' features, such as LBP [Ojala et al., 2002], GIST [Oliva & Torralba, 2001], and PHOG [Bosch et al., 2007]. These features, however, are obsolete and do not always yield satisfactory results on classification datasets. Recently, features learned by CNN has resulted in state-of-the-art performance in various recognition tasks [Krizhevsky et al., 2012a; Donahue et al., 2014b; Girshick et al., 2014b; Chatfield et al., 2014]. In this paper, we validate the efficacy of EP with CNN features rather than the simple features used in our earlier work. The second difference is that in this paper experiments are conducted on eight standard classification datasets, instead of only four in [Dai & Van Gool, 2013] and three in [Dai et al., 2012b]. While we focus on image classification and clustering, our framework of feature learning from unlabeled data can be used for other tasks as well. For instance, Tang et al. [2013] extended the idea to generate hashing functions for efficient image retrieval.

The rest of this paper is organized as follows. Section 4.1 reports on related work. Section 4.2 is devoted to two motivating observations. Section 4.3 describes our method, followed by experiments in Sec.4.4. Sec.4.5 concludes the paper.

4.1 Related Work

Our method is generally relevant to semi-supervised learning, ensemble learning, image feature learning, and image clustering.

Semi-supervised Learning: SSL aims at enhancing the performance of recognition systems by exploiting an additional set of unlabeled data. Due to its great practical value, SSL has a rich literature [Chapelle et al., 2006; Zhu & Goldberg, 2009]. Amongst existing methods, the simplest methodology for SSL is based on the self-training scheme

[Blum & Mitchell, 1998] where the system iterates between training recognition models with current 'labeled' training data and augmenting the training set by adding its highly confident predictions in the set of unlabeled data; the process starts from human labeled data and stops until some termination condition is reached, *e.g.* the maximum number of iterations. Guillaumin et al. [2010] and Shrivastava et al. [2012] presented two methods in this stream for image classification. While obtaining promising results, they both require additional supervision: Guillaumin et al. [2010] need image tags and Shrivastava et al. [2012] image attributes.

The second group of SSL methods is based on label propagation over a graph, where nodes represent data examples and edges reflect their similarities. The optimal labels are those that are maximally consistent with the supervised class labels and the graph structure. Well known examples include Harmonic-Function [Zhu et al., 2003], Local-Global Consistency [Zhou et al., 2004], Manifold Regularization [Belkin et al., 2006], and Eigenfunctions [Fergus et al., 2009]. While having strong theoretical support, these methods are unable to exploit the power of discriminative learning for image classification.

Another group of methods utilize the unlabeled data to regularize the classifying functions – enforcing the boundaries to pass through regions with a low density of data samples. The most notable methods are transductive SVMs [Joachims, 1999], Semi-supervised SVMs [Bennett & Demiriz, 1998], and semi-supervised random forests [Leistner et al., 2009]. These methods have difficulties to extend to large-scale applications, and developing an efficient optimization for them is still an open question. Readers are referred to [Zhu & Goldberg, 2009] for a thorough overview of SSL.

Ensemble Learning: Our method learns the representation from an ensemble of prototype sets, thus sharing ideas with ensemble learning (EL). EL builds a committee of base learners, and finds solutions by maximizing the agreement. Popular ensemble methods that have been extended to semi-supervised scenarios are Boosting [Kumar Mallapragada et al., 2009] and Random Forests [Leistner et al., 2009]. However, these methods still differ significantly from ours. They focus on the problem of improving classifiers by using unlabeled data. Our method learns new representations for images using all data available. Thus, it is independent of the classification method. The reason we use EL is to capture rich visual attributes from a series of prototype sets, and to mitigate the deficiency of the sampled prototype sets. Other work close to ours is Random Ensemble Metrics [Kozakaya et al., 2011], where images are projected to randomly subsampled training classes for supervised distance learning.

Supervised Feature Learning: Over the past years, a wide spectrum of features, from pixel-level to semantic-level, have been designed and used for different vision tasks. Due to the semantic gap, recent work extract high-level features, which go beyond single images and are probably impregnated with semantic information. Notable examples are Image Attributes [Farhadi et al., 2009], Classemes [Torresani et al., 2010], and Object Bank [Li et al., 2010]. While getting pleasing results, these methods all require additional

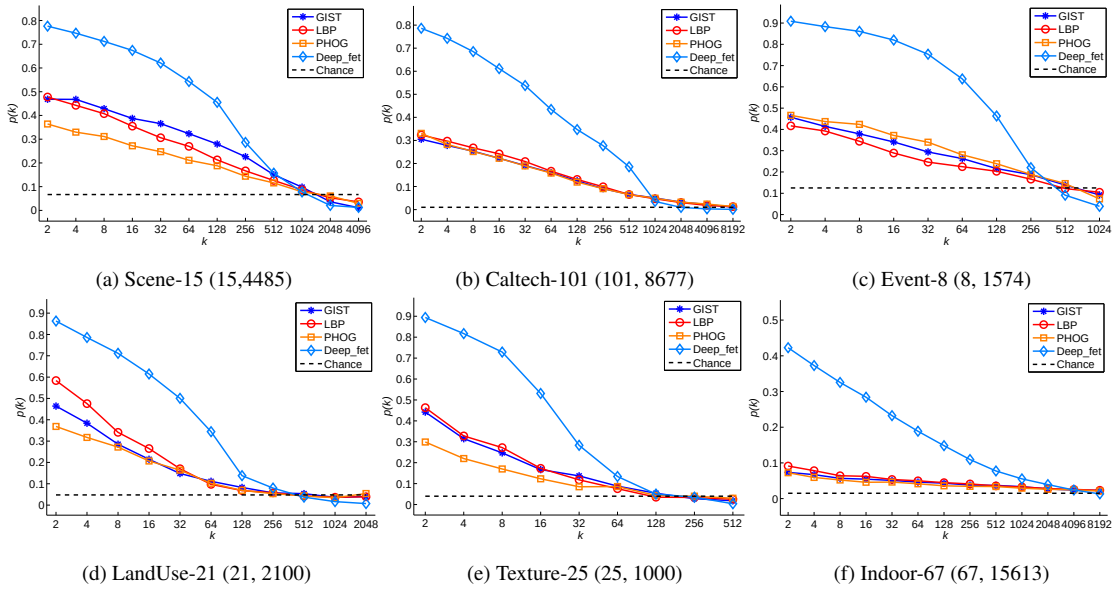
labeled training data, which is exactly what we want to avoid. There have been attempts, *e.g.* [Sharmanska et al., 2012; Yu et al., 2013], to avoid the extra attribute-level supervision, but they still require canonical class-level supervision. Our representation learning however, is fully unsupervised.

Unsupervised Feature Learning: Our method is akin to methods which learn middle- or high-level image representation in an unsupervised manner. [Coates et al., 2011] employs k-means mining filters of image patches and then applys the filters for feature computation. [Dosovitskiy et al., 2014] generates surrogate classes by augmenting each patch with its transformed versions under a set of transformations such as translation, scaling, and rotation, and trains a CNN on top of these surrogate classes to generate features. The idea is very similar to ours, but our surrogate classes are generated by augmenting seed images with their close neighbors. The learning methods are also different. [Singh et al., 2012] discovers a set of representative patches by training discriminative classifiers with small, compact patch clusters from one dataset, and testing them on another dataset to find similar patches. The found patches are then used to train new classifiers, which are applied back to the first dataset. The process iterates and terminates after rounds, resulting in a set of representative patches and their corresponding 'filters'. The idea of learning 'filters' from compact clusters shares similarities with what we do, but our clusters are images rather than patches.

Image Clustering: A plethora of methods have been developed for image clustering. Fergus et al. [2003] modeled objects as constellations of visual parts and estimated parameters using the expectation-maximization algorithm for unsupervised recognition. Sivic et al. [2005] proposed using aspect models to discover object classes from an unordered image collection. Later on, Sivic et al. [2008] used Hierarchical Latent Dirichlet Allocation to automatically discover object class hierarchies. For scene class discovery, Dai et al. [2010] proposed to combine information projection and clustering sampling. These methods assume explicit distributions for the samples. Image classes, nevertheless, are arranged in complex and widely diverging shapes, making the design of explicit models difficult. An alternative strand, which is more versatile in handling structured data, builds on similarity-based methods. Dueck & Frey [2007] applied the affinity propagation algorithm of [Frey & Dueck, 2007] for unsupervised image categorization. Grauman & Darrell [2006] developed partially matching image features to compute image similarity and used spectral methods for image clustering. The main difficulty of this strand is how to measure image similarity as the semantic level goes up. Readers are referred to [Tuytelaars et al., 2010] for a survey.

4.2 Observations

In this section, we motivate our approach and explain why it is working. We experimentally verify our assumptions: First, given a standard distance metric over images, do the



(a) Scene-15 (15,4485) (b) Caltech-101 (101, 8677) (c) Event-8 (8, 1574)

(d) LandUse-21 (21, 2100) (e) Texture-25 (25, 1000) (f) Indoor-67 (67, 15613)

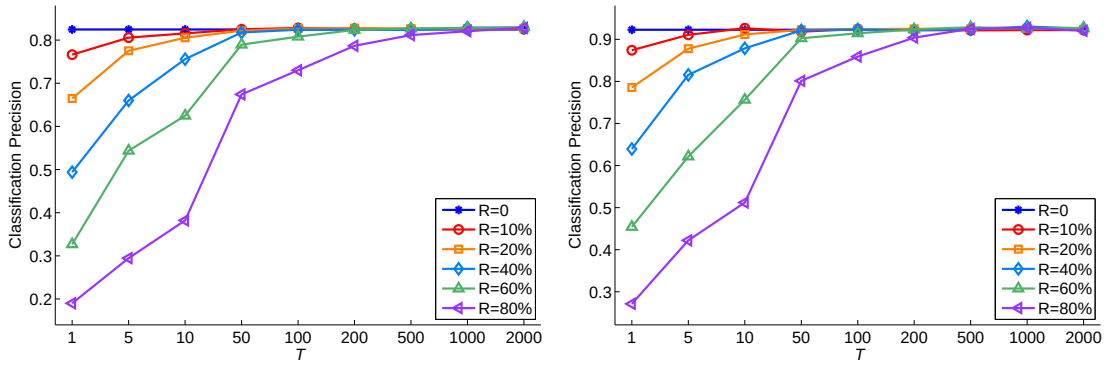
***Figure 4.2**: The label co-occurrence probability $p(k)$: frequency of images having the same label with their k_{th} neighbors. Results on six datasets are shown. The number of classes and the number of images of the datasets are shown as well.*

assumptions *local-consistency* and *exotic-inconsistency* hold, and to what extent? Second, is ensemble learning able to cancel out the deficiency of the individual training sets, given that the number of such training sets are sufficiently large and the deficiency modes of them are different or 'orthogonal'?

4.2.1 Observation 1

The assumptions of *local-consistency* and *exotic-inconsistency* do hold for real image datasets. An ideal image representation along with a distance metric should ensure that all images of the same class are more similar to each other than to those of other classes. However, this does not strictly hold for most of vision systems in reality. In this section, we want to verify whether the relaxed assumptions *local-consistency* and the *exotic-inconsistency* hold. These state images are very likely from the same class as their close neighbors, and very likely from different classes than those far from them. In order to examine the assumptions, we tabulate how often an image is from the same class as its k^{th}-nearest neighbor. We refer to the frequency as label co-occurrence probability $p(k)$. $p(k)$ is averaged across images and class labels in the dataset. Four features were tested: GIST [Oliva & Torralba, 2001], PHOG [Bosch et al., 2007], LBP [Ojala et al., 2002], and the CNN feature [Chatfield et al., 2014]. The Euclidean distance is used here.

Fig. 4.2 shows the results on six datasets (Datasets and features will be introduced in Sec.4.4). The results reveal that using the distance metric in conventional ways (*e.g.* clustering by k-means and spectral methods) will result in very noisy training sets, because



(a) Scene-15 (b) Event-8

Figure 4.3: *Classification accuracy of ensemble learning on the the Scene-15 dataset [Lazebnik et al., 2006] and the Event-8 dataset [Li & Fei-Fei, 2007], for varying training label noise R and varying number of training trials T. Experiments on other datasets obtain the same trend. Ensemble learning is able to cancell out the deficiency of the training sets even it is very severe (e.g. $R = 80\%$), given that the deficiency modes are different or 'orthogonal' and the number of training sets are sufficiently large. The figure is best viewed in color.*

the label co-occurrence probability $p(k)$ drops very quickly with k. Sampling in the very close neighborhood of a given image is likely to generate more instances of the same class, whereas sampling far-away tends to gather samples of different classes. This suggests that samples along with a few very close neighbors, namely "compact" image clusters, can form a training set for a single class, and a set of such image clusters far away from each other in feature space can serve as good prototype sets for different classes. Furthermore, sampling in this way provides the chance of creating a large number of diverse prototype sets, due to the small size of each sampled prototype set. Also, from this figure, it is evident that the CNN feature performs significantly better than the rest, which suggests that using the CNN feature in our system is recommendable.

4.2.2 Observation 2

Ensemble learning is able to cancel out or substantially mitigate the deficiency of individual training sets, given that the number of such training sets is sufficiently large and the modes of the deficiency are different or 'orthogonal'.

We examined this idea in supervised image categorization. Given the ground truth data divided into training and test sets: $\mathcal{D} = \{\mathcal{D}^{\text{train}}, \mathcal{D}^{\text{test}}\}$, (i) we artificially synthesized a set of weak training sets (training sets with different modes of deficiency) $\mathcal{D}_t^{\text{train}}, t = 1, \ldots, T$ from training data $\mathcal{D}^{\text{train}}$, and (ii) ensemble learning was then performed on these sets and its performance on test data classification was measured.

In order to guarantee the diversity of the training sets (for ensemble learning), each weak training set $\mathcal{D}_t^{\text{train}}$ is formed by randomly taking 30% of the images in $\mathcal{D}^{\text{train}}$, and randomly re-assigning labels of a fixed percentage R of these images. Hence, $R = 0$ corresponds to the upper performance bound as every sample is assigned its true label. A classifier is trained for each of these weak training sets. At test time, each of these classifiers returns the class label of each image in $\mathcal{D}^{\text{test}}$. The winning label is the mode of the results returned by all the classifiers. Fig. 4.3 evaluates this for the Scene-15 dataset [Lazebnik et al., 2006]. Logistic Regression is used as the classifiers with the CNN feature [Chatfield et al., 2014] as input. When the label noise percentage R is low, the classification precision starts out high and levels quickly with T, as one would expect. But interestingly, for R even as high as 80%, the classification precision, which starts low, converges to a similarly high precision given sufficient weak training sets T (≈ 500). This suggests that ensemble learning is able to cancel out the deficiency of individual training sets. It learns the essence of image classes when the modes of deficiency are different for different training sets, and given a sufficiently large number of such training sets.

We were inspired by the two observations, and would like to investigate whether the assumptions of *local-consistency* and *exotic-inconsistency* are enough to generate a set of such weak training sets in an unsupervised manner, with which ensemble learning is able to learn useful visual attributes for semi-supervised image classification and image clustering.

4.3 Our Approach

The training data consists of both labeled data $\mathcal{D}_l = \{(\mathbf{x}_i, y_i)\}_{i=1}^{l}$ and unlabeled data $\mathcal{D}_u = \{\mathbf{x}_i\}_{i=l+1}^{l+u}$, where $\mathbf{x}_i$ denotes the feature vector of image i, $y_i \in \{1, ..., C\}$ represents its label, and C is the number of classes. For image clustering, $l = 0$, and u is the total number of images. Most previous semi-supervised learning (SSL) methods learn a classifier $\phi : \mathcal{X} \mapsto \mathcal{Y}$ from $\mathcal{D}_l$ with a regulation term learned from $\mathcal{D}_u$. Our method learns a new image representation $\mathbf{f}$ from all known data $\mathcal{D} = \mathcal{D}_l \cup \mathcal{D}_u$, and trains standard classifier models ϕ with $\mathbf{f}$. $\mathbf{f}_i$ is a vector of similarities of image i to a series of sampled image prototypes.

Assume that Ensemble Projection (EP) learns knowledge from T prototype sets $\mathcal{P}^{t,t\in\{1,...,T\}} = \{(s_i^t, c_i^t)\}_{i=1}^{rn}$, where $s_i^t \in \{1, ..., l+u\}$ is the index of the i^{th} chosen image, $c_i^t \in \{1, ..., r\}$ is the pseudo-label indicating which prototype s_i^t belongs to. r is the number of prototypes (surrogate classes) in $\mathcal{P}^t$, and n the number of images sampled for each prototype (class) (*e.g.* $r = 3$ and $n = 3$ in Fig. 4.1). Below, we first present our sampling method for creating a single prototype set $\mathcal{P}^t$ in the tth trial, followed by EP.

4.3.1 Max-Min Sampling

As stated, we want the prototypes to be inter-distinct and intra-compact, so that each one represents a different visual concept. To this end, we design a 2-step sampling method, termed Max-Min Sampling. The Max step is based on the *exotic-inconsistency* and caters for the inter-distinct property; the Min-step is based on the *local-consistency* assumption and caters for the intra-compact requirement. In particular, we first sample a skeleton of the prototype set, by looking for image candidates that are strongly spread out, i.e. at large distances from each other. We then enrich the skeleton to a prototype set by including the closest neighbors of the skeleton images. The algorithm for creating $\mathcal{P}^t$ is given in Algo.1. For the skeleton, we sampled m hypotheses – each one consists of r randomly sampled images. For each hypothesis, the average pairwise distance between the r images is then computed. Finally, we take the hypothesis yielding the largest average mutual distance as the skeleton. This simple procedure guarantees that the sampled seed images are far from each other. Once the skeleton is created, the Min-step extends each seed image to an image prototype by introducing its n nearest neighbors (including itself), in order to enrich the characteristics of each image prototype and reduce the risk of introducing noisy images. The pseudo-labels are shared by all images specifying the same prototype. It is worth pointing out that the randomized Max-step may not generate the optimal skeleton. However, it serves its purpose well. For one thing, we do not need the optimal one – we only need the prototypes to be *far apart*, not *farthest apart*. Moreover, randomization allows diverse visual concepts to be captured in different $\mathcal{P}^t$'s. The influence of the optimality of each single skeleton is tested in Sec. 4.4.1. The Euclidean distance is also used here.

4.3.2 Ensemble Projection

We now explore the use of the image prototype sets created in Section 4.3.1 for a new image representation. Because the prototypes are compact in feature space, each of them implicitly defines a visual concept (image attribute). This is especially true when the dataset $\mathcal{D}$ is sufficiently large, which is to be expected given the vast number of unlabeled images that are available. Since the information carried by a single prototype set $\mathcal{P}^t$ is quite limited and noisy, we borrow an idea from ensemble learning (EL), namely to create an ensemble of T such sets to accumulate wisdom from a brood set of training images. A sanity check of this was already presented for a simulated situation in Sec.4.2.2.

As is well-known, EL benefits from the precision of its base learners and their diversity. To obtain a good precision, discriminative learning method is employed for the base learner $\phi_t(.)$; logistic regression is used in our implementation to project each input image $\mathbf{x}$ to the image prototypes to measure the similarities. This choice is both due to its training efficiency and because lower capacity models are better suited for the sparse,

Algorithm 1: Max-Min Sampling in t^{th} trial

Data: Dataset $\mathcal{D}$
Result: Prototype set $\mathcal{P}^t$

begin
  $\hat{e} = 0$; /* Max-step */
  while *iterations* $\leq m$ **do**
    $\mathcal{V} = \{r$ random image indexes$\}$;
    $e = \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{V}} \text{dis}(\mathbf{x}_i, \mathbf{x}_j)$;
    if $e > \hat{e}$ **then**
      $\hat{e} = e$;
      $\hat{\mathcal{V}} = \mathcal{V}$;
    end
  end
  for $i \leftarrow 1$ **to** r **do** /* Min-step */
    $\mathbf{s}_i^t =$ stacked indexes of the n nearest neighbors of $\mathcal{V}(i)$ in $\mathcal{D}$;
    $\mathbf{c}_i^t = (i, i, ..., i) \in \mathbb{R}^n$;
  end
  $\mathbf{s}^t = (\mathbf{s}_1^t, ..., \mathbf{s}_r^t) \in \mathbb{R}^{rn}$ $\mathbf{c}^t = (\mathbf{c}_1^t, ..., \mathbf{c}_r^t) \in \mathbb{R}^{rn}$;
  $\mathcal{P}^t = \{(s_i^t, c_i^t)\}_{i=1}^{rn}$;
end

small-size datasets under consideration. To achieve a high diversity, randomness is introduced in different trials of Max-Min Sampling to create an ensemble of diverse prototype sets, so that a rich set of image attributes are captured. The vector of all similarities is then concatenated and used as a new image representation $\mathbf{f}$ for the final classification. A standard classifier (*e.g.* SVMs, Boosting, or Random Forest) can then be trained on $\mathcal{D}_l$ with the learned feature $\mathbf{f}$ for the semi-supervised classification, as unlabeled data has already been explored when obtaining $\mathbf{f}$. Likely, image clustering is performed by injecting the learned feature to a standard clustering method. The whole procedure of EP is presented in Algo.2. By now, the whole pipeline in Fig.4.1 has been explained.

4.4 Experiments

The effectiveness of the approach is evaluated in the situations of: (1) semi-supervised image classification, where the amount of labeled data is sparse relative to the total amount of data; and (2) image clustering, where no labeled data is provided. In this section, we will first introduce the datasets and the features used, followed by experimental results for the two tasks and their corresponding analysis.

Algorithm 2: Ensemble Projection

Data: Dataset $\mathcal{D}$ with image presentation $\mathbf{x}_i$
Result: Dataset $\mathcal{D}$ with image presentation $\mathbf{f}_i$

begin
  for $t \leftarrow 1$ **to** T **do**
    Sample $\mathcal{P}^t = \{(s_i^t, c_i^t)\}_{i=1}^{rn}$ using Algo. 1 ;
    Train classifiers $\phi^t(.) \in \{1, ..., r\}$ on $\mathcal{P}^t$;
  end
  for $i \leftarrow 1$ **to** $l + u$ **do**
    for $t \leftarrow 1$ **to** T **do**
      Obtain projection vector: $\mathbf{f}_i^t = \phi^t(\mathbf{x}_i)$;
    end
    $\mathbf{f}_i = ((\mathbf{f}_i^1)^\top, ..., (\mathbf{f}_i^T)^\top)^\top$;
  end
end

Datasets: The method is evaluated on diverse classification tasks: texture classification, object classification, scene classification, event classification, style classification, and satellite image classification. Eight standard datasets are used for the evaluation:

- Texture-25 [Lazebnik et al., 2005]: 25 texture classes, with 40 samples per class.
- Caltech-101 [Fei-Fei et al., 2004]: 101 object classes, with 31 to 800 images per class, and 8677 images in total,
- STL-10 [Coates et al., 2011]: 10 object classes including airplane, bird, car, cat, deer, dog, horse, monkey, ship, truck, with 500 training images per class, 800 test images per class, and 100000 unlabeled images for unsupervised learning.
- Scene-15 [Lazebnik et al., 2006]: 15 scene classes with both indoor and outdoor environments, 4485 images in total. Each class has 200 to 400 images.
- Indoor-67 [Quattoni & Torralba, 2009]: 67 indoor classes such as shoe shop, mall and garage, with a total of 15620 images and at least 100 images per class.
- Event-8 [Li & Fei-Fei, 2007]: 8 sports event classes including rowing, badminton, polo, bocce, snowboarding, croquet, sailing, and rock climbing, with a total of 1574 images.
- Building-25 [Xu et al., 2014]: 25 architectural styles such as American craftsman, Baroque, and Gothic, with 4794 images in total.
- LandUse-21 [Yang & Newsam, 2010]: 21 classes of satellite images in terms of land usage, such as agricultural, airplane, forest. There are 2100 images in total, with 100 images per class.

Features: The following three features were used in our earlier papers [Dai et al., 2012b; Dai & Van Gool, 2013] due to their simplicity and low dimensionality: GIST [Oliva & Torralba, 2001], Pyramid of Histogram of Oriented Gradients (PHOG) [Bosch et al., 2007], and Local Binary Patterns (LBP) [Ojala et al., 2002]. However, these features are obsolete and yield results inferior than alternative features recently developed for image classification. In this paper, we replaced them with the CNN features [Donahue et al., 2014b; Chatfield et al., 2014]. These were obtained from an off-the-shelf CNN pre-trained on the ImageNet data. They were chosen as CNN features have achieved state-of-the-art performance for image classification [Krizhevsky et al., 2012a; Dosovitskiy et al., 2014]. For implementation, we used the MatConvNet [Vedaldi & Lenc, 2014] toolbox, with a 21-layer CNN pre-trained model being used. The convolutional results at layer 16 were stacked as the CNN feature vector, with dimensionality of 4096. We also tested the LLE-coded SIFT feature [Wang et al., 2010]. However, it is not on par with the CNN features.

Competing methods: For semi-supervised classification, six classifiers were adopted to evaluate the method, with three baselines: k-NN, Logistic Regression (LR), and SVMs with RBF kernels, and three semi-supervised classifiers: Harmonic Function (HF) [Zhu et al., 2003], LapSVM [Belkin et al., 2006], and Anchor Graph (AG) [Liu et al., 2010]. HF formulates the SSL learning problem as a Gaussian Random Field on a graph for label propagation. LapSVM extends SVMs by including a smoothness penalty term defined on the Laplacian graph. AG aims to address the scalability issue of graph-based SSL, and constructs a tractable large graph by coupling anchor-based label prediction and adjacency matrix design. For image clustering, we compare our learned feature to the original CNN feature with two standard clustering algorithms: k-means and Spectral Clustering. Existing systems for image clustering often report performance on relatively easy datasets and it is hard to compare with them on these standard classification datasets.

Experimental settings: We conducted four sets of experiments: (1) compare our method with competing methods for semi-supervised image classification on the eight datasets, where the unlabeled images are from the same class as the labeled ones; (2) evaluate the robustness of our method against the choice of its parameters and classifier models in the context of semi-supervised image classification; (3) evaluate the performance of our method for the task of self-taught image classification on the STL-10 dataset, where the feature is learned from the unlabeled images and the performance is tested on the labeled set; and (4) evaluate our method for the task of image clustering on the eight datasets.

For all experimental setups except (2), the same set of parameters were used for all the classifiers. We used $k = 1$ for the k-NN classifier, L2-regularized LR of LIBLINEAR [Fan et al., 2008] with $C = 15$, and the SVMs with RBF kernel of LIBSVM [Chang & Lin, 2011] with $C = 15$ and the default g, *i.e.* $g = 1/4096$. For LapSVM, we used the scheme suggested by Belkin et al. [2006]: γ_A was set as the inductive model, and γ_I was set as $\frac{\gamma_I l}{(l+u)^2} = 100\gamma_A l$. For HF, the weight matrix was computed with the Gaussian function $e^{-||\mathbf{x}_i - \mathbf{x}_j||^2/2\sigma^2}$, where σ is automatically set by using the self-tuning

Methods	Scene-15	LandUse-21	Texture-25	Building-25	Event-8	Caltech-101	Indoor-67	STL-10
k-NN	62.4	69.6	81.0	31.9	76.2	70.0	21.1	55.9
k-NN + **EP**	75.6	75.6	<u>84.5</u>	35.7	87.3	71.5	26.6	65.6
LR	73.0	<u>78.0</u>	85.9	38.1	85.5	**81.5**	31.9	65.4
LR + **EP**	**80.0**	**80.6**	**87.5**	**42.1**	**90.4**	<u>80.9</u>	**36.6**	73.0
SVMs	73.0	73.0	81.4	39.6	84.1	77.9	33.2	64.6
SVMs + **EP**	<u>79.8</u>	76.6	84.4	<u>40.0</u>	<u>88.3</u>	76.0	<u>34.9</u>	70.9
HF	45.2	39.1	67.9	18.5	15.6	70.6	7.4	10.3
HF + **EP**	61.5	52.3	74.6	21.2	27.1	75.8	12.1	38.1
AG	72.8	51.3	50.5	34.8	51.0	67.7	24.7	**76.0**
AG + **EP**	78.3	58.5	32.1	37.5	50.5	66.3	24.9	<u>74.9</u>

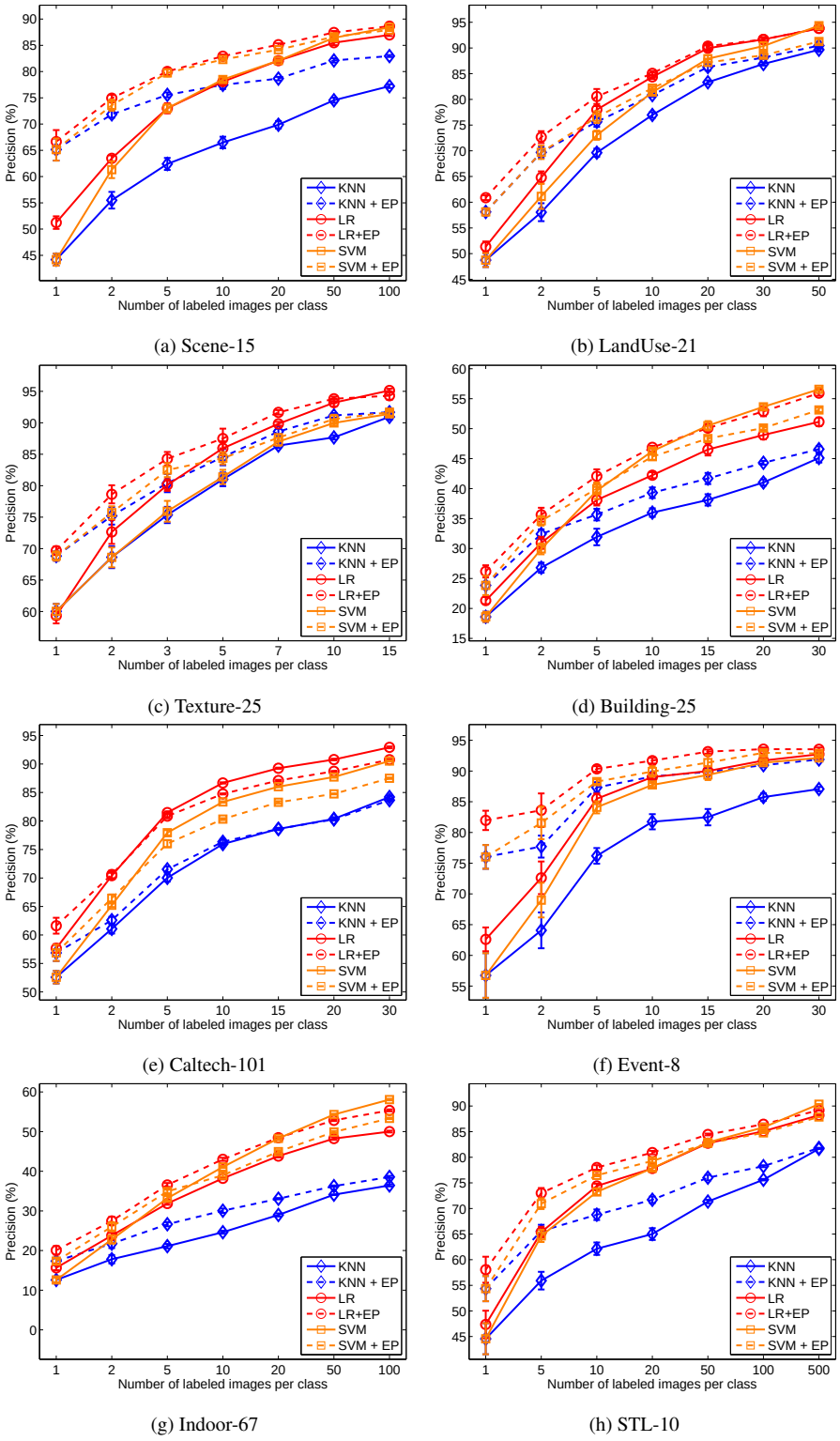
Table 4.1: *Precision (%) of image classification on the eight datasets, with* 5 *labeled training examples per class. "+ EP" indicate that classifiers working with our learned feature as input rather than the original CNN. The best performance is indicated in* ***bold****, and the second best is* <u>underlined</u>*.*

method [Zelnik-Manor & Perona, 2004]. For AG, we followed the suggestion from the original work [Liu et al., 2010] and used the following for both our learned feature and the original CNN feature: 1000 anchors and features reduced to 500 dimensions via PCA.

As to the parameters of our method, a wide variety of values for them were tested in experimental setup (2). In experimental setups (1), (3) and (4), we fixed them to the following values: $T = 100$, $r = 30$, $n = 6$, and $m = 50$, which leads to a feature vector of 3000 dimensions. Note that the learned feature may contain redundancy across different dimensions, as some prototype sets are similar to others. We leave the task of selecting useful features to the discriminative classifiers.

4.4.1 Semi-supervised Image Classification

In this section, we evaluate all methods across all datasets for semi-supervised image classification. Different numbers of training images per class were tested: Scene-15 and Indoor-67 with {1, 2, 5, 10, 20, 50, 100}, LandUse-21 with {1, 2, 5,10, 20, 30, 50}, Texture-25 with {1, 2, 3, 5, 7, 10 , 15}, Building-25, Event-8, and Caltech-101 with {1, 2, 5, 10, 15, 20, 30}, and STL-10 with {1, 5, 10, 20, 50, 100, 500}. The different choices are due to the different structures of the datasets: different number of classes and different number of images per class. In keeping with most existing systems for semi-supervised classification [Zhu et al., 2003; Zhou et al., 2004; Liu et al., 2010; Fergus et al., 2009; Ebert et al., 2010; Pitelis et al., 2014], we evaluate the method in the transductive manner, where we take the training and test samples as a whole, and randomly choose labeled samples from the whole dataset to learn and infer labels of other samples whose labels



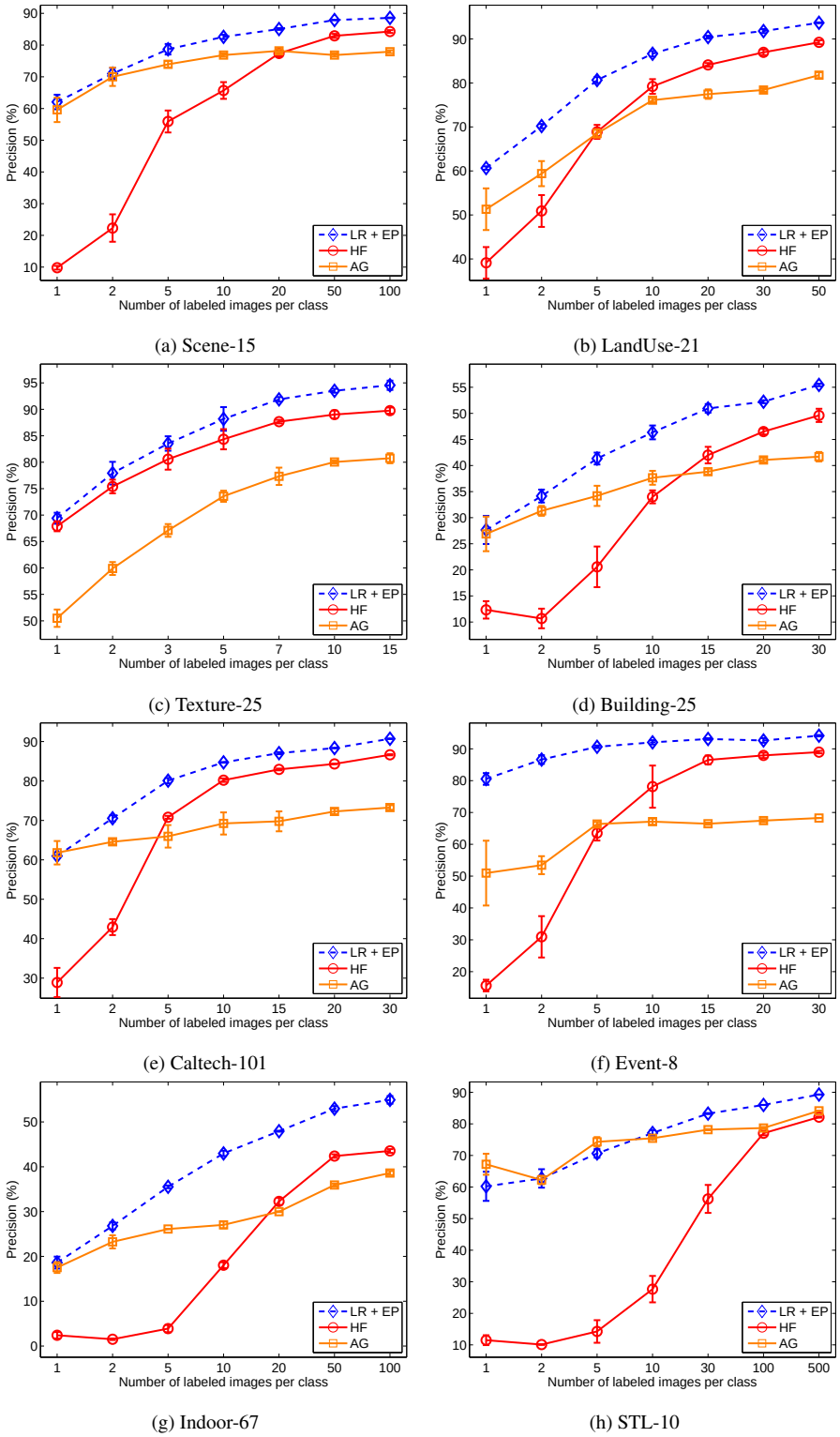
(a) Scene-15 (b) LandUse-21

(c) Texture-25 (d) Building-25

(e) Caltech-101 (f) Event-8

(g) Indoor-67 (h) STL-10

Figure 4.4: *Classification results of Ensemble Projection (EP) on the eight datasets, where three classifiers are used: k-NN, Logistic Regression, and SVMs with RBF kernels. All methods were tested with two feature inputs: the original deep feature and the learned feature by EP on top of it (indicated by "+ EP").*



(a) Scene-15 (b) LandUse-21

(c) Texture-25 (d) Building-25

(e) Caltech-101 (f) Event-8

(g) Indoor-67 (h) STL-10

Figure 4.5: *Classification results of Ensemble Projection (EP) on the eight datasets, where three classifiers are used: k-NN, Logistic Regression, and SVMs with RBF kernels. All methods were tested with two feature inputs: the original deep feature and the learned feature by EP on top of it (indicated by "+ EP").*

are held back as the unlabeled samples. The reported results are the average performance over 5 runs with random labeled-unlabeled splits.

Comparison to baselines: Fig. 4.4 shows the results of the three baseline classifiers with our learned feature and the original CNN feature as input, and Table 4.1 lists the results of all methods when 5 labeled training samples are available for each class. From the figure, it is easy to observe that the three plain classifiers k-NN, LR and SVMs perform consistently better when working with our feature than working with the original CNN features. This is, of course, not a fair comparison, as our feature has been learned with the help of unlabeled samples, while the CNN features not. However, this experiment serves as a good sanity check: given the access to the unlabeled samples, does the proposed feature learning improve the performance of the system over the original feature? The figure shows clear advantages of our method over the original CNN feature across different datasets and classifiers. The most pronounced improvement occurs in the scenarios where a small number of labeled training samples is available, *e.g.* from 1 to 5. This is exactly what the method is designed for – classification tasks where the labeled training samples are sparse relative to the available unlabeled samples. Since LR performs generally the best when working with our learned feature, we will take LR + EP as our method to compare to other SSL methods. The comparison is made in the next section.

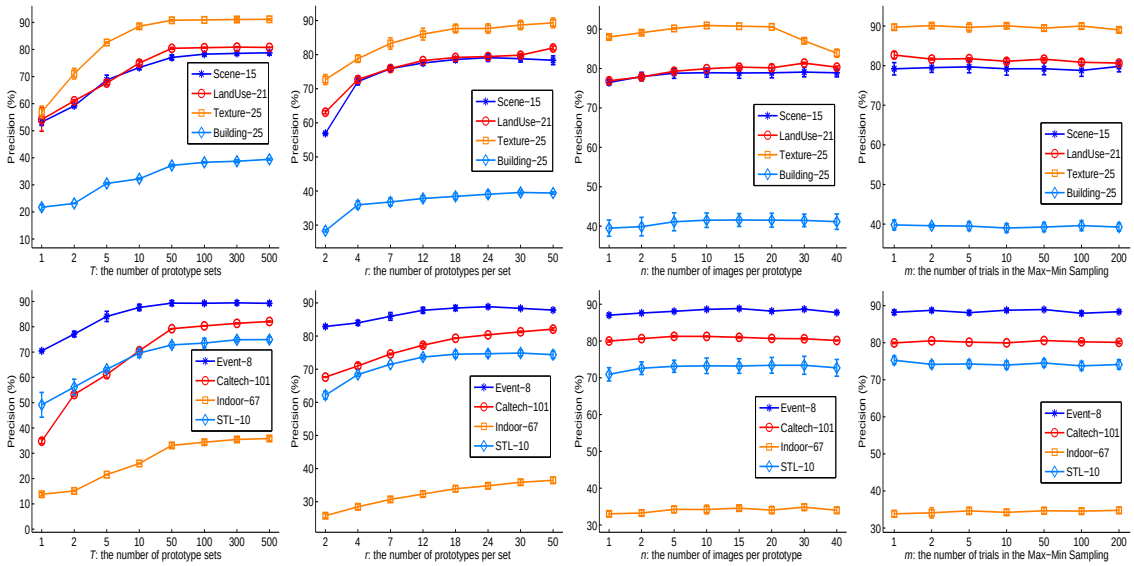
Comparison to other SSL Methods: In this section, we compare our method (LR + EP) with the three SSL methods HF, AG, and LapSVM. The classification precision is reported for HF and AG, while the mean average precision (mAP) of C rounds of binary classification is used for LapSVM. This is because the implementation of LapSVM from the authors performs binary classification [Belkin et al., 2006]. Because LapSVM is computationally expensive, we only compare our method to it for the scenario where 5 labeled training samples per class are used.

Fig. 4.5 shows the results of our method (LR + EP) and that of HF and AG, and Table 4.1 lists the precision of the methods when 5 labeled training examples per class are used. Table 4.2 lists the mAP of our method, HF and LapSVM, when 5 labeled training samples are available for each class. The figure and the tables show that our method outperforms the competing SSL methods consistently for semi-supervised image classification. For instance, if 5 labeled training examples per class are used, our method (LR + EP) improves over the best competing method AG by 7.2% in terms of precision on Scene-15, and by 11.9% on Indoor-67. This suggests that our method can achieve superior results for semi-supervised image classification, even when combined with very standard classifiers. It can be found from the figure and tables that graph-based SSL methods such as HF and AG are not very stable. This is mainly due to their sensitivity to the graph structure, which was observed in [Kingma et al., 2014] as well.

The superior performance of our method can be ascribed to two factors: (1) in addition to the *local-consistency* assumption, our method also exploits the *exotic-inconsistency* assumption; (2) the discriminative projections abstract high-level attributes from the sam-

Methods	Scene-15	LandUse-21	Texture-25	Building-25	Event-8	Caltech-101	Indoor-67	STL-10
LR + **EP**	**84.8**	**85.6**	**95.1**	**39.2**	**91.7**	**73.1**	**33.2**	**81.5**
HF	81.4	84.2	94.1	37.9	89.5	71.6	25.1	78.1
LapSVM	79.2	82.3	91.4	35.8	86.2	56.4	29.3	69.3

***Table 4.2**: MAP (%) of semi-supervised classification on the eight datasets, with 5 labeled training examples per class. "LR + EP" indicate Logistic Regression with our learned feature as input. The other two classifiers use the original CNN feature as input. The best number is indicated in **bold**.*



(a) T (b) r (c) n (d) m

***Figure 4.6**: Performance of EP as a function its parameters T, r, n, and m, where LR is employed with 5 labeled training images per class.*

pled prototypes, *e.g.* being more "yellow-smooth" than "dark-structured". As already proven in fully supervised scenarios [Farhadi et al., 2009; Quattoni et al., 2008], prototype-linked, attribute-based features are very helpful for image classification.

We further investigate the complementarity of our learned feature with other SSL methods for semi-supervised classification. It is interesting to see from the bottom panel of Table 4.1 that using such combinations boosts the performance also. This suggests that our scheme of exploiting unlabeled data and the previous ones doing so capture complementary information. However, using the standard Logistic Regression generally yields the best results for our learned feature.

Robustness to Parameters

In this section, we examine the influence of the parameters of our method on its classification performance. They are the total number of prototype sets T, the number of prototypes in each set r, the number of images in each prototype n, and the number of skeleton hypotheses m used in Max-Min Sampling. LR was used as the classifier here. The parameters were evaluated as follows. Each time the value of one changes while the other ones being kept fixed to the values described in the experimental settings.

Fig. 4.6 shows the results over a range of their values. The figure shows that the performance of our method increases pretty fast with T, but then stabilizes quickly. It implies that the method benefits from exploiting more "novel" visual attributes (image prototypes). After T increases to some threshold (*e.g.* 50 for the eight datasets), basically no new attributes are added, and performance stops going up much. For r, the figure shows that the performance generally increases with it. This is expected because a large r leads to precise attribute assignment. In other words, a large r generates more prototypes per set, thus increasing the possibility of linking every image to its desirable attribute. However, we seen that when r outpaces 24, the increase is not worth the computing time. A larger r would lead to confusing attributes, as it starts to draw very similar or even identical samples into different prototypes. Also, a large r results in high-dimensional features, which in turn cause over-fitting.

For n, a similar trend was obtained – as n increases, the characteristics of the prototypes are enriched, thus boosting the performance. But beyond some threshold (*e.g.* 10 in our experiments), more noisy images are introduced, thus degrading the performance. One possible solution to further enrich the training samples of each prototype is to perform image transformations such as *translation*, *rotation*, and *scaling* to the seed images, and to add the transformed images into the prototype. This technique of enriching training data has been successfully used recently for image classification [Paulin et al., 2014] and for feature learning [Dosovitskiy et al., 2014]. For m, Fig. 4.6 shows that it does not affect the performance as much as the three parameters analyzed so far. This does not mean that there is no need to use the *exotic-inconsistency* assumption. Instead, it suggests that a random selection of r images from a dataset of $l + u$ images already fulfills the requirement of the assumption: images should be apart from each other. This is generally true because $r << l + u$ holds for the datasets considered.

Although the performance of EP will be affected by the choice of its parameters, we can see from Fig. 4.6 that each of the parameters has a wide range of reasonable values to choose from. It is not difficult to choose a set of parameter values that produces better results than competing methods (*c.f.* Fig. 4.6 and Table 4.1). Also, the parameters are quite intuitive and their roles are similar to the parameters of some other methods: analogues of m, n and T can be found in RANSAC, k-NN, and Bagging, for instance.

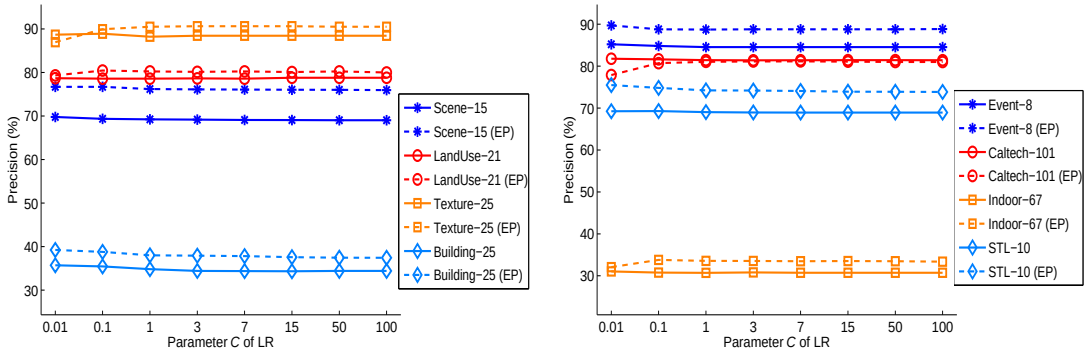


***Figure 4.7**: Comparison of our learned feature to the CNN feature Chatfield et al. [2014], with different LR models.*

Robustness to Classifier Models

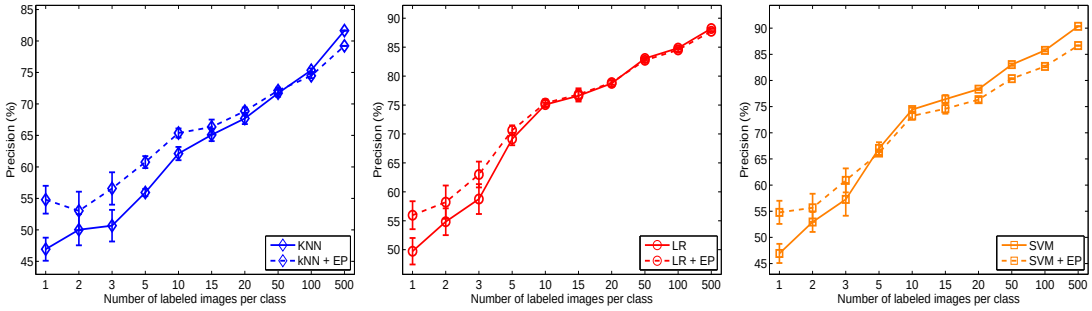
In this section, we evaluate the robustness of our learned features against classifier models. Different values of the balancing parameter C between model accuracy and model complexity were tested for the LR classifier across the eight datasets. 5 labeled training examples per class were used. A set of values $\{0.01, 0.1, 1, 5, 15, 50, 100\}$ were tested for the parameter C of LR. Fig. 4.7 shows the results. It is evident from the figure that our learned feature consistently outperforms the original CNN feature over a large range of parameter values for the classifier models. This property is important for semi-supervised classification, as labeled data is limited in this scenario and probably cannot afford model selection techniques such as Cross-Validation.

Efficiency

Although additional time is needed for feature learning (the direct use of the original feature needs no training at this stage), our method is efficient. The efficiency is due to two reasons: 1) Training logistic regression is very efficient; and 2) the performance of our method stabilizes quickly with respect to T as Fig.4.6 shows. The training on the datasets takes $2-6$ minutes on a Core i5 2.80 GHz desktop PC. Furthermore, our method is inherently parallelizable and can take advantage of multi-core processors. It is worth noting that this extra-training time is compensated by using a simpler classifier such as logistic regression for the classification.

4.4.2 Self-taught Image Classification

In order to evaluate the generality of our method, we tested it in a more general scenario, where the unlabeled data is the set of $100,000$ unlabeled images from the STL-10 dataset. Projection functions were learned from this unlabeled dataset and the performance was



(a) KNN (b) LR (c) SVMs

Figure 4.8: *Self-taught classification results on dataset STL-10, where EP is learned from the unlabeled images. The classifiers were tested with deep features, and our learned feature from it (indicated by "+ EP").*

tested on the STL-10 dataset. Again, we held the training image and test images as a whole, and chose only a small fraction as training images (for the classifiers) with others as test images for evaluation. The average accuracy of 5 runs with random training-test splits was reported. Fig. 4.8 shows the classification performance with different numbers of labeled training images per class. From the figure and table, it can be observed that our learned feature from the random image collection still outperforms the original CNN feature when the number of labeled training images is small. This is a very helpful property for semi-supervised learning, as it happens quite often that one has no prior access to the data to be classified. The success could be ascribed to the fact that the "universal visual world" (the random image collection) contains abundant high-level, valuable visual attributes such as "blue and open" in some image clusters and "textured and man-made" in others. Exploiting these "hidden" visual attributes is very beneficial for narrowing down the semantic gap between low-level features and high-level classification tasks.

However, the figure also shows that as the number of labeled training images increases, the advantage of our learned feature vanishes. The method even produces worse results than the original CNN feature when the number of training samples is large. This is to be expected as the method is designed to improve classification systems by exploiting unlabeled data. Therefore, when a sufficient number of labeled images are available, introducing additional unlabeled ones may hurt the system. This is a general, open problem for semi-supervised learning (self-taught learning) [Li & Zhou, 2011]. One possible solution is to study when the classification systems should switch from semi-supervised learning to fully supervised learning. Another solution could be to use the labeled training images directly as the skeleton to generate the prototype sets. This strategy, however, is more limited than ours, and is difficult to use for tasks, such as image clustering, where no labeled samples are available. We leave these issues as future work.

Methods	Scene-15	LandUse-21	Texture-25	Building-25	Event-8	Caltech-101	Indoor-67	STL-10
k-means	65.5	56.3	59.2	39.3	76.7	67.7	33.1	57.0
k-means + **EP**	71.5	63.6	**73.1**	**43.8**	**87.3**	69.4	37.0	63.5
Spectral Clustering	69.6	59.8	66.6	33.3	82.7	68.2	31.5	52.8
Spectral Clustering + **EP**	**73.6**	**65.2**	70.1	41.0	86.5	**70.7**	**37.2**	**66.4**

***Table 4.3**: Purity (%) of image clustering on the eight datasets, where the CNN feature [Chatfield et al., 2014] and our learned feature from it (indicated by + EP) are used. The best results are indicated in **bold**, and the second best is underlined.*

4.4.3 Image Clustering

In this section, we evaluated our learned feature for the task of image clustering. Given a collection of images without any labels, the task is to group them so that images in the same group are more (semantically) similar to each other than to those in other groups. We follow existing work [Sivic et al., 2005; Tuytelaars et al., 2010; Dai et al., 2010, 2012b; Faktor & Irani, 2012] and evaluate the task on the image classification datasets, in particular on the eight datasets used for semi-supervised image classification. To the best of our knowledge, we are the first to evaluate the performance of image clustering on as many as eight standard classification datasets, some of which are still very challenging for supervised image classification. Most clustering methods have been tested only on relatively simple datasets, such as 4, 7 and 20 classes of the Caltech dataset, and 5 classes of the ETH shape dataset.

Since our main aim is to validate whether the proposed learning is able to boost the performance of the original feature for image clustering, we chose two standard clustering algorithms – Spectral Clustering and k-means – to compare the two features. As to the implementation, we use the parallel implementation of Chen et al. [2011] for Spectral Clustering and the vl-feat library of Vedaldi & Fulkerson [2008] for k-means algorithm. Since Spectral Clustering and k-means both require the number of clusters as a parameter, we set it to the number of semantic classes of the datasets, leading to weakly-supervised image clustering.

Table 4.3 lists the results of the two features when combined with k-means and Spectral Clustering. Purity is used as the evaluation criterion, which measures the percentage of images from the dominant class within their clusters, averaged over all clusters. The dominant class of a cluster is the (semantic) class that has more image members than other classes in the cluster. It is easy to see from the table that features learned by EP outperform the original CNN features for image clustering by a considerable margin. For instance, when k-means is used, EP outperforms the CNN feature by 9.6% on Event-8, and by 6.5% on STL-10; when Spectral Clustering is used, the improvement is 4.0% on Scene-15, and 5.7% on Indoor-67. Again, our feature is learned from the original CNN feature, but goes beyond one single image and captures the *similarity* relationship among

images. The superior performance of the learned feature suggests that it is worth some effort to analyze properties of the datasets to learn a better feature representation before performing image clustering. This is useful for the task of clustering, as all the data is available to use from the very beginning. This pre-processing step of analyzing datasets has not yet raised much attention in the community. We hope that this work will stimulate more efforts in this direction.

4.5 Conclusion

This paper has tackled the problem of feature learning for the task of semi-supervised image classification and image clustering. We proposed as novel concept the *exotic-inconsistency* assumption and designed a simple, yet effective feature learning method to use it along with *local-consistency* to exploit the available, unlabeled data. By using the assumptions, we generate a diverse set of training data for surrogate classes to learn visual attributes in a discriminative way. By doing so, images are classified and linked to the surrogate classes – images are represented with their affinities to a rich set of discovered image attributes for classification and clustering. Experiments on eight standard datasets showed the superior performance of the learned feature for both semi-supervised image classification and image clustering. In addition, the method is conceptually simple, computationally efficient, and flexible to use. The future work is to extend the method to image segmentation and image semantic labeling.

5

conclusion

wawawa

Bibliography

(????a). Cmu sphinx. http://cmusphinx.sourceforge.net/.

(????b). Cmu sphinx. http://www.speech.cs.cmu.edu/tools/lmtool-new.html.

(2015). Smile - smart photo annotation. https://play.google.com/store/apps/details?id=com.neuromorphic.retinet.smile.

Agrawal, P., Carreira, J., & Malik, J. (2015). Learning to see by moving. In *International Conference on Computer Vision*, (pp. 37–45).

Aodha, O. M., Humayun, A., Pollefeys, M., & Brostow, G. J. (2013). Learning a confidence measure for optical flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(5), 1107–1120.

Arbelaez, P., Pont-Tuset, J., Barron, J., Marques, F., & Malik, J. (2014). Multiscale combinatorial grouping. In *CVPR*.

Baker, S., Scharstein, D., Lewis, J., Roth, S., Black, M., & Szeliski, R. (2011). A database and evaluation methodology for optical flow. *IJCV*, *92*(1), 1–31.

Bearman, A., Russakovsky, O., Ferrari, V., & Fei-Fei, L. (2015). What's the point: Semantic segmentation with point supervision. *arXiv:1506.02106*.

Belkin, M., Niyogi, P., & Sindhwani, V. (2006). Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *JMLR*, *7*(36), 2399–2434.

Bell, S., Upchurch, P., Snavely, N., & Bala, K. (2013). Opensurfaces: A richly annotated catalog of surface appearance. In *SIGGRAPH*.

Bennett, K. P., & Demiriz, A. (1998). Semi-supervised support vector machines. In *Advances in Neural Information Processing Systems*, (pp. 368–374).

Berg, T. L., Berg, A. C., Edwards, J., Maire, M., White, R., Teh, Y.-W., Learned-Miller, E., & Forsyth, D. A. (2004). Names and faces in the news. In *CVPR*.

Bilen, H., & Vedaldi, A. (2016). Weakly supervised deep detection networks. In *CVPR*.

Blum, A., & Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, COLT' 98, (pp. 92–100).

Boiman, O., Shechtman, E., & Irani, M. (2008). In defense of nearest-neighbor based image classification. In *CVPR*.

Bosch, A., Zisserman, A., & Muoz, X. (2007). Image classification using random forests and ferns. In *ICCV*.

Branson, S., Wah, C., Schroff, F., Babenko, B., Welinder, P., Perona, P., & Belongie, S. (2010). Visual recognition with humans in the loop. In *ECCV*, (pp. 438–451).

Brostow, G. J., Fauqueur, J., & Cipolla, R. (2009). Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, *30*(2), 88–97.

Bucilua, C., Caruana, R., & Niculescu-Mizil, A. (2006). Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, (pp. 535–541). ACM.

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

Chapelle, O., Schölkopf, B., & Zien, A. (Eds.) (2006). *Semi-Supervised Learning*. Cambridge, MA: MIT Press.

Chatfield, K., Simonyan, K., Vedaldi, A., & Zisserman, A. (2014). Return of the devil in the details: Delving deep into convolutional nets. In *BMVC*.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2015). Semantic image segmentation with deep convolutional nets and fully connected crfs.

Chen, W.-Y., Song, Y., Bai, H., Lin, C.-J., & Chang, E. (2011). Parallel spectral clustering in distributed systems. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *33*(3), 568–586.

Chen, X., & Gupta, A. (2015a). Webly supervised learning of convolutional networks. In *ICCV*.

Chen, X., & Gupta, A. (2015b). Webly supervised learning of convolutional networks. In *ICCV*.

Chen, X., Shrivastava, A., & Gupta, A. (2013). Neil: Extracting visual knowledge from web data. In *IEEE International Conference on Computer Vision*, (pp. 1409–1416).

Chen, Y., Dai, D., Pont-Tuset, J., & Van Gool, L. (2016). Scale-aware alignment of hierarchical image segmentation. In *Computer Vision and Pattern Recognition (CVPR)*.

Cheng, M.-M., Zheng, S., Lin, W.-Y., Vineet, V., Sturgess, P., Crook, N., Mitra, N. J., & Torr, P. (2014). Imagespirit: Verbal guided image parsing. *ACM Trans. Graph.*, *34*(1), 3:1–3:11.

Coates, A., Ng, A. Y., & Lee, H. (2011). An analysis of single-layer networks in unsupervised feature learning. In *International Conference on Artificial Intelligence and Statistics*, (pp. 215–223).

Collins, B., Deng, J., Li, K., & Fei-Fei, L. (2008). Towards scalable dataset construction: An active learning approach. In *ECCV*.

Cordts, M., Omra, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., & Schiele, B. (2016). Cityscapes dataset for semantic urban scene understanding. In *CVPR*.

Dai, D., Kroeger, T., Li, W., & Van Gool, L. (2016a). Draw&tell: Efficient annotation for semantic image segmentation by drawing and speaking. In *in submission to ECCV*.

Dai, D., Kroeger, T., Timofte, R., & Van Gool, L. (2015a). Metric imitation by manifold transfer for efficient vision applications. In *IEEE Conference on Computer Vision and Pattern Recognition*, (pp. 3527–3536).

Dai, D., Prasad, M., Leistner, C., & Gool, L. V. (2012a). Ensemble partitioning for unsupervised image categorization. In *ECCV*.

Dai, D., Prasad, M., Leistner, C., & Van Gool, L. (2012b). Ensemble partitioning for unsupervised image categorization. In *European Conference on Computer Vision*, (pp. 483–496).

Dai, D., Riemenschneider, H., & Van Gool, L. (2014). The synthesizability of texture examples. In *CVPR*.

Dai, D., & Van Gool, L. (2013). Ensemble projection for semi-supervised image classification. In *International Conference on Computer Vision*, (pp. 2072–2079).

Dai, D., Wang, Y., Chen, Y., & Van Gool, L. (2016b). Is image super-resolution helpful for other vision tasks? In *WACV*.

Dai, D., Wu, T., & Zhu, S. C. (2010). Discovering scene categories by information projection and cluster sampling. In *IEEE Conference on Computer Vision and Pattern Recognition*, (pp. 483–496).

Dai, J., He, K., & Sun, J. (2015b). Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *ICCV*.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *CVPR*.

Deng, J., Russakovsky, O., Krause, J., Bernstein, M. S., Berg, A., & Fei-Fei, L. (2014). Scalable multi-label annotation.

Divvala, S., Farhadi, A., & Guestrin, C. (2014). Learning everything about anything: Webly-supervised visual concept learning. In *CVPR*.

Doersch, C., Gupta, A., & Efros, A. A. (2015). Unsupervised visual representation learning by context prediction. In *International Conference on Computer Vision*, (pp. 1422–1430).

Dollár, P., & Zitnick, C. L. (2013). Structured forests for fast edge detection. In *ICCV*.

Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., & Darrell, T. (2014a). Decaf: A deep convolutional activation feature for generic visual recognition. In *ICML*.

Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., & Darrell, T. (2014b). Decaf: A deep convolutional activation feature for generic visual recognition. In *International Conference on Machine Learning*.

Dosovitskiy, A., T. Springenberg, J., Riedmiller, M., & Brox, T. (2014). Discriminative unsupervised feature learning with convolutional neural networks. In *Advances in Neural Information Processing Systems*, (pp. 766–774).

Dueck, D., & Frey, B. J. (2007). Non-metric affinity propagation for unsupervised image categorization.

Dutt Jain, S., & Grauman, K. (2013). Predicting sufficient annotation strength for interactive foreground segmentation. In *ICCV*.

Ebert, S., Fritz, M., & Schiele, B. (2012). Semi-supervised learning on a budget: Scaling up to large datasets. In *ACCV*.

Ebert, S., Larlus, D., & Schiele, B. (2010). Extracting structures in image collections for object recognition. In *European Conference on Computer Vision*, (pp. 720–733).

Everingham, M., Van Gool, L., Williams, C., Winn, J., & Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *IJCV*, *88*(2), 303–338.

Faktor, A., & Irani, M. (2012). "clustering by composition" - unsupervised discovery of image categories. In *ECCV*.

Fan, R.-E., Chang, K. W., Hsieh, C. J., Wang, X. R., & Lin, C. J. (2008). Liblinear: A library for large linear classification. *JMLR*, *9*(6/1/2008), 1871–1874.
URL http://portal.acm.org/citation.cfm?id=1442794

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

Fei-Fei, L., Fergus, R., & Perona, P. (2004). Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories. In *Workshop on Generative-Model Based Vision*.

Felzenszwalb, P. F., Girshick, R. B., & McAllester, D. (2010). Cascade object detection with deformable part models. In *CVPR*.

Fergus, R., Perona, P., & Zisserman, A. (2003). Object class recognition by unsupervised scale-invariant learning. vol. 2, (pp. 264–271).

Fergus, R., Weiss, Y., & Torralba, A. (2009). Semi-supervised learning in gigantic image collections. In *NIPS*.

Fernando, B., Habrard, A., Sebban, M., & Tuytelaars, T. (2013). Unsupervised visual domain adaptation using subspace alignment. In *ICCV*.

Frey, B. J., & Dueck, D. (2007). Clustering by passing messages between data points. *Science*, *315*(5814), 972–976.
URL http://www.ncbi.nlm.nih.gov/pubmed/17218491

Freytag, A., Rodner, E., & Denzler, J. (2014). Selecting influential examples: Active learning with expected model output changes. In *ECCV*.

Geiger, A., Lenz, P., Stiller, C., & Urtasun, R. (2013). Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*.

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014a). Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*.

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

Gong, Y., Kumar, S., Rowley, H. A., & Lazebnik, S. (2013). Learning binary codes for high-dimensional data using bilinear projections. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Gopalan, R., Li, R., & Chellappa, R. (2011). Domain adaptation for object recognition: An unsupervised approach. In *ICCV*.

Grauman, K., & Darrell, T. (2006). Unsupervised learning of categories from sets of partially matching image features.

Graves, A., & Jaitly, N. (2014). Towards end-to-end speech recognition with recurrent neural networks. In *ICML*.

Gregor, K., Danihelka, I., Graves, A., & Wierstra, D. (2015). Draw: A recurrent neural network for image generation. In *ICML*.

Guadarrama, S., Krishnamoorthy, N., Malkarnenkar, G., Venugopalan, S., Mooney, R., Darrell, T., & Saenko, K. (2013). Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. In *ICCV*.

Guillaumin, M., Verbeek, J. J., & Schmid, C. (2010). Multimodal semi-supervised learning for image classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, (pp. 902–909).

Gulshan, V., Rother, C., Criminisi, A., Blake, A., & Zisserman, A. (2010). Geodesic star convexity for interactive image segmentation. In *CVPR*.

Gupta, A., & Davis, L. S. (2008). Beyond nouns: Exploiting prepositions and comparative adjectives for learning visual classifiers. In *ECCV*, (pp. 16–29). Springer.

Gupta, S., Hoffman, J., & Malik, J. (2015). Cross modal distillation for supervision transfer. *arXiv:1507.00448*.

H. Zhang, M. M., A. Berg, & Malik, J. (2006). Svm-knn: Discriminative nearest neighbor classification for visual category recognition.

Hariharan, B., Arbeláez, P., Bourdev, L., Maji, S., & Malik, J. (2011). Semantic contours from inverse detectors. In *ICCV*.

Hartmann, W., Havlena, M., & Schindler, K. (2014). Predicting matchability. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (pp. 9–16).

Hazen, T. J., Sherry, B., & Adler, M. (2007). Speech-based annotation and retrieval of digital photographs. In *INTERSPEECH*.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*.

Hinton, G., Vinyals, O., & Dean, J. (2014). Distilling the knowledge in a neural network. In *Deep Learning and Representation Learning Workshop, NIPS*.

Hoffman, J., Guadarrama, S., Tzeng, E., Hu, R., Donahue, J., Girshick, R., Darrell, T., & Saenko, K. (2014). LSDA: Large scale detection through adaptation. In *NIPS*.

Hoiem, D., Efros, A. A., & Hebert, M. (2008). Putting objects in perspective. *International Journal of Computer Vision*, *80*(1), 3–15.

Hoogs, A., Rittscher, J., Stein, G., & Schmiederer, J. (2003). Video content annotation using visual analysis and a large semantic knowledgebase. In *CVPR*.

Jain, M., van Gemert, J. C., & Snoek, C. G. (2015). What do 15,000 object categories tell us about classifying and localizing actions? In *Computer Vision and Pattern Recognition (CVPR)*, (pp. 46–55).

Jain, P., & Kapoor, A. (2009). Active learning for large multi-class problems. In *IEEE Conference on Computer Vision and Pattern Recognition*, (pp. 762–769).

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., & Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. In *ACM International Conference on Multimedia*.

Joachims, T. (1999). Transductive inference for text classification using support vector machines. In *International Conference on Machine Learning*, (pp. 200–209).

Joshi, A. J., Porikli, F., & Papanikolopoulos, N. (2009a). Multi-class active learning for image classification. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, (pp. 2372–2379). IEEE.

Joshi, A. J., Porikli, F., & Papanikolopoulos, N. (2009b). Multi-class active learning for image classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, (pp. 2372–2379).

Kalashnikov, D., Mehrotra, S., Xu, J., & Venkatasubramanian, N. (2011). A semantics-based approach for speech annotation of images. *IEEE Trans. Knowl. Data Eng.*, *23*(9), 1373–1387.

Karpathy, A., & Fei-Fei, L. (2015). Deep visual-semantic alignments for generating image descriptions. In *CVPR*.

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., & Fei-Fei, L. (2014). Large-scale video classification with convolutional neural networks. In *CVPR*.

Kendall, A., Badrinarayanan, V., & Cipolla, R. (2015). Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. *arXiv preprint arXiv:1511.02680*.

Khoreva, A., Benenson, R., Omran, M., Hein, M., & Schiele, B. (2016). Weakly supervised object boundaries. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kingma, D. P., Mohamed, S., Rezende, D. J., & Welling, M. (2014). Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, (pp. 3581–3589).

Kondermann, C., Mester, R., & Garbe, C. (2008). A statistical confidence measure for optical flows. In *ECCV*.

Kopf, J., Kienzle, W., Drucker, S., & Kang, S. B. (2012). Quality prediction for image completion. *ACM Trans. Graph.*, *31*(6).

Kozakaya, T., Ito, S., & Kubota, S. (2011). Random ensemble metrics for object recognition. In *International Conference on Computer Vision*, (pp. 1959–1966).

Krähenbühl, P., & Koltun, V. (2011). Efficient inference in fully connected crfs with gaussian edge potentials. In *NIPS*.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012a). Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, (pp. 1097–1105).

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012b). Imagenet classification with deep convolutional neural networks. In *NIPS*.

Kulis, B., Saenko, K., & Darrell, T. (2011). What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *CVPR*.

Kulkarni, G., Premraj, V., Ordonez, V., Dhar, S., Li, S., Choi, Y., Berg, A. C., & Berg, T. L. (2013). Babytalk: Understanding and generating simple image descriptions. *PAMI*, *35*(12), 2891–2903.

Kumar Mallapragada, P., Jin, R., Jain, A., & Liu, Y. (2009). Semiboost: Boosting for semi-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *31*(11), 2000–2014.

Lampert, C., Nickisch, H., & Harmeling, S. (2009). Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*.

Laput, G. P., Dontcheva, M., Wilensky, G., Chang, W., Agarwala, A., Linder, J., & Adar, E. (2013). Pixeltone: a multimodal interface for image editing. In *CHI*.

Lazebnik, S., Schmid, C., & Ponce, J. (2005). A sparse texture representation using local affine regions. *PAMI*, *27*(8), 1265–1278.

Lazebnik, S., Schmid, C., & Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*.

Leistner, C., Saffari, A., Santner, J., & Bischof, H. (2009). Semi-supervised random forests. In *International Conference on Computer Vision*, (pp. 506–513).

Li, L.-J., & Fei-Fei, L. (2007). What, where and who? classifying event by scene and object recognition. In *International Conference on Computer Vision*, (pp. 1–8).

Li, L.-J., Su, H., Xing, E. P., & Li, F.-F. (2010). Object bank: A high-level image representation for scene classification & semantic feature sparsification. In *Advances in Neural Information Processing Systems*, (pp. 1378–1386).

Li, W., Dai, D., Tan, M., Xu, D., & Van Gool, L. (2016). Fast algorithms for linear and kernel svm+. In *Computer Vision and Pattern Recognition (CVPR)*.

Li, Y.-F., & Zhou, Z.-H. (2011). Towards making unlabeled data never hurt. In *International Conference on Machine Learning*, (pp. 175–188).

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollr, P., & Zitnick, C. (2014). Microsoft coco: Common objects in context. In *ECCV*.

Liu, B., & He, X. (2015). Multiclass semantic video segmentation with object-level active inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (pp. 4286–4294).

Liu, W., He, J., & Chang, S.-F. (2010). Large graph construction for scalable semi-supervised learning. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, (pp. 679–686).

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *CVPR*.

Maas, A. L., Xie, Z., Jurafsky, D., & Ng, A. Y. (2015). Lexicon-free conversational speech recognition with neural networks. In *Proceedings the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Martinovic, A., Knopp, J., Riemenschneider, H., & Van Gool, L. (2015). 3d all the way: Semantic segmentation of urban scenes from start to end in 3d. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (pp. 4456–4465).

Mottaghi, R., Chen, X., Liu, X., Cho, N.-G., Lee, S.-W., Fidler, S., Urtasun, R., & Yuille, A. (2014). The role of context for object detection and semantic segmentation in the wild. In *CVPR*.

Ojala, T., Pietikäinen, M., & Mäenpää, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*(7), 971–987.

Oliva, A., & Torralba, A. (2001). Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV*, *42*(3), 145–175.

Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. *IEEE Trans. on Knowl. and Data Eng.*, *22*(10), 1345–1359.

Papadopoulos, D., Clarke, A., Keller, F., & Ferrari, V. (2014). Training object class detectors from eye tracking data. In *ECCV*.

Papadopoulos, D. P., Uijlings, J. R. R., Keller, F., & Ferrari, V. (2016). We don't need no bounding-boxes: Training object class detectors using only human verification. In *CVPR*.

Papandreou, G., Chen, L.-C., Murphy, K., & Yuille, A. L. (2015). Weakly- and semi-supervised learning of a deep convolutional network for semantic image segmentation. In *ICCV*.

Park, M.-G., & Yoon, K.-J. (2015). Leveraging stereo matching with learning-based confidence measures. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, (pp. 101–109).

Pathak, D., Krähenbühl, P., & Darrell, T. (2015a). Constrained convolutional neural networks for weakly supervised segmentation. In *ICCV*.

Pathak, D., Shelhamer, E., Long, J., & Darrell, T. (2015b). Fully convolutional multi-class multiple instance learning. In *ICLR*.

Paulin, M., Revaud, J., Harchaoui, Z., Perronnin, F., & Schmid, C. (2014). Transformation pursuit for image classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, (pp. 3646–3653).

Pinheiro, P. H. O., & Collobert, R. (2015). From image-level to pixel-level labeling with convolutional networks. In *CVPR*.

Pitelis, N., Russell, C., & Agapito, L. (2014). Semi-supervised learning using an unsupervised atlas. In *Machine Learning and Knowledge Discovery in Databases*, vol. 8725, (pp. 565–580).

Prest, A., Leistner, C., Civera, J., Schmid, C., & Ferrari, V. (2012). Learning object class detectors from weakly annotated video. In *IEEE Conference on Computer Vision and Pattern Recognition*, (pp. 3282–3289).

Quattoni, A., Collins, M., & Darrell, T. (2008). Transfer learning for image classification with sparse prototype representations. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Quattoni, A., & Torralba, A. (2009). Recognizing indoor scenes. In *CVPR*.

Raina, R., Battle, A., Lee, H., Packer, B., & Ng, A. Y. (2007). Self-taught learning: transfer learning from unlabeled data. In *International Conference on Machine Learning*, (pp. 759–766).

Rokach, L. (2010). Ensemble-based classifiers. *Artificial Intelligence Review*, *33*(1-2), 1–39.

Romero, A., Ballas, N., Kahou, S. E., Chassang, A., Gatta, C., & Bengio, Y. (2014). Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*.

Rosch, E. (1978). Principles of categorization. *Cognition and Categorization*, (pp. 27–48).

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A., & Fei-Fei, L. (2015a). Imagenet large scale visual recognition challenge. *IJCV*, (pp. 1–42).

Russakovsky, O., Li, L.-J., & Fei-Fei, L. (2015b). Best of both worlds: human-machine collaboration for object annotation. In *CVPR*.

Russell, B. C., Torralba, A., Murphy, K. P., & Freeman, W. T. (2008a). Labelme: A database and web-based tool for image annotation. *IJCV*, *77*(1-3), 157–173.

Russell, B. C., Torralba, A., Murphy, K. P., & Freeman, W. T. (2008b). Labelme: a database and web-based tool for image annotation. *International Journal of Computer Vision*, *77*(1-3), 157–173.

Rusu, A. A., Colmenarejo, S. G., Gulcehre, C., Desjardins, G., Kirkpatrick, J., Pascanu, R., Mnih, V., Kavukcuoglu, K., & Hadsell, R. (2016). Policy distillation. In *ICLR*.

Saini, P., & Kaur, P. (2013). Automatic speech recognition: A review. *IJETT*.

Sharmanska, V., Quadrianto, N., & Lampert, C. (2012). Augmented attribute representations. In *European Conference on Computer Vision*, (pp. 242–255).

Shotton, J., Winn, J., Rother, C., & Criminisi, A. (2009). Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *IJCV*.

Shrivastava, A., Singh, S., & Gupta, A. (2012). Constrained semi-supervised learning using attributes and comparative attributes. In *European Conference on Computer Vision*, (pp. 369–383).

Silberman, N., Hoiem, D., Kohli, P., & Fergus, R. (2012). Indoor segmentation and support inference from rgbd images. In *ECCV*.

Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *ICLR*.

Singh, S., Gupta, A., & Efros, A. A. (2012). Unsupervised discovery of mid-level discriminative patches. In *European Conference on Computer Vision*, (pp. 73–86).

Sivic, J., Russell, B. C., Efros, A. A., Zisserman, A., & Freeman, W. T. (2005). Discovering objects and their location in images. In *ICCV*.

Sivic, J., Russell, B. C., Zisserman, A., Freeman, W. T., & Efros, A. A. (2008). Unsupervised discovery of visual object class hierarchies.

Srihari, R., & Zhang, Z. (2000). Show&tell: a semi-automated image annotation system. *MultiMedia, IEEE*, *7*(3), 61–71.

Tang, T.-Y., Tzu-Wei, H., & Chen, H.-T. (2013). Random exemplar hashing. In *DDS*.

Thomason, J., Venugopalan, S., Guadarrama, S., Saenko, K., & Mooney, R. (2015). Integrating language and vision to generate natural language descriptions of videos in the wild. In *COLING*.

Torresani, L., Szummer, M., & Fitzgibbon, A. (2010). Efficient object category recognition using classemes. In *European Conference on Computer Vision*, (pp. 776–789).

Tuytelaars, T., Lampert, C. H., Blaschko, M. B., & Buntine, W. (2010). Unsupervised object discovery: A comparison. *International Journal of Computer Vision*, *88*(2), 284–302.

Vedaldi, A., & Fulkerson, B. (2008). VLFeat: An open and portable library of computer vision algorithms. http://www.vlfeat.org/.

Vedaldi, A., & Lenc, K. (2014). Matconvnet – convolutional neural networks for matlab. *CoRR*, *abs/1412.4564*.

Verbeek, J., & Triggs, B. (2007). Region classification with markov field aspect models. In *CVPR*.

Vezhnevets, A., Buhmann, J., & Ferrari, V. (2012). Active learning for semantic segmentation with expected change. In *CVPR*.

Vijayanarasimhan, S., & Grauman, K. (2012). Active frame selection for label propagation in videos. In *ECCV*.

Vinyals, O., Toshev, A., Bengio, S., & Erhan, D. (2015). Show and tell: A neural image caption generator. In *CVPR*.

Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *CVPR*.

Von Ahn, L. (2006). Games with a purpose. *Computer*, *39*(6), 92–94.

Wang, J., Yang, J., Yu, K., Lv, F., Huang, T., & Gong, Y. (2010). Locality-constrained linear coding for image classification. In *CVPR*.

Wang, X., & Gupta, A. (2015). Unsupervised learning of visual representations using videos. In *International Conference on Computer Vision*, (pp. 2794–2802).

Weston, J., Ratle, F., Mobahi, H., & Collobert, R. (2012). Deep learning via semi-supervised embedding. In *Neural Networks: Tricks of the Trade*, (pp. 639–655). Springer.

Xiao, J., Hays, J., Ehinger, K. A., Oliva, A., & Torralba, A. (2010). Sun database: Large-scale scene recognition from abbey to zoo.

Xu, J., Schwing, A. G., & Urtasun, R. (2015a). Learning to segment under various forms of weak supervision. In *CVPR*.

Xu, L., Ren, J., Yan, Q., Liao, R., & Jia, J. (2015b). Deep edge-aware filters. In *ICML*.

Xu, Z., Tao, D., Zhang, Y., Wu, J., & Tsoi, A. (2014). Architectural style classification using multinomial latent logistic regression. In *European Conference on Computer Vision*, (pp. 600–615).

Yang, M., Dai, D., Shen, L., & Van Gool, L. (2014). Latent dictionary learning for sparse representation based classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, (pp. 4138–4145).

Yang, Y., & Newsam, S. (2010). Bag-of-visual-words and spatial extensions for land-use classification. In *International Conference on Advances in Geographic Information Systems*, (pp. 270–279). ACM.

Yao, A., Gall, J., Fanelli, G., & Gool, L. V. (2011). Does human action recognition benefit from pose estimation? In *British Machine Vision Conference (BMVC)*.

Yao, A., Gall, J., Leistner, C., & Van Gool, L. (2012). Interactive object detection. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, (pp. 3242–3249). IEEE.

Yu, F. X., Cao, L., Feris, R. S., Smith, J. R., & Chang, S.-F. (2013). Designing category-level attributes for discriminative visual recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, (pp. 771–778).

Zelnik-Manor, L., & Perona, P. (2004). Self-tuning spectral clustering. In *Advances in Neural Information Processing Systems*, (pp. 1601–1608).

Zhang, P., Wang, J., Farhadi, A., Hebert, M., & Parikh, D. (2014). Predicting failures of vision systems. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (pp. 3566–3573).

Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C., & Torr, P. (2015). Conditional random fields as recurrent neural networks. In *ICCV*.

Zhou, D., Bousquet, O., Lal, T. N., Weston, J., & Schlkopf, B. (2004). Learning with local and global consistency. In *Advances in Neural Information Processing Systems*, (pp. 321–328).

Zhu, X., Ghahramani, Z., & Lafferty, J. (2003). Semi-supervised learning using gaussian fields and harmonic functions. In *International Conference on Machine Learning*, (pp. 912–919).

Zhu, X., & Goldberg, A. B. (2009). *Introduction to Semi-Supervised Learning*.

Zitnick, C. L., & Doll, P. (2014). Edge Boxes : Locating Object Proposals from Edges. In *ECCV*.

List of Publications

Journal Publications

1. A. Yao, J. Gall, L. Van Gool. Coupled Action Recognition and Pose Estimation from Multiple Views. *International Journal of Computer Vision (IJCV)*, 2012. 100(1), 16-37.

2. J. Gall, A. Yao, N. Razavi, L. Van Gool and V. Lempitsky. Hough Forests for Object Detection, Tracking and Action Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2011. 33(11), 2188-2202.

3. A.Y.J. Yao and W. Einhaeuser. Colour aids late but not early stages of rapid natural scene recognition. *Journal of Vision*, 2008. 8(16):12, 1-13.

Refereed Conference Proceedings

1. A. Yao, J. Gall, C. Leistner and L. Van Gool. Interactive Object Detection. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

2. A. Yao, J. Gall, L. Van Gool, and R. Urtasun. Learning Probabilistic Non-Linear Latent Variable Models for Tracking Complex Activities. *Neural Information Processing Systems (NIPS)*, 2011.

3. A. Yao, J. Gall, G. Fanelli and L. Van Gool. Does Human Action Recognition Benefit from Pose Estimation? In *Proceedings British Machine Vision Conference (BMVC)*, 2011.

4. A. Yao, D. Uebersax, J. Gall and L. Van Gool. Tracking People in Broadcast Sports. In *Proceedings German Association for Pattern Recognition (DAGM)*, 2010.

5. J. Gall, A. Yao and L. Van Gool. 2D Action Recognition Serves 3D Human Pose Estimation. In *Proceedings European Conference on Computer Vision (ECCV)*, 2010.

6. A. Yao, J. Gall and L. Van Gool. a Hough Transform-Based Voting Framework for Action Recognition. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

Others

1. D. Waltisberg, A. Yao, J. Gall and L. Van Gool. Variations of a Hough-Voting Action Recognition System. *Proceedings International Conference on Pattern Recognition (ICPR) Contests*, 2010.

2. G. Fanelli, A. Yao, P.L. Noel, J. Gall and L. Van Gool. Hough Forest-Based Facial Expression Recognition from Video Sequences. *International Workshop on Sign, Gesture and Activity (SGA)*, 2010.

Curriculum Vitae

Personal Data

Name	Stefan Saur
Date of birth	1^{st} December 1979
Place of birth	Buchen (Odenwald), Germany
Citizenship	German

Education

2005 – 2009	*ETH Zurich, Computer Vision Laboratory, Switzerland*
	Doctoral studies
2004	*National University of Singapore, Singapore*
	Semester abroad
2000 – 2005	*University of Karlsruhe, Germany*
	Studies of Electrical Engineering and Information Technology
	Graduation with the degree Dipl.-Ing.
1990 – 1999	*Ganztagsgymnasium Osterburken, Germany*

Work Experience

2005 – 2008	*ETH Zurich, Computer Vision Laboratory, Switzerland*
	Teaching and research assistant
2000 – 2005	*Siemens AG, Germany*
	Several internships, semester project, and master thesis
2001 – 2009	*Webdesign, self-employed*

Awards

2004 *Baden-Württemberg Stipendium*, Landesstiftung Baden-Württemberg
2000 *IPP award*, University of Karlsruhe, Germany