Unsupervised High-level Feature Learning by Ensemble Projection for Semi-supervised Image Classification and Image Clustering ☆

Dengxin Dai*, Luc Van Gool

Computer Vision Lab, ETH Zürich, CH-8092, Switzerland

Abstract

This paper investigates the problem of image classification with limited or no annotations, but abundant unlabeled data. The setting exists in many tasks such as semi-supervised image classification, image clustering, and image retrieval. Unlike previous methods, which develop or learn sophisticated regularizers for classifiers, our method learns a new image representation by exploiting the distribution patterns of all available data. Particularly, a rich set of visual prototypes are sampled from all available data, and are taken as surrogate classes to train discriminative classifiers; images are projected via the classifiers; the projected values, similarities to the prototypes, are stacked to build the new feature vector. The training set is noisy. Hence, in the spirit of ensemble learning we create a set of such training sets which are all diverse, leading to diverse classifiers. The method is dubbed Ensemble Projection (EP). EP captures not only the characteristics of individual images, but also the relationships among images. It is conceptually simple and computationally efficient, yet effective and flexible. Experiments on nine standard datasets show that: (1) EP outperforms previous methods for semi-supervised image classification; (2) EP produces promising results for self-taught image classification, where unlabeled samples are a random collection of images rather than being from the same distribution as the labeled ones; and (3) EP improves over the original features for image clustering. The code of the method is available at `www.vision.ee.ethz.ch/~daid/EnProDeepFets`.

Keywords: High-level Feature Learning, Learning with Limited Supervision, Semi-supervised Image Classification, Image Clustering, Ensemble Learning, Ensemble Projection

1. Introduction

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 [1, 2, 3, 4, 5, 6] heavily rely on manually labeled training data, which is expensive

*I am corresponding author

Email addresses: dai@vision.ee.ethz.ch (Dengxin Dai), vangool@vision.ee.ethz.ch (Luc Van Gool)

URL: www.vision.ee.ethz.ch/~daid/ (Dengxin Dai)

and sometimes impractical to acquire. Despite substantial efforts towards efficient annotation by developing online games [7] or appealing software tools [8], collecting training data for classification 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 unsupervised feature learning [9, 10, 11, 12], semi-supervised learning [13, 14, 15], active learning [16, 17], transfer learning [18, 19], weakly-supervised learning [20, 21], self-taught learning [22], and image clustering [23, 24].

In this paper, we are interested in the problem of image classification with limited or no annotation. Instead of regularizing the classifiers like most of the previous methods [25, 26, 27, 28], we learn a new feature representation using all available data (labeled + unlabeled). Specifically, we aim to learn a new feature representation by exploiting the distribution patterns of the data to be handled. The setting assumes the availability of unlabeled data in the same or a similar distribution as the test data. This form of weak supervision is naturally available in applications such as semi-supervised image classification and image clustering, where data in the same or a similar distribution as the test data is available. The learned feature is specifically tuned for the data distribution of interest and performs better for the data than the standard features the method started with. The features to start with for our method can be hand-crafted features [29, 30, 31, 3], learned features in a supervised manner [32, 33, 34] or learned features in an unsupervised way [9, 10, 11, 12, 35].

Learning with unlabeled data has been quite successful in many fields, for instance in semi-supervised learning (SSL) [36, 37, 13, 28, 38, 39], in image clustering [40, 41, 24], and in unsupervised feature representation learning [9, 10, 12, 35]. Typically these methods build 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 training process for the classifiers or feature representations. In this paper, we propose another way to exploit the *local-consistency* assumption to learn a new feature representation. 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 classifier choice. 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.

1.1. 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 classes 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 [42], that states that an object's class is determined by its *similarity* to prototypes which

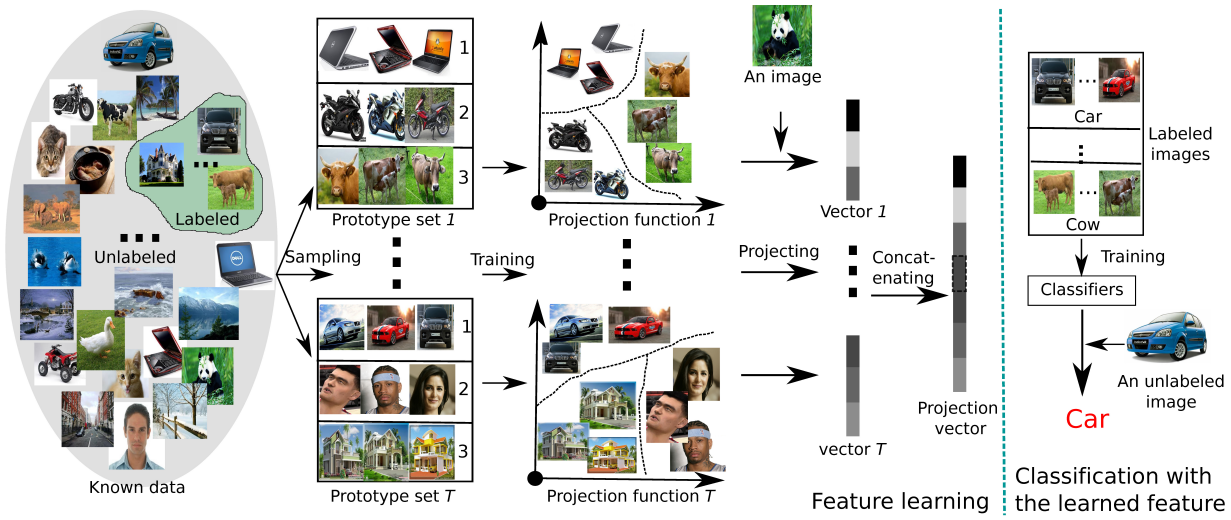


Figure 1: The pipeline of Ensemble Projection (EP). EP consists of unsupervised feature learning (left panel) and plain classification or clustering (right panel). For feature learning, EP samples an ensemble of T diverse prototype sets from all known images and learns discriminative classifiers on them 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.

represent object classes. The theory has been used successfully in transfer learning [18], when labeled data of different classes are available. An important question is whether the theory can also be used for feature representation learning 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. In keeping with Eleanor Rosch's prototype theory [42], 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-consistency*, 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 Section 3.1. 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 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 [43] 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 Section 3.2, 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 [42]. We call the method Ensemble Projection (EP). Its schematic diagram is sketched in Figure 1.

1.2. Contributions

EP was evaluated on nine 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, EP is compared to the original features it started with. Two standard clustering methods are used: k-means and Spectral Clustering. The experiments show that: (1) EP improves over the original features by exploiting the data distribution of interest, 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; (3) EP improves over the original features for image clustering.

This paper is an extension of our conference papers [44, 15]. In addition to putting the two tasks, image clustering and semi-supervised image classification, into the same framework, this paper brings several new contributions. First, in the conference papers, EP was validated only with hand-crafted features, such as LBP [30], GIST [29], and PHOG [31]. These features, however, are obsolete for image classification. Recently, features learned by CNN has resulted in state-of-the-art performance in various classification tasks [6, 32, 33, 34]. In this paper, we validate the efficacy of EP also with CNN features. Second, experiments are conducted on nine standard classification datasets instead of only four in [15]. Third, more analyses and insights are given. Our feature learning method can be used for other tasks as well. For instance, [45] extended the idea to generate hashing functions for efficient image retrieval.

The rest of this paper is organized as follows. Section 2 reports on related work. Section 3 describes the observations that motivate the method. Section 4 is devoted to the approach, followed by experiments in Section 5. Section 6 concludes the paper.

2. Related Work

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

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 [46], Classemes [47], and Object Bank [48]. 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.* [49, 50], to avoid the extra attribute-level supervision, but they still require canonical class-level supervision. Our representation learning however, is fully unsupervised. The pre-trained

CNN features [6, 32, 33, 34] have shown state-of-the-art performance on various classification tasks. Our feature learning is complementary to their methods. As shown in the experiment, our method can improve on top of the CNN features by exploiting the distribution patterns of the data to be classified. Although the technique of fine-tuning can boost the performance of CNN features for the specific tasks at hand [51, 52], it needs labeled data of a moderate size, which is not always available in our setting. Our method can be understood as unsupervised feature enhancing or fine-tuning.

Unsupervised Feature Learning: Our method is akin to methods which learn middle- or high-level image representation in an unsupervised manner. [9] employs k-means mining filters of image patches and then applys the filters for feature computation. [10] 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. Other forms of weak supervision have also been exploited to learn good feature representation without human labeled data, and they all obtain very promising results. For instance, [11] uses the spatial relationships of image windows in an image as the supervision to train a neural network; [35] exploits the tracking results of objects in videos to guide the training of a neural network to learn feature representations; and [53] exploits the ego-motion of cameras for the training. These methods aim for general feature representation. Our method, however, is designed to 'tune' or enhance vision features specifically for the datasets on which the vision tasks are performed.

Filter Learning: [54] 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. Learning with images is mostly useful for holistic image recognition, while learning with patches are more useful for object localization as patches are more focused on the local scales of an image [54, 55]. An extention of our method to patches is interesting.

Semi-supervised Learning: SSL aims at enhancing the performance of classification systems by exploiting an additional set of unlabeled data. Due to its great practical value, SSL has a rich literature [56, 38]. Amongst existing methods, the simplest methodology for SSL is based on the self-training scheme [57] where the system iterates between training classification 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. [14] and [58] presented two methods in this stream for image classification. While obtaining promising results, they both require additional supervision: [14] need image tags and [58] 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 [59], Local-Global Consistency [36], Manifold Regularization

[60], and Eigenfunctions [13]. 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 [26], Semi-supervised SVMs [25], and semi-supervised random forests [28]. 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 [38] 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 [27] and Random Forests [28]. 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 [61], where images are projected to randomly subsampled training classes for supervised distance learning.

Image Clustering: A plethora of methods have been developed for image clustering. [62] modeled objects as constellations of visual parts and estimated parameters using the expectation-maximization algorithm for unsupervised classification. [23] proposed using aspect models to discover object classes from an unordered image collection. Later on, [63] used Hierarchical Latent Dirichlet Allocation to automatically discover object class hierarchies. For scene class discovery, [24] 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. [41] applied the affinity propagation algorithm of [64] for unsupervised image categorization. [40] 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 [65] for a survey.

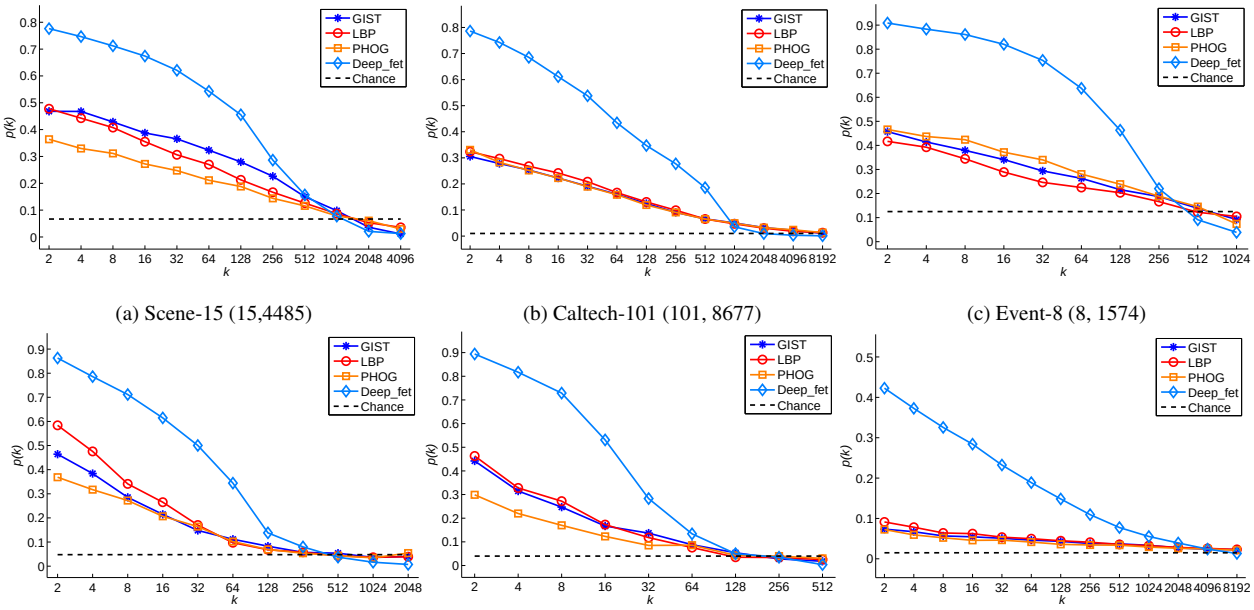
3. Observations

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 assumptions *local-consistency* and *exotic-consistency* 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'?

3.1. Observation 1

The assumptions of *local-consistency* and *exotic-consistency* 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-consistency* 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 [29], PHOG [31], LBP [30], and the CNN feature [34]. The Euclidean distance is used here.

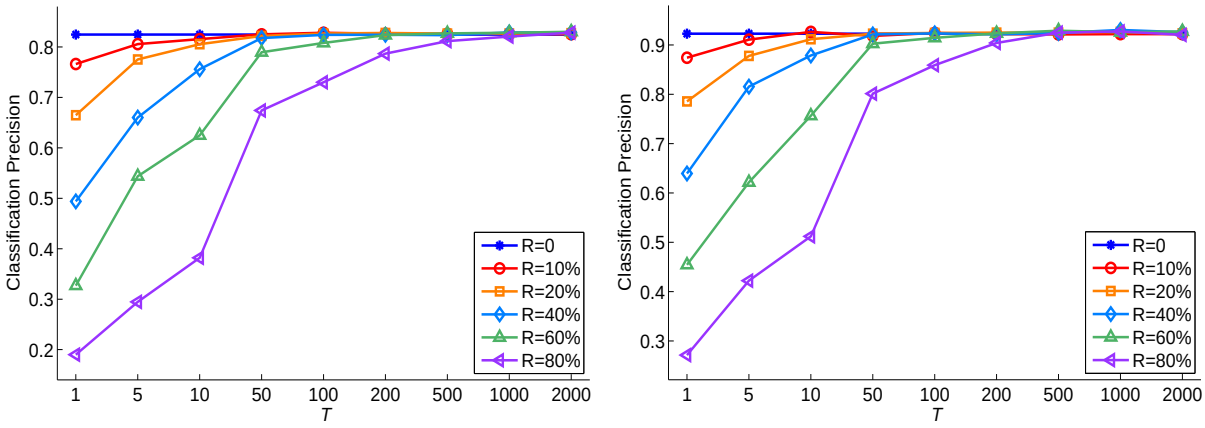


(a) Scene-15 (15,4485) (b) Caltech-101 (101, 8677) (c) Event-8 (8, 1574)

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

Figure 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.

Figure 2 shows the results on six datasets (Datasets and features will be introduced in Section 5). 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 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.



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

Figure 3: Classification accuracy of ensemble learning on the the Scene-15 dataset [1] and the Event-8 dataset [66], 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.

3.2. 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 'oracle' performance 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. Figure 3 evaluates this for the Scene-15 dataset [1]. Logistic Regression is used as the classifiers with the CNN feature [34] 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). The reason this works is that $R = 80\%$ is still able to generate training sets which are better than randomly generated ones ($R = 93.3\% = 1 - 1/15$ for random guessing on Scene-15 with 15 classes). The results suggest 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-consistency* 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. Our Approach

The training data consist 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.

Let us assume that Ensemble Projection (EP) learns knowledge from T prototype sets $\mathcal{P}^t = \{(s_i^t, c_i^t)\}_{i=1}^{r \times n}$, where $t \in \{1, ..., T\}$, $s_i^t \in \{1, ..., l + u\}$ 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 Figure 1). Below, we first present our sampling method for creating a single prototype set $\mathcal{P}^t$ in the tth trial, followed by EP.

4.1. 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-consistency* 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 Algorithm1. 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 Section 5.1.1. The Euclidean distance is used here, but it is easy to change to other distance metrics if needed.

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 $\hat{\mathcal{V}}(i)$ in $\mathcal{D}$;
    $\mathbf{c}_i^t = (i, i, ..., i) \in \mathbb{R}^n$;
  end
  $\mathbf{s}^t = (\mathbf{s}_1^t, ..., \mathbf{s}_r^t) \in \mathbb{R}^{r \times n}$ $\mathbf{c}^t = (\mathbf{c}_1^t, ..., \mathbf{c}_r^t) \in \mathbb{R}^{r \times n}$;
  $\mathcal{P}^t = \{(s_i^t, c_i^t)\}_{i=1}^{r \times n}$;
end

4.2. Ensemble Projection

We now explore the use of the image prototype sets created in Section 4.1 for a new image representation. Because the prototypes are compact in feature space, each of them implicitly defines a visual concept. This is especially true when the dataset $\mathcal{D}$ is 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 Section 3.2.

As is well-known [67], 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, 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 Algorithm2. By now, the whole pipeline in Figure1 has been explained.

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 $\mathcal{P}^t = \{(s_i^t, c_i^t)\}_{i=1}^{r \times n}$using Algorithm 1 ;
    Train classifiers $\phi^t(.) \in [0,1]^r$ on $\mathcal{P}^t$; /* classification scores */
  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

5. 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.

Datasets: The method is evaluated on diverse classification tasks: texture classification [68], object classification [69, 9, 70], scene classification [1, 2], event classification [66], style classification [71], and satellite image classification [72, 73]. Nine standard datasets are used for the evaluation:

- Event-8 [66]: 8 sports event classes including rowing, badminton, polo, bocce, snowboarding, croquet, sailing, and rock climbing, with a total of 1574 images.
- LandUse-21 [72]: 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.
- Texture-25 [68]: 25 texture classes, with 40 samples per class.
- Scene-15 [1]: 15 scene classes with both indoor and outdoor environments, 4485 images in total. Each class has 200 to 400 images.
- Building-25 [71]: 25 architectural styles such as American craftsman, Baroque, and Gothic, with 4794 images in total.
- Caltech-101 [69]: 101 object classes, with 31 to 800 images per class, and 8677 images in total.

- Indoor-67 [2]: 67 indoor classes such as shoe shop, mall and garage, with a total of 15620 images and at least 100 images per class.
- STL-10 [9]: 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.
- ImageNet-50 [70]: 50 classes randomly sampled from the 1000 classes of ImageNet, each class has 1000 images, *i.e.* 50000 images in total.

Features: The following three features were used in our earlier papers [44, 15] due to their simplicity and low dimensionality: GIST [29], Pyramid of Histogram of Oriented Gradients (PHOG) [31], and Local Binary Patterns (LBP) [30]. 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 [32, 34]. 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 [6, 10]. For implementation, we used the MatConvNet [74] 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 [3]. 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) [59], LapSVM [60], and Anchor Graph (AG) [75]. 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 nine 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 nine 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 [76] with $C = 15$, and the SVMs with RBF kernel of LIBSVM [77] with $C = 15$ and the default g, *i.e.* $g = 1/4096$. For LapSVM, we used the scheme suggested by [60]: 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

Methods	Scene-15	LandUse-21	Texture-25	Building-25	Event-8	Caltech-101	Indoor-67	STL-10	ImageNet-50
k-NN	62.4 (1.4)	69.6 (1.0)	81.0 (1.3)	31.9 (1.7)	76.2 (1.5)	70.0 (0.8)	21.1 (0.7)	55.9 (2.1)	47.9 (2.3)
k-NN+**EP**	75.6 (0.6)	75.6 (1.2)	84.5 (1.6)	35.7 (1.2)	87.3 (1.0)	71.5 (0.6)	26.6 (0.6)	65.6 (1.5)	55.4 (1.9)
LR	73.0 (1.2)	78.0 (0.8)	85.9 (1.3)	38.1 (1.1)	85.5 (1.0)	**81.5** (0.2)	31.9 (0.4)	65.4 (1.3)	61.1 (1.5)
LR+**EP**	**80.0** (0.8)	**80.6** (1.7)	**87.5** (1.9)	**42.1** (1.4)	**90.4** (0.5)	80.9 (0.5)	**36.6** (0.6)	73.0 (1.2)	**64.5** (1.4)
SVMs	73.0 (1.0)	73.0 (1.1)	81.4 (1.4)	39.6 (1.7)	84.1 (1.2)	77.9 (0.5)	33.2 (0.6)	64.6 (1.4)	44.8 (2.2)
SVMs+**EP**	79.8 (0.8)	76.6 (1.4)	84.4 (1.2)	40.0 (0.9)	88.3 (0.4)	76.0 (0.7)	34.9 (0.5)	70.9 (1.3)	62.1 (1.9)
HF	45.2 (0.3)	39.1 (3.6)	67.9 (1.0)	18.5 (3.6)	15.6 (1.8)	70.6 (0.3)	7.4 (2.9)	10.3 (0.1)	—
HF+**EP**	61.5 (0.2)	52.3 (1.6)	74.6 (1.0)	21.2 (3.0)	27.1 (7.7)	75.8 (0.4)	12.1 (0.3)	38.1 (8.2)	—
AG	72.8 (1.1)	51.3 (4.7)	50.5 (1.6)	34.8 (1.0)	51.0 (10.2)	67.7 (1.4)	24.7 (0.4)	**76.0** (0.2)	—
AG+**EP**	78.3 (1.1)	58.5 (0.1)	32.1 (2.5)	37.5 (0.0)	50.5 (5.6)	66.3 (2.2)	24.9 (1.5)	74.9 (1.6)	—

Table 1: Precision (%) of image classification on the nine 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 underlined.

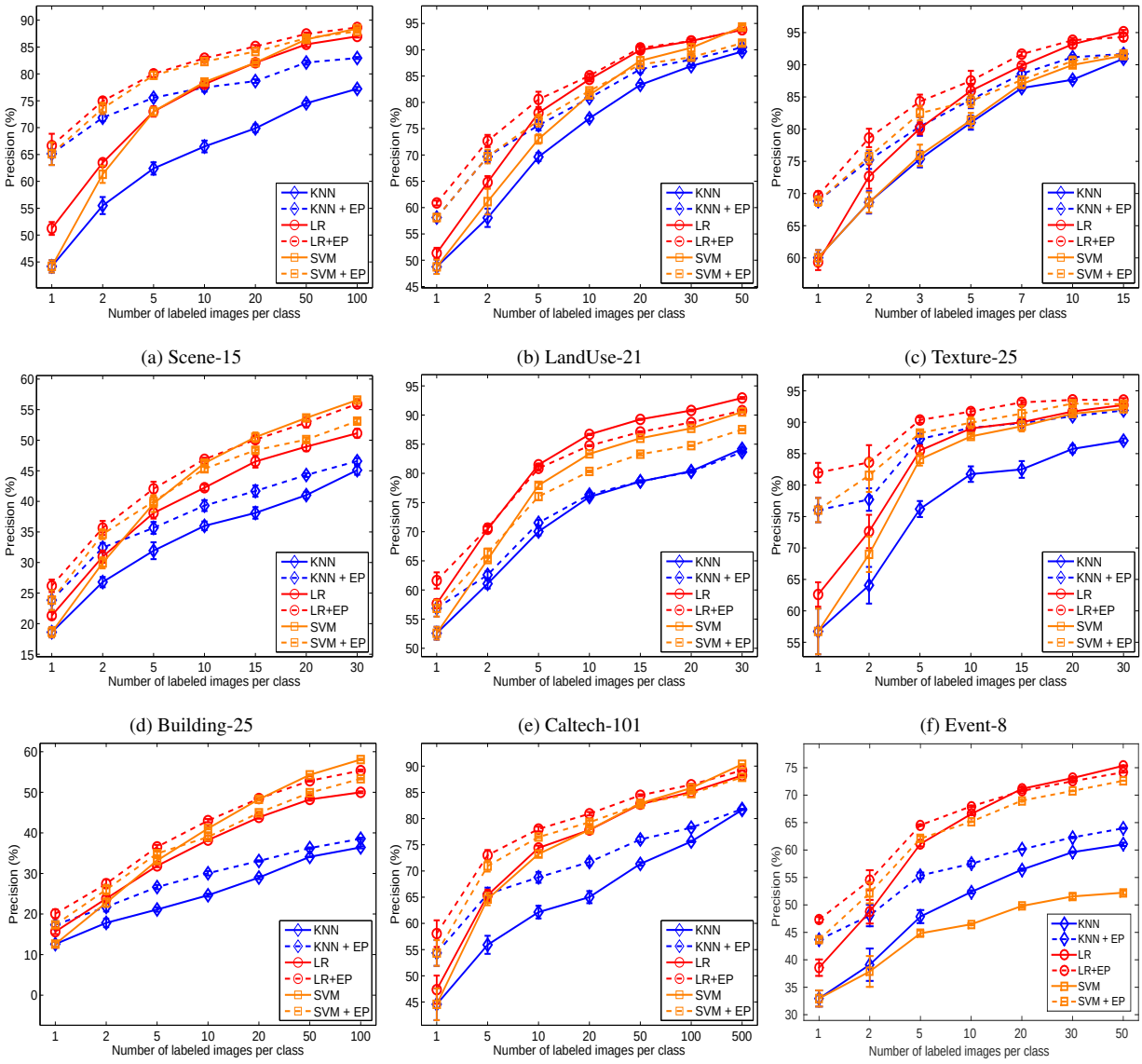
$e^{-||\mathbf{x}_i-\mathbf{x}_j||^2/2\sigma^2}$, where σ is automatically set by using the self-tuning method [78]. For AG, we followed the suggestion from the original work [75] and used the following for both our learned feature and the original CNN feature: 1000 anchors and features reduced to 500 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.

5.1. 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}, STL-10 with {1, 5, 10, 20, 50, 100, 500}, and ImageNet-50 with {1, 2, 5, 10, 20, 30, 50}. 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 [59, 36, 75, 13, 79, 80], 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 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: Figure 4 shows the results of the three baseline classifiers with our learned feature and the original CNN feature as input, and Table 1 lists the results of all methods



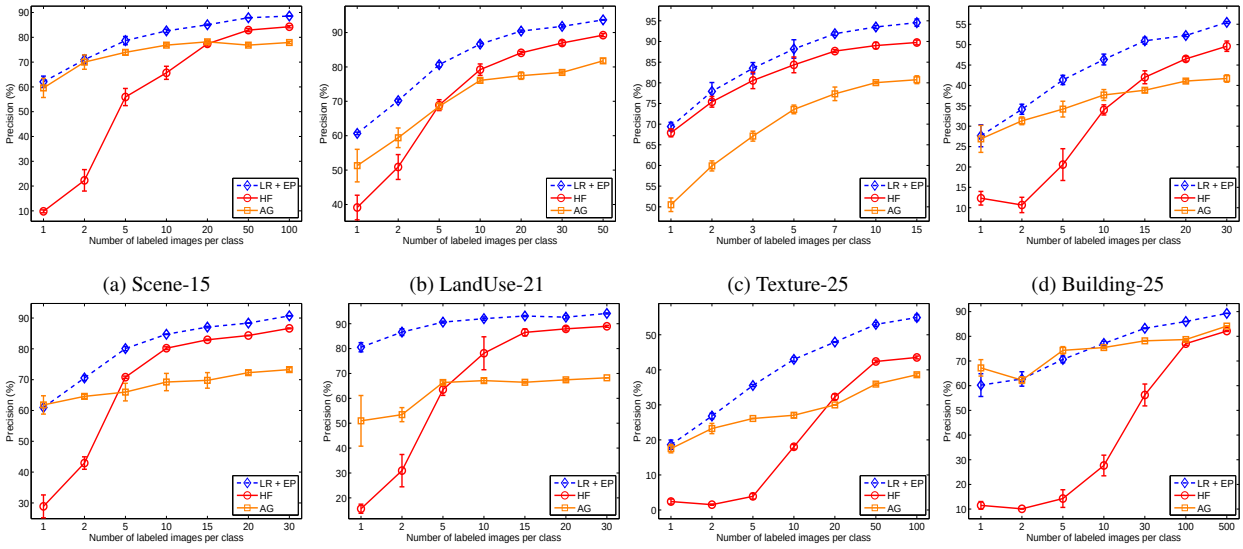
(a) Scene-15 (b) LandUse-21 (c) Texture-25

(d) Building-25 (e) Caltech-101 (f) Event-8

(g) Indoor-67 (h) STL-10 (i) ImageNet-50

Figure 4: Classification results of Ensemble Projection (EP) on the nine 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").

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 very 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



(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 5: Classification results of Ensemble Projection (LR+EP) on eight of the nine datasets, compared to two semi-supervised classifiers HF and AG with the original CNN features.

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.

We also compared EP to Principle Component Analysis (PCA). We applied PCA to reduce the dimensionality of the original CNN features to 3000 (from 4096), and compared it to the features learned by EP. It is found that PCA does not improve the performance of the original features, and performs worse than EP for unsupervised feature learning. This is mainly because EP exploits the local structures of data manifold in a discriminative manner, while PCA exploits the global structure of data distribution via a 'generative' manner. The former usually is more flexible and suitable for classification.

Comparison to other SSL Methods: In this section, we compare our method (LR + EP) with the three SSL methods HF, AG, and LapSVM. We choose not to evaluate these SSL methods on ImageNet-50, due to their high computational cost. 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 [60]. Because LapSVM is computationally expensive, we only compare our method to it for the scenario where 5 labeled training samples per class are used.

Figure 5 shows the results of our method (LR + EP) and that of HF and AG, and Table 1 lists the precision of the methods when 5 labeled training examples per class are used. Table 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

Methods	Scene-15	LandUse-21	Texture-25	Building-25	Event-8	Caltech-101	Indoor-67	STL-10
LR + **EP**	**84.8 (1.3)**	**85.6 (1.0)**	**95.1 (0.8)**	**39.2 (1.6)**	**91.7 (0.6)**	**73.1 (0.3)**	**33.2 (0.1)**	**81.5 (0.8)**
HF	81.4 (1.9)	84.2 (0.9)	94.1 (0.1)	37.9 (1.1)	89.5 (0.9)	71.6 (0.2)	25.1 (0.2)	78.1 (1.0)
LapSVM	79.2 (2.2)	82.3 (0.5)	91.4 (0.6)	35.8 (1.0)	86.2 (0.8)	56.4 (0.6)	29.3 (0.1)	69.3 (1.2)

Table 2: MAP (%) of semi-supervised classification on eight of the nine 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**.

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 [39] as well.

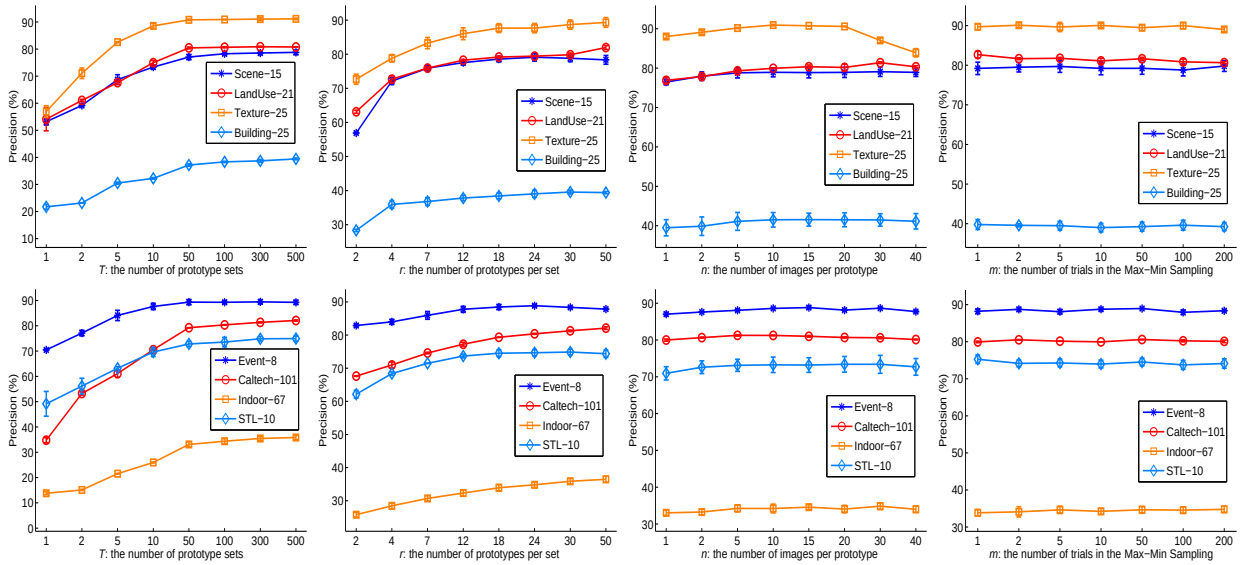
The superior performance of our method to other SSL methods can be ascribed to two factors: (1) in addition to the *local-consistency* assumption, our method also exploits the *exotic-consistency* assumption; (2) the discriminative projections abstract high-level attributes from the sampled prototypes, *e.g*. being more "yellow-smooth" than "dark-structured". As already proven in fully supervised scenarios [46, 18], prototype-linked, attribute-based features are very helpful for image classification. The superior performance of our method to the original feature [34] is that our method learns the statistics of the to-be-classified dataset, while standard CNN features are trained on a different dataset, though very large. The exploitation of dataset-specific properties by EP can be understood as feature enhancing or fine-tuning in an unsupervised manner.

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 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.

5.1.1. 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.

Figure 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 considered datasets), basically no new attributes are added, and



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

Figure 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.

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 [81] and for feature learning [10]. For m, Figure 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-consistency* 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 Figure 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.* Figure 6 and Table 1). Also, the parameters are quite intuitive and their roles are

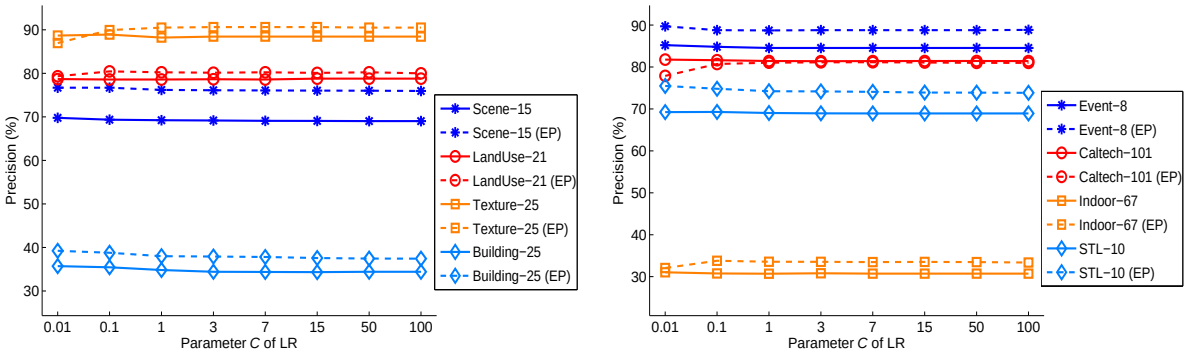


Figure 7: Comparison of our learned feature to the CNN feature [34], with different LR models.

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.

5.1.2. Robustness to Classifier Models

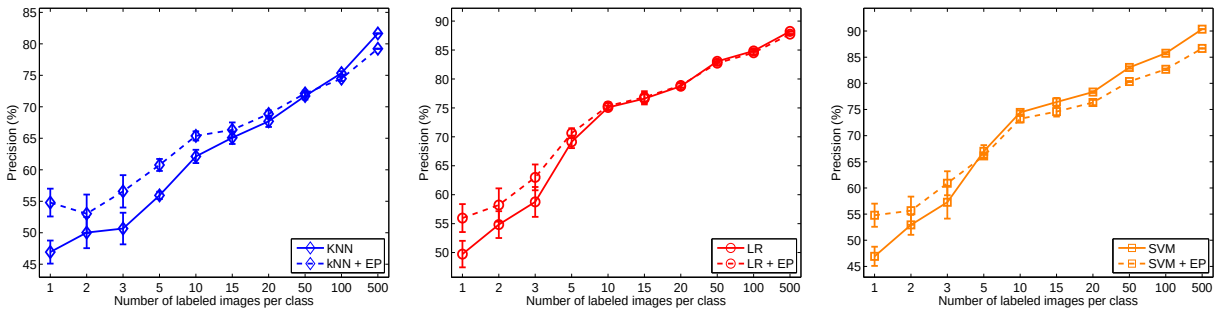
In this section, we evaluate the robustness of our learned features against classifier models. Different values of the balancing parameter C between model accuracy and model complexity were tested for the LR classifier across the eight datasets. 5 labeled training examples per class were used. A set of values $\{0.01, 0.1, 1, 5, 15, 50, 100\}$ were tested for the parameter C of LR. Figure 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. We have tried $k = 3$, and $k = 5$ for the k-NN method, and they yield similar results as $k = 1$. 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.

5.1.3. 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 Figure 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.

5.2. 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 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. Figure 8 shows the classification performance with different numbers of labeled training images per class. From the



(a) k-NN (b) LR (c) SVMs

Figure 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").

Methods	Scene-15	LandUse-21	Texture-25	Building-25	Event-8	Caltech-101	Indoor-67	STL-10	ImageNet-50
k-means	65.5	56.3	59.2	39.3	76.7	67.7	33.1	57.0	49.9
k-means + **EP**	<u>71.5</u>	<u>63.6</u>	**73.1**	**43.8**	**87.3**	<u>69.4</u>	<u>37.0</u>	<u>63.5</u>	**53.4**
Spectral Clustering	69.6	59.8	66.6	33.3	82.7	68.2	31.5	52.8	49.5
Spectral Clustering + **EP**	**73.6**	**65.2**	<u>70.1</u>	<u>41.0</u>	<u>86.5</u>	**70.7**	**37.2**	**66.4**	<u>53.3</u>

Table 3: Purity (%) of image clustering on the nine datasets, where the CNN feature [34] and our learned feature from it (indicated by + EP) are used. The best results are indicated in **bold**, and the second best is <u>underlined</u>.

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) [82]. 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.

5.3. 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 [23, 65, 24, 44, 83] and evaluate the task on the image classification datasets, in particular on the nine datasets used for semi-supervised image classification.

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 [84] for Spectral Clustering and the vl-feat library of [85] 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 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.

6. Conclusion and Discussion

This paper has tackled the problem of feature learning for the tasks of semi-supervised image classification and image clustering. We proposed a simple, yet effective feature learning method to exploit the available, unlabeled data. By using two consistency 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 nine 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.

Ensemble Projection (EP) learns knowledge by exploiting local neighbourhood structures. Local neighbourhood structures have been extensively exploited for feature embedding [86, 87, 21]. EP can be understood as a method for feature embedding, but rather in a discriminative manner. EP contracts the distance among images that are highly likely from the same class, and expands

the distance among images that are very likely from different classes. EP can also be understood from the perspective of curriculum learning [88]. EP takes the salient ('easy') image clusters as surrogate classes and learns a discriminative classifier to further group the rest of images. The grouping information is finally used for the task of imag classification and clustering. Providing theoretical evidences for EP and applying it to other vision tasks constitute our future work.

Acknowledgements. The work is supported by the ERC Advanced Grant Varcity (#273940) and the bilateral collaboration with Toyota.

References

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

[2] A. Quattoni, A. Torralba, Recognizing indoor scenes, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 413–420.

[3] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, Y. Gong, Locality-constrained linear coding for image classification, in: IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 3360–3367.

[4] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, A. Torralba, Sun database: Large-scale scene recognition from abbey to zoo., in: IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 3485–3492.

[5] M. Yang, D. Dai, L. Shen, L. Van Gool, Latent dictionary learning for sparse representation based classification, in: IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 4138–4145.

[6] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.

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

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

[9] A. Coates, A. Y. Ng, H. Lee, An analysis of single-layer networks in unsupervised feature learning, in: International Conference on Artificial Intelligence and Statistics, 2011, pp. 215–223.

[10] A. Dosovitskiy, J. T. Springenberg, M. Riedmiller, T. Brox, Discriminative unsupervised feature learning with convolutional neural networks, in: Advances in Neural Information Processing Systems, 2014, pp. 766–774.

[11] C. Doersch, A. Gupta, A. A. Efros, Unsupervised visual representation learning by context prediction, in: International Conference on Computer Vision, 2015, pp. 1422–1430.

[12] N. Srivastava, E. Mansimov, R. Salakhudinov, Unsupervised learning of video representations using lstms, in: International Conference on Machine Learning, 2015.

[13] R. Fergus, Y. Weiss, A. Torralba, Semi-supervised learning in gigantic image collections, in: Advances in Neural Information Processing Systems, 2009, pp. 522–530.

[14] M. Guillaumin, J. J. Verbeek, C. Schmid, Multimodal semi-supervised learning for image classification, in: IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 902–909.

[15] D. Dai, L. Van Gool, Ensemble projection for semi-supervised image classification, in: International Conference on Computer Vision, 2013, pp. 2072–2079.

[16] P. Jain, A. Kapoor, Active learning for large multi-class problems, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 762–769.

[17] A. J. Joshi, F. Porikli, N. Papanikolopoulos, Multi-class active learning for image classification, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 2372–2379.

[18] A. Quattoni, M. Collins, T. Darrell, Transfer learning for image classification with sparse prototype representations, in: IEEE Conference on Computer Vision and Pattern Recognition, 2008.

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

[20] A. Prest, C. Leistner, J. Civera, C. Schmid, V. Ferrari, Learning object class detectors from weakly annotated video, in: IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 3282–3289.

[21] D. Dai, T. Kroeger, R. Timofte, L. Van Gool, Metric imitation by manifold transfer for efficient vision applications, in: IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3527–3536.
[22] R. Raina, A. Battle, H. Lee, B. Packer, A. Y. Ng, Self-taught learning: transfer learning from unlabeled data, in: International Conference on Machine Learning, 2007, pp. 759–766.
[23] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, W. T. Freeman, Discovering objects and their location in images, in: International Conference on Computer Vision, 2005, pp. 370–377.
[24] D. Dai, T. Wu, S. C. Zhu, Discovering scene categories by information projection and cluster sampling, in: IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 483–496.
[25] K. P. Bennett, A. Demiriz, Semi-supervised support vector machines, in: Advances in Neural Information Processing Systems, 1998, pp. 368–374.
[26] T. Joachims, Transductive inference for text classification using support vector machines, in: International Conference on Machine Learning, 1999, pp. 200–209.
[27] P. Kumar Mallapragada, R. Jin, A. Jain, Y. Liu, Semiboost: Boosting for semi-supervised learning, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (11) (2009) 2000–2014.
[28] C. Leistner, A. Saffari, J. Santner, H. Bischof, Semi-supervised random forests, in: International Conference on Computer Vision, 2009, pp. 506–513.
[29] A. Oliva, A. Torralba, Modeling the shape of the scene: A holistic representation of the spatial envelope, International Journal of Computer Vision 42 (3) (2001) 145–175.
[30] T. Ojala, M. Pietikäinen, T. Mäenpää, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (7) (2002) 971–987.
[31] A. Bosch, A. Zisserman, X. Muoz, Image classification using random forests and ferns, in: International Conference on Computer Vision, 2007, pp. 1–8.
[32] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, T. Darrell, Decaf: A deep convolutional activation feature for generic visual recognition, in: International Conference on Machine Learning, 2014.
[33] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: IEEE Conference on Computer Vision and Pattern Recognition, 2014.
[34] K. Chatfield, K. Simonyan, A. Vedaldi, A. Zisserman, Return of the devil in the details: Delving deep into convolutional nets, in: British Machine Vision Conference, 2014.
[35] X. Wang, A. Gupta, Unsupervised learning of visual representations using videos, in: International Conference on Computer Vision, 2015, pp. 2794–2802.
[36] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, B. Schlkopf, Learning with local and global consistency, in: Advances in Neural Information Processing Systems, 2004, pp. 321–328.
[37] J. Weston, F. Ratle, R. Collobert, Deep learning via semi-supervised embedding, in: International Conference on Machine Learning, 2008, pp. 639–655.
[38] X. Zhu, A. B. Goldberg, Introduction to Semi-Supervised Learning, 2009.
[39] D. P. Kingma, S. Mohamed, D. J. Rezende, M. Welling, Semi-supervised learning with deep generative models, in: Advances in Neural Information Processing Systems, 2014, pp. 3581–3589.
[40] K. Grauman, T. Darrell, Unsupervised learning of categories from sets of partially matching image features, in: IEEE Conference on Computer Vision and Pattern Recognition, 2006, pp. 19–25.
[41] D. Dueck, B. J. Frey, Non-metric affinity propagation for unsupervised image categorization, in: International Conference on Computer Vision, 2007, pp. 1–8.
[42] E. Rosch, Principles of categorization, Cognition and Categorization (1978) 27–48.
[43] L. Rokach, Ensemble-based classifiers, Artificial Intelligence Review 33 (1-2) (2010) 1–39.
[44] D. Dai, M. Prasad, C. Leistner, L. Van Gool, Ensemble partitioning for unsupervised image categorization, in: European Conference on Computer Vision, 2012, pp. 483–496.
[45] T.-Y. Tang, H. Tzu-Wei, H.-T. Chen, Random exemplar hashing, in: DDS, 2013.
[46] A. Farhadi, I. Endres, D. Hoiem, D. Forsyth, Describing objects by their attributes, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009.
[47] L. Torresani, M. Szummer, A. Fitzgibbon, Efficient object category recognition using classemes, in: European Conference on Computer Vision, 2010, pp. 776–789.
[48] L.-J. Li, H. Su, E. P. Xing, F.-F. Li, Object bank: A high-level image representation for scene classification &

semantic feature sparsification, in: Advances in Neural Information Processing Systems, 2010, pp. 1378–1386.
[49] V. Sharmanska, N. Quadrianto, C. Lampert, Augmented attribute representations, in: European Conference on Computer Vision, 2012, pp. 242–255.
[50] F. X. Yu, L. Cao, R. S. Feris, J. R. Smith, S.-F. Chang, Designing category-level attributes for discriminative visual recognition, in: IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 771–778.
[51] M. Oquab, L. Bottou, I. Laptev, J. Sivic, Learning and transferring mid-level image representations using convolutional neural networks, in: IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1717–1724.
[52] J. Yosinski, J. Clune, Y. Bengio, H. Lipson, How transferable are features in deep neural networks?, in: Advances in Neural Information Processing Systems, 2014, pp. 3320–3328.
[53] P. Agrawal, J. Carreira, J. Malik, Learning to see by moving, in: International Conference on Computer Vision, 2015, pp. 37–45.
[54] S. Singh, A. Gupta, A. A. Efros, Unsupervised discovery of mid-level discriminative patches, in: European Conference on Computer Vision, 2012, pp. 73–86.
[55] M. Juneja, A. Vedaldi, C. Jawahar, A. Zisserman, Blocks that shout: Distinctive parts for scene classification, in: IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 923–930.
[56] O. Chapelle, B. Schölkopf, A. Zien (Eds.), Semi-Supervised Learning, MIT Press, Cambridge, MA, 2006.
[57] A. Blum, T. Mitchell, Combining labeled and unlabeled data with co-training, in: Proceedings of the Eleventh Annual Conference on Computational Learning Theory, COLT' 98, 1998, pp. 92–100.
[58] A. Shrivastava, S. Singh, A. Gupta, Constrained semi-supervised learning using attributes and comparative attributes, in: European Conference on Computer Vision, 2012, pp. 369–383.
[59] X. Zhu, Z. Ghahramani, J. Lafferty, Semi-supervised learning using gaussian fields and harmonic functions, in: International Conference on Machine Learning, 2003, pp. 912–919.
[60] M. Belkin, P. Niyogi, V. Sindhwani, Manifold regularization: A geometric framework for learning from labeled and unlabeled examples, JMLR 7 (36) (2006) 2399–2434.
[61] T. Kozakaya, S. Ito, S. Kubota, Random ensemble metrics for object recognition, in: International Conference on Computer Vision, 2011, pp. 1959–1966.
[62] R. Fergus, P. Perona, A. Zisserman, Object class recognition by unsupervised scale-invariant learning, in: IEEE Conference on Computer Vision and Pattern Recognition, 2003, pp. 264–271.
[63] J. Sivic, B. C. Russell, A. Zisserman, W. T. Freeman, A. A. Efros, Unsupervised discovery of visual object class hierarchies, in: IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.
[64] B. J. Frey, D. Dueck, Clustering by passing messages between data points., Science 315 (5814) (2007) 972–976.
[65] T. Tuytelaars, C. H. Lampert, M. B. Blaschko, W. Buntine, Unsupervised object discovery: A comparison, International Journal of Computer Vision 88 (2) (2010) 284–302.
[66] L.-J. Li, L. Fei-Fei, What, where and who? classifying event by scene and object recognition., in: International Conference on Computer Vision, 2007, pp. 1–8.
[67] Z.-H. Zhou, Ensemble Methods: Foundations and Algorithms, 2012.
[68] S. Lazebnik, C. Schmid, J. Ponce, A sparse texture representation using local affine regions, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (8) (2005) 1265–1278.
[69] L. Fei-Fei, R. Fergus, P. Perona, Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories., in: IEEE Conference on Computer Vision and Pattern Recognition, WS, 2004.
[70] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, ImageNet: A Large-Scale Hierarchical Image Database, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248–255.
[71] Z. Xu, D. Tao, Y. Zhang, J. Wu, A. Tsoi, Architectural style classification using multinomial latent logistic regression, in: European Conference on Computer Vision, 2014, pp. 600–615.
[72] Y. Yang, S. Newsam, Bag-of-visual-words and spatial extensions for land-use classification, in: International Conference on Advances in Geographic Information Systems, ACM, 2010, pp. 270–279.
[73] D. Dai, W. Yang, Satellite image classification via two-layer sparse coding with biased image representation, IEEE Geosci. Remote Sensing Lett. 8 (1) (2011) 173–176.
[74] A. Vedaldi, K. Lenc, Matconvnet – convolutional neural networks for matlab, in: Proceeding of the ACM Int.

Conf. on Multimedia, 2015.
[75] W. Liu, J. He, S.-F. Chang, Large graph construction for scalable semi-supervised learning, in: International Conference on Machine Learning, 2010, pp. 679–686.
[76] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, C.-J. Lin, Liblinear: A library for large linear classification, J. Mach. Learn. Res. 9 (2008) 1871–1874.
[77] C.-C. Chang, C.-J. Lin, LIBSVM: A library for support vector machines, ACM Transactions on Intelligent Systems and Technology 2 (2011) 1–27.
[78] L. Zelnik-Manor, P. Perona, Self-tuning spectral clustering, in: Advances in Neural Information Processing Systems, 2004, pp. 1601–1608.
[79] S. Ebert, D. Larlus, B. Schiele, Extracting structures in image collections for object recognition, in: European Conference on Computer Vision, 2010, pp. 720–733.
[80] N. Pitelis, C. Russell, L. Agapito, Semi-supervised learning using an unsupervised atlas, in: Machine Learning and Knowledge Discovery in Databases, Vol. 8725, 2014, pp. 565–580.
[81] M. Paulin, J. Revaud, Z. Harchaoui, F. Perronnin, C. Schmid, Transformation pursuit for image classification, in: IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 3646–3653.
[82] Y.-F. Li, Z.-H. Zhou, Towards making unlabeled data never hurt, in: International Conference on Machine Learning, 2011, pp. 175–188.
[83] A. Faktor, M. Irani, "clustering by composition" - unsupervised discovery of image categories, in: European Conference on Computer Vision, 2012, pp. 474–487.
[84] W.-Y. Chen, Y. Song, H. Bai, C.-J. Lin, E. Chang, Parallel spectral clustering in distributed systems, Pattern Analysis and Machine Intelligence, IEEE Transactions on 33 (3) (2011) 568–586.
[85] A. Vedaldi, B. Fulkerson, VLFeat: An open and portable library of computer vision algorithms, `http://www.vlfeat.org/` (2008).
[86] S. T. Roweis, L. K. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (5500) (2000) 2323–2326.
[87] X. He, D. Cai, S. Yan, H.-J. Zhang, Neighborhood preserving embedding, in: International Conference on Computer Vision (ICCV), Vol. 2, 2005, pp. 1208–1213.
[88] Y. Bengio, J. Louradour, R. Collobert, J. Weston, Curriculum learning, in: International Conference on Machine Learning (ICML), 2009, pp. 41–48.