Semi-Supervised Learning on a Budget: Scaling up to Large Datasets

Sandra Ebert, Mario Fritz, and Bernt Schiele

Max Planck Institute for Informatics
Saarbrucken, Germany

Abstract. Internet data sources provide us with large image datasets which are mostly without any explicit labeling. This setting is ideal for semi-supervised learning which seeks to exploit labeled data as well as a large pool of unlabeled data points to improve learning and classification. While we have made considerable progress on the theory and algorithms, we have seen limited success to translate such progress to the large scale datasets which these methods are inspired by. We investigate the computational complexity of popular graph-based semi-supervised learning algorithms together with different possible speed-ups. Our findings lead to a new algorithm that scales up to 40 times larger datasets in comparison to previous approaches and even increases the classification performance. Our method is based on the key insights that by employing a density-based measure unlabeled data points can be selected similar to an active learning scheme. This leads to a compact graph resulting in an improved performance up to 11.6% at reduced computational costs.

1 Introduction

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

Due to the active research on semi-supervised learning, the understanding of theory and algorithms in this area have greatly improved. One of the most promising frameworks is graph-based label propagation which leads to many insights [1] as well as high performance algorithms [2, 3]. However, those algorithms typical come with a quadratic complexity that is contradictory to the initial goal to scale up to large datasets. The "the-more-data-the-better" strategy that usually increases the performance of SSL [4] can often be not applied due to the prohibitive time and space complexity.

In this work, we question this strategy and show that we can indeed increase the performance with a more careful selection of *unlabeled* data. As a result we get similar or even better performance with only a fraction of all unlabeled data. This advantage becomes particularly evident when using large datasets like

ILSVRC 2010 with $1,000$ categories and more than a million images. In contrast to previous selection approaches [5, 6] that are only applicable to mid-sized data collections with up to $30,000$ data points, we are able to handle 40 times larger datasets. A further advantage of our selection method is that we can efficiently combine label propagation with active learning to further improve performance. In the context of active learning graph size plays a crucial role and thus our effective selection of unlabeled data becomes even more advantageous.

Contributions. First, we introduce two selection strategies (Sec. 3). We compare these criteria to previous methods and show on mid-sized datasets that we improve these approaches when we consider more realistic datasets with occlusions, truncations, and background clutter (Sec. 6). After that, we illustrate on a subset of ILSVRC 2010 with 100 classes that we get better performance when using only a representative subset of all images instead of all unlabeled data. We also show that our approach is able to process the entire ILSVRC 2010 dataset with $1,000$ classes and more than one million images. Finally, we conclude our work in Sec. 7 by applying graph propagation in combination with active learning resulting in increased performance.

2 Related Work

Large-scale computer vision has become more and more prominent in recent research. There is many work utilizing vast amount of images from the internet in order to improve one specific object category [7], to generate new datasets within an active learning framework [8], or to use it for image retrieval [9]. For image classification, ILSVRC 2010 [10] with $1,000$ classes and more than one million images is currently one of the most difficult datasets according to size and number of classes. Although, there are many approaches addressing this dataset most of them focus more on faster and better image description [11], analyze semantic similarities [12], or evaluate the scalability of knowledge transfer [13]. However, there are surprisingly few works that consider more advanced classification schemes beyond linear classifiers.

In contrast, semi-supervised learning (SSL) in particular graph-based methods are made to leverage labeled as well as unlabeled data to improve performance of classification. We observe a large progress towards algorithmic contributions [14, 15]. More recently, there is also a focus on improving graph construction – the most critical part of these algorithms. Previous works propose a better weighting function [16, 17], make use of discriminative algorithms like SVM [18], or remove noise of the data [1]. But although there is a common believe that more unlabeled data helps for learning, there is almost no work that address the scalability issue to take advantage of this huge available amount.

Main problem is that graph-based algorithms come with a quadratic runtime and space complexity. Previous work proposes methods to reduce the dimensionality of the used image descriptors [19], or classify with an approximation [20]. Other works reduce the amount of unlabeled data to approximate the distance matrix [21, 22], or to construct a smaller graph that represents the entire data

data distribution [5, 6, 23]. In this work, we build on this idea. But instead of representing the entire data space we focus on the data regions that are more relevant for our image classification task.

3 General SSL-framework

This section briefly introduces our SSL setup consisting of label propagation [14] extended with active learning [24] to further improve the performance.

3.1 Label propagation (LP)

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

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

We use an iterative procedure [14] to propagate labels through this graph

$$Y_m^{(t+1)} = \alpha \mathcal{S} Y_m^{(t)} + (1 - \alpha) Y_m^{(0)} \quad \text{with } 1 \leq m \leq c, \tag{2}$$

with Y_m^* the limit of this sequence. The initial label vector is set as follows $Y_m^{(0)} = (y_1^m, ..., y_l^m, 0, ..., 0)$ with $y_i^m \in \{1, -1\}$ for the labeled data and zero otherwise. Parameter $\alpha \in (0, 1]$ controls the overwriting of the original labels. Finally, the prediction of the data $\hat{Y} \in \mathcal{L}$ is obtained by $\hat{Y} = \text{argmax}_{1 \leq m \leq c} Y_m^*$.

3.2 Active Learning (AL)

Similar to [24], we combine uncertainty (exploitation) and density (exploration) criteria. For uncertainty, we use entropy over the class posterior $P(\tilde{y}_{ij}|x)$ by normalizing the prediction values from Eq. 2:

$$\mathcal{H}(x_i) = -\sum_{j=1}^{c} P(\tilde{y}_{ij}|x_i) \log P(\tilde{y}_{ij}|x_i). \tag{3}$$

For the density-based sampling, we employ the graph density criteria introduced by [24]. This criteria make use of the symmetric k-NN graph to find dense regions and is defined by the sum of all neighboring nodes divided by the number of neighbors

$$\mathcal{D}(x_i) = \frac{\sum_j W_{ij}}{\sum_j P_{ij}}, \tag{4}$$

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

4 Graph enhancement techniques

As motivated above, graph-based SSL-techniques are quadratic in the number of data samples. Therefore, we are interested in techniques that benefit from more unlabeled data while simultaneously minimizing the runtime. After reviewing previous techniques (Sec. 4.1) we propose two novel techniques (Sec. 4.2) that can be scaled to 40 times larger datasets than any previous techniques that we aware of due to their lower computational complexity (Sec. 4.3).

4.1 Previous techniques

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

There are several other approaches that propose a graph construction with a representative unlabeled subset called anchor graph. In [6] *k-means* cluster centroids are used as anchor points which can be advantageous when the clusters represent one class each. Otherwise they introduce many shortcuts between different classes. We show experimentally that *k-means* works well for datasets with a smooth manifold structure but fails for more difficult data collections.

The second approach [5] finds representative unlabeled data in a *greedy* fashion by repeatedly selecting the sample that is farthest from the current subset S consisting of the training set L and the already selected unlabeled data Z: $\arg\min_{j \in Z \setminus S} \sum_{i \in L \cup S} W_{ij}$, with W a similarity matrix for all images using L1 distance and a Gaussian weighting function. This method covers the entire data space without introducing redundant information and works well as long as there are not too many outliers in the data collection.

All methods aim to represent the entire unlabeled data space independently from the task itself. If the unlabeled data is representative for the test data as it is the case for ETH80 (Sec. 5), these methods work well. However, when the ratio between test samples and unlabeled data is very small as it is often the case for large datasets, these approaches fail to focus on the relevant part of the distribution thus not achieving optimal performance.

4.2 Novel techniques to enrich graph structure

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

We consider three scenarios for extension: 1) training set only; 2) test set only; and 3) training+test set. The first scenario leads only to local improvements because of the small amount of labeled data. Additionally, this approach becomes problematic if the neighborhoods around the labels are sparse as it leads to many false neighbors. Enhancing the area around the test data only improves the results for the same reasons. Experimentally, we observed that enriching the neighborhood of both training and test works best so that we report only results for this setting in the following.

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

$$Z_{pool} \leftarrow \{x_j\} \text{ with } x_j \text{ the k nearest neighbors of } x_i \in T. \tag{5}$$

We order these data points by their graph density $\mathcal{D}$ from Eq. 4

$$r(x_i) = m_i, \quad \text{where} \quad m_i \leq m_j \Leftrightarrow \mathcal{D}(x_i) \geq \mathcal{D}(x_j) \tag{6}$$

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

$$Z \leftarrow \{x_i\} \text{ where } r(x_i) \text{ belongs to the } |Z| \text{ smallest scores} \tag{7}$$

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

(ii) *NN.* Beside this positive behavior regarding our set T, this method still does not scale well to large datasets (see Sec. 4.3) as we have to calculate the entire distance matrix. For this reason, we propose a second criteria *NN* that can be seen as an approximation of *dense*. This selection technique needs only the distances between $x_i \in T$ to all unlabeled data $x_i \in U$ with $U = N \setminus T$ and all data N. Usually, we have $|T| \ll |U|$ so that the runtime is moderate. To enhance T, we select the first k nearest neighbors for each $x_i \in T$, i.e.,

$$Z_{pool} \leftarrow \{x_{i_k}\} \text{ with } x_{i_k} \text{ the k nearest neighbors of } x_i \in T \tag{8}$$

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

4.3 Runtime complexity

In the following, we briefly analyze the runtime complexity of all introduced graph enhancement techniques and then compare their runtime behavior in the context of label propagation (see Fig. 1). Given $|N| = |T| + |U|$ images with T the original dataset consisting of training and test set and U the pool of unexplored and unlabeled data. The runtime of *k-means* is directly linked to the number of clusters, i.e., $O(|Z||U|m)$ with $|Z|$ the number of anchor points ($\sim$ number of added data) and m the dimensionality of the image descriptor. With increasing unlabeled data volume, memory and and runtime requirements increase disproportionately as can be seen in Fig. 1 (left).

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

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

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

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

Fig. 1 visualizes on the left side the runtime of the several graph enhancement methods including the *random* baseline for the dataset IM100 introduced in the next section. This is a subset of ILSVRC 2010 with 100 classes and approx. $130,000$ images. We plot number of added images against the expected runtime.

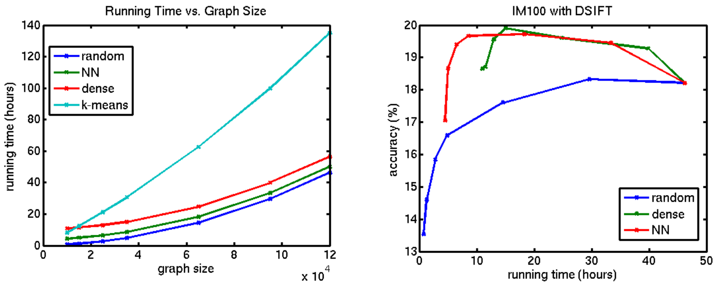

Fig. 1. Left: Complexity for selecting $|S|$ unlabeled data $x \in U$ with m dimensions of the image descriptor given a fixed training and test set $|T|$ and label propagation. Right: Complexity against performance of IM100 (see Sec. 5) for DSIFT.

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

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

On the right side of Fig. 1, we plot runtime against classification performance for the same dataset. *k-means* and *greedy* cannot be applied on this large unlabeled pool due to the runtime and space complexity. Most interestingly we see for a given time budget that we achieve better performance than *random*. For example if we look at 20 hours for *random* that corresponds to a graph size of $65,000$ images, we get a performance of 17.6%. In contrast, *dense* and *NN* need only a graph size of $25,000$ to get a higher performance with 19.9% and 19.6% respectively. This emphasizes our claim that we are not only faster but also obtain better performance with a more representative subset of the unlabeled data. Although ”the-more-data-the-better“ strategy actually leads to a mostly consistent improvement (blue curve) the final performance is clearly below the results achieved with our methods (red and green curve). This loss of performance is often a consequence of added images that connect many images from different classes bringing them mistakingly close together.

5 Dataset and image representation

In our experiments, we analyze four different datasets with increasing dataset size and number of classes. Example images are shown in Fig. 2. ETH80 [25]

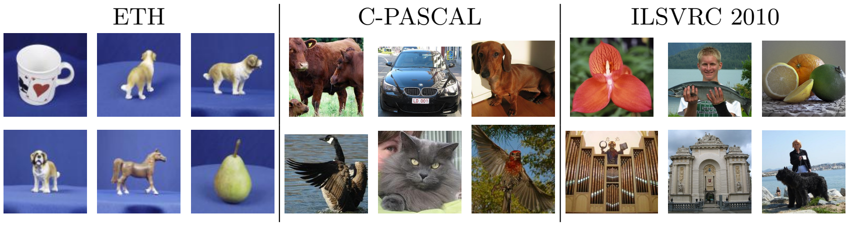

Fig. 2. Example images for ETH (left), C-PASCAL (middle), and ILSVRC 2010 (right)

contains 3,280 images divided in 8 object classes with 10 instances per class. Each instance is imaged from 41 viewpoints in front of a uniform background.

Cropped PASCAL (C-PASCAL) is introduced in [4]. Bounding box annotations of the PASCAL VOC challenge 2008 training set are used to extract the objects such that classification can be evaluated in a multi-class setting. In this paper, we use the larger PASCAL VOC challenge 2011 with 8,900 images of aligned objects from 20 classes but with varying object poses, challenging appearances, background clutter, and truncation.

IM100 is subset of the ILSVRC 2010 challenge one of the state-of-the-art datasets for large-scale image classification. IM100 contains 100 classes with approx. $130,000$ images. Finally, we show also results on ILSVRC 2010 with $1,000$ categories and approx. 1.26 million images. Objects can be anywhere in an image and images contain background clutter, occlusions, or truncations.

For all datasets, we evaluate three different image descriptors to show that our insights generalize to several settings. Gist (960 dimensions) is computed by using the code of [26]. Dense SIFT (DSIFT) and spatial dense SIFT (SpDSIFT) are extracted with the implementation VLFeat proposed by [27]. SIFT features are calculated on a regular grid and quantized into 1000 visual words. For SpDSIFT, we use a subdivision of 4×4 that are concatenated to a final histogram representation with $9,000$ dimensions.

6 Experiments

In our experiments, we select randomly 5 training samples and 45 test samples per class that serves as the original dataset T. This setting exactly corresponds to the classical semi-supervised setting with 10% labeled data [2, 16, 4]. The remaining images of these datasets are considered as the data pool U from which we select unlabeled data to enrich T. We run all experiments 5 times with 5 different sets T and evaluate the performance on the test set only. Therefore, we are able to compare our results independently from the amount of added data. In the following, we analyze each dataset separately.

ETH80. Fig. 3 shows for all three image descriptors graph quality (GQ, first row) and accuracy after label propagation (second row) without (solid lines) and with (dashed lines) active learning (AL). Graph quality denotes the average

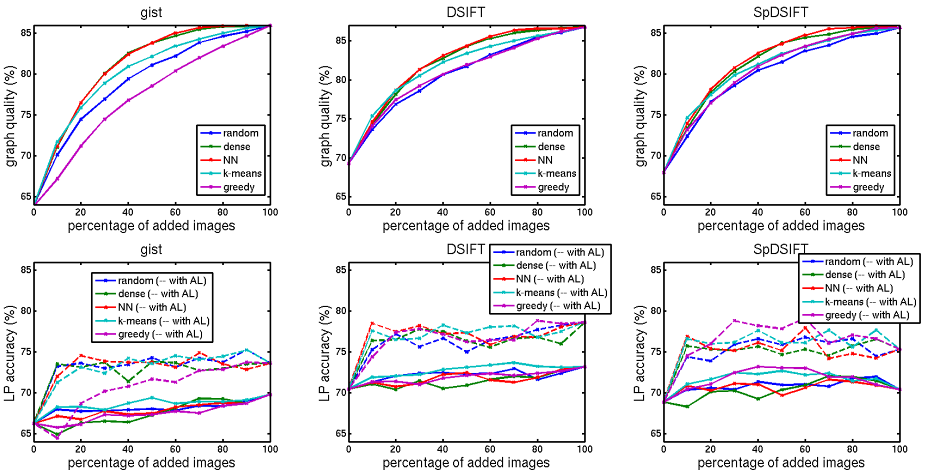

Fig. 3. Graph quality (first row) and LP accuracy (second row) for ETH80 with (dashed lines) and without (solid lines) active learning for different number of added images: Gist (left), dense SIFT (middle), and spatial dense SIFT (right)

number of correct nearest neighbors in our symmetric k-NN graph structure for the training and test data and serves only as a theoretic measure as we need to know all labels for this evaluation. For AL, we start with one training example per class randomly selected from our fixed training set of 5 samples per class, and request in average 4 labels per class from the remaining training set plus the additional unlabeled set.

We observe that the graph quality starts saturating after 60% to 70% added data. The performance of all selection methods is similar including the *random* baseline. This can be explained by the smooth manifold structure of the dataset. There are almost no outliers in this dataset so that our test set benefits from almost all images equally. For LP, we see a consistent improvement when active learning is used[1]. Tab. 1 shows graph quality (GQ) and accuracies with 50% ($\approx$ 1500) additional unlabeled data. For DSIFT with *NN* selection we improve LP without AL from 72.4% to 77.3% with AL. *k-means* performs slightly better for LP without AL. The cluster centers seem to be good anchor points for the test data. Our density selection criteria shows on average slightly worse performance for LP without AL probably due to the oversampling of dense regions (e.g. apples and tomatoes are high density regions which are preferred by this criteria).

C-PASCAL. This dataset corresponds to a more difficult classification problem with many outliers and overlapping classes. We observe for both GQ and LP (Fig. 4) a large performance gap between our selection methods and previous methods. For SpDSIFT and DSIFT, *k-means* and *greedy* are even worse than the random baseline, e.g., LP+AL decreases for SpDSIFT from 28.3% with *ran-*

[1] as this is true also for all other datasets we show only the performance for active learning in the following.

	Gist			DSIFT			SpDSIFT		
method	GQ	LP	+AL	GQ	LP	+AL	GQ	LP	+AL
random	81.5	68.0	**74.3**	82.1	72.33	75.0	82.3	70.9	75.9
dense	83.0	67.2	73.8	84.1	70.9	76.3	83.0	70.3	74.7
NN	**83.3**	67.4	73.7	**84.1**	72.4	**77.3**	**83.5**	69.7	75.2
k-means [6]	82.5	**69.4**	73.6	83.6	**73.1**	**77.3**	82.9	72.7	76.1
greedy [5]	78.1	67.3	71.7	81.7	72.1	76.2	82.2	**73.1**	**77.8**

Table 1. Graph quality (GQ) and LP accuracy without and with (+AL) active learning for ETH80 after adding 50% unlabeled images.

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

Table 2. Graph quality (GQ) and LP accuracy without and with (+AL) active learning for C-PASCAL after adding 50% unlabeled images.

dom to 20.1% with *k-means*, and to 23.5% with *greedy*. For *k-means*, this drop is a direct consequence of the used cluster centroids. Many clusters contain more than one class so that these clusters connect all examples of those classes and bring them closer together. In contrast, *greedy* focus more on outliers.

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

IM100. In the following, we analyze a subset of ILSVRC 2010 with approx. $130,000$ images. This subset is large enough to increase the amount of unlabeled data by a factor of 25 but also small enough to run SSL on the entire dataset. *k-means* and *greedy* cannot be applied to this dataset due to their time and space complexities (see Sec. 3). Similar to all previous subsections, we show GQ and LP+AL in Fig. 5 for different numbers of added data (graph size), and Tab. 3 contains results when adding 20% unlabeled data.

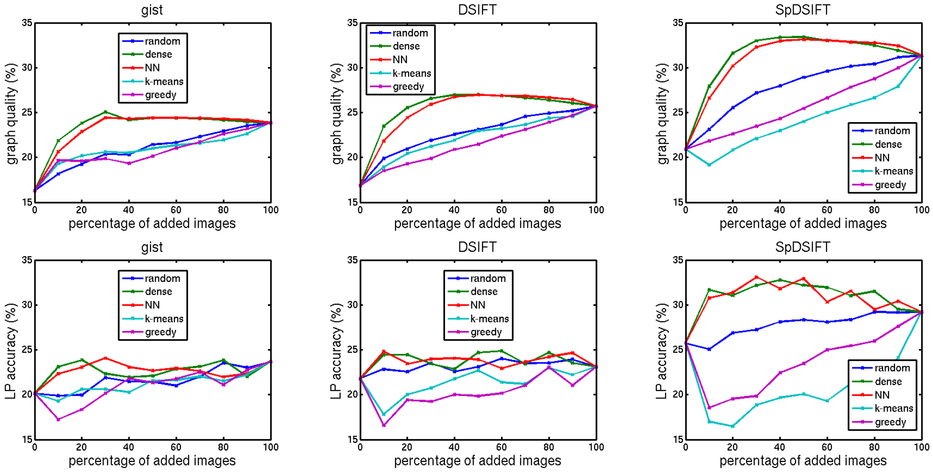

Fig. 4. Graph quality (first row) and LP accuracy (second row) for C-PASCAL with (dashed lines) and without (solid lines) active learning for different number of added images: Gist (left), dense SIFT (middle), and spatial dense SIFT (right)

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

Table 3. Graph quality (GQ) and LP accuracy without and with (+AL) active learning for IM100 after adding 30,000 unlabeled images (≈ 23%).

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

ILSVRC 2010. Finally, we run LP on the entire ILSVRC 2010 challenge with 1,000 classes. We start with our set T given by 5 training samples and 45 test sample per class, i.e., 50,000 images (Tab. 4, first line). After that, we continuously add 50,000 unlabeled data from the pool of the remaining 1.2 million images. Tab. 4 shows graph quality (GQ), top 1, and top 5 accuracy for LP+AL and the difference to *random* selection. For computational reason, we apply only *NN*. To further increase the speed of AL, we use batch active learning with a batch size of 100 labels per query. So that we request 400 times a batch of 100 labels to get in average 5 labels per class.

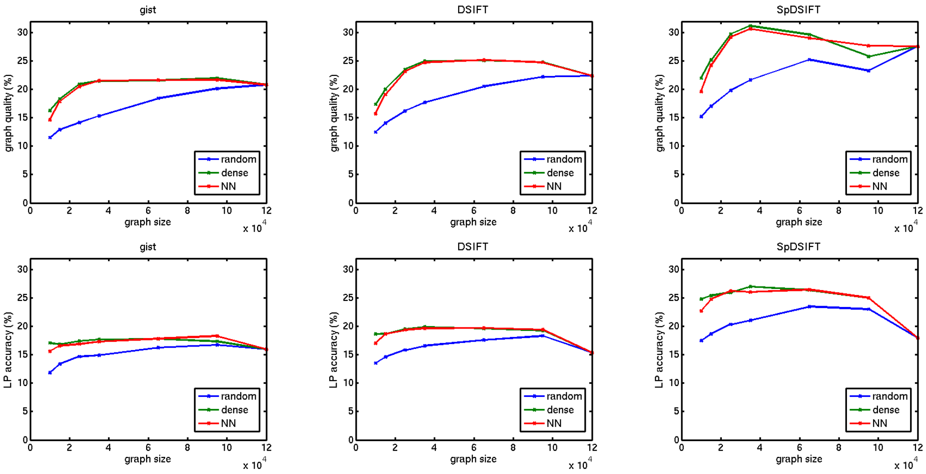

Fig. 5. Graph quality (first row) and LP accuracy (second row) for IM100 with active learning for different number of added images: Gist (left), dense SIFT (middle), and spatial dense SIFT (right)

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

Table 4. ILSVRC 2010 with *random* and *NN* enrichment for DSIFT: graph quality (GQ), top 1 and top 5 accuracy after LP with AL, and the difference to *random*.

For comparison, we run also a linear SVM on the base setting with $50,000$ images and with different parameters. The best performance we observe is 0.22% averaged over 5 different runs. In contrast with LP without enrichment we get 2.8% top 1 accuracy. This large difference can be explained by the additional graph structure we used in SSL. According to the selection criteria, we improve increasingly our graph quality (GQ). For $50,000$ additional unlabeled images we note a difference between *random* and *NN* of +1.7% while for $250,000$ added images this difference increase to +4.4%. We also observe an improvement for LP. For $150,000$ additional images, we increase LP from 4.2% with *random* to 5.5% with *NN*. However, LP benefits only limited from this improving structure. One explanation might be that we run a batch AL instead of a single AL. Usually these batch AL show worse performance in comparison to single AL.

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

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

7 Conclusion

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

Tab. 5 summarizes our main insights from this paper on the dataset IM100. First of all, we see a consistent improvement when adding more unlabeled data. For SpDSIFT, we increase from 14.7% to 16.6% with randomly added 25,000 unlabeled data points to finally 16.7% when adding all available data. But these results are clearly below the performance of 17.8% that we achieve with our novel criteria *NN*. This fact becomes even more obvious in combination with active learning where we improve SpDSIFT with our new criteria by 11.6% to 26.3% while we increase this performance only by 5.9% when applying "the-more-data-the-better" strategy.

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

References

1. Hein, M., Maier, M.: Manifold Denoising. In: NIPS. (2006)
2. Zhou, D., Huang, J.: Learning from Labeled and Unlabeled Data on a Directed Graph. In: ICML. (2005)

3. Liu, W., Chang, S.: Robust multi-class transductive learning with graphs. In: CVPR. (2009)
4. Ebert, S., Larlus, D., Schiele, B.: Extracting Structures in Image Collections for Object Recognition. In: ECCV. (2010)
5. Delalleau, O., Bengio, Y., Le Roux, N.: Efficient non-parametric function induction in semi-supervised learning. In: AISTATS. (2005)
6. Liu, W., He, J., Chang, S.: Large graph construction for scalable semi-supervised learning. In: ICML. (2010)
7. Schroff, F., Criminisi, A., Zisserman, A.: Harvesting Image Databases from the Web. In: ICCV. (2007)
8. Collins, B., Deng, J., Li, K., Fei-Fei, L.: Towards scalable dataset construction : An active learning approach. In: ECCV. (2008)
9. Kulis, B., Grauman, K.: Kernelized locality-sensitive hashing for scalable image search. In: ICCV. (2009)
10. Deng, J., Dong, W., Socher, R., Li-Jia Li, Li, K., Fei-Fei, L.: ImageNet: A large-scale hierarchical image database. In: CVPR. (2009)
11. Perronnin, F., Liu, Y., J: Large-scale image retrieval with compressed Fisher vectors. In: CVPR. (2010)
12. Deselaers, T., Ferrari, V.: Visual and Semantic Similarity in ImageNet. In: CVPR. (2011)
13. Rohrbach, M., Stark, M., Schiele, B.: Evaluating Knowledge Transfer and Zero-Shot Learning in a Large-Scale Setting. In: CVPR. (2011)
14. Zhou, D., Schölkopf, B., Bousquet, O., Lal, T.N., Weston, J.: Learning with Local and Global Consistency. In: NIPS. (2004)
15. Sindhwani, V., Niyogi, P., Belkin: Beyond the point cloud: from transductive to semi-supervised learning. ML (2005)
16. Zhu, X., Ghahramani, Z., Lafferty, J.: Semi-supervised learning using gaussian fields and harmonic functions. In: ICML. (2003)
17. Wang, F., Zhang, C.: Label propagation through linear neighborhoods. TKDE **1** (2007) 55–67
18. Zhang, Z., Wang, J., Zha, H.: Adaptive Manifold Learning. TPAMI (2011) 1–14
19. Torralba, A., Fergus, R., Weiss, Y.: Small codes and large image databases for recognition. In: CVPR. (2008)
20. Fergus, R., Weiss, Y., Torralba, A.: Semi-supervised learning in gigantic image collections. In: NIPS. (2009)
21. Zhang, Z., Zha, H., Zhang, M., Tech, G.: Spectral Methods for Semi-supervised Manifold Learning. In: CVPR. (2008)
22. Zhang, K., Kwok, J.T., Parvin, B.: Prototype vector machine for large scale semi-supervised learning. In: ICML. (2009)
23. Li, Y.F., Zhou, Z.H.: Towards Making Unlabeled Data Never Hurt. In: ICML. (2011)
24. Ebert, S., Fritz, M., Schiele, B.: Reinforced Active Learning: An Object Class Learning-By-Doing Approach. In: CVPR. (2012)
25. Leibe, B., Schiele, B.: Analyzing Appearance and Contour Based Methods for Object Categorization. In: CVPR. (2003)
26. Oliva, A., Torralba, A.: Modeling the shape of the scene: A holistic representation of the spatial envelope. IJCV (2001)
27. Vedaldi, A., Fulkerson, B.: VLFEAT: An Open and Portable Library of Computer Vision Algorithms (2008)