Scalable Multitask Representation Learning for Scene Classification

Maksim Lapin, Bernt Schiele
Max Planck Institute for Informatics
{mlapin,schiele}@mpi-inf.mpg.de

Matthias Hein
Saarland University
hein@cs.uni-saarland.de

Abstract

The underlying idea of multitask learning is that learning tasks jointly is better than learning each task individually. In particular, if only a few training examples are available for each task, sharing a jointly trained representation improves classification performance. In this paper, we propose a novel multitask learning method that learns a low-dimensional representation jointly with the corresponding classifiers, which are then able to profit from the latent inter-class correlations. Our method scales with respect to the original feature dimension and can be used with high-dimensional image descriptors such as the Fisher Vector. Furthermore, it consistently outperforms the current state of the art on the SUN397 scene classification benchmark with varying amounts of training data.

1. Introduction

It is often observed that the best performance is obtained with high dimensional image descriptors [3]. Recent work [15] showed that feature encoding based on the Fisher Kernel [8], which combines the benefits of generative and discriminative approaches, yields significantly better results when compared to the standard bag of visual (BoV) words model on a number of computer vision benchmarks. The Fisher Vector (FV) encoding was also shown recently to be crucial in achieving state of the art performance on the MIT Scene 67 dataset [9]. However, as discussed in [15], one of the biggest disadvantages of the FV compared to the BoV is that the FV is dense. When combined with high dimensionality (in our experiments we work with over 260K dimensional features), this obviously presents a scalability problem not only storage-, but also computation-wise.

The de facto standard to address image, object or scene classification today is to train separate classifiers in a one-vs-all regime. On the other hand, it has long been argued that co-learning of class representations and transferring knowledge across classes is a key-ingredient in scaling to a large number of categories as well as in learning from a small number of training examples per class. This calls for a multitask learning framework where each binary classifier becomes a separate task and all classifiers as well as the representation are learned jointly. In contrast to the independent one-vs-all training, this enables the classifiers to exploit potential inter-class correlations.

While there has been significant progress in the area of multitask learning in the last decade [1, 2, 13, 10, 14] both on the theoretical as well as the algorithmic side, most of the proposed methods do not scale well to very large feature dimensions. Thus, in order to enable inter class transfer for such high-dimensional representations which are encountered in computer vision problems today, we propose, as our first main contribution, a new scalable formulation of multitask representation learning. It jointly learns a linear mapping into a lower dimensional space which is then used to build the classifiers for each class. In order to deal with the resulting large scale optimization problem, we adapt the recently developed stochastic dual coordinate ascent (SDCA) method [16, 17, 18]. The SDCA algorithm can be applied to smooth as well as Lipschitz losses (*e.g.* the hinge loss), it has a clean stopping criterion (the duality gap) and fast convergence rate which is superior to that of the vanilla stochastic gradient descent. It is also important that the method operates on dual variables, because the number of training examples in our setting is much smaller than the feature dimension. Our adaptation is very efficient as variable updates for the hinge loss can be computed in closed form.

There is also a connection between multitask learning [13] and supervised dictionary learning [12]. However, dictionary learning is additionally aiming at optimizing the reconstruction error, while our primary goal is to find a new representation where the classes are well separated.

Apart from our framework for multitask representation learning (MTL-SDCA), we apply, as a second main contribution, our novel approach on the challenging scene classification SUN397 benchmark [21]. An important ingredient for the best performance on this dataset is the high dimensional Fisher Vector encoding which achieves excellent performance even with a *single* image descriptor (SIFT). The latter is obtained when the feature extraction pipeline is carefully designed following the current best practices

[3, 9, 15]. With our novel approach we improve upon the state of the art thus validating both efficiency as well as effectiveness of our MTL-SDCA method. Furthermore, we validate that the approach performs well even in the case when only little training data is available, as it is expected from a multitask learning method.

2. Multitask Representation Learning

In this section we introduce the multitask representation learning framework and discuss its scalable solution via stochastic dual coordinate ascent (SDCA) methods. We discuss a general multitask setting first, even though we later specialize to a multiclass classification problem where we jointly learn the representation and the classifiers.

We first fix some notation and then introduce the problem. Let $\{(x_i, y_{ti}) : 1 \leq t \leq T, 1 \leq i \leq n\}$ be the input/output pairs of the multitask learning problem, where $x_i \in \mathbb{R}^d$, $y_{ti} \in \{\pm 1\}$, T is the number of tasks, and n is the number of training examples per task. We assume that all tasks have the same training examples even though this can be easily generalized. The setting we have in mind is that the feature space is high dimensional, which is quite common in computer vision problems, *e.g.* one has $d \geq 10^5$ with the Fisher Vector encoding [15], see Section 4. We learn a matrix U in $\mathbb{R}^{d\times k}$ with $k \ll d$ which is used to generate the new low dimensional representation of the data $z_i = U^\top x_i$. Moreover, we learn linear predictors w_t in $\mathbb{R}^k$ that operate on the lower dimensional data representation. Let X in $\mathbb{R}^{d\times n}$ be the matrix of stacked feature vectors x_i, W in $\mathbb{R}^{k\times T}$ the matrix of stacked predictors w_t, $K = X^\top X$ the Gram matrix, and $M = W^\top W$.

Multitask Representation Learning: We formulate multitask representation learning as the following optimization problem:

$$\min_{U\in\mathbb{R}^{d\times k}} \frac{1}{T}\sum_{t=1}^{T} \min_{w_t\in\mathbb{R}^k} P_{U,t}(w_t) + \frac{\mu}{2}\|U\|_F^2\,, \qquad (1)$$

where the objective for task t given a fixed U is

$$P_{U,t}(w_t) = \frac{1}{n}\sum_{i=1}^{n} \ell_{ti}\left(\langle w_t, U^\top x_i\rangle\right) + \frac{\lambda}{2}\|w_t\|_2^2\,, \qquad (2)$$

and $\lambda > 0$, $\mu > 0$ are the regularization parameters, ℓ_{ti} is a convex margin-based loss function (*e.g.* hinge loss, that is $\ell_{ti}(a) := \max(0, 1 - y_{ti}a)$ for $a \in \mathbb{R}$), and $\|\cdot\|_F$ denotes the Frobenius norm. We keep the general notation for the loss function but fix it to be the hinge loss in the following.

The inner subproblems are standard independent one-vs-all SVMs trained in a *lower dimensional subspace* which is determined by the matrix $U \in \mathbb{R}^{d\times k}$. The latter is learned jointly for all $t = 1, \ldots, T$ which facilitates knowledge transfer across the tasks. This is of particular interest when the amount of training examples per task is limited and at least some of the tasks are related.

Algorithm 1 MTL-SDCA

Input: data $\{(x_i, y_{ti})\}$, initial $U^{(0)}$, parameters λ, μ, ϵ
Let: $W^{(0)} = \mathbf{0}$.
repeat $\{s = 1, \ldots\}$
  for $t = 1$ **to** T **do**
    Let $w_t^{(s)} = \arg\min_w P_{U,t}(w)$, see (2)
      (via SDCA [16] on $\{({U^{(s-1)}}^\top x_i, y_{ti})\}$)
  end for
  Let $U^{(s)} = \arg\min_U P_W(U)$, see (4)
    (via SDCA with updates (7))
until change in variables is below ϵ

Let us discuss the relation to multitask feature learning formulation proposed in [1]:

$$\min_{\substack{U\in\mathbb{R}^{d\times d}, UU^T=\mathbf{I}\\ W\in\mathbb{R}^{d\times T}}} \sum_{t=1}^{T}\sum_{i=1}^{n} \ell_{ti}\left(\langle w_t, U^\top x_i\rangle\right) + \gamma\|W\|_{2,1}^2\,,$$

where $\|W\|_{2,1}^2 = \sum_{i=1}^{d} \left\|w_{(i)}\right\|_2$ and $w_{(i)} \in \mathbb{R}^T$ are the rows of W. The key difference is that we work with a low-dimensional representation $U \in \mathbb{R}^{d\times k}$ whereas the method above works with a square matrix $U \in \mathbb{R}^{d\times d}$ and enforces certain features to be discarded via the sparsity inducing penalizer $\|W\|_{2,1}^2$, which also couples the tasks. While [1] has the strong theoretical guarantee of convergence to the global optimum, they are by construction not able to scale to a high dimensional feature representation since $U \in \mathbb{R}^{d\times d}$ is a square, dense matrix and thus requires $O(d^2)$ memory. Our approach is scalable as our matrix requires only $O(k\,d)$ memory and $k \ll d$. Moreover, we enforce the coupling of the tasks directly by requiring that U maps to a low-dimensional subspace. Thus we do not need to additionally enforce the coupling of the tasks via a sparsity enforcing regularizer on the predictors w_t. This allows us to formulate the optimization problem in a way that it reduces to standard multiclass SVM when $\mu = 0$, which is not possible in the framework of [1].

2.1. MTL-SDCA Algorithm

The optimization problem (1) of our multitask representation learning framework is biconvex, that means it is convex in $W \in \mathbb{R}^{k\times T}$ for fixed U and vice versa. It is not jointly convex in U and W. This is common to most multitask formulations and the standard optimization method is block coordinate descent (see Algorithm 4.1 in [7]), that is we alternate between fixing U and optimizing W and then fixing W and optimizing U. Each subproblem is convex and one achieves monotonic descent in each iteration. This

guarantees (sub)-convergence to a critical point of the objective of (1), see [7], which is the standard convergence result for non-convex problems.

We solve both convex subproblems via two variants of stochastic dual coordinate ascent (SDCA) [18, 17], which is currently among the state of the art methods in large scale optimization. The final algorithm **MTL-SDCA** for multi-task representation learning is summarized in Algorithm 1.

The scalability of our approach crucially depends on the algorithm for learning $U \in \mathbb{R}^{d\times k}$. The choice of an algorithm that solves the *dual* problem is primarily motivated by our experimental setting for the SUN scene classification dataset. We use dense high dimensional feature vectors with the number of dimensions d being an order of magnitude larger than the number of training examples n, thus making dual optimization a natural choice.

For simplicity, we describe the algorithm in terms of primal variables U and W. However, to be computationally efficient, our implementation works only with the corresponding dual variables α and precomputed kernel matrices K and M, which in our setting fit into memory. The actual U and W are never computed at any stage. Further details can be found in the supplementary material at our website.

Learning W: Note that learning the predictor matrix $W \in \mathbb{R}^{k\times T}$ when U is fixed is the easier subproblem as the problems for each task decouple. Thus they can be trained in parallel via any solver for a standard SVM and the choice of SDCA here is more a matter of convenience.

Learning U: We show how the matrix U can be learned efficiently via an adaptation of the SDCA algorithm [16].

Let W be fixed. Then problem (1) reduces to

$$\min_{U\in\mathbb{R}^{d\times k}} P_W(U), \tag{3}$$

$$P_W(U) = \frac{1}{nT}\sum_{i,t}\ell_{ti}\left(\langle w_t, U^\top x_i\rangle\right) + \frac{\mu}{2}\|U\|_F^2. \tag{4}$$

The analogy to the SVM now comes from the fact that

$$\langle w_t, U^\top x_i\rangle = \langle U, x_i w_t^\top\rangle,$$

and thus we can see U as the weight vector of an SVM for the feature representation $x_i w_t^\top$. Moreover, note that the Frobenius norm of U is nothing else but the Euclidean norm of the matrix U rearranged as a vector. However, as this correspondence might not be obvious, we give the explicit details of the SDCA for this case.

The Fenchel dual problem of (3) is

$$\max_{\alpha\in\mathbb{R}^{T\times n}} D_W(\alpha), \tag{5}$$

$$D_W(\alpha) = \frac{1}{nT}\sum_{i,t}-\ell_{ti}^*(-\alpha_{ti}) - \frac{\mu}{2}\left\|\frac{1}{\mu nT}\sum_{i,t}\alpha_{ti}x_i w_t^\top\right\|_F^2,$$

where ℓ_{ti}^* is the convex conjugate of ℓ_{ti}, e.g. for the hinge loss one has

$$\ell_{ti}^*(-b) = \begin{cases} -y_{ti}b & 0\le y_{ti}b\le 1,\\ \infty & \text{otherwise.}\end{cases}$$

Let

$$U(\alpha) = \frac{1}{\mu nT}\sum_{i,t}\alpha_{ti}x_i w_t^\top, \tag{6}$$

then it is well known that $U^* = U(\alpha^*)$, where U^* is the solution of the primal problem (3) and α^* is the solution of the dual problem (5).

The dual problem is solved via SDCA. At every step s, an i in $\{1,\dots,n\}$ and a t in $\{1,\dots,T\}$ are chosen uniformly at random and an update of $\alpha_{ti}^{(s)}$ is computed as

$$\alpha_{ti}^{(s)} = \alpha_{ti}^{(s-1)} + \Delta\alpha_{ti},$$

where $\Delta\alpha_{ti}$ is the stepsize chosen to achieve maximal ascent of the dual objective $D_W(\alpha)$ when all other variables are fixed. To achieve maximal dual ascent one has to solve

$$\Delta\alpha_{ti} = \arg\max_{a\in\mathbb{R}} -\ell_{ti}^*\big(-(\alpha_{ti}^{(s-1)}+a)\big) - a\big\langle U(\alpha^{(s-1)}), x_i w_t^\top\big\rangle - \frac{a^2}{2\mu nT}\left\|x_i w_t^\top\right\|_F^2,$$

which for the hinge loss can be computed in closed form.

Efficient updates of α: In analogy to [17], we provide a *closed form solution* for $\Delta\alpha_i$ when $\ell_{ti}(a) = \phi_\gamma(y_{ti}a)$ is the smooth hinge loss, where ϕ_γ is defined as in [17]:

$$\phi_\gamma(a) = \begin{cases} 0 & a\ge 1,\\ 1-a-\gamma/2 & a\le 1-\gamma,\\ \frac{1}{2\gamma}(1-a)^2 & \text{otherwise.}\end{cases}$$

In our experiments, we always set $\gamma = 0$ which recovers the standard (non-smooth) hinge loss used in the SVM. The formula for the update $\Delta\alpha_{ti}$ is given below:

$$\Delta\alpha_{ti} = y_{ti}\max\left(-y_{ti}\alpha_{ti}^{(s-1)}, \min\left(1-y_{ti}\alpha_{ti}^{(s-1)}, \frac{1-y_{ti}x_i^\top U(\alpha^{(s-1)})w_t - \gamma y_{ti}\alpha_{ti}^{(s-1)}}{\frac{1}{\mu nT}\|x_i\|_2^2\|w_t\|_2^2+\gamma}\right)\right). \tag{7}$$

Note that the norms $\|x_i\|_2^2$ and $\|w_t\|_2^2$ are directly available from the precomputed kernel matrices K and M, and the inner product $x_i^\top U(\alpha^{(s-1)})w_t$ can be computed via (6). For details, see the supplementary material.

Initialization: In all our experiments we fix $k = T$. This choice is motivated by a two layer architecture, where the output of all one-vs-all SVMs is fed into a second layer of

one-vs-all classifiers. For this particular case there is a natural initialization for U, namely we can use $U^{(0)} = W_{\text{SVM}}$, where $W_{\text{SVM}} \in \mathbb{R}^{d \times T}$ is the matrix of stacked predictors w_t, which have been trained in the original feature representation. This choice is used in all experiments.

Stopping criterion: We use the relative duality gap defined as $(P(U(\alpha)) - D(\alpha))/\max(|P(U(\alpha))|, |D(\alpha)|)$ with $\epsilon = 10^{-3}$ as stopping criterion for both subproblems of learning W and U. In the master problem, we stop when the change in dual variables of the two subproblems is below $\epsilon = 10^{-3}$ as measured by the root-mean-square error (RMSE) defined as RMSE(Δ) = $\sqrt{(\sum_{i=1}^{m} \Delta_i^2)/m}$.

2.2. MTL-SDCA Extensions

Our method fully benefits from the generality of the SDCA framework which can be applied to different loss functions and different regularizers [16, 17]. We discuss a few examples below.

Other scalar losses: The method can be directly applied to other loss functions, *e.g.* the squared loss, for which a closed form solution for updates can also be derived. Even if there is no closed form solution (*e.g.* there is none for the logistic loss), then $\Delta\alpha_i$ can still be computed via a few iterations of the Newton method.

Other regularizers: Another straightforward generalization would be the introduction of ℓ_1/ℓ_2 regularization, also known as the elastic net [22], which would require keeping a copy of the primal variable and performing standard ℓ_1 shrinkage after every update.

Structured losses: Finally, the SDCA method can be also applied to structured loss functions. A classic example would be the loss used *e.g.* in Latent SVMs [6].

3. Initial Experiments

Before discussing the scene classification problem for the high-dimensional SUN397 dataset in Section 4, we begin with a first set of experiments on two datasets where a direct comparison to other methods is possible. Our algorithm is compared against two baselines: *(a)* the multitask feature learning method of Kang *et al.* [10] that outperformed a number of multitask baselines on the data used here (see Table 2 in [10]), and *(b)* a single task learning approach termed **STL-SDCA**. The latter corresponds to the standard one-vs-all technique where the binary SVMs are trained via SDCA [16]. The main purpose of the experiment is two-fold. First, we want to compare to a state of the art multitask learning algorithm, and second, we want to experimentally verify that the shared multitask representation learned within our framework and given by $U^\top x$ can be superior to single task learning in the original feature space.

We use two handwritten digit recognition datasets (subsets of USPS and MNIST) provided by Kang *et al.* [10] and follow their evaluation protocol: parameters are tuned on a validation set, which is not used for training, and performance is evaluated on a fixed test set of 500 examples. Training and validation subsets are sampled randomly 5 times from a fixed set of 1500 examples. Results are reported in Table 1.

On the USPS dataset (upper part of Table 1), single task and multitask learning algorithms perform on par when 100 training examples are used per class (Kang *et al.* outperforms our methods in this case). When the amount of training data is successively reduced from 100, over 50, 20, 10, to 5 examples, the performance of the STL-SDCA as well as our MTL-SDCA approach decreases as expected. However, the advantage of the multitask algorithm (MTL-SDCA) w.r.t. to the single task version (STL-SDCA) becomes evident as it consistently improves performance. Similarly, we observe improvement for our MTL-SDCA approach on the MNIST dataset, where our methods also outperform the multitask algorithm of Kang *et al.* [10].

Discussion: The results suggest that our approach is competitive with the state of the art multitask method of Kang *et al.* The benefit of multitask learning is more pronounced on smaller training sets, which agrees both with the general intuition behind multitask learning and the related theoretical results of Maurer *et al.* [13].

4. SUN397 Experiments

This section reports our main experimental results on SUN397 [21] which is a challenging scene classification benchmark containing over 100K images of 397 categories. We also discuss important implementation details that lead to the state of the art performance on this dataset.

4.1. Experimental Setup

We follow the protocol of Xiao *et al.* [21] and use 5, 10, 20, and 50 images per class for training and 50 images per class for testing. We use the 10 splits provided on the website of the dataset[1] and measure top-1 recognition accuracy. We report mean accuracy and standard deviation over the 10 splits. We treat every training subset in each split as an independent dataset and run the whole experimental pipeline (including feature extraction, codebook learning, model selection, *etc.*) on each of them separately.

Our feature extraction pipeline follows closely the one described in [15]. Images are resized to 100K pixels if larger and approximately 10K descriptors are extracted per image from 24×24 patches on a regular grid every 4 pixels at 5 scales $2^{-2:.5:0}$. We use 128-dim SIFT descriptors of Lowe [11] and 96-dim Local Color Statistic (LCS) descriptors of Clinchant *et al.* [4].

The descriptors are processed by PCA as discussed below and we use on the order of 10^6 descriptors to learn the

[1] http://people.csail.mit.edu/jxiao/SUN/

Method	Dataset	Ntrain=5	Ntrain=10	Ntrain=20	Ntrain=50	Ntrain=100
Kang *et al.* [10]						**91.6 (0.3)**
STL-SDCA	USPS [10]	69.4 (0.6)	76.3 (1.1)	83.7 (0.2)	88.5 (0.5)	90.8 (0.3)
MTL-SDCA		**71.4 (0.7)**	**77.2 (0.5)**	**84.6 (0.4)**	**90.0 (0.5)**	90.6 (0.2)
Kang *et al.* [10]						84.8 (0.3)
STL-SDCA	MNIST [10]	65.6 (0.7)	73.6 (0.8)	**79.8 (1.0)**	83.1 (0.6)	85.7 (0.4)
MTL-SDCA		**66.2 (0.7)**	**74.0 (1.0)**	79.7 (0.9)	**83.4 (0.6)**	**86.0 (0.2)**

Table 1: Mean accuracy (%) across 5 splits on two handwritten digit recognition datasets (numbers in parenthesis show standard deviation scaled by $1/\sqrt{5}$ as reported in [10]). **Ntrain** - number of training examples per class. The original images were preprocessed with PCA reducing dimensionality to $d = 87$ (USPS) and $d = 64$ (MNIST) retaining 95% of the variance.

				Ntrain=5			Ntrain=10			Ntrain=20			Ntrain=50		
LCS	PN	L2	PCA	Lin	Sqr	Chi	Lin	Sqr	Chi	Lin	Sqr	Chi	Lin	Sqr	Chi
			64	18.5	20.6	20.8	26.0	28.8	28.8	33.6	35.8	36.0	43.2	45.1	45.7
.			64	18.6	20.8	20.8	27.0	29.3	29.1	35.2	37.6	37.5	45.0	47.2	47.2
.	.		64	18.6	20.5	20.6	27.2	29.2	29.3	35.3	37.3	37.4	45.0	47.4	47.3
.		.	64	18.2	19.4	19.5	26.4	28.8	28.7	36.7	**39.2**	39.1	44.0	45.9	46.0
.	.	.	64	18.5	20.4	20.3	26.7	29.1	29.2	34.1	36.8	36.7	44.4	46.1	46.1
.			128	19.0	21.4	21.4	26.9	29.6	29.8	35.8	39.1	39.0	44.7	47.4	47.6
.	.		128	18.6	21.0	21.1	26.8	29.5	29.5	35.3	38.3	38.0	44.3	47.0	47.2
.		.	128	19.5	21.8	21.8	28.0	30.6	30.8	35.9	38.2	38.2	45.8	48.0	48.3
.	.	.	128	20.0	22.3	**22.5**	28.5	**31.2**	**31.2**	36.1	38.6	38.6	46.2	48.3	**48.4**

Table 2: STL-SDCA accuracy (%) on the first split of the SUN397 [21] dataset. **LCS** - Local Color Statistic descriptor; **PN** - LCS with power normalization; **L2** - LCS with ℓ_2-normalization; **PCA** - independent PCA of SIFT and LCS to 64-dim each, or joint PCA to 128-dim; **Lin/Sqr/Chi** - linear/Hellinger/χ^2 kernel; **Ntrain** - number of training examples per class.

PCA projections. Finally, descriptors are encoded via the Fisher Vector (FV) and pooled over a spatial pyramid with 4 regions (the entire image and three horizontal stripes). The codebook for FV is given by a GMM with 256 Gaussians learned via the EM algorithm. This yields the following feature dimensions of the final descriptor: $d = 131,072$ (SIFT) and $d = 262,144$ (SIFT+LCS).

We use VLFeat 0.9.17 [20] and our feature extraction scripts are based on the code provided with the library (located in `apps/recognition`). This way we follow currently established best practices in image classification.

Both STL-SDCA and MTL-SDCA solvers were implemented in C++ and are available at our website.

4.2. Feature Implementation Details

This section explores the impact of several implementation details on the final performance. Sánchez *et al.* [15] provide extensive evaluation of the effects of PCA, ℓ_2-normalization, power normalization ($\text{sign}(z)\,|z|^\rho$, $0 < \rho \leq 1$), and other parameters on the PASCAL VOC 2007 dataset. While their findings suggest that these details have significant effect on the final performance, a similar evaluation was not done on SUN397 and it is also not clear which options, in particular for the LCS descriptor, would perform best. We aim to fill this gap in this section.

To save computation time and to avoid overfitting to other splits, we perform all experiments in this section on the first split only. We set the SVM parameter C by 2-fold cross-validation and retrain models on the full training subsets. Results are summarized in Table 2.

Impact of PCA: When both SIFT and LCS descriptors are used, there are two ways to perform PCA preprocessing: reduce each descriptor to 64-dim independently and then concatenate, or perform a joint PCA reducing the combined descriptor to 128-dim. We observe that performing PCA jointly is generally better (except for the Ntrain=20 setting) and we use this strategy in our further experiments.

Impact of power normalization: We observe that performing power normalization with $\rho = 0.5$ (a.k.a. "square rooting") on the LCS descriptor improves classification performance when it is combined with the ℓ_2-normalization and joint PCA. This setting yields the best accuracy.

Impact of ℓ_2-normalization: The results for ℓ_2-normalization seem to depend on the way PCA preprocessing is done and generally improve performance when dimensionality reduction is performed jointly.

Method	Features	Ntrain=5	Ntrain=10	Ntrain=20	Ntrain=50
Xiao *et al*. [21]	12 combined	14.5	20.9	28.1	38.0
Su and Jurie [19]	Context+Semantic				35.6 (0.4)
Donahue *et al*. [5]	$DeCAF_6$				40.9 (0.3)
Sánchez *et al*. [15]	SIFT	19.2 (0.4)	26.6 (0.4)	34.2 (0.3)	43.3 (0.2)
STL-SDCA, Lin	SIFT	17.4 (1.5)	25.8 (0.2)	33.6 (0.3)	43.2 (0.2)
STL-SDCA, Sqr	SIFT	20.4 (0.3)	28.2 (0.3)	35.9 (0.3)	45.1 (0.3)
STL-SDCA-Stacked, Sqr	SIFT	20.6 (0.4)	28.4 (0.3)	36.1 (0.3)	45.3 (0.3)
MTL-SDCA, Sqr	**SIFT**	**20.8 (0.4)**	**28.9 (0.4)**	**37.6 (0.3)**	**46.9 (0.3)**
Sánchez *et al*. [15]	SIFT+LCS	21.1 (0.3)	29.1 (0.3)	37.4 (0.3)	47.2 (0.2)
STL-SDCA, Sqr	SIFT+LCS	21.0 (0.5)	29.2 (0.3)	37.8 (0.6)	47.2 (0.4)
STL-SDCA-Stacked, Sqr	SIFT+LCS	21.1 (0.4)	29.3 (0.3)	37.9 (0.6)	47.3 (0.4)
MTL-SDCA, Sqr	**SIFT+LCS**	**21.2 (0.2)**	**29.4 (0.4)**	**38.5 (0.5)**	**47.9 (0.5)**
STL-SDCA, Sqr	SIFT+LCS+PN	20.4 (0.6)	29.0 (0.4)	37.4 (0.4)	47.1 (0.3)
STL-SDCA-Stacked, Sqr	SIFT+LCS+PN	20.8 (0.3)	29.1 (0.4)	37.5 (0.4)	47.2 (0.4)
MTL-SDCA, Sqr	**SIFT+LCS+PN**	**20.9 (0.4)**	**29.2 (0.4)**	**38.2 (0.4)**	**48.1 (0.4)**
STL-SDCA, Sqr	SIFT+LCS+L2	21.4 (0.4)	29.8 (0.5)	38.2 (0.4)	47.9 (0.3)
STL-SDCA-Stacked, Sqr	SIFT+LCS+L2	21.6 (0.3)	30.0 (0.5)	38.3 (0.4)	48.0 (0.4)
MTL-SDCA, Sqr	**SIFT+LCS+L2**	**21.7 (0.3)**	**30.3 (0.5)**	**39.0 (0.4)**	**49.0 (0.5)**
STL-SDCA, Sqr	SIFT+LCS+PN+L2	22.1 (0.6)	30.5 (0.6)	38.8 (0.3)	48.4 (0.2)
STL-SDCA-Stacked, Sqr	SIFT+LCS+PN+L2	22.3 (0.6)	30.7 (0.6)	38.9 (0.3)	48.5 (0.2)
MTL-SDCA, Sqr	**SIFT+LCS+PN+L2**	**22.4 (0.5)**	**31.0 (0.7)**	**39.5 (0.3)**	**49.5 (0.3)**

Table 3: Mean accuracy (%) and standard deviation across 10 splits on the SUN397 [21] dataset. **STL-SDCA** - single task learning (one-vs-all SVMs trained via SDCA [16]); **STL-SDCA-Stacked** - two layer architecture (second layer SVMs are trained on the outputs of the first layer); **MTL-SDCA** - multitask learning method described in Algorithm 1; **Lin/Sqr/Chi** - linear/Hellinger/χ^2 kernel; **LCS** - Local Color Statistic descriptor; **PN** - LCS with power normalization; **L2** - LCS with ℓ_2-normalization; **Ntrain** - number of training examples per class.

Impact of the kernel map: We compare three SVM kernels: linear, Hellinger, and χ^2-kernel. The Hellinger kernel in our setting is equivalent to performing power normalization with $\rho = 0.5$ on the *full* feature vector, *i.e*. both SIFT and LCS combined. We observe that the Hellinger kernel performs better than the linear one and is comparable to the χ^2-kernel at significantly lower computational cost. We thus avoid the χ^2-kernel in our further experiments.

4.3. Baselines

In this section we aim to establish a baseline by reproducing the results of Sánchez *et al*. [15], which is the current state of the art method on SUN397. We first show that this strong baseline can be improved using the feature tuning techniques discussed in Section 4.2. As before, the SVM parameter C for single task learning is selected by 2-fold cross-validation and the final model is retrained on the full training subset. Results are given in Table 3.

First, we are able to confirm that the Fisher Vector encoding has striking performance even when only a single type of descriptor (SIFT) is used to represent an image. Using only SIFT, this baseline (STL-SDCA, Sqr) yields an average of 45.1% accuracy across 10 splits and is further improved to 48.4% when color (LCS+PN+L2) information is added (we use the accuracies obtained for $Ntrain = 50$ in Table 3, but similar improvements are obtained for the other cases). We note that these results are obtained using the Hellinger kernel, joint PCA on both SIFT and LCS, and performing power- and ℓ_2-normalization of the LCS feature. This baseline already outperforms the best published results ([15] obtained 43.3% accuracy using SIFT only and 47.2% accuracy using SIFT+LCS) and also outperforms the method proposed by the authors of the dataset [21] (38.0%), as well as more recent work [19, 5] (35.6% and 40.9% respectively). We note that the DeCAF features in [5] were learned on ImageNet data which may explain why a deep convnet is outperformed in this case.

The question we ask next is whether a jointly learned lower dimensional representation can exploit commonalities across scene classes to further improve performance.

4.4. Multitask Learning

The downside of having a dense high dimensional image descriptor (apart from the scalability issues) is that it

also captures a significant amount of noise irrelevant to the given object category. Hence, when the number of training examples is small, it is difficult to identify features that generalize well and separate them from noise. The situation becomes even worse when there are highly related tasks that are trained using the one-vs-all approach.

The nature of the SUN dataset is such that there are intrinsically related classes that have very similar visual appearance. Let us consider an example. There are three different categories related to art: "art school", "art studio", and "art gallery". Visual differences between these classes are rather subtle and are likely to be dominated by non-discriminative information in the high dimensional image descriptor. A classifier trained in the one-vs-all regime is thus likely to pick a random subset of features that just happen to discriminate between these related classes on few examples and will not generalize well.

Our multitask learning approach, on the other hand, addresses this issue by forcing all classifiers to first agree on a significantly lower dimensional subspace of features and only then attempt to discriminate between the classes.

One natural baseline for comparison in this case is a two layer feed-forward architecture where the outputs of SVMs from the first layer are used as features (inputs) to the SVMs in the second layer. We refer to this approach as **STL-SDCA-Stacked**. Note that the matrix of the first layer predictors in this case is fixed and the resulting subspace cannot be influenced by the second layer predictors. On the contrary, our MTL method allows the matrix U to be iteratively updated thus propagating information from the second layer.

The regularization parameters for both the STL-SDCA-Stacked and the MTL-SDCA methods were tuned on the first split of SUN397 and then fixed for the other 9 splits.

When looking at the results obtained by STL-SDCA-Stacked in Table 3, it becomes evident that the improvement over single task is minor (.1%–.2%), but consistent. This gives us hope that there are inter-class correlations that could be exploited, even though the considered stacked architecture may be suboptimal in this case.

Let us now discuss the results of our multitask approach.

Top-1 accuracy: Results in Table 3 clearly indicate superiority of a learned representation that is shared across multiple classes. MTL-SDCA is consistently better for every training subset and all choices of image descriptors. Furthermore, the improvement is more significant than for the stacked single task approach.

Take for example the performance for Ntrain=50. MTL-SDCA achieves 46.9% using SIFT only and 49.5% using SIFT with LCS+PN+L2. This is better than the best published results as well as our strong baselines reported above. While the improvement is not particularly strong (1.6% and 1% correspondingly when compared to the stacked classifier), it is consistent across all settings (SIFT vs. SIFT+LCS and different amounts of training data).

Top-K accuracy: Because there are intrinsically ambiguous classes (like the art scenes mentioned above, or a factory and assembly line scenes, or different types of shops, *etc*.) we believe that the top-1 accuracy is a suboptimal performance measure on this dataset. We thus extend our evaluation by reporting mean top-K accuracy for each $K = 1, \ldots, 20$ in Figure 1.

Again we observe that MTL-SDCA consistently improves classification performance not only for every image descriptor and every training subset, but also for every number of allowed guesses K. Moreover, the improvement is more significant at $K \geq 3$, *e.g*. using SIFT only and Ntrain=20 examples per class, MTL-SDCA improves top-5 accuracy by 3.7% and top-15 by 5%.

Finally, we also compare to the estimated human performance based on the top-1 accuracy of the AMT workers (68.48%), which is computed from the confusion matrix of "good workers" provided by Xiao *et al*. [21]. We observe that on average already 3-4 guesses are sufficient to reach human performance on this dataset.

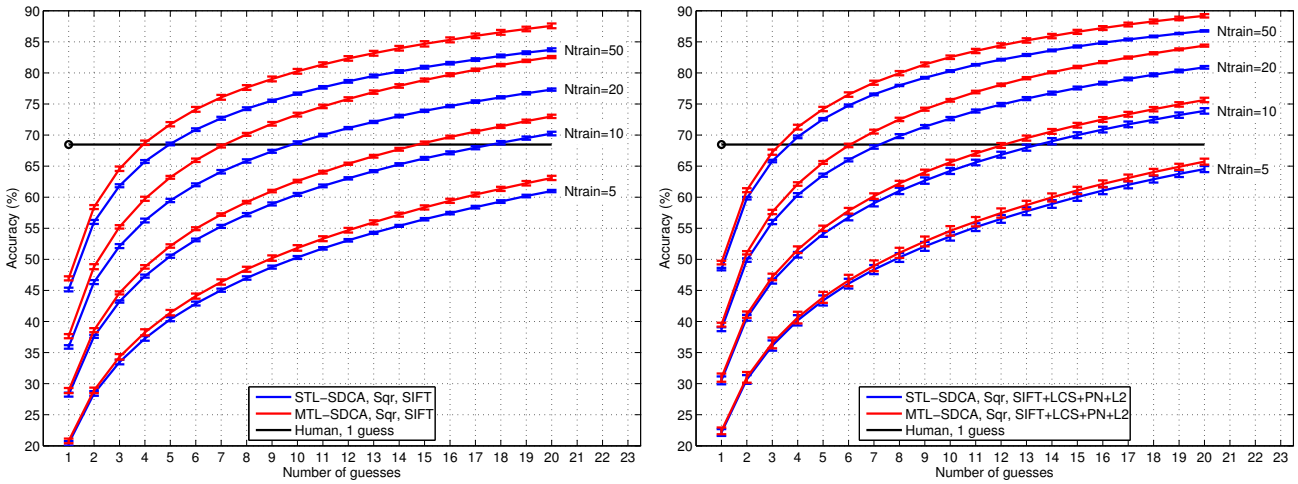
Runtime analysis: The overhead of multitask learning is relatively small (approximately a factor of 4) if the cost for computing the kernel matrices is taken into account, and is close to negligible (6%–12%) when complete image classification pipeline is considered (since most of the time is spent on computation of image descriptors). Further details can be found in the supplementary material.

5. Conclusion

We proposed a novel multitask representation learning scheme that scales to high-dimensional feature representations (such as the Fisher Vector) which are often used to obtain the best performance in object and scene classification tasks. The principle idea is to jointly learn a low dimensional representation that is shared across all classes and thus allows to leverage existing inter-class correlations.

Such inter-class relations exist in many tasks in computer vision. The running example of this paper is scene classification where different scene types, such as *e.g*. art schools and art studios, share common visual features. Thus jointly learning their respective representation has the potential to increase both robustness and accuracy, as well as to allow training classifiers from relatively small sample sizes.

Our multitask approach outperforms the state of the art on the SUN397 dataset and consistently improves classification performance over the respective single task baselines. Moreover, the improvement is even more evident when performance is evaluated via the top-K accuracy for $K > 1$, which we interpret as the ability of the MTL method to discover groups of related classes. We also conducted experiments concerning feature implementation details, which



(a) SIFT

(b) SIFT + Color

Figure 1: Mean top-K accuracy (%) and standard deviation across 10 splits on the SUN397 [21] dataset. The number of guesses K is varied between 1 and 20. **STL-SDCA** - single task learning (one-vs-all SVMs trained via SDCA [16]); **MTL-SDCA** - multitask learning method described in Algorithm 1; **LCS** - Local Color Statistic descriptor; **PN** - LCS with power normalization; **L2** - LCS with ℓ_2-normalization; **Sqr** - Hellinger kernel; **Ntrain** - number of training examples per class.

helped to improve performance. Note, however, that the proposed multitask learning method is not tied to a particular choice of features and can be applied with other image descriptors than the ones used in this work.

References

[1] A. Argyriou, T. Evgeniou, and M. Pontil. Convex multi-task feature learning. *Machine Learning*, 73(3):243–272, 2008. 1, 2

[2] R. Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997. 1

[3] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. In *BMVC*, 2011. 1, 2

[4] S. Clinchant, G. Csurka, F. Perronnin, and J.-M. Renders. XRCE's participation to ImagEval. In *ImageEval workshop at CVIR*, 2007. 4

[5] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition. *arXiv:1310.1531*, 2013. 6

[6] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, 2008. 4

[7] J. Gorski, F. Pfeuffer, and K. Klamroth. Biconvex sets and optimization with biconvex functions: a survey and extensions. *Math. Meth. Oper. Res.*, 66(3):373–407, 2007. 2, 3

[8] T. Jaakkola, D. Haussler, et al. Exploiting generative models in discriminative classifiers. In *NIPS*, 1999. 1

[9] M. Juneja, A. Vedaldi, C. Jawahar, and A. Zisserman. Blocks that shout: distinctive parts for scene classification. In *CVPR*, 2013. 1, 2

[10] Z. Kang, K. Grauman, and F. Sha. Learning with whom to share in multi-task feature learning. In *ICML*, 2011. 1, 4, 5

[11] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004. 4

[12] J. Mairal, F. Bach, and J. Ponce. Task-driven dictionary learning. *TPAMI*, 34(4):791–804, 2012. 1

[13] A. Maurer, M. Pontil, and B. Romera-Paredes. Sparse coding for multitask and transfer learning. In *ICML*, 2013. 1, 4

[14] B. Romera-Paredes, A. Argyriou, N. Berthouze, and M. Pontil. Exploiting unrelated tasks in multi-task learning. In *AISTATS*, 2012. 1

[15] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek. Image classification with the Fisher vector: theory and practice. *IJCV*, pages 1–24, 2013. 1, 2, 4, 5, 6

[16] S. Shalev-Shwartz and T. Zhang. Proximal Stochastic Dual Coordinate Ascent. *arXiv:1211.2717*, 2012. 1, 2, 3, 4, 6, 8

[17] S. Shalev-Shwartz and T. Zhang. Accelerated Proximal Stochastic Dual Coordinate Ascent for Regularized Loss Minimization. *arXiv:1309.2375*, 2013. 1, 3, 4

[18] S. Shalev-Shwartz and T. Zhang. Stochastic dual coordinate ascent methods for regularized loss minimization. *JMLR*, 14:567–599, 2013. 1, 3

[19] Y. Su and F. Jurie. Improving image classification using semantic attributes. *IJCV*, 100(1):59–77, 2012. 6

[20] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. `http://www.vlfeat.org/`, 2008. 5

[21] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. SUN database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010. 1, 4, 5, 6, 7, 8

[22] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *J. R. Stat. Soc.*, 67(2):301–320, 2005. 4