# LAPLACIAN DEEP KERNEL LEARNING FOR IMAGE ANNOTATION

*Mingyuan Jiu, Hichem Sahbi*

LTCI, CNRS, Télécom ParisTech, Université Paris-Saclay, France

## ABSTRACT

Semi-supervised learning seeks to build accurate classification machines by taking advantage of both labeled and unlabeled data. This learning scheme is useful especially when labeled data are scarce while unlabeled ones are abundant. Among the existing semi-supervised learning algorithms, Laplacian support vector machines (SVMs) are known to be particularly powerful but their success is highly dependent on the choice of kernels.

In this paper, we propose an algorithm that designs kernels as a part of Laplacian SVM learning. The proposed kernels correspond to *deep* multi-layered combinations of elementary kernels which capture simple – linear – as well as intricate – nonlinear – relationships between data. Our optimization process finds both the parameters of the deep kernels and the Laplacian SVMs in a unified framework resulting into highly discriminative and accurate classifiers. When applied to the challenging ImageCLEF2013 Photo Annotation benchmark, the proposed deep kernels show significant and consistent gain compared to existing elementary kernels as well as standard multiple kernels.

***Index Terms***— Multiple kernel learning, semi-supervised learning, Laplacian SVMs, image annotation.

## 1. INTRODUCTION

Learning from labeled and unlabeled data, a.k.a semi supervised learning (SSL), has gained a considerable attention, in the last decade, for different application domains [1, 2, 3] including manifold learning [4] and object category recognition [3]. Initially introduced by [5], SSL is mainly targeted to training problems with scarce labeled data. The general recipe of SSL consists in building inference machines by optimizing an empirical loss together with a regularization criterion. While the former relies on few labeled data, the latter uses abundant unlabeled sets in order to model the topology of the manifold enclosing data and to smooth the learned decision criteria. In practice, regularization criteria are implemented using two major principles [6]: the first one suggests that close data in a high density area of the input space should have similar labels while the second one considers that the decision boundary should exist in the low density areas of the input space. Following these two principles, several SSL algorithms have been proposed in the literature mainly for classification tasks.

Transductive SVM [1] is one of these SSL techniques which learns a decision criterion by optimizing binary-valued class memberships that minimize a hinge loss both on labeled and unlabeled data. This optimization process, even though combinatorial, can be solved iteratively in two steps: first, an inductive SVM is used to infer the values of the binary memberships on the unlabeled data. Then, the parameters of SVMs are updated accordingly using quadratic programming. Resulting from the joint (and non-convex) optimization of memberships and SVM parameters, the learning process is not guaranteed to reach a global optimum. Other methods, as in [7], reshape semi-supervised SVMs as mix-integer programming problems that minimize hinge loss criteria on unlabeled data w.r.t binary choices of labels for classification; however, this scheme is not applicable to large scale datasets. As an alternative (and more tractable solution), Laplacian SVM is introduced in [2]; it combines the geometry of the marginal distribution of data as a regularization term inside SVM objective function and provides a closed form solution.

Among these well studied kernel-based SSL models, Laplacian SVM is particularly successful and also tractable. However, its success is highly dependent on the choice of kernels. The latter are defined as symmetric functions that measure similarity between any two data, and when they are positive semi-definite, they can be expressed as inner products in high dimensional Hilbert spaces. A relevant kernel should reserve a high similarity *iff* two data belong to the same class. Although different standard kernels exist in the literature (including linear, Gaussian, etc. [8]), it is not always possible to known a priori which kernel is suitable for a given task and domain specific knowledge is required in order to handcraft appropriate kernels.

Much effort has recently been undertaken in order to automatically design suitable kernels from training data [9, 10, 3, 11, 12]. For instance authors in [13] learn explicit transductive kernel maps using Laplacian regularization between labeled and unlabeled data while in [3] authors learn context-dependent kernels that improve similarity by integrating the context. Another family of kernel design al-

This work was supported in part by a grant from the research agency ANR (Agence Nationale de la Recherche) under the MLVIS project.

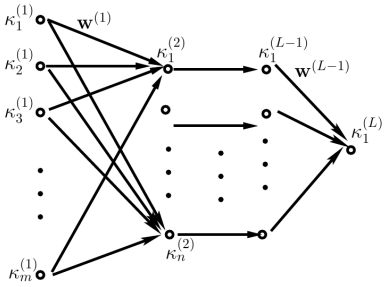


**Fig. 1**. The deep kernel network.

gorithms, known as multiple kernel learning (MKL), seeks to learn discriminative similarities from multiple elementary kernels; indeed, several works [9, 14, 15, 16, 17] focus on learning a linear combination of elementary kernels, by optimizing convex problems. However, these MKL algorithms are only restricted to linear combinations. Other (including nonlinear) combinations have recently been investigated [18, 19, 20, 21, 22]; Cortes et al. [19] propose nonlinear combinations of polynomial kernels while Cho and Saul [18] develop Arc-cosine kernels that mimic the computation of large neural nets. Zhuang et al. [23] propose a two layer nonlinear MKL framework, where exponentials are used as activation functions and authors in [24] extend this work to more than two layers using a semi-supervised setting.

In this paper, we propose a SSL algorithm that learns deep kernel networks as a part of Laplacian SVMs. This work combines the strength of deep networks *with* the high generalization capabilities of Laplacian SVMs resulting into effective classifiers as shown through image annotation experiments. This contribution is also related to our previous work [24] but has major updates: indeed, the regularization criterion used in this paper is different from the one in [24]; the latter, smooths kernel maps (in the Hilbert space) while the former takes into account the topological structure of (labeled and unlabeled) data and smooths the outputs of the decision criteria. This results into improved label prediction accuracy, compared to [24], as shown later in this paper which is organized as follows: first, we briefly present our deep kernel network in Section 2 and then we introduce, in Section 3, the semi-supervised SVM algorithm that learns both the parameters of deep kernels and Laplacian SVMs. The experimental results on the ImageCLEF2013 annotation database are shown in Section 4, followed by the conclusion.

## 2. OVERVIEW OF DEEP KERNEL NETWORKS

A deep kernel network [23, 24] in essence is a multi-layer perceptron. It is fed with a vector of elementary kernel values between pairs of data in the input layer, and it includes several layers that evaluate intermediate multiple kernels, and finally it provides a final kernel value in the output layer. Fig.1 shows an example of its architecture. The recursive feed forward process calculates a nonlinear activation function over linear combinations of kernel values in the previous layers until the output. This recursive form is defined as: $\{\kappa_p^{(l)}(\cdot,\cdot) = g\big(\sum_q \mathbf{w}_{p,q}^{(l-1)} \ \kappa_q^{(l-1)}(\cdot,\cdot)\big)\}$, here $\kappa_q^{(l-1)}(\cdot,\cdot)$ stands for a kernel value at unit $q$ and layer $(l-1)$, and $\{\mathbf{w}_{p,q}^{(l-1)}\}_q$ correspond to the weights connecting layers $(l-1)$ and $l$ (at unit $p$), which are initialized with positive values. In this deep kernel network, $g(\cdot)$ is a nonlinear activation function; for instance, the hyperbolic or exponential functions [23]. The former are chosen in order to make learning numerically stable and also to preserve the positive definiteness of all intermediate and output kernels. This network learns an implicit kernel map representation for the inputs, instead of deep explicit feature representations such as convolutional neural networks [25].

The learning process involves two sets of parameters: weights of the deep kernel network and SVM parameters. As a joint optimization of these parameters is difficult (and not convex), we adopt, instead, an alternating optimization strategy. First, we fix the weights in the deep network and optimize the SVM parameters (using LIBSVM [26]); then, we fix the parameters of SVMs and we update the weights in the deep network using gradient descent and back-propagation [25]. In this work, we use Laplacian SVMs on top of the deep network. Again, this choice is motivated by the strong generalization capabilities of Laplacian SVMs when modeling the topology of the manifold enclosing labeled as well as unlabeled data in the training process (through Laplacian regularization), and this results into highly effective classifiers.
In the subsequent section, we introduce the mathematical details about the evaluation of the backward information from Laplacian SVMs and the update of the weights in the deep network.

## 3. LEARNING

Considering a multi-class problem (with $K$ classes), we define $\mathcal{L} = \{(\mathbf{x}_i, \mathbf{y}_i^k)\}_{i=1}^{\ell}$ as a labeled training set and $\mathcal{U} = \{\mathbf{x}_i\}_{i=\ell+1}^{\ell+u}$ as an unlabeled test set. In this definition, $\mathbf{y}_i^k$ stands for the membership of data $\mathbf{x}_i$ to the $k^{th}$ class, i.e. $\mathbf{y}_i^k = 1$ iff $\mathbf{x}_i$ belongs to class $k$ and $-1$ otherwise. Our goal is to train a set of classifiers $\{f_k\}_{k=1}^{K}$ on top of a deep kernel network (including $L$ layers) in order to predict whether a given test data $\mathbf{x}_i \in \mathcal{U}$ belongs to the class $k$ depending on the sign of $f_k(\mathbf{x}_i)$.

The general form of our SSL objective function has two major parts: the first one is the standard empirical loss (classification error of the learned SVMs) while the second one is a regularizer that controls the smoothness of SVM parameters as well as outputs of these classifiers. Although both [24] and this work consider the topological structure of data into learning, the two approaches are conceptually different. Indeed, the method proposed in this paper smooths explicitly the outputs of the SVMs while the approach in [24] considers

| | SIFT | C-SIFT | RGB-SIFT | OPP-SIFT | COLORHIST | GETLF | GIST | GIST2 | HSVHIST | LBP |
|---|---|---|---|---|---|---|---|---|---|---|
| lin | 36.7/23.3/52.1 | 36.9/22.7/52.9 | 39.0/**23.4**/52.5 | 37.9/22.7/52.1 | 34.7/18.5/47.6 | 33.4/17.4/45.1 | 33.2/15.4/40.0 | 34.7/18.5/47.6 | 30.8/14.7/39.5 | 31.0/19.4/38.3 |
| pol | 33.4/17.7/49.4 | 37.1/19.3/52.1 | 32.7/18.6/48.8 | 32.1/18.7/48.7 | 36.7/21.7/51.5 | 37.7/21.5/51.3 | 34.5/17.8/48.2 | 35.2/18.9/48.3 | 36.2/17.3/46.9 | 35.5/21.1/46.1 |
| RBF | 35.4/22.7/52.0 | 39.4/22.1/53.5 | 36.0/**23.4**/52.8 | 37.6/23.3/52.5 | 41.1/22.0/53.2 | 36.6/19.5/50.8 | 36.8/18.5/49.0 | 33.8/19.4/48.6 | 37.2/18.1/49.0 | 36.3/20.8/50.3 |
| HI | 39.0/20.8/53.4 | **39.5**/21.9/**54.9** | 39.0/21.6/53.7 | 36.0/21.7/51.4 | 35.2/22.4/52.9 | 34.0/20.1/49.5 | 34.3/17.7/48.0 | 34.5/19.8/48.6 | 36.4/16.9/50.1 | 35.4/18.8/49.3 |

**Table 1**. This table shows the baseline performance of Laplacian SVMs (in %) for different elementary kernels; triple scores (././.) correspond respectively to MF-S, MF-C and mAP performances.

a regularization term that smooths only the kernel maps in the high dimensional Hilbert space. As the regularization in [24] has no direct impact on the outputs of the SVMs, the regularization used in this paper in highly preferred and turns out to be more effective (as shown later in experiments).

Considering the previous statements, our objective function is defined as

$$\min_{\mathbf{w},\{f_k\}} \sum_{k=1}^{K} C_k \sum_{i=1}^{\ell} \max\left(0, 1 - \mathbf{y}_i^k f_k(\mathbf{x}_i)\right) + \frac{1}{2}\left\|f_k\right\|_{\mathcal{H}}^2 + \lambda \sum_{i=1}^{\ell+u} \sum_{j \in \mathcal{N}_i} A(\mathbf{x}_i, \mathbf{x}_j)(f_k(\mathbf{x}_i) - f_k(\mathbf{x}_j))^2, \tag{1}$$

with $\mathbf{w}$ being the weights of the deep kernel network and $C_k$, $\lambda$ balances between empirical loss and regularization. The first term, in this objective function, is the hinge loss defined on $\mathcal{L}$, the second term is the regularizer of the SVM parameters, and the third term controls the smoothness of the SVM outputs. In Eq. (1), $\mathcal{N}_i$ is the set of neighbors of $\mathbf{x}_i$ and $A(\mathbf{x}_i, \mathbf{x}_j)$ is a similarity between any $\mathbf{x}_i$, $\mathbf{x}_j$ defined as

$$A(\mathbf{x}_i, \mathbf{x}_j) = \begin{cases} \frac{1}{2}[H_{ij} + S_{ij}] & \forall i,j \in \{1,\ldots,\ell\} \\ H_{ij} & \text{otherwise,} \end{cases} \tag{2}$$

where $H_{ij}$, $S_{ij}$ respectively correspond to histogram intersection and Jaccard similarity; the latter is defined as the ratio between label "intersection" and "union", i.e., $S_{ij} = \sum_{k=1}^{K}[(\mathbf{y}_i^k = +1) \wedge (\mathbf{y}_j^k = +1)] / \sum_{k=1}^{K}[(\mathbf{y}_i^k = +1) \vee (\mathbf{y}_j^k = +1)]$. Considering a matrix form of the regularizer, Eq. (1) can be rewritten as

$$\min_{\mathbf{w},\{f_k\}} \sum_{k=1}^{K} C_k \sum_{i=1}^{\ell} \max\left(0, 1 - \mathbf{y}_i^k f_k(\mathbf{x}_i)\right) + \frac{1}{2}\left\|f_k\right\|_{\mathcal{H}}^2 + \lambda \mathbf{f}_k^{\intercal} \mathbf{L} \mathbf{f}_k, \tag{3}$$

where $\mathbf{L}$ is the graph Laplacian and $\mathbf{f}_k$ is a vector that gathers the outputs of the $k^{th}$ SVM classifier applied to all data in $\mathcal{L} \cup \mathcal{U}$. Following the representer theorem (see [2]), the solution of Eq. (3) can be written as $f_k(\mathbf{x}) = \sum_{i=1}^{\ell+u} \alpha_i^k \kappa_1^{(L)}(\mathbf{x}, \mathbf{x}_i)$. By introducing Lagrange multipliers, (3) is transformed into

$$\min_{\mathbf{w}} \max_{\alpha,\beta} J = \min_{\mathbf{w}} \max_{\alpha,\beta} \sum_{k=1}^{K} \frac{1}{2} \alpha^{k\intercal} \left(\mathbf{K}_1^{(L)} + 2\lambda\, \mathbf{K}_1^{(L)} \mathbf{L} \mathbf{K}_1^{(L)}\right) \alpha^k - \alpha^{k\intercal} \mathbf{K}_1^{(L)} \mathbf{D}^{\intercal} \mathbf{Y}^k \beta^k + \sum_{i=1}^{\ell} \beta_i^k, \tag{4}$$

here $\mathbf{K}_1^{(L)}$ is the Gram matrix associated to the output layer of the deep network. The vector $\alpha^k$ corresponds to the $(\ell+u)$ SVM parameters, $\mathbf{D} = [\mathbf{I}\ 0]$ is an $\ell \times (\ell+u)$ matrix with $\mathbf{I}$ being the $\ell \times \ell$ identity matrix and $\mathbf{Y}^k = \text{diag}(\mathbf{y}_1^k, \mathbf{y}_2^k, \ldots, \mathbf{y}_\ell^k)$. The derivative of Eq. (4) w.r.t $\alpha^k$ implies

$$\alpha^k = (\mathbf{I} + 2\lambda \mathbf{L} \mathbf{K}_1^{(L)})^{-1} \mathbf{D}^{\intercal} \mathbf{Y}^k \beta^{k*}. \tag{5}$$

Substituting Eq. (5) into (4), we get its corresponding dual problem which is solved using a standard SVM solver. Then, the optimal $\beta^{k*}$ is used in order to evaluate $\alpha^k$ via Eq. (5).

Following an alternating (and iterative) optimization, we fix $\alpha^k$ (as described above) and then we estimate the weights $\{\mathbf{w}_{p,q}^{(l)}\}$ of the intermediate and the output kernels, using the chain rule and the following gradient

$$\frac{\partial J}{\partial \mathbf{K}_1^{(L)}} = \sum_{k=1}^{K} \frac{1}{2} \alpha^k \alpha^{k\intercal} + \lambda(\mathbf{L} + \mathbf{L}^{\intercal}) \mathbf{K}_1^{(L)} \alpha^k \alpha^{k\intercal} - \left[\alpha^k (\mathbf{Y}^k \beta^k)^{\intercal}\ \ 0_{(\ell+u)\times u}\right]. \tag{6}$$

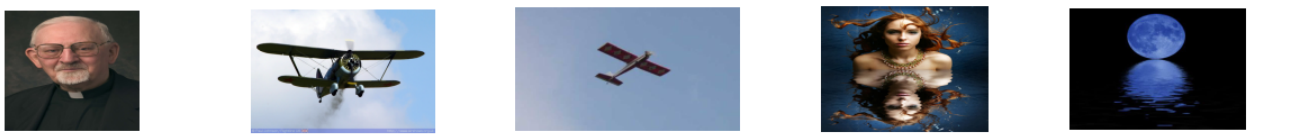
Finally, we update the deep kernel weights by standard back-propagation (more details can be found in [24]). This alternating optimization of the Laplacian SVM parameters and the deep kernel weights continues until convergence.

## 4. EXPERIMENTS

In this section, we show the impact of the proposed deep kernel network on the performance of image annotation using the challenging ImageCLEF2013 Photo Annotation benchmark [27]. The underlying task corresponds to a multi-label classification problem, in other words, one or several semantic concepts are assigned to a given test image; in total 95 concepts are considered. The ImageCLEF2013 database has three sets: training, dev and test sets. As the ground truth is available (released) only on the dev set, we consider the latter (with 1000 images) for evaluation and we randomly split it into two halves; one for training and the other for testing. Then, we train "one-versus-all" Laplacian SVMs (for all concepts) on top of the deep kernel network and we use the signs of these SVMs to assign concepts to test images.

Performances are measured using the F-scores (harmonic means of recall and precision) at the concept and the sample levels (resp. MF-C and MF-S) as well as the mean Average Precision (mAP) [27]. We consider a combination of ten different features (provided by ImageCLEF2013 challenge) and four elementary kernels (linear, polynomial with 2 orders, Gaussian[1] and histogram intersection) as an input to our deep

[1]With a scale parameter set to average Euclidean distance between data.



| concept | GT | single | sup | semi |
|---|---|---|---|---|
| cityscape | N | Y | N | N |
| daytime | N | Y | N | N |
| elder | Y | N | N | N |
| female | N | N | Y | Y |
| indoor | Y | N | N | Y |
| male | Y | N | N | Y |
| outdoor | N | Y | N | Y |
| person | Y | N | Y | Y |
| portrait | Y | N | N | N |
| shadow | N | Y | N | N |
| sky | N | Y | N | Y |
| snow | N | Y | N | N |

| concept | GT | single | sup | semi |
|---|---|---|---|---|
| airplane | Y | N | N | N |
| cloud | Y | N | N | Y |
| daytime | Y | Y | Y | Y |
| forest | N | N | N | Y |
| outdoor | Y | Y | Y | Y |
| plant | N | Y | Y | Y |
| sky | Y | Y | Y | Y |
| smoke | Y | N | N | N |
| snow | N | Y | Y | Y |
| sport | N | Y | N | N |
| tree | N | N | N | Y |
| vehicle | Y | N | N | Y |
| water | N | Y | N | N |

| concept | GT | single | sup | semi |
|---|---|---|---|---|
| airplane | Y | N | N | N |
| cityscape | N | Y | N | N |
| cloudless | Y | N | N | Y |
| daytime | Y | Y | Y | Y |
| male | N | N | N | Y |
| outdoor | Y | Y | Y | Y |
| plant | N | Y | N | Y |
| sea | N | N | Y | Y |
| sky | Y | Y | Y | Y |
| snow | N | Y | N | N |
| tree | N | N | N | Y |
| vehicle | Y | N | N | Y |
| water | N | N | Y | Y |

| concept | GT | single | sup | semi |
|---|---|---|---|---|
| building | N | Y | N | N |
| child | N | Y | N | N |
| cityscape | N | Y | N | N |
| female | Y | N | N | N |
| furniture | N | Y | N | N |
| indoor | N | Y | N | N |
| male | N | Y | N | Y |
| nighttime | N | Y | Y | Y |
| outdoor | N | Y | N | Y |
| person | Y | Y | Y | Y |
| reflection | Y | N | N | Y |
| sculpture | N | Y | N | N |
| snow | N | Y | N | N |
| water | Y | N | N | Y |

| concept | GT | single | sup | semi |
|---|---|---|---|---|
| cityscape | N | Y | N | N |
| male | N | N | N | Y |
| moon | Y | N | Y | Y |
| nighttime | Y | Y | Y | Y |
| outdoor | Y | Y | N | Y |
| reflection | Y | N | N | Y |
| sea | N | Y | Y | Y |
| shadow | N | Y | N | N |
| silhouette | N | Y | Y | N |
| sky | Y | N | N | Y |
| smoke | N | N | Y | Y |
| snow | N | Y | N | N |
| sun | N | Y | N | N |
| sunrise/set | N | Y | Y | N |
| water | Y | Y | Y | Y |

**Fig. 2**. This figure shows annotated samples using histogram intersection kernel on C-SIFT features ("single"), supervised learning on 4-layer network ("sup") and semi-supervised learning on 3-layer network ("semi") respectively. "GT" means ground-truth. "Y" (resp. "N") stands for the presence (resp. absence) of a given concept in an image.

kernel network. Tab.1 shows the (baseline) performances of Laplacian SVMs using different combinations of elementary kernels and features. We observe that histogram intersection (applied to C-SIFT) provides the best baseline performances.

| Architecture | method | MF-S | MF-C | mAP |
|---|---|---|---|---|
| Baseline | GMKL [17] | 41.3 | 24.3 | 49.0 |
| | 2LMKL [23] | 45.0 | 25.8 | 54.0 |
| | semi DKL[24] | 46.8 | 29.5 | 58.5 |
| 2-layer KL | sup | 45.0 | 25.8 | 54.0 |
| | semi | 46.8 | 28.6 | **58.9** |
| 3-layer KL | sup | 46.0 | 29.5 | 55.8 |
| | semi | **47.8** | **30.0** | 58.6 |
| 4-layer KL | sup | 46.6 | 29.6 | 56.3 |
| | semi | 45.5 | 28.4 | 58.4 |
| 5-layer KL | sup | 46.5 | 29.3 | 56.2 |
| | semi | 46.6 | 28.1 | 57.3 |

**Table 2**. The performance of different baselines and Laplacian deep kernel learning w.r.t different number of layers in the deep network.

We train our deep kernel network using the method described in Section 3. The input layer has 40 units (resulting from the combination of ten features and four elementary kernels) while each intermediate layer has 80 units. In all these units, different activation functions are adopted; hyperbolic for intermediate layers and exponential for the output layer. Note that hyperbolic functions act as normalizers and make the learning process numerically (more) stable[2].

As a matter of comparison we also consider a supervised setting in order to train our deep kernel network (i.e., $\lambda = 0$). For both supervised and semi supervised learning, we chose the penalty parameter $C_k$ using 3-fold cross validation (on the training set) with values ranging from $2^{-4}$ to $2^8$. Similarly to $C_k$, we select the best parameter ($\lambda$ in $[2^{-10}, 2^4]$) for the semi supervised setting. All the performances, including different MKL methods ([17, 23]) and SSL on 3-layer deep kernel network in [24], are depicted in Tab. 2. Some annotation examples are also shown in Fig. 2. From these results, we observe that

i) SSL with the deep kernel network (mainly with 3 layers) outperforms SSL with elementary kernels and other comparative approaches; indeed, the underling MF-S, MF-C and mAP scores reach 47.8%/30.0%/58.6% respectively and this corresponds to a substantial gain compared to histogram intersection on the C-SIFT features.

ii) The accuracy of the classifiers increases when the learned kernel is sufficiently deep, and stabilizes afterwards. The best results for supervised learning are achieved with a 4-layer network (46.6%/29.6%/56.3%), but we still get an extra gain when using semi supervised learning with a 3-layer network. In these results, SSL of deep kernel networks with Laplacian SVM, obtains the best overall performances.

iii) For 4 and 5-layer networks, the performance of Laplacian SVMs slightly decreases compared to 3-layer network; this may result from the lack of labeled data compared to the complexity (number of parameters) of 4 and 5 layer networks.

## 5. CONCLUSION

We introduced in this paper a novel semi supervised deep kernel learning algorithm. The proposed method considers the topology of labeled and unlabeled data as a part of kernel design resulting into better discrimination. This learning is achieved by optimizing an objective function that mixes empirical error and Laplacian regularization. Alternating optimization is used and provides us with the parameters of the Laplacian SVMs as well as the weights of the deep networks that define the best nonlinear combination of multiple elementary kernels. Experiments conducted on the challenging ImageCLEF2013 Photo Annotation benchmark show that the proposed kernel design framework is effective compared to different baselines as well as supervised deep kernel learning.

[2]This observation is made after extensive comparison w.r.t the previous work in [24] where exponential functions were instead applied in the intermediate layers; and this required extensive tuning to obtain convergence.

## 6. REFERENCES


[1] T. Joachims, "Transductive inference for text classification using support vector machines," in *ICML*, 1999.

[2] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *JMLR*, vol. 7, pp. 2399–2434, 2006.

[3] H. Sahbi, J.-Y. Audibert, and R. Keriven, "Context-dependent kernels for object classification," *PAMI*, vol. 33, pp. 699–708, 2011.

[4] Belkin M. and Niyogi P., "Semi-supervised learning on riemannian manifolds," vol. 56, pp. 209–239, 2004.

[5] V. Vapnik, "Statistical learning theory," 1998.

[6] H. Narayanan and M. Belkin, "On the relation between low density separation, spectral clustering and graph cuts," *In NIPS*, 2006.

[7] Bennet K. and A. Demirez, "Semi-supervised support vector machines," in *NIPS*, 1998.

[8] J. Shawe-Taylor and N. Cristianini, "Kernel methods for pattern analysis," *Cambriage University Press*, 2004.

[9] G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M. I. Jordan, "Learning the kernel matrix with semi-definite programming," *JRML*, vol. 5, pp. 27–72, 2004.

[10] C. Corinna, M. Mehryar, and R. Afshin, "Two-stage learning kernel algorithms," in *ICML*, 2010.

[11] H. Sahbi and X. Li, "Context-based support vector machines for interconnected image annotation," in *ACCV*, 2011, pp. 214–227.

[12] J. Mairal, P. Koniusz, Z. Harchaoui, and C. Schmid, "Convolutional kernel networks," in *NIPS*, 2014.

[13] P. Vo and H. Sahbi, "Transductive kernel map learning and its application to image annotation," in *ICCV*, 2007.

[14] F. Bach, G. Lanckriet, and M. Jordan, "Multiple kernel learning, conic duality, and the smo algorithm," in *ICML*, 2004.

[15] A. Rakotomamonjy, F. Bach, Canu S., and Grandvalet Yves, "Simplemkl," *JMLR*, vol. 9, pp. 2491–2521, 2008.

[16] F. Bach, "Exploring large feature spaces with hierarchical multiple kernel learning," in *NIPS*, 2009.

[17] M. Varma and B. Babu, "More generality in efficient multiple kernel learning," in *ICML*, 2009.

[18] Y. Cho and L. Saul, "Kernel methods for deep learning," in *NIPS*, 2009, pp. 1–9.

[19] C. Cortes, M. Mohri, and A. Rostamizadeh, "Learning non-linear combinations of kernels," in *NIPS*, 2009.

[20] M. Gönen and E. Alpaydin, "Localized multiple kernel learning," in *ICML*, 2008.

[21] P. Gehler and S. Nowozin, "Infinite kernel learning," in *NIPS workshop on Automatic Selection of Kernel Parameters*, 2008.

[22] E. V. Strobl and S. Visweswaran, "Deep multiple kernel learning," in *ICMLA*. IEEE, 2013, pp. 414–417.

[23] J. Zhuang, I. Tsang, and S. Hoi, "Two-layer multiple kernel learning," in *ICML*, 2011, pp. 909–917.

[24] M. Jiu and H. Sahbi, "Semi supervised deep kernel design for image annotation," in *ICASSP*, 2015.

[25] Y. LeCun, L. Botto, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[26] C.-C. Chang and C.-J. Lin, "Libsvm: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 1–27, 2011.

[27] M. Villegas, R. Paredes, and Thomee B., "Overview of the imageclef 2013 scalable concept image annotation subtask," in *CLEF 2013 Evaluation Labs and Workshop*, 2013.