Context-Aware Deep Kernel Networks for Image Annotation

Mingyuan Jiu, Hichem Sahbi

Context-Aware Deep Kernel Networks for Image Annotation

Mingyuan Jiu[a,*], Hichem Sahbi[b]

[a]*School of Information Engineering, Zhengzhou University, Zhengzhou, China*
[b]*Sorbonne University, UPMC, CNRS, LIP6, F-75005 Paris, France*

Abstract

Context plays a crucial role in visual recognition as it provides complementary clues for different learning tasks including image classification and annotation. As the performances of these tasks are currently reaching a plateau, any extra knowledge, including context, should be leveraged in order to seek significant leaps in these performances. In the particular scenario of kernel machines, context-aware kernel design aims at learning positive semi-definite similarity functions which return high values not only when data share similar contents, but also similar structures (a.k.a. contexts). However, the use of context in kernel design has not been fully explored; indeed, context in these solutions is handcrafted instead of being learned.

In this paper, we introduce a novel deep network architecture that learns context in kernel design. This architecture is fully determined by the solution of an objective function mixing a content term that captures the intrinsic similarity between data, a context criterion which models their structure and a regularization

*Corresponding author
Email addresses: iemyjiu@zzu.edu.cn (Mingyuan Jiu), hichem.sahbi@lip6.fr (Hichem Sahbi)

Published in Neurocomputing

term that helps designing smooth kernel network representations. The solution of this objective function defines a particular deep network architecture whose parameters correspond to different variants of learned contexts including layer-wise, stationary and classwise; larger values of these parameters correspond to the most influencing contextual relationships between data. Extensive experiments conducted on the challenging ImageCLEF Photo Annotation, Corel5k and NUS-WIDE benchmarks show that our deep context networks are highly effective for image classification and the learned contexts further enhance the performance of image annotation.
Keywords: Deep kernel learning, context-aware kernel networks, deep learning, image annotation

---

1. Introduction

Following the rapid development of electronic devices and social medias, there is an exponential growth of image and video collections in the web and this makes their manual annotation and search completely out of reach. This rapid growth greatly motivates the need for automatic solutions that help analyzing and indexing these large collections [1, 2, 3, 4, 5, 6]. Among these solutions, image category recognition is a major challenge, which aims at describing contents of images with multiple semantic concepts (a.k.a. keywords, categories or classes) [7, 8], for different use-cases including image retrieval, human-computer interaction, autonomous driving, etc. Image annotation is challenging as concepts are usually diverse ranging from simple objects, to abstract notions, through highly interacting scene parts; hence, learning *pure visual content models (without context) is clearly insufficient* [9].

Most of the existing image annotation techniques are based on machine learning. These methods build decision functions that learn the intricate relationships between images and their semantic concepts using variety of models including Support Vector Machines (SVMs) [10, 11], Nearest Neighbor classifiers [12, 13], deep networks[14, 15, 16, 17, 18, 19, 20, 21], etc. and the membership of images to different semantic concepts is decided by the scores of these models. Among the aforementioned machine learning techniques, SVMs are highly effective especially for tasks exhibiting scarce categories and training data. Their general principle consists in mapping nonlinearly separable data from input spaces into high (possibly infinite) dimensional spaces and finding hyperplanes that separate these data while maximizing their margin. This mapping is achieved (either explicitly or implicitly) using particular similarity functions referred to as kernels [22]. The latter, defined as symmetric positive semi-definite (p.s.d) functions, should reserve high values only when data share similar semantics and vice-versa. Several kernels have been proposed in the literature including linear, polynomial and RBF as well as histogram intersection [23, 24]. These functions can also be combined in order to learn more relevant similarities using multiple [25], additive [26] and deep kernels [27], as well as deep kernel map networks [28] which significantly reduce their computational complexity. Nonetheless, standard kernels and their combinations rely mainly on the visual content of images which is highly variable and insufficient to capture the semantics of images especially when labeled training data are scarce. Hence, context should also be leveraged in order to further improve the discrimination power of the learned kernels. As shown through this paper, context kernels should reserve high values not only when images share similar content but also comparable context, and when the latter is learned, the

accuracy of the resulting kernels is further improved.

Given an image as a lattice of cells, with each one being described with a feature vector, our goal is to design a kernel that captures the similarity between these cells while modeling their context (see also [29, 30, 31, 32, 33]). Our design principle is based on the minimization of an objective function mixing (i) a content term that captures visual similarity, (ii) a context criterion which models image structure, and (iii) a regularizer. The solution of this objective function defines the architecture of a deep kernel network[1] whose parameters correspond to the learned context. Note that this formulation is different from [31, 32] as context in this related work is handcrafted while in our proposed method, it is learned in order to further enhance classification performances (see Section 3). Note also that the approach proposed in this paper extends the preliminary work in [19] in two aspects; on the one hand, different context settings are investigated including stationary and classwise, both resulting into an extra gain in performances. On the other hand, more comprehensive analysis and experiments are achieved using diverse and larger benchmarks including ImageCLEF, Corel5k and NUS-WIDE. Considering these issues, the main contributions of this paper include:

- A novel method that learns effective (context-aware) kernels using deep networks. While the early formulation in [32, 31] relies on rigid (fixed) contexts, the ones proposed in this paper are learned instead of being handcrafted. This is achieved as a part of an "end-to-end" training framework which shows superior performances compared to handcrafted contexts.

[1]The advantage of this kernel network resides also in its computational complexity which scales linearly w.r.t. the size of the data while in standard kernel-based methods, this complexity is at least quadratic.

- The study of different variants of contexts including *layerwise, stationary and classwise*. Layerwise contexts are learned with different parameters through the layers of our deep network while stationary ones assume shared contextual relationships between all the layers and this reduces the actual number of parameters. Besides, classwise contexts are also investigated in order to build class-dependent parameters; this generates multiple branches of contexts each one dedicated to a particular category in image classification.

- All these statements are corroborated through extensive experiments in image classification using the challenging ImageCLEF, Corel5k and NUS-WIDE benchmarks.

The rest of this paper is organized as follows: first, we review the related work about context learning in Section 2, and then we revisit our previous context-aware kernel design [31] in Section 3. In Section 4, we introduce our two main contributions: i) a deep network that learns explicit context-aware kernel representations, and ii) different variants of context learning (including layerwise, stationary and classwise) which model context and further enhance the classification performances. In particular, classwise context learning makes it possible to build specific contexts for different classes. In Section 5, we show the performance and comparison of our method on the ImageCLEF Photo Annotation, Corel5k and NUS-WIDE benchmarks. Finally, we conclude the paper in Section 6 and we provide possible extensions for a future work.

2. Related work

Prior to detail our main contribution (in sections 3 and 4), we discuss in this section the related work both in image annotation and context modeling.

2.1. Image annotation

State of the art in image annotation can be categorized into two major families of methods: discriminative and generative (see for instance [34, 35, 36, 37]). The latter aim at modeling the joint distribution between observed data and their ground truth and use maximum a priori/posteriori to infer concepts on unseen data while the former seek to learn dependencies between images and their classes through decision functions[2] that map visual features into semantic concepts. Amongst the models used for image category recognition, those based on SVMs are particularly successful. In these methods, each concept is treated as an independent class and a binary SVM is trained to predict the membership of its underlying concept into a given test image [10, 41].

Different SVM-based solutions have been proposed in the literature in order to further enhance the performance of image annotation. SVM ranking is used in [38] to achieve image annotation where relevant concepts are ranked higher than irrelevant ones while semi-supervised Laplacian SVM is considered in [42] in order to propagate concepts from labeled to unlabeled images. SVMMN [43] is also proposed in order to improve the accuracy of SVM-based image annotation; it seeks to learn maximum margin classifiers with a minimum number of samples by modeling smoothness both at the sample and the concept levels together with a correlation criterion across classes. Other methods proceed differently by

[2] e.g. SVMs [10, 38], decision trees [39], artificial neural networks [40], etc.

learning discriminative kernels that improve the performance of binary SVMs; for instance, shallow and deep multiple kernel learning [27] are proposed for image annotation in order to combine different standard kernels. Context-dependent kernels [32] are also proposed for multi-class image annotation; a variant of this method in [31] makes it possible to learn explicit representations of these kernels while being highly efficient. Our proposed method, in this paper, follows this line and allows learning kernel representations *together* with their geometric contextual relationships using deep parametric networks. The learned representations of these networks also preserve the similarity of the original kernels while being highly efficient and effective.

With the resurgence of deep convolutional neural networks (CNNs) [44, 45], a further "impressive" progress has recently been observed in the aforementioned image classification methods. This gain comes essentially from the high accuracy of the learned CNNs that capture the visual content of images better than the widely used handcrafted representations [46]. The common aspect of these techniques consists in revisiting and rebuilding annotation methods by learning classifiers[3] on top of CNN representations instead of handcrafted ones (see for instance [47, 48, 49]). Other more recent work encodes structural semantic information to produce sequential labels by combining CNNs with recurrent neural networks (RNNs) [50] through semantically regularized layers [51], semantic graph embedding [52], dynamic LSTM label ordering [53], as well as label transfer in latent semantic spaces [54]. Similarly to this related work, our proposed framework is also built on top of CNNs but seeks to design representations by

[3]These classifiers include SVMs, voting, K-nearest neighbors, self-defined Bayesian models, etc.

integrating and *learning context in deep kernel networks*.

2.2. Context modeling

Context, as a complementary clue, has attracted a lot of interest in different computer vision applications ranging from 3D scene understanding [55], to scene parsing [56], through object and person re-identification [1], etc. Early work includes "shape context" [57] which models the spatial relationships between image primitives (mainly interest points) in order to design handcrafted shape representations. Later, the pyramid matching kernel, introduced in [58], models similarities between multi-layout feature representations and the context-aware keypoint extractor (CAKE), in [2], makes it possible to describe and retrieve keypoints which are representative within a certain image context. A priori knowledge of human part relationships have also been leveraged into CNNs in order to design a spatially-constrained deep learning framework [33] that captures more discriminative features for part segmentation. Context-aware kernel and its deep map variants [31, 32] are proposed in order to design kernels accounting for "image-image" relationships. In these works, handcrafted contexts are combined with different models but their design is not end-to-end, which results into sub-optimal contexts.

In context learning, most of the related work models image relationships using K-nearest neighbors (KNNs), where nearest neighbors depend on a learned similarity between images and their labels, such as TagProp [12], 2PKNN [13], etc. Our approach differs from this related work in that we aim to learn structures within images rather than between images. Other related work, based on attention [59, 60], seeks to learn the most prominent areas in images for classification or graph neural networks (GNNs) [61, 62, 63]. This is achieved in order to ex-

plicitly learn graph relationships; for instance, multiple laplacians over skeleton graphs [64] are learned in GNNs for human action recognition, hierarchical multi-graphs [65] are learned in graph convolutional networks (GCNs) for image classification, as well as semantic-specific graphs [66] which are learned using GNNs for multi-label image recognition, where each node corresponds to a salient area endowed with a semantic label. In our proposed method, we design context-aware kernel representations by modeling not only image content but also context which is learned as a part of kernel network design instead of being handcrafted.

In the particular scenario of image annotation, several other methods, leveraging contextual information, have been proposed in the literature. Authors in [3] consider undirected graphical models that jointly exploit low-level features and contextual information (as concept co-occurrences and spatial correlation statistics) to classify local image blocks into predefined concepts. Zhang et al. [4] propose a region annotation framework that exploits the semantic correlation of segmented image regions; this method assigns each segmented region to one concept and learns the relationships between labels and region locations using PSA. A hybrid annotation approach based on visual attention mechanism and conditional random fields is proposed in [5] in order to pay more attention to the salient regions during the annotation process. A tri-relational graph-based method (including image and region as well as label graphs) is proposed for web image annotation [6], and a spatial regularization network is introduced in [67] in order to exploit both semantic and spatial relationships between labels using image-level supervision only.

As stated earlier, our contribution is different from the aforementioned related work in the way context is learned as a part of deep kernel network design. From

the methodological point of view, our work is rather related to Convolutional Kernel Networks (CKN) [68] which explicitly learn kernel maps for gaussian functions using convolutional networks. However, our work is conceptually different from CKN: on the one hand, our approach is not restricted to gaussian kernels and is capable of learning a more general class of kernels. On the other hand, no context is considered in CKN while our method incorporates contexts explicitly in kernel design.

3. Context-aware kernel representations

In this section, we briefly describe the preliminary work about context-aware kernels as well as its explicit map network. A context-aware kernel models the similarity between images using not only their content but also their context. The latter is relevant especially when the visual content of images with the same semantic concepts is noisy and highly variable.

Following this goal, considering $\{\mathcal{I}_p\}_p$ as a collection of images, a kernel function κ (whose associated gram matrix denoted as $\mathbf{K}$ with $\mathbf{K}_{\mathcal{I}_p,\mathcal{I}_q} = \kappa(\mathcal{I}_p, \mathcal{I}_q)$) is learned by minimizing the objective function [31]

$$\min_{\mathbf{K}} \text{tr}(-\mathbf{K}\mathbf{S}') - \alpha \sum_{c=1}^{C} \text{tr}(\mathbf{K}\mathbf{P}_c\mathbf{K}'\mathbf{P}_c^{'}) + \frac{\beta}{2}||\mathbf{K}||_2^2, \tag{1}$$

here $\beta > 0$, $\alpha \geq 0$, $'$ and $\text{tr}(.)$ stand for matrix transpose and trace operator respectively, $\mathbf{S}$ refers to a context-free kernel matrix between data in $\mathcal{X}$ (e.g., linear kernel, RBF kernel, etc.) while $\{\mathbf{P}_c\}_{c=1}^{C}$ denotes C intrinsic adjacency matrices that capture different spatial relationships among images. The left-hand side term of Eq. (1) is a fidelity criterion that encourages high kernel values for visually similar pairs $\{(\mathcal{I}_p, \mathcal{I}_q)\}_{\mathcal{I}_p,\mathcal{I}_q}$ while the second term makes the kernel values between

these pairs stronger or weaker depending on the similarity of their neighbors. Finally, the right-hand side term acts as a regularizer that controls the smoothness of the learned kernel solution.

The optimization problem in Eq. (1) admits the closed-form kernel solution

$$\mathbf{K}^{(t+1)} = \mathbf{S} + \gamma \sum_{c=1}^{C} \mathbf{P}_c \mathbf{K}^{(t)} \mathbf{P}_c^{'}, \tag{2}$$

with $\mathbf{K}^{(0)} = \mathbf{S}$, $\gamma = \alpha/\beta$ and t being an iteration number. In this solution, γ controls the influence of the context and in practice it is chosen in order to guarantee the convergence of the kernel solution to a fixed point (see more details in [31]). Resulting from the p.s.d of $\mathbf{S}$ and the closure of the p.s.d with respect to the sum and the product, all the kernel matrices $\{\mathbf{K}^{(t)}\}_t$ defined in Eq. (2) are also p.s.d. Therefore, each kernel solution can be expressed as an inner product $\mathbf{K}^{(t)} = \mathbf{\Phi}^{(t)'}\mathbf{\Phi}^{(t)}$, with $\mathbf{\Phi}^{(t)}$ being an explicit feature map that takes data in $\mathcal{X}$ from an input space into a high dimensional Hilbert space, then Eq. (2) can be equivalently rewritten as

$$\mathbf{\Phi}^{(t+1)} = \left(\mathbf{\Phi}^{'(0)} \;\; \gamma^{\frac{1}{2}}\mathbf{P}_1\mathbf{\Phi}^{'(t)} \;\; \ldots \;\; \gamma^{\frac{1}{2}}\mathbf{P}_C\mathbf{\Phi}^{'(t)}\right)'. \tag{3}$$

From Eq. (3), it is clear that the dimensionality of $\mathbf{\Phi}^{(t)}$ is not constant and increases as t evolves. However, when γ is properly upper-bounded, the inner product $\mathbf{\Phi}^{'(t)}\mathbf{\Phi}^{(t)}$ is guaranteed to converge to a fixed-point provided that T (with $t \leq T$) is sufficiently (but not very) large. Now considering the adjacency matrices $\{\mathbf{P}_c\}_c$ and a fixed T for all images, one may update the explicit kernel maps $\{\mathbf{\Phi}_{\mathcal{I}}^{(t)}\}_{t,\mathcal{I}}$ recursively "image-by-image"; this makes the kernel map evaluation inductive and efficiently extendable to any image.

As shown through this paper, our proposed method differs from [31] in multiple aspects. Firstly, context is defined within images rather than between images.

Secondly, in contrast to [31] (which considers instead fixed handcrafted adjacency matrices), context is learned using deep kernel networks that fit a given image annotation task. Finally, different types of contexts (layerwise vs stationary, global vs classwise) are considered in order to mitigate the growth of the actual number of training parameters and to further enhance generalization.

4. Context Learning with Deep Networks

The method described in Section 3 leverages context in kernel design and is totally unsupervised as ground truth is not employed neither in kernel design nor in context definition. In order to further explore the potential of this method, we consider a supervised setting that allows us to learn more discriminating contexts as a part of kernel network design and this turns out to be more effective as shown later in experiments.

4.1. From context-aware kernels to deep context networks

Considering $\mathcal{S}_p = \{\mathbf{x}_1^p, \ldots, \mathbf{x}_n^p\}$ as a set of non-overlapping cells taken from a regular grid in the image $\mathcal{I}_p$ (see Fig. 1) and $\mathcal{X} = \cup_p \mathcal{S}_p$ as the cells from all the images; without a loss of generality, n is assumed constant for all images. We measure the similarity between any two images $\mathcal{I}_p$ and $\mathcal{I}_q$ using the convolution kernel defined as $\mathcal{K}(\mathcal{I}_p, \mathcal{I}_q) = \sum_{i,j} \kappa(\mathbf{x}_i^p, \mathbf{x}_j^q)$ over the elementary kernel κ (e.g. linear, polynomial, RBF kernel, etc.). Since κ is a p.s.d function, $\mathcal{K}$ is also p.s.d. Note that $\mathcal{K}$ captures the similarity between two images without necessarily aligning their cells, and this makes $\mathcal{K}$ translation and deformation resilient. However, most of elementary kernels focus mainly on the visual content of primitives and ignore their contextual relationships. *A more relevant kernel κ should provide*

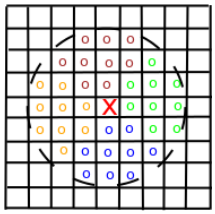

Figure 1: This figure shows the handcrafted neighborhood system with four orientations to build the context-aware kernels. Red cross in the center means a particular cell in the regular grid, and colored circles around it within a radius of 3 stand for its 4 different sectors of neighbors (i.e. $C = 4$).

high similarity values not only when primitives share close visual content but also similar context.

In order to define context between image cells, we consider $\{\mathbf{P}_c\}_{c=1}^{4}$ as a *typed* neighborhood system that captures spatial relationships through four different relative cell locations (namely "above", "below", "left" and "right"; see also Fig. 1). Given a reference cell $\mathbf{x}$; *if* $\mathbf{x}'$ *is within a predefined range from* $\mathbf{x}$ *and with a relative position typed as* c then $\mathbf{P}_{c,\mathbf{x},\mathbf{x}'} \leftarrow 1$; *otherwise* $\mathbf{P}_{c,\mathbf{x},\mathbf{x}'} \leftarrow 0$. This neighborhood system can also be made invariant to different rigid transformations including rotation and scaling[4].

Given two cells $\mathbf{x}$, $\mathbf{x}'$ in $\mathcal{X}$ and following Eqs. (2) and (3), one may rewrite the kernel definition $\mathbf{K}^{(t)}_{\mathbf{x},\mathbf{x}'}$ at iteration t as

$$\mathbf{K}^{(t)}_{\mathbf{x},\mathbf{x}'} = \phi_t(\phi_{t-1}(...\phi_1(\phi_0(\mathbf{x})))).\phi_t(\phi_{t-1}(...\phi_1(\phi_0(\mathbf{x}'))))), \tag{4}$$

with $\phi_t(\mathbf{x}) = \mathbf{\Phi}^{(t)}_{\mathbf{x}}$. According to the definition of the convolution kernel $\mathcal{K}$, the

[4]One may estimate a "characteristic" orientation and scale of a given cell using the SIFT descriptor [69], and thereby make the typed adjacency matrices of the neighborhood system $\{\mathbf{P}_c\}_c$ rotation, scale and also translation invariant.

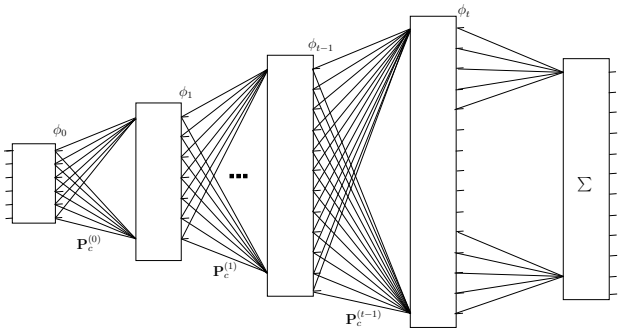

Figure 2: This figure shows the "unfolded" multi-layered kernel map network with increasing dimensionality that captures larger and more influencing contexts.

similarity between two images $\mathcal{I}_p, \mathcal{I}_q$ can be rewritten as

$$\begin{aligned} \mathcal{K}(\mathcal{I}_p, \mathcal{I}_q) &= \sum_{\mathbf{x} \in \mathcal{S}_p} \sum_{\mathbf{x}' \in \mathcal{S}_q} \mathbf{K}^{(t)}_{\mathbf{x},\mathbf{x}'} \\ &= \sum_{\mathbf{x} \in \mathcal{S}_p} \phi_t(\phi_{t-1}(...\phi_1(\phi_0(\mathbf{x})))) \\ &\quad \cdot \sum_{\mathbf{x}' \in \mathcal{S}_q} \phi_t(\phi_{t-1}(...\phi_1(\phi_0(\mathbf{x}')))). \end{aligned} \tag{5}$$

Eq. (5) defines an inner product between two recursive kernel maps. Each one corresponds to a multi-layered neural network (see Fig. 2) whose layers deliver feature maps with increasing dimensionalities that correspond to larger and more influencing contexts; a final layer is added in order to pool these feature maps through all the cells of a given image. It is easy to see that the architecture in Fig. 2 is similar to the ones widely used in deep learning with some differences; on the one hand, as discussed earlier, the number of units and the depth are respectively determined by (i) the dimensionality of the kernel maps $\{\mathbf{\Phi}^{(t)}_{\mathbf{x}}\}_t$ and (ii) the asymptotic behavior of our kernel solution; in other words, by the maximum number of iterations that guarantees the convergence of Eq. (2). In practice, we found that T-layers (with $T = 5$) are enough in order to observe this convergence on the

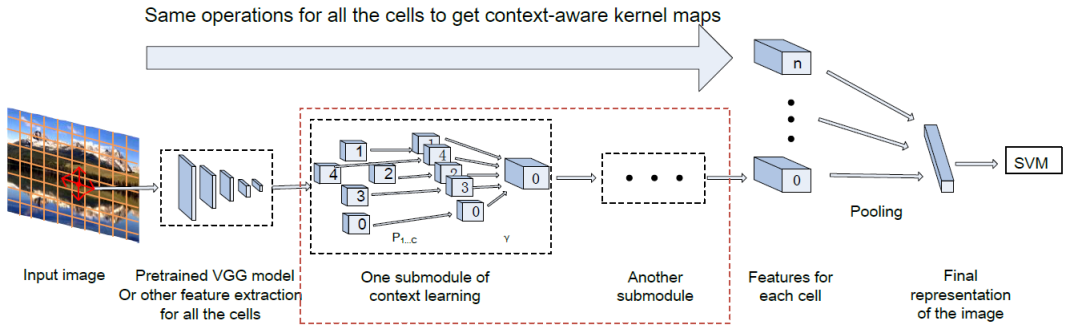

Figure 3: This figure shows the whole architecture and flowchart of our deep context learning. Given an input image (divided into cells), cells are first described using the pre-trained VGG-net. Afterwards, the context-based kernel map of a given cell (for instance cell 0), at a given iteration, is obtained by combining the kernel maps of its neighboring cells (namely cells 1, 2, 3 and 4), obtained at the previous iteration, as shown in the red dashed rectangle and also in Eq. (3). At the end of this iterative process, the kernel maps of all the cells are pooled together in order to obtain the global representation of the input image, prior to achieve its classification. Note that the network shown in the red rectangle, together with the pooling layer, correspond to the deep net shown in Fig. 2.

images. On the other hand, the parameters of this deep network correspond to the entries of the neighborhood system $\{\mathbf{P}_c\}_c$ and the largest parameters capture the most influencing contextual relationships.

Considering the limit of Eq. (2) as $\tilde{\mathbf{K}}$ and the underlying map in Eq. (3) as $\tilde{\mathbf{\Phi}}$, the convolution kernel $\mathcal{K}$ between two given images $\mathcal{I}_p$ and $\mathcal{I}_q$ can be written as

$$\mathcal{K}(\mathcal{I}_p, \mathcal{I}_q) = \langle \phi_{\mathcal{K}}(\mathcal{S}_p), \phi_{\mathcal{K}}(\mathcal{S}_q) \rangle, \tag{6}$$

with

$$\phi_{\mathcal{K}}(\mathcal{S}_p) = \sum_{\mathbf{x} \in \mathcal{S}_p} \tilde{\mathbf{\Phi}}_{\mathbf{x}}, \tag{7}$$

so each constellation of cells in an image $\mathcal{I}_p$ can be represented by a deep explicit kernel map $\phi_{\mathcal{K}}(\mathcal{S}_p)$. It is worth noticing that the maps in Eqs. (3) and (7) rely on the initial setting of $\{\mathbf{\Phi}^{(t)}\}_t$ (i.e., when $t = 0$). The latter could be obtained *exactly* for some kernels (including polynomial and histogram intersection) or approximated for others (such as RBF) using kernel principal component analysis [31].

In order to fully investigate the potential of the feature maps in Eq. (7), we consider in the subsequent section an "end-to-end" framework that learns the neighborhood system $\{\mathbf{P}_c\}_c$. The underlying context network is learned on top of another pre-trained CNN; as shown later in Section 5, this context learning process enhances further the performance of image annotation.

4.2. Deep context learning

In this section, we extend the approach described earlier in order to learn context. Considering N training images $\{\mathcal{I}_p\}_{p=1}^N$ belonging to K different classes, we define $\mathbf{Y}_k^p$ ($k \in \{1, \ldots, K\}$) as the class membership of a given image $\mathcal{I}_p$: here

$\mathbf{Y}_k^p = +1$ iff $\mathcal{I}_p$ belongs to the class k and $\mathbf{Y}_k^p = -1$ otherwise. For each class k, we train a binary SVM on top of the deep context network in order to decide about the membership of k into test images. The objective function associated to these multi-class SVMs (shown subsequently) makes it possible to define an "end-to-end" framework that learns *not only* the SVM parameters but also the weights associated to the neighborhood system $\{\mathbf{P}_c\}_c$.

Considering the pairs of training images and their labels $\{(\phi_{\mathcal{K}}(\mathcal{S}_p), \mathbf{Y}_k^p)\}_p$, the loss associated to the multi-class SVMs is defined as

$$\min_{\{w_k\}_k, \{\mathbf{P}_c\}_c} \sum_{k=1}^{K} \frac{1}{2}||w_k||^2 + C_k \sum_{p=1}^{N} \max\left(0, 1 - \mathbf{Y}_k^p f_k(\mathcal{S}_p)\right), \tag{8}$$

with $C_k > 0$, $f_k(\mathcal{S}_p) = w_k' \phi_{\mathcal{K}}(\mathcal{S}_p)$ and w_k the training parameters of f_k. The first term, of the above objective function, is an ℓ_2 regularization that seeks to maximize the margins of $\{f_k\}_k$ while the second criterion corresponds to the hing loss. Eq. (8) makes it possible to learn at least two variants of contexts $\{\mathbf{P}_c\}_c$: global and classwise. In what follows, we first describe how to learn global contexts and then we update our scheme in section 4.4 in order to make these contexts class-dependent.

As it is difficult to *jointly* optimize the two sets of parameters $\{w_k\}_k, \{\mathbf{P}_c\}_c$, we adopt an alternating optimization procedure: at each iteration, we fix $\{\mathbf{P}_c\}_c$ and we learn $\{w_k\}_k$ and then vice-versa. When fixing $\{w_k\}_k$, the parameters $\{\mathbf{P}_c\}_c$ are learned using backpropagation and gradient descent. Let $\mathbb{1}_{\{\}}$ denote the indicator function; considering the kernel maps $\{\phi_{\mathcal{K}}(\mathcal{S}_p)\}_p$ of training images and the gradient of the loss (8) — also denoted as E — w.r.t the output of the context network

$$\frac{\partial E}{\partial \phi_{\mathcal{K}}} = -\sum_{p=1}^{N} \sum_{k=1}^{K} C_k \mathbf{Y}_k^p w_k \mathbb{1}_{\{1 - \mathbf{Y}_k^p w_k' \phi_{\mathcal{K}}(\mathcal{S}_p) > 0\}}, \tag{9}$$

we apply (i) the chain rule in order to backpropagate $\{\frac{\partial E}{\partial \mathbf{P}_c}\}_c$ and then (ii) gradient descent to update the neighborhood system $\{\mathbf{P}_c\}_c$. When fixing $\{\mathbf{P}_c\}_c$, the parameters $\{w_k\}_k$ of the primal form of $\{f_k\}_k$ are given by

$$w_k = \sum_{p=1}^{N} \mathbf{Y}_k^p \alpha_k^p \phi_{\mathcal{K}}(\mathcal{S}_p), \tag{10}$$

here $\{\alpha_k^p\}$ correspond to the parameters of the dual form of Eq. (8) trained using LIBSVM [70]. Note that this iterative optimization procedure is performed till convergence, i.e., when the values of the two sets of parameters $\{w_k\}_k, \{\mathbf{P}_c\}_c$ remain stable through iterations and this is observed in less than 100 iterations in practice.

4.3. Layerwise vs stationary contexts

In Section 4.2, context matrices $\{\mathbf{P}_c\}_c$ are layerwise learned and this increases the total number of training parameters. Actually, layerwise contexts are not totally independent; for instance, one may infer, using transitive closure, high-order contexts from low-order ones. Considering stationary context matrices $\{\mathbf{P}_c\}_c$ through all the layers (written for short as $\mathbf{P}$), the underlying gradient is obtained using the chain rule as

$$\frac{\partial E}{\partial \mathbf{P}} \;=\; \frac{\partial E}{\partial \phi_{\mathcal{K}}} \frac{\partial \phi_{\mathcal{K}}}{\partial \mathbf{P}}, \tag{11}$$

with the left-hand side term being defined in Eq. (9). For stationary $\mathbf{P}$, the right-hand side term can be expanded by averaging (or equivalently summing) the gradients of all the instances of $\mathbf{P}$ through layers $t \in \{1, \ldots, T\}$, and using again the chain rule, as

$$\frac{\partial \phi_{\mathcal{K}}}{\partial \mathbf{P}} \;=\; \sum_{t=1}^{T} \frac{\partial \phi_{\mathcal{K}}}{\partial \phi_T} \left(\frac{\partial \phi_T}{\partial \phi_{T-1}} \cdots \frac{\partial \phi_{t+1}}{\partial \phi_t} \right) \frac{\partial \phi_t}{\partial \mathbf{P}}. \tag{12}$$

It is easy to see that the general form of the "between parentheses" term in Eq. (12) reduces to 1 when $t = T$ and $\frac{\partial \phi_T}{\partial \phi_{T-1}}$ when $t = T - 1$. Since $\left(\frac{\partial \phi_T}{\partial \phi_{T-1}} \dots \frac{\partial \phi_{t+1}}{\partial \phi_t}\right)$ vanishes very quickly as T increases, we add skip connections between each layer and the ouput of our deep network. With this slight update of the architecture, the gradient flies back from the output to each layer t, not only through the intermediate layers but also *directly*. Hence

$$\frac{\partial \phi_{\mathcal{K}}}{\partial \mathbf{P}} = \sum_{t=1}^{T} \frac{\partial \phi_{\mathcal{K}}}{\partial \phi_T} \left(\frac{\partial \phi_T}{\partial \phi_{T-1}} \dots \frac{\partial \phi_{t+1}}{\partial \phi_t}\right) \frac{\partial \phi_t}{\partial \mathbf{P}} + \frac{\partial \phi_{\mathcal{K}}}{\partial \phi_t} \frac{\partial \phi_t}{\partial \mathbf{P}}. \tag{13}$$

In practice, as the right-hand side term in Eq. (13) is equivalent to the left-hand side one (when t reaches T) and is several orders of magnitude larger (when $t < T$), we consider a surrogate form which keeps only the dominant direction of the gradient in Eq. (13), i.e.,

$$\frac{\partial \phi_{\mathcal{K}}}{\partial \mathbf{P}} \approx \sum_{t=1}^{T} \frac{\partial \phi_{\mathcal{K}}}{\partial \phi_t} \frac{\partial \phi_t}{\partial \mathbf{P}}. \tag{14}$$

Finally, context parameters are updated by gradient descent using the shared gradients in Eqs. (11) and (14) for all the instances of $\mathbf{P}$ through all the layers; note that the initial values in $\mathbf{P}$ are also shared through all the layers so maintaining shared gradients, across epochs, guarantees stationary context at convergence and reduces the actual number of training parameters.

4.4. Global vs classwise contexts

As shown subsequently in experiments (see section 5), the impact of context learning is already well established with a *global* neighborhood system. However, this gain could further be enhanced if one considers instead a classwise context. Indeed, the rational resides in the fact that scenes may have different structures depending on their concepts. *Generally speaking, the notion of sharing intermediate*

representations in deep nets is highly valid when considering the shared intrinsic properties of the learned object categories, however, context is extrinsic, so making the latter class-dependent is complementary and rather more appropriate.

In order to investigate the validity of this conjecture, we update the model slightly: as the objective function E can be written as the sum of classwise losses, one may split its gradient into K terms each one corresponding to a particular class. Then, using these K gradients, we update the underlying neighborhood systems (now indexed by their classes and denoted as $\{\mathbf{P}_c^k\}_{c,k}$). This results into K different context network maps[5] $\{\phi_{\mathcal{K}}^k(.)\}_k$ used to evaluate the underlying SVMs. Note that during the learning process, we adopt a warm-start in order to accelerate the convergence of our iterative algorithm. Indeed, the learned parameters $\{\mathbf{P}_c^k\}_{c,k}$ are initially set using the global learned context $\{\mathbf{P}_c\}_c$ and updated at each iteration in K-steps; each step includes (i) a backpropagation of $\{\frac{\partial E}{\partial \mathbf{P}_c^k}\}_c$ through the layers of the k^{th} deep context network using the chain rule and (ii) a gradient descent in order to update the underlying classwise neighborhood system $\{\mathbf{P}_c^k\}_c$. Once all these contexts learned (and fixed), the parameters $\{w_k\}_k$ are updated as shown earlier, in section 4.2, with the only difference that maps used in Eq. (10) are now class-dependent.

5. Experimental Results

In this section, we evaluate the performance of our deep context-aware kernel network on image annotation, using different variants of contexts. Image annotation is a multi-task classification problem; given an image $\mathcal{I}$, the goal is to assign

[5] also indexed by their classes.

a list of concepts (a.k.a. keywords) to $\mathcal{I}$ depending on the values of the underlying classifiers. We conducted the experiments on three challenging image annotation benchmark as follows:

- ImageCLEF [71]: it consists of more than 250k images belonging to 95 concepts and is split into training, dev and test data; we only consider the dev set, which includes 1,000 images equally split between training and testing, as the ground-truth is released on this dev set only.

- Corel5k [72]: it consists of 4,999 annotated images with a vocabulary of up to 200 concepts and is split into two subsets: 4,500 images for training and the rest for testing. Following the standard protocol in [72], each test image is annotated with up to 5 keywords.

- NUSWIDE [73]: it includes 269,648 images collected from Flickr, and most of them are manually annotated with an average of 2.4 keywords taken from 81 concepts in total. Following [74], 150k images are used for training (among them 10% for validation), and the remaining for testing. Each test image is annotated with at most 3 keywords.

Performances are measured using the mean precision and recall over keywords (respectively denoted as $\mathbf{P}$ and $\mathbf{R}$), the F-scores (referred to as $\mathbf{F}$) and the number of keywords with non-zero recall (denoted as $\mathbf{N}_+$) both at the sample and concept levels (denoted respectively as MF-S and MF-C) as well as the mean Average Precision (mAP) for ImageCLEF; higher values of these measures imply better performances. In what follows, we first study the impact of the proposed network w.r.t. different settings on ImageCLEF, then we compare the optimal setting of our network architecture against the related work.

5.1. Network analysis on ImageCLEF

In order to study the impact of the proposed network architecture w.r.t. different settings, we show experiments on ImageCLEF. Images in this dataset are rescaled to their median dimension (of 400×500 pixels), and then partitioned into regular grids of $W \times H$ cells[6]. Each cell is encoded using two types of features: *handcrafted* and *learned*; the former include the bag-of-word (BoW) histogram (evaluated on dense SIFT features and a code-book of 500 words) while the latter include deep VGG-net features. More precisely, the used VGG-net corresponds to "imagenet-vgg-m-1024" [75]; this model is pretrained on ImageNet and consists in five convolutional and three fully-connected layers. The outputs of the second fully-connected layer are used to describe the content of the cells in the regular grids.

In order to learn the context-aware kernel networks, we consider four different types of geometric relationships (see Fig. 1) and different context-free kernel maps including linear (LIN), polynomial (POLY) and histogram intersection (HI). The explicit maps of linear and polynomial kernels can be exactly obtained using identity and tensor product respectively while for histogram intersection these maps can be approximated using decimal-to-unary projections [31]. Using the above setting, we learn maximum margin classifiers on top of the deep context-aware kernel networks, and predict the presence of concepts into test images depending on the scores of the underlying classifiers.

Impact of network depth. As already discussed, the depth of the network is defined by the number of iterations T in Eqs. (2) and (3); larger values of T imply

[6] as shown later, different granularities are considered for W and H.

	2-layer-net	3-layer-net	4-layer-net	5-layer-net
RE	23.97	4.08	0.99	0.25

Table 1: This table shows the relative errors (in %) of different layers in the deep context-aware kernel networks. These performances correspond to "Layerwise & Global" (L-Global) learned contexts.

		MF-S	MF-C	mAP	runtime
2-layer	Handcrafted CA	40.93	25.20	48.78	-
	L-Global CA	42.64	25.04	50.98	45.31
3-layer	Handcrafted CA	40.74	25.28	48.51	-
	L-Global CA	44.22	26.43	52.02	235.10
4-layer	Handcrafted CA	41.03	25.55	48.99	-
	L-Global CA	44.11	26.57	52.43	1021.71
5-layer	Handcrafted CA	40.93	25.54	48.68	-
	L-Global CA	44.27	25.12	52.15	7729.90

Table 2: This table shows the performance of handcrafted vs. learned deep context-aware kernel networks for a grid of 8×10 cells; we compare handcrafted vs. "layerwise & global" (L-Global) contexts on ImageCLEF. The MF-S/MF-C/mAP measures are provided (in %) and runtime performances (in *s*) for one "forward-backward" iteration of backpropagation.

deeper and convergent networks but the underlying feature maps become high dimensional. In contrast, low values of T make the network relatively shallow and compact but not convergent. Hence, the appropriate setting of the network depth (i.e., T) should tradeoff *convergence* and *compacity*; this also impacts the number of training parameters (thereby generalization) and the computational efficiency of the resulting network.

In practice, we measure the convergence of the network, between two consecutive

Grid	1×1	2×2	4×5	8×10
Forward time				

Table 3: This table shows forward runtime (in *s*) of 3-layer network including feature extraction by VGG-net and network forward.

layers, using the following relative error (RE) criterion

$$\mathrm{RE}^{(t)} = \frac{1}{|\mathcal{S}_p||\mathcal{S}_q|} \sum_{\mathbf{x} \in \mathcal{S}_p} \sum_{\mathbf{x}' \in \mathcal{S}_q} \frac{|\mathbf{K}^{(t)}_{\mathbf{x},\mathbf{x}'} - \mathbf{K}^{(t-1)}_{\mathbf{x},\mathbf{x}'}|}{|\mathbf{K}^{(t)}_{\mathbf{x},\mathbf{x}'} + \mathbf{K}^{(t-1)}_{\mathbf{x},\mathbf{x}'}|}, \tag{15}$$

this measure is evaluated using the initial normalized neighborhood system, i.e., $\{\mathbf{P}_{c,\mathbf{x},\mathbf{x}'} / \sum_{\mathbf{x}'' \in \mathcal{N}_c(\mathbf{x})} \mathbf{P}_{c,\mathbf{x},\mathbf{x}''}\}_c$ with $\mathcal{N}_c(\mathbf{x})$ being a subset of neighbors of $\mathbf{x}$ (a regular grid of 8×10 cells is used, and $C = 4$, $r = 1$). Tab. 1 shows this error as the network becomes deeper; it is clear that convergence is obtained for reasonably (not very) deep networks. Tab. 2 shows the underlying performances with handcrafted and learned contexts. From these results, we observe that when context is handcrafted, the impact of the depth is marginal while for learned context this impact is noticeable. Runtime performances are also reported and correspond to the cost of one "forward-backward" iteration during backpropagation; these performances are obtained on a workstation with 4 Intel-Xeon CPUs of 3.2GHz and 64G memory. It is clear that the learning process becomes cumbersome as the network gets deeper; hence, in order to make the learning reasonably tractable, we consider in the remainder of this paper a network architecture with three layers. We also show, in Tab. 3, the run-time (including feature extraction with VGG-net and the forward steps) of 3-layer context networks for different cell grids.

Impact of the neighborhood system. In order to model different context granularities, we consider multiple instances of regular grids (with $W \times H$ in $\{2 \times 2, 4 \times$

$5\}$) and different settings of the radius r in the neighborhood system [7]. It is clear that r is constrained by W and H; for instance, when the latter are equal to 1, the radius r cannot exceed 1 so this particular configuration (shown in the first row of Tab. 5) corresponds to a holistic (*one-cell-grid*) context-free (CF) network. From Tab. 5, we first observe a global negative impact of finer grids on the performances of CF networks on both BoW and VGG features. Indeed, finer cells — deprived from context — are not sufficiently discriminating and hence powerless to capture the semantic of images. However, context-aware (CA) networks, even-though applied to finer cells, allow us to recover and enhance the discrimination power of these cells and also to substantially overtake the original CF performances by a significant margin, especially when context is learned; this gain is consistently *the highest through all the original kernel maps*, particularly when cells are encoded with VGG and when $H \times W = 2 \times 2$ (with r being necessarily set to 1). For larger $H \times W$, we observe a moderate (and sometimes a negative) impact of larger r on performances. We conjecture that, as the radius gets larger, cells in the neighborhood system $\{\mathcal{N}_c(.)\}_c$ are more and more densely connected; as a result, it becomes more difficult to learn relevant context from a huge combinatorial set of possible relationships in $\{\mathcal{N}_c(.)\}_c$.

Impact of stationary and classwise contexts. In the following experiments, the initial weights of classwise and stationary context networks are taken from the learned global context. Tab. 5 show the results of different context variants using several kernel map initializations and features, while Tab. 4 shows the statistical dependencies between different network pairs. From all these results, we ob-

[7] We empirically found that grids (with more than 8×10 cells) degrade performances, so larger grids were not investigated in this work.

Comparison	MF-S	MF-C	mAP
L-Global CA vs. Handcrafted CA	100	72.2	100
S-Global CA vs. L-Global CA	83.3	88.8	50
L-Classwise CA vs. L-Global CA	100	100	33.3
S-Classwise CA vs. S-Global CA	100	100	66.7
S-Classwise CA vs. L-Classewise CA	38.9	38.9	88.9

Table 4: This table shows the statistical dependencies (in %) between different variants of learned and handcrafted contexts. For each comparison ("A vs. B"), we measure the percentage of times "A is not worse than B" in a pool of 18 runs (using evaluation measures of Tab. 5).

serve that the learned context networks overtake the handcrafted ones over all the settings in MF-S/mAP and 75% of the settings in MF-C. We also observe that stationary context networks are able to further enhance the performances of the global context networks for most of the settings, however the gain in mAP is less marked. In the subsequent results, the learned classwise context networks boost further the performances for most of the settings compared to the other (global) variants; for instance, MF-S/MF-C/mAP values raise from 44.20/25.91/52.51 to 45.35/27.31/52.60 (using polynomial kernel map with BoW features, $r = 3$ and $W \times H = 4 \times 5$) and from 55.01/42.56/66.45 to 56.41/44.87/66.67 (using polynomial kernel map with VGG features, $r = 3$ and $W \times H = 4 \times 5$). In sum, *learned classwise context networks are more positively influencing than learned global ones, whether context is stationary or not* through different layers.

5.2. Qualitative results

In what follows, we show the experimental results of context-aware kernel networks with different initial kernel maps on ImageCLEF and Corel5k (see Tab. 5 and Tab. 6). "LIN", "POLY" and "HI" stand respectively for linear, polynomial and histogram intersection functions as initial kernel map, while r corresponds

Cells	r	Method	BoW features			VGG features		
			LIN	POLY	HI	LIN	POLY	HI
1×1	-	CF	40.24/24.21/49.31	43.76/27.26/52.27	44.16/25.76/53.78	52.84/42.30/66.75	55.70/44.69/71.02	54.13/43.66/70.52
2×2	-	CF	39.55/24.36/47.40	41.61/24.73/50.27	44.61/27.30/53.71	53.82/42.32/68.04	56.82/44.41/70.66	55.15/43.69/69.85
	1	Handcrafted CA	42.23/25.23/51.24	42.93/26.13/52.33	45.93/28.39/54.74	54.47/43.24/69.56	56.53/45.23/71.43	55.18/42.97/70.43
		L-Global CA	43.50/26.05/51.68	43.48/26.41/52.93	46.10/27.22/55.15	56.01/45.26/70.49	58.29/46.76/72.06	58.26/44.99/71.54
		S-Global CA	43.59/26.23/51.69	43.78/26.73/52.94	46.51/28.35/54.75	56.07/45.32/70.45	58.58/46.89/72.15	58.43/45.48/71.64
		L-Classwise CA	44.34/27.78/51.52	45.17/28.21/52.91	46.22/27.31/55.16	57.36/47.36/70.35	58.87/47.98/72.08	58.88/46.51/71.54
		S-Classwise CA	44.15/27.21/51.72	44.79/27.84/52.92	48.61/29.60/55.29	57.69/47.24/70.44	58.97/47.77/72.14	58.60/46.03/71.68
4×5	-	CF	40.37/24.89/48.01	42.27/26.56/50.12	43.39/28.53/52.40	51.24/38.34/63.21	52.55/39.96/64.71	51.33/37.96/63.64
	1	Handcrafted CA	41.67/26.05/50.95	42.88/26.40/51.60	44.94/29.22/53.50	51.71/38.97/63.98	53.09/39.66/65.01	51.61/38.94/64.15
		L-Global CA	44.74/26.85/52.86	44.39/26.04/52.82	45.77/27.02/54.74	53.55/41.09/65.36	54.80/41.98/66.11	53.26/39.11/65.37
		S-Global CA	43.81/27.30/52.14	44.63/26.66/52.74	46.09/28.34/54.60	53.57/41.43/65.43	54.63/42.32/66.22	53.70/40.85/65.52
		L-Classwise CA	46.16/28.04/52.48	46.03/27.49/52.70	47.59/28.63/54.05	55.96/44.25/65.33	56.66/44.13/66.50	55.50/41.33/64.96
		S-Classwise CA	45.64/29.48/52.56	46.34/29.10/53.06	47.79/29.97/54.79	55.91/43.44/65.14	57.06/44.84/66.72	55.60/42.62/66.04
	3	Handcrafted CA	41.88/26.26/50.97	43.31/26.98/51.78	44.78/29.42/53.78	51.89/39.35/64.45	52.85/39.84/65.11	51.70/38.36/64.36
		L-Global CA	43.81/27.97/51.80	44.20/25.91/52.51	45.81/28.65/54.91	54.55/40.67/65.35	55.01/42.56/66.45	53.65/39.51/65.44
		S-Global CA	44.00/28.02/51.81	44.59/27.31/52.63	46.09/28.72/54.72	54.14/40.66/65.27	55.16/42.26/66.30	53.89/40.12/65.37
		L-Classwise CA	44.90/28.78/51.72	45.35/27.31/52.60	47.65/30.80/54.75	55.63/42.84/65.34	56.41/44.87/66.67	55.24/42.51/65.31
		S-Classwise CA	44.42/28.72/51.73	44.67/27.39/52.62	46.75/29.57/54.82	55.14/42.23/65.39	55.49/42.68/66.30	55.04/41.92/65.42

Table 5: The performance (in $\%$) of different variants of context-aware kernel networks on ImageCLEF. The triple $\cdot/\cdot/\cdot$ stands for MF-S/MF-C/mAP.

to the radius of the disk that supports context. In these results, "L-Global", "S-Global", "L-Classwise" and "S-Classwise" stand respectively for "Layerwise & Global", "Stationary & Global", "Layerwise & Classwise" and "Stationary & Classwise" contexts. For Corel5k dataset, we also rescale images to the median dimension of 400×500 pixels, and partition each image into a regular grid of 2×2 and 4×5 cells. Each cell is again described with the same features used on ImageCLEF. Since categories are highly imbalanced in Corel5k, we learn ensembles of binary SVMs; for each concept, ten SVMs are trained on all the positive data (belonging to that concept) and a random subset (from the remaining negative data) whose cardinality is three times larger than the positive set. The global decision score on a given test image, w.r.t. a given concept, is taken as the average score of the ten underlying SVMs.

Cells	r	Method	BoW features			VGG features		
			LIN	POLY	HI	LIN	POLY	HI
1×1	-	CF	22.99/15.06/13.05/122	26.90/18.47/16.26/147	22.69/18.62/14.38/131	42.26/29.10/28.15/179	46.91/31.04/30.15/191	46.49/30.70/29.41/195
2×2	-	CF	23.36/16.21/13.75/128	26.12/17.98/14.89/137	22.76/16.94/13.33/125	44.75/35.25/31.88/190	43.96/34.91/31.49/188	42.88/33.77/29.42/186
	1	Handcrafted CA	24.66/17.53/14.60/129	25.68/18.46/15.17/137	24.66/17.95/15.04/131	43.48/35.71/32.07/189	43.70/34.95/31.56/186	44.15/33.85/30.03/188
		L-Global CA	24.88/17.77/14.83/135	25.37/18.77/15.20/137	24.51/19.26/15.73/131	43.19/35.73/31.94/189	44.65/35.76/32.40/189	42.73/33.89/29.64/186
		S-Global CA	26.29/18.74/15.88/138	26.10/19.53/16.28/141	25.58/19.89/16.52/138	44.70/36.66/33.07/193	45.50/36.32/32.78/194	45.28/34.63/31.02/194
		L-Classwise CA	26.42/18.45/15.75/138	25.35/19.25/15.76/138	26.11/20.42/16.71/142	43.74/36.41/32.59/190	45.31/36.36/32.76/192	43.51/34.93/30.37/188
		S-Classwise CA	25.97/18.68/16.01/138	25.79/19.26/16.17/140	25.58/19.92/16.56/138	44.18/36.86/32.88/192	45.69/37.05/33.16/193	45.73/34.84/31.08/195
4×5	-	CF	21.71/15.95/13.01/126	22.30/17.55/13.82/130	20.61/15.39/12.42/120	43.44/31.56/29.50/184	43.86/32.27/29.98/183	42.12/31.05/29.04/182
	1	Handcrafted CA	22.93/17.02/14.04/129	23.49/18.26/14.81/132	22.30/16.72/13.87/123	44.08/32.03/30.03/182	43.70/32.24/30.03/182	41.93/31.28/29.21/183
		L-Global CA	23.54/17.90/14.71/131	23.40/18.19/14.72/132	23.29/17.39/14.95/125	43.45/32.79/30.52/182	43.95/33.56/31.06/185	42.02/31.28/29.21/183
		S-Global CA	25.84/18.66/16.23/139	24.35/18.83/15.44/134	25.58/19.04/16.36/139	44.20/33.00/30.73/186	44.25/34.16/31.34/186	43.07/31.60/29.45/187
		L-Classwise CA	26.05/18.68/16.14/140	26.58/19.38/17.09/142	24.44/18.62/15.70/137	44.44/33.45/31.21/187	44.23/33.69/31.16/187	42.71/31.18/29.29/184
		S-Classwise CA	26.02/18.72/16.35/139	23.70/18.68/15.45/133	25.70/18.73/16.22/139	44.08/33.21/30.61/188	44.26/33.68/31.12/186	42.83/31.51/29.29/186
	3	Handcrafted CA	22.87/17.64/14.42/127	23.11/18.70/15.11/131	23.00/16.99/14.19/126	43.87/33.03/30.47/184	43.31/32.02/29.76/182	42.29/31.03/29.01/185
		L-Global CA	25.12/19.55/16.39/136	23.72/19.55/15.71/136	24.98/18.51/15.42/135	43.45/34.10/31.07/183	43.22/32.97/30.47/182	41.25/32.45/29.25/182
		S-Global CA	25.02/19.25/16.12/137	24.82/19.66/16.16/140	25.37/18.42/15.58/135	44.21/33.08/30.52/185	43.08/34.04/30.95/184	43.00/31.41/29.48/185
		L-Classwise CA	25.27/19.26/16.36/137	25.30/20.02/16.79/141	25.46/18.64/15.74/136	43.29/34.81/31.25/183	43.82/33.51/30.95/184	41.28/32.61/29.26/182
		S-Classwise CA	24.74/19.13/16.05/136	25.16/19.73/16.47/141	25.56/18.71/15.70/137	44.00/32.92/30.57/185	43.60/34.15/31.06/185	42.96/31.37/29.38/185

Table 6: The performance (in $\%$) of different deep context networks on Corel5k. A quadruplet $\cdot/\cdot/\cdot/\cdot$ stands for $\mathbf{R}/\mathbf{P}/\mathbf{F}/\mathbf{N}_+$.

On Corel5k, we observe for most of the settings in Tab. 6, a clear gain of different context-nets when trained on top of BoW features; indeed, the gain in $\mathbf{R/P/F/N}_+$ reaches $1.0/0.7/1.1/2$ points for layerwise global contexts, $2.1/1.9/1.8/14$ for layerwise classwise contexts, $3.3/2.3/2.5/16$ for stationary global ones and $3.4/2.0/2.4/16$ for stationary classwise contexts, all obtained using histogram intersection initial map and a grid of 4×5 cells with $r = 1$. We also observe a clear gain when using VGG features; this gain reaches $1.0/0.8/0.8/3$ points for layerwise global context, $1.6/1.4/1.2/6$ for layerwise classwise one, $1.8/1.4/1.2/8$ for stationary global context and $2.0/2.1/1.6/7$ for stationary classwise context, all obtained using polynomial initial map and a grid of 2×2 cells with $r = 1$. It is worth noticing that the gain of classwise context is not always consistent due to the large number of training parameters (w.r.t. the size of training data) compared to stationary context which is relatively less subject to overfitting as its parameters are shared.

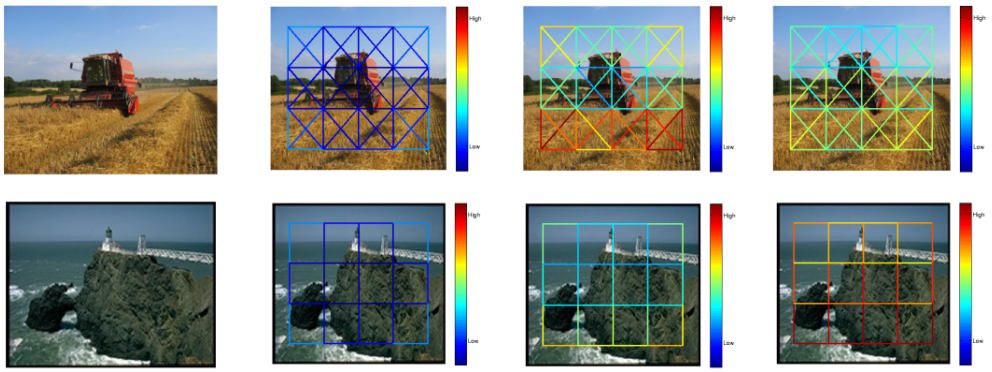

Figure 4: This figure shows different context variants on two examples taken from ImageCLEF (top) and Corel5k (bottom); original images (first column), handcrafted contexts (second column), learned global context in the first layer (third column), learned global context in the second layer (fourth column). These results are obtained using a linear kernel map initialization and VGG features with $r = 1$ and a grid of 4×5 cells on the two datasets. Handcrafted context matrices are obtained by normalizing each row (cell) in these matrices by the number of its spatial neighbors, that's why the cells in the four corners have larger values. In all these results, the importance of the context of a given cell is shown with colored connections to its neighbors using a particular color-map; warmer colors (close to red) correspond to important relationships while the cooler ones are less important (better to zoom the PDF version).

In order to visually analyze the learned contexts, we accumulate and display the weights involved in $\{\mathbf{P}_c\}_c$ (following the spatial support shown in Fig. 1). We investigate two aspects: the interpretation of context evolution through layers and the interpretation of different context variants. Fig. 4 (second, third and fourth columns) respectively describe the handcrafted and the learned $\{\mathbf{P}_c\}_c$ in the first and the second layers of the underlying network; values in $\{\mathbf{P}_c\}_c$ are superimposed on images from ImageCLEF and Corel5k. This display shows that *first* layer context is less meaningful than the *second* layer one, possibly resulting from the fact that the *latter* captures higher-order and more influencing spatial

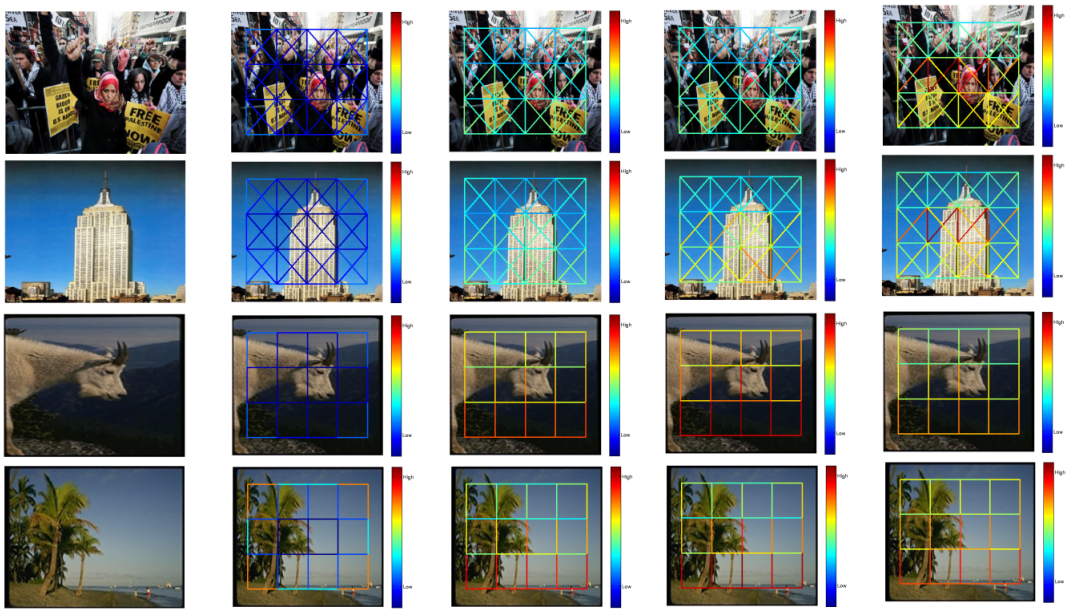

Figure 5: This figure shows original images (first column), handcrafted (second column), learned global (third column), learned classwise contexts (fourth column) and learned stationary (fifth column) using VGG features with $r = 1$ and grids of 4×5 cells for ImageCLEF (the top two rows) and Corel5k (the bottom two rows) databases; linear kernel maps are used on these two datasets. For stationary context, it is clear that the importance of the underlying prominent area is strengthened compared to layerwise context and for classwise context, the contribution of background is weakened while the underlying prominent area boosted. For instance, regarding the concepts "cityscape" and "mountain" (shown in the second and third rows respectively), it is clear that the areas around these concepts are strengthened compared to the other areas in these scenes (better to zoom the PDF version).

Kernel	MF-S	MF-C	mAP
GMKL([76])	41.3	24.3	49.1
2LMKL([77])	45.0	25.8	54.0
LDMKL ([27])	47.8	30.0	58.6
Handcrafted CA ([31])	56.5	45.2	71.4
L-Global CA (proposed)	58.3	46.8	72.1
S-Global CA (proposed)	58.6	46.9	72.2
L-Classwise CA (proposed)	58.9	48.0	72.1
S-Classwise CA (proposed)	59.0	47.8	72.1

Table 7: This table shows comparison of performances (in %) between different kernel-learning methods on ImageCLEF. The best results, on both global and classwise contexts, are obtained using polynomial kernel and VGG features on a grid of 2×2 cells with $r = 1$.

and structural relationships compared to the *former* which relies only on immediate neighbors in $\{\mathbf{P}_c\}_c$. Besides, due to the chain rule, the gradient w.r.t. the first layer context is more quickly vanishing and this makes its evolution through the iterations of back-propagation relatively less important compared to the gradient in the second layer. Fig. 5 is a visualization of handcrafted vs. learned contexts (including layerwise, classwise and stationary) taken from the second layer of their respective networks superimposed on two images from ImageCLEF and Corel5k; it is clear that when contexts are learned, some spatial cell-relationships are *amplified* while others are *attenuated* and this reflects their importance in the underlying image classification tasks.

5.3. Comparisons w.r.t. other methods

In what follows, we compare our method against the related work on ImageCLEF, Corel5K and NUS-WIDE. We show these comparisons using the best

Learned Input feat.	Method	Setting	Classifier	Kernel learning	Context	$\mathbf{R}$	$\mathbf{P}$	$\mathbf{N}_+$
No	CRM [78]	Generative	-	No	No	19	16	107
	InfNet [79]	Generative	-	No	No	24	17	112
	JEC-15 [8]	Discriminative	KNN	No	Yes (fixed)	33	28	140
	FT DMN [28]	Discriminative	SVM	Yes	No	35	21	168
	3-layer DKN [27]	Discriminative	SVM	Yes	No	38	26	158
	TagProp-σML [12]	Discriminative	KNN	No	Yes (learned)	42	33	160
	wTKML [80]	Discriminative	SVM	Yes	Yes (fixed)	42	21	173
	KSVM-VT [81]	Discriminative	SVM	No	Yes (fixed)	42	32	179
	LDMKL [27]	Discriminative	SVM	Yes	Yes (fixed)	44	29	179
	SKL-CRM [82]	Generative	-	Yes	No	46	39	184
	2PKNN-GSR [83]	Discriminative	KNN	No	No	43.5	39.5	189
	2PKNN-ML [13]	Discriminative	KNN	No	Yes (learned)	46	44	191
	MLDL [84]	Discriminative	KNN	No	Yes	49	45	198
	MVRSC (Visual + Textual) [85]	Generative	KNN	No	Yes	54.3	42.9	-
	L-Global CA (proposed)	Discriminative	SVM	Yes	Yes (learned)	25	19	137
	S-Global CA (proposed)	Discriminative	SVM	Yes	Yes (learned)	26	20	141
	L-Classwise CA (proposed)	Discriminative	SVM	Yes	Yes (learned)	26	20	142
	S-Classwise CA (proposed)	Discriminative	SVM	Yes	Yes (learned)	26	20	138
Yes	ResNet [45]	Discriminative	SVM	No	No	35	22	161
	FT DMN [28]	Discriminative	SVM	Yes	No	38	23	169
	CNN (Caffe-Net) [47]	Discriminative	LR	No	No	41	32	166
	3-layer DKN [27]	Discriminative	SVM	Yes	No	43	25	180
	LNR+TagProp-σSD (ResNet + Textual) [86]	Discriminative	KNN	No	No	49	39	181
	LNR+2PKNN (ResNet + Textual) [86]	Discriminative	KNN	No	Yes (learned)	52	43	192
	CCA-KNN (VGG-16 + Textual) [47]	Discriminative	KNN	No	No	52	42	201
	L-Global CA (proposed)	Discriminative	SVM	Yes	Yes (learned)	45	36	189
	S-Global CA (proposed)	Discriminative	SVM	Yes	Yes (learned)	46	36	194
	L-Classwise CA (proposed)	Discriminative	SVM	Yes	Yes (learned)	45	36	192
	S-Classwise CA (proposed)	Discriminative	SVM	Yes	Yes (learned)	46	37	193

Table 8: Extra comparison of the proposed deep context-aware kernel networks w.r.t. the related works on Corel5k (in % for $\mathbf{R}$ and $\mathbf{P}$). In this table, FT stands for Fine-Tuned. Results corresponding to BoW features (referred to as "proposed" in the first part of the table) are obtained using (i) the polynomial kernel map on a grid of 2×2 cells with $r = 1$ for Global contexts and (ii) the HI kernel map on a grid of 2×2 cells with $r = 1$ for Classwise contexts. Results corresponding to deep VGG features (referred to as "proposed" in the second part of the table) are obtained using the polynomial kernel map on a grid of 2×2 cells with $r = 1$ for all the learned context variants.

setting of our context network (i.e., based on a grid of 2×2 cells and $r = 1$, as shown in Tables. 5 and 6).

ImageCLEF dataset: We compare the performance of the proposed approach to other kernel-based methods in Tab. 7. This comparison involves the most related kernel design techniques including: general multiple kernel learning (GMKL), two-layer multiple kernel learning (2LMKL) and Laplacian-based semi-supervised learning on 3-layer kernel network (LDMKL). The proposed classwise context-aware kernel networks obtain the best performance. The first row of Fig. 6 shows some image instances and their annotation results respectively using context-free, handcrafted and learned (layerwise vs. stationary and global vs. classwise) context networks.

Corel5k dataset: Tab. 8 shows a comparison of the proposed approach against the related work, using both handcrafted and learned deep feature settings. Further informations about the learning settings, classifiers, relation to kernel learning and context modeling are also given. From these results, we observe that the proposed approaches are competitive in comparison to the kernel-based and deep learning approaches. These comparative methods (namely LDMKL [27], wTKML [80], TagPop σML [12], SKL-CRM [82], 2PKNN-GSR [83], MLDL [84] MVRSC [85]) rely on a battery of handcrafted features (GIST, 8 types of BoWs, etc.). Among these methods, k-NN in TagPop σML [12], 2PKNN-ML [13] and Laplacian operators in DMKL [27] are used for context modeling, MLDL [84] considers label consistency and partial-identical label embedding in the multi-label dictionary learning, MVRSC [85] adopts spectral clustering in the multiple feature and semantic space. In contrast, our method — in spite of using a single BoW — is still competitive; this is essentially due to the discrimination power of the

learned contexts which catch-up with these extensively-tuned handcrafted techniques. Further comparisons involving deep features show a clearer trend and a better improvement against other methods including CNN [47], ResNet [45], DKN [27], DMN [28], LNR+TagProp-σSD and LNR+2PKNN [86] as well as CCA-KNN [47]. Although the latters model bi-modal "image and textual" contexts using Listwise Neural Ranking (LNR) and Canonical Correlation Analysis (CCA), the proposed approach, which relies only on visual features, is still competitive.

NUS-WIDE dataset: Following the best experimental settings in ImageCLEF and Corel5k, we further show the results on a large dataset. The images in NUSWIDE are resized to a reference dimension of 400×500 pixels, and partitioned using a regular grid of 2×2 cells. Each cell is encoded with the deep features extracted from the pre-trained VGG model and the linear kernel map initialization is used to train our models. Similarly to Corel5k, ensemble SVMs are used for training (using all the positive samples and random subsets of negative samples with equal sizes), and the average scores of these ensemble SVMs are taken in order to check the presence of different concepts on test images. Tab. 9 shows the performance of the proposed network as well as state-of-the-art methods. From this table, we observe that i) compared to context-free and handcrafted context-aware kernels, our proposed networks improve the F-scores and more noticeably the recall, ii) classwise contexts, either learned layerwise or stationary, achieve higher gain in recall with comparable precision and F-scores w.r.t. to the global contexts, and iii) compared to other state-of-the-art methods, mainly those based on CNNs and KNNs, our method outperforms CNN+Softmax [74] based on AlexNet and shows competitive performance against CNN+RNN in [50].

Method	$\mathbf{R}$	$\mathbf{P}$	$\mathbf{F}$	$\mathbf{N}_+$
CNN+RNN [50]	30.4	40.5	34.7	-
CNN+Softmax [74]	31.2	31.7	31.5	80
CNN+WARP [74]	35.6	31.7	33.5	78
TagPop σSD [12]	35.0	42.0	38.0	-
2PKNN [13]	39.0	42.0	40.0	-
CNN+Logistic [87]	45.0	45.6	45.3	-
LNR+2PKNN [86]	46.0	44.0	45.0	80
S-CNN-RNN [51]	50.2	55.7	52.8	-
ResNet-101 [45]	56.8	46.9	47.0	-
ResNet-SRN [67]	58.9	48.2	48.9	-
SINN [88]	60.6	58.3	59.4	-
CF	42.2	40.1	32.0	81
Handcrafted CA	45.0	38.5	32.7	81
L-Global CA (proposed)	45.8	38.1	32.9	81
S-Global CA (proposed)	45.3	38.2	32.3	81
L-Classwise CA (proposed)	46.8	37.9	32.3	81
S-Classwise CA (proposed)	46.8	38.0	32.3	81

Table 9: Comparison of the proposed deep context-aware kernel networks w.r.t. the other works on the NUS-WIDE benchmark. Our results are obtained using the linear kernel map on a grid of 2×2 cells with $r = 1$ for all the learned context variants.

Other methods including LNR+2PKNN (as well as S-CNN-RNN, ResNet-SRN and SINN[8]) [86] show superior performances as they rely not only on deep visual

[8]S-CNN-RNN and ResNet-SRN add semantic and label spatial regularization between CNNs and RNNs while SINN uses different side information of tags, groups and labels to model the semantic correlation between concepts and a bidirectional RNN-like model is adopted to integrate all these informations.

features but also on other sources of multimodal semantic information including text in order to define the context.

Following the state-of-the-art, extra modalities (mainly textual information, label relationships, etc.) could also be considered. We believe that adding extra textual information could bring an extra-gain to our context learning; this issue, out of the main scope of this paper, will be addressed as a future work. Finally, Fig. 6 shows image instances and their annotation results respectively using context-free, handcrafted vs. learned context networks from the test set of ImageCLEF, Corel5k and NUS-WIDE datasets.

6. Conclusion

In this paper we introduce a novel deep context-aware kernel network that considers context learning as as part of kernel design. The proposed method is based on a particular deep network architecture whose parameters — trained "end-to-end" — model the contextual relationships between visual patterns (cells) into images. Different variants of contexts are investigated including layerwise, stationary and classwise. While stationary contexts allow us to reduce the actual number of training parameters, classwise ones make it possible to further enhance the performances by making context class-dependent. Extensive experiments conducted on the challenging ImageCLEF, Corel5k and NUS-WIDE benchmarks, show a clear and a consistent gain of classifiers trained on top of the learned context networks w.r.t. classifiers trained using handcrafted context networks as well as context-free ones. As a future work, we are currently investigating the issues of (i) the integration of attention mechanisms into our context networks in order to model primitive saliency in images and (ii) the use of a priori knowledge (mainly

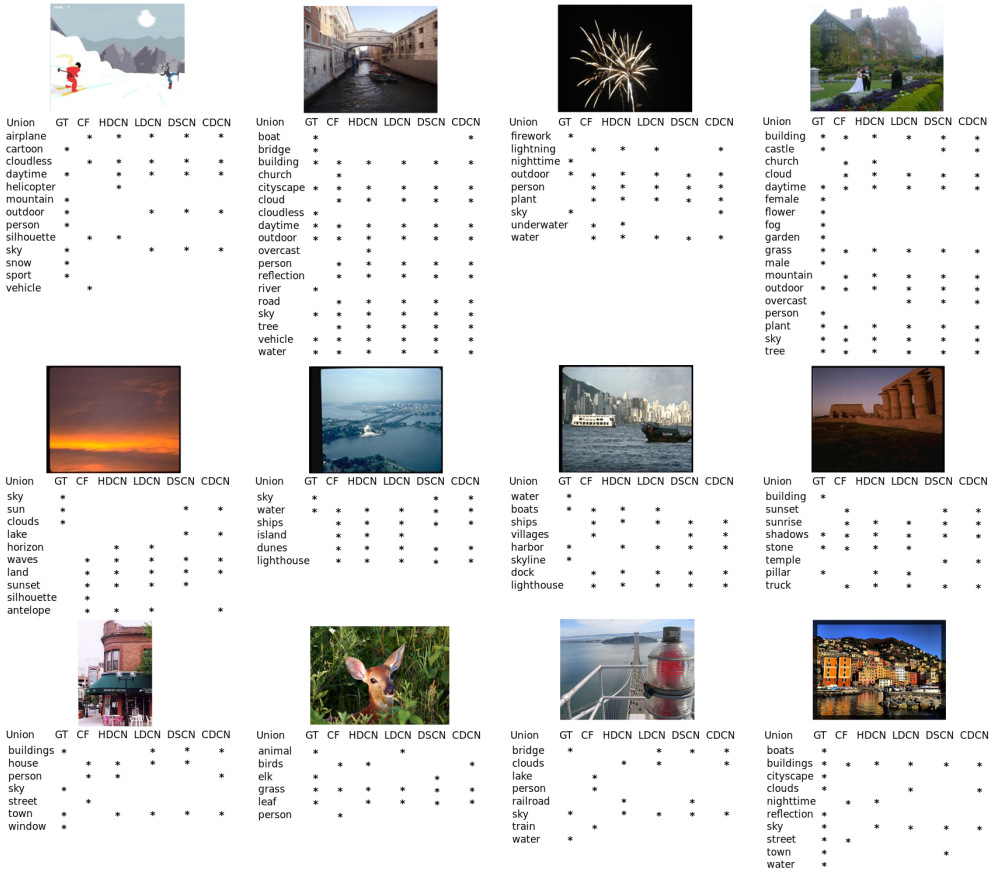

Figure 6: These figures show some annotation examples using (from left-to-right on each image): context-free kernels ("CF"), handcrafted and learned (layerwise, stationary and classwise) context-aware kernel networks respectively denoted as HDCN, LDCN, DSCN and CDCN. "GT" stands for ground-truth annotations and the stars refer to the presence of a given concept in a test image. Results shown in the first row correspond to ImageCLEF while those in the second and third row respectively to Corel5k and NUS-WIDE. All these results are obtained using polynomial kernel map initialization and VGG features on a grid of 2×2 cells with $r = 1$.

semantic structures) in context learning; we believe that these two extensions will further enhance the performances.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (grant numbers 61806180 and U1804152); the Key Research Projects of Henan Higher Education Institutions in China (grant number 19A520037); and also the research agency ANR (Agence Nationale de la Recherche) of France under the MLVIS project (grant number ANR-11-BS02-0017).

References

[1] D. Li, X. Chen, Z. Zhang, K. Huang, Learning deep context-aware features over body and latent parts for person re-identification, in: Proceedings of *CVPR*, 2017, pp. 7398–7407.

[2] P. Martins, P. Carvalho, C. Gatta, Context-aware features and robust image representations, Journal of Visual Communication and Image Representation 25 (2) (2014) 339–348.

[3] K. S. Arun, V. K. Govindan, A context-aware semantic modeling framework for efficient image retrieval, International Journal of Machine Learning and Cybernetics 8 (4) (2017) 1259–1285.

[4] J. Zhang, Y. Mu, S. Feng, K. Li, Y. Yuan, C.-H. Lee, Image region annotation based on segmentation and semantic correlation analysis, IET Image Processing 12 (8) (2018) 1331–1337.

[5] C. Jin, Q. M. Sun, S. W. Jin, A hybrid automatic image annotation approach, Multimedia Tools and Applications 78 (9) (2019) 11815–11834.

[6] J. Zhang, T. Tao, Y. Mu, H. Sun, D. Li, Z. Wang, Web image annotation based on tri-relational graph and semantic context analysis, Engineering Applications of Artificial Intelligence 81 (2019) 313–322.

[7] K. Barnard, P. Duygulu, N. de Freitas, D. Forsyth, Matching words and pictures, The Journal of Machine Learning Research 3 (2003) 1107–1135.

[8] A. Makadia, V. Pavlovic, S. Kumar, A new baseline for image annotation, in: Proceedings of *ECCV*, 2008, pp. 316–329.

[9] H. Sahbi, J.-Y. Audibert, J. Rabarisoa, R. Keriven, Context-dependent kernel design for object matching and recognition, in: Proceedings of *CVPR*, 2008.

[10] K. Goh, E. Chang, B. Li, Using one-class and two-class svms for multiclass image annotation, IEEE Transactions on Knowledge and Data Engineering 17 (2005) 1333–1346.

[11] X. Qi, Y. Han, Incorporating multiple svms for automatic image annotation, IEEE Transactions on Knowledge and Data Engineering 40 (2007) 728–741.

[12] M. Guillaumin, T. Mensink, J. Verbeek, C. Schmid, Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation, in: In Proceedings of *ICCV*, 2009, pp. 316–329.

[13] Y. Verma, C. Jawahar, Image annotation using metric learning in semantic neighbourhoods, in: Proceedings of *ECCV*, 2012, pp. 836–849.

[14] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, Communications of the ACM 60 (6) (2017) 84–90.

[15] L. Deng, D. Yu, Deep learning: methods and applications, Foundations and Trends in Signal Processing 7 (3).

[16] R. Girshick, J. Donahue, T. Darrell, M. J., Crich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of *CVPR*, 2014, pp. 580–587.

[17] I. Goodfellow, Y. Bengio, A. Courville, Deep learning, in: MIT Press, 2016.

[18] R. K. Srivastava, K. Greff, J. Schmidhuber, Training very deep networks, in: Proceedings of *NIPS*, 2015, pp. 2377–2385.

[19] M. Jiu, H. Sahbi, L. Qi, Deep context networks for image annotation, in: Proceedings of *ICPR*, 2018, pp. 2422–2427.

[20] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, Going deeper with convolutions, in: Proceedings of *CVPR*, 2015, pp. 1–9.

[21] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, Imagenet large scale visual recognition challenge, International Journal of Computer Vision 115 (3) (2015) 211–252.

[22] J. Shawe-Taylor, N. Cristianini, Kernel methods for pattern analysis, Cambriage University Press (2004).

[23] A. Barla, F. Odone, A. Verri, Histogram intersection kernel for image classification, in: Proceedings of *ICPR*, 2003, pp. III–513.

[24] S. Maji, A. Berg, J. Malik, Classification using intersection kernel support vector machines is efficient, in: Proceedings of *CVPR*, 2008, pp. 1–8.

[25] F. Bach, G. Lanckriet, M. Jordan, Multiple kernel learning, conic duality, and the smo algorithm, in: Proceedings of *ICML*, 2004, pp. 1–6.

[26] A. Vedaldi, A. Zisserman, Efficient additive kernels via explicit feature maps, IEEE Transactions on Pattern Analysis and Machine Intelligence 34 (2012) 480–492.

[27] M. Jiu, H. Sahbi, Nonlinear deep kernel learning for image annotation, IEEE Transactions on Image Processing 26(4) (2017) 1820–1832.

[28] M. Jiu, H. Sahbi, Deep representation design from deep kernel networks, Pattern Recognition 88 (2019) 447–457.

[29] S. Belongie, J. Malik, J. Puzicha, Shape matching and object recognition using shape contexts, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (2001) 509–522.

[30] X. He, R. Zimel, M. Carreira, Multiscale conditional random fields for image labeling, in: Proceedings of *CVPR*, 2004, pp. II–II.

[31] H. Sahbi, Imageclef annotation with explicit context-aware kernel maps, International Journal of Multimedia Information Retrieval (2015) 113–128.

[32] H. Sahbi, J.-Y. Audibert, R. Keriven, Context-dependent kernels for object classification, IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (2011) 699–708.

[33] M. Jiu, C. Wolf, G. Taylor, A. Baskurt, Human body part estimation from depth images via spatially-constrained deep learning, Pattern Recognition Letteres 50 (2014) 122–129.

[34] Y. Liu, D. Zhang, G. Lu, A survey of content-based image retrieval with high-level semantics, Pattern Recognition 40 (1) (2007) 262–282.

[35] D. Zhang, M. Islam, G. Lu, A review on automatic image annotation techniques, Pattern Recognition 45 (2012) 346–362.

[36] Q. Cheng, Q. Zhang, P. Fu, C. Tu, S. Li, A survey and analysis on automatic image annotation, Pattern Recognition 79 (2018) 242–259.

[37] P. K. Bhagat, P. Choudhary, Image annotation: Then and now, Image and Vision Computing 80 (2018) 1–23.

[38] D. Grangier, S. Bengio, A discriminative kernel-based approach to rank images from text queries, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (2008) 1371–1384.

[39] R. Wong, C. Leung, Automatic semantic annotation of real-world web images, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (2008) 1933–1944.

[40] K. Kuroda, M. Hagiwara, An image retrieval system by impression words and specific object names-iris, Neurocomputing 43 (2002) 3–14.

[41] C. Cusano, G. Ciocca, S. R., Image annotation using svm, in: Proceedings of the Internet Image IV, vol. 5304, SIPE, 2004.

[42] M. Jiu, H. Sahbi, Laplacian deep kernel learning for image annotation, in: Proceedings of *ICASSP*, 2016, pp. 1551–1555.

[43] Y. Liu, K. Wen, Q. Gao, X. Gao, F. Nie, SVM based multi-label learning with missing labels for image annotation, Pattern Recognition 78 (2018) 307–317.

[44] Y. LeCun, L. Botto, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of IEEE 86 (11) (1998) 2278–2324.

[45] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of *CVPR*, 2016, pp. 770–778.

[46] L. Zheng, Y. Yang, Q. Tian, SIFT Meets CNN: A Decade Survey of Instance Retrieval, IEEE Transactions on Pattern Analysis and Machine Intelligence 40 (5) (2018) 1224–1244.

[47] V. N. Murthy, S. Maji, R. Manmatha, Automatic image annotation using deep learning representations, in: International Conference on Multimedia Retrieval, 2015, pp. 603–606.

[48] Y. Niu, Z. Lu, J.-R. Wen, T. Xiang, S.-F. Chang, Multi-modal multi-scale deep learning for large-scale image annotation, IEEE Transactions on Image Processing 28 (4) (2019) 1720–1731.

[49] Y. Ma, Y. Liu, Q. Xie, L. Li, CNN-feature based automatic image annotation method, Multimedia Tools and Applications 78 (3) (2019) 3767–3780.

[50] J. Wang, Y. Yang, J. Mao, Z. Huang, C. Huang, W. Xu, Cnn-rnn: A unified framework for multi-label image classification, in: Proceedings of *CVPR*, 2016, pp. 2285–2294.

[51] F. Liu, T. Xiang, T. M. Hospedales, W. Yang, C. Sun, Semantic regularisation for recurrent image annotation, in: Proceedings of *CVPR*, 2017, pp. 2872–2880.

[52] R. You, Z. Guo, L. Cui, X. Long, Y. Bao, S. Wen, Cross-modality attention with semantic graph embedding for multi-label classification, arXiv:1912.0787 (2019).

[53] V. Yazici, A. Gonzalez-Garcia, A. Ramisa, B. Twardowski, J. van de Weijer, Orderless recurrent models for multi-label classification, in: Proceedings of *CVPR*, 2020, pp. 13437–13446.

[54] U. Tiberio, B. Lamberto, S. Lorenzo, D. Alberto, Automatic image annotation via label transfer in the semantic space, Pattern Recognition (2017) 144–157.

[55] Y. Zhang, M. Bai, P. Kohli, S. Izadi, J. Xiao, Deepcontext: Context-encoding neural pathways for 3d holistic scene understanding, in: Proceedings of *ICCV*, 2017, pp. 1201–1210.

[56] W.-C. Hung, Y.-H. Tsai, X. Shen, Z. Lin, K. Sunkavalli, X. Lu, M.-H. Yang, Scene parsing with global context embedding, in: Proceedings of *ICCV*, 2017, pp. 2650–2658.

[57] S. Belongie, J. Malik, J. Puzicha, Shape Matching and Object Recognition

Using Shape Contexts, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (24) (2002) 509–521.

[58] K. Grauman, T. Darrell, The pyramid match kernel: Efficient learning with sets of features, The Journal of Machine Learning Research 8 (2007) 725–760.

[59] P. Velicković, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio, Graph attention networks, in: Proceedings of *ICLR*, 2018, pp. 1–12.

[60] L. Mi, Z. Chen, Hierarchical graph attention network for visual relationship detection, in: Proceedings of *CVPR*, 2020, pp. 13883–13892.

[61] M. Defferrard, X. Bresson, P. Vandergheynst, Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering, in: Proceedings of *NIPS*, 2016, pp. 3844–3852.

[62] T. N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: Proceedings of *ICLR*, 2017, pp. 1–14.

[63] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, P. S. Yu, A Comprehensive Survey on Graph Neural Networks, IEEE Transactions on Neural Networks and Learning Systems 32 (1) (2021) 4–24.

[64] A. Mazari, H. Sahbi, MLGCN: Multi-laplacian graph convolutional networks for human action recognition, in: Proceedings of *BMVC*, 2019, pp. 1–16.

[65] B. Knyazev, X. Lin, M. R. Amer, G. W. Taylor, Image classification with

hierarchical multigraph networks, in: Proceedings of *BMVC*, 2019, pp. 1–13.

[66] T. Chen, M. Xu, X. Hui, H. Wu, L. Lin, Learning semantic-specific graph representation for multi-label image recognition, in: Proceedings of *CVPR*, 2019, pp. 522–531.

[67] F. Zhu, H. Li, W. Ouyang, N. Yu, X. Wang, Learning spatial regularization with image-level supervisions for multi-label image classification, in: Proceedings of *CVPR*, 2017, pp. 2027–2036.

[68] J. Mairal, P. Koniusz, Z. Harchaoui, C. Schmid, Convolutional kernel networks, in: Proceedings of *NIPS*, 2014.

[69] D. Lowe, Object recognition from local scale-invariant features, in: Proceedings of *ICCV*, Vol. 2, 1999, pp. 1150–1157.

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

[71] M. Villegas, R. Paredes, T. B., Overview of the imageclef 2013 scalable concept image annotation subtask, in: CLEF, 2013.

[72] P. Duygulu, K. Barnard, N. de Freitas, D. Forsyth, Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary, in: Proceedings of *ECCV*, 2002, pp. 97–112.

[73] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, Y.-T. Zheng, Nus-wide: A real-world web image database from national university of singapore, in: Proceedings of *CIVR*, 2009, pp. 1–9.

[74] G. Y., Y. Jia, T. Leung, A. Toshev, S. Ioffe, Deep convolutional ranking for multilabel image annotation, arXiv:1312.4894 (2014).

[75] K. Chatfield, K. Simonyan, A. Vedaldi, A. Zisserman, Return of the devil in the details: Delving deep into convolutional nets, in: Proceedings of *BMVC*, 2014.

[76] M. Varma, B. Babu, More generality in efficient multiple kernel learning, in: Proceedings of *ICML*, 2009, pp. 1065–1072.

[77] J. Zhuang, I. Tsang, S. Hoi, Two-layer multiple kernel learning, in: Proceedings of *ICML*, 2011, pp. 909–917.

[78] V. Lavrenko, R. Manmatha, J. Jeon, A model for learning the semantics of pictures, in: Proceedings of *NIPS*, 2003.

[79] D. Metzler, R. Manmatha, A inference network approach to image retrieval, in: Proceedings of *CIVR*, 2004, pp. 42–50.

[80] P. Vo, H. Sahbi, Transductive kernel map learning and its application to image annotation, in: Proceedings of *BMVC*, 2012.

[81] Y. Verma, C. Jawahar, Exploring svm for image annotation in presence of confusing labels, in: Proceedings of *BMVC*, 2013.

[82] S. Moran, V. Lavrenko, Sparse kernel learning for image annotation, in: Proceedings of *ICMR*, 2014, pp. 113–120.

[83] Q. Ji, L. Zhang, X. Shu, J. Tang, Image annotation refinement via 2P-KNN based group sparse reconstruction, Multimedia Tools and Applications 78 (10) (2019) 13213–13225.

[84] X. Jing, F. Wu, Z. Li, R. Hu, D. Zhang, Multi-Label Dictionary Learning for Image Annotation, IEEE Transactions on Image Processing 25 (6) (2016) 2712–2725.

[85] M. Zamiri, H. Sadoghi Yazdi, Image annotation based on multi-view robust spectral clustering, Journal of Visual Communication and Image Representation 74 (October 2020) (2021) 103003.

[86] W. Zhang, H. Hu, H. Hu, Neural ranking for automatic image annotation, Multimedia Tools and Applications 77 (17) (2018) 22385–22406.

[87] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, Y.-T. Zheng, Love thy neighbors: Image annotation by exploiting image metadata, in: Proceedings of *ICCV*, 2015, pp. 4624–4632.

[88] H. Hu, G. Zhou, Z. Deng, Z. Liao, G. Mori, Learning structured inference neural networks with label relations, in: Proceedings of *CVPR*, 2016, pp. 2960–2968.