HAL open science

DIABLO: Dictionary-based attention block for deep metric learning

Pierre Jacob, David Picard, Aymeric Histace, Edouard Klein

To cite this version:

Pierre Jacob, David Picard, Aymeric Histace, Edouard Klein. DIABLO: Dictionary-based attention block for deep metric learning. Pattern Recognition Letters, 2020, 10.1016/j.patrec.2020.03.020 . hal-02510473

HAL Id: hal-02510473

https://hal.science/hal-02510473v1

Submitted on 22 Aug 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

Version of Record: https://www.sciencedirect.com/science/article/pii/S0167865520300982
Manuscript_4c535412bf325e680dc31784cbf91ac1

Pattern Recognition Letters
journal homepage: www.elsevier.com

DIABLO: Dictionary-based attention block for deep metric learning

Pierre Jacob[a,**], David Picard[a,b], Aymeric Histace[a], Edouard Klein[c]

[a] *ETIS UMR 8051, Université Paris Seine, UCP, ENSEA, CNRS, F-95000, Cergy, France*
[b] *LIGM, UMR 8049, École des Ponts, UPE, Champs-sur-Marne, France*
[c] *Pôle Judiciaire de la Gendarmerie Nationale, 5 boulevard de l'Hautil, 95000 Cergy, France*

ABSTRACT

Recent breakthroughs in representation learning of unseen classes and examples have been made in deep metric learning by training at the same time the image representations and a corresponding metric with deep networks. Recent contributions mostly address the training part (loss functions, sampling strategies, *etc.*), while a few works focus on improving the discriminative power of the image representation. In this paper, we propose *DIABLO*, a dictionary-based attention method for image embedding. *DIABLO* produces richer representations by aggregating only visually-related features together while being easier to train than other attention-based methods in deep metric learning. This is experimentally confirmed on four deep metric learning datasets (Cub-200-2011, Cars-196, Stanford Online Products, and In-Shop Clothes Retrieval) for which *DIABLO* shows state-of-the-art performances.
© 2020 Elsevier Ltd. All rights reserved.

1. Introduction

Deep Metric Learning (DML) is an important yet challenging topic in the Computer Vision community, that has a broad-spectrum in terms of applications such as: person or vehicle identification (Zhou et al., 2017), visual product search (Liu et al., 2016; Oh Song et al., 2016) or multi-modal retrieval (Wehrmann and Barros, 2018; Carvalho et al., 2018). By learning the image representations and an embedding space together, DML methods produce compact representations where visually-related images (*e.g.*, images of the same car model) are close to each other and dissimilar images (*e.g.*, images of two cars from the same brand but from different models) are distant.

Recent contributions mainly address the training part of deep metric learning, proposing loss functions (*e.g.* Angular loss (Wang et al., 2017)), sampling strategies (*e.g.*, DAMLRMM (Xu et al., 2019)) and ensemble methods (*e.g.*, BIER (Opitz et al., 2017)). All of these methods are built upon a backbone network such as GoogleNet (Szegedy et al., 2015) to extract the local features from which the image representations and the corresponding metric are computed. Nowadays, global average pooling is the most used pooling strategy to compute the image representations. This is due to interesting properties such as full back-propagation of the gradient or localization ability without direct supervision (Zhou et al., 2016). In almost every method, these representations are computed using the mean of the deep features and used as they are.

However, the average is also known to be a non-robust representation because it is very sensitive to outliers and to sampling problems (Jacob et al., 2019b). A famous solution is to strengthen this representation in order to compute the average on only visually-related features, using a set of attention maps such as ABE (Kim et al., 2018). ABE is based on what we next call *dimension-wise selection with pre-attention* - this method shows very good results with few additional parameters, through both pooling of visually-related features and feature denoising. Also, ABE is trained with a divergence loss which ensures that the attention maps are complementary by enforcing two attention maps to be dissimilar, even for visually-related images. Due to this optimization criterion and the trade-off parameter, both the training procedure and the parameterization become more complex. NetVLAD (Arandjelovic et al., 2016) is based on what we next call *feature-wise selection with post-attention* - this method aggregates visually-similar features using a structural constraint based on a dictionary strategy. However, the *feature-wise selection* does not contribute to feature denoising.

In this paper, we introduce the method *DIABLO*, a DIctionary-based Attention BLOck, that produces robust im-

**Corresponding author:
e-mail: pierre.jacob@ensea.fr (Pierre Jacob)

© 2020 published by Elsevier. This manuscript is made available under the CC BY NC user license
https://creativecommons.org/licenses/by-nc/4.0/

age representations by taking advantage of both ABE and NetVLAD benefits. We evaluate attention strategies named *pre-attention* and *post-attention* in the DML setup, together with two selection strategies named *feature-wise* and *dimension-wise* selection. We show in practice that *DIABLO* consistently improves the state-of-the-art on four DML datasets (Cub-200-2011, Cars-196, Stanford Online Products and In-Shop Clothes Retrieval).

The remaining of the paper is organized as follows: in the next section, we present the related work on deep metric learning and support our proposition. In section 3, we give an overview of the proposed architecture, then we detail the attention strategies and the feature-wise and dimension-wise selection methods. In section 4, we show ablation studies on the selection and attention strategies, the dictionary size and the comparison to ABE. Finally, in section 5 we compare our approach to the state-of-the-art methods on four image retrieval datasets (Cub-200-2011, Cars-196, Stanford Online Products and In-Shop Clothes Retrieval).

2. Related work

In DML, we learn the image representations and an embedding together in such a way that the Euclidean distance corresponds with the semantic content of the images. The standard strategy is to extract deep features using a backbone network such as GoogleNet (Szegedy et al., 2015) and to learn the target representation with a linear projection on the average of these deep features. The whole network is fine-tuned to solve the metric learning task according to three criteria: a similarity-based loss function, a sampling strategy, and an ensemble method.

Standard loss functions rely on pairs or triplets of similar/dissimilar samples. Most recent ones extend these formulations by considering larger tuples (Ustinova and Lempitsky, 2016) or by improving the design (Wang et al., 2017). The batch construction can either be done by random sampling, mining strategies (Xu et al., 2019), proxy-based approximations (Movshovitz-Attias et al., 2017) or generation (Lin et al., 2018). Finally, ensemble methods have recently become a popular way of improving the performances of DML architectures (Opitz et al., 2017, 2018).

The latest DML approaches consider using a codebook strategy (Arandjelovic et al., 2016) or even attention maps (Kim et al., 2018). Kim et al. (2018) propose ABE, an attention-based ensemble method to enforce diversity within the embedding space. With this aim, they train a set of attention blocks to select several feature map entries per block. These blocks are then trained with a divergence loss function, which ensures that all attention maps are complementary. Their proposed design relies on what we call a *dimension-wise selection with pre-attention*. they select a set of dimensions from the feature maps using a given attention map. Then, these selected dimensions are further processed with Inception blocks. We call this selection schema a *pre-attention* strategy - and we argue that one of its properties results in denoising the deep features. Thus, this improves the image representations by post-processing features that contain only relevant information. However, there are two major drawbacks in using a divergence loss. On the one hand, we need to cross-validate the additional trade-off hyper-parameter. On the other hand, the training time is increased because the divergence loss plays the opposite role of the similarity-based loss function.

Arandjelovic et al. (2016) propose NetVLAD which takes advantage of a dictionary strategy that avoids the divergence loss. In their case, the optimization constraint is replaced by a structural one, that simplifies the training procedure by directly optimizing the task-dependent loss function. However, NetVLAD's design relies on a strategy that we call *feature-wise selection with post-attention*. Feature-wise selection ensures that only visually-related features are pooled together. This selection strategy does not have the denoising property of the *dimension-wise selection*. Also, in the case of NetVLAD, the pre-processing is a centering and an intra-projection, that can be improved using non-linear transformation learned by a CNN.

In *DIABLO*, we show the benefits of dimension-wise selection versus feature-wise selection with pre-attention or post-attention strategies. With this aim, we propose to replace the divergence loss in ABE by a structural constraint, which is based on a dictionary strategy. This dictionary is trained without direct supervision as it is the case with NetVLAD. We show that it leads to better results than ABE on four DML datasets.

3. Proposed method

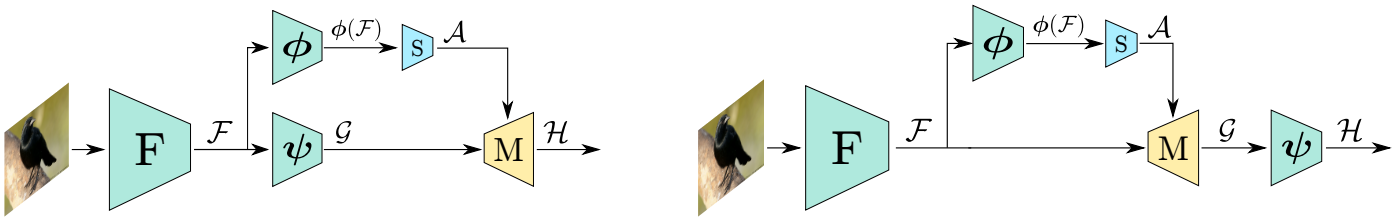
In this section, we start by giving an overview of *DIABLO*. Then, we describe the attention maps computation and the two selection strategies named feature-wise and dimension-wise selection.

3.1. Method overview

We give an overview of the method illustrated in Figure 1. We start by extracting a deep feature map $\boldsymbol{\mathcal{F}} \in \mathbb{R}^{h \times w \times c}$ using the local feature extractor F where h and w are the height and width of the feature map and c is the deep feature dimension. We further process the extracted feature map using a non-linear function $\boldsymbol{\phi}$ implemented by a convolutional neural network (CNN). In order to compute the N attention maps $\boldsymbol{\mathcal{A}}$ where N is the dictionary size, we pass the feature map into the selection block S, that is either the feature-wise selection (subsection 3.3) or the dimension-wise selection (subsection 3.4).

In the post-attention setup (Figure 1a), the feature map $\mathcal{F}$ is further processed with a non-linear function $\boldsymbol{\psi}$ (implemented by a CNN) and transformed into a feature map $\mathcal{G}$. It is then combined with the attention maps $\boldsymbol{\mathcal{A}}$ using the block M to produce the N new feature maps. In the pre-attention setup (Figure 1b), the feature map $\mathcal{F}$ is directly combined with the attention maps $\boldsymbol{\mathcal{A}}$ using the block M. In this case, we further process these N feature maps using a non-linear function $\boldsymbol{\psi}$. The different blocks M used to combine the attention maps and the original feature map are illustrated in Figure 2.

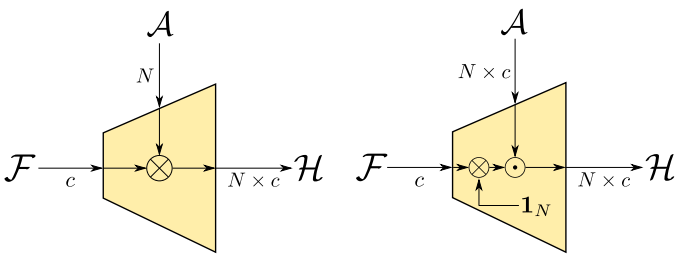
These N feature maps are pooled using a global average pooling with adding an embedding layer for each branch. So, the output representation is obtained by concatenating these N branches.



(a) Post-attention strategy

(b) Pre-attention strategy

Fig. 1: Illustration of post and pre-attention strategies. In pre-attention, we first compute the attention maps and then we process each of them. In post-attention, the feature map is further processed before computing the attention maps. In pre-attention, we first compute the attention maps and then we process each of them. F is a feature extractor and $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$ are two non-linear transformation of the deep features. The blocks $\mathcal{S}$ correspond to either Equation 7 or Equation 11. The blocks $\mathcal{M}$ are illustrated in Figure 2.



(a) Feature-wise selection

(b) Dimension-wise selection

Fig. 2: Illustration of the merging block M for feature-wise and dimension-wise selection strategies. The feature-wise selection is directly implemented using Kronecker product ($\otimes$). The dimension-wise selection first duplicate N times the feature map before computing the Hadamard product ($\odot$) with the attention maps. Reported dimensions are without the spatial dimensions $h \times w$.

3.2. Attention strategies

This section focuses on computing a set of attention maps $\mathcal{A}$ using one of the selection blocks S and its corresponding dictionary $\mathcal{D}$ as well as the feature map $\mathcal{F} \in \mathbb{R}^{h \times w \times c}$. For this purpose, we process the feature map $\mathcal{F}$ with a non-linear function $\boldsymbol{\phi} : \mathbb{R}^c \rightarrow \mathbb{R}^m$ implemented by a CNN. Then, the set of attention maps are computed by a selection block S such that $\mathcal{A} = \mathrm{S}(\boldsymbol{\phi}(\mathcal{F}); \mathcal{D})$. $\mathcal{A}$ is used with one of the following attention strategies.

3.2.1. Post-attention

In the post-attention strategy illustrated in Figure 1a, we process the feature map $\mathcal{F}$ into an intermediate feature map $\mathcal{G}$:

$$\mathcal{G} = \boldsymbol{\psi}(\mathcal{F}) \tag{1}$$

Then, we combine this feature map $\mathcal{G}$ with the attention maps $\mathcal{A}$ using the merging block M:

$$\mathcal{H} = M(\mathcal{G}; \mathcal{A}) \tag{2}$$

with $\mathcal{H}$ representing the set of feature maps obtained by the post-attention strategy. From these, we perform a spatial pooling of the local features and we add an embedding layer to generate the corresponding image representation.

The main idea of post-attention consists of aggregating only the related features, unlike global pooling that aggregates unrelated features. For example, features that describe the background are aggregated using a different attention block than those which represent the desired object. One can note that this approach is strongly related to NetVLAD for which the function $\boldsymbol{\psi}$ is a centering. However, it differs from NetVLAD in two ways: First, because it learns a non-linear clustering of the local features using the function $\boldsymbol{\phi}$; second because it learns a non-linear pre-processing of the deep features using the function $\boldsymbol{\psi}$.

3.2.2. Pre-attention

In the pre-attention strategy illustrated in Figure 1b, we perform the operation in reverse order. In order to do so, we combine the feature map $\mathcal{F}$ with the set of attention maps $\mathcal{A}$ using the merging block M to produce the set of N (one per attention block) feature maps $\mathcal{G}$:

$$\mathcal{G} = M(\mathcal{F}; \mathcal{A}) \tag{3}$$

Then, we process these N feature maps using a non-linear function $\boldsymbol{\psi}$ to generate the feature maps $\mathcal{H}$:

$$\mathcal{H} = \boldsymbol{\psi}(\mathcal{G}) \tag{4}$$

From these N feature maps, we pool the local features and we add an embedding layer to generate the corresponding image representation.

The pre-attention strategy is similar to a refinement approach. Indeed, the attention selects dimensions or features, and a refinement function $\boldsymbol{\psi}$ improves these extracted features before that they are aggregated. Hence, the attention maps role is to select only the relevant information for a given attention block. The function $\boldsymbol{\psi}$ is trained to refine this information so that it improves generalization.

3.3. Feature-wise selection

In this section, we give details about the computation of the selection block S and the merging block M for the feature-wise selection illustrated in Figure 2a. We start from a feature map $\mathcal{F} \in \mathbb{R}^{h \times w \times c}$ from which we compute a set of attention maps $\mathcal{A} = \{\mathcal{A}^{(n)} \in \mathbb{R}^{h \times w \times c}\}_n$ using a dictionary $\mathcal{D}$ such that $\mathcal{A} = \mathrm{S}(\mathcal{F}; \mathcal{D})$. Before computing the feature-wise selection, the feature map $\mathcal{F}$ is processed by a non-linear function $\boldsymbol{\phi} : \mathbb{R}^c \rightarrow \mathbb{R}^m$ implemented by a CNN. We denote $\boldsymbol{f}_{i,j} \in \mathbb{R}^c$ a feature from $\mathcal{F}$ at spatial location (i, j) and $\boldsymbol{\phi}(\boldsymbol{f}_{i,j}) \in \mathbb{R}^m$ its transformation in $\boldsymbol{\phi}(\mathcal{F})$ that is in the same spatial location.

3.3.1. Selection block

In the feature-wise selection, the objective is to assign each feature $\boldsymbol{f}_{i,j}$ from the feature map $\mathcal{F}$ to one of the dictionary entries $\mathcal{D} = \{\boldsymbol{d}^{(n)} \in \mathbb{R}^m\}_n$. To do so, we consider the cosine similarity between the transformed feature $\boldsymbol{\phi}(\boldsymbol{f}_{i,j})$ and the dictionary entry:

$$s(\boldsymbol{\phi}(\boldsymbol{f}_{i,j}), \boldsymbol{d}^{(n)}) = \frac{\left\langle \boldsymbol{\phi}(\boldsymbol{f}_{i,j}) \; ; \; \boldsymbol{d}^{(n)} \right\rangle}{\|\boldsymbol{\phi}(\boldsymbol{f}_{i,j})\|_2 \|\boldsymbol{d}^{(n)}\|_2} \tag{5}$$

Using the cosine similarity has a main advantage when compared to the Euclidean distance: during the training, it is more stable by means of the ℓ_2-normalization.

From this similarity measure, we compute $\mathcal{A}^{(n)}_{i,j,k}$, the weight for the n-th feature-wise attention map and k-th dimension of the feature in spatial location (i, j):

$$\mathcal{A}^{(n)}_{i,j,k} = \begin{cases} 1 & \text{if } \boldsymbol{d}^{(n)} = \underset{\boldsymbol{d}^{(l)} \in \mathcal{D}}{\arg\max}\; s(\boldsymbol{\phi}(\boldsymbol{f}_{i,j}), \boldsymbol{d}^{(l)}) \\ 0 & \text{else.} \end{cases} \tag{6}$$

However, this one-hot encoder is not differentiable due to the arg max operator. We relax the constraint using the soft-max operator to train in an end-to-end way the deep network:

$$\mathcal{A}^{(n)}_{i,j,k} = \frac{\mathrm{e}^{\alpha \; s(\boldsymbol{\phi}(\boldsymbol{f}_{i,j}), \boldsymbol{d}^{(n)})}}{\sum_l \mathrm{e}^{\alpha \; s(\boldsymbol{\phi}(\boldsymbol{f}_{i,j}), \boldsymbol{d}^{(l)})}} \tag{7}$$

such that, α is a hyper-parameter to control the hardness of the assignment. For this formulation, we rely on a given feature $\boldsymbol{f}_{i,j}$ that is assigned to a dictionary entry $\boldsymbol{d}^{(n)}$ according to the similarity between $\boldsymbol{\phi}(\boldsymbol{f}_{i,j})$ and $\boldsymbol{d}^{(n)}$. We show in section 5 that the feature-wise selection increases the performance of the attention-based models when compared to the baseline model (without attention map). Then, we detail the block M to merge the feature-wise selection based attention map with the raw features.

3.3.2. Merging block

The combination of the attention maps $\mathcal{A}$ and the feature map $\mathcal{F}$ is illustrated in Figure 2a. For the k-th dimension of the n-th feature map in spatial location (i, j), the corresponding entry in $\mathcal{H}$, $\mathcal{H}^{(n)}_{i,j,k}$ is computed using the following equation:

$$\mathcal{H}^{(n)}_{i,j,k} = \mathcal{A}^{(n)}_{i,j,k} \, \mathcal{F}_{i,j,k} \tag{8}$$

It must be considered that $\mathcal{A}^{(n)}_{i,j,k}$ has the same value independently from the value of k. Thus, Equation 8 is easily implemented using the Kronecker product ($\otimes$) as illustrated in Figure 2a:

$$\mathcal{H}_{i,j} = \mathcal{A}_{i,j} \otimes \mathcal{F}_{i,j} \tag{9}$$

where $\mathcal{A}_{i,j}$ is composed by the N-th assignment weights and $\boldsymbol{f}_{i,j} \in \mathbb{R}^c$ is the feature for all spatial locations (i, j).

3.4. Dimension-wise selection

In this section, we extend the feature-wise selection from the subsection 3.3 to the dimension-wise selection. Similarly, we give details about the attention maps computation through the selection block S before that we explain the merging strategy with the block M.

3.4.1. Selection block

The selection block is composed of a set of directions per dictionary entry $\mathcal{D} = \{\boldsymbol{d}^{(n)}_k \in \mathbb{R}^m\}_{n,k}$ of size $N \times c$, on the contrary to the feature-wise selection that has a dictionary $\mathcal{D} = \{\boldsymbol{d}^{(n)} \in \mathbb{R}^m\}_n$ of size N. Thus, for a given feature $\boldsymbol{f}_{i,j} \in \mathcal{F}$ in spatial location (i, j), the cosine similarity is computed between the transformed feature $\boldsymbol{\phi}(\boldsymbol{f}_{i,j})$ and the k-th direction of the n-th dictionary entry $\boldsymbol{d}^{(n)}_k \in \mathcal{D}$:

$$s(\boldsymbol{\phi}(\boldsymbol{f}_{i,j}), \boldsymbol{d}^{(n)}_k) = \frac{\left\langle \boldsymbol{\phi}(\boldsymbol{f}_{i,j}) \; ; \; \boldsymbol{d}^{(n)}_k \right\rangle}{\|\boldsymbol{\phi}(\boldsymbol{f}_{i,j})\|_2 \|\boldsymbol{d}^{(n)}_k\|_2} \tag{10}$$

Then, one entry $\mathcal{A}^{(n)}_{i,j,k}$ from the attention map $\mathcal{A}$ is computed using the following equation:

$$\mathcal{A}^{(n)}_{i,j,k} = \frac{\mathrm{e}^{\alpha \; s(\boldsymbol{\phi}(\boldsymbol{f}_{i,j}), \boldsymbol{d}^{(n)}_k)}}{\sum_l \mathrm{e}^{\alpha \; s(\boldsymbol{\phi}(\boldsymbol{f}_{i,j}), \boldsymbol{d}^{(l)}_k)}} \tag{11}$$

The attention map has an attention weight for each dimension of the input feature and for each of the N attention blocks. section 5 highlights that the dimension-wise selection produces stronger image representations than the feature-wise selection, showing much higher performances.

3.4.2. Merging block

The merging block M is used in the case of dimension-wise selection as illustrated in Figure 2b. The entry $\mathcal{H}^{(n)}_{i,j,k}$ in $\mathcal{H}$ is computed for the k-th dimension of the n-th feature map in the spatial location (i, j) using the following equation:

$$\mathcal{H}^{(n)}_{i,j,k} = \mathcal{A}^{(n)}_{i,j,k} \, \mathcal{F}_{i,j,k} \tag{12}$$

Note that $\mathcal{A}^{(n)}_{i,j,k}$ depends on the value of k. The computation can easily be re-written using the Kronecker product and the Hadamard product ($\odot$) as illustrated in Figure 2b. Using the Kronecker product, we duplicate N times the local feature $\boldsymbol{f}_{i,j}$ Then, $\mathcal{H}$ is computed using the element-wise product between $\mathcal{A}$ and the duplicated feature map:

$$\mathcal{H} = (\mathcal{F} \otimes \mathbf{1}_N) \odot \mathcal{A} \tag{13}$$

3.5. Implementation details

For a fair comparison with other methods, we use a pre-trained GoogleNet (Szegedy et al., 2015) on ImageNet. The embedding size is fixed to 512 for all models. As it is done in common practice, we set the triplet margin $\alpha = 0.1$, the contrastive and the binomial margin $\beta = 0.5$ and the negative sample weight $C = 25$ in the binomial deviance. We follow the same training procedure as state-of-the-art methods (Opitz et al., 2017; Kim et al., 2018): training hyper-parameters (learning rate, batch size, *etc.*) are empirically chosen based on the training loss after a few epochs. Model hyper-parameters (number of layers, *etc.*) are set to comparable values as the ones of models reported in ABE (Kim et al., 2018). We use the following data augmentation on the images: multi-resolution where the size is uniformly sampled in [0.8, 1.8] times the crop size as well as random 256×256 crop and horizontal flip during the

training. During testing, we re-scale the images to 256×256. We use Adam (Kingma and Ba, 2015) with the default parameters and a learning rate of 10^{-5}. For all models, the function $\boldsymbol{\psi}$ is shared across the attention maps to reduce the large number of parameters induced. In practice, it still leads to strong experimental results and avoids over-fitting on small datasets, for instance: Cub-200-2011 and Cars-196.

4. Ablation studies

In this section, we compare our approach to the baseline in terms of model complexity and computation. Then, we present the evaluation protocol of our ablation studies. Ablation studies are performed to evaluate the benefits of pre and post-attention, the assignment approaches (feature and dimension-wise), the dictionary strategy and the dictionary size.

4.1. Model complexity and computation cost

In this section, we give the architecture choices to compute the functions $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$ for the post-attention and for the pre-attention architectures. Then, we analyze the induced computations and the complexity introduced by the dictionary approach.

In post-attention, the GoogleNet backbone extracts the feature map including and up to the max-pooling between the fourth and the fifth scales. To compute $\boldsymbol{\phi}$, we add the two inception blocks named '5a' and '5b' upon these features. The function $\boldsymbol{\psi}$ is also composed by two independent inception blocks '5a' and '5b'. The attention maps and the weighted features are computed using one of the two proposed strategies. Then, each branch is pooled using a global average pooling as well as an embedding layer of size $512/N$ and a ℓ_2-norm are added. The output representation is the concatenation of these N branches which leads to a 512d representation.

In pre-attention, we use the GoogleNet features including and up to the max-pooling between the third and the fourth scales. The non-linear function $\boldsymbol{\phi}$ is composed of the five inception blocks from the fourth scale of GoogleNet pre-trained on ImageNet. The refinement function $\boldsymbol{\psi}$ is composed by the fourth and fifth scales of GoogleNet, they are shared for each map but they are independent from $\boldsymbol{\phi}$. The attention maps and the weighted feature maps are computed using either dimension-wise or feature-wise selection. As it the case with pre-attention, each branch is pooled using a global average pooling, an embedding layer of size $512/N$ and a ℓ_2-norm are added before the concatenation of all branches in order to produce the full 512d representation.

These choices directly follow ABE (Kim et al., 2018) and we refer the reader to the related paper for more ablations on the architecture, including the multi-head approach, the attention module and the sharing of parameters across the attention modules.

All additional parameters are included in the computation of the function $\boldsymbol{\phi}$. By using the five Inception blocks from the fourth scale of GoogleNet, this leads to 3.5M additional parameters. Note that these parameters are shared across the dictionary entries and this drastically reduces the number of parameters. Also, note that the function $\boldsymbol{\psi}$ is already included in the number of parameters of GoogleNet.

In terms of computation, the most important additional computations come from the function $\boldsymbol{\psi}$ which is computed on each attention map. The computation of the fourth and fifth Inception scales are estimated to 0.7 Gflop (see Kim et al. (2018), Table 1). In comparison, the whole GoogleNet requires 1.6 Gflop to produce the image embedding. Note that in the case of the pre-attention, all parameters of $\boldsymbol{\psi}$ are shared across the attention maps, which leads to fewer additional parameters. Overall, these choices lead to higher experimental results for both ABE and *DIABLO* compared to the baseline.

4.2. Model selection protocol

In this section, we detail the evaluation protocol for all ablation studies on the Cub-200-2011 dataset that are performed to select the best model. We perform 10 random train-val splits on the training set of Cub-200-2011 for deep metric learning: We randomly choose 50 classes for the training set and we keep the rest of the classes for the validation set. Then, we train each model on the training set and we select the model that gives the best performances on the validation set. We then compute Recall@K on the testing set of Cub-200-2011 for each train-val split. All reported results in Table 4 and Table 3 are the average and the standard-deviation of Recall@K on the testing set for the ten runs.

4.3. Feature selection and attention strategy

First, we evaluate the benefit of the pre and post-attention strategies with respect to the assignment strategy. To that end, we fix the dictionary size to 8 and we use the training procedure from subsection 3.5. Results are reported in Table 3 for the dataset Cub-200-2011 with binomial loss. We remark that all strategies with the exception of the feature-wise selection with pre-attention improve over the baseline and this confirms the benefit of attention maps. This experiment also shows that feature-wise attention and dimension-wise attention impact differently the model.

In Feature-wise attention, the post-attention leads to stronger representations with +1.7% on Recall@1 over the pre-attention. We argue that selecting features make better sense with post-attention than pre-attention: only related features are aggregated together with post-attention whereas in pre-attention, the refinement part mostly processes sparse feature maps. In the case of dimension-wise attention, both approach provide stronger results with +3.8% and +4.3% over the best feature-wise strategy, even though it is still better to use the pre-attention approach. We argue that pre-attention is better with dimension-wise selection because the refinement part processes denoised features. Indeed, certain dimensions may be useless for a given dictionary entry and the dimension-wise approach can select only the relevant dimensions. Then, the refinement part is trained with sparse vectors which contain only the relevant information. Moreover, feature-wise selection with post-attention leads to aggregate sparse vector together, which might bring to sub-optimal results because some dimensions are rarely used.

Table 1: Comparison with the state-of-the-art on Cub-200-2011 and Cars-196 datasets using GoogleNet as feature extractor. Results are in percents. State-of-the-art results are in bold and results which improve the baseline are underlined.

	Cub-200-2011						Cars-196					
R@	1	2	4	8	16	32	1	2	4	8	16	32
RML (Roy et al., 2019)	52.3	64.5	75.3	84.0	-	-	73.2	82.2	88.6	92.2	-	-
Angular loss (Wang et al., 2017)	54.7	66.3	76.0	83.9	-	-	71.4	81.4	87.5	92.1	-	-
DAML (Duan et al., 2018)	52.7	65.4	75.5	84.3	-	-	75.1	83.8	89.7	93.5	-	-
HDML (Zheng et al., 2019)	53.7	65.7	76.7	85.7	-	-	79.1	87.1	92.1	95.5	-	-
DAMLRMM (Xu et al., 2019)	55.1	66.5	76.8	85.3	-	-	73.5	82.6	89.1	93.5	-	-
HDC (Yuan et al., 2017)	53.6	65.7	77.0	85.6	91.5	95.5	73.7	83.2	89.5	93.8	96.7	98.4
BIER (Opitz et al., 2017)	55.3	67.2	76.9	85.1	91.7	95.5	78.0	85.8	91.1	95.1	97.3	98.7
DVML (Lin et al., 2018)	52.7	65.1	75.5	84.3	-	-	82.0	88.4	93.3	96.3	-	-
HTG (Zhao et al., 2018)	59.5	71.8	81.3	88.2	-	-	76.5	84.7	90.4	94.0	-	-
HTL (Ge, 2018)	57.1	68.8	78.7	86.5	92.5	95.5	81.4	88.0	92.7	95.7	97.4	99.0
A-BIER (Opitz et al., 2018)	57.5	68.7	78.3	86.2	91.9	95.5	82.0	89.0	93.2	96.1	97.8	98.7
JCF (Jacob et al., 2019a)	60.1	72.1	81.7	88.3	-	-	82.6	89.2	93.5	96.0	-	-
HORDE (Jacob et al., 2019b)	59.4	71.0	81.0	88.0	93.1	96.5	83.2	89.6	93.6	96.3	98.0	98.8
ABE (Kim et al., 2018)	60.6	71.5	79.8	87.4	-	-	85.2	90.5	94.0	96.1	-	-
Contrastive (Ours)	58.7	69.7	79.4	87.0	92.6	96.1	78.5	85.9	90.9	94.4	96.7	98.1
Contrastive + *DIABLO*	62.3	73.6	82.6	89.2	94.0	96.9	84.8	90.5	94.3	96.6	98.1	98.9
Triplet (Ours)	55.9	67.0	77.7	86.1	92.0	95.5	73.1	81.7	87.9	92.9	95.9	97.6
Triplet + *DIABLO*	59.6	70.6	80.3	87.7	92.7	96.2	75.0	83.4	89.4	93.5	96.4	98.1
Binomial (Ours)	59.6	70.8	81.0	88.1	93.1	96.2	78.8	86.2	91.5	94.8	97.0	98.4
Binomial + *DIABLO* $N = 8$	62.8	73.9	82.4	89.3	94.0	96.7	85.0	90.8	94.0	96.4	98.0	98.9
Binomial + *DIABLO* $N = 16$	**63.9**	**74.3**	**82.4**	**88.8**	**94.0**	**96.8**	**85.4**	**91.3**	**95.0**	**97.2**	**98.5**	**99.1**

Table 2: Comparison with the state-of-the-art on Stanford Online Products and In-Shop Clothes Retrieval using GoogleNet as feature extractor. Results in percents. State-of-the-art results are in bold and results which improve the baseline are underlined.

	Stanford Online Products				In-Shop Clothes Retrieval					
R@	1	10	100	1000	1	10	20	30	40	50
Angular loss Wang et al. (2017)	70.9	85.0	93.5	98.0	-	-	-	-	-	-
DAML (Duan et al., 2018)	68.4	83.5	92.3	-	-	-	-	-	-	-
HDML (Zheng et al., 2019)	68.7	83.2	92.4	-	-	-	-	-	-	-
RML (Roy et al., 2019)	69.2	83.1	92.7	-	-	-	-	-	-	-
DAMLRMM (Xu et al., 2019)	69.7	85.2	93.2	-	-	-	-	-	-	-
HDC (Yuan et al., 2017)	69.5	84.4	92.8	97.7	62.1	84.9	89.0	91.2	92.3	93.1
BIER (Opitz et al., 2017)	72.7	86.5	94.0	98.0	76.9	92.8	95.2	96.2	96.7	97.1
DVML (Lin et al., 2018)	70.2	85.2	93.8	-	-	-	-	-	-	-
HTG (Zhao et al., 2018)	-	-	-	-	80.3	93.9	95.8	96.6	97.1	-
HORDE (Jacob et al., 2019b)	72.6	85.9	93.7	97.9	84.4	95.4	96.8	97.4	97.8	98.1
HTL (Ge, 2018)	74.8	88.3	94.8	98.4	80.9	94.3	95.8	97.2	97.4	97.8
A-BIER (Opitz et al., 2018)	74.2	86.9	94.0	97.8	83.1	95.1	96.9	97.5	97.8	98.0
ABE (Kim et al., 2018)	76.3	88.4	94.8	98.2	87.3	96.7	97.9	98.2	98.5	98.7
JCF (Jacob et al., 2019a)	77.4	89.9	95.8	98.6	-	-	-	-	-	-
Contrastive (Ours)	75.0	87.9	94.5	98.1	89.0	97.2	98.0	98.4	98.6	98.7
Contrastive + *DIABLO* $N = 8$	**77.8**	**89.5**	**95.3**	**98.4**	**91.3**	**98.1**	**98.7**	**99.0**	**99.1**	**99.1**
Triplet (Ours)	70.6	85.7	94.0	98.2	85.6	96.5	97.7	98.2	98.4	98.6
Triplet + *DIABLO* $N = 8$	73.5	87.8	95.0	98.5	87.4	97.2	98.1	98.6	98.8	98.9

Table 3: Impact of the pre-attention or post-attention on the performances for the three proposed attention strategies. Reported Recall@1 (R@1) is on the Cub-200-2011 dataset in percent.

		Feature		Dimension	
	Baseline	Pre-att	Post-att	Pre-att	Post-att
R@1	52.9 ± 0.2	52.3 ± 0.4	**54.0** ± 0.4	**58.3** ± 0.2	57.8 ± 0.1

Table 4: Impact of the dictionary size for the feature-wise post-attention mapping and for the dimension-wise pre-attention mapping. Reported results are Recall@1 (R@1) on the Cub-200-2011 dataset in percent.

Feat. + Post-att.				
N	2	4	8	16
R@1	53.4 ± 0.3	53.3 ± 0.3	**54.0** ± 0.4	48.0 ± 0.4
Dim. + Pre-att.				
N	2	4	8	16
R@1	57.1 ± 0.3	57.4 ± 0.3	58.3 ± 0.2	**58.9** ± 0.3

4.4. Comparison to ABE

In a second time, we want to evaluate the impact of the structural constraints imposed by the dictionary (Equation 7 and Equation 11) by comparing our method to ABE (Kim et al., 2018). In ABE, the authors show that a M-head approach already provides strong results on the Cars-196 dataset with 76.1% (+8.9%) Recall@1 for $M = 8$. However, this architecture tends to overfit due to the large number of parameters. Then, they propose an enhanced version named ABE that takes advantage of attention maps. The divergence loss increases the

performance from 69.7% without the divergence loss to 85.2% (+15.5%) in Recall@1 (see Table 2 in (Kim et al., 2018)).

Among the drawback of the divergence loss, despite of the additional hyper-parameter, figure that optimizing the loss generates gradients which are in opposition with the metric learning loss ones. Indeed, it is designed to reduce the similarity between different branches even when the images are similar. We solve this issue in *DIABLO* where this optimization constraint is replaced by a structural constraint (softmax in Equation 7 and Equation 11). The orthogonality is ensured by the design of *DIABLO*, which allows to simply remove the divergence loss at the price of a reduced expressiveness: feature map entries can be chosen independently in ABE whereas in *DIABLO* they are constrained to only one dictionary entry. In Table 1 and Table 2, *DIABLO* performs well compared to ABE with similar results on the Cars-196 dataset (-0.2% compared to ABE) and higher ones on other datasets such as Cub-200-2011 (+2.2%), Stanford Online Products (+1.5%) and Inshop Clothes Retrieval (+3.4%) for this set of parameters.

4.5. Dictionary size

In this section, we evaluate the impact of the dictionary size on *DIABLO*. We evaluate both the feature-wise and the dimension-wise with dictionary sizes in $\{2, 4, 8, 16\}$. Recall@1 on the Cub-200-2011 dataset are reported in Table 4.

In post-attention with feature-wise selection, extreme combinations (*e.g.*, with 16 branches with 32 dimensions) lead to results under the baseline. Thus, to increase the performances of such attention strategy, there is a compromise between the representation size of each branch and the number of branches (+1.1% over the baseline). In pre-attention with dimension-wise selection, all parameter combinations for this approach lead to better results than the baseline (+4.9% to +5.4%). The number of branches increases the performance with the log of the dictionary size on the contrary to the previous strategy.

5. Comparison to the state-of-the-art

In this section, we compare *DIABLO* to the state-of-the-art on 4 DML datasets, named Cub-200-2011 (Wah et al., 2011), Cars-196 (Krause et al., 2013), Stanford Online Products (Oh Song et al., 2016) and In-Shop Clothes Retrieval (Liu et al., 2016). For Cub-200-2011, Cars-196 and Stanford Online Products, we follow the standard splits from (Oh Song et al., 2016) and for In-Shop Clothes Retrieval we follow the one from (Liu et al., 2016). Especially, the Cub-200-2011 training set is composed of the first 100 classes (*i.e.*, the training and the validation sets from the ablation studies) for a total of 5864 images and its testing set is composed of the last 100 classes for a total of 5924 images. The Cars-196 training set is composed of the first 98 classes for a total of 8054 images and its testing set is composed of the last 98 classes for a total of 8131 images. The Stanford Online Products training set is composed of 11318 classes for a total of 59551 images and its testing set is composed of 11316 classes for a total of 60502 images. Finally, the In-Shop Clothes Retrieval training set is composed of 3997 classes for total of 25882 images and its testing set is composed 3985 classes and it is split into query set of 17218 images and a collection set of 12612 images. We report the Recall@K which evaluates, for a given query, if there is at least one image with the same label in the top-K retrieved images. We use $K \in \{1, 2, 4, 8, 16, 32\}$ for Cub-200-2011 and Cars-196, $K \in \{1, 10, 100, 1000\}$ for Stanford Online Products and $K \in \{1, 10, 20, 30, 40, 50\}$ for In-Shop Clothes Retrieval.

We report the results for Cub-200-2011 and Cars-196 in Table 1 and in Table 2 for Stanford Online Products and In-Shop Clothes Retrieval. *DIABLO* consistently improves the already strong baseline on the four datasets and for three different loss functions. *E.g.*, using the binomial loss, the baseline is improved from 59.6% to 63.9% (+4.3%) on Cub-200-2011 and from 78.8% to 85.4% (+6.6%) on the Cars-196 dataset. The same observation is made for both the other loss functions on these datasets.

Moreover, *DIABLO* leads to better results when compared to the similar approach ABE. Their best reported approach, ABE-8, is outperformed by *DIABLO* with $N = 8$ and $c = 64$ (total dimension 512) by 2.2% in R@1 on Cub-200-2011, by 1.5% on Stanford Online Products and by 4% on In-Shop Clothes Retrieval. We also report results with $N = 16$ and $c = 32$ (total dimension 512) which are further improved on these datasets: 1.1% in Recall@1 on the Cub-200-2011 and by 0.4% on the Cars-196 dataset, leading to the state-of-the-art on the four deep metric learning datasets.

6. Conclusion

In this paper, we have presented a dictionary-based attention method named *DIABLO* which consistently improves DML models. An ablation study is undertaken to evaluate the benefits of the feature-wise and the dimension-wise selections for two attention strategies named pre-attention and post-attention. We show that *DIABLO* consistently outperforms the baseline for three different loss functions on four datasets (Cub-200-2011, Cars-196, Stanford Online Products, and Inshop Clothes Retrieval). Moreover, it outperforms the current state-of-the-art methods on the four datasets.

Acknowledgments

Authors would like to acknowledge the COMUE Paris Seine University, the Cergy-Pontoise University and M2M Factory for their financial and technical support.

References

Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T., Sivic, J., 2016. Netvlad: Cnn architecture for weakly supervised place recognition, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1437 – 1451.

Carvalho, M., Cadène, R., Picard, D., Soulier, L., Thome, N., Cord, M., 2018. Cross-modal retrieval in the cooking context: Learning semantic text-image embeddings, in: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, pp. 35 – 44.

Duan, Y., Zheng, W., Lin, X., Lu, J., Zhou, J., 2018. Deep adversarial metric learning, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2780 – 2789.

Ge, W.a., 2018. Deep metric learning with hierarchical triplet loss, in: The European Conference on Computer Vision (ECCV), pp. 272 – 288.

Jacob, P., Picard, D., Histace, A., Klein, E., 2019a. Efficient codebook and factorization for second order representation learning, in: 2019 IEEE International Conference on Image Processing (ICIP), pp. 849 – 853.

Jacob, P., Picard, D., Histace, A., Klein, E., 2019b. Metric learning with horde: High-order regularizer for deep embeddings, in: The IEEE International Conference on Computer Vision (ICCV), pp. 6539 – 6548.

Kim, W., Goyal, B., Chawla, K., Lee, J., Kwon, K., 2018. Attention-based ensemble for deep metric learning, in: The European Conference on Computer Vision (ECCV), pp. 760 – 777.

Kingma, D.P., Ba, J., 2015. Adam: A method for stochastic optimization. International Conference on Learning Representations (ICLR) .

Krause, J., Stark, M., Deng, J., Fei-Fei, L., 2013. 3d object representations for fine-grained categorization, in: 4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13), Sydney, Australia. pp. 554 – 561.

Lin, X., Duan, Y., Dong, Q., Lu, J., Zhou, J., 2018. Deep variational metric learning, in: The European Conference on Computer Vision (ECCV), pp. 714 – 729.

Liu, Z., Luo, P., Qiu, S., Wang, X., Tang, X., 2016. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1096 – 1104.

Movshovitz-Attias, Y., Toshev, A., Leung, T.K., Ioffe, S., Singh, S., 2017. No fuss distance metric learning using proxies, in: The IEEE International Conference on Computer Vision (ICCV), pp. 360 – 368.

Oh Song, H., Xiang, Y., Jegelka, S., Savarese, S., 2016. Deep metric learning via lifted structured feature embedding, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4004 – 4012.

Opitz, M., Waltner, G., Possegger, H., Bischof, H., 2017. Bier - boosting independent embeddings robustly, in: The IEEE International Conference on Computer Vision (ICCV), pp. 5199 – 5208.

Opitz, M., Waltner, G., Possegger, H., Bischof, H., 2018. Deep metric learning with BIER: boosting independent embeddings robustly. IEEE Transactions on Pattern Analysis and Machine Intelligence .

Roy, S.K., Harandi, M., Nock, R., Hartley, R., 2019. Siamese networks: The tale of two manifolds, in: The IEEE International Conference on Computer Vision (ICCV), pp. 3046 – 3055.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2015. Going deeper with convolutions, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1 – 9.

Ustinova, E., Lempitsky, V., 2016. Learning deep embeddings with histogram loss, in: Lee, D.D., Sugiyama, M., Luxburg, U.V., Guyon, I., Garnett, R. (Eds.), Advances in Neural Information Processing Systems 29, pp. 4170–4178.

Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S., 2011. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001. California Institute of Technology.

Wang, J., Zhou, F., Wen, S., Liu, X., Lin, Y., 2017. Deep metric learning with angular loss, in: The IEEE International Conference on Computer Vision (ICCV), pp. 2612 – 2620.

Wehrmann, J., Barros, R.C., 2018. Bidirectional retrieval made simple, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7718 – 7726.

Xu, X., Yang, Y., Deng, C., Zheng, F., 2019. Deep asymmetric metric learning via rich relationship mining, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4076 – 4085.

Yuan, Y., Yang, K., Zhang, C., 2017. Hard-aware deeply cascaded embedding, in: The IEEE International Conference on Computer Vision (ICCV), pp. 814 – 823.

Zhao, Y., Jin, Z., Qi, G.j., Lu, H., Hua, X.s., 2018. An adversarial approach to hard triplet generation, in: The European Conference on Computer Vision (ECCV), pp. 508 – 524.

Zheng, W., Chen, Z., Lu, J., Zhou, J., 2019. Hardness-aware deep metric learning, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 72 – 81.

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A., 2016. Learning deep features for discriminative localization, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2921 – 2929.

Zhou, J., Yu, P., Tang, W., Wu, Y., 2017. Efficient online local metric adaptation via negative samples for person re-identification, in: The IEEE International Conference on Computer Vision (ICCV), pp. 2439 – 2447.