Metric Learning With HORDE: High-Order Regularizer for Deep Embeddings

Pierre Jacob[1], David Picard[1,2], Aymeric Histace[1], Edouard Klein[3]
[1]ETIS UMR 8051, Université Paris Seine, UCP, ENSEA, CNRS, F-95000, Cergy, France
[2]LIGM, UMR 8049, École des Ponts, UPE, Champs-sur-Marne, France
[3]C3N, Pôle Judiciaire de la Gendarmerie Nationale, 5 boulevard de l'Hautil, 95000 Cergy, France
{pierre.jacob, picard, aymeric.histace}@ensea.fr

Abstract

Learning an effective similarity measure between image representations is key to the success of recent advances in visual search tasks (e.g. verification or zero-shot learning). Although the metric learning part is well addressed, this metric is usually computed over the average of the extracted deep features. This representation is then trained to be discriminative. However, these deep features tend to be scattered across the feature space. Consequently, the representations are not robust to outliers, object occlusions, background variations, etc. In this paper, we tackle this scattering problem with a distribution-aware regularization named HORDE[1]. This regularizer enforces visually-close images to have deep features with the same distribution which are well localized in the feature space. We provide a theoretical analysis supporting this regularization effect. We also show the effectiveness of our approach by obtaining state-of-the-art results on 4 well-known datasets (Cub-200-2011, Cars-196, Stanford Online Products and Inshop Clothes Retrieval).

1. Introduction

Deep Metric Learning (DML) is an important yet challenging topic in the Computer Vision community with numerous applications such as visual product search [15, 18], multi-modal retrieval [1, 31], face verification and clustering [22], person or vehicle identification [14, 38]. To deal with such applications, a DML method aims to learn an embedding space where all the visually-related images (*e.g.*, images of the same car model) are close to each other and dissimilar ones (*e.g.*, images of two cars from the same brand but from different models) are far apart.

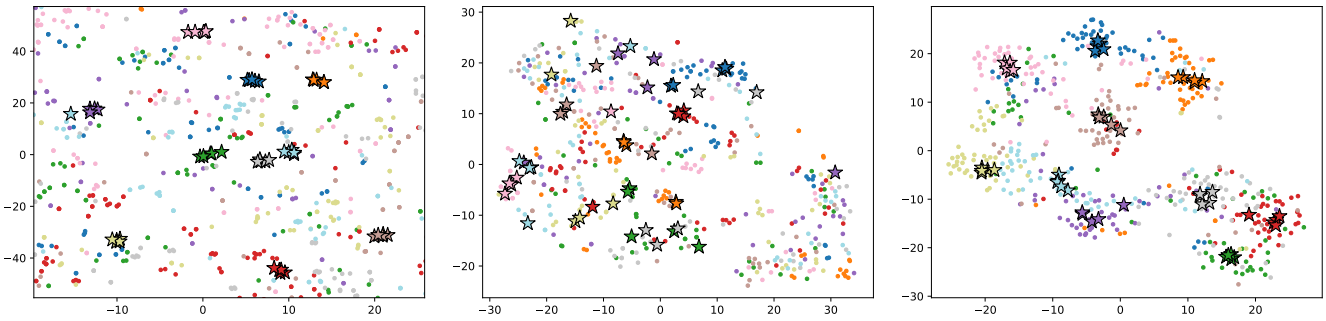
Recent contributions in DML can be divided into three categories. A first category includes methods that focus on batch construction to maximize the number of pairs or triplets available to compute the similarity (*e.g.*, N-pair loss [23]). A second category involves the design of loss functions to improve the generalization (*e.g.*, binomial deviance [26]). The third category covers ensemble methods that tackle the embedding space diversity (*e.g.*, BIER [19]).

This similarity metric is trained jointly with the image representation which is computed using deep neural network architectures such as GoogleNet [25] or BN-Inception [8]. For all of these networks, the image representations are obtained by the aggregation of the deep features using a Global Average Pooling [37]. Hence, the deep features are summarized using the sample mean, and the training process makes sure that the sample mean is discriminative enough for the target task.

Our insight is that ignoring the characteristics of the deep feature distribution leads to a lack of distinctiveness in the deep features. We illustrate this phenomenon in Figure 1. In Figure 1a, we train a DML model on MNIST and plot both the deep features and the image representations from a set of images sampled from the training set. We observe that the representations are perfectly organized while the deep features are in contrast scattered in the entire space. As the representations are obtained using the sample mean only, they are sensitive to outliers or sampling problems (occlusions, illumination, background variation, etc.), which we refer to as the *scattering problem*. We illustrate this problem in Figure 1b where the representations are computed using the same architecture but by sampling only $1/6$-th of the original deep features. As we can see, the resulting representations are no longer correctly organized.

In this paper, we propose *HORDE*, a High-Order Regularizer for Deep Embeddings which tackles this scattering problem. By minimizing (resp. maximizing) the distance between high-order moments of the deep feature distributions, this DML regularizer enforces deep feature distributions from similar (resp. dissimilar) images to be nearly identical (resp. to not overlap). As illustrated in Figure 1c, our *HORDE* regularizer produces well localized features,

[1]Code is available at https://github.com/pierre-jacob/ICCV2019-Horde



(a) Representations using all deep features

(b) Representations using $1/6$-th of deep features

(c) Representations using $1/6$-th of deep features trained with *HORDE*

Figure 1: 2D visualizations of representations (stars) and deep features (points) using t-SNE computed from a DML architecture on MNIST dataset with **one color per class**. Representations and features come from the **training set**. Figure 1a shows discriminative representations but with scattered deep features (Remark the scale of the axes). Figure 1b shows representations computed with $1/6$-th of the deep features, leading to a disorganized space. Figure 1c shows the same model trained with *HORDE*: the deep features are well concentrated and the representations computed using $1/6$-th of the deep features are organized according to the classes (best viewed on a computer screen).

leading to robust image representations even if they are computed using only $1/6$-th of the original deep features.

Our contributions are the following: First, we propose a High-Order Regularizer for Deep Embeddings (*HORDE*) that reduces the scattering problem and allows the sample mean to be a robust representation. We provide a theoretical analysis in which we support this claim by showing that *HORDE* is a lower bound of the Wasserstein distance between the deep feature distributions while also being an upper-bound of their Maximum Mean Discrepancy. Second, we show that *HORDE* consistently improves DML with varying loss functions, even when considering ensemble methods. Using *HORDE*, we are able to obtain state of the art results on four standard DML datasets (Cub-200-2011 [27], Cars-196 [12], In-Shop Clothes Retrieval [15] and Stanford Online Products [18]).

The remaining of this paper is organized as follows: In section 2, we review recent works on deep metric learning and how our approach differs. In section 3, after an overview of our proposed method, we present the practical implementation of *HORDE* as well as a theoretical analysis. In section 4 we compare our proposed architecture with the state-of-the-art on four image retrieval datasets. We show the benefit of *HORDE* regularization for different loss functions and an ensemble method. In section 5 we conduct extensive experiments to demonstrate the robustness of our regularization and its statistical consistency.

2. Related Work

In DML, we jointly learn the image representations and an embedding in such a way that the Euclidean distance corresponds with the semantic content of the images. Current approaches use a pre-trained CNN to produce deep features, then they aggregate these features using Global Average Pooling [37]. Finally they learn the target representation with a linear projection. The whole network is fine-tuned to solve the metric learning task according to three criteria: a loss function, a sampling strategy and an ensemble method.

Regarding the loss function, popular approaches consider pairs [3] or triplets [22] of similar/dissimilar samples. Recent works generalize these loss functions to larger tuples [2, 18, 23, 26] or improve the design [28, 29, 34]. The sampling of the training tuples receive plenty of attention [18, 22, 23], either through mining [7, 22], proxy based approximations [16, 17] or hard negative generation [4, 13]. Finally, ensemble methods have recently become an increasingly popular way of improving the performances of DML architectures [11, 19, 33, 35]. Our proposed *HORDE* regularizer is a complementary approach. We show in section 4 that it consistently improves these popular DML models.

Recent approaches also consider a distribution analysis for DML [21, 13]. Contrarily to us, they only consider the distribution of the representations to design a loss function or a hard negative generator but they do not take into account the distribution of the underlying deep features. Consequently, they do not address the scattering problem. More precisely, Magnet loss [21] proposes to better represent a given class manifold by learning a K-mode distribution instead of the standard uni-mode assumption. To that aim, the per-class distribution is approximated using K-means clustering. The proposed loss tries to minimize the distance between a representation and its nearest class mode and tries to maximize the distance between all modes of all other

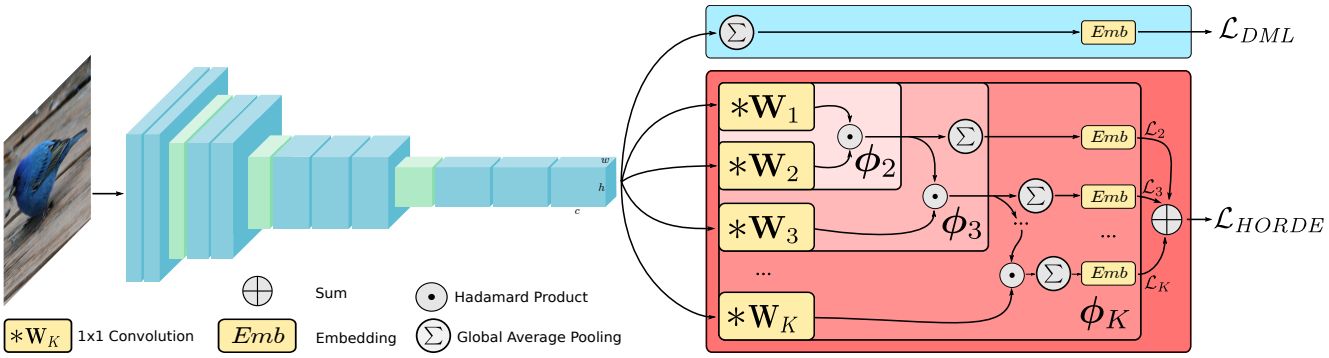


Figure 2: Global overview of our *HORDE* architecture. The deep convolutional neural network extracts $h \times w \times c$ deep features. The standard architecture (top blue block) relies on a global average pooling and an embedding before computing the $\mathcal{L}_{\text{DML}}$ loss. The bottom red block is our *HORDE* regularizer, composed by the approximation of all high-order moments ϕ_k, global average pooling and embeddings before computing the sum of each $\mathcal{L}_k$ loss.

classes. However, since the magnet loss is directly applied to the sample means of the deep features, it leads to the scattering problem illustrated in Figure 1. In DVML [13], the authors assume that the representations follow a per-class Gaussian distribution. They propose to estimate the parameters of these distributions using a variational auto-encoder approach. Then, by sampling from a Gaussian distribution with the learned parameters, they are able to generate artificial hard samples to train the network. However, no assumption is made on the distribution of the deep features, which leads to the scattering problem illustrated in Figure 1 (see also [13], Figure 1). In contrast, we show that focusing on the distribution of the deep features reduces the scattering problem and improves the performances of DML architectures.

In the next section, we first give an overview of the proposed *HORDE* regularization. Then, we describe the practical implementation of the high-order moments computation. Finally, we give theoretical insights which support the regularization effect of *HORDE*.

3. Proposed High-Order Regularizer

We first give an overview of the proposed method in Figure 2. We start by extracting a deep feature map of size $h \times w \times c$ using a CNN where h and w are the height and width of the feature map and c is the deep features dimension. Following standard DML practices, these features are aggregated using a Global Average Pooling to build the image representation and are projected into an embedding space before a similarity-based loss function is computed over these representations (top-right blue box in Figure 2).

In *HORDE*, we directly optimize the distribution of the deep features by minimizing (respectively maximizing) a distance between the deep feature distributions of similar images (respectively dissimilar images). We approximate the deep feature distribution distance by computing high-order moments (bottom-right red box in Figure 2). We recursively approximate the high-order moments and we compute an embedding after each of these approximations. Then, we apply a DML loss function on each of these embeddings.

3.1. High-order computation

In practice, the computation of high-order moments is very intensive due to their high dimension. Furthermore, it has been shown in [9, 19] that an independence assumption over all high-order moment components is unrealistic. Hence, we rely on factorization schemes to approximate their computation, such as Random Maclaurin (RM) [10]. The RM algorithm relies on a set of random projectors to approximate the inner product between two high-order moments. In the case of the second-order, we sample two independent random vectors $\boldsymbol{w}_1, \boldsymbol{w}_2 \sim \mathcal{W}$ where $\mathcal{W}$ is a uniform distribution in $\{-1, +1\}$. For two non random vectors $\boldsymbol{x}$ and $\boldsymbol{y}$, the inner product between their second-order moments can be approximated as:

$$\begin{aligned}\mathbb{E}_{\boldsymbol{w}_1,\boldsymbol{w}_2\sim\mathcal{W}}[\phi_2(\boldsymbol{x})\phi_2(\boldsymbol{y})] &= \langle \boldsymbol{x}\ ;\ \boldsymbol{y}\rangle^2 \\ &= \langle \boldsymbol{x}\otimes\boldsymbol{x}\ ;\ \boldsymbol{y}\otimes\boldsymbol{y}\rangle \end{aligned} \quad (1)$$

where $\otimes$ is the Kronecker product, $\mathbb{E}_{\boldsymbol{w}_1,\boldsymbol{w}_2\sim\mathcal{W}}$ is the expectation over the random vectors $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ which follow the distribution $\mathcal{W}$ and $\phi_2(\boldsymbol{x}) = \langle \boldsymbol{w}_1\ ;\ \boldsymbol{x}\rangle \langle \boldsymbol{w}_2\ ;\ \boldsymbol{x}\rangle$. This approach easily holds to estimate any inner product between K-th moments:

$$\begin{aligned}\mathbb{E}_{\boldsymbol{w}_k\sim\mathcal{W}}[\phi_K(\boldsymbol{x})\phi_K(\boldsymbol{y})] &= \langle \boldsymbol{x}\ ;\ \boldsymbol{y}\rangle^K \\ &= \left\langle \underbrace{\boldsymbol{x}\otimes\cdots\otimes\boldsymbol{x}}_{K \text{ times}}\ ;\ \underbrace{\boldsymbol{y}\otimes\cdots\otimes\boldsymbol{y}}_{K \text{ times}} \right\rangle \end{aligned} \quad (2)$$

where $\phi_K(\boldsymbol{x})$ is computed as:

$$\phi_K(\boldsymbol{x}) = \prod_{k=1}^{K} \langle \boldsymbol{w}_k \; ; \; \boldsymbol{x} \rangle \tag{3}$$

In practice, we approximate the expectation of this quantity by using the sample mean over d sets of these random projectors. That is, we sample independent random matrices $\boldsymbol{W}_1, \boldsymbol{W}_2, ..., \boldsymbol{W}_K \in \mathbb{R}^{c \times d}$ and we compute the vector $\boldsymbol{\phi}_K(\boldsymbol{x}) \in \mathbb{R}^d$ that approximates the K-th moments of $\boldsymbol{x}$ with the following equation:

$$\boldsymbol{\phi}_K(\boldsymbol{x}) = \left(\boldsymbol{W}_1^\top \boldsymbol{x}\right) \odot \left(\boldsymbol{W}_2^\top \boldsymbol{x}\right) \odot \cdots \odot \left(\boldsymbol{W}_K^\top \boldsymbol{x}\right) \tag{4}$$

where $\odot$ is the Hadamard (element-wise) product. Thus, the inner product between the K-th moments is:

$$\langle \boldsymbol{x} \; ; \; \boldsymbol{y} \rangle^K \approx \frac{1}{d} \langle \boldsymbol{\phi}_K(\boldsymbol{x}) \; ; \; \boldsymbol{\phi}_K(\boldsymbol{y}) \rangle \tag{5}$$

However, Random Maclaurin produces a consistent estimator independently of the analyzed distributions, and thus also encodes non informative high-order moment components. To ignore these non-informative components, the projectors $\boldsymbol{W}_k$ can be learned from the data. However, the high number of parameters in $\mathcal{O}(K^2cd)$ makes it difficult to learn a consistent estimator, as we empirically show in subsection 5.2. We solve this problem by computing the high-order moment approximation using the following recursion:

$$\boldsymbol{\phi}_k(\boldsymbol{x}) = \boldsymbol{\phi}_{k-1}(\boldsymbol{x}) \odot \left(\boldsymbol{W}_k^\top \boldsymbol{x}\right) \tag{6}$$

This last equation leads to the proposed cascaded architecture for *HORDE* summarized in Algorithm 1. We empirically show in subsection 5.2 that this recursive approach produces a consistent estimator of the informative high-order moment components.

Then, the *HORDE* regularizer consists in computing a DML-like loss function on each of the high-order moments, such that similar (respectively dissimilar) images have similar (respectively dissimilar) high-order moments:

$$\mathcal{L}_{HORDE} = \sum_{k=2}^{K} \mathcal{L}_k \left(\mathbb{E}_{\boldsymbol{x} \sim \mathcal{I}}[\boldsymbol{\phi}_k(\boldsymbol{x})], \mathbb{E}_{\boldsymbol{y} \sim \mathcal{J}}[\boldsymbol{\phi}_k(\boldsymbol{y})] \right) \tag{7}$$

In practice, we cannot compute the expectation $\mathbb{E}_{\boldsymbol{x} \sim \mathcal{I}}[\boldsymbol{\phi}_k(\boldsymbol{x})]$ since the distribution of $\boldsymbol{x}$ is unknown. We propose to estimate it using the empirical estimator:

$$\mathcal{L}_{HORDE}(\mathcal{I}, \mathcal{J}) = \sum_{k=2}^{K} \mathcal{L}_k \left(\frac{1}{|\mathcal{I}|} \sum_{\boldsymbol{x}_i \in \mathcal{I}} \boldsymbol{\phi}_k(\boldsymbol{x}_i), \frac{1}{|\mathcal{J}|} \sum_{\boldsymbol{x}_j \in \mathcal{J}} \boldsymbol{\phi}_k(\boldsymbol{y}_j) \right) \tag{8}$$

Algorithm 1 High-order moments computation

Require: $\boldsymbol{W}_1, \ldots, \boldsymbol{W}_K$ sampled from $\{-1; +1\}$
Ensure: K first moments approximations
procedure APPROXMOMENTS($\boldsymbol{x}$)
  $\boldsymbol{\phi}_2(\boldsymbol{x}) \leftarrow \frac{1}{\sqrt{d}} \left(\boldsymbol{W}_1^\top \boldsymbol{x}\right) \odot \left(\boldsymbol{W}_2^\top \boldsymbol{x}\right)$
  $k \leftarrow 3$
  while $k \leq K$ **do**
    $\boldsymbol{\phi}_k(\boldsymbol{x}) = \boldsymbol{\phi}_{k-1}(\boldsymbol{x}) \odot \left(\boldsymbol{W}_k^\top \boldsymbol{x}\right)$
    $k \leftarrow k + 1$
  end while
  return $\boldsymbol{\phi}_2(\boldsymbol{x}), \ldots, \boldsymbol{\phi}_K(\boldsymbol{x})$
end procedure

where $\{\boldsymbol{x}_i \in \mathcal{I}\}$ and $\{\boldsymbol{x}_j \in \mathcal{J}\}$ are the sets of deep features extracted from images $\mathcal{I}$ and $\mathcal{J}$.

Hence, the DML model is trained on a combination of a standard DML loss and the *HORDE* regularizer on pairs of images $\mathcal{I}$ and $\mathcal{J}$:

$$\mathcal{L}(\mathcal{I}, \mathcal{J}) = \mathcal{L}_{DML}(\mathcal{I}, \mathcal{J}) + \mathcal{L}_{HORDE}(\mathcal{I}, \mathcal{J}) \tag{9}$$

This can easily be extended to any tuple based loss function. In practice, we use the same DML loss function for *HORDE* ($\forall k, \mathcal{L}_k = \mathcal{L}_{DML}$).

Remark also that at inference time, the image representation $\boldsymbol{\phi}_1(\mathcal{I})$ consists only of the sample mean of the deep features:

$$\boldsymbol{\phi}_1(\mathcal{I}) = \frac{1}{|\mathcal{I}|} \sum_{\boldsymbol{x}_i \in \mathcal{I}} \boldsymbol{x}_i, \tag{10}$$

and the *HORDE* part of the model can be discarded.

3.2. Theoretical analysis

In this section, we show that optimizing distances between high-order moments is directly related to the Maximum Mean Discrepancy (MMD) [6] and the Wasserstein distance. We consider the Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}$ of distributions $f : \Omega \mapsto \mathbb{R}^+$ defined on the compact $\Omega \subset \mathbb{R}^c$, endowed with the Gaussian kernel $k(\boldsymbol{x}, \boldsymbol{y}) = e^{-\gamma \|\boldsymbol{x} - \boldsymbol{y}\|^2}$. An image is then represented as a distribution $\mathcal{I} \in \mathcal{H}$ from which we can sample a set of deep features $\{\boldsymbol{x}_i \in \Omega\}_i$. We denote $\mathbb{E}_{\boldsymbol{x} \sim \mathcal{I}}[\boldsymbol{x}] \in \mathbb{R}^c$ the expectation of $\boldsymbol{x}$ sampled from $\mathcal{I}$. The high-order moments are denoted using their vectorized forms, that is $\mathbb{E}_{\boldsymbol{x} \sim \mathcal{I}}[\boldsymbol{x}^{\otimes k}] \in \mathbb{R}^{c^k}$ where $\boldsymbol{x}^{\otimes 2} = \boldsymbol{x} \otimes \boldsymbol{x}$, $\boldsymbol{x}^{\otimes 3} = \boldsymbol{x} \otimes \boldsymbol{x} \otimes \boldsymbol{x}$, *etc.* By extension, we use $\mathbb{E}_{\boldsymbol{x} \sim \mathcal{I}}[\boldsymbol{x}^{\otimes 1}]$ for the mean. We assume that all moments exist for every distributions in $\mathcal{H}$ and we note, $\forall \mathcal{I} \in \mathcal{H}$:

$$\max_k \| \mathbb{E}_{\boldsymbol{x} \sim \mathcal{I}}[\boldsymbol{x}^{\otimes k}] \|^2 = K < \infty \tag{11}$$

Following [6], the MMD between two distributions $\mathcal{I}$ and $\mathcal{J}$ is expressed as:

$$\text{MMD}(\mathcal{I}, \mathcal{J}) = \sup_{T} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{I}}[T(\boldsymbol{x})] - \mathbb{E}_{\boldsymbol{y}\sim\mathcal{J}}[T(\boldsymbol{y})] \quad (12)$$

The MMD searches for a transform T that maximizes the difference between the expectation of two distributions. Intuitively, a low MMD implies that both distributions are concentrated in the same regions of the feature space.

In the following theorem, we show that the distance over high-order moments is an upper-bound of the squared MMD (the proof mainly follows [6]):

Theorem 1. *There exists $A \in \mathbb{R}^{+*}$ such that, for every distributions $\mathcal{I}, \mathcal{J} \in \mathcal{H}$, the MMD is bounded from above by the p first moments of $\mathcal{I}$ and $\mathcal{J}$ by:*

$$MMD^2(\mathcal{I}, \mathcal{J}) \leq A \sum_{k=1}^{p} \left\| \mathbb{E}_{\boldsymbol{x}\sim\mathcal{I}}[\boldsymbol{x}^{\otimes k}] - \mathbb{E}_{\boldsymbol{y}\sim\mathcal{J}}[\boldsymbol{y}^{\otimes k}] \right\|^2 + 1 + o(\frac{\gamma^p K}{p!}) \quad (13)$$

Proof. As the MMD is a distance on the RKHS $\mathcal{H}$ [6], the square of the MMD can be re-written such as:

$$\text{MMD}^2(\mathcal{I}, \mathcal{J}) = \|\mathbb{E}_{\boldsymbol{x}\sim\mathcal{I}}[\boldsymbol{\phi}(\boldsymbol{x})] - \mathbb{E}_{\boldsymbol{y}\sim\mathcal{J}}[\boldsymbol{\phi}(\boldsymbol{y})]\|^2_{\mathcal{H}} \quad (14)$$

where $\boldsymbol{\phi}$ is defined using the kernel trick $k(\boldsymbol{x}, \boldsymbol{y}) = \langle \boldsymbol{\phi}(\boldsymbol{x}) \,;\, \boldsymbol{\phi}(\boldsymbol{y}) \rangle$. Then, we can approximate the Gaussian kernel using its Taylor expansion:

$$\begin{aligned} k(\boldsymbol{x}, \boldsymbol{y}) &= \exp(-\gamma\|\boldsymbol{x}\|^2 - \gamma\|\boldsymbol{y}\|^2) \exp(2\gamma \langle \boldsymbol{x} \,;\, \boldsymbol{y} \rangle) \\ &= \exp(-\gamma\|\boldsymbol{x}\|^2 - \gamma\|\boldsymbol{y}\|^2) \sum_{k=0}^{+\infty} \frac{(2\gamma)^k}{k!} \langle \boldsymbol{x}; \boldsymbol{y} \rangle^k \\ &\leq 1 + \sum_{k=1}^{+\infty} a_k \left\langle \boldsymbol{x}^{\otimes k}; \boldsymbol{y}^{\otimes k} \right\rangle \end{aligned} \quad (15)$$

where $a_k = \frac{(2\gamma)^k}{k!} > 0$. Thus, we can define $\boldsymbol{\phi}$ as the direct sum of all weighted and vectorized moments:

$$\boldsymbol{\phi}(\boldsymbol{x}) = \bigoplus_{k=1}^{+\infty} \sqrt{a_k} \boldsymbol{x}^{\otimes k} \quad (16)$$

As all moments exist, we can swap the expectation and the direct sum. Moreover, since the sequence $a_k = \frac{(2\gamma)^k}{k!} \to 0$ when $k \to +\infty$ and the moments are bounded by K, the higher-order moment contributions become negligible compared to the p first moments. Thus, we have:

$$\begin{aligned} \text{MMD}^2(\mathcal{I}, \mathcal{J}) &\leq 1 + \sum_{k=1}^{+\infty} a_k \|\mathbb{E}_{\boldsymbol{x}\sim\mathcal{I}}[\boldsymbol{x}^{\otimes k}] - \mathbb{E}_{\boldsymbol{y}\sim\mathcal{J}}[\boldsymbol{y}^{\otimes k}]\|^2 \\ &\leq A \sum_{k=1}^{p} \|\mathbb{E}_{\boldsymbol{x}\sim\mathcal{I}}[\boldsymbol{x}^{\otimes k}] - \mathbb{E}_{\boldsymbol{x}\sim\mathcal{J}}[\boldsymbol{x}^{\otimes k}]\|^2 \\ &\quad + 1 + o(\frac{\gamma^p K}{p!}) \end{aligned} \quad (17)$$

where $A = \max_k a_k$. □

This result implies that regularizing high-order moments to be similar enforces similar images to have deep features sampled from similar distributions. Thus, deep features from similar images have a higher probability of being concentrated in the same regions of the feature space.

Next, we show a converse relation between high-order moments and the Wasserstein distance:

Theorem 2. *There exists $a \in \mathbb{R}^{+*}$ such that, for every distributions $\mathcal{I}, \mathcal{J} \in \mathcal{H}$, the squared Wasserstein distance is bounded from below by the p first moments of $\mathcal{I}$ and $\mathcal{J}$ by:*

$$W_1^2(\mathcal{I}, \mathcal{J}) \geq a \sum_{k=1}^{p} \left\| \mathbb{E}_{\boldsymbol{x}\sim\mathcal{I}}[\boldsymbol{x}^{\otimes k}] - \mathbb{E}_{\boldsymbol{y}\sim\mathcal{J}}[\boldsymbol{y}^{\otimes k}] \right\|^2 - o(\frac{\gamma^p}{p!}) \quad (18)$$

Proof. Similarly to the Theorem 1, we can lower-bound the Gaussian kernel using its Taylor expansion:

$$k(\boldsymbol{x}, \boldsymbol{y}) \geq \alpha \sum_{k=1}^{+\infty} a_k \left\langle \boldsymbol{x}^{\otimes k}; \boldsymbol{y}^{\otimes k} \right\rangle$$

where $\alpha = \exp(-2\gamma K)$ and $a_k = \frac{(2\gamma)^k}{k!} > 0$. Then, by using the definition of $\boldsymbol{\phi}$ from Equation 16, a lower-bound for the MMD is:

$$\begin{aligned} \text{MMD}^2(\mathcal{I}, \mathcal{J}) &\geq \alpha\, a' \sum_{k=1}^{p} \|\mathbb{E}_{\boldsymbol{x}\sim\mathcal{I}}[\boldsymbol{x}^{\otimes k}] - \mathbb{E}_{\boldsymbol{y}\sim\mathcal{J}}[\boldsymbol{y}^{\otimes k}]\|^2 \\ &\quad - o(\frac{\gamma^p K}{p!}) \end{aligned} \quad (19)$$

where $a' = \min_k a_k$. Finally, the MMD is a lower-bound of the Wasserstein distance [24]:

$$\sqrt{K} W_1(\mathcal{I}, \mathcal{J}) \geq \text{MMD}(\mathcal{I}, \mathcal{J}) \quad (20)$$

By combining Equation 19 and Equation 20, we get the expected lower-bound:

$$W_1^2(\mathcal{I}, \mathcal{J}) \geq a \sum_{k=1}^{p} \|\mathbb{E}_{\boldsymbol{x}\sim\mathcal{I}}[\boldsymbol{x}^{\otimes k}] - \mathbb{E}_{\boldsymbol{y}\sim\mathcal{J}}[\boldsymbol{y}^{\otimes k}]\|^2 - o(\frac{\gamma^p}{p!}) \quad (21)$$

where $a = \frac{\alpha\, a'}{K}$. □

		Cub-200-2011						Cars-196					
Backbone	R@	1	2	4	8	16	32	1	2	4	8	16	32
Loss functions or mining strategies													
GoogleNet	Angular loss [29]	54.7	66.3	76.0	83.9	-	-	71.4	81.4	87.5	92.1	-	-
	HDML [36]	53.7	65.7	76.7	85.7	-	-	79.1	87.1	92.1	95.5	-	-
	DAMLRMM [32]	55.1	66.5	76.8	85.3	-	-	73.5	82.6	89.1	93.5	-	-
	DVML [13]	52.7	65.1	75.5	84.3	-	-	82.0	88.4	93.3	96.3	-	-
	HTL [5]	57.1	68.8	78.7	86.5	92.5	95.5	81.4	88.0	92.7	95.7	97.4	99.0
	contrastive loss (Ours)	55.0	67.9	78.5	86.2	92.2	96.0	72.2	81.3	88.1	92.6	95.6	97.8
	contrastive loss + *HORDE*	57.1	69.7	79.2	87.4	92.8	96.3	76.2	85.2	90.8	95.0	97.2	98.8
	Triplet loss (Ours)	50.5	63.3	74.8	84.6	91.2	95.0	65.2	75.8	83.7	89.4	93.6	96.5
	Triplet loss + *HORDE*	53.6	65.0	76.0	85.2	91.1	95.3	74.0	82.9	89.4	93.7	96.4	98.0
	Binomial Deviance (Ours)	55.9	67.6	78.3	86.4	92.3	96.1	78.2	86.0	91.3	94.6	97.1	98.3
	Binomial Deviance + *HORDE*	**58.3**	**70.4**	**80.2**	**87.7**	**92.9**	**96.3**	81.5	**88.5**	92.7	95.4	**97.4**	98.6
	Binomial Deviance + *HORDE*†	**59.4**	**71.0**	**81.0**	**88.0**	**93.1**	**96.5**	**83.2**	**89.6**	**93.6**	**96.3**	**98.0**	**98.8**
BN-Inception	Multi-similarity loss [30]	65.7	77.0	**86.3**	**91.2**	**95.0**	97.3	84.1	90.4	94.0	96.5	98.0	98.9
	contrastive loss + *HORDE*	**66.3**	76.7	84.7	90.6	94.5	96.7	83.9	90.3	94.1	96.3	98.3	99.2
	contrastive loss + *HORDE*†	**66.8**	**77.4**	85.1	91.0	94.8	97.3	**86.2**	**91.9**	**95.1**	**97.2**	**98.5**	**99.4**
Ensemble Methods													
GoogleNet	HDC [35]	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 [19]	55.3	67.2	76.9	85.1	91.7	95.5	78.0	85.8	91.1	95.1	97.3	98.7
	A-BIER [20]	57.5	68.7	78.3	86.2	91.9	95.5	82.0	89.0	93.2	96.1	97.8	98.7
	ABE [11]	60.6	71.5	79.8	87.4	-	-	85.2	90.5	94.0	96.1	-	-
	ABE (Ours)	60.0	71.8	81.4	88.9	93.4	96.6	79.2	87.1	92.0	95.2	97.3	98.7
	ABE + *HORDE*	**62.7**	**74.3**	**83.4**	**90.2**	**94.6**	**96.9**	**86.4**	**92.0**	**95.3**	**97.4**	**98.6**	**99.3**
	ABE + *HORDE*†	**63.9**	**75.7**	**84.4**	**91.2**	**95.3**	**97.6**	**88.0**	**93.2**	**96.0**	**97.9**	**99.0**	**99.5**

Table 1: Comparison with the state-of-the-art on Cub-200-2011 and Cars-196 datasets. Results in percents. † means that the test scores are computed using all the high-order moments (concatenation + PCA to the embedding size).

		Stanford Online Products				In-Shop Clothes Retrieval					
Backbone	R@	1	10	100	1000	1	10	20	30	40	50
GoogleNet	Angular loss [29]	70.9	85.0	93.5	98.0	-	-	-	-	-	-
	HDML [36]	68.7	83.2	92.4	-	-	-	-	-	-	-
	DAMLRMM [32]	69.7	85.2	93.2	-	-	-	-	-	-	-
	DVML [13]	70.2	85.2	93.8	-	-	-	-	-	-	-
	HTL [5]	**74.8**	**88.3**	**94.8**	**98.4**	80.9	94.3	95.8	97.2	97.4	97.8
	Binomial Deviance (Ours)	67.4	81.7	90.2	95.4	81.3	94.2	95.9	96.7	97.2	97.6
	Binomial Deviance + *HORDE*	72.6	85.9	93.7	97.9	**84.4**	**95.4**	**96.8**	**97.4**	**97.8**	**98.1**
BN-Inception	Multi-similarity loss [30]	78.2	90.5	96.0	98.7	89.7	97.9	98.5	98.8	99.1	99.2
	contrastive loss + *HORDE*	**80.1**	**91.3**	**96.2**	**98.7**	**90.4**	97.8	98.4	98.7	98.9	99.0

Table 2: Comparison with the state-of-the-art on Stanford Online Products and In-Shop Clothes Retrieval. Results in percents.

Hence, regularizing high-order moments to be dissimilar enforces dissimilar images to have deep features sampled from different distributions. As such, deep features are more distinctive as they are sampled from different regions of the feature space for dissimilar images. This is illustrated in Figure 1c ($p = 5$) compared to Figure 1a ($p = 1$).

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

We present the benefits of our method by comparing our results with the state-of-the-art on four datasets, namely CUB-200-2011 (CUB) [27], Cars-196 (CARS) [12], Stanford Online Products (SOP) [18] and In-Shop Clothes Retrieval (INSHOP) [15]. We report the Recall@K (R@K) on the standard DML splits associated with these datasets. Following standard practices, we use GoogleNet [25] as a backbone network and we add a fully connected layer at the end for the embedding. For CUB and CARS, we train *HORDE* using 5 high-order moments with 5 classes and 8 images per instance per batch. For SOP and INSHOP, we use 4 high-order moments with a batch size of 2 images and 40 different classes as there are classes with only 2 images in these datasets. We use 256×256 crops and the following data augmentation at training time: multi-resolution where the resolution is uniformly sampled in $[80\%, 180\%]$ of the crop size, random crop and horizontal flip. At inference time, we only use the images resized to 256×256. For *HORDE*, we use 8192 dimensions for all high-order moments and we fix all embedding dimensions to 512. Finally, we take advantage of the high-order moments at testing time by concatenating them together. To be fair with other methods, we reduce their dimensionality to 512 using a PCA.

k	1	2		3			4				5					6					
n	1	1	2	1	2	3	1	2	3	4	1	2	3	4	5	1	2	3	4	5	6
R@1	55.9	**57.8**	58.6	56.8	58.0	56.9	**57.8**	58.8	57.6	56.1	57.4	57.7	56.8	56.3	53.3	57.4	57.9	57.1	55.6	54.4	50.7
R@2	67.6	69.5	70.4	68.1	69.4	68.7	69.2	70.6	70.0	68.5	68.8	69.9	69.3	68.1	65.4	**69.9**	70.6	70.5	68.9	66.2	63.0
R@4	78.3	79.0	79.8	78.3	78.8	78.1	78.6	79.9	79.2	78.1	78.7	78.8	79.2	78.0	75.9	**79.4**	80.0	79.9	78.7	76.5	74.0
R@8	86.4	86.7	87.2	86.2	86.7	86.6	86.5	87.2	87.0	85.5	**87.0**	87.1	87.1	86.5	84.2	86.9	87.4	87.4	86.7	85.4	82.5

Table 3: Impact of the high order moments as regularizers. We report the Recall@K on CUB. k is the number of chosen orders at training time, and n is the order used at testing time to evaluate the performances. $k = n = 1$ is the baseline.

k	1	2		3			4				5					6					
n	1	1	2	1	2	3	1	2	3	4	1	2	3	4	5	1	2	3	4	5	6
R@1	55.9	57.0	53.4	57.6	54.7	50.6	57.9	55.4	52.3	47.6	58.1	55.9	53.1	48.4	43.7	**58.4**	55.7	52.9	47.8	43.9	40.5
R@2	67.6	68.3	65.4	69.9	67.0	63.0	69.5	67.1	65.0	60.2	**70.3**	67.7	65.0	60.8	56.0	69.9	67.6	64.9	59.9	56.0	53.0
R@4	78.3	78.3	75.8	79.1	76.8	73.6	79.6	77.5	75.2	71.0	**79.9**	78.2	75.5	72.8	67.2	79.8	78.0	75.6	70.2	67.2	64.7
R@8	86.4	86.2	84.2	87.0	84.7	82.4	87.1	85.8	83.6	80.2	87.1	85.2	83.9	81.7	78.0	**87.3**	85.6	83.8	79.6	77.5	75.2

Table 4: Impact of the high order moments when all parameters are trained. We report the Recall@K on CUB. k is the number of chosen orders at training time, and n is the order used at testing time. $k = n = 1$ is the baseline.

These results are annotated with a †.

First, we show in the upper part of Table 1 that *HORDE* significantly improves three popular baselines (contrastive loss, triplet loss and binomial deviance). These improvements allow us to claim state of the art results for single model methods on CUB with $\mathbf{58.3}\%$ R@1 (compared to 57.1% R@1 for HTL [5]) and second best for CARS.

We also present ensemble method results in the second part of Table 1. We show that *HORDE* is also a benefit to ensemble methods by improving ABE [11] by 2.7% R@1 on CUB and 7.2% R@1 on CARS. To the best of our knowledge, this allows us to outperform the state of the art methods on both datasets with $\mathbf{62.7}\%$ R@1 on CUB and $\mathbf{86.4}\%$ R@1 on CARS, despite our implementation of ABE underperforming compared to the results reported in [11].

Note that both single models and ensemble ones are further improved by using the high-order moments at testing: +1.1% on CUB and +1.7% on CARS for the single models + *HORDE* and +1.2% on CUB and +1.6% on CARS for ABE + *HORDE*.

Furthermore, we show that *HORDE* generalizes well to large scale datasets by reporting results on SOP and INSHOP in Table 2. *HORDE* improves our baseline binomial deviance by 5.2% R@1 on SOP and 3.1% R@1 on INSHOP. This improvement allows us to claim state of the art results for single model methods on INSHOP with $\mathbf{84.2}\%$ R@1 (compared to 80.9% R@1 for HTL) and second best on SOP with 72.6% R@1 (compared to 74.8% R@1 for HTL). Remark also that *HORDE* outperforms HTL on 3 out of 4 datasets.

We also report some results with the BN-Inception [8]. Our model trained with *HORDE* and contrastive loss leads to similar results compared to the recent MS loss with mining [30] on smaller datasets while on larger datasets we outperform it by 1.9% on SOP and by 0.7% on INSHOP. By using the high-order moments are testing, performances are further increased and outperforms MS loss with mining by 1.1% on CUB and by 2.1% on CARS.

Finally, we show some example queries and their nearest neighbors in Figure 3 on the test split of CUB.

5. Ablation study

In this section, we provide an ablation study on the different contributions of this paper. We perform 3 experiments on the CUB dataset [27]. The first experiment shows the impact of high-order regularization on a standard architecture while the high-order moments are consistently approximated using the Random Maclaurin approximation. The second experiment illustrates the benefit of learning the high-order moments projection matrices. The last experiment confirms the statistical consistency of our cascaded architecture when the parameters are learned.

5.1. Regularization effect

In this section, we assess the regularization impact of *HORDE*. To that aim, we use the baseline detailed in section 4 and we train the architecture with a number of high-order moments varying from 2 to 6. In this first experiment, the computation of the high-order moments does not rely on the cascade computation approach of Equation 6. Instead, the matrices to approximate the high-order moments are untrainable and sampled using the Random Maclaurin method of Equation 4. Remark also that the embedding layers on all high-order moments are not added. We use the binomial deviance loss with the standard parameters [26]. The results are shown in Table 3.

First, we can see that *HORDE* consistently improves the baseline from 1% to 2% in R@1. These results corroborate the insights of our theoretical analysis in section 3 and also provide a quantitative evaluation of the behavior observed in Figure 1 on the retrieval ranking. When consider-

k	1	2		3			4				5					6					
n	1	1	2	1	2	3	1	2	3	4	1	2	3	4	5	1	2	3	4	5	6
R@1	55.9	57.0	53.4	57.9	56.1	54.2	57.6	55.4	54.3	53.0	**58.3**	56.3	56.0	54.7	52.4	57.9	56.6	55.8	55.0	53.9	51.6
R@2	67.6	68.3	65.4	69.4	67.9	66.2	69.3	67.2	66.0	65.2	**70.4**	68.7	68.1	66.9	64.7	69.5	68.8	68.3	67.7	65.2	64.0
R@4	78.3	78.3	75.8	79.2	77.8	76.4	79.5	77.2	77.0	75.8	**80.2**	78.5	78.3	76.9	75.6	79.6	76.6	77.9	77.9	75.3	74.4
R@8	86.4	86.2	84.2	86.6	85.3	84.4	87.1	85.6	84.4	84.1	**87.7**	86.3	86.0	85.4	84.1	87.0	86.4	85.6	84.8	84.0	83.7

Table 5: Impact of the cascaded architecture when all parameters are trained using Algorithm 1. We report the Recall@K on CUB. k is the number of chosen orders at training time, and n is the order used at testing time. $k = n = 1$ is the baseline.

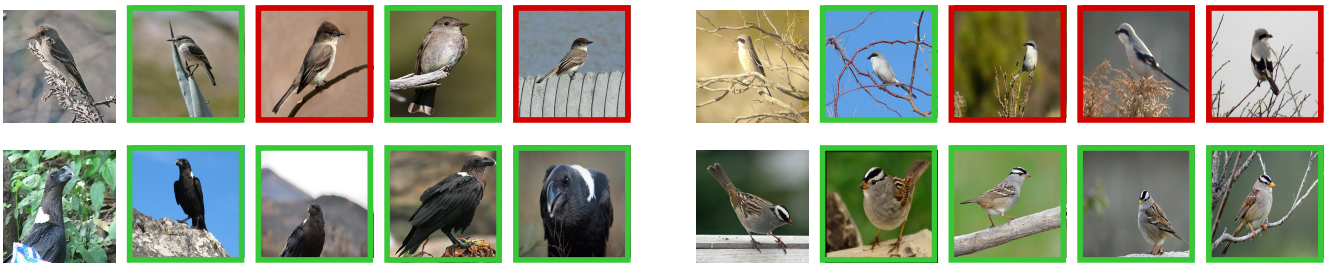


Figure 3: Qualitative results on CUB for *HORDE*. Correct results are highlighted green (incorrect in red).

ing the high-order moments as representations, we observe improved results with respect to the baseline for orders 2 and 3. Note however that the reported high-order results are not comparable to the first order as the similarity measure is computed on the 8192 dimensional representations. While adding orders higher than 2 does not seem interesting in terms of performances, we found that the training process is more stable with 5 or 6 orders than only 2. This is observed in practice by measuring the Recall@K with $K \geq 8$ which tend to vary less between training steps. Moreover on the CUB dataset, while the baseline requires around 6k steps to reach the best results, we usually need 1k steps less to reach higher accuracy with *HORDE*.

5.2. Statistical consistency

To evaluate the impact of estimating only informative high-order moments, we first train the projection matrices and the embeddings but without the cascade architecture and report the results in Table 4.

In this second experiment, we empirically show that such scheme also increases the baseline by at least 1% in R@1. Notably, by focusing on the most informative high-order moment components, *HORDE* further improves the performances of the untrainable *HORDE* from 57.8% to 58.4%. However, the retrieval performances of the high-order representations are heavily degraded compared to Table 3. We interpret these results as an inconsistent estimations of the high-order moments due to overfitting the model. For example, the 6% loss in R@1 for the third-order moment between the first and the second experiments suggests a reduced interest for even higher-order moments.

For the third experiment, we report the results of our cascaded architecture in Table 5. Interestingly, the high-order moments computed from the cascaded architecture perform almost identically to those computed from the untrained method Table 3 but with a smaller dimension. Moreover, we keep the performance improvement of the second experiments of Table 4. This confirms that the proposed cascaded architecture does not overfit its estimations of the high-order moments while still improving the baseline. Finally, this cascaded architecture only produces a small computational overhead during the training compared to the architecture without the cascade.

6. Conclusion

In this paper, we have presented *HORDE*, a new deep metric learning regularization scheme which improves the distinctiveness of the deep features. This regularizer, based on the optimization of the distance between the distributions of the deep features, provides consistent improvements to a wide variety of popular deep metric learning methods. We give theoretical insights that show *HORDE* upper-bounds the Maximum Mean Discrepancy and lower-bounds the Wasserstein distance. The computation of high-order moments is tackled using a trainable Random Maclaurin factorization scheme which is exploited to produce a cascaded architecture with small computation overhead. Finally, *HORDE* achieves very competitive performances on four well known datasets.

Acknowledgements

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

References

[1] Micael Carvalho, Rémi Cadène, David Picard, Laure Soulier, Nicolas Thome, and Matthieu Cord. 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*, 2018. 1

[2] Weihua Chen, Xiaotang Chen, Jianguo Zhang, and Kaiqi Huang. Beyond triplet loss: A deep quadruplet network for person re-identification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2

[3] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005. 2

[4] Yueqi Duan, Wenzhao Zheng, Xudong Lin, Jiwen Lu, and Jie Zhou. Deep adversarial metric learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[5] Weifeng Ge. Deep metric learning with hierarchical triplet loss. In *The European Conference on Computer Vision (ECCV)*, September 2018. 6, 7

[6] Arthur Gretton, Karsten M Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex J Smola. A kernel method for the two-sample-problem. In *Advances in neural information processing systems*, pages 513–520, 2007. 4, 5

[7] Ben Harwood, Vijay Kumar B G, Gustavo Carneiro, Ian Reid, and Tom Drummond. Smart mining for deep metric learning. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 2

[8] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning*, Jul 2015. 1, 7

[9] Hervé Jégou and Ondrej Chum. Negative evidences and co-occurrences in image retrieval: the benefit of PCA and whitening. In *The European Conference on Computer Vision (ECCV)*, Oct. 2012. 3

[10] Purushottam Kar and Harish Karnick. Random feature maps for dot product kernels. In *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, Apr 2012. 3

[11] Wonsik Kim, Bhavya Goyal, Kunal Chawla, Jungmin Lee, and Keunjoo Kwon. Attention-based ensemble for deep metric learning. In *The European Conference on Computer Vision (ECCV)*, September 2018. 2, 6, 7

[12] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Dec 2013. 2, 6

[13] Xudong Lin, Yueqi Duan, Qiyuan Dong, Jiwen Lu, and Jie Zhou. Deep variational metric learning. In *The European Conference on Computer Vision (ECCV)*, September 2018. 2, 3, 6

[14] Hongye Liu, Yonghong Tian, Yaowei Yang, Lu Pang, and Tiejun Huang. Deep relative distance learning: Tell the difference between similar vehicles. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 1

[15] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 1, 2, 6

[16] Yair Movshovitz-Attias, Alexander Toshev, Thomas K. Leung, Sergey Ioffe, and Saurabh Singh. No fuss distance metric learning using proxies. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 2

[17] Hyun Oh Song, Stefanie Jegelka, Vivek Rathod, and Kevin Murphy. Deep metric learning via facility location. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2

[18] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 1, 2, 6

[19] Michael Opitz, Georg Waltner, Horst Possegger, and Horst Bischof. Bier - boosting independent embeddings robustly. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 1, 2, 3, 6

[20] Michael Opitz, Georg Waltner, Horst Possegger, and Horst Bischof. Deep metric learning with BIER: boosting independent embeddings robustly. *IEEE transactions on pattern analysis and machine intelligence*, 2018. 6

[21] Oren Rippel, Manohar Paluri, Piotr Dollar, and Lubomir Bourdev. Metric learning with adaptive density discrimination. *International Conference on Learning Representations (ICLR)*, May 2016. 2

[22] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 1, 2

[23] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in Neural Information Processing Systems 29*, Dec 2016. 1, 2

[24] Bharath K. Sriperumbudur, Arthur Gretton, Kenji Fukumizu, Bernhard Schölkopf, and Gert R.G. Lanckriet. Hilbert space embeddings and metrics on probability measures. *J. Mach. Learn. Res.*, Aug 2010. 5

[25] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 1, 6

[26] Evgeniya Ustinova and Victor Lempitsky. Learning deep embeddings with histogram loss. In *Advances in Neural Information Processing Systems 29*, Dec 2016. 1, 2, 7

[27] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 2, 6, 7

[28] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[29] Jian Wang, Feng Zhou, Shilei Wen, Xiao Liu, and Yuanqing Lin. Deep metric learning with angular loss. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 2, 6

[30] Xun Wang, Xintong Han, Weilin Huang, Dengke Dong, and Matthew R. Scott. Multi-similarity loss with general pair weighting for deep metric learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 6, 7

[31] Jnatas Wehrmann and Rodrigo C. Barros. Bidirectional retrieval made simple. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 1

[32] Xinyi Xu, Yanhua Yang, Cheng Deng, and Feng Zheng. Deep asymmetric metric learning via rich relationship mining. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4076 – 4085, June 2019. 6

[33] Hong Xuan, Richard Souvenir, and Robert Pless. Deep randomized ensembles for metric learning. In *The European Conference on Computer Vision (ECCV)*, September 2018. 2

[34] Baosheng Yu, Tongliang Liu, Mingming Gong, Changxing Ding, and Dacheng Tao. Correcting the triplet selection bias for triplet loss. In *The European Conference on Computer Vision (ECCV)*, September 2018. 2

[35] Yuhui Yuan, Kuiyuan Yang, and Chao Zhang. Hard-aware deeply cascaded embedding. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 2, 6

[36] Wenzhao Zheng, Zhaodong Chen, Jiwen Lu, and Jie Zhou. Hardness-aware deep metric learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 72 – 81, June 2019. 6

[37] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 1, 2

[38] Jiahuan Zhou, Pei Yu, Wei Tang, and Ying Wu. Efficient online local metric adaptation via negative samples for person re-identification. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 1